Variational Quantum Computing: Optimization and Geometry

by

Roeland Wiersema

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Physics

Waterloo, Ontario, Canada, 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Sophia Economou Professor, Department of Physics, Virginia Tech
Supervisor:	Juan Felipe Carrasquilla Álvarez Adjunct Assistant Professor, Department of Physics, University of Waterloo Associate Professor, Department of Physics, ETH Zürich
Co-Supervisor:	Roger Melko Professor, Department of Physics, University of Waterloo Associate Faculty, Perimeter Institute Affiliate Member, Institute for Quantum Computing
Internal Member:	Anton Burkov Professor, Department of Physics, University of Waterloo Affiliate Member, Perimeter Institute
Internal-External Member:	Michael Brannan Associate Professor, Department of Pure Mathematics, University of Waterloo Associate Member, Institute for Quantum Computing
Other Member(s):	Timothy Hsieh Faculty Member, Perimeter Institute Adjunct Faculty, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Roeland Wiersema was the sole author for Chapters 1, 2, 3 and 4. This thesis consists of several manuscripts written for publication. Exceptions to sole authorship of material are as follows:

Chapter 1

The research projects described in sections 2.2 and 2.3 were conceived by professor Carrasquilla, professor Kim and Professor Yuen. Writing, coding and running experiments was done by Roeland Wiersema with contributions from Dr. Zhou and Yvette de Sereville. The research described in section 2.4 was conceived by professor Carrasquilla and professor Aolita. Coding was done largely by Dr. Guerini with contributions from Roeland Wiersema. The manuscript was written by Roeland Wiersema. The relevant citations are:

- **Wiersema**, **Roeland**, Cunlu Zhou, Yvette de Sereville, Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen. Exploring entanglement and optimization within the hamiltonian variational ansatz. *PRX Quantum*, 1(2):020319, 12 2020
- **Wiersema**, **Roeland**, Cunlu Zhou, Juan Felipe Carrasquilla, and Yong Baek Kim. Measurement-induced entanglement phase transitions in variational quantum circuits. *SciPost Phys.*, 14(6):147, 6 2023
- **Wiersema**, **Roeland**, Leonardo Guerini, Juan Felipe Carrasquilla, and Leandro Aolita. Circuit connectivity boosts by quantum-classical-quantum interfaces. *Phys. Rev. Res.*, 4:043221, Dec 2022

Chapter 2

The research projects described in sections 3.2 and 3.3 were conceived by Nathan Killoran and Roeland Wiersema. Writing, coding and running experiments was done by Roeland Wiersema with contributions from professor Carrasquilla, Dr. Wierichs and Dylan Lewis for section 3.3. The relevant citations are:

- **Roeland Wiersema** and Nathan Killoran. Optimizing quantum circuits with Riemannian gradient flow. *Phys. Rev. A*, 107(6):062421, Jun 2023
- **Wiersema**, **Roeland**, Dylan Lewis, David Wierichs, Juan Carrasquilla, and Nathan Killoran. Here comes the SU(N): multivariate quantum gates and gradients. *Quantum*, 8:1275, March 2024

Chapter 3

The project was conceived by Roeland Wiersema and professor Bakalov. Computer calculations and numerical simulations were performed by Roeland Wiersema, with some assistance from professor Kemper. Mathematical theorems were derived by professor Bakalov with the assistance of Dr. Kökcü. The main part of the manuscript was written mostly by Roeland Wiersema, while the Supplemental Materials were written by professor Bakalov. The figures were made by Roeland Wiersema, and the tables by professor Kemper and Dr. Kökcü. Relevant citations:

- **Wiersema**, **Roeland**, Efekan Kökcü, Alexander F Kemper, and Bojko N Bakalov. Classification of dynamical Lie algebras for translation-invariant 2-local spin systems in one dimension. *arXiv:2203.05690*, 9 2023

Abstract

Quantum computing potentially offers unprecedented computational capabilities that transcend the limitations of classical computing paradigms. Despite its conceptual inception over three decades ago, recent years have witnessed remarkable progress in the realization of physical quantum computers, spurring a surge of research activity in the field. Although fault-tolerance devices remain unrealized, modern quantum hardware is getting less noisy, which allows us to investigate quantum algorithms that require only short depth circuits. One particular class of algorithms that falls into this category are variational quantum algorithms, which treat a quantum computer as a black box with tunable parameters that can be optimized via a classical optimization routine. This thesis delves into the realm of variational quantum algorithms and explores their optimization properties, trainability and geometric properties. Through a blend of numerical experiments, geometric insights, and mathematical analysis, it provides a comprehensive exploration of variational quantum algorithms paving the way for future advancements in variational quantum computing.

Acknowledgments

A week after my arrival in Toronto I met Juan for the first time and one of the first things he said to me was something along the lines of: "I'm glad you're here, we're going to do some fun Science." At that point I knew that I had made right choice in moving to Canada from the Netherlands. I want to thank Juan for his excellent supervision during the last four years, which included many great discussions and indeed a lot of fun science. I'm also extremely grateful that he let me pursue internships with a variety of different companies and institutes, which have greatly shaped my academic interests and skills. Finally, I know that the freedom that I have experienced during my PhD is not a given, and I am thankful that I got the opportunity to pursue my own interests, start my own projects (of which there were probably too many) and find my own collaborators. I want to thank Roger for letting me be a part of the PIQUIL at Perimeter, which has felt like a second academic home in addition to the Vector Institute.

I feel like I have been extremely blessed with my collaborators, who in addition to being great scientists, were also really nice people that were very fun to work with. I learned so many things from and always felt like I was in a very wholesome and supportive environment. From talking to many of my peers in the last four years I know that this is not something to take for granted. I want to give a special shoutout to Nathan Killoran. When applying for my internship at Xanadu, I remember telling him that I "wanted to do something with geometry." I am very grateful that we got to do just that in the last couple of years and were able to pursue so many cool research directions in this spirit.

I would like to thank my friends in the Netherlands, Canada and everywhere else in the world for making this PhD thing a little easier. Jelle, I am still working on the Dyson sphere, but it may take one or two postdocs to complete.

Schuyler, je bent een onmisbare steun geweest tijdens mijn PhD, ik vind je lief. Mams, dankzij jou heb ik ondanks de thuissituatie toch mijn eigen weg kunnen volgen. Ik ben je eeuwig dankbaar dat je zo lang voor paps hebt gezorgd en nu eindelijk je eigen weg hebt gevonden. Paps, ik ben heel trots dat jij ondanks alles wat er is gebeurd door bent gegaan, gelukkig bent, en je eigen ding blijft doen.

Translation: Schuyler, Your unwavering support has been invaluable throughout my PhD, I love you. Mom, thanks to you I was able to follow my own path despite everything we have been through as a family. I am eternally grateful that you took care of dad for so long and have now finally found your own way. Dad, I am very proud that despite everything that has happened, you have kept going forward, are happy, and continue to do your own thing.

Dedication

This thesis is dedicated to my favorite guinea pig in the world.

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Statement of Contributions iv

Abstract vi

Acknowledgments vii

Dedication viii

List of Figures xvi

List of Tables xix

List of Abbreviations xx

Mathematical Notation xxi

1 Introduction 1

2 Ground state approximations with Variational Quantum Circuits 5

2.1 Variational Quantum Computing 5

2.1.1 Variational Quantum Eigensolver 7
2.1.2 Gradients of quantum gates 9
2.2 The Hamiltonian Variational Ansatz 12
2.2.1 The ansatz 13
2.2.2 Models 14
2.2.3 Entanglement 18
2.2.4 The ansatz space through the lens of entanglement spectrum 20
2.2.5 Overparameterization in HVA 22
2.2.6 Ameliorated barren plateaus in HVA 27
2.2.7 The entangling power of HVA circuits 28
2.2.8 Conclusion 29
2.3 Measurement-induced entanglement phase transitions in quantum gradients 32
2.3.1 Variational quantum circuits undergoing measurements 34
2.3.2 Measurement–induced entanglement phase transitions 35
2.3.3 Projective gradients and barren plateaus 38
2.3.4 Conclusion 41
2.4 Quantum-Classical-Quantum interfaces 43
2.4.1 Positive Operator Valued Measures 44
2.4.2 Interfaces for hybrid classical-quantum circuits 46
2.4.3 Numerical experiments 50
2.4.4 Random Walk Metropolis-Hastings for negativity minimization . . 50
2.4.5 Simulation of long-range maximal Bell violations 53
2.4.6 The Transverse Field Ising-model circuit 55
2.4.7 Conclusion 56

3 Riemannian geometry in Variational Quantum Computing 61

3.1 Differential Geometry and Lie groups 61

3.1.1 Differentiable Manifolds 61

3.1.2 Tangent spaces 62

3.1.3 Cotangent vectors (one-forms) 63

3.1.4 Riemannian manifolds 63

3.1.5 Riemannian gradients 64

3.1.6 Lie groups 67

3.1.7 SU(N) and its Lie algebra 68

3.1.8 The exponential map and its gradient 69

3.2 Riemmannian Gradient Flows in variational quantum circuits 71

3.2.1 Gradient flows in quantum circuits 72

3.2.2 Gradient flows on Lie groups 72

3.2.3 Exact Riemannian gradient flow in quantum circuits 76

3.2.4 Approximate Riemannian gradient flow in quantum circuits 78

3.2.5 Numerical examples 81

3.2.6 Conclusion 83

3.3 Here comes the SU(N): multivariate quantum gates and gradients 85

3.3.1 $\mathbb{SU}(N)$ gates 86

3.3.2 Obtaining the gradient 87

3.3.3 Comparison with Riemannian gradient flow 94

3.3.4 Comparison with decomposed unitaries 96

3.3.5 Resource estimation 101

3.3.6 Conclusion 104

4 A Lie algebra perspective of Variational Quantum Computing 106
4.1 Lie algebras . . . 106
4.1.1 The Lie algebras $\mathfrak{su}(N)$, $\mathfrak{so}(N)$, and $\mathfrak{sp}(N)$ . . . 108
4.1.2 Dynamical Lie Algebras . . . 110
4.2 Classification of dynamical Lie algebras in one dimension . . . 112
4.2.1 Summary of the main results . . . 114
4.3 Background . . . 117
4.3.1 Preliminaries . . . 117
4.3.2 2-local spin systems in one dimension . . . 118
4.3.3 Growing the dynamical Lie algebras . . . 120
4.4 Method . . . 121
4.4.1 The power sets . . . 121
4.4.2 Symmetries of the power sets . . . 123
4.5 Results . . . 125
4.5.1 Main theorem . . . 125
4.5.2 Sketch of the proof . . . 127
4.5.3 Example: $\mathfrak{a}_9(n)$ . . . 128
4.5.4 Periodic boundary conditions . . . 129
4.5.5 Permutation-invariant subalgebras . . . 131
4.6 Discussion . . . 131
4.6.1 Relevance for variational quantum computing . . . 132
4.6.2 Relevance for quantum control . . . 140
4.6.3 Relevance for spin systems . . . 140
4.7 Conclusion . . . 142

5 Conclusion 145

References 147

APPENDICES **184**

A Quantum computing preliminaries **185**

A.1 Pauli strings and $\mathfrak{su}(2^n)$ 185

A.2 Projective measurements 187

A.3 Informationally Complete POVMs 191

B Additional details for the Hamiltonian Variational Ansatz **193**

B.1 Computational Details 193

B.2 Dynamics of Entanglement Entropy during Optimization 194

C Additional details for measurement-induced entanglement phase transitions in quantum gradients **198**

C.1 Finite-scaling analysis and data collapse 198

C.2 Mutual information 200

C.3 Projective gradients 201

C.4 A practical optimization algorithm with projective measurements 205

C.5 Data collapse of the projective gradients 208

D Additional details for Quantum-Classical-Quantum interfaces **209**

D.1 Interfaces for hybrid classical-quantum circuits with Frames 209

D.1.1 Dual frame decomposition 215

D.2 Finite statistics estimator 215

D.3 Locally Purified Density Operators 216

D.4 Circuit cutting via the Hubbard-Stratonovich transformation 218

E Additional details Riemannian gradient flow **222**

E.1 Dynamical Lie algebra gradient flow 222

F Additional details here comes the SU(N) 224

F.1 The generalized parameter-shift rule . . . 224

F.2 Alternative differentiation of SU(N) gates . . . 226

F.3 The Stochastic parameter-shift rule . . . 226

F.4 Finite differences . . . 227

F.5 Decomposing effective generators for differentiation . . . 229

F.6 Gate speed limit . . . 230

F.7 Proof of Theorem 3.3.1 . . . 232

F.7.1 Special case of SU(2) . . . 232

F.8 Unique spectral gaps of Dynamical Lie Algebras . . . 234

F.8.1 Proof of Theorem 3.3.2 . . . 234

F.8.2 Examples . . . 237

F.9 Errors due to truncation . . . 238

G Additional details DLA classification 241

G.1 Involutions of $\mathfrak{su}(2^n)$ . . . 241

G.2 Statement of Results . . . 244

G.2.1 Subalgebras of $\mathfrak{su}(4)$ up to symmetry . . . 244

G.2.2 List of all 202 subalgebras of $\mathfrak{su}(4)$ . . . 248

G.2.3 Identifying the subalgebras with known spin systems . . . 251

G.2.4 Extending subalgebras of $\mathfrak{su}(4)$ to $\mathfrak{su}(2^n)$ . . . 252

G.2.5 Subalgebras of $\mathfrak{su}(8)$ . . . 253

G.2.6 Subalgebras of $\mathfrak{su}(2^n)$ . . . 255

G.3 Proofs . . . 257

G.3.1 Plan of the proof of Theorem 4.5.1 . . . 257

G.3.2 Inclusions and isomorphisms . . . 260

G.3.3 Frustration graphs . . . 263

G.3.4 Stabilizers, commutants, and centralizers . . . 270

G.3.5 Upper bounds for $\mathfrak{a}_k(n)$ 275
G.3.6 Lower bounds for $\mathfrak{a}_k(n)$ 279
G.3.7 Identifying the Lie algebras $\mathfrak{g}_k(n)^{\theta_k}$ 284
G.3.8 Periodic boundary conditions 292
G.3.9 Permutation-invariant subalgebras 297

List of Figures

2.1 HVA quantum circuits of depth $L = 1$. 16
2.2 Choice of the bipartition of a 1D chain. 19
2.3 Average entanglement spectrum of HVA quantum states. 21
2.4 Change of the entanglement spectrum of the final layer during the optimization. 23
2.5 Overparameterization in HVA. 24
2.6 Ratio of random initializations that converge to the ground state. 25
2.7 The mean iteration time to convergence as a function of depth. 26
2.8 Variance of the gradients of a single Z_1Z_2 term with respect to θ_0 as a function of the number of qubits at initialization. 28
2.9 Infidelities found after optimization for the MHS Hamiltonian. 30
2.10 Quantum circuit undergoing projective measurements. 33
2.11 Schematic depiction of the variational circuits studied. 37
2.12 Data collapse of the average entanglement entropies. 38
2.13 Variance of the projective gradients taken with respect to the first parameter in the circuit. 40
2.14 Schematic depiction of the QCQ interfaces. 45
2.15 Monte Carlo random walk for interface negativity optimization of the ZZ gate used in Section 2.4.6. 52
2.16 Improvement of energy-estimator variance for the 8-qubit TFIM circuit experiment of Figure 2.18b 54
2.17 CHSH violation as a function the number of qubits. 56

2.18 Comparison of QCQ interface simulation with both noisy and noiseless TFIM circuits. 57
2.19 Comparison of double QCQ interface simulation with both noisy and noiseless TFIM circuits. 58
2.20 Comparison of a QCQ interface simulation with both noisy and noiseless circuits for a 20 qubit TFIM circuit. 59

3.1 Schematic depiction of the gradient of a function $E(\boldsymbol{\theta})$ on $\mathbb{R}^2$ 65
3.2 Difference between the Riemannian gradient flow and Euclidean gradient flow. 73
3.3 Schematic depiction of the compatibility condition and subspace restriction. 77
3.4 Numerical examples on two qubits. 80
3.5 Non-zero components of the Riemannian gradient versus the optimization step. 82
3.6 Comparison of Riemannian gradient optimization versus gradient-based VQE for the 4-qubit transverse field Ising-model. 84
3.7 Schematic depiction of our gradient method. 91
3.8 Rewriting the gradient calculation. 91
3.9 Gradients of $E(\boldsymbol{\theta})$ for a single $\mathbb{SU}(2)$ gate and a random single-qubit Hamiltonian, in the limit of infinitely many shots on quantum hardware. 93
3.10 Gradients of $E(\boldsymbol{\theta})$ as in Figure 3.9 but for finitely many shots on quantum hardware. 95
3.11 Comparing the decomposed unitary with the $\mathrm{SU}(N)$ parameterization. . . 99
3.12 Comparison of the update of circuit parameters from various initial parameters acting on the initial state $\rho = |0\rangle\langle 0|$. 100
3.13 Comparison of decomposed gates versus $\mathbb{SU}(N)$ gates in bricklayer circuits for random 10-qubit Hamiltonians and various depths. 102

4.1 Scaling of the DLAs of spin chains with open boundary conditions. 116
4.2 Growing a Lie algebra by adding a site to the chain. 120
4.3 The three types of Lie algebras in the classification. 122
4.4 Examples of brick layer circuits that fall within our classification. 134

4.5 ADAPT-VQE circuit growing heuristic. 135
4.6 Permutation-invariant topology. 136
4.7 Barren plateaus in variational quantum circuits. 137
4.8 Overparameterization of variational quantum circuits. 138

B.1 Dynamics of entanglement entropy at each layer during optimization (TFIM). 195
B.2 Dynamics of entanglement entropy at each layer during optimization (XXZ). 196
B.3 Scaling of the entanglement entropy of the converged state after $L/2$ and L layers. 197

C.1 Quantum mutual information between two qubits A and B separated by a distance r on a chain of length 16. 200
C.2 Data collapse for the projective gradients at $p_c = 0.25$ and $p_c = 0.5$ for the XXZ-HVA and HAE circuits, respectively. 208

D.1 Illustration of the lower bound of Equation (D.17). 219

E.1 Comparison of Riemannian gradient optimization versus gradient-based VQE for the 4-qubit transverse field Ising-model. 223

F.1 Error of the central difference gradients with $\delta = 0.5, 0.75, 1.0$ for the single-qubit example from Figures 3.9 and 3.10. 228

G.1 Frustration graphs for several examples in our classification for $n = 4$. (a) A frustration graph given by a line. (b) A frustration graph consisting of two disjoint lines. (c) A circular frustration graph. 263
G.2 Frustration graphs for $\mathfrak{a}_2(n)$ and $\mathfrak{a}_4(n)$. 267
G.3 Visualization of the frustration graphs of generators of certain Lie algebras. 268

List of Tables

2.1 Overparameterization threshold $\tilde{L}(n)$ for TFIM and XXZ model with different system sizes n. 27

4.1 List of generators of the DLAs in Theorem 4.5.1 and examples of conventional spin models that have the equivalent DLA. 144

F.1 Examples of DLAs and the size of the root spaces. 238

G.1 List of all subalgebras of $\mathfrak{su}(4)$ up to symmetry $S_3 \times \mathbb{Z}_2$. 245

G.2 Conventional spin models corresponding to the dynamical Lie algebras discussed in the main text. 251

G.3 Proofs and where to find them. 257

List of Abbreviations

DLA Dynamical Lie Algebra 3, 106, 145

HEA Hardware Efficient Ansatz 34

HVA Hamiltonian Variational Ansatz 12

MHS modified Haldane-Shastry 13

MP Marchenko-Pastur 19

NISQ Noisy Intermediate-Scale Quantum 2, 43

POVM Positive Operator Valued Measurement 44

QAOA Quantum Approximation Optimization Algorithm 13

QCQ Quantum-Classical-Quantum 43

TFIM Transverse-field Ising Model 13, 119

VQE Variational Quantum Eigensolver 8, 71, 132, 145

Mathematical Notation

Symbol	Meaning
$\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\boldsymbol{x}$	Vector of real numbers
H	Hamiltonian
$A^\dagger$	Hermitian conjugate
$\mathcal{H}$	Finite-dimensional complex Hilbert space
$\lvert\psi\rangle$	Quantum state
$\mathbb{C}\mathrm{P}$	Complex projective space
grad	Riemannian gradient
n	Number of qubits
$\mathcal{M}$	Differentiable Manifold
$\overleftarrow{\prod}$ ($\overrightarrow{\prod}$)	Product ordered from right (left) to left (right)
$T_x\mathcal{M}$	Tangent space at point $x \in \mathcal{M}$
$\mathfrak{g}$, $\mathfrak{su}$, $\mathfrak{a}$	Lie algebra
$\mathcal{P}_n$	The set of n-qubit Pauli strings
span	Real span
$\mathrm{span}_{\mathbb{C}}$	Complex span
$i\,\mathrm{span}$	Real span multiplied with the imaginary unit

Chapter 1

Introduction

Quantum computing has been described as the next frontier of computing in the information age. With the number of transistors that can be fit into a square millimeter reaching physical limits, scientists are looking for alternative methods of compute. Although the founding ideas of quantum computing are already more than three decades old [7], recent years have seen significant advances to build physical quantum computers, which has led to an explosion of research in the field.

One of the big promises of quantum computing is to provide an (almost) exponential [8] or polynomial speedup [9] over classical algorithms using polynomial resources to manipulate and measure a quantum state. Developing these algorithms turns out to be quite challenging, and the few known algorithms that are provably faster than the best known classical algorithms make use of very specific properties of a quantum states. In fact, in the last couple of years we have seen several works that "dequantize" a class of sparse linear algebra quantum algorithms by developing a classical counterpart with similar performance [10]. Nonetheless, the zoo of quantum algorithms keeps expanding as more novel algorithms are found.

A particular exciting area of quantum computing is the field of *quantum simulation* [11], where a quantum computer is used to simulate the dynamics of a physical quantum system [12]. Such simulations could be used to study condensed matter physics, quantum chemistry and high energy physics, which currently require a large amount of computational resources [13]. For most quantum algorithms, one requires a digital quantum computer, which can in principle prepare an arbitrary quantum state and apply any unitary operations given enough resources. However, for the purposes of quantum simulation, an analog quantum device could be enough to study non-trivial physics. In these so-called

analog quantum computers, one is limited to the dynamics of a particular controllable Hamiltonian. Examples of such platforms are D-Wave's quantum annealer [14], which performs a quantum quench of a tunable spin glass and a Rydberg atom array where cold atoms are manipulated in optical tweezers [15]. Although provable speedups are much harder to establish for the purposes of quantum simulation, it is believed that this will be an area where quantum computers will outshine classical methods.

The main issue that prevents us from accessing this nascent field of computing is noise. Unlike classical computers, which are largely unaffected by random errors, quantum computers are constantly affected by different sources of noise. To combat this, computations need to be error-corrected at all times to keep calculations accurate. To achieve this, quantum error correction codes have been developed, which enable the development of logical qubits constructed out of multiple of their noisy counterparts [16]. In the last couple of years, these ideas have been taken from theory into the lab, and we are starting to see the building blocks of a fault-tolerant quantum computer getting developed [17, 18, 19]. Nonetheless, it will likely take many more years for us to develop this holy grail of a fault-tolerant quantum computer.

Even though fault-tolerant devices do not exist yet, attempts have been made in the last couple of years to make use of the noisy devices that are currently available. This is era of quantum computing has been dubbed the Noisy Intermediate-Scale Quantum (NISQ) computing era [20]. On the one hand, it has been shown noisy machines can do something non-trivial, although their usefulness for practical applications seems highly limited at the moment. Additionally, similar to the dequantization of known quantum algorithms, the classical computing community has made great efforts to challenge and debunk claims of quantum supremacy [21, 22, 23]. Nonetheless, a wide variety of NISQ algorithms has been developed that aims to use the limited resources of noisy devices to solve a classically challenging problem.

A particular class of NISQ algorithms treats the quantum computer as a black box, with parameterized operations that can be individually controlled. These operations are then optimized with a gradient-based method in order to minimize a specific cost function. This approach to quantum computing is called *variational quantum computing* [24, 25] which is close in spirit to the deep learning, where one attempts to optimize a black box function over a non-convex cost landscape. Somewhat miraculously, this approach seems to work extremely well given the right neural network architecture. It seems that given enough data, one is typically able to find high quality local minima of the cost function that leads to an accurate model of the data. The hope is that similar behavior will occur in parameterized quantum circuits, even when they can only perform noisy operations. To turn this hope into a reality, we need a detailed analysis of variational quantum algorithms,

both in theory and application to uncover if there is merit to this approach of quantum computing. This is the main point of this thesis and the work contained herein.

We investigate variational quantum algorithms, with a focus on its optimization properties. The goal of this research program is to investigate if variational quantum circuits can be optimized reliably and used to solve non-trivial problems. Although variational quantum algorithms are proposed in the NISQ context, where noise is omnipresent and only shallow algorithms can realistically be performed on current hardware, we almost exclusively considered an idealized setting. In particular, we consider the case where observables are measured exactly, gradients calculated with floating point precision, and we stayed far away from any practical implementations on a real quantum device (with the exception of Section 2.4). This is not out of laziness, but by choice. The focus of this work has been to investigate the power of variational quantum computing under the most favorable conditions possible, since if it does not work under these circumstances, what hope is there that it will work under noisy conditions?

This thesis is divided in three parts, with an introductory chapter for each chapter. The first part (Chapter 2) consists of numerical studies to investigate the power of variational algorithms. We considered the main algorithmic paradigm in the field and apply it to various problems in quantum many-body physics. Within the context of these problems, we sought to understand when the circuit can be optimized efficiently, or in machine learning terminology, is trainable. We connect overparameterization phenomena in Machine Learning to the quantum setting and explore different ways to get around barren plateau problems in the cost landscape.

The second part of this thesis is a more explorative body of work, focused on developing variational algorithms using ideas in Riemannian geometry (Chapter 3). Although standard Riemannian optimization ideas quickly made its way into the field [26], our exploration has been quite distinct in character from others works. The core perspective that we considered is that instead of thinking about training a variational quantum circuit as an optimization problem in some real parameter space, we can think about it as an optimization problem on the special unitary group SU. This approach has been quite fruitful and led to a series of works that has gotten more esoteric over time, culminating in a work that constructs quantum gates from the homogeneous spaces one obtains by taking a quotient of Lie groups.

The third part of this thesis (Chapter 3) is the most mathematical part of this work. Although the author is no mathematician by any stretch of the imagination, the body of research in the first two chapters led to fundamental questions that required a mathematical approach. In particular, a central object called the Dynamical Lie Algebra (DLA) of a

variational quantum circuit is intricately related to its optimization properties, and we will encounter this object in both Chapter 2 and Chapter 3. A DLA is a Lie algebra that is generated by the terms in a Hamiltonian, which subsequently tells us about the Lie group that is explored during the quantum dynamics generated by that Hamiltonian. The term DLA comes from the quantum control literature [27], where one is mainly concerned with the question, "Can I generate the full special unitary group given the Hamiltonian of my system?", since doing so gives one access to universal quantum computing. In the variational quantum computing context, most works have only considered identifying the DLA of particular circuits. The approach in Chapter 3 is fundamentally different from this perspective, since we seek to answer the question: "What is the set of all possible DLAs, and how do we generate them?" In mathematical terms, we have provided a classification of DLAs and subsequently of classes of variational quantum circuits.

Chapter 2

Ground state approximations with Variational Quantum Circuits

2.1 Variational Quantum Computing

To establish notation, we briefly introduce some core concepts in quantum computing [28]. We denote a quantum state with the ket $|\psi\rangle$. Unless stated otherwise, $|\psi\rangle$ is an element of an n-qubit Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ where we set $N = 2^n$. We can think of the ket $|\psi\rangle$ as an element of the $\mathcal{H}$, and the bra $\langle\psi|$ as an element of the dual space $\mathcal{H}^*$ so that $\langle\psi| : \mathcal{H} \to \mathbb{C}$, i.e. a bra is a function that takes in a vector, and produces a (complex) scalar. Throughout this thesis we will assume that $|\langle\psi|\psi\rangle| = 1$, hence a quantum state is normalized with respect to the ℓ_1-norm. Bras and kets can be transformed into each other via complex transposition

$$\langle\psi| = |\psi\rangle^\dagger .$$

The canonical basis for $\mathcal{H}$ is given by a tensor product of the states

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

which gives

$$\mathcal{H} = \text{span}_{\mathbb{C}} \left\{ |b\rangle = \bigotimes_{i=1}^{n} |b_i\rangle , \quad b_i \in \{0,1\}^n \right\}.$$

This basis is called the *computational basis*.

Let $\mathcal{L}(\mathcal{H})$ be the space of linear operators on $\mathcal{H}$. An equivalent description of a quantum system is via a *density matrix* $\rho \in \mathcal{L}(\mathcal{H})$ which is positive, semi-definite and trace one. If the state has rank one, it can be written as

$$\rho = |\psi\rangle\langle\psi| , \tag{2.1}$$

and is called a *pure state*. A *mixed state*, on the other hand, is a convex linear combination of pure states

$$\rho = \sum_{i}^{n} p_i \, |\psi_i\rangle\langle\psi_i| , \tag{2.2}$$

with $\sum_i^n p_i = 1$ and $0 < p_i \leq 1$.

A quantum computer is a physical device that prepares an initial state $|\psi_0\rangle$ and transform it to some output state $|\psi\rangle$ via a linear transformation U,

$$|\psi_f\rangle = U \, |\psi_0\rangle .$$

For this operation to result in a valid quantum state, we need U to be norm-preserving. Such transformations are given to us in the form of unitary operations U, that satisfy $UU^\dagger = I_N$ with I_N the identity matrix of size $N \times N$.

Although, we are free to use any unitary to transform the state, we are typically limited to unitary operations called *quantum gates*, that act locally on a subsystem of the full Hilbert space. Due to the tensor product nature of $\mathcal{H}$, these gates also act as a tensor representation. For example, for a one-qubit gate acting on qubit i, we have

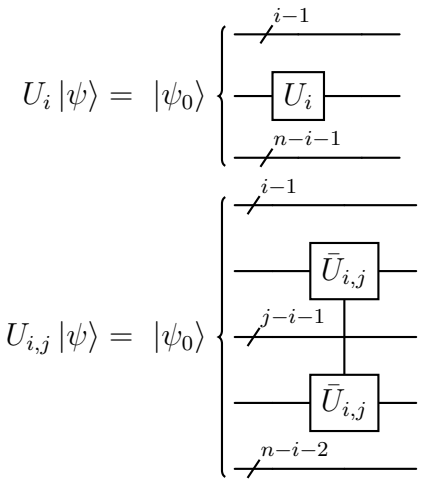
$$U_i \, |\psi\rangle = I_{2^{(i-1)}} \otimes U \otimes I_{2^{n-i-1}} .$$

Similarly, for a two qubit gate acting on $i < j$,

$$U_{i,j} = I_{2^{(i-1)}} \otimes \bar{U}_{i,j} \otimes I_{2^{n-j-2}},$$

where $\bar{U}_{i,j}$ acts only on the subspace spanned by $|b_i\rangle \otimes |b_j\rangle$, $\forall i, j \in \{0, 1\}$ and behaves like the identity elsewhere. This notation quickly becomes too involved when we are applying multiple gates, so the identity tensor products are usually dropped. Additionally, one can

replace the mathematical notation with so-called circuit diagrams,



where the lines indicate local states $|b_i\rangle$ gates are applied from left to right to the state.

A *quantum circuit* consists of multiple gates $\{U_l\}$, with $l = 1, \ldots, L$, acting on different parts of the system.

$$U = \prod_{l=1}^{\overleftarrow{L}} U_l, \tag{2.3}$$

where $\prod^{\overleftarrow{}}$ indicates the ordering from right to left. If the unitaries U_l are chosen from a specific gate set called a *universal* gate set, then any unitary $U \in \mathrm{U}(N)$ can be written as a product of these gates. In the worst case, this will require a number of quantum gates that is exponential in n. In practice however, we can only apply a polynomial number of quantum gates. Designing an efficient (polynomial) quantum circuit to prepare a specific target state is the central goal of quantum computing [28].

2.1.1 Variational Quantum Eigensolver

Instead of designing a quantum circuit from a particular set of fixed quantum gates, we instead parameterize the gates in the circuit. Such a quantum circuit is called a *variational quantum circuit* and corresponds to a product of parameterized unitaries

$$U(\boldsymbol{\theta}) = \prod_{l=1}^{\overleftarrow{L}} U_l(\boldsymbol{\theta}_l), \tag{2.4}$$

with $\boldsymbol{\theta}_l \in \mathbb{R}^{d_l}$ and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L\}$. One can apply the circuit to an initial state

$$U(\boldsymbol{\theta}) \left|\psi_0\right\rangle = \left|\psi(\boldsymbol{\theta})\right\rangle, \tag{2.5}$$

to obtain the variational quantum state $|\psi(\boldsymbol{\theta})\rangle$. Throughout the literature, variational quantum circuits have obtained different names, such as quantum neural network and parameterized quantum circuit.

This *variational quantum state* can be used in a variety of different contexts, such as ground state problems, real-time dynamics and quantum machine learning [24, 25]. A key algorithm in all these approaches is Variational Quantum Eigensolver (VQE) [29], a hybrid classical-quantum algorithm for finding eigenstates of a quantum many-body Hamiltonian H, which is typically a linear combination operators O_i

$$H = \sum_i O_i,$$

where each O_i is Hermitian. Due to physical constraints, the O_i typically act locally, i.e. only on a subsystem of the entire system. We will often make use of the following set of Hermitian operators called the *Pauli matrices*

$$\sigma^0 = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \sigma^1 = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma^2 = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma^3 = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

including the identity matrix I, which form a basis for the real vector space of 2×2 Hermitian matrices. A *Pauli string* is a tensor product of n Pauli matrices of the form

$$a = A^1 \otimes A^2 \otimes \cdots \otimes A^n, \qquad A^j \in \mathcal{P}_1. \tag{2.6}$$

We denote the set of all such Pauli strings by $\mathcal{P}_n := \{I, X, Y, Z\}^{\otimes n}$. The Pauli strings form a basis over the real numbers for any Hermitian operator that of size $\mathbb{C}^N \times \mathbb{C}^N$ (see also Appendix A.1. In this thesis, we will mostly consider spin Hamiltonians that can be described in terms of Pauli strings.

According to the variational principle of quantum mechanics, a variational quantum state $|\psi(\boldsymbol{\theta})\rangle$ provides an upper bound on the ground state energy,

$$E_{\text{ground}} \leq \left\langle\psi(\boldsymbol{\theta})\right| H \left|\psi(\boldsymbol{\theta})\right\rangle = E(\boldsymbol{\theta}), \tag{2.7}$$

Since the variational energy $E(\boldsymbol{\theta})$ is always larger than the ground state energy, we can approximate the ground state by minimizing $E(\boldsymbol{\theta})$ with respect to the parameters $\boldsymbol{\theta}$. The quantum part of VQE is thus the measurement of the energy $E(\boldsymbol{\theta})$ on a variational quantum

state $|\boldsymbol{\theta}\rangle$. The hybrid aspect of VQE is that these observables can be used in a classical optimization routine which can be used to find the optimal parameters $\boldsymbol{\theta}^*$ that minimize the variational energy:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \; E(\boldsymbol{\theta}). \tag{2.8}$$

This can provide us with a good approximation of the true ground state if $U(\boldsymbol{\theta})$ is expressive enough to capture the ground state and the if optimization is successful. An alternative notation for the variational energy is

$$E(\boldsymbol{\theta}) = \mathrm{Tr}\left\{U(\boldsymbol{\theta})\rho_0 U^\dagger(\boldsymbol{\theta})H\right\}, \tag{2.9}$$

where $\rho_0 = |\psi_0\rangle\langle\psi_0|$ is an initial state density matrix. Throughout this thesis we will only consider the choice $|\psi_0\rangle = |0\rangle$, and absorb any pre-processing of the state into $U(\boldsymbol{\theta})$.

To estimate the energy $E(\boldsymbol{\theta})$ during the optimization we estimate the observables that make up the Hamiltonian H. With the notation

$$\langle\psi(\boldsymbol{\theta})|\, O_i \,|\psi(\boldsymbol{\theta})\rangle = \langle O_i\rangle_{\boldsymbol{\theta}},$$

we have

$$\langle\psi(\boldsymbol{\theta})|\, H \,|\psi(\boldsymbol{\theta})\rangle = \sum_i \langle O_i\rangle_{\boldsymbol{\theta}}.$$

The practical implementation of such measurements is discussed in Appendix A.2.

2.1.2 Gradients of quantum gates

To solve the optimization problem in Equation (2.8), one typically uses the gradient descent algorithm, which is a discretization of the gradient flow $\dot{\boldsymbol{\theta}} = \nabla E(\boldsymbol{\theta})$:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \epsilon\, \nabla E(\boldsymbol{\theta}), \tag{2.10}$$

where ϵ is called the learning rate or step size. In practice, there exists a zoo of gradient-based optimizers, with Equation (2.10) being the simplest implementation.

Before we can perform the gradient descent, we need to obtain the gradient of $E(\boldsymbol{\theta})$ with respect to the gate parameters. As it turns out, the gradient of a parameterized

quantum gate has a simple form if our parameterized gates are given by the exponent of an idempotent operator:

$$U_l(\theta_l) = \exp\left\{-i\frac{\theta_l}{2}A\right\} = \cos(\theta_l/2)I + i\sin(\theta_l/2)A, \quad \text{if } A^2 = I. \tag{2.11}$$

which is a generalization of Euler's formula. The gradient with respect to θ can then easily be derived:

$$\partial_{\theta_l} U_l(\theta_l) = -\frac{i}{2}AU(\theta_l).$$

Consider the cost function Equation (2.7) in a quantum circuit with L gates.

$$E(\boldsymbol{\theta}) = \langle 0| \overrightarrow{\prod}_{l=1}^{L} U^\dagger(\theta_l) H \overleftarrow{\prod}_{l=1}^{L} U(\theta_l) |0\rangle .$$

Differentiating with respect to θ_l gives two terms from the product rule

$$\begin{aligned}\partial_{\theta_l} E(\boldsymbol{\theta}) &= \langle 0| U^\dagger(\theta_1) \dots \partial_{\theta_l} U^\dagger(\theta_l) \dots U^\dagger(\theta_L)\, H \overleftarrow{\prod}_{l=1}^{L} U(\theta_l) |0\rangle + \text{c.c.} \\ &= -\frac{i}{2} \langle \psi'| \left[A, U^\dagger(\theta_l)\, \mathcal{C} U(\theta_l)\right] |\psi'\rangle ,\end{aligned}$$

with

$$\begin{aligned}\langle \psi'| = \langle 0| \overleftarrow{\prod}_{l=1}^{l-1} U^\dagger(\theta_l), \quad |\psi'\rangle = \overrightarrow{\prod}_{l=1}^{l-1} U(\theta_l) |0\rangle \\ \mathcal{C} = \overleftarrow{\prod}_{l+1}^{L} U^\dagger(\theta_l) H \overrightarrow{\prod}_{l+1}^{L} U(\theta_l).\end{aligned}$$

As a direct consequence of the Euler relation in Equation (2.11)

$$[A, O] = i\left[U^\dagger\left(\frac{\pi}{2}\right) O\, U\left(\frac{\pi}{2}\right) - U^\dagger\left(-\frac{\pi}{2}\right) O\, U\left(-\frac{\pi}{2}\right)\right],$$

and so we can rewrite the commutator as

$$\left[A, U^\dagger(\theta_l)\, \mathcal{C} U(\theta_l)\right] = \frac{1}{2}\left[U^\dagger\left(\frac{\pi}{2}\right) U^\dagger(\theta_l)\, \mathcal{C} U(\theta_l) U\left(\frac{\pi}{2}\right) - U^\dagger\left(-\frac{\pi}{2}\right) U^\dagger(\theta_l)\, \mathcal{C} U(\theta_l) U\left(-\frac{\pi}{2}\right)\right].$$

Using that $U(\phi)U(\theta) = U(\phi+\theta)$ we find the gradient as the difference between the expectation value of two circuits, where the gate parameter θ_l is shifted by $\frac{\pi}{2}$.

$$\partial_{\theta_l} E(\boldsymbol{\theta}) = \frac{1}{2} \langle \psi'| \left[\overbrace{U^\dagger\left(\theta_l + \frac{\pi}{2}\right) \mathcal{C}\, U\left(\theta_l + \frac{\pi}{2}\right)}^{E^{+,l}(\boldsymbol{\theta})} - \overbrace{U^\dagger\left(\theta_l + \frac{-\pi}{2}\right) \mathcal{C}\, U\left(\theta_l + \frac{-\pi}{2}\right)}^{E^{-,l}(\boldsymbol{\theta})}\right] |\psi'\rangle .$$

So evaluating the gradient of an expectation value with respect to θ_l requires us to evaluate two expectation values

$$\partial_{\theta_l} E(\boldsymbol{\theta}) = \frac{1}{2}\left(E^{+,l}(\boldsymbol{\theta}) - E^{-,l}(\boldsymbol{\theta})\right). \tag{2.12}$$

This rule is known in the literature as the parameter shift rule [30, 31] which is at the core of variational optimization (see Equation (2.12)), since it allows us to construct circuits to calculate the gradients of the cost function in Equation (2.7). We emphasize that the above relation is analytic. Although it seems that this is some finite-shift approximation, the gradient is in fact exact. There are generalizations of the parameter-shift rule to more general settings where we do not require $A^2 = I$, and we will discuss these methods in both Section 2.3 and Section 3.3.

2.2 The Hamiltonian Variational Ansatz

As we described in Section 2.1.1, a VQE algorithm contains three ingredients: A variational quantum circuit ansatz consisting of a set of ordered operators with parameters $\boldsymbol{\theta}$, an energy function given by the expectation value of a local Hamiltonian H composed of local measurements on the variational circuit state and a classical optimizer. A natural first approach is the random quantum circuit ansatz [32, 33, 34], capable of expressing a wide variety of states. However, this was shown to be problematic for gradient-based optimization strategies due to the barren plateau phenomenon [35, 36, 37, 38], which causes the optimization of randomly initialized circuits to get stuck on flat areas in the cost landscape where gradients are exponentially small. These observations suggest that an effective ansatz for VQE requires a circuit that is problem-specific, such that the optimization landscape of the problem is not hindered by barren plateaus. For quantum many-body problems, Ref. [39] suggests a novel variational circuit that is now called the Hamiltonian Variational Ansatz (HVA). While there is no rigorous proof that HVA will be an effective ansatz, recent work has demonstrated that HVA is rather effective for several one- and two-dimensional quantum many-body models [40, 41]. It is thus an intriguing question to further understand the empirically observed effectiveness of HVA.

For the purpose of understanding the effectiveness of such ansätze, it is useful to note that quantum entanglement provides a window into the capabilities of several families of numerical techniques and algorithms aimed at understanding the properties of quantum many-body states, as well as helps us delineate the boundary between quantum states that can be simulated classically and those which call for quantum simulators and quantum computers for their accurate description. For instance, for a one-dimensional (1D) gapped local Hamiltonian, the entanglement entropy of the ground state obeys an area law, i.e., the entanglement entropy grows proportional to the boundary area of the system instead of the system size [42]. This remarkable result allows us to combat the exponential scaling of the Hilbert space, since this area law provides evidence that the relevant physics of a system only takes place in a restricted part of the full state space. These observations have inspired a variety of variational numerical methods, most notably, Tensor Network approaches such as Matrix Product State (MPS), Multiscale Entanglement Renormalization and Projected Entangled Pair States [43], but also deep learning inspired variational approaches, which have been successful at representing quantum many-body states [44, 45, 46, 47].

In this section we study various entanglement properties of HVA and present several results on the favorable features of HVA that shed light on the underlying reasons for its effectiveness for solving natural many-body problems. Our findings suggest that HVA is highly expressive but yet structured enough to allow for efficient optimization.

Through the study of two prototypical models in condensed matter physics, namely the 1D Transverse-field Ising Model (TFIM) and XXZ models, we find that entanglement entropy and entanglement spectrum can shed light onto the initialization and optimization properties of HVA in the context of the VQE algorithm. Whereas HVA provides a restricted and effective state space for the TFIM which yields ground state approximations largely insensitive to the circuit initialization, the 1D XXZ model ansatz requires a careful parameter initialization for its successful optimization. Through the study of the dynamics of entanglement spectrum during the optimization of the XXZ model we find that initializing the HVA near the identity operator enables a restricted and effective subspace during optimization that yields accurate approximations to the ground state with fast convergence. Furthermore, we show evidence that the gradient vanishing problem in HVA, especially if the HVA is initialized near the identity operator, is mild or entirely absent in comparison to the random circuit ansatz, where barren plateaus in the energy landscape cause gradients to decay exponentially with increasing system size. We also explore the over-parameterization phenomena in HVA and observe a "computational phase transition" between an under-parameterized and over-parameterized regime where the optimization landscape of HVA crosses over to a regime with faster convergence and absence of low-quality solutions. Lastly, as a demonstration of the entangling power and effectiveness of HVA, we study a modified Haldane-Shastry (MHS) Hamiltonian which has long-range interactions and a power-law scaling entanglement entropy [48]. We observe that HVA can find approximations to the ground state of the MHS Hamiltonian reaching fidelities $> 99\%$ for system sizes $n = 4, 8, 12, 16$ and circuit depths $L = n$. Our findings point to important features of HVA that will lead to a deeper understanding of its effectiveness, and point the way to developing more sophisticated ansätze for other many-body problems, as well as more informed optimization strategies. Moreover,we establish a substantial connection between quantum entanglement and the efficacy of HVA and show how entanglement properties such as the entanglement spectrum can be used to study variational quantum circuits. Furthermore, the surprising phenomenon of over-parameterization in HVA signals a nontrivial connection with deep neural networks which merits further investigation.

2.2.1 The ansatz

As with other variational methods for approximating the ground state, a key ingredient to the success of the method is finding a good parameterization scheme of the wave function. Ideally, the manifold of states parameterized by the ansatz of choice contains the ground state of interest, and this ground state can be reached using a numerical optimization. The HVA [39] is a quantum circuit ansatz as defined in Equation (2.4) inspired by the Quantum

Approximation Optimization Algorithm (QAOA) [49] and adiabatic computation [50]. Instead of using only two (non-commuting) operators as in QAOA, HVA uses more terms of the Hamiltonian. More specifically,

$$H = \sum_s H_s, \tag{2.13}$$

where we assume that each pair of H_s and $H_{s'}$ do not commute, i.e., $[H_s, H_{s'}] \neq 0$. A depth-L HVA is given by

$$U(\boldsymbol{\theta}, \{H_s\}) = \prod_{l=1}^{\overleftarrow{L}} \left(\prod_s \exp\{-i\theta_{s,l} H_s\} \right). \tag{2.14}$$

For the initial state $|\psi_0\rangle$, we take the ground state of one of the terms in Equation (2.13), i.e. H_{s_0}. When ordering the unitaries, we make sure that H_{s_0} is not the first H_s acting on $|\psi_0\rangle$. The order of the other unitaries is ambiguous, as long as we make sure that subsequent layers do not commute, since otherwise they can be absorbed into a single layer which leads to a redundancy of parameters. Note that due to the periodicity of the complex exponent, we can restrict the parameters to $[0, 2\pi]$, although in the case of certain symmetries, this restriction can be made tighter without losing expressive power [40]. Since these circuits are model specific, the properties of the circuit can vary per problem.

Our goal is to find the ground state of H, which we try to achieve by performing a VQE optimization (see Section 2.1.1 for more details):

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \; E(\boldsymbol{\theta}),$$

where $E(\boldsymbol{\theta})$ is the variational energy

$$E(\boldsymbol{\theta}) = \langle \psi(\boldsymbol{\theta}) | \, H \, | \psi(\boldsymbol{\theta}) \rangle \, .$$

2.2.2 Models

Transverse Field Ising-Model

The TFIM is a paradigmatic model for studies of quantum magnetism. The Hamiltonian for the one-dimensional chain is given by:

$$H_{\text{TFIM}} = -\sum_{i=1}^{n} (Z_i Z_{i+1} + g X_i) = H_{zz} + g H_x, \tag{2.15}$$

with $H_{zz} = -\sum_{i=1}^{n} Z_i Z_{i+1}$ and $H_x = -\sum_{i=1}^{n} X_i$ where we assume $g > 0$ and use periodic boundary conditions $Z_{n+1} \equiv Z_1$. The matrices Z_i and X_i correspond to Pauli matrices $X, Z \in \mathcal{P}_1$ tensored with the identity (see Equation (A.2)). The Hamiltonian has a $\mathbb{Z}_2$ symmetry, so it is invariant under the operation of flipping all spins.

For $g < 1$, the system is in a ferromagnetic phase where the Hamiltonian favors spin alignment along the z direction. For $g > 1$ the system transitions to a disordered paramagnetic phase. In the limit that $g \to \infty$, the X term dominates the Hamiltonian, and the ground state becomes $|+\rangle^{\otimes n}$. At $g = 1$ there is a critical point, and the system becomes gapless in the thermodynamic limit.

A depth-L HVA circuit for the TFIM corresponds to

$$U_{\mathrm{TFIM}}(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \prod_{l=1}^{\overleftarrow{L}} \exp\left\{-i\frac{\gamma_l}{2} H_x\right\} \exp\left\{-i\frac{\beta_l}{2} H_{zz}\right\}. \tag{2.16}$$

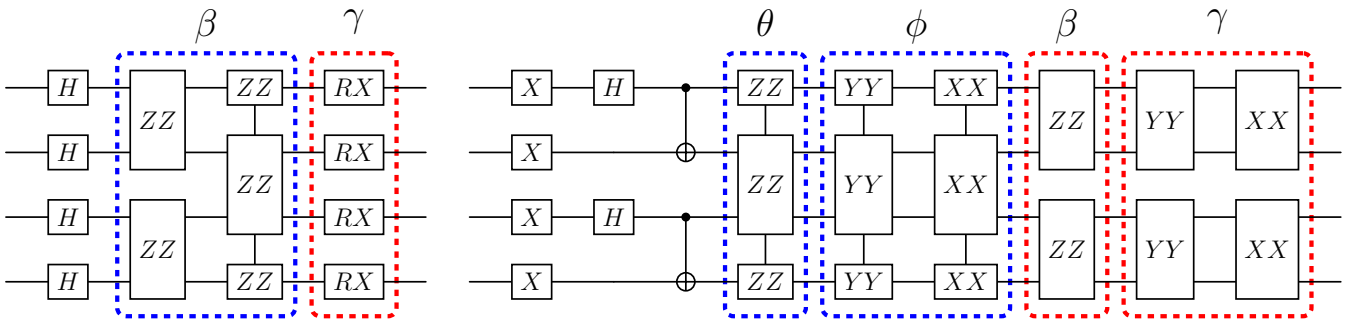
Hence for a depth-L circuit, we have $2L$ parameters. Figure 2.1a illustrates the corresponding quantum circuit for $n = 4$ and $L = 1$. Note that we choose $|\psi_0\rangle$ in Equation (2.14) to be the ground state of $H_x = -\sum_{i=1}^{n} X_i$, i.e., $|\psi_0\rangle = |+\rangle^{\otimes n}$. The HVA circuit of Equation (2.16) is the same as the QAOA ansatz used in [49] for solving the MaxCut problem. By using the Jordan-Wigner transformation, it was shown that the ground state can be represented accurately with a depth $L = n/2$ circuit for the case that $g = 0$ [51]. For the case that $g \neq 0$, there is only numerical evidence to support this claim [40, 52].

XXZ-model

Another prototypical model for studying quantum magnetism is the XXZ model. For the 1D XXZ model, the Hamiltonian is given by

$$\begin{aligned} H_{\mathrm{XXZ}} &= \sum_{i=1}^{n} \left(X_i X_{i+1} + Y_i Y_{i+1} + \Delta Z_i Z_{i+1}\right) \\ &= H_{xx} + H_{yy} + \Delta H_{zz}, \end{aligned} \tag{2.17}$$

with $H_{xx} = \sum_{i=1}^{N} X_i X_{i+1}$, $H_{yy} = \sum_{i=1}^{N} Y_i Y_{i+1}$ and $H_{zz} = \sum_{i=1}^{N} Z_i Z_{i+1}$. Here we also use $Y \in \mathcal{P}_1$ and use Y_i to describe the Pauli Y matrix tensored with the identity. Again we assume periodic boundary conditions. The parameter Δ controls the spin anisotropy in the model. For $\Delta = 1$, this model has a SU(2) symmetry, which implies that the Hamiltonian



(a) TFIM (b) XXZ

Figure 2.1: **HVA quantum circuits of depth $L = 1$.** (a) TFIM circuit. The first layer of Hadamard gates, represented by H, are used to construct the initial $|+\rangle$ state. The ZZ gates are 2-local qubit rotation gates of the form $\mathrm{ZZ} = \exp\{i\beta_l/2\, Z_i Z_j\}$. The RX gates are single qubit rotation gates $\mathrm{RX} = \exp\{i\gamma_l/2\, X_i\}$. (b) The XXZ model circuit. Here, the X gates are given by $\mathrm{X} = X_i$. Together with a single Hadamard gate and a CNOT on even links, we prepare the $|\Psi^-\rangle$ Bell state. The 2-local qubit rotations are all of the form $\mathrm{AA} = \exp\{-ix/2\, \sigma_i^a \sigma_j^a\}$, with $x = \theta, \phi, \beta, \gamma$ depending on whether the links are even or odd and ZZ or XX, YY.

commutes with the generators of SU(2) acting on the full Hilbert space. The generators for this N-dimensional representation of SU(2) is the set

$$\mathcal{A} = \left\{ P_x := \sum_i^n X_i, P_y = \sum_i^n Y_i, P_z = \sum_i^n Z_i \right\},$$

for which it is easy to verify that we have $[P_\alpha, P_\beta] = 2i\epsilon_{\alpha\beta\gamma} P_\gamma$ with $\alpha, \beta, \gamma \in \{x, y, z\}$, which are the commutation relations of SU(2). It is also easy to confirm that

$$[H, P_\alpha] = 0,$$

which implies that the eigenstates of H are also eigenstates of the symmetry SU(2). For $\Delta \neq 1$, this symmetry gets reduced to a $U(1) \times \mathbb{Z}_2$ symmetry. For $1 < |\Delta|$ the system is in the XY quasi-long-range ordered state and becomes gapless in the thermodynamic limit. At $|\Delta| = 1$ there is a phase transition to the Néel ordered state. This model can be solved exactly using the Bethe-ansatz for $n \to \infty$ [53].

Inspired by [40], we decompose the 1D chain into even and odd links and separate the

Hamiltonian into two parts,

$$H^{\text{even}} = H^{\text{even}}_{xx} + H^{\text{even}}_{yy} + H^{\text{even}}_{zz}$$
$$H^{\text{odd}} = H^{\text{odd}}_{xx} + H^{\text{odd}}_{yy} + H^{\text{odd}}_{zz},$$

where the indices only run over non-overlapping bonds:

$$H^{\text{even}}_{\alpha\alpha} = \sum_{i=1}^{n/2} \sigma^{\alpha}_{2i-1}\sigma^{\alpha}_{2i} \quad \text{and} \quad H^{\text{odd}}_{\alpha\alpha} = \sum_{i=1}^{n/2} \sigma^{\alpha}_{2i}\sigma^{\alpha}_{2i+1},$$

where $\sigma^{\alpha} = X, Y, Z$ for $\alpha = x, y, z$, respectively. Our numerical experiments indicate that separately parameterizing these bonds gives better performance when studying the anisotropic system $\Delta \neq 1$. Additionally, we parameterize H_{xx}, H_{yy} and H_{zz} terms with their own respective parameter. The reason for this is that for $\Delta \neq 1$ the anisotropy in the model cannot be accounted for by a single parameter. A depth-L HVA circuit for the XXZ model corresponds to

$$U_{\text{XXZ}}(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \prod_{l=1}^{L} G(\theta_l, H^{\text{odd}}_{zz}) G(\phi_l, H^{\text{odd}}_{xx}) G(\phi_l, H^{\text{odd}}_{yy})$$
$$G(\beta_l, H^{\text{even}}_{zz}) G(\gamma_l, H^{\text{even}}_{xx}) G(\gamma_l, H^{\text{even}}_{yy}), \tag{2.18}$$

where

$$G(x, H) = \exp\Big\{-i\frac{x}{2}H\Big\}.$$

Hence, for a depth-L circuit, we have $4p$ parameters. Figure 2.1b illustrates a quantum circuit for $n = 4$ and $L = 1$. We choose the initial state $|\psi_0\rangle$ in Equation (2.14) to be the ground state of H^{even}, i.e., $|\psi_0\rangle = \bigotimes_{i=1}^{n/2} \frac{1}{\sqrt{2}} (|01\rangle - |10\rangle)_{2i-1,2i} = \bigotimes_{i=1}^{n/2} |\Psi^-\rangle$, which is in the correct symmetry sector [54]. It was shown in [40] that the Heisenberg chain (i.e., $\Delta = 1$) can be solved accurately with HVA with $L = n/2$. Note that for the case of $\Delta = 1$, one can use a single parameter for $H_{xx} + H_{yy} + H_{zz}$.

We consider the problem of approximating the ground state at the critical points $g = 1$ and $\Delta = 1$ for the TFIM and XXZ model respectively since their particular entanglement scaling properties makes them harder to approximate with classical methods [55], such as the Density Matrix Renormalization Group. Due to the criticality of the aforementioned systems at these order values, the energy spectrum becomes gapless in the thermodynamic limit and hence there is a logarithmic correction of $S \propto \log n$ to the area law of entanglement entropy. A matrix product state with bond dimension D bounds the entanglement of the state to $S \leq 2 \log D$, so the necessary bond dimension to express the ground state grows polynomially in a Density Matrix Renormalization Group calculation [55].

Performance Metrics

We use the *fidelity* $\mathcal{F}$ between the VQE optimized state $|\psi(\boldsymbol{\theta}^*)\rangle$ and the true ground state $|\psi_{\text{ground}}\rangle$ obtained from exact diagonalization:

$$\mathcal{F} = |\langle\psi(\boldsymbol{\theta}^*)|\psi_{\text{ground}}\rangle|^2.$$

Note that for the models studied in here, $|\psi_{\text{ground}}\rangle$ is always non-degenerate. If the square root of the fidelity is $> 99.9\%$, we assume that we have successfully found the ground state. When assessing the quality of an optimized HVA circuit, the fidelity is a strong indicator of the success for solving the ground state problem, since the infidelity upper bounds the difference between the ground state and variational expectation value of any observable. Let $1 - \mathcal{F} < \epsilon$

$$\left|\langle O\rangle_{\text{ground}} - \langle O\rangle_{\boldsymbol{\theta}}\right| \leq 2c\sqrt{\epsilon(1-\epsilon)} + \epsilon,$$

where c is the operator norm of O [56] and $\langle O\rangle_{\text{ground}} - \langle O\rangle_{\boldsymbol{\theta}}$ is the difference in expectation value between the ground state and variational state.

2.2.3 Entanglement

In the context of quantum many-body physics, quantum correlations play a central role in our current understanding of the equilibrium and out-of-equilibrium properties of several systems in condensed matter. The source of these correlations is inherently non-local, and can be traced back to the presence of entanglement in the quantum state. In this section we introduce several commonly used entanglement quantities in quantum many-body physics.

In classical systems, one uses entropy to quantify our lack of knowledge of the state of the system due to thermal fluctuations. However, for a quantum system at zero temperature, the entropy of a subsystem has a different origin: entanglement. To quantify it, we use the bipartite *entanglement entropy* [42], which is defined as the *von Neumann entropy* of the reduced density matrix ρ_A. To obtain this reduced density matrix, we divide the system into two subsystems A and B and trace out subsystem B,

$$\rho_A(|\psi\rangle) = \text{Tr}_B(|\psi\rangle\langle\psi|), \tag{2.19}$$

where $|\psi\rangle$ is a pure state. For example, for an 8-spin model on a ring, a typical bipartition is given in Figure 2.2.

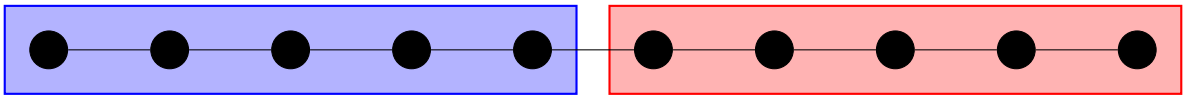


Figure 2.2: **Choice of the bipartition of a 1D chain.** Division of the full system into two subsystems A (blue) and B (red) on a one-dimensional chain.

The von Neumann entropy generalizes the concept of Shannon entropy to quantum states, and is given by

$$S(\rho) := -\mathrm{Tr}(\rho \log \rho). \tag{2.20}$$

Since a bipartite quantum state can always be rewritten using the Schmidt decomposition,

$$|\psi\rangle = \sum_{k=0}^{K} e^{-\frac{1}{2}\xi_k} \left|\psi_A^k\right\rangle \otimes \left|\psi_B^k\right\rangle, \tag{2.21}$$

with $\left\langle \psi_A^k | \psi_A^m \right\rangle = \left\langle \psi_B^k | \psi_B^m \right\rangle = \delta_{km}$ and K the size of the smallest subsystem, the von Neumann entropy of ρ_A reduces to [57]

$$S(\rho_A) = \sum_{k=0}^{K} \xi_k \exp\{-\xi_k\}, \tag{2.22}$$

In recent years, the importance of entanglement in condensed matter physics has been elucidated in several systems through the study of the scaling behavior of the entanglement entropy, which has enabled the identification and characterization of exotic phases of matter such as topological quantum states [58] and quantum spin liquids [59, 60].

Fully characterizing the entanglement properties of a system cannot be done by looking solely at the entanglement entropy [57, 61, 62]. The so-called *entanglement spectrum* has a much richer structure, and has been used to study many-body localization [61], observable thermalization [63], irreversibility in quantum circuits [62], and preparation of ground states of non-integrable quantum models [64]. In addition, the entanglement spectrum has been used to study the properties of variational methods such as the Restricted Boltzmann Machine [48]. The entanglement spectrum is defined as the eigenvalue spectrum of the *entanglement Hamiltonian*

$$H_{\mathrm{ent}}(\rho) = -\log \rho. \tag{2.23}$$

From Equation (2.21) it follows directly that this Hamiltonian has eigenvalues ξ_k. For random quantum states distributed according to the Haar measure, the entanglement spectrum follows the Marchenko-Pastur (MP) distribution [65, 66]. This distribution describes

the asymptotic average density of eigenvalues of Wishart matrices, i.e., matrices of the form XX^* where X be $m \times n$ random matrices.

Finally, the *Page entropy* [67] describes the average entanglement entropy over randomly drawn pure states in the entire Hilbert space, and is given by

$$S_{Page}(n_A, n_B) = -\frac{n_A - 1}{2n_B} + \sum_{k=n_B+1}^{n_A n_B} \frac{1}{k} \approx \log(n_A) - \frac{n_A}{2n_B}, \tag{2.24}$$

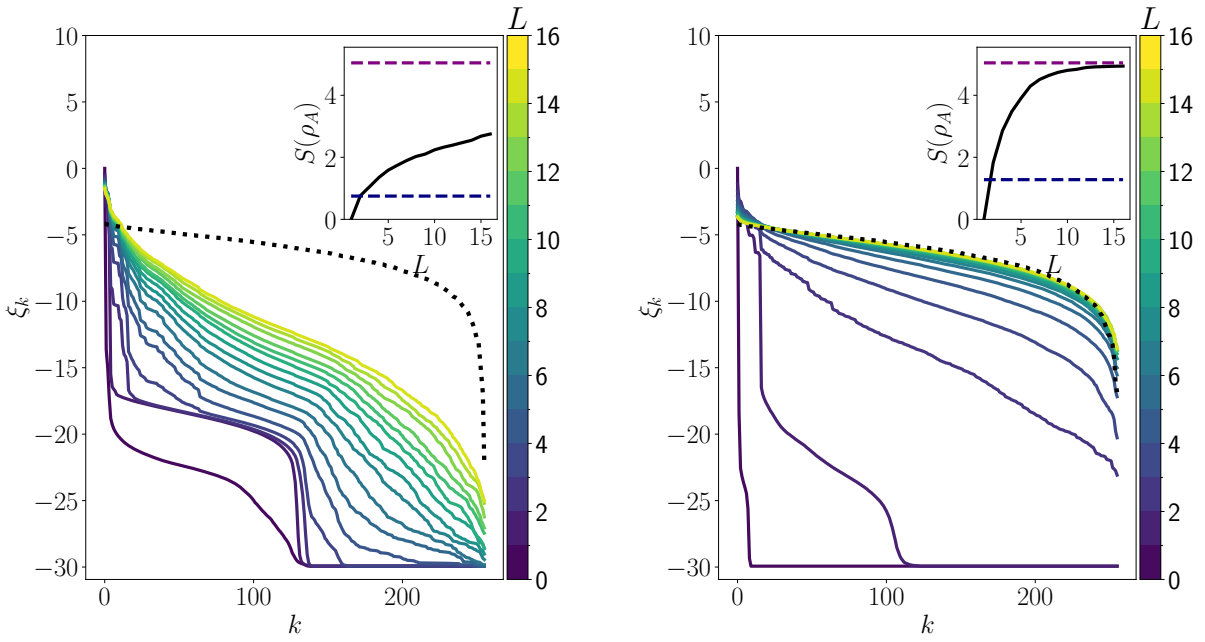
where n_A and n_B are the dimensions of subsystem A and B, respectively.

2.2.4 The ansatz space through the lens of entanglement spectrum

The effectiveness of a VQE optimization is determined by two factors. First, one requires an expressive enough *ansatz space* that contains the ground state. Within the context of HVA, the ansatz space of a specific model H and depth L refers to the set of all possible quantum states that can be reached by applying a depth-L HVA circuit corresponding to H to a fixed initial state $|\psi_0\rangle$ which depends on the model. Secondly, the non-convex cost landscape induced by the variational energy of Equation (2.7) must be favorable, in the sense that the optimization does not get stuck in local minima and can reliably reach the ground state.

Here, we investigate the properties of the ansatz space by examining the entanglement spectra of HVA quantum states generated with random parameters sampled uniformly in the range $[0, \pi]$ for the TFIM and $[0, 2\pi]$ for the XXZ model. For each model, we sample 5000 sets of parameters and calculate the entanglement spectrum of the resulting state. If the spectrum of the sampled states follows a distribution close to the Marchenko-Pastur distribution, a random HVA state has entanglement spectrum that resembles that of a Haar random state. On the contrary, a distribution far away from the MP distribution indicates a restricted manifold of states that has a non-random structure. We hypothesize that the shape of the average entanglement spectrum can give insight into the performance of the VQE optimization by revealing the structure of the ansatz space.

Figure 2.3 shows the average entanglement spectrum for a state in the ansatz space of circuits with depths ranging from $1, 2, \ldots, n$ for the 16-qubit TFIM and XXZ models. From the insets we see that both ansätze have enough entangling power to express the ground state, even for low depth circuits. For the TFIM with 16 qubits (Figure 2.3a), we see that for all L the HVA spectrum is further away from the MP distribution, and the



(a) TFIM

(b) XXZ model

Figure 2.3: **Average entanglement spectrum of HVA quantum states.** We show layer $L = 1$ (bottom line in purple) to $L = n$ (top line in yellow) over 5000 random parameter initializations. ξ_k denotes the k-th eigenvalue of H_{ent}. The eigenvalues are arranged in descending order and cut off at $\xi_k = -30$. The black lines in the insets show how close the average entanglement entropy is to the Page-entropy (purple dashed line) as a function of increasing circuit depth. The lower blue dashed line in the inset indicates the entanglement entropy of the ground state. We see that the average HVA state is more entangled than the ground states of interest.

HVA space corresponding to the TFIM appears to be a manifold of states with restricted entanglement structure. In contrast, for the XXZ model, we see that the average spectra gets closer to the MP distribution as L increases. This suggests that the HVA space for the XXZ model is not as restricted as for the TFIM. This can be understood directly by looking at the circuit complexity, which for the XXZ model contains more gates and parameters per layer. However, this is necessary because the XXZ model is inherently a much richer model in terms of physics, and it may be necessary for HVA space to accommodate more variety of states.

We now turn to examining the entanglement features of the XXZ model HVA states explored during optimization. For the variational minimization of Equation (2.7) we use a gradient descent algorithm (see Appendix B.1 for details). Since the cost function is non-convex, the quality of the solution will vary significantly between different starting points in parameter space. We compare the following initialization strategies:

1. A completely random-state initialization, where all parameters are sampled as $\theta_l \sim \mathcal{U}(0, 2\pi)$.

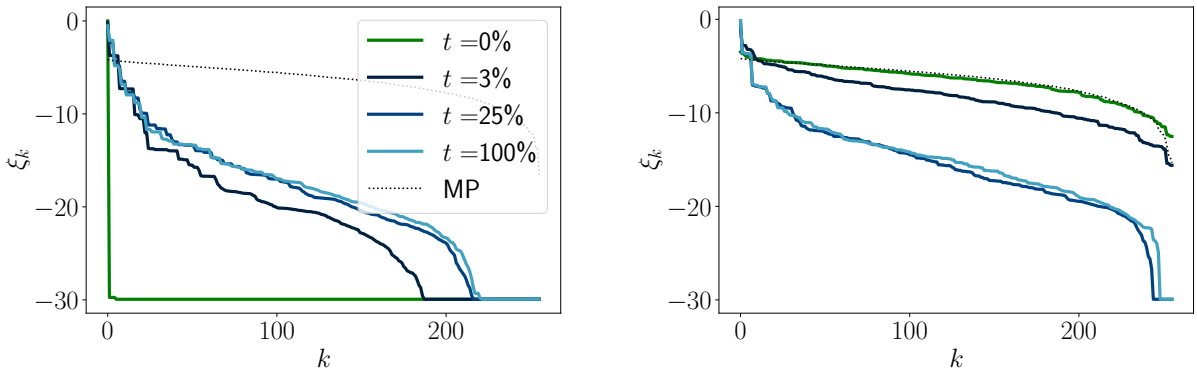
2. An identity initialization. We set all parameters equal to π, so that our circuit is equal to the identity circuit and a global phase i.

Our approach of starting close to the identity is similar to the block identity initialization strategy discussed in [68], however, we study a simpler version by setting all parameters equal to π. For both parameter initializations, we extract the final layer state from the circuit at multiple times during the optimization and calculate its entanglement spectrum with Equation (2.23). Not surprisingly, our experiments indicate that a random start is prone to getting stuck in a local minimum, due to our local optimization strategies combined with a non-convex energy landscape.

To study this finding in more detail, we study the dynamics of the entanglement spectrum for different initialization strategies. In Figure 2.4a we see that an identity state initialization stays far away from the MP distribution at all times, indicating that we are accessing a highly structured restricted subspace of the full HVA space. Additionally, this initialization reaches a state with $\mathcal{F} = 99.9\%$. On the contrary, the random state initialization in Figure 2.4b starts close to the MP distribution and then moves to a more structured, local minima with $\mathcal{F} = 70\%$. We conclude that even though the shape of the entanglement spectrum from Figure 2.3b indicates a possible large unstructured ansatz space, a local optimization is still capable of finding the ground state if we choose a suitable parameter initialization. We further investigate the qualitative properties of the optimization dynamics in Appendix B.2. In the next section, we will see that the disadvantage of starting at a bad initial point can be overcome by making the circuit sufficiently deep, a process known as *overparameterization*.

2.2.5 Overparameterization in HVA

Overparameterization is a phenomenon in certain types of non-convex optimization problems. For an overparameterized model, the optimization landscape becomes dramatically



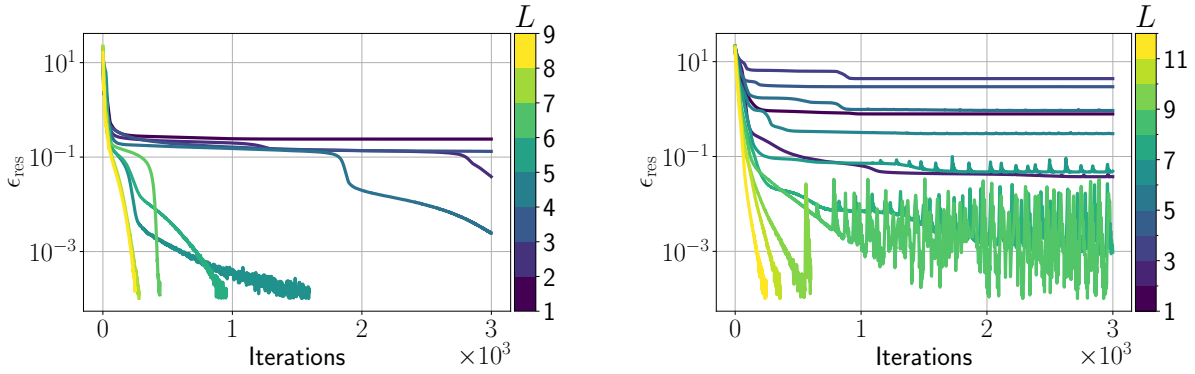
(a) (b)

Figure 2.4: **Change of the entanglement spectrum of the final layer during the optimization.** Both figures are for a 16 qubit XXZ model with a depth $L = n/2$ circuit. The times are percentages of the total optimization time. Figure (a) correspond to a converged state of fidelity, whereas figure (b) corresponds to a $\approx 70\%$ fidelity state. (a) The identity state initialization remains far away from the MP distribution at all times during the optimization and convergence to state with $> 99.9\%$ with the ground state. Since this initialization strategy starts with the identity circuit, we find the $t = 0\%$ state to be a product state, as indicated by the single eigenvalue. (b) The random initialization starts close to the MP distribution and converges to a local minimum with $\approx 70\%$ fidelity.

better (e.g., almost trap free or almost-convex) as the number of parameters reaches some threshold. In most cases the rate of convergence also becomes better, sometimes even exponentially faster after passing this threshold.

Overparameterization has been studied extensively in the classical deep neural network literature [69, 70]. For example, in [69] it was shown that under certain mild assumptions, the optimization landscape of a deep neural network is almost-convex in a large neighborhood of a random starting point. As a consequence the stochastic gradient descent algorithm can almost always find an accurate solution.

For VQE algorithms it is clear that growing the ansatz allows us to cover more of the Hilbert space, which could lead to finding better minima in the cost landscape. However, it is not clear if this minimum can be found consistently due to the non-convexity of the energy landscape. Hence, a deeper understanding of the energy landscape with increasing



(a) TFIM

(b) XXZ model

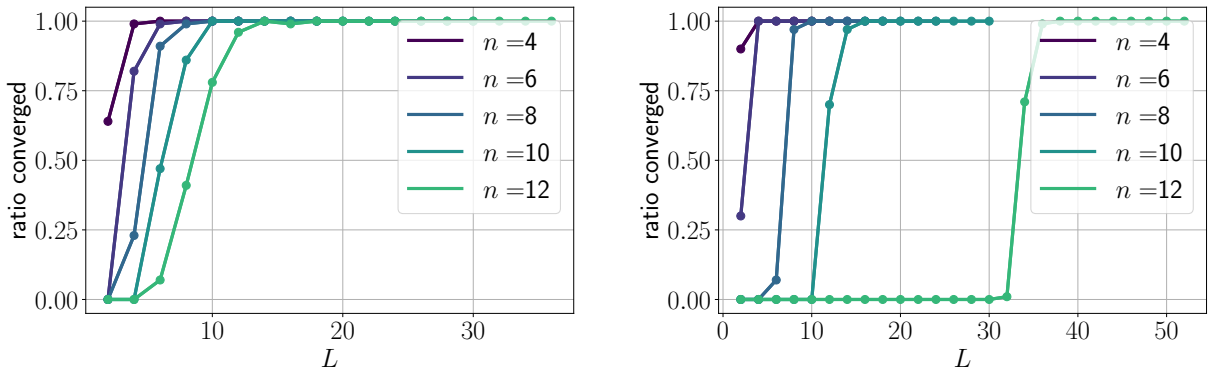
Figure 2.5: **Overparameterization in HVA.** Each line corresponds the VQE optimization at depth L that took the most iterations to converge out of 100 random initializations. Both figures are for $n = 12$ qubits. The rapid oscillations in figure (b) are artifacts of the Adam optimizer and are less severe as the circuit depth increases. Due to our stopping criterion, we know that if the number of iterations is smaller than 3000, then $\epsilon_{\text{res}} \leq 10^{-4}$ and so the model did converge to a good ground state approximation.

depth is required. There is some work on overparameterization in the context of controllable quantum systems with *unconstrained* time-varying controls [71, 72, 73], where the authors show that there are no suboptimal local minima in the optimization landscape. For the case of a *constrained* controllable quantum system, a recent work [74] considers the problem of learning N-dimensional Haar random unitaries by gradient descent using general alternating operator ansatz of the form $e^{-i\gamma_L A}e^{-i\beta_L B}\cdots e^{-i\gamma_1 A}e^{-i\beta_1 B}$, where A and B are matrices sampled from the Gaussian Unitary Ensemble [75]. The authors show that gradient descent always converges to an accurate solution when the number of parameters is N^2 or greater, and a "computational phase transition" is observed between an under-parameterization ($< N^2$) and overparameterization ($> N^2$) regimes.

Since HVA also has the form of an alternating operator ansatz, and the problem of finding the ground states can also be seen as a constrained quantum control problem, we expect a similar overparameterization phenomenon in our setting. To investigate this, we randomly sample 100 initial parameters $\boldsymbol{\theta}$ (uniformly drawn from the interval $[0, \pi]$ for the TFIM and $[0, 2\pi)$ for the XXZ model) and perform the optimization for increasing values of L. Here, we set the stopping criterion for the optimization to $\epsilon_{\text{res}} = E(\boldsymbol{\theta})_p - E_{\text{ground}} <$

10^{-4} and the maximum number of iterations to 3000. Indeed, Figure 2.5 shows that the overparameterization phenomenon also occurs in HVA for the 12-qubit TFIM and XXZ model. We find that for both the TFIM and XXZ model, gradient descent from all 100 random starting points converges to an accurate solution once the depth L reaches a certain threshold $\tilde{L}(n)$.

Moreover, we also observe a "computational phase transition" around this threshold where the convergence speed becomes exponentially fast, i.e., the decrease of the residue energy as a function of the number of iterations. However, this threshold $\tilde{L}(n)$ is not tight, i.e., for depth $L < \tilde{L}(n)$ it is possible that gradient descent still converges to a high fidelity state. This indicates that in the setting of finding ground states using HVA, the problem is more structured and gradient descent is effective. In Figure 2.7 we see that for all system sizes, the mean number of iterations eventually converges to about 100 iterations. In addition, we can find the overparameterization thresholds $\tilde{L}(n)$ in Table 2.1 for the TFIM and XXZ model with different system sizes. Our data suggests that $\tilde{L}(n)$ has a polynomial scaling, which is compatible with the analogous parameter count required to express critical 1D ground states with an MPS. A more detailed view of this data can be found in Figure 2.6, which shows that all random initializations converge to the ground state after a certain threshold $\tilde{L}(n)$.



(a) TFIM

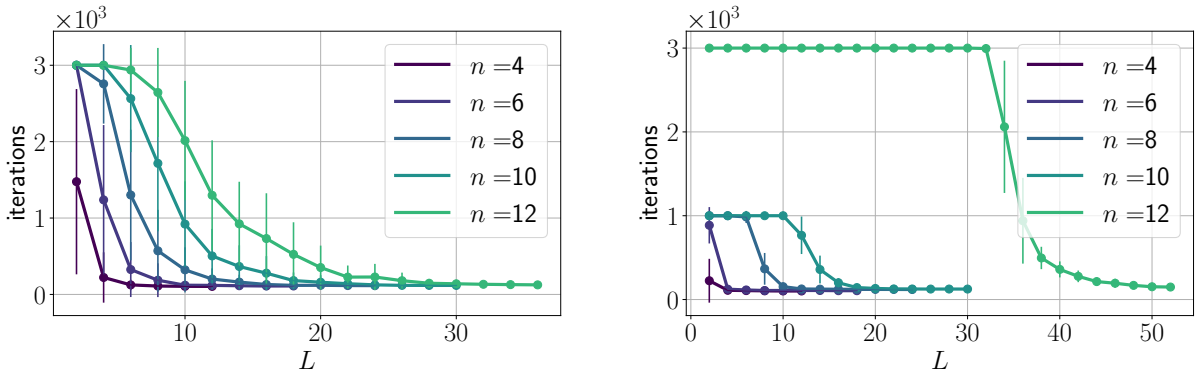
(b) XXZ model

Figure 2.6: **Ratio of random initializations that converge to the ground state.** We consider a run converged if $\epsilon_{\text{res}} \leq 10^{-4}$.

This is a striking difference with [74] where the number of parameters to achieve overparameterization is $(2^n)^2$. From Figure 2.7 we can also see that the iteration time decreases

substantially as L increases, which saturates to around 100 iterations after a certain L for all n.

We tried to observe the overparameterization phenomenon in random quantum circuits (RQC), which has layers that consists of arbitrarily chosen single and two qubit gates. We observed there is no indication or evidence that the landscape of RQC gets better as one increases the depth. On the contrary, in our experiments with random circuits of comparable depths to our HVA circuits, we were unable to observe the same overparameterization phenomenon. This can be explained from the barren plateau point of view and the lack of structure in the ansatz space.



(a) TFIM (b) XXZ model

Figure 2.7: **The mean iteration time to convergence as a function of depth.** Error bars indicate the standard deviation over 100 different initializations. For both models there is a clear cutoff where the number of iterations saturate. Note that if the number of iterations is smaller than 3000, then we know that $\epsilon_{\text{res}} \leq 10^{-4}$, indicating that the optimization has converged to a good ground state approximation. We see that the error bars decrease systematically with depth. For both models, there is a critical L after which all random initializations converge to a good ground state approximation. Moreover, for depth $L = 34$ and $L = 52$ for the TFIM and XXZ model respectively, the number of iterations to find the ground state is of the order of 100 iterations for every starting point.

	TFIM	XXZ model
n	$\tilde{L}(n)$	$\tilde{L}(n)$
4	6	4
6	6	4
8	8	8
10	10	12
12	14	36

Table 2.1: **Overparameterization threshold $\tilde{L}(n)$ for TFIM and XXZ model with different system sizes n.** By threshold, we mean that when $L \geq \tilde{L}(n)$, all the random initializations converged to an accurate solution.

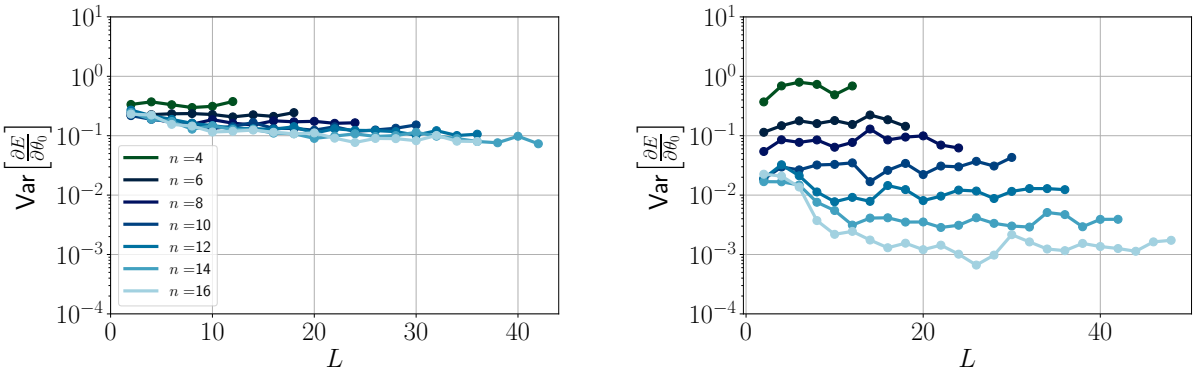
2.2.6 Ameliorated barren plateaus in HVA

In Ref. [35], a *barren plateau phenomenon* was observed for VQE on random quantum circuits, where all gradients are exponentially close to zero with overwhelmingly high probability, making local optimization within the ansatz space very challenging. The barren plateau phenomenon is due to the fact that RQCs consisting of single- and two-qubit gates form a 2-design, which means that gradients of the energy objective function will obey the same concentration of measure properties as if the circuits were Haar-random unitaries.

In contrast to the RQC ansatz, we show that the optimization landscape of HVA is much more favorable. This is clearly illustrated when optimizing the HVA corresponding to the TFIM: to begin with, as discussed in Section 2.2.4 the manifold of states has a much more restricted entanglement structure than a typical, Haar-random state – this already indicates that the HVA circuits do not form 2-designs, and thus do not obey the same kind of concentration of measure phenomenon as RQCs. On the other hand, the entanglement spectrum of the ansatz space corresponding to the XXZ model does not immediately rule out the same barren plateau behavior as exhibited by RQCs.

Nonetheless, we determined that the barren plateau problem is significantly ameliorated in the TFIM and mild in the XXZ model. In Figure 2.8, we calculated the variance of gradients as a function of qubits number n and depth L over 20 random points per n and per L. For the TFIM, the flatness of the variance curve indicates no barren plateau problem. However, for the XXZ model, we see an exponential decay, but this decay is not as strong as in RQCs [35]. The scaling of the mean gradient magnitudes follows a similar pattern. Nonetheless, we can reliably find an accurate solution when choosing an identity start, where the barren plateau problem is absent. Indeed, sampling gradients close to the identity initialization gives a constant gradient variance for all n. This indicates that

the vanishing gradient problem can be circumvented by choosing a suitable initialization strategy.



(a) TFIM

(b) XXZ model

Figure 2.8: **Variance of the gradients of a single Z_1Z_2 term with respect to θ_0 as a function of the number of qubits at initialization.** The number of samples used per n for each L is 20. (a) For the TFIM, the gradient variance decay is almost constant for all n. (b) The XXZ model gradient variance is still exponential, although the effect is not as pronounced as for the RQCs of [35], where the $n = 16$ variance is two orders of magnitude smaller.

2.2.7 The entangling power of HVA circuits

For a 1D gapped quantum system, the entanglement entropy of the ground state obeys an area law [76, 77, 78], i.e., the entanglement entropy grows proportionally to the boundary area $|\partial I|$ of the subsystem ρ_A:

$$S(\rho_A) = \mathcal{O}(|\partial I|).$$

In 1D, the boundary area $|\partial I|$ is either 1 (for an open chain) or 2 (for a closed chain), and the area law simply says that the entanglement entropy should be constant as n increases. For a 1D conformally invariant gapless (critical) system, the entanglement entropy of the ground state has a logarithmic scaling instead [79], i.e.,

$$S(\rho_A) = \mathcal{O}(\log(n)).$$

Entangling power is an important factor for characterizing the expressiveness and efficiency of many variational ansätze in condensed matter physics. It characterizes how much entanglement (measured by the entanglement entropy) can be generated by the variational circuit. For example, in the Matrix Product State representation, the entangling power is limited by the so-called *bond dimension* D which affects the expressive power and computational cost of the ansatz. For a 1D gapped system with energy gap ϵ, the ground state can be approximated well by an MPS with sublinear bond dimension $D = \exp\left(\tilde{O}(\frac{\log^{3/4} n}{\epsilon^{1/4}})\right)$ [80]. In the case of HVA, the amount of entanglement generated by the circuit depends on the depth L of the circuit. Indeed, we observed in Figure 2.3 numerically that the HVA circuits for the TFIM and XXZ model have enough entangling power to express the ground states.

As a demonstration that the full entangling power of HVA can be utilized effectively, we solve for the ground state of the so-called modified Haldane-Shastry (MHS) Hamiltonian. This model has long range interactions and is expected to have power-law entanglement scaling in the ground state [81, 82]. The MHS Hamiltonian is given by

$$H_{\text{MHS}} = \sum_{j<k}^{n} \frac{1}{d_{jk}^2}(-X_j X_k - Y_j Y_k + Z_j Z_k),$$

where $d_{jk} = \frac{n}{\pi}|\sin(\pi(j-k)/n)|$. Due to the form of the Hamiltonian, we can use the same HVA Equation (2.18) as for the XXZ model. In Figure 2.9 we see that it is possible to find the ground state with $> 99.7\%$ fidelity using a depth $L = n$ circuit for $n = 4, 8, 12, 16$.

2.2.8 Conclusion

We have shed light on some of the desirable properties of HVA as a critical ingredient in the variational quantum eigensolver algorithm. In particular, we show evidence that there are only mild or entirely absent barren plateaus in HVA. This is strikingly different from the commonly used random quantum circuits. Moreover, we also observe an overparameterization phenomenon in HVA. Similar to what was observed in the deep neural networks, the optimization landscape of HVA becomes increasingly better as the ansatz is overparameterized and eventually becomes trap free as the overparameterization reaches a certain threshold. In contrast with the case of learning Haar random unitaries, we observe that such threshold in HVA scales at most polynomially with the system size. Finally, we provide numerical evidence that HVA can be used to find the ground state of the MHS Hamiltonian, which has a power-law scaling entanglement. We believe that our findings point to important features of HVA that will lead to a deeper understanding of its effec-

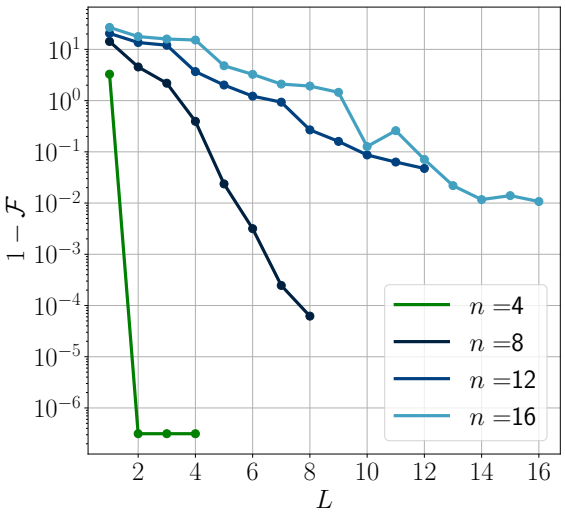


Figure 2.9: **Infidelities found after optimization for the MHS Hamiltonian.** The circuit is initialized with an identity start. For the 4-qubit case we get close to machine precision, and hence the fidelities are unstable.

tiveness, and point the way to developing more sophisticated ansätze for other many-body problems, as well as more informed optimization/initialization strategies.

As for future work, since most 1D quantum many-body systems can be simulated efficiently with classical methods, the crucible for HVA will be 2D systems. If low-depth circuits are capable of reproducing non-trivial 2D quantum states, then one can start thinking when a quantum advantage can be reached for systems where classical methods are computationally expensive or even ineffective. The effectiveness of the identity initialization, both in terms of the absence of vanishing gradients and reliability of finding a good ground state approximation is striking. Scrutinizing the mechanism for why this is the case will require a deeper understanding of the energy landscape of HVA. Our preliminary results for the XXZ model and TFIM on rectangular lattices show that this initialization strategy remains effective even for 2D systems.

Lastly, the overparameterized regime is a double-edged sword. On the one hand, it implies that we can improve the energy landscape by increasing the depth of the circuit, ameliorating the effects of local minima. On the other hand, the growth in circuit depth, may well nullify this increase in performance due to the longer coherence times required

and multiplicative gate errors. In order to assess how useful this regime is for hardware implementations, it would require an understanding of the effect that noise has on the optimization in the overparameterized regime. The recent work [83] of Wang et al. indicates that for a class of VQE ansätze including the quantum alternating operator ansatz, there could be severe noise-induced barren plateaus when the number of layers scales polynomially. However, for the practical performance of general HVA, a more careful analysis of the trade-off between the benefits of overparameterization and the detrimental effects of noise-induced barren plateaus is needed. Moreover, research on designing more effective variational quantum circuits based on HVA should also be pursued.

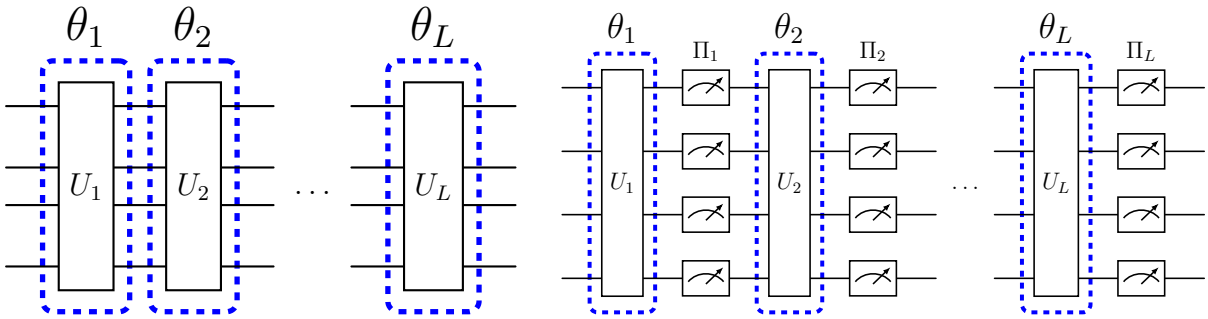
2.3 Measurement-induced entanglement phase transitions in quantum gradients

Recently, significant progress has been made in understanding the evolution of quantum entanglement in random unitary quantum circuits undergoing intermediate projective measurements. In these circuits, random nearest neighbor two-qubit gates locally entangle qubits, which generally leads to volume-law entanglement growth. When such a system is measured at randomly selected locations throughout the circuit, the measured subsystems become disentangled from the rest of the state. One might expect that this leads to a simple decrease in the coefficient of the entanglement growth volume law. However, the competition between local entanglement creation and non-local disentanglement induces a phase transition in the entanglement growth from a volume to an area law at a critical measurement rate p_c [84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94]. Moreover, it appears that this critical behavior is universal, independent of the specific implementation of both the unitary or measurement dynamics. A significant amount of theoretical understanding has been gained about the properties of entanglement phase transitions in random unitary circuits [89, 95] by mapping such systems to well-defined statistical mechanics models.

In this section, we connect research in condensed matter theory and variational quantum computing, by showing that measurement-induced entanglement phase transitions take place in two prototypical variational quantum circuits used within VQE [29]. This variational quantum algorithm is used throughout the literature to approximate quantum many-body ground states [40, 41, 52, 1, 96], perform quantum chemistry simulations [32, 97, 98, 99, 100] or in quantum machine learning approaches [101, 102, 103, 104]. Our motivation to investigate the measurement-induced entanglement transitions in variational quantum circuits are twofold. First, most of the quantum ground states of interacting many-body systems follow the area-law entanglement (up to a logarithmic correction). However, ballistic growth of entanglement in time evolution implies that circuits used in VQE can rapidly develop much more entanglement than what may be needed to simulate these ground states of interest [105, 106, 1]. Secondly, as we discussed in Section 2.2, it is known that randomly initialized variational quantum circuits tend to approximate unitary 2-designs, which are known to have exponentially decaying gradients with respect to the gate parameters as a function of system size. It has been shown that there is a close relation between entanglement production in a circuit and barren plateaus, hence it is natural to consider constraining the amount of entanglement during parts of the variational optimization as a useful strategy for increasing the trainability of variational circuits [107, 108, 109]. In fact, constraining the amount of entanglement that two-qubit gates can generate has been shown to be a potential strategy for avoiding barren plateaus[108]. We anticipate

that the inclusion of interspersed measurements in variational quantum circuits may additionally offer a way to control their quantum entanglement, which could be used as a strategy to overcome barren plateaus. Quantum hardware that allows for intermediate measurements can potentially be used to test these ideas in practice [110, 111, 112].

Here, we give numerical evidence to show that a measurement-induced entanglement phase transition takes place in the variational quantum circuits, and coincides with a "landscape transition", a change from a landscape with severe barren plateaus to a landscape with mild or no barren plateaus. This suggests that VQE with intermediate projective measurements can potentially be used to avoid barren plateaus and improve current optimization strategies. In deriving our results, we also provide a modified parameter shift rule for calculating the quantum gradients with intermediate projective measurements that may lead to the development of such algorithms.



(a) VQE Circuit (b) Projective VQE Circuit

Figure 2.10: **Quantum circuit undergoing projective measurements.** (a) Schematic representation of the circuit $U(\boldsymbol{\theta})$. Each layer $U_m(\theta_m)$ can consist of multiple gates with multiple parameters, hence θ_m is a vector of parameters. (b) For a circuit undergoing projective measurements, we apply a projector Π_m to all qubits. Whether we apply a measurement (green circle) or not is determined by flipping a coin with probability p. Once we decide that a measurement will be applied, we sample the projector according to the quantum probability $\tilde{p}_{m,i} = \mathrm{Tr}\{\Pi_{m,i}\rho\}$.

2.3.1 Variational quantum circuits undergoing measurements

We start with the parameterized n-qubit circuit of Equation (2.4),

$$U(\boldsymbol{\theta}) = \prod_{l=1}^{\overleftarrow{L}} U_l(\theta_l), \tag{2.25}$$

We want to find an approximation of the ground state of H by a VQE optimization. We therefore calculate the variational energy of Equation (2.9),

$$E(\boldsymbol{\theta}) = \mathrm{Tr}\left\{U(\boldsymbol{\theta})\rho_0 U^\dagger(\boldsymbol{\theta})H\right\},$$

with $\rho_0 = |\psi_0\rangle\langle\psi_0|$, and solve the optimization problem of Equation (2.8)

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \; E(\boldsymbol{\theta}).$$

As with other variational methods, the choice of ansatz $U(\boldsymbol{\theta})$ is crucial since the ground states must be reachable from the initial state by application of this unitary. There exists a variety of proposals, including the HVA [40, 41, 1, 52, 96] which we discussed in Section 2.2 and the Hardware Efficient Ansatz (HEA) [39, 32, 113], a type of random quantum circuit. The former exploits the structure of the Hamiltonian for the unitary ansatz design, whereas the latter aims to provide a hardware-friendly parameterization with enough degrees of freedom to capture a variety of states.

We are interested in studying random ensembles of typical VQE circuits undergoing projective measurements and the entanglement properties of the states they produce. Given a state ρ, a projective measurement in the computational basis results in

$$\rho' = \frac{\Pi_i \rho \Pi_i}{\mathrm{Tr}\left\{\Pi_i \rho\right\}}, \tag{2.26}$$

where $\Pi_i = |i\rangle\langle i|$ are the projectors onto the Z (computational) basis (see also Appendix A.2). Which projector Π_i is applied depends on the quantum probability $\mathrm{Tr}\left\{\Pi_i \rho\right\}$.

Consider the circuit in Equation (2.25). After each layer l, with probability p (the measurement rate), we apply a projective measurement to each qubit. For L layers, we then obtain the variational state

$$\rho_L(\boldsymbol{\theta}) = \left(\prod_{l=1}^{\overleftarrow{L}} \Pi_l U_l(\theta_l)\right) \rho_0 \left(\prod_{l=1}^{\overrightarrow{L}} U_l^\dagger(\theta_l)\Pi_l\right) p_L^{-1}(\boldsymbol{\theta}), \tag{2.27}$$

where $p_L(\boldsymbol{\theta})$ is the probability of obtaining the state $\rho_L(\boldsymbol{\theta})$ given the L sets of measurements performed, see also Section 2.3. The projective measurement is represented by the projector $\Pi_l = \Pi_{l,0} \otimes \ldots \otimes \Pi_{l,n}$ where $\Pi_{l,i} \in \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ if we perform a measurement and $\Pi_{l,i} = I$ otherwise. Here, $\rho_L(\boldsymbol{\theta})$ is the normalized state obtained after applying the circuit with intermediate measurements. Each projector Π_l has 3^n different configurations, hence there will be a total of 3^{NL} possible states $\rho_L(\boldsymbol{\theta})$. Note that each state $\rho_L(\boldsymbol{\theta})$ corresponds to a pure state. Also, we want to emphasize that we are not performing any optimization; we consider the variational circuit at initialization.

2.3.2 Measurement–induced entanglement phase transitions

Given a state produced by quantum circuit interspersed with intermediate measurements, we can calculate the bipartite von Neumann entanglement entropy of Equation (2.20) as a function of the number of qubits n and the measurement rate p and denote this by

$$S(n, p) = \mathrm{Tr}_A\{\rho_L(\boldsymbol{\theta})\},$$

where $\rho_L(\boldsymbol{\theta})$ is depends on n and p and A is one of the subsystems in the bipartition of Figure 2.2.

We know that the projective measurements disentangle the system over any length scale due to a local projection onto a single state. As a result, the unitary dynamics locally entangles nearest neighbor qubits, whereas measurements globally disentangle the different subsystems by projecting out part of the state. This competition induces a dynamical phase transition between a volume and area law regime of entanglement scaling at a critical measurement rate p_c [84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94]. Although the critical point p_c can vary between different types of random unitary dynamics and measurement schemes, the critical exponent characterizing the correlation length scale divergence $\xi \propto (p - p_c)^{-\nu}$ appears to be the same for different models at $\nu \approx 4/3$. This critical exponent can be derived by considering toy models and mapping the projective dynamics to a two-dimensional percolation model, which is exactly solvable [85, 86, 90, 95, 89].

Central to the investigations on phase transitions induced by measurements is the concept of steady state entanglement dynamics [85, 86, 87]. Given a circuit with a number of qubits n, we are primarily interested in the late time behavior when $L \to \infty$. In this infinite depth (long time) limit we expect the system to evolve into a steady state, characterized by a typical value of entanglement entropy that depends on the measurement rate p, but not the dynamics at finite times. In order to characterize this regime, we can investigate the average entanglement entropy as a function of depth for different values of

p. For the moderate system sizes considered here, we observe steady state entanglement dynamics at $L = 16$.

For our numerical study, we investigate the projective entanglement dynamics of the XXZ-chain HVA and the HEA, whose circuits are depicted in Figure 2.11. Notice that the dynamics in the HVA is specified by a Hamiltonian in contrast to random unitaries. The HVA for the XXZ model is of particular interest since the XXZ Hamiltonian is Bethe-ansatz integrable, i.e. there exists an analytical solution for the energy spectrum. Additionally, the entanglement properties of these systems undergoing quenches can be understood analytically [114, 106]. For such models, it is still an open question if the corresponding unitary dynamics interspersed with measurements will produce a measurement-induced entanglement phase transitions [89]. Here, we address a closely related model, where the unitary dynamics are generated by random quenches under the XXZ Hamiltonian. For the HEA, we expect that the behavior is close to that of random circuits [115].

Since phase transitions only occur in the thermodynamic limit $n \to \infty$, we have to take care of the finite-size effects in analyzing our numerical data. To account for finite-size effects, we fit the scaling form [86, 85, 89]

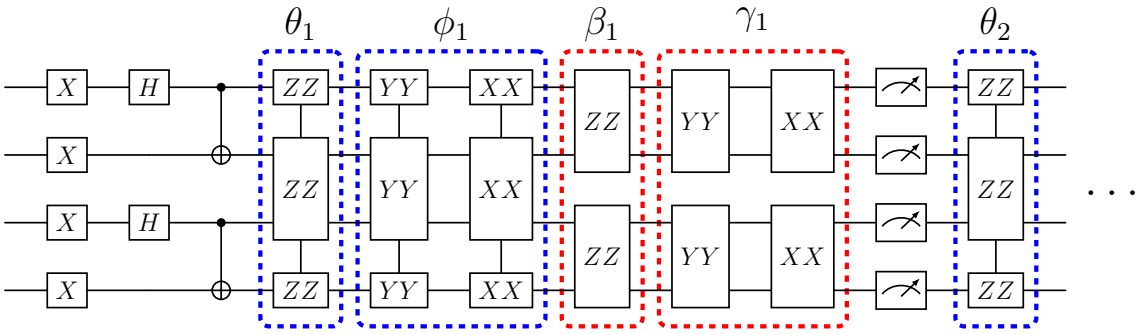
$$S(n, p, \nu) - S(n, p_c, \nu) = f(n^{1/\nu}(p - p_c)),$$

where f is a scaling function, to get a data collapse of the individual circuits of size n. To determine p_c and ν, we minimize a Chi-squared statistic between the scaling form above and the data, and use a statistical bootstrap to verify the integrity of the fit. In Figure 2.12 we find critical exponents close to the previously mentioned value of $\nu \approx 4/3$. To extrapolate the critical exponent to the thermodynamic limit, we do a linear fit of ν as a function of $1/n'$ where $n_{\max}/2 \leq n' \leq n_{\max}$ is the largest value of n in the data set. The intercept then gives us the value of ν for $n' \to \infty$ [85]. The details of our statistical estimation procedure are outlined in Appendix C.1. In addition to the finite scaling analysis, we can investigate the quantum mutual information,

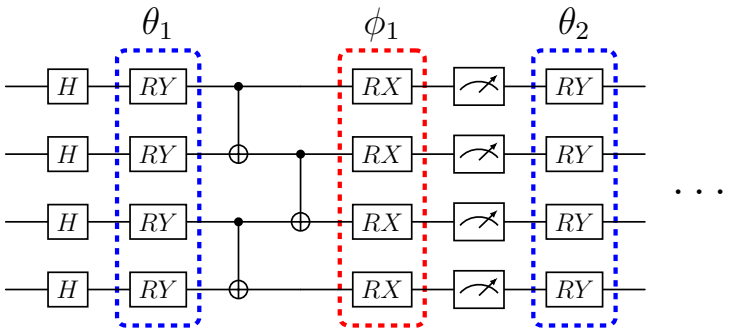
$$I(A, B) = S_A(n, p) + S_B(n, p) - S_{A\cup B}(n, p),$$

between qubits A and B separated by a distance r, which we expect to peak at a critical point due to subsystem correlations becoming non-negligible. From these data, we find similar critical measurement rates $p_c \approx 0.25$ and $p_c \approx 0.5$ for the XXZ-HVA and HEA, respectively. In Appendix C.2 we give further details on this procedure.

The results in Figure 2.12 suggest that an entanglement phase transition takes place in two prototypical circuits used in variational quantum algorithms. Although we have studied static circuits here where no optimization takes place, we can investigate how the



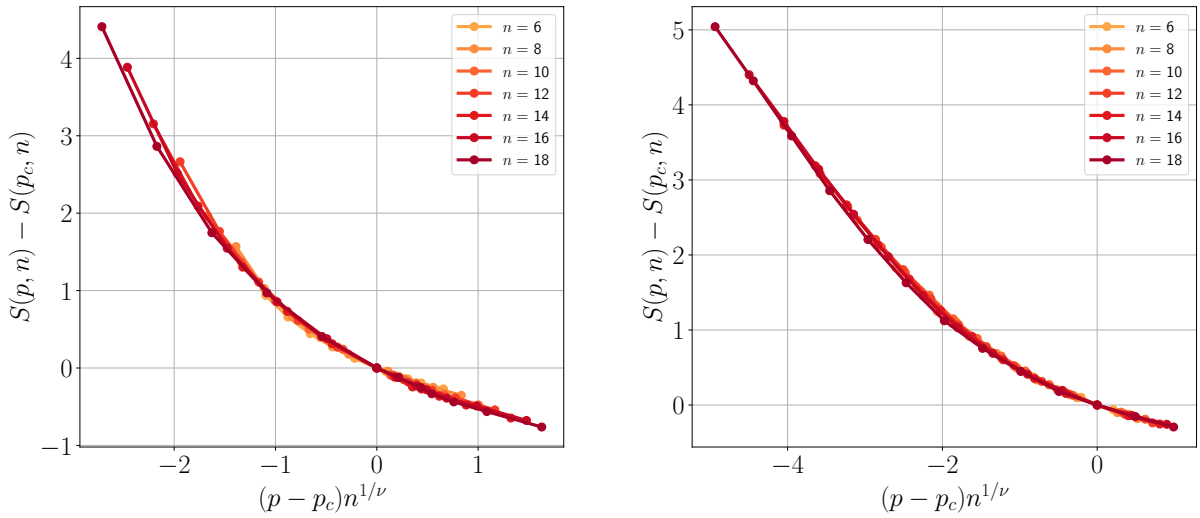
(a) XXZ



(b) HEA

Figure 2.11: **Schematic depiction of the variational circuits studied.** (a) For the XXZ-HVA, we prepare a Bell state on the even sites and alternatingly and apply ZZ, YY and XX two-qubit rotations on odd and even bonds in the chain. See Section 2.2.2 for more details. (b) The initial state in the HEA consists of the equal superposition followed by L layers of low-depth entangling unitaries. These unitaries consists of n Pauli-Y rotations on each qubit, a chain of nearest neighbor CNOTs and n Pauli-X rotations on each qubit. All $2N$ rotations are controlled by individual parameters $\theta_{i,l}, \phi_{i,l}$, where $i = 1, \ldots, n$ indicates the qubit and $l = 1, \ldots, L$ indicates the layer. After each layer, we perform a projective measurement according to Equation (2.26) with probability p on each qubit (indicated by the green circles here), bringing the average number of measurements in the circuits to NMp.

projective measurements affect the gradients with respect to the gate parameters in the circuit.



(a) XXZ-HVA (b) HEA

Figure 2.12: **Data collapse of the average entanglement entropies.** (a) For the XXZ-HVA, we find $p_c = 0.25$ and $\nu \approx 1.22 \pm 0.24$. (b) For the HEA, we find $p_c \approx 0.5$ and $\nu \approx 1.26 \pm 0.23$. The error bars are calculated as the difference between the critical exponent in the thermodynamic extrapolation and the finite-size data collapse. The average $S(p,n)$ is obtained by averaging over 3×10^3 circuit realizations with all circuit parameters sampled uniformly in $(0, 2\pi)$. After each layer, we apply a computational basis measurement with probability p. Due to the difficulty in simulating large systems, we restrict ourselves to $n = 6, 8, \ldots, 18$.

2.3.3 Projective gradients and barren plateaus

The variational energy $E(\boldsymbol{\theta})$ is typically a non-convex function of the gate parameters $\boldsymbol{\theta}$. In practice one typically uses a gradient-based method to find a minimum of the cost function. To calculate the necessary gradients with respect to the layer parameters, we can employ hardware-friendly methods, most of which rely on the usage of the so-called parameter-shift rule which we discussed in Section 2.1.2 [30, 31, 116, 117, 118, 119, 120, 121]. In its standard form, this allows one to calculate the gradient with respect to the parameters of

a gate generated by a Pauli operator as

$$\partial_{\theta_l} E(\boldsymbol{\theta}) = \frac{1}{2}\left(E^{+,l}(\boldsymbol{\theta}) - E^{-,l}(\boldsymbol{\theta})\right),$$

where we use $\pm, l$ to denote the expectation value under a circuit $U(\boldsymbol{\theta})$ where parameter l has been shifted by $\pm\pi/2$. In other words, the gradient can be calculated by shifting the parameter θ_l by $\pm\pi/2$ and calculating the difference of the expectation values of H under the shifted circuits. Unfortunately, this kind of gradient calculation is plagued by barren plateaus in the cost landscape: gradients with respect to the gate parameters vanish exponentially with the number of qubits in the circuit, preventing us from optimizing the circuit. To mitigate this problem, a variety of recent works are aimed at finding ways to avoid these regions where optimization is hard [108, 37, 68, 122, 115, 123, 124].

Here, we investigate the barren plateau problem under the influence of projective measurements, more specifically the variance of the gradients in the XXZ-HVA and the HEA with intermediate projective measurements. There has been prior work on gradient through non-unitary quantum circuits. For instance, in [125] the quantum natural gradient [26] is extended to quantum channels. Additionally, in [126] measurement-based VQE is investigated, but only in the context where an entangled cluster state is prepared and measurements are directly part of the algorithm [127]. None of these works consider quantum gradients through a circuit undergoing projective measurements, which is the case we consider here.

The gradient with respect to a single parameter θ_l of the expectation value of a Hermitian operator H undergoing a set of measurements is given by

$$\partial_{\theta_l} E(\boldsymbol{\theta}) = \partial_{\theta_l} \mathrm{Tr}\left\{H\rho_L(\boldsymbol{\theta}\right\},$$

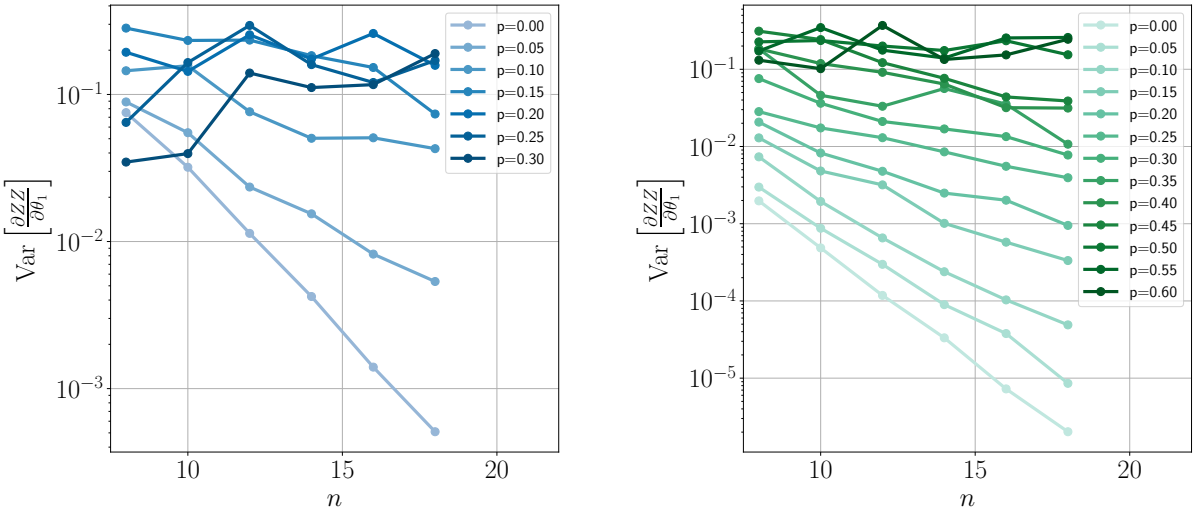
where $\rho_L(\boldsymbol{\theta})$ is given in Equation (2.27)). In Appendix C.3, we show that the full projective gradient can be written as

$$\partial_{\theta_l} E(\boldsymbol{\theta}) = \frac{1}{2}\left(\left(E^{+,l}(\boldsymbol{\theta}) - E(\boldsymbol{\theta})\right)\frac{p_L^{+,l}}{p_L} - \left(E^{-,l}(\boldsymbol{\theta}) - E(\boldsymbol{\theta})\right)\frac{p_L^{-,l}}{p_L}\right). \tag{2.28}$$

The probabilities p_L and $p_L^{\pm,l}$ are the probabilities of obtaining $\rho_L(\boldsymbol{\theta})$ and $\rho_L(\theta_1, \ldots, \theta_l \pm \pi/2, \ldots \theta_L)$, respectively. Note that obtaining these probabilities will be difficult and require a large number of measurements, since estimating the ratio $p_L^{-,l}/p_L$ requires full knowledge of the wave function.

To investigate the severity of the barren plateau effect, we consider the same circuits as in Figure 2.12 and examine the projective gradients of Equation (2.28) with respect to the

expectation value of $H = Z_0 Z_1$. We calculate the projective gradients for the first circuit parameter (θ_1 in the first parameterized layer in both the HVA and HEA (see Figure 2.11)). We consider a depth $L = 16$ circuit for system sizes $n = 8, \ldots 18$. In Figure 2.13a and Figure 2.13b, we observe that the gradient variances in both the XXZ-HVA and HEA transition from exponentially decaying to a constant as the measurement rate increases. This transition coincides with the critical measurement rate for the volume-area law transition (See Appendix C.5). Therefore, we see that the measurement-induced entanglement phase transitions induces a landscape transition in the circuit from mild/severe barren plateaus to no barren plateaus.



(a) XXZ-HVA (b) HEA

Figure 2.13: **Variance of the projective gradients taken with respect to the first parameter in the circuit.** θ in the first parameterized layer in both the HVA and HEA, see Figure 2.11. The variances are estimated over 10^3 samples where for each data point, we randomly choose measurements with probability p and uniformly sample the gate parameters. The gradient is then calculated exactly from Equation (2.28). We emphasize that these gradients are thus calculated with respect to the individual pure states resulting from measuring the state during the application of the circuit. For the 1D XXZ-HVA circuits with depth $L = 16$ (a) and the 1D HEA circuit with depth $L = 16$ (b) the gradient variance becomes constant as the measurement rate p increases.

This landscape transition can serve as the motivation for a projective gradient VQE algorithm where the early optimization of the circuit is done with projective gradients to escape barren plateaus due to initialization. However, calculating the gradients in Equation (2.28) is exponentially hard in the number of layers L, since we need accurate estimates of p_L and $p_L^{\pm,l}$.

On the contrary, the mixture of all pure states $\rho_L(\boldsymbol{\theta})$ has a simple gradient formula that can be calculated in practice, as we show in Appendix C.4. This approach does not require post-selection and can be applied by ignoring the results of the measurements. The resulting ensemble however, corresponds to system at infinite temperature [85, 87, 86]. It is known that such a high temperature ensemble will suffer again from barren plateaus [83]. Additionally, we require a pure state as the outcome of our optimization algorithm, which will require annealing the measurement rate to zero during the optimization. Any useful variational algorithm with intermediate measurements must not remix all projective states but still be efficiently calculable. We leave the exploration of this class of algorithms as future work.

2.3.4 Conclusion

We demonstrated the existence of a measurement-induced entanglement phase transition in variational quantum circuits which coincides with a "landscape transition" in the behavior of quantum gradients. As mentioned earlier, the exponentially-vanishing quantum gradients in presence of volume-law entanglement growth, the so-called barren plateau, is a serious obstacle in the applications of variational quantum circuits. Our work suggests that intermediate projective measurements may provide a useful knob to control the barren plateau issue. Inclusion of the measurement protocol in the quantum-classical hybrid algorithm would be a timely development given that quantum computing hardware companies like IBM and Honeywell now allow their users to perform mid-circuit measurements, enabling the real-time logic required for performing these algorithms in an experimental setting [110, 111, 112]. In particular, the HVA quantum circuits considered in here could be implemented in the quantum hardware. For the projective gradient VQE, the exponential sum in Equation (2.28) currently inhibits the number of measurements that can be performed in practice. A detailed analysis of when and how a projective circuit optimization can be practical and "advantageous" would be an excellent topic of future study.

For a practical implementation of a projective gradient VQE algorithm, note that the scheme we provided here is quite general and many extensions and modifications are possible. For instance, the projective measurements used in here can be replaced by general

Positive Operator Value Measures (POVM) or parameterized measurements. Additionally, we have focused on one-dimensional quantum circuits where the measurement-induced entanglement transition belongs to the same universality class as in the random unitary circuits. It would be interesting to consider moderately sized quantum circuits with a two-dimensional topology, and see if a similar phase transition appears there and investigate the universality class.

2.4 Quantum-Classical-Quantum interfaces

The paradigm of NISQ devices limits quantum algorithms to circuits of low qubit numbers, low depth, and low connectivity [20]. This poses serious concerns on the actual usefulness of quantum computers in the near term and has thus ignited a both experimental and theoretical quest to make use of NISQ hardware [128, 129, 24].

To combat the noise in these systems, subsequent variants incorporated the idea of quantum error mitigation [130, 131, 132, 133]. This refers to schemes whereby noisy experimental implementations (e.g., in different noise regimes or with different gate choices), together with suitable classical post-processing, are used to simulate a target, noiseless quantum circuit of limited size. This offers a NISQ alternative to quantum error correction (which requires large-scale quantum circuits), where full fault tolerance is achieved by actively correcting errors on the quantum hardware during the execution of the computation. Here, we achieve error correction by removing noisy gates from circuit at the cost of an increase in the sampling budget.

More recently, a different type of hybrid method has been put forward [134, 135, 136, 137, 138, 139]. There, a classical algorithm calls a quantum computer as a sub-routine to simulate a larger quantum circuit. However, the cost of this is that both the number of queries to the quantum sub-routine and the classical post-processing runtime unavoidably grow exponentially with the size of the target circuit. Moreover, a particularly challenging aspect of NISQ devices is their inability to run algorithms that require high, long-range connectivity among the constituent qubits. In most NISQ hardware, long-range gates are synthesized by a long sequence of nearest-neighbor gates. This drastically inflates the circuit depth and causes large infidelity due to noise accumulation incurred during the syntheses. This is a crucial limitation in the NISQ era.

Here, we take a conceptually different direction from previous hybrid schemes: instead of assembling a large quantum circuit from small pieces, we simulate a high-connectivity circuit from circuits with low connectivity and depth. To that end, we introduce the notion of Quantum-Classical-Quantum (QCQ) interfaces. A QCQ interface for a gate U corresponds to a local measurement on the qubits on which U acts followed by a re-preparation of those same qubits in a random product state that depends on U. In other words, the interface performs a hybrid quantum-classical simulation of U. Each interface introduces a multiplicative statistical overhead that, as we prove below, is independent of the on-chip distance between the qubits. Hence, for fixed number of interfaces, e.g., the longer the range of the target gates is, the more drastic the reduction in depth attained is at the expense of a constant overall statistical overhead.

More technically, our interfaces combine state-of-the-art state estimation based on single-qubit random measurements [46, 140] with quasi-probability representations based on frames [141, 142]. Such representations have been used for classically simulating a quantum circuit with Monte Carlo sampling techniques [143, 144, 145]. In particular, our algorithm can be seen as a hybrid version of the scheme of Ref. [145] where everything is quantum except for a subset of gates that one wishes to "cut out" of the experimental circuit. Here, we choose such subset in terms of the on-chip qubit distance. However, other relevant choices may be due simply to error mitigation or hardware-specific limitations. As most quasi-probability schemes, our method suffers from the infamous sign problem [146, 147, 148]. Remarkably, the severity of the problem depends only on the number of interfaces and not the on-chip distance between the qubits. Moreover, as by-product contribution, in order to minimize the average sign of our quasi-probability representation, we develop a Metropolis-Hastings simulated-annealing algorithm based on random walks in the space of a dual Positive Operator Valued Measurement (POVM). We implement such walks through a convenient, long-known parameterization of generalized inverse matrices [149]. This allows us to decrease the sample complexity overhead per interface by almost a factor of four relative to the canonical POVM choice, constituting a practical tool of general relevance for sign-problem mitigation [150, 151].

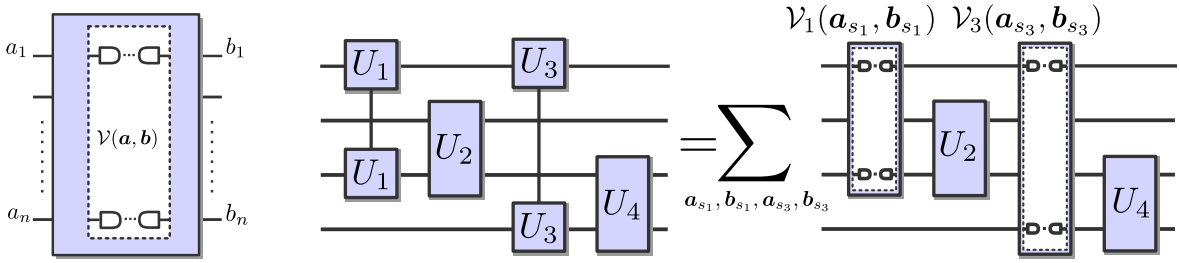
2.4.1 Positive Operator Valued Measures

Here, we give a high level description of our method and leave the formal treatment in terms of frame theory [141, 142] for the Appendix (See Appendix D.1).

We consider an n-qubit system $\mathcal{S}$ described by a density matrix ρ. This density matrix can be fully described via the measurement statistics of an informationally complete positive operator valued measure (IC-POVM) $\boldsymbol{M} = \{M_{\boldsymbol{a}}\}_{\boldsymbol{a}\in\{1,\ldots,m\}^n}$, which can be constructed by taking the tensor product of single qubit IC-POVMS, $M_{\boldsymbol{a}} = M_{a_1} \otimes \ldots \otimes M_{a_N}$, where $M_{a_i} \geq 0$ and $\sum_{a_i=1}^{m} M_{a_i} = I$ [28, 152] (see Appendix A.3). For each operator $M_{\boldsymbol{a}}$ we can define a dual IC-POVM element $\tilde{M}_{\boldsymbol{a}}$ such that the following equality holds

$$\rho = \sum_{\boldsymbol{a}} p_\rho(\boldsymbol{a}) \tilde{M}_{\boldsymbol{a}}, \tag{2.29}$$

where $p_\rho(\boldsymbol{a}) := \mathrm{Tr}\{M_{\boldsymbol{a}}\rho\}$ is the probability of measurement outcome $\boldsymbol{a}$ on ρ. Equation (2.29) is the basis of classical-shadow tomography, a powerful technique to get compact classical representations of states from measurements [153, 140]. Note that Equation (2.29) also works if $M_{\boldsymbol{a}}$ acts on a subset of all n qubits, e.g. $\rho = \sum_{\boldsymbol{a}} p_\rho(\boldsymbol{a})(\tilde{M}_{\boldsymbol{a}} \otimes \rho_{\mathrm{red}}(\boldsymbol{a}))$ where $\rho_{\mathrm{red}}(\boldsymbol{a})$ is the normalized state on the rest of the system after applying $M_{\boldsymbol{a}}$.



(a) A QCQ interface (b) Replacing multiple gates with QCQ interfaces

Figure 2.14: **Schematic depiction of the QCQ interfaces.**(a) A QCQ interface $\mathcal{V}(\boldsymbol{a},\boldsymbol{b})$ applying the identity operator $U_k = I$ between qubits 1 and n. We measure the POVM $\boldsymbol{M}$ on both qubits reprepare them in a product state that depends on the simulated gate and the outcome $\boldsymbol{a}$. The other $n-2$ qubits are left untouched. (b) An exemplary 4-qubit circuit (left) is simulated by a hybrid quantum-classical circuit (right), where the non nearest-neighbor gates $\mathcal{U}_1$ and $\mathcal{U}_3$ are substituted by QCQ interfaces ($\mathcal{V}_1(\boldsymbol{a}_{s_1},\boldsymbol{b}_{s_1})$ and $\mathcal{V}_3(\boldsymbol{a}_{s_3},\boldsymbol{b}_{s_3})$, respectively). The summation over $(\boldsymbol{a}_{s_1},\boldsymbol{b}_{s_1},\boldsymbol{a}_{s_3},\boldsymbol{b}_{s_3})$ represents the average over all interface outcomes sampled (see text).

The dual POVM elements $\tilde{M}_{\boldsymbol{a}}$ can be expressed in terms of the $M_{\boldsymbol{b}}$'s as

$$\tilde{M}_{\boldsymbol{a}} = \sum_{\boldsymbol{b}} \tilde{T}_{\boldsymbol{a},\boldsymbol{b}} M_{\boldsymbol{b}}, \tag{2.30}$$

where $\tilde{T} = \mathfrak{T}\,T\,\mathfrak{T}$ and $T_{\boldsymbol{a},\boldsymbol{b}} := \mathrm{Tr}\{M_{\boldsymbol{a}} M_{\boldsymbol{b}}\}$, hence T is an $m^n \times m^n$ matrix. The matrix $\mathfrak{T}$ has to satisfy the equation $T = T\mathfrak{T}T$, i.e. $\mathfrak{T}$ is a generalized inverse of T [154].

By virtue of Eqs. (2.29) and (2.30), we can then express any ρ as an affine combination of product states by normalizing the POVM elements.

This fact has been used to reconstruct quantum states [46], processes [155], and overlaps [156] from single-qubit measurements. Additionally, this has been used to simulate quantum circuits [157] with generative machine learning models, where $\mathfrak{T}$ was taken as the canonical pseudo-inverse of T. However, other choices of $\mathfrak{T}$ are possible. The columns of $\mathfrak{T}$ are normalized, but in general, its elements can be positive or negative, hence we can understand it as a quasi-probability distribution [145]. The negativity of $\mathfrak{T}$ has important consequences for the sample complexity of our algorithm.

2.4.2 Interfaces for hybrid classical-quantum circuits

Our goal is to simulate observable measurements on quantum circuits using hybrid classical-quantum ones. More precisely, we are given an observable O, an n-qubit input state $\rho_0 := |0\rangle\langle 0|$, and a target circuit $C := \{U_k\}_{k=1,\dots,K}$, consisting of single- or two-qubit unitary gates U_k. We denote by $s_k \subset \mathcal{S}$ the subset of qubits on which U_k acts, and by $\boldsymbol{a}_{s_k}$ a corresponding sub-string of POVM measurement outcomes on s_k. There are K gates $\{U_k\}_{k=k_1,\dots,k_K} \subset C$ where $k_i \leq L$ in the circuit that we intend to replace, because they are for instance particularly experimentally demanding for NISQ implementations or do not match the native hardware connectivity of the device.

The case we explicitly study below is that of two-qubit gates on qubits far apart in the connectivity graph in question. We want to estimate the expectation value $\mathrm{Tr}\{\rho_L O\}$ of O on the output state $\rho_L := U_L \dots U_1 \, \rho_0 \, U_1^\dagger \dots U_L^\dagger$ by substituting every U_k by a classical simulation of it. Our main tool to achieve this are interfaces between quantum objects and their (classical) representations. Note that we can also use a partial state reconstruction. The first one is based on Equation (2.29):

Definition 2.4.1 (*Quantum-classical interfaces*)**.** *We refer to a QC interface on s_k as the assignment of a classical snapshot $\tilde{M}_{\boldsymbol{a}_{s_k}}$ according to the measurement outcome $\boldsymbol{a}_{s_k}$ of a factorable POVM $\boldsymbol{M}$ on a state ρ, occurring with probability $p_\rho(\boldsymbol{a}_{s_k}) := \mathrm{Tr}\left\{M_{\boldsymbol{a}_{s_k}} \rho\right\}$.*

Hence, the QC interface is a list of bitstrings corresponding to measurement outcomes on the subsystem s_k (see Appendix A.2).

From the outcomes $\boldsymbol{a}_{s_k}$ of the QC interface, we can use Equation (2.30) to reconstruct the state by importance sampling $\boldsymbol{b}_{s_k}$. To achieve this, we first define the normalized states $\bar{\rho}_{\boldsymbol{b}} := M_{\boldsymbol{b}}/\mathrm{Tr}\{M_{\boldsymbol{b}}\}$. Next, we rewrite $\tilde{T}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}$ as

$$\tilde{T}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}} =: \left\|\tilde{T}_{\boldsymbol{a}_{s_k}}\right\|_1 \mathrm{sgn}\big(\tilde{T}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\big)\, p(\boldsymbol{b}_{s_k}|\boldsymbol{a}_{s_k}), \tag{2.31}$$

where $\left\|\tilde{T}_{\boldsymbol{a}_{s_k}}\right\|_1 := \sum_{\boldsymbol{b}_{s_k}} \left|\tilde{T}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\right|$ is the l_1-norm of the rows $\tilde{T}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}$, and $p(\boldsymbol{b}_{s_k}|\boldsymbol{a}_{s_k}) := \left|\tilde{T}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\right| / \left\|\tilde{T}_{\boldsymbol{a}_{s_k}}\right\|_1$ the conditional probability distribution obtained by taking the absolute value of the rows and normalizing appropriately. By construction, $p(\circ|\boldsymbol{a}_{s_k})$ is a valid probability distribution, which allows us to quantum Monte Carlo simulate $\tilde{M}_{\boldsymbol{a}_{s_k}}$ by sampling $\boldsymbol{b}_{s_k}$ [145]. This leads us to the definition of our second interface:

Definition 2.4.2 (*Classical-quantum interface*)**.** *We refer as CQ interface on s_k as the repreparation of the state $\bar{\rho}_{\boldsymbol{b}_{s_k}}$, with probability $p(\boldsymbol{b}_{s_k}|\boldsymbol{a}_{s_k})$, given a classical snapshot $\tilde{M}_{\boldsymbol{a}_{s_k}}$. Each sampled pair $(\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k})$ is assigned the value $\left\|\tilde{T}_{\boldsymbol{a}_{s_k}}\right\|_1 sgn\big(\tilde{T}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\big)\mathrm{Tr}\{M_{\boldsymbol{b}}\}$.*

The CQ interface is thus a collection of bitstrings indicating which state to reprepare on s_k, while we keep track of the signs and norms of $\tilde{T}$. One can combine the QC with the CQ interface to represent ρ by measuring and repreparing states on s_k. The main contribution of our work is going beyond this identity. To do this, we absorb the action of a gate U_k acting on s_k into the measurement-and-repreparation of ρ by defining $\tilde{T}^{U_k} := \mathfrak{T}\, T^{U_k}\, \mathfrak{T}$, where

$$T^{U_k} := \mathrm{Tr}\left\{U_k M_{\boldsymbol{a}_{s_k}} U_k^\dagger M_{\boldsymbol{b}_{s_k}}\right\}.$$

We provide a derivation of this quantity in Appendix D.1. This leads us to our final definition:

Definition 2.4.3 (*Quantum-classical-quantum interface*)**.** *We refer as a QCQ interface on s_k given a gate U_k as the measurement of $\boldsymbol{M}$ with outcome $\boldsymbol{a}_{s_k}$, followed by the repreparation of $\bar{\rho}_{\boldsymbol{b}_{s_k}}$ with probability $p_{U_k}(\boldsymbol{b}_{s_k}|\boldsymbol{a}_{s_k})$. Each sampled pair $(\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k})$ is assigned the value $v_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}} := \left\|\tilde{T}^{U_k}_{\boldsymbol{a}_{s_k}}\right\|_1 sgn\big(\tilde{T}^{U_k}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\big)\mathrm{Tr}\{M_{\boldsymbol{b}}\}$. We represent this interface by $\mathcal{V}_k(\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k})$.*

Note that we can place an interface at any point in the circuit to replace a gate. For example, we can perform the gates $\{U_1 \dots U_{l_1-1}\}$ to our initial state ρ_0, create an interface $\mathcal{V}_{l_1}(\boldsymbol{a}_{s_{l_1}}, \boldsymbol{b}_{s_{l_1}})$ and then apply the rest of the circuit $\{U_{l_1+1} \dots U_K\}$ to the reprepared state $\bar{\rho}_{s_{l_1}}$. By combining Eqs. Equation (2.29), Equation (2.30) and Equation (2.31) we can obtain the following equation for the expectation value of an observable O via a QCQ interface:

$$\begin{aligned}\mathrm{Tr}\{\rho O\} = \sum_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}} & p_\rho(\boldsymbol{a}_{s_{l_1}})\, v_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}} \mathrm{Tr}\left\{M_{\boldsymbol{b}_{s_k}}\right\} \\ & \times \mathrm{Tr}\left\{U_K \dots U_{l_1+1} \bar{\rho}_{\boldsymbol{b}_{s_k}} U_{l_1+1} \dots U_K O\right\}, \end{aligned} \tag{2.32}$$

We can extend the single QCQ interface example above to multiple interfaces by applying subsequent measurement-and-repreparation steps and multiplying the norms and signs $v_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}$ of each interface accordingly.

Equation (2.32) and its generalization to multiple interfaces can be experimentally calculated with a finite statistics estimator $O^*_{M_s}$ over M_s runs (see Appendix D.2.). We refer to M_s as the *sample complexity* of our protocol. Clearly, the estimation of observables via QCQ interfaces comes at a cost. In particular, the multiplicative factors of $v_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}$ increase the variance of the observable estimator $O^*_{M_s}$, hence we need more runs M_s to get an accurate estimate of $\mathrm{Tr}\{\rho_K O\}$. In practice, M_s needs to be chosen to guarantee that the statistical error and significance level (failure probability) of the estimation are

respectively given by target values ε and δ. The entire procedure is sketched by the pseudocode in Algorithm 1.

To quantify the runtime of the algorithm given ε and δ, we define the *interface negativity* of the gate U_k and the *total forward interface negativity* of the entire circuit C respectively as

$$\mathcal{N}_{U_k} \coloneqq \max_{\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k}} \left\| \tilde{T}^{U_{s_k}}_{\boldsymbol{a}_{s_k}} \right\|_1 \mathrm{Tr}\left\{ M_{\boldsymbol{b}_{s_k}} \right\} \quad \text{and} \quad \mathcal{N}_{\rightarrow} \coloneqq \prod_{k \in K} \mathcal{N}_{U_k}. \tag{2.33}$$

This allows us to state the following theorem.

Theorem 2.4.1. *[Correctness and sample complexity] The finite-statistics average O_M^* of Algorithm 1 is an unbiased estimator of $\mathrm{Tr}\{\rho_L\, O\}$ (See Appendix D.2). Moreover, if*

$$M_s \geq \mathcal{N}_{\rightarrow}^2 \times \frac{2\,\|O\|^2\,\log\left(2/\delta\right)}{\varepsilon^2}, \tag{2.34}$$

with $\|O\|$ the operator norm of O, then, with probability at least $1-\delta$, the statistical error of O_M^* is at most ε.

The proof follows straightforwardly from the Hoeffding bound. We note that the factor $\frac{2\,\|O\|^2\,\log\left(2/\delta\right)}{\varepsilon^2}$ in Equation (2.34) is the equivalent sample complexity bound one would obtain if $\mathrm{Tr}\{\rho_L\, O\}$ was estimated from measurements on the actual state ρ_L. Hence, $\mathcal{N}_{\rightarrow}^2$ quantifies the runtime overhead introduced by the interfaces. In that regard, the interface negativities play the same role in our hybrid classical-quantum simulation as the negativities of Ref. [145] in fully classical simulations with quasi-probability representations. An innovative and advantageous feature of Equation (2.33) is the presence of the POVM-element trace $\mathrm{Tr}\left\{ M_{\boldsymbol{b}_{s_k}} \right\}$ in $\mathcal{N}_{U_k}$, which comes from the state repreparation. Indeed, since $\mathrm{Tr}\left\{ M_{\boldsymbol{b}_{s_k}} \right\} < 1$, the $\mathcal{N}_{U_k}$'s (and therefore also $\mathcal{N}_{\rightarrow}$) are significantly smaller than their counterparts for fully classical simulations [145]. This is consistent with the intuition that hybrid classical-quantum Monte Carlo simulations should cause lower sample-complexity increases than fully classical ones. Our bound is similar to sample complexity of the space-like circuit cuts in [139] and [158], but is not restricted specific gates.

Either way, the most relevant property for our purposes is that $\mathcal{N}_{\rightarrow}^2$ (and therefore also M_s) is independent not only of the numbers of gates L or qubits n but most importantly, of the connectivity-graph distance between the qubits on which the interfaces act. In other words, for a fixed budget of measurement runs, simulating a gate U_k with a QCQ interface increases the statistical error at most by a constant factor $\mathcal{N}_{U_k}$, regardless how far apart in the circuit the qubits s_k are. In contrast, experimentally synthesizing U_k with noisy

nearest-neighbor gates would give a systematic error due to infidelity accumulation that grows with the distance between those qubits.

With regard to the limitations of our method, we note that $\mathcal{N}_{\rightarrow}^2$ grows exponentially with the number K of interfaces used. We can therefore only simulate a limited a number of gates before the number of measurement-and-repreparation steps becomes too large to perform in practice. Additionally, the forward negativity depends on $\|\tilde{T}_{\boldsymbol{a}_{s_k}}^{U_{s_k}}\|_1$ which increases with the number of qubits onto which the simulated gate acts. However, we are usually only interested in simulating two qubit gates, where this effect is small. Even with these drawbacks, Algorithm 1 constitutes a better alternative for many circuits than the bare NISQ implementation. Also, Theorem 2.4.1 provides a direct way to get a sense of whether implementing a QCQ interface will be too difficult to perform in practice, since we can obtain an upper bound on the number of shots required to perform an accurate simulation of a certain gate. We study relevant exemplary circuits with such trade-offs in the next sections.

Finally, note that $\mathcal{N}_{\rightarrow}^2$ is POVM-dependent. This is crucial to the efficiency of classical simulations [146, 147, 148]. For instance, in quantum Monte Carlo, it is known that the statistical overhead due to negative (quasi-)probabilities can be ameliorated [151] or even removed [150] by local base changes. Something similar applies here: the interface negativities depend not only on the choice of POVM, but also on how we construct the dual POVM elements.

Algorithm 1: Hybrid classical-quantum simulation with QCQ interfaces.

Input: ρ_0, C, O, ε, δ
Output: $O_{M_s}^*$ s.t. $\left|O_{M_s}^* - \mathrm{Tr}\left[O\,\rho_L\right]\right| \leq \varepsilon$ with probability at least $1-\delta$.
Initialize $O_{M_s}^* = 0$, $v = 1$, and M_s as in Equation (2.34).
for $m \in (1, \ldots, M_s)$ **do**
  for $k \in (1, \ldots, L)$ **do**
    if $k \in \{l_1, \ldots, l_L\}$ **then**
      Apply a QCQ interface for U_k on qubits s_k, obtaining the pair $(\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k})$;
      $v \leftarrow v \times v_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}$, with $v_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}$ as in Definition 2.4.3.
    else
      Apply the gate U_k on qubits s_k.
  Measure O, obtaining the measurement outcome (eigenvalue of O) o;
  $O_{M_s}^* \leftarrow O_{M_s}^* + o \times v$.
$O_{M_s}^* \leftarrow \frac{O_{M_s}^*}{M_s}$.

2.4.3 Numerical experiments

Here, we provide numerical experiments to validate the procedure outlined in Algorithm 1. Throughout the rest of this section, we take $\{M_{\boldsymbol{a}}\}_{\boldsymbol{a}}$ to be the Pauli-6 IC-POVM,

$$\{M_{\boldsymbol{a}}\}_{\boldsymbol{a}}^{\text{Pauli-6}} := \bigcup_{i=x.y,z} \left\{ \frac{1}{3} |\uparrow_i\rangle\langle\uparrow_i| , \frac{1}{3} |\downarrow_i\rangle\langle\downarrow_i| \right\},$$

where the vectors $|\uparrow_i\rangle , |\downarrow_i\rangle$ correspond to the eigenvectors of the Pauli operators with eigenvalue ± 1 respectively. Note that this POVM can be implemented in an experimental setting without the usage of ancilla qubits (see Appendix A.3).

For our simulations, we make use of full density matrix simulations and Locally Purified Density Operator tensor networks [159] (see Appendix D.3). For the latter, we choose the bond and Kraus dimensions D and κ, respectively, such that the simulation errors are under control, and we end up with a high fidelity ($> 99.9\%$) state approximation. To simulate realistic experimental settings, we apply noise to the two-qubit gates in our circuit. In particular, for each numerical experiment reported in this section, we implement noisy CNOTs throughout our circuits by applying single-qubit depolarizing channels $\mathcal{E} : \rho \mapsto \mathcal{E}(\rho)$ to both the control and target qubit of the CNOT gate. We apply depolarizing noise in the CNOTs with $\lambda_{\text{unit}} = 0.005$. These values correspond to experimentally realistic values of Google's Sycamore quantum processor [160]. At the end of the circuit we estimate observables $\text{Tr}\{\rho O\}$ exactly, i.e. without further sampling bitstrings but relying on the full state representation.

Since we are considering two-qubit gates for our numerical experiments, our interfaces only act on two-qubit systems. Hence, for our measurement-and-repreparation step, hence we only need to store bitstrings $\boldsymbol{a}_{s_l}$ of length 2, as well as the $6^2 \times 6^2$ overlap matrix $\tilde{T}^{U_k}$.

2.4.4 Random Walk Metropolis-Hastings for negativity minimization

To improve the sample complexity of our algorithm, we use a Monte Carlo algorithm to minimize the interface negativities. We first note that the matrix $\tilde{T}^{U_k}$ defined under Equation (2.30) defines a domain over which to optimize such negativity. Similar optimizations have been used for alleviating the sign problem in partition-function estimations [150, 151]. In our setting, we use a convenient parameterization of generalized inverse matrices by Rao [149] to propose new dual POVM elements for an adaptive random walk

Metropolis-Hastings algorithm. This allows us to decrease the multiplicative sample complexity overhead per interface by almost a factor of four relative to the canonical dual POVM (corresponding to $\tilde{T} = T^{-1}$, with T^{-1} the pseudo-inverse of T), which reduces the number of samples required by a factor of four.

In particular, we present a method to minimize the sample-complexity overhead by the interface of a unitary gate U exploiting the freedom in the choice of dual POVM, namely the choice of $\mathfrak{T}$ subject to Equation (D.4). For concreteness, we focus on the case where all POVM elements have the same trace, so $\mathrm{Tr}\left[M_{\boldsymbol{b}}\right] = 1/D$ for all $\boldsymbol{b}$, with D the number of POVM elements. Moreover, we optimize a modified version of the interface negativity n_U where, instead of maximizing $\left\|\tilde{T}^U_{\boldsymbol{a}}\right\|_1$ over $\boldsymbol{a}$ (as in Equation (D.9), we average $\left\|\tilde{T}^U_{\boldsymbol{a}}\right\|_1^2$ over $\boldsymbol{a}$). Such an average is the sample-complexity overhead directly given by the Hoeffding bound for when the sampled random variables can lie within segments of different lengths. The reason for this modification is that, while in Theorem D.1.1 we are interested in the worst-case complexity, here we are interested in the more practical problem of the average case.

For optimizing $\tilde{T}^U$ over $\mathfrak{T}$, we express it as $\tilde{T}^U = \mathfrak{T}_1\, T^U \mathfrak{T}_2$, with T^U given by $T^{U_k} := \mathrm{Tr}\left[U_k M_{\boldsymbol{a}_{s_k}} U_k^\dagger M_{\boldsymbol{b}_{s_k}}\right]$. Note that by not enforcing that $\mathfrak{T}_1 = \mathfrak{T}_2$, we are explicitly allowing for the more general case of possibly different input and output dual POVMs. Hence, we want to solve the constrained non-convex optimization

$$\min_{\mathfrak{T}} \quad \frac{1}{D}\sum_{\boldsymbol{a}} \left\|\left(\mathfrak{T}_1\, T^U \mathfrak{T}_2\right)_{\boldsymbol{a}}\right\|_1^2, \tag{2.35}$$

$$\text{s.t.} \quad T = T\mathfrak{T}_i T, \text{ for } i = 1, 2, \tag{2.36}$$

where $\left(\mathfrak{T}_1\, T^U \mathfrak{T}_2\right)_{\boldsymbol{a}}$ is a shorthand notation for the $\boldsymbol{a}$-th row of $\mathfrak{T}_1\, T^U \mathfrak{T}_2$ and $\left\|\left(\mathfrak{T}_1\, T^U \mathfrak{T}_2\right)_{\boldsymbol{a}}\right\|_1$ its l_1-norm. Equation (2.36) is a necessary but not sufficient condition for $\mathfrak{T}_i$ to be the Penrose-Moore pseudo-inverse of T. Indeed, such condition implies that $\mathfrak{T}_i$ is a so-called generalized inverse of T [161, 154]. So, the first question we need to consider is how to variationally explore the space of generalized inverses of T in a practical way.

Fortunately, this question has been previously studied. In particular, in Ref. [149] it was shown that for an arbitrary matrix $A \in \mathbb{R}^{m\times n}$ and given any particular generalized inverse A^- of it, every generalized inverse B^- can be obtained from some $C \in \mathbb{R}^{m\times n}$ by the map

$$B^-(C) := A^- + C - A^- A C A A^-. \tag{2.37}$$

That is, the entire space of generalized inverses is parameterized by C. This leads us to a practical way to obtain a random walk across the space of generalized inverses: In the first iteration, take the Penrose-Moore pseudo-inverse A^{-1} as starting generalized inverse and a randomly sampled C. This produces the first B^{-}. As inputs for the second iteration, use the first iteration's output B^{-} as generalized inverse and a fresh, independently sampled C. This produces a new B^{-}. Then continue to iterate.

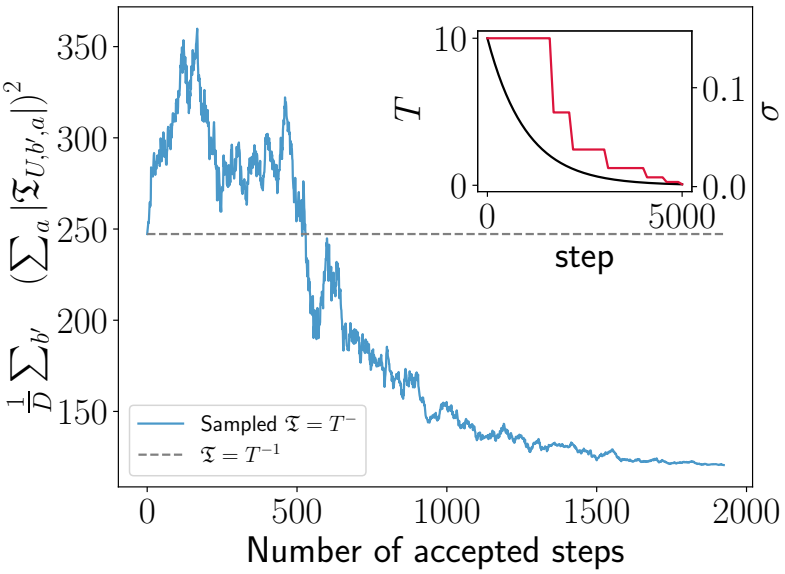


Figure 2.15: **Monte Carlo random walk for interface negativity optimization of the ZZ gate used in Section 2.4.6.** The total number of steps in the annealing schedule is 5000 . The gray dashed line indicates the mean average squared negativity of the pseudo-inverse, whereas the blue line indicates the one for the newly accepted $\mathfrak{T}$'s during the Monte Carlo random walk. The inset shows the adaptive scheme that fine-tunes the search with the temperature and variance, given in black and red, respectively.

Using this recipe for $A = T$ and $A^{-1} = T^{-1}$, we can ergodically explore the space of generalized inverses $\mathfrak{T}_i$ of T. In turn, the resulting random walk can be used as Markov Chain Monte Carlo dynamics for a simulated-annealing optimization [162, 163] that approximates a solution to Equation (2.35). More precisely, for each random walk iteration, we (probabilistically) accept or reject the newly produced $\mathfrak{T}_i$ via a standard Metropolis-Hastings algorithm with $\frac{1}{D}\sum_{\boldsymbol{a}} \left\| \left(\mathfrak{T}_1 T^U \mathfrak{T}_2\right)_{\boldsymbol{a}} \right\|_1^2$ as energy function.

For U a two-qubit gate and the Pauli-6 POVM, each dual-overlap matrix can be expressed as $\mathfrak{T}_i = \mathfrak{T}_i^{(1)} \otimes \mathfrak{T}_i^{(2)}$, where $\mathfrak{T}_i^{(1)}$ and $\mathfrak{T}_i^{(2)}$ are respectively the 6×6 real dual-overlap matrices of the two qubits on which U acts. We can independently sample all four matrices, $\mathfrak{T}_1^{(1)}$, $\mathfrak{T}_1^{(2)}$, $\mathfrak{T}_2^{(1)}$, and $\mathfrak{T}_2^{(2)}$. Hence, the search-space dimension is $4 \times 6 \times 6 = 142$.

For the simulated-annealing schedule, we take random matrices $C \sim \mathcal{N}(0, \sigma^2)^{6\times 6}$. We set the initial temperature to be $T = 10$ and decrease it with a factor 0.999 at each Monte Carlo step. In addition to the temperature, the Monte dynamics are controlled by the variance σ^2 of the normal distribution $\mathcal{N}(0, \sigma^2)^{6\times 6}$ for C. We start with a large initial $\sigma^2 = 0.1$ to coarsely explore the search space. However, as the temperature decreases, we want to refine the search without freezing the Monte Carlo dynamics. Therefore, we use an adaptive scheme where σ^2 is decreased according to the acceptance ratio. Specifically, we halve the value of σ if the acceptance ratio per 100 MCS is smaller than 0.23, a well-known heuristic for continuous-variable MCMC [164]. The search is terminated if the negativity decreases less than 10^{-2} after 100 accepted steps.

As a result, we consistently find dual frames whose averaged squared negativities are about half the value of the canonical dual frame from the pseudo-inverse (see Figure 2.15). This is also observed to greatly improve the sample complexity in practice (see Figure 2.16).

2.4.5 Simulation of long-range maximal Bell violations

As a proof of principle experiment, we show that a maximally entangled state simulated with our method attains the maximal violation of the Clauser-Horne-Shimony-Holt (CHSH) inequalities as expected. The CHSH inequalities constrain a set of four correlators in an Alice (A) and Bob (B) type experiment and provide a condition to check if the correlations between the observations of Alice and Bob can be explained by a local theory, or necessitate a non-local theory such as quantum mechanics [165]. Consider the quantity

$$\begin{aligned} f_{\text{CHSH}}(A, B) =& C^{00}(A, B) + C^{01}(A, B) + C^{10}(A, B) \\ & - C^{11}(A, B) \end{aligned}$$

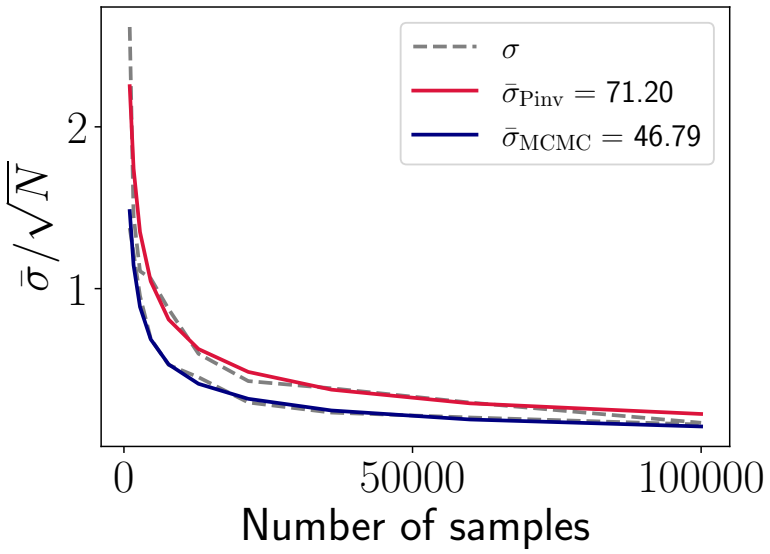


Figure 2.16: **Improvement of energy-estimator variance for the 8-qubit TFIM circuit experiment of Figure 2.18b**. Sample variance is estimated over 50 runs. The red line shows sample variance corresponding to the canonical dual frame given by the pseudo-inverse of T. In blue, we see the variance of the energy corresponding to the dual frame obtained from the Monte Carlo search.

where

$$C^{00}(A,B) = \frac{1}{\sqrt{2}}(-\langle Z_A Z_B\rangle - \langle Z_A X_B\rangle)$$
$$C^{01}(A,B) = \frac{1}{\sqrt{2}}(-\langle X_A Z_B\rangle - \langle X_A X_B\rangle)$$
$$C^{10}(A,B) = \frac{1}{\sqrt{2}}(\langle Z_A Z_B\rangle - \langle Z_A X_B\rangle)$$
$$C^{11}(A,B) = \frac{1}{\sqrt{2}}(\langle X_A Z_B\rangle - \langle Z_A X_B\rangle)$$

are the correlations obtained from the state shared by Alice and Bob. The observables X and Z are the Pauli matrices. We call $f_{\text{CHSH}}(A,B)$ the Bell polynomial. The CHSH inequality is given by $f_{\text{CHSH}}(A,B) \leq 2$, which if satisfied, implies that a local hidden variable theory can explain the observed correlations. On the other hand, for $f_{\text{CHSH}}(A,B) > 2$

we have to invoke quantum theory to explain the correlations. The maximum value of $f_{\mathrm{CHSH}}(A, B)$ is $2\sqrt{2}$ which is obtained for a maximally entangled two qubit state.

We consider the Bell state $|\Phi^+\rangle = \frac{1}{2}(|00\rangle + |11\rangle)$ which has the maximum CHSH violation $f_{\mathrm{CHSH}}(A, B) = 2\sqrt{2}$. We consider the case where the state is prepared on two qubits separated by a distance d. Applying the CNOT between these distant qubits requires implementing a swap chain to bring the two states close together. In Figure 2.17 we compare the CHSH violation of the Bell state simulated with our algorithm and one prepared with a circuit containing a noisy swap chain. We see that the CHSH violation is only affected by the statistical fluctuations of our method and therefore approximates the maximum value independent of the distance between the qubits.

2.4.6 The Transverse Field Ising-model circuit

As a practical example of implementing our method in an experimentally realistic setting, we investigate the ground state of a prototypical model for quantum magnetism: the transverse field Ising-model (TFIM) on a one dimensional ring. The Hamiltonian of the TFIM for the 1D chain is given by

$$H_{\mathrm{TFIM}} = -\sum_{i=1}^{n} \left[Z_i Z_{i+1} + g X_i \right], \tag{2.38}$$

where we assume periodic boundary conditions and set $g = 1$. The ground state of H can be approximated reliably with a depth $L = n/2$ HVA circuit (see Section 2.2) [40, 52, 1].

To evaluate the accuracy of the state reconstruction, we compare the finite statistics estimator of the energy $\left\langle \hat{H}_M \right\rangle$ from our algorithm with the ground-state energy $E_{\mathrm{gs}} = \langle \psi_{\mathrm{gs}} | H | \psi_{\mathrm{gs}} \rangle$ from exact diagonalization.

We consider three setups: First, we consider the $n = 4$ and $n = 8$ qubit TFIM chains where the last long range ZZ gate (in the 2nd and 4th layer respectively) is classically simulated with our algorithm (See Figure 2.18). Next, we apply our method twice for the same circuits, with simulation of both the last and first-to-last long range ZZ gate (See Figure 2.19). Finally, we consider the ground state of an $n = 20$ TFIM chain, where we only apply the first two layers of the circuit and simulate the second long range ZZ gate (See Figure 2.20). For all simulations, we confirm that we can greatly improve the final energy estimates by making use of QCQ interfaces at the cost of M_s measurement-and-reprepare steps.

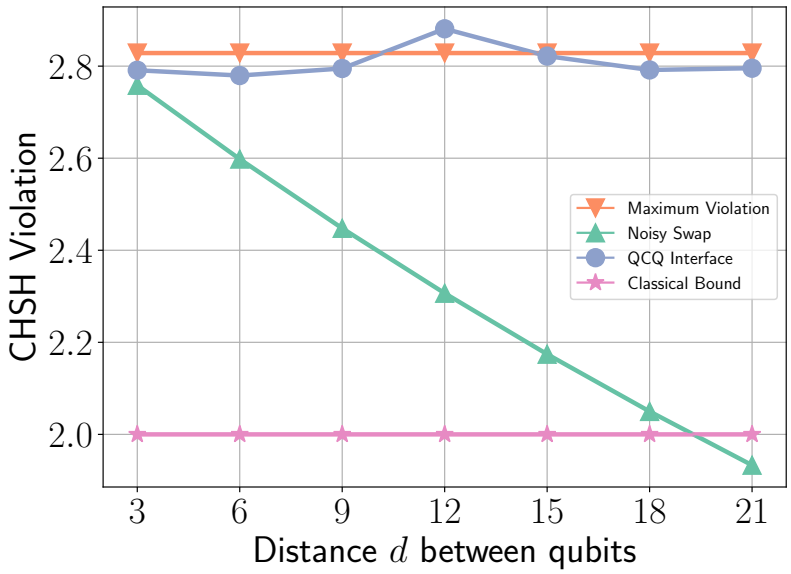
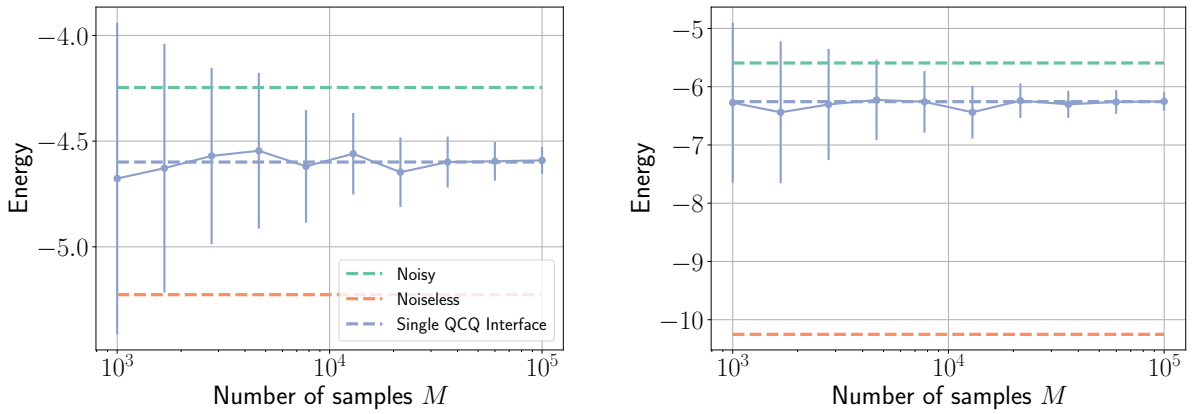


Figure 2.17: **CHSH violation as a function the number of qubits.** These results where obtained with an LPDO simulation where $D = 12$ and $\kappa = 24$. In addition to the gate noise, we apply a depolarizing channel to simulate measurement noise with $\lambda_{\text{meas}} = 0.01$ and repreparation noise with $\lambda_{\text{reprep}} = 0.005$. The classical bound (pink) and maximal violation (green) are 2 and $2\sqrt{2}$ respectively for all d. We see that the violation in the noisy circuit (green) decreases linearly with the number of qubits as a result of the $4(d-2)+1$ noisy swap gates required to prepare the state. Our algorithm provides the maximum CHSH violation up to statistical fluctuations independent of the distance between the qubits. This comes at a cost of sampling $M = 60000$ measurement-and-repreparation steps to estimate the violation.

2.4.7 Conclusion

We have introduced a rigorous framework of hybrid quantum-classical interfaces for quantum-circuit simulations. We applied a specific variant of these gadgets – which we dub quantum-classical-quantum (QCQ) interfaces – to simulate long-range gates in low-connectivity devices without using swap-gate ladders. QCQ interfaces replaces an experimentally problematic gate (e.g. a very long-range one) by single-qubit random measurements and state-preparations sampled according to a classical quasi-probability simulation of the ideal target gate. This procedure eliminates long swap-gate ladders which would otherwise be



(a) (b)

Figure 2.18: **Comparison of QCQ interface simulation with both noisy and noiseless TFIM circuits.** (a) $n = 4$ and (b) $n = 8$ qubits obtained with a full density matrix simulation. Each dot represents the average energy $\mathbb{E}\left[\left\langle \hat{H}_M \right\rangle\right]$ estimated over 50 separate instances. The error bars indicate the standard deviation. As the number of samples M_s increases, the statistical fluctuations of our method become small in accordance with the central limit theorem. We can determine the scaling of the size of the error bars by fitting $\sigma = \bar{\sigma}/\sqrt{N_{\text{samples}}}$. While for 4 qubits $\bar{\sigma} \approx 27.8$, for 8 qubits we have $\bar{\sigma} \approx 76.5$. This scaling only depends on the mean negativity, which differs between the two circuits because we apply a different ZZ rotation on each circuit. The energy of the noiseless circuit (orange dashed line) corresponds to the ground-state energy E_{gs}. The noisy circuit (green dashed line) shows the energy obtained when we apply depolarizing channels with $\lambda_{\text{unit}} = 0.005$ to the CNOT gates in the circuit. We see that for both the 4 and 8 qubit our algorithm provides a significant improvement on the final estimated energy of the circuit for a reasonable number of measure-and-reprepare steps. In (b) we observe that the large number of number of noisy CNOTs dominates the simulation, hence the improvement is not as significant as for 4 qubit.

required to physically synthesize the target gate. This results in a drastic increase in gate fidelity. The final output of the scheme is an estimate of the expectation value of a given observable on the output of the target high-connectivity circuit.

The quasi-probability distribution used is given by a POVM representation of the gate simulated at each interface. As any sampling scheme based on non-positive quasi-

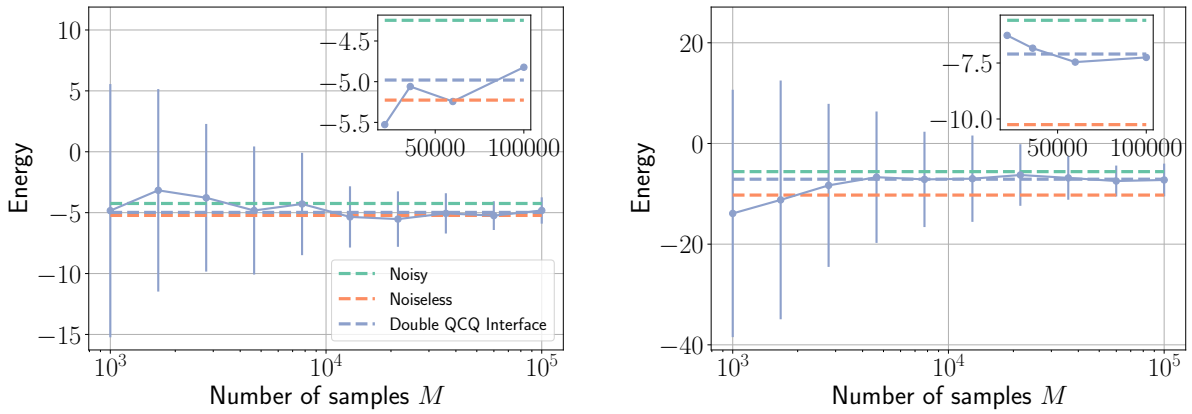


Figure 2.19: **Comparison of double QCQ interface simulation with both noisy and noiseless TFIM circuits.** (a) $n = 4$ and (b) $n = 8$ qubits. These results were obtained with a full density matrix simulation. In (a), we see that we can almost approximate the true ground-state energy of the 4 qubit state, because the only noisy operations are the 12 CNOTs required for implementing the 6 nearest neighbor ZZ gates in layers 1 and 2. In (b) we see a more significant improvement over the energies from Figure 2.18b, but still the noise dominates. Since we apply the QCQ method twice, the standard deviation $\sigma = \bar{\sigma}/\sqrt{N_{\text{samples}}}$ of the error bars increases quadratically, as per Equation (2.33). We find $\bar{\sigma} \approx 333.1$ and $\bar{\sigma} \approx 856.8$ for 4 and 8 qubits respectively.

probabilities, our method suffers from the sign problem. Because of this, the overall sample complexity grows exponentially with the number of interfaces applied. However, the statistical overhead per interface is independent of the on-chip distance between the qubits on which the interface acts. To ameliorate the sign problem, we developed a Metropolis-Hastings simulated-annealing algorithm based on random walks in the space of dual POVMs. This allowed us to decrease the statistical overhead per interface by almost a factor of two over that of the canonical dual POVM. This is potentially interesting on its own beyond the current scope and further optimization is possible. All together, we show that any circuit with a limited number of gates to cut out can be simulated at the expenses of a moderate overall overhead in sample complexity. As examples, we explicitly considered a Bell-state preparation circuit for two qubits increasingly far apart and variational ground-state solvers for the transverse-field Ising model on ring lattices. The former involves a single long-range gate, whereas the latter contains one such gate per variational

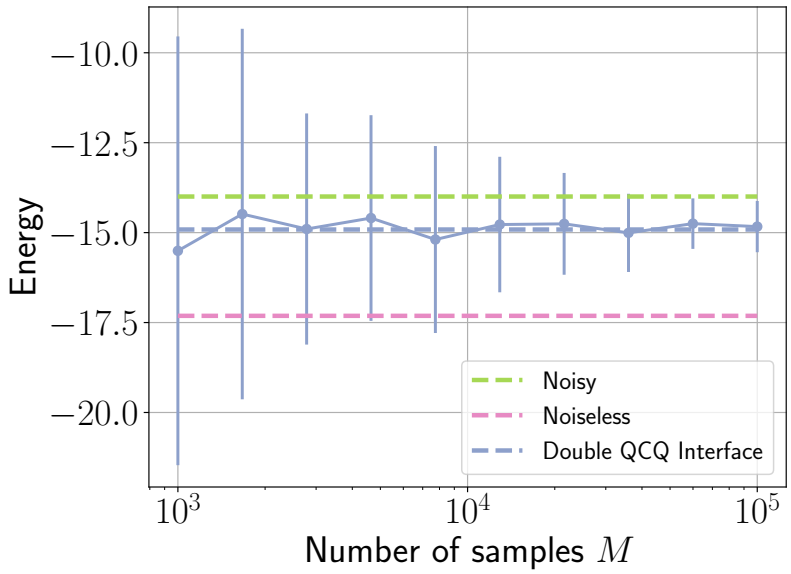


Figure 2.20: **Comparison of a QCQ interface simulation with both noisy and noiseless circuits for a 20 qubit TFIM circuit.** These results where obtained with an LPDO simulation where $D = 50$ and $\kappa = 50$. Only two of the 8 layers of the circuit are simulated here, to keep simulation errors under control. The sample variance $\hat{\sigma} \approx 195.0$.

layer.

Interestingly, the quasi-probability approach we use here is not the only route to gate simulation. In [166], similar in spirit to [145], quantum circuits are simulated via Monte Carlo simulation. However, instead of using the language of frames, a Hubbard-Stratanovich transformation is applied. In this context, the sign problem manifests itself in the form of a complex action that inhibits the efficient simulation of a large number of gates. A potential fruitful direction of future work would be to investigate the limits of this alternative gate simulation approach (see Appendix D.4).

Importantly, our method requires platforms supporting mid-circuit measurements and state preparations, which are readily provided by some quantum hardware companies such as, e.g., IBM and Honeywell [110, 111]. This may pave the way to implement our method in a practical setting in the near future. However, the efficacy of our method will rely on the speed and accuracy of intermediate measurements. Although our numerical experiments for the CHSH violation indicate that our algorithm is insensitive to imperfect measurements,

slow measurements may be more problematic since NISQ devices only have a limited coherence time.

Finally, we emphasize that our framework is not restricted to connectivity boosts only. It could also be applied to any gate that is too noisy for a given platform or combined with error-correcting codes to remove a gate that is particularly difficult to implement fault-tolerantly by the code. Another interesting application that will be studied elsewhere is circuit-depth boosts, where a deep circuit is simulated by shallower experimental circuits together with classical simulations of entire slices of the target circuit. In conclusion, our framework provides a versatile toolbox for both error-mitigation and circuit boosts well suited for noisy, intermediate-scale quantum hardware.

Chapter 3

Riemannian geometry in Variational Quantum Computing

3.1 Differential Geometry and Lie groups

To establish notation, we briefly summarize some of the key concepts in differential geometry needed for our purposes. There exist many excellent references on the topic, see [167, 168] for the physicist-friendly references and [169, 170] for the more technical expositions on the subject. Since this is a physics thesis, we will not spend too much time discussing the beautiful mathematics of differential geometry, and only introduce the necessary concepts.

3.1.1 Differentiable Manifolds

An n-dimensional (topological) manifold $\mathcal{M}$ is a set that locally looks like $\mathbb{R}^n$. This local description is given by charts φ_a, which smoothly map open subsets U_a of the manifold onto coordinate patches in $\mathbb{R}^n$. So if $\{x^i\}$ are the coordinates in $\mathbb{R}^n$, and $p \subseteq U_a \in \mathcal{M}$, then

$$\varphi(p) = \boldsymbol{x}. \tag{3.1}$$

If all charts between two subsets of the manifold are compatible, the manifold is differentiable. For the purposes of this work, all manifolds will be C^∞ differentiable. The beauty of differential geometry is that you can work in a "coordinate-free" description where you

never choose coordinates, however, due to the applied nature of this thesis we will mostly work in an explicit coordinate chart.

3.1.2 Tangent spaces

In calculus, we learn about function and its derivatives on a real vector space $\mathbb{R}^n$. Given that differentiable manifolds can be described locally via charts, many ideas of calculus carry over directly to the manifold setting, with the caveat that they are defined locally.

A prime example of this is the idea of a *tangent space* $T_p\mathcal{M}$ at a point $p \in \mathcal{M}$. We can think of the tangent space as a collection of vectors that provide the possible directions one can move in on the manifold from point p. Consider a curve $\gamma : I \to \mathcal{M}$ where $I = (-a, a)$ is an open subset of $\mathbb{R}$ and $\mathcal{M}$ is a differentiable manifold. We can construct a curve on $\mathcal{M}$ so that $\gamma(0) = p$. Then we can ask, what is the derivative of a function $f : \mathcal{M} \to \mathbb{R}$ in the direction of this curve? By working in a chart (U, ϕ), $p \in U$ and $\phi(p) = \{x^i\}$ where x^i is the ith coordinate of the vector $\phi(p)$, we find

$$\left.\frac{df(\gamma(t))}{dt}\right|_{t=0} = \frac{\partial(f \circ \phi^{-1})}{\partial x^i}\left.\frac{d(\phi \circ \gamma)(t)}{dt}\right|_{t=0} = \frac{\partial f}{\partial x^i}\left.\frac{dx^i(\gamma(t))}{dt}\right|_{t=0}. \tag{3.2}$$

This allows us to define a tangent vector at p as

$$\boldsymbol{v} = \sum_i^n v^i \frac{\partial}{\partial x^i} \equiv \sum_i^n v^i \partial_i, \quad v^i = \left.\frac{dx^i(\gamma(t))}{dt}\right|_{t=0}. \tag{3.3}$$

So a tangent vector is an operator that differentiates a function in the direction of some curve $\gamma(t)$ going through a point p as $\boldsymbol{v}(f)(p)$. There exist many such curves, and these curves form an equivalence class. The collection of these equivalence classes is then called the tangent space $T_p\mathcal{M}$ of $\mathcal{M}$ at p, which is a vector space that be spanned by a basis of differential operators $\{\partial/\partial x_i\} \equiv \{\partial_i\}$. This may be somewhat surprising, since we are not used to think about derivatives as elements that can span a vector space, however this is what tangent vectors represent: operators that can differentiate a function on $\mathbb{R}^n$.

We can change the coordinates with a simple transformation. To see this, consider two tangent vectors $\boldsymbol{v}, \boldsymbol{u}$ defined in different charts (U_a, φ_a), (U_b, φ_b) such that $p \in U_a \cap U_b$ and

$\varphi_a(p) = \boldsymbol{x}$, $\varphi_b(p) = \boldsymbol{y}$. Then

$$\sum_i^n v^i \frac{\partial}{\partial x^i} = \sum_j^n u^j \frac{\partial}{\partial y^j}$$
$$v^i = \sum_j^n u^j \frac{\partial x^i}{\partial y^j}. \tag{3.4}$$

The object $\partial x_i / \partial y_i$ should look familiar: it is the *Jacobian matrix* of the coordinates $\boldsymbol{x}$ with respect to the coordinates $\boldsymbol{y}$ which tells us how tangent vectors in one coordinate system transform to another coordinate system.

3.1.3 Cotangent vectors (one-forms)

Since $T_p\mathcal{M}$ is a vector space, there exists a dual vector space $T_p^*\mathcal{M}$ called the cotangent space. Elements of the cotangent space are called cotangent vectors or *one-forms* $\boldsymbol{\omega}$: $T_p\mathcal{M} \to \mathbb{R}$, which accept a tangent vector and produce real number. Similar to the tangent vectors in Equation (3.3), a one-form can be expanded into a basis $\{dx^i\}$ that is dual to $\{\partial_i\}$,

$$\partial_i dx^j = \delta_i^j, \tag{3.5}$$

so that

$$\boldsymbol{\omega} = \sum_i^n \omega_i dx^i. \tag{3.6}$$

One-forms are also known as covectors. In the following, we will drop the explicit basis vectors ∂_i and dx^i and denote the components of tangent vectors $\boldsymbol{v}$ and one-forms $\boldsymbol{\omega}$, with an upper index v^i and lower index ω_i, respectively. We will also make use of the Einstein summation convention,

$$\sum_i v_i u^i \equiv v_i u^i. \tag{3.7}$$

3.1.4 Riemannian manifolds

At this point, we know how to take a manifold $\mathcal{M}$, equip it with coordinates and calculate derivatives at points of the manifold. In order to talk about distances between points we

need to define a *metric*, which is a positive-definite bilinear tensor that takes two points in a tangent space and produces a scalar. Once we equip $\mathcal{M}$ with a metric, it becomes a *Riemannian manifold*.

Specifically, a Riemannian manifold is a manifold $\mathcal{M}$ equipped with a symmetric, non-degenerate metric $g : T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$. Hence, the metric takes two vectors in the tangent space at a point p and produces a positive number. For example, given $\boldsymbol{v}, \boldsymbol{u} \in T_p\mathcal{M}$ we get

$$g(\boldsymbol{v}, \boldsymbol{u}) = v^i u^j g_{ij}.$$

The components g_{ij} of g are always defined with respect to a basis $\{\partial^i\}$, so $g_{ij} = g_{ij}(\partial^i, \partial^j)$ and we can represent g as an $n \times n$ matrix. Consider now the function $g(\boldsymbol{v},) : T_p\mathcal{M} \to \mathbb{R}$. Remember that a one-form $\boldsymbol{\omega}$ is function $\boldsymbol{\omega} : T_p\mathcal{M} \to \mathbb{R}$. Hence, we can use the metric to turn tangent vectors into one-forms:

$$g(\boldsymbol{v},) = v^i g_{ij} = v_j,$$

so that

$$g(\boldsymbol{v}, \boldsymbol{u}) = v_j u^j.$$

Since the metric is symmetric and non-degenerate, we can define the inverse metric g^{-1} as providing a distance measure between one-forms,

$$g^{-1}(\boldsymbol{\omega}, \boldsymbol{\eta}) = \omega_i \eta_j g^{ij} = \omega_i \eta^i.$$

Remember that the metric is defined locally at each point p. In order to calculate distances between two points p and p', we need to define a curve that connects these points. If we take $\gamma(t)$ to be the curve such that $\gamma(0) = p$ and $\gamma(1) = p'$, then we can calculate the distance between these points by adding up all infinitesimal line elements along the curve

$$L = \int_0^1 \sqrt{g(\gamma'(t), \gamma'(t))} dt. \tag{3.8}$$

3.1.5 Riemannian gradients

Now that we have defined a metric, we can talk in a meaningful way about the concept of a gradient. To do so, we will consider a function $f : \mathcal{M} \to \mathbb{R}$ at a point p and a tangent vector $\boldsymbol{v}$. We can define a curve $\gamma : (-a, a) \to \mathcal{M}$ with $\gamma(0) = p$ such that $E(\gamma(t)) = \text{constant}$ for all t. Differentiating with respect to t then gives

$$(\partial_i f(\gamma(t)))\gamma'(t)\big|_{t=0} = 0,$$

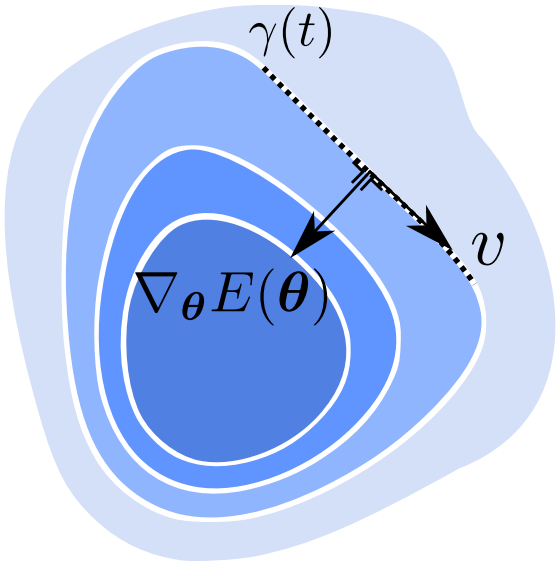


Figure 3.1: **Schematic depiction of the gradient of a function $E(\boldsymbol{\theta})$ on $\mathbb{R}^2$.** The gradient is always orthogonal to the curve $\gamma(t)$ where the function is constant. Here, $\boldsymbol{v}$ corresponds to $\gamma'(t)$ at some point p.

In other words, the gradient of a function produces a vector orthogonal to the level curves through a point (see Figure 3.1). To define the gradient in local coordinates, we need the metric to establish what it means for two vectors to be orthogonal.

This can be achieved with the following construction. Instead of the curve $\gamma(t)$ on $\mathcal{M}$, we can pick local coordinates $\boldsymbol{x}$ around p and establish the notion of a gradient in an explicit coordinate system. To start, we consider the differential df, which takes the function f and creates a one-form in the chosen coordinate system $\{\partial_i\}$,

$$df = \partial_i f(x^1, \ldots, x^n) dx^i.$$

When we apply this function to a tangent vector, we get

$$df(\boldsymbol{v}) = \partial_i f(x^1, \ldots, x^n) v^j dx^i.$$

Remember that $v^j \equiv v^j \partial_j$ and $\partial_j dx^i = \delta^i_j$, which gives

$$df(\boldsymbol{v}) = \partial_i f(x^1, \ldots, x^n) v^j \delta^i_j. \tag{3.9}$$

In other words, the differential gives us the rate of change of the function f, at the point p in the direction of v. If we change coordinates, $\boldsymbol{x} \to \boldsymbol{y}$, we can always use the Jacobian

matrix of Equation (3.4) to account for the change. The (Riemannian) gradient is then implicitly defined as

$$g(\text{grad } f, \boldsymbol{v}) = df(\boldsymbol{v}), \tag{3.10}$$

which is known as the compatibility condition. In other words the Riemannian gradient is a tangent vector that preserves the inner product with any other tangent vector $\boldsymbol{v}$. We find

$$(\text{grad } f)^i v^j g_{ij} = \partial_i f(x^1, \dots, x^n) v^j \delta^i_j,$$

so

$$(\text{grad } f)^i g_{ij} = \partial_i f(x^1, \dots, x^n) \delta^i_j.$$

So in vector form

$$\text{grad } f = g^{-1} \cdot (\nabla f), \tag{3.11}$$

where we defined $(\nabla f)^i = \partial_i f(x^1, \dots, x^n)$. If g is the euclidean metric, then the Riemannian gradient becomes the standard gradient operator in calculus. However, Equation (3.11) is works for any manifold and a valid metric g.

Analogous to the gradient flow in a Euclidean space that we saw in Section 2.1.2, we can use the Riemannian gradient to construct a Riemannian gradient flow

$$\dot{\boldsymbol{x}} = \text{grad } f(\boldsymbol{x}).$$

Riemannian gradient flows show up in different fields of optimization. For example, consider a probability density function $p(x|\boldsymbol{\theta})$ for a random variable X which takes values in Ω, parameterized by $\boldsymbol{\theta} \in \mathbb{R}^n$. The manifold of probability distributions p is smooth, and the induced metric on the coordinate space $\mathbb{R}^n$ is given by the Fisher metric [171]:

$$g_{ij} = -\int_\Omega \frac{\partial^2 \log p(x|\boldsymbol{\theta})}{\partial\theta_i \, \partial\theta_j} p(x|\boldsymbol{\theta}) \, dx. \tag{3.12}$$

If we plug this metric into Equation (3.11), we obtain Natural Gradient Descent [172], a well-known second order optimization method for statistical models.

The quantum analogue of this can be constructed as follows. Global phases are physically unobservable, so quantum states are actually *rays* in a projective complex Hilbert space $\mathbb{C}\text{P}^{N-1}$. This is formalized via the equivalence relation

$$|\psi\rangle \sim \lambda \, |\psi\rangle \,, \quad \lambda \in \mathbb{C}, |\psi\rangle \in \mathcal{H} \tag{3.13}$$

Given some parameterization $|\psi(\boldsymbol{\theta})\rangle$, the induced metric on $\mathbb{C}\mathrm{P}^{N-1}$ is given by (see [173, 174])

$$g_{ij} = \langle \partial_i \psi(\boldsymbol{\theta}) | \partial_j \psi(\boldsymbol{\theta}) \rangle - \langle \partial_i \psi(\boldsymbol{\theta}) | \psi(\boldsymbol{\theta}) \rangle \langle \partial_j \psi(\boldsymbol{\theta}) | \psi(\boldsymbol{\theta}) \rangle , \tag{3.14}$$

which is known as the Fubini-Study metric. The resulting Riemannian gradient flow of Equation (3.11) is known as Stochastic Reconfiguration in Variational Monte Carlo [175] and Quantum Natural Gradient in variational quantum computing [26].

3.1.6 Lie groups

A Lie group is differentiable manifold that is equipped with a group structure. Remember that a group G is a set of objects that satisfies the following properties

1. $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ for all $a, b, c \in G$ (associativity)
2. $e \cdot g = g$ (existence of the identity)
3. $g^{-1} \cdot g = e$ (existence of inverse)

For this group to be a differentiable manifold, we need both the group product $g\cdot$ and its inverse $g^{-1}\cdot$ to be differentiable in the sense of Section 3.1.1. Lie groups are omnipresent in physics. They are at the foundation of the standard model, can be used to describe the internal symmetries of spin systems and are at the heart of universality theorems in quantum computing.

In this work we will only be considered with finite-dimensional groups. We will always work with the fundamental matrix representation of the group (the smallest possible representation of the group), unless specified otherwise. The most relevant ones in the quantum computing context are the following subgroups of the group of complex invertible matrices $\mathrm{GL}(N, \mathbb{C})$

$$\mathrm{U}(N) = \{X \in \mathbb{C}^{N\times N} | X^\dagger X = I\}$$
$$\mathrm{O}(N) = \{X \in \mathbb{R}^{N\times N} | X^T X = I\},$$

which are called the *unitary* and *orthogonal* group, respectively. We also have the *symplectic group*

$$\mathrm{Sp}(2N, \mathbb{C}) = \{X \in \mathbb{C}^{2N\times 2N} : X^T J X = J\}, \quad J = \begin{pmatrix} 0 & I_N \\ -I_N & 0 \end{pmatrix}. \tag{3.15}$$

In the quantum context we typically work with the compact counterpart of $\mathrm{Sp}(2N, \mathbb{C})$,

$$\mathrm{Sp}(N) = \mathrm{Sp}(2N, \mathbb{C}) \cap \mathrm{U}(2N), \tag{3.16}$$

which is the set of unitary operators that is also symplectic.

3.1.7 $\mathrm{SU}(N)$ and its Lie algebra

Consider the special unitary Lie group $\mathrm{SU}(N)$:

$$\mathrm{SU}(N) = \{X \in \mathbb{C}^{N\times N} | X^\dagger X = I,\, \det\{X\} = 1\}. \tag{3.17}$$

In general, quantum gates belong to the unitary group $\mathrm{U}(N)$, which drops the determinant condition and thus allows for an additional global phase. Restricting ourselves to $\mathrm{SU}(N)$ therefore is physically equivalent. Consider a curve $X(t) : \mathbb{R} \to \mathrm{SU}(N)$, $t \in (-a, a)$ such that $X(0) = I$ and $d/dtX(t)|_{t=0} \equiv \dot{X}(0) = \Omega$. If we differentiate the unitarity condition with respect to t, we obtain

$$\begin{aligned} \frac{d}{dt}(X^\dagger(t)X(t))\big|_{t=0} &= 0 \\ \dot{X}^\dagger(t)X(t) + X^\dagger(t)\dot{X}(t)|_{t=0} &= 0 \\ \Omega^\dagger + \Omega &= 0, \end{aligned}$$

and so we find that $\Omega^\dagger = -\Omega$, i.e., Ω is a skew-Hermitian matrix.

The matrices Ω which result from such curves are elements of the *Lie algebra*

$$\mathfrak{su}(N) = \{\Omega \in \mathbb{C}^{N\times N} | \Omega^\dagger = -\Omega,\, \mathrm{Tr}\,\{\Omega\} = 0\}. \tag{3.18}$$

The second condition, $\mathrm{Tr}\{\Omega\} = 0$, can be found by realizing that the Lie algebra is connected to a Lie group via the exponential map: $e^{tX} \in \mathrm{SU}(N)$, $\forall t \in \mathbb{R}$ and $X \in \mathfrak{su}(N)$ and so $1 = \det\{e^X\} = e^{\mathrm{Tr}\{X\}}$. Furthermore, from Equation (3.2), we can see that we have constructed the tangent space $T_I\,\mathrm{SU}(N)$; the tangent space of the special unitary group at the identity. To keep things in the realm of differential geometry, we will defer more background and details about Lie algebras to Chapter 4, since they are the central object of study there. For now, it's only important to know that $\mathfrak{su}(N)$ is the tangent space of the unitary group at the identity.

Instead of considering a curve going through the identity at $t = 0$, we can consider a curve $X(t)$ such that $X(0) = U$ and $\dot{X}(0) = \Xi$. We can repeat the same argument as before,

$$\begin{aligned}\dot{X}^\dagger(t)X(t) + X^\dagger(t)\dot{X}(t)|_{t=0} &= 0\\ \Xi^\dagger U + U^\dagger \Xi &= 0.\end{aligned}$$

We quickly see that this equation is solved by setting

$$\Xi = \Omega U,$$

with $\Omega \in \mathfrak{su}(N)$. Hence, the tangent space at U is given by

$$T_U \operatorname{SU}(N) = \{\Omega U |\, \Omega \in \mathfrak{su}(N)\}.$$

In other words, we can always move to the tangent space at a point U from the tangent space at the identity by multiplying with U on the left. Note that the left multiplication is a choice; we could have also picked $\Xi = U\Omega$, but this does not affect the final results.

3.1.8 The exponential map and its gradient

The following is due to [176, Chapter 1, Theorem 5]. Let G be a matrix Lie group $G \in \operatorname{GL}(N, \mathbb{C})$ with a corresponding Lie algebra $\mathfrak{g}$. Define conjugation by $h \in G$ to be the transformation $c_h : G \to G$ given by $c_h(g) = hgh^{-1}$. Note that c is an (inner) automorphism of G, since it is an isomorphism from G onto itself. Let $\exp : \mathfrak{g} \to G$ be the exponential map from the Lie algebra to the group. Taking $X \in \mathfrak{g}$ and $t \in \mathbb{R}$, the differential of the conjugation map at the identity is

$$d(c_h)(X) = \frac{d}{dt}(he^{tX}h^{-1})\bigg|_{t=0} = hXh^{-1},$$

which maps an element of the Lie algebra to another element of the Lie algebra. This map is called the *adjoint representation* $\operatorname{Ad}_h : \mathfrak{g} \to \mathfrak{g}$ and takes $X \mapsto hXh^{-1}$. Given $X, Y \in \mathfrak{g}$, we compute

$$\begin{aligned}\frac{d}{dt}\operatorname{Ad}_{e^{tX}} Y\bigg|_{t=0} &= \frac{d}{dt} e^{tX} Y e^{-tX}\bigg|_{t=0}\\ &= XY - YX\\ &= [X, Y]\\ &= \operatorname{ad}_X Y.\end{aligned}$$

The operator ad: $\mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ is the Lie bracket on $\mathfrak{g}$, which for our purposes will be the standard commutator. It now follows that

$$\begin{aligned}\frac{d}{dt}\mathrm{Ad}_{e^{tX}}Y &= \frac{d}{dt}\left(e^{tX}Ye^{-tX}\right) \\ &= Xe^{tX}Ye^{-tX} + e^{tX}Ye^{-tX}(-X) \\ &= \mathrm{ad}_X(\mathrm{Ad}_{e^{tX}}Y).\end{aligned}$$

With the boundary condition $\mathrm{Ad}_{e^{tX}}|_{t=0} = \mathrm{Id}$ we find that the above differential equation is solved by

$$\mathrm{Ad}_{e^X} = e^{\mathrm{ad}_X}, \tag{3.19}$$

at $t = 1$.

3.2 Riemmannian Gradient Flows in variational quantum circuits

As discussed in Section 2.1.1, in a typical VQE setup one calculates gradients with respect to gate parameters in a quantum circuit to minimize a cost function that depends on the variational state. Since these approaches often involve minimizing non-convex cost functions, the choice of optimizer can greatly affect the result [177]. Unlike in deep learning, where backpropagation can remain effective despite a large number of parameters, calculating gradients in a variational quantum circuit quickly becomes inefficient. This is due to the fact that the gradients for single parameters cannot be calculated concurrently, but require additional circuit evaluations for each parameter [30, 31].

Gradient-based methods can be improved by considering additional structure of the model under consideration. For instance, when dealing with a statistical model, one can make use of the Fisher information to quantify the statistical distance between probability distributions [178]. This induces a metric on parameter space, which provides the direction of steepest descent with respect to the information geometry [172]. The resulting gradient is called the natural gradient, and forms the basis of a second order optimization algorithm called natural gradient descent. This method is used in deep learning [179, 180] but can also be extended to variational quantum Monte Carlo [181] and variational quantum circuits, where the distance between rays in Hilbert space provides an analogue of the Fisher information [26, 182].

Optimization algorithms that rely on the Fisher information fall into the category of Riemannian optimization algorithms [183, 184]. However, they are limited to optimizing over a real parameter space $\mathbb{R}^d$ with a non-Euclidean metric. Riemannian optimization has a much broader application: we can consider minimizing a function over a differentiable manifold $\mathcal{M}$ equipped with a non-degenerate, positive metric. This construction is more general, and allows one to take the structure of the manifold into account during the optimization. Such applications have been considered in the context of quantum control [185, 186, 187, 188, 189, 190], tensor networks [191, 192] or optimization of neural networks [193, 194, 195, 196, 197]. In the quantum circuit setting, the Riemannian manifold perspective has been considered to study the computational complexity of constructing specific circuits by approximating geodesics on the unitary group [198, 199]. In this section, we introduce the optimization of quantum circuits over the special unitary group $\mathrm{SU}(N)$ using Riemannian gradient flows [185]. We show the resulting algorithm can produce quantum circuits with favorable optimization properties but which may be exponentially deep. To obtain a practically feasible circuit optimizer, we make approxima-

tions that keep gate costs under control. We explore several toy problems to illustrate the properties of the resulting exact and approximate Riemannian gradient flow.

3.2.1 Gradient flows in quantum circuits

Consider the cost function of Equation (2.9),

$$E(\boldsymbol{\theta}) = \mathrm{Tr}\left\{HU(\boldsymbol{\theta})\rho_0 U(\boldsymbol{\theta})^\dagger\right\}, \tag{3.20}$$

where $U(\boldsymbol{\theta})$ is a variational quantum circuit with d parameters. To solve the optimization problem $\min_{\boldsymbol{\theta}} E(\boldsymbol{\theta})$, we can consider the flow

$$\dot{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}), \tag{3.21}$$

where $\nabla_{\boldsymbol{\theta}} = \sum_i^d \partial_{\theta_i}$ is the standard gradient operator. Equation (3.21) provides a differential equation for the evolution of the parameters based on the gradient of the function at a point $\boldsymbol{\theta}$. As shown in Section 2.1.2, this flow equation can be discretized as

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \epsilon \nabla_{\boldsymbol{\theta}^{(k)}} E(\boldsymbol{\theta}^{(k)}), \tag{3.22}$$

where ϵ is the step size that controls the precision of the discretization. Using this equation to update the parameters of $U(\boldsymbol{\theta})$ is called steepest descent, since we follow the gradient of the function to a minimum, since the gradient always points in the direct orthogonal to the level curves of the function. As a result, infinitesimal steps in the direction of the gradient will decrease the function's value until we reach a local minimum (under reasonable assumptions)[200].

As we have seen in Chapter 2, an important point in obtaining a successful VQE optimization is that the parameterization of the variational circuit $U(\boldsymbol{\theta})$ must be chosen appropriately. This implies that one must try different ansätze and assume that the state of interest can be expressed with the chosen ansatz. However, due to the non-convexity of the cost landscape, we have no guarantees that our optimizer can find a good approximation to the desired state.

3.2.2 Gradient flows on Lie groups

Since a quantum circuit is ultimately a unitary operation, we can think of an n-qubit quantum circuit U as an element of the special unitary group $\mathrm{SU}(N)$ (remember that

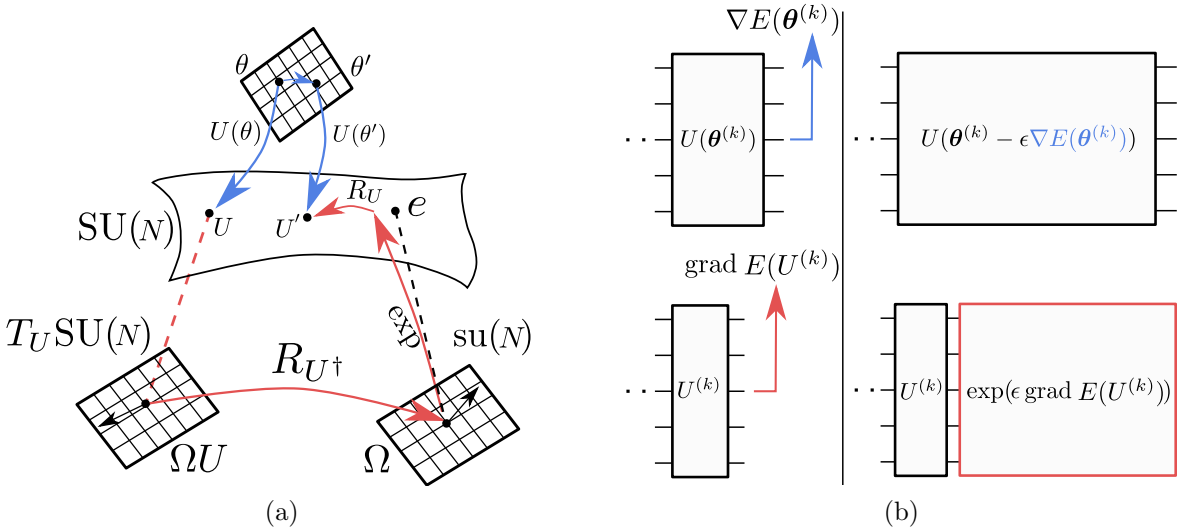


Figure 3.2: **Difference between the Riemannian gradient flow and Euclidean gradient flow.** (a) At the top, we have a mapping from real parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ to a unitary in $U \in \mathrm{SU}(N)$. Gradient updates in the parameter space from θ to θ' result in a new unitary U' on the group. Starting instead at U, we first obtain the Riemannian gradient at U in the tangent space $T_U\mathrm{SU}(N)$. Since the Riemannian gradient can be written as ΩU with $\Omega \in \mathfrak{su}(N)$, we can move to the Lie algebra $\mathfrak{su}(N)$ by multiplying the Riemannian gradient with $U^\dagger$ from the right. Then, the exponential map and subsequent right multiplication with U projects the Riemannian gradient back onto the manifold which results in a new unitary U'. (b) At the top, we see how a standard gradient flow optimizes a quantum circuit: The circuit stays fixed and the gradient is calculated via the parameter-shift rule. Next, the free parameters describing the unitary are updated via gradient descent. In the bottom figure, we see that a step of the Riemannian optimization corresponds to appending a new unitary to the original circuit.

$N = 2^n$). Hence, instead of minimizing the cost Equation (3.20) over $\mathbb{R}^d$ for a particular parameterization $U(\boldsymbol{\theta})$, we can instead directly optimize over $\mathrm{SU}(N)$. To achieve this, we rewrite Equation (3.20) as a scalar function on the special unitary group, $E : \mathrm{SU}(N) \to \mathbb{R}$, to obtain

$$E(U) = \mathrm{Tr}\left\{HU\rho_0 U^\dagger\right\}, \tag{3.23}$$

where $U \in \mathrm{SU}(N)$. To solve the optimization problem $\min_U E(U)$, we can consider the Riemannian gradient flow

$$\dot{U} = \mathrm{grad}\, E(U).$$

We now need to determine the Riemannian gradient $\mathrm{grad}\, E(U)$. As we have pointed out in Section 3.1.6, $\mathrm{SU}(N)$ is a Lie group, which carries a differentiable manifold structure. We can therefore use the language of differential geometry to define a gradient on the group [167, 168, 169, 170]. In Section 3.1.7, we showed that the tangent space $T_U\mathrm{SU}(N)$ of the manifold at a point U is a vector space that consists of a collection of vectors $\Omega \in T_U\mathrm{SU}(N)$ that provide the possible directions one can move in on the manifold from point U. The tangent vectors Ω can be defined as derivatives of curves going through the point U and are fully described by the Lie algebra $\mathfrak{su}(N)$. Once we introduce a metric, the manifold becomes a Riemannian manifold, with well-defined notions of angles and distance that allow use to define a gradient (see Section 3.1.4).

Here, we will explicitly construct $\mathrm{grad}\, E(U)$. First, we remember that the tangent space of $\mathrm{SU}(N)$ at the identity element $U_0 = I_N$ is given by the Lie algebra $\mathfrak{su}(N)$, the set of $N \times N$ skew-Hermitian matrices Ω with $\mathrm{Tr}\{\Omega\} = 0$. The elements of $T_U\mathrm{SU}(N)$ can then be found by right multiplying an element of the Lie algebra with U:

$$T_U\mathrm{SU}(N) \coloneqq \{\Omega U | \Omega \in \mathfrak{su}(N)\}.$$

With the notion of a tangent vector on $\mathrm{SU}(N)$, we can enforce the compatibility condition of Equation (3.10) to derive the resulting Riemannian gradient flow on $\mathrm{SU}(N)$.

We start with the left hand side of Equation (3.10),

$$dE(U)(\Omega U) = \mathrm{Tr}\left\{HU\rho_0 d(U^\dagger)(\Omega U) + Hd(U)(\Omega U)\rho_0 U^\dagger\right\},$$

where we simply used the product rule with the differential. Remember that Ω is a tangent vector, hence we can define a curve $\gamma(t)$, with $t \in (-a, a)$ such that $\gamma(0) = U$ and $\dot{\gamma}(t) = \Omega U$. We the see that

$$d(U)(\Omega U) = \left.\frac{d\gamma(t)}{dt}\right|_{t=0} = \Omega U, \quad d(U^\dagger)(\Omega U) = \left.\frac{d\gamma^\dagger(t)}{dt}\right|_{t=0} = U^\dagger\Omega^\dagger,$$

hence

$$\begin{aligned} dE(U)(\Omega U) &= \mathrm{Tr}\left\{HU\rho_0 U^\dagger \Omega^\dagger + H\Omega U\rho_0 U^\dagger\right\} \\ &= \mathrm{Tr}\left\{(-HU\rho_0 U^\dagger + U\rho_0 U^\dagger H)\Omega\right\} \\ &= \mathrm{Tr}\left\{\left[U\rho_0 U^\dagger, H\right]\Omega\right\}, \end{aligned}$$

where in the second line we used the cyclicity of the trace and the fact that $\Omega^\dagger = -\Omega$.

For the right hand side, we need to define a metric. For $\mathrm{SU}(N)$, there exists a (bi-invariant) metric $g : T_U\mathrm{SU}(N) \times T_U\mathrm{SU}(N) \to \mathbb{R}$ [185, 201]. This metric is given by

$$g(\Omega U, \zeta U) = g(\Omega, \zeta) = \mathrm{Tr}\left\{\Omega^\dagger \zeta\right\}, \quad \forall \Omega, \zeta \in T_U\mathrm{SU}(N).$$

Plugging this in, we then find

$$\begin{aligned} dE(U)(\Omega U) &= g(\mathrm{grad}\, E(U), \Omega U) \\ \mathrm{Tr}\left\{\left[U\rho_0 U^\dagger, H\right]\Omega\right\} &= \mathrm{Tr}\left\{(\mathrm{grad}\, E(U))^\dagger \Omega U\right\}, \end{aligned}$$

from which we conclude that

$$\begin{aligned} (\mathrm{grad}\, E(U))^\dagger &= U^\dagger\left[U\rho_0 U^\dagger, H\right] \\ \mathrm{grad}\, E(U) &= -\left[U\rho_0 U^\dagger, H\right]U. \end{aligned}$$

Flipping the sign to find the minimum of Equation (3.23) instead of the maximum gives the Riemannian gradient flow

$$\dot{U} = \mathrm{grad}\, E(U) = \left[U\rho_0 U^\dagger, H\right]U. \tag{3.24}$$

Analogous to the gradient in $\mathbb{R}^d$, the Riemannian gradient flow of Equation (3.24) converges to a critical point of $E(U)$ on $\mathrm{SU}(N)$ by descending along the level curves of the function [186]. To numerically compute the flow, we need to discretize the gradient steps.

The commutator $\left[U\rho_0 U^\dagger, H\right]$ is a skew-Hermitian matrix in the tangent space of $\mathrm{SU}(N)$ at U, hence left multiplication of U with the commutator will in general not keep us on the manifold. To perform a discrete gradient update step, we have to *retract* the Riemannian gradient from the tangent space onto $\mathrm{SU}(N)$ [202]. In contrast, for the Euclidean case of Equation (3.21) where $\mathcal{M} = \mathbb{R}^d$ this is not necessary since the tangent space of $\mathbb{R}^d$ coincides with the manifold at all points: $T_\theta\mathbb{R}^d \cong \mathbb{R}^d$.

The canonical retraction for our setting is the Lie exponential map $\exp_U : T_U\mathrm{SU}(N) \to \mathrm{SU}(N)$ given by $\Omega \mapsto \exp_U\{\Omega\}$, so that $\exp_U\{t\Omega\}$ for $t \in [0, 1]$ describes a unique geodesic

at U with initial "velocity" $\Omega \in T_U\mathrm{SU}(N)$. The operator $\exp_U$ can be decomposed as follows. We realize that $\operatorname{grad} E(U) = \Omega U$ with $\Omega = \big[U\rho_0 U^\dagger, H\big]$, hence right multiplication with the inverse $U^\dagger$ yields an element of the Lie algebra. Taking $\exp\{\Omega\}$ and multiplying with U from the right then produces the retracted gradient (see Figure 3.2a). If we discretize Equation (3.24) and perform the retraction, we finally obtain

$$U^{(k+1)} = \exp\Big\{\epsilon\big[U^{(k)}\rho_0 U^{(k)\dagger}, H\big]\Big\}U^{(k)}, \tag{3.25}$$

where ϵ is the step size and $U^{(k)} \in \mathrm{SU}(N)$ the unitary at step k.

To analyze the convergence properties of Equation (3.25), we rely on the fact that the map $\rho_0 \mapsto U^{(k)}\rho_0 U^{(k)\dagger}$ can be understood as a so-called double bracket flow on the adjoint orbits of the group [201, 203, 204].

Double bracket flows can be used to solve a variety of tasks such as sorting lists [205], describing Toda flows [206], or diagonalizing Hamiltonians in many-body physics [207, 208, 209]. Additionally, they have been studied in the context of quantum gate design [210]. The properties of this optimization scheme are well understood, in particular, if the Hamiltonian H is non-degenerate there exist exactly $N!$ minima on $\mathrm{SU}(N)$, and $(N-1)!$ global minima. Amazingly, only the global minima are stable attractors of the optimization dynamics, and one can show that almost all points will converge to these minima given a suitable step size [186]. Hence the Riemannian gradient flow is guaranteed to find the ground state of a non-degenerate Hamiltonian H.

3.2.3 Exact Riemannian gradient flow in quantum circuits

If $U^{(k)}$ in Equation (3.25) is implemented by a quantum circuit, then left multiplication of $U^{(k)}$ with the retracted Riemannian gradient is nothing more than appending a set of gates to that circuit, as illustrated in Figure 3.2b. However, it should come as no surprise that an implementation of the Riemannian gradient flow on a quantum computer will require an exponential number of gates as the number of qubits n increases, since an element in the Lie algebra $\mathfrak{su}(2^n)$ is described by $4^n - 1$ parameters in general. Nevertheless, we describe an approach for implementing the Riemannian gradient in a circuit in order to set up an approximate scheme that requires only a polynomial number of operations.

An exact approach to implement the Riemannian gradient on a quantum circuit is to decompose the skew-Hermitian operators $\big[U^{(k)}\rho_0 U^{(k)\dagger}, H\big]$ in terms of a basis of the Lie algebra $\mathfrak{su}(2^n)$. One such basis is the set of Pauli strings $\mathcal{P}_n := \{I, X, Y, Z\}^{\otimes n}$ (see Appendix A.1. We denote a Pauli string multiplied with the imaginary unit i by $P_j = ia$ with $a \in \mathcal{P}_n$, which ensures that all P_j are skew-Hermitian.

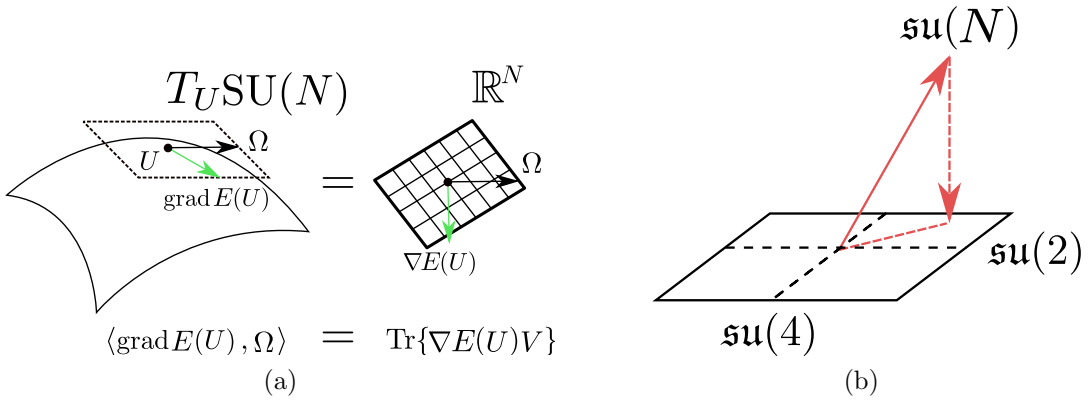


Figure 3.3: **Schematic depiction of the compatibility condition and subspace restriction.** (a) Restricting the algebra to a subspace and projecting the Riemannian gradient onto this subspace. Schematically, one can also break down the projected subspace into further component subspaces (represented for simplicity as single lines). (b) The compatibility condition. By taking the Euclidean inner product as the reference inner product, we can enforce the invariance of the inner product under a change of metric and explicitly construct grad $E(U)$.

We can write the commutator in the exponent of Equation (3.25) in terms of this basis,

$$\left[U^{(k)}\rho_0 U^{(k)\dagger}, H\right] = -\frac{1}{2^n}\sum_{j=1}^{4^n-1} \mathrm{Tr}\left\{\left[U^{(k)}\rho_0 U^{(k)\dagger}, H\right] P_j\right\} P_j. \tag{3.26}$$

The coefficients

$$c_j^{(k)} = \mathrm{Tr}\left\{\left[U^{(k)}\rho_0 U^{(k)\dagger}, H\right] P_j\right\} = \langle [H, P_j] \rangle_{\rho_k}, \tag{3.27}$$

with $\rho_k = U^{(k)}\rho_0 U^{(k)\dagger}$ can then be calculated on a quantum device with the parameter-shift rule [30, 31, 116, 119] (see also Section 2.1.2):

$$\langle [H, P_j] \rangle = i\left\langle V^\dagger(\frac{\pi}{2}) H V(\frac{\pi}{2}) - V^\dagger(-\frac{\pi}{2}) H V(-\frac{\pi}{2}) \right\rangle_{\rho_k}, \tag{3.28}$$

with $V(t) = \exp\{itP_j/2\}$ and the expectation value is calculated with respect to the state $U^{(k)}\rho_0 U^{(k)\dagger}$. Hence estimating the coefficients $c_j^{(k)}$ requires taking the gradient of

$$E(U,t) = \langle \psi_0 | U^{(k)\dagger} V^\dagger(t) H V(t) U^{(k)} | \psi_0 \rangle,$$

with respect to t. The resulting Riemannian gradient flow can be compactly written as

$$U^{(k+1)} \approx \prod_{j=1}^{4^n-1} \exp\Big\{-\epsilon c_j^{(k)} P_j\Big\} U^{(k)}, \tag{3.29}$$

where we absorbed the exponential factor into ϵ and took the sum out of the product via the Trotter formula at the cost of an error of $\mathcal{O}(\epsilon^2)$. In addition to requiring $4^n - 1$ estimates of $c_j^{(k)}$, this also requires applying the corresponding multi-qubit gates generated by all Pauli strings of size n, which will be very difficult in practice.

Note that instead of splitting the exponent of the sum with a Trotter decomposition, we could directly use a Cartan decomposition algorithm, e.g., the Khaneja-Glaser or D'Alessandro decomposition to recursively decompose the Riemannian gradient into products of single- and two-qubit unitaries [211, 212, 213].

3.2.4 Approximate Riemannian gradient flow in quantum circuits

To circumvent the exponential resources required for the exact Riemannian gradient, we consider an approximation scheme that requires only a polynomial number of parameters and gates. A natural approximation is restricting the Riemannian gradient to a subspace $\mathfrak{m} \subseteq \mathfrak{su}(2^n)$ via an orthogonal projection onto $\mathfrak{m}$. We show this schematically in Figure 3.3a. If we let $\{K_j\} \subset \mathcal{P}_n$ for $j = 1, \ldots, m$ be a basis of the subspace $\mathfrak{m}$, then from Equation (3.29) we obtain the local Riemannian gradient flow

$$U^{(k+1)} \approx \prod_{j=1}^{m} \exp\Big\{-\epsilon c_j^{(k)} K_j\Big\} U^{(k)}, \tag{3.30}$$

where now $c_j^{(k)} = \langle [H, K_j] \rangle_{\rho_k}$. This approximation gives us control over which directions in the Lie algebra we want to explore. Depending on the choice of $\mathfrak{m}$, we append a sequence of m gates at each optimization step. This approximation is an example of a stochastic Riemannian gradient algorithm [214].

Interestingly, the approximate Lie algebra optimization coincides with some VQE approaches for particular choices of $\mathfrak{m}$. For instance, if we restrict the Riemannian gradient to single qubit Paulis ,

$$\mathcal{P}_n^{1-\text{local}} := \{I^{\otimes i-1} \otimes \sigma_i \otimes I^{\otimes n-i} | \sigma_i \in \{X, Y, Z\}\},$$

where σ_i acts on qubit i, then we are performing a variant of the circuit structure learning algorithm called Rotosolve, [215, 216, 217] where instead of minimizing the expectation value $\langle H \rangle$ per added gate, we follow the Riemannian gradient with a step ϵ. Additionally, we can choose the subspace $\mathfrak{m}$ in such a way that the terms in the product Equation (3.30) become two-qubit gates. For example, we could take the subspace to consist only of 2-local Paulis,

$$\mathcal{P}_n^{2-\text{local}} := \{I \otimes \sigma_i \otimes I \ldots I \otimes \sigma_j \otimes I | \sigma_i, \sigma_j \in \{X, Y, Z\}\},$$

which contains $|\mathcal{P}^n_{2-\text{local}}| = 9n(n-1)$ terms. We can also consider the nearest neighbor 2-local Paulis,

$$\mathcal{P}_n^{2-\text{local n.n.}} := \{I \otimes \sigma_i \otimes \sigma_j \otimes \ldots \otimes I | \sigma_i, \sigma_j \in \{X, Y, Z\}\},$$

which contains $|\mathcal{P}_n^{2-\text{local n.n.}}| = 9(n-1)$ terms. If instead of appending all K_j in our set we only append the unitary generated by the K_j with the largest $c_j^{(k)}$, we are performing a popular meta-heuristic first introduced in [113] called ADAPT-VQE. The difference being that we do not re-optimize the parameters of previous layers at each step. Additionally, the Lyapunov control strategy FALQON [218] can be understood as a Trotterized time evolution where the stepsize of the drift Hamiltonian is set to the Riemannian gradient.

Analogous to these methods, the choice of operator pool that will be appended to the circuit will affect the quality of the ground state approximation. From the optimal control literature, we know that if $\{K_j\}$ spans $\mathfrak{su}(2^n)$ under nested commutation, the system is controllable and any state can be reached given sufficient depth [219, 220]. In the ADAPT-VQE setting, there are various proposals for which operators K_j to choose if one considers a fermionic Hamiltonian [221, 222, 223].

With the subspace restriction, the fixed point analysis becomes highly non-trivial. Although we still have the same convergence criterion as before, $\text{grad } E(U)|_{\mathfrak{m}} = 0$ can be satisfied if the Riemannian gradient has nonzero components orthogonal to the restricted subspace of the algebra, i.e., $\text{grad } E(U)|_{\mathfrak{m}} \in \mathfrak{p}$ where $\mathfrak{su}(N) = \mathfrak{p} \perp \mathfrak{m}$. As a result, we lose the global minima guarantees from the exact optimization. However, with the right choice of subspace, it is possible that the local Riemannian gradient information is enough to give a good approximation of the global minimum of Equation (3.23). Recently, a randomized algorithm was proposed that randomly twirls the local directions with a random unitary, which recovers the global convergence guarantees [224].

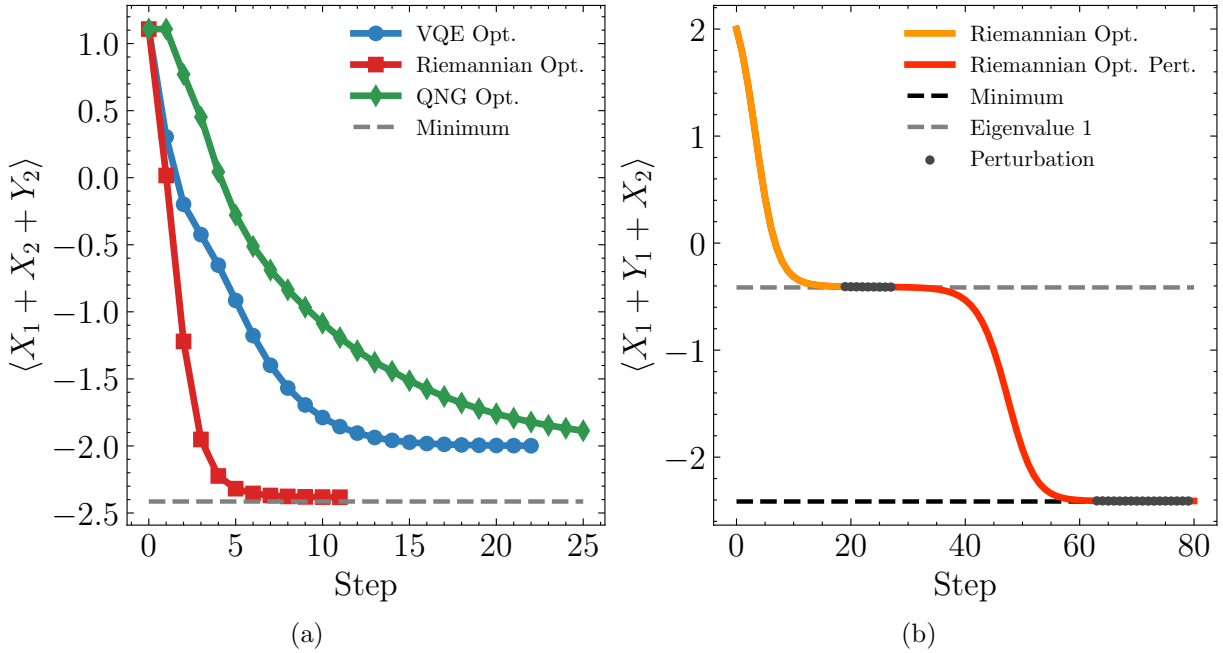


Figure 3.4: **Numerical examples on two qubits.** (a) Two-qubit example for the Hamiltonian $H = X_1 + X_2 + Y_2$. The circuit at initialization for both the VQE and Riemannian optimzer is given by the Hardware Efficient Ansatz [32]. This circuit ansatz consists of two RY gates with initial parameters $(0.1, 2.1)$, followed by two RZ gates with initial parameters $(0.3, 0.1)$ and finally a CNOT where the second qubit is the target. The step size for both the Riemannian gradient and parameter-shift VQE are $\epsilon = 0.5$. The VQE optimization gets stuck in a local minimum, whereas, the Riemannian gradient-flow optimizer rapidly reaches the optimal solution of $\langle H \rangle \approx -2.40$. We see that the use of the quantum natural gradient (QNG) does not improve the overal convergence [26].(b) Two-qubit example for the exact Riemannian gradient for the Hamiltonian $H = X_1 + Y_1 + X_2$. The circuit at initialization consists of two Hadamards on each qubit. The learning rate is set at $\epsilon = 0.2$. After 20 steps, the optimization gets stuck in an eigenstate. We generate a stochastic 4×4 matrix $X \sim \mathcal{N}(0, 0.1)^{4\times4}$ and obtain a random direction in the Lie algebra $K = \frac{i}{2}(X - X^T)$. After 5 perturbations, we escape the saddle point, and the optimization reaches the ground state of H.

3.2.5 Numerical examples

Here, we provide several numerical experiments on toy models to test the Riemannian gradient descent algorithm. Our Riemannian and VQE optimization procedures minimize the costs in Equation (3.23) and Equation (3.20), respectively. We have implemented the Riemannian optimizer in PennyLane as the `LieAlgebraOptimizer` [225].

First, we consider the exact Riemannian gradient flow, which can be implemented on a circuit for small system sizes. In Figure 3.4a, we compare the optimizer with the parameter-shift rule for a two-qubit circuit (see Section 2.1.2). We see that the Riemannian gradient flow can reach the ground state of a simple Hamiltonian, whereas the VQE optimization can only reach a sub-optimal solution.

To further illustrate the optimization properties of the exact Riemannian gradient flow, we study a two-qubit example in Figure 3.4b where the optimization gets stuck in an eigenstate, which corresponds to a saddle point in the optimization landscape. After performing a small perturbation in the Lie algebra, we escape the saddle point minimum and converge to the ground state.

In Figure 3.5, we see a simple example of the approximate Riemannian gradient flow, where we restricted the full Lie algebra to a subset of directions. We see that after a few steps, we get close to the minimum of the function. For this example, the Lie algebra restriction still allows us to reach the ground state of the Hamiltonian.

Although the local approximation provides an accurate solution for the previous toy example, we can run into issues for more non-trivial problems, as we see in the final example. We consider the problem of finding the ground state of the Transverse Field Ising model on four qubits, whose Hamiltonian is given in eq. (2.15). We assume periodic boundary conditions and set $g = 1$. The ground state of this model can be reached with a depth $n/2$ ansatz for an n-qubit chain using gradient-based VQE [40, 52, 1]. We find that the approximate Riemannian gradient optimizer can get close to the ground state. But unlike standard VQE, we cannot approximate the ground state closer than 10^{-2}, as can be seen in Figure 3.6.

Here, we see a limitation of the approximate Riemannian gradient flow. If we restrict the Lie algebra to $\mathfrak{su}(2)$ and $\mathfrak{su}(4)$ operators, the Riemannian gradient only has a local view of the cost landscape, and cannot access higher-order Lie algebra directions. On the contrary, VQE can access these directions, since the ansatz is often universal, i.e., made from a product of single and two-qubit unitaries. In principle, the unitary that is implemented by such an ansatz could have a generator $W(\theta)$ such that

$$U_{\mathrm{VQE}}(\theta) = \exp\{-iW(\theta)\},$$

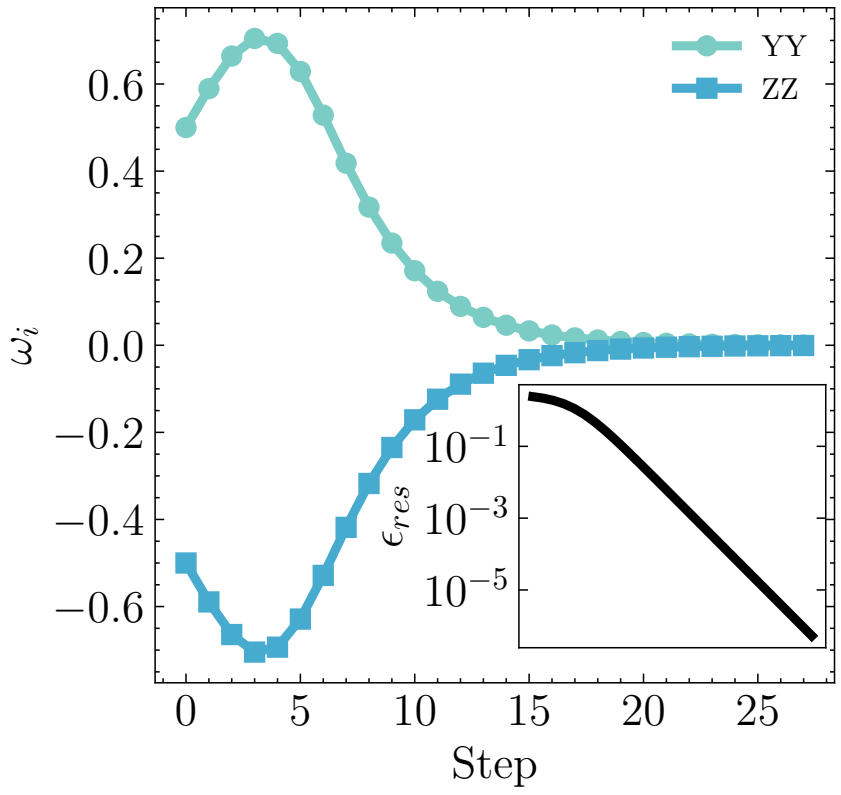


Figure 3.5: **Non-zero components of the Riemannian gradient versus the optimization step.** The initial circuit consists of two Hadamard gates. The Hamiltonian is $H = X_1 + Y_1 Z_2$. At each step in the optimization, the Riemannian gradient $\text{grad}\, E(U) = -\left[U^{(k)} \rho_0 U_k^\dagger, H\right]$ only has components in the YY and ZZ direction, keeping the state in the submanifold spanned by the states reachable by (XX, YY, ZZ). We can therefore restrict the Lie algebra to the subspace $\mathfrak{m}$ spanned by $\{YY, ZZ\}$ and perform the approximate Riemannian gradient flow. At each step, we need to calculate $\{\omega_k^{YY}, \omega_k^{ZZ}\}$. In the inset we see the residual energy $\epsilon_{res} = E_0 - \langle H \rangle$ versus the optimization steps. As the optimization progresses, we get exponentially closer to the ground state of H.

that can explore additional $\mathfrak{su}(N)$ directions in the Lie algebra for $N = 8, 16, \ldots$, albeit with a restricted parameterization.

A bottleneck for gradient-based VQE is that the number of circuit evaluations per optimization step scales linearly in the number of parameters, which is difficult in practice since parallel evaluation of quantum gradients requires multiple quantum devices. The

approximate Riemannian gradient flow does not suffer from this issue, since the amount of circuit evaluations is constant independent of circuit depth: we only require $|\mathfrak{m}|$ gradient calculations at each step. However, the Riemannian gradient flow may produce a circuit that is much deeper than the VQE ansatz, since we are appending gates to the circuit at each step k.

Ultimately, the approximate Riemannian gradient flow may not provide an accurate approximation to the ground state of a given Hamiltonian H. However, it could dynamically produce an ansatz that serves as a good starting point for further VQE optimization, similar to ADAPT-VQE.

3.2.6 Conclusion

We proposed Riemannian gradient flows in the context of variational quantum circuits. We showed that one can perform these types of optimizations on a quantum circuit, with strong convergence guarantees holding for exponentially deep variants of this algorithm. The usefulness of the local approximations to the Riemannian gradient flow merits further investigation in order to understand the power of this class of algorithms.

We hope that this alternative optimization paradigm can lead to new variational quantum algorithms, and provide insight for existing variational methods in noisy intermediate-scale quantum hardware. Additionally, we believe that the differential geometry and quantum control perspective can be a fruitful direction of research to further our understanding of the optimization properties such algorithms [227, 228]. In particular, these ideas could be used to investigate overparameterization in VQE [74, 1, 229, 230, 231]. Perhaps the global convergence guarantees of double bracket flows can be used to understand the convergence properties of deep quantum circuits and provide deeper insight into the power and limitations of VQE.

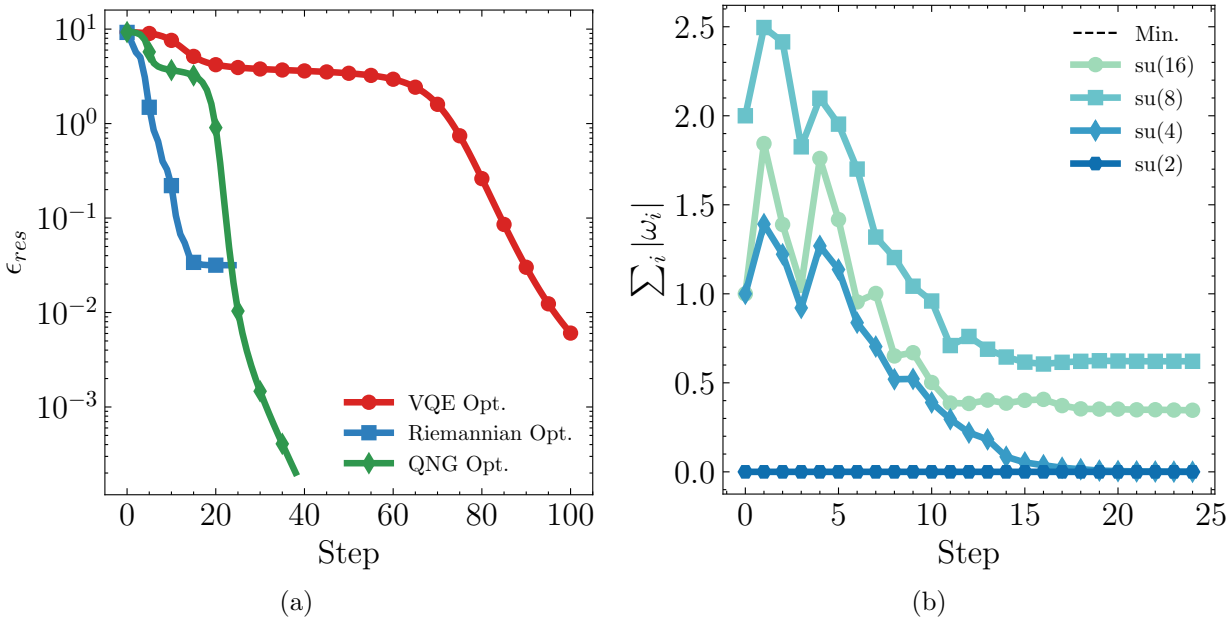


Figure 3.6: **Comparison of Riemannian gradient optimization versus gradient-based VQE for the 4-qubit transverse field Ising-model.** The Riemannian gradient circuit is initialized with a Hadamard on each qubit. To minimize gate costs, we use an adaptive scheme to reduce the amount of gates appended at each step of the Riemannian optimization. We obtain the $c_j^{(k)}$'s on all qubits or pairs of qubits for $\mathfrak{su}(2)$ and $\mathfrak{su}(4)$, respectively. Then, we select the largest $c_j^{(k)}$ and use we use a structure optimization algorithm to calculate the optimal step size ϵ [226, 216, 116, 119]. The gradient-based VQE optimizer has step size $\epsilon = 0.01$. Finally, we append a single gate corresponding to the chosen Lie algebra direction with this step size. (a) The residual energy $\epsilon_{res} = E_0 - E(U)$ plateaus for the Riemannian gradient close to the ground state energy. We verify that the optimizer is not stuck in an eigenstate close to the ground state, and so the optimization gets stuck due to the projection of the gradient onto the local algebra. The VQE optimization on the other hand is still getting closer to the ground state. In addition, the QNG optimizer finds the ground state much more rapidly than the vanilla gradient descent optimizer by taking the geometry of the Hilbert space into account. (b) Here, we plot the magnitude of all components of the Riemannian gradient versus the optimization steps. We see that Riemannian gradient becomes zero in the $\mathfrak{su}(4)$ direction, but higher order Lie algebra directions are still non-zero. This explains why we cannot converge close to the ground state: we need to access higher order elements of the Lie algebra. The transverse field Ising-model has symmetries that we can exploit. In particular, we can use the dynamical Lie algebra of the model to construct a Riemannian gradient flow within a subgroup, see Appendix E.1.

3.3 Here comes the SU(N): multivariate quantum gates and gradients

Typically, the gates in variational quantum circuits are parameterized unitary matrices generated by single Pauli-string operators that can locally rotate a state around some axis: $U(t) = \exp\{itG\}$, where t is a gate parameter and G a Pauli string. For a specific family of cost functions, there exist a variety of methods that allow one to obtain the gradient with respect to t [232, 30, 31, 233, 118, 116, 119] on quantum hardware. With these gradients, the cost function can be minimized via any gradient-descent-based algorithm.

Instead of considering a gate generated by a single Pauli string, one can construct more general parameterized gates that can perform an arbitrary rotation in SU(N), the special unitary group. These general SU(N) rotations are used in a variety of quantum algorithms [234, 235, 236, 32]. In practice, rotations in SU(N) can be implemented by composing several simple parameterized gates together into a more complicated one. For example, for single and two-qubit gates (where $N = 2, 4$, respectively), there exist several general decomposition schemes of such gates into products of single-qubit gates and CNOTs [211, 237, 238, 239, 240, 213]. In practice, this compilation comes with hardware-specific challenges, since quantum hardware usually has a set of native gates into which all others have to be decomposed [241, 242].

Choosing the right parameterization for a function is important because it can significantly affect the properties of its gradients. Reparameterizing functions to obtain more useful gradients is a well-known method in statistics and machine learning. For example, in restricted maximum likelihood methods one can ensure numerical stability of quasi-Newton methods by decomposing covariance matrices into Cholesky factors [243]. In addition, methods like auxiliary linear transformations [244], batch normalization [245] and weight normalization [246] are used to improve the gradients in neural networks. In variational inference, the reparameterization trick [247] is at the core of variational autoencoder approaches and allows for gradients for stochastic back-propagation [248, 249]. Therefore, it may be worthwhile to investigate alternative parameterizations of quantum gates for variational quantum algorithms.

Here, we propose a family of parameterized unitaries called $\mathbb{SU}(N)$ gates and provide a method to evaluate their gradients on quantum hardware. In doing so, we generalize the prior literature one step further, since many past schemes can be understood as special cases of our proposal [232, 30, 31, 233, 118, 116, 119]. We provide numerical results to support the validity of our approach and give several examples to illustrate the capabilities of the $\mathbb{SU}(N)$ gate. We show that this gate satisfies the quantum speed limit and that

it is easier to optimize compared to $\mathrm{SU}(N)$ parameterizations that consist of products of gates. We argue that this is the case because the product of unitaries creates a "bias" in the Lie algebra that deforms the cost landscape. In addition, we highlight the connections between our formalism and the properties of semisimple Lie algebras and establish a bound on the computational complexity of the gradient estimation using tools from representation theory.

3.3.1 $\mathbb{SU}(N)$ gates

As we saw in Section 3.2, we can understand a quantum circuit as an element of the unitary group. Similarly, we can consider the quantum gates that the circuit consists of as elements of the unitary group. Note that all the following works for any $N > 1$, but here we restrict ourselves to the qubit case. We are interested in constructing a quantum gate that parameterizes all of $\mathrm{SU}(N)$. To achieve this, we make use of the theory of Lie algebras. We will not be concerned with the formal treatment of this topic, which can be found in many excellent textbooks [250, 251, 176].

To construct our gate, we realize that $\mathrm{SU}(N)$ is a (semisimple) Lie group and so there exists a unique connection between its elements and the Lie algebra $\mathfrak{su}(N)$ via the so-called Lie correspondence, or Lie's third theorem [252, 176]. In particular, each $g \in \mathrm{SU}(N)$ can be identified with an $A \in \mathfrak{su}(N)$ via the exponential map $g = \exp\{A\}$. For our purposes, we can understand the Lie algebra $\mathfrak{su}(N)$ as a vector space of dimension $N^2 - 1$ that is closed under the commutator, $[A, B] = AB - BA \in \mathfrak{su}(N)$ for $A, B \in \mathfrak{su}(N)$. For $\mathfrak{su}(N)$, we choose as a basis the tensor products of Pauli matrices $\mathcal{P}_n$ multiplied by the imaginary unit i. This then gives the following parameterization of $\mathrm{SU}(N)$:

$$U(\boldsymbol{\theta}) = \exp\{A(\boldsymbol{\theta})\}, \quad A(\boldsymbol{\theta}) = \sum_m \theta_m G_m, \tag{3.31}$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_{N^2-1}) \in \mathbb{R}^{N^2-1}$ and $\{G_m\}$ where $G_m = ia$ with $a \in \mathcal{P}^{(n)}$. To distinguish between the group and the gate, we call the parameterization in Equation (3.31) an $\mathbb{SU}(N)$ gate. The coordinates $\boldsymbol{\theta}$ are called the canonical coordinates, which uniquely parameterize U through the Lie algebra $\mathfrak{su}(N)$. Since we typically cannot implement the above gate in hardware, we will have to be decompose it via a standard unitary decomposition algorithm [211, 237, 238, 239, 240, 213]. We emphasize here that even though the gate will be decomposed, it is parameterized as an exponential map. Hence, we can understand Equation (3.31) as a change of coordinates from the $\mathrm{SU}(N)$ gate decomposition.

If we do not want to parameterize all of $\mathrm{SU}(N)$, we can instead parameterize a more restricted Hamiltonian by setting some of the parameters θ_m to zero. This makes Equation (3.31) a natural parameterization of several Hamiltonians available on modern quantum hardware platforms. These Hamiltonians often have multiple independently tunable fields which can be active at the same time and do not necessarily commute. One typically has local control on each qubit and access to an interacting Hamiltonian between pairs of qubits, depending on the topology of the quantum device [253]. The interacting pair can for example be a ZZ interaction for Josephson flux qubits [254], a Heisenberg interaction for nuclear spins in doped silicon [255] or an XY interaction in quantum dots interacting with a cavity [256].

To use this gate in a gradient-based variational quantum algorithm, we have to be able to obtain partial derivatives of $U(\boldsymbol{\theta})$ with respect to each parameter θ_l. Although there exist a variety of works that provide analytical expressions for gradients through quantum circuits via the parameter-shift rule [232, 30, 31, 233, 118, 116, 119, 257], these works almost uniformly assume that the gate is of the form $U(\theta) = \exp\{i\theta P\}$ (see also Equation (2.12)), where P is a Hermitian operator. As far as we are aware, the only methods to obtain gradients of Equation (3.31) with respect to $\boldsymbol{\theta}$ are the stochastic and Nyquist parameter-shift rules of [233] and [234], respectively. The first approach relies on an integral identity for bounded operators that is estimated via Monte Carlo [258], whereas the latter is based on a theorem in Fourier analysis [259].

3.3.2 Obtaining the gradient

Here, we provide a new approach to obtain the gradient of Equation (3.31) that makes use of differentiable programming, which is efficient for gates acting on a small number of qubits. We are interested in calculating the following gradient:

$$\frac{\partial}{\partial\theta_l}U(\boldsymbol{\theta}) = \frac{\partial}{\partial\theta_l}\exp\{A(\boldsymbol{\theta})\}.$$

We know that this derivation must result in an element of the tangent space of $\mathrm{SU}(N)$,

$$T_U\,\mathrm{SU}(N) = \{U\Omega|\,\Omega \in \mathfrak{su}(N)\}.$$

Where we emphasize that we use the convention of left multiplication by U for the tangent vector as opposed to right multiplication in Section 3.2.

Consider a curve $X(t) : \mathbb{R} \to \mathrm{SU}(N)$, $t \in (-a, a)$ such that $X(0) = I$ and $d/dtX(t)|_{t=0} \equiv \dot{X}(0) = \Omega$. We define the following parameterized matrix function $Y : \mathbb{R} \times \mathbb{R} \to \mathrm{SU}(N)$,

$$Y(s,t) = e^{-sX(t)} \frac{\partial}{\partial t} e^{sX(t)},$$

where $X(t)$ is a curve on $\mathfrak{su}(N)$. We then find

$$\begin{aligned}
\frac{\partial Y(s,t)}{\partial s} &= e^{-sX(t)}(-X(t))\frac{\partial}{\partial t}e^{sX(t)} + e^{-sX(t)}\frac{\partial}{\partial t}(X(t)e^{sX(t)}) \\
&= -e^{-sX(t)}X(t)\frac{\partial}{\partial t}e^{sX(t)} + e^{-sX(t)}X(t)\frac{\partial}{\partial t}e^{sX(t)} + e^{-sX(t)}\frac{dX(t)}{dt}e^{sX(t)} \\
&= e^{-sX(t)}\frac{dX(t)}{dt}e^{sX(t)} \\
&= \mathrm{Ad}e^{-sX(t)}\frac{dX(t)}{dt}.
\end{aligned}$$

Then with equation Equation (3.19), we find

$$\frac{\partial Y(s,t)}{\partial s} = e^{-\mathrm{ad}sX(t)}\frac{dX(t)}{dt}.$$

Using that $Y(0,t) = 0$, we find by integration

$$Y(1,t) = \int_0^1 ds\, \frac{\partial Y(s,t)}{\partial s}.$$

Estimating the above integral forms the basis of the stochastic parameter-shift rule of [121] (see Appendix F.3). Continuing,

$$\begin{aligned}
Y(1,t) &= \int_0^1 ds \sum_{n=0}^{\infty} \frac{(-1)^n s^n}{n!}(\mathrm{ad}X)^n \frac{dX(t)}{dt} \\
&= \left[\sum_{n=0}^{\infty} \frac{(-1)^n s^{n+1}}{(n+1)!}(\mathrm{ad}X)^n\right]_{s=0}^{s=1} \frac{dX(t)}{dt} \\
&= \left(\sum_{n=0}^{\infty} \frac{(-1)^n}{(n+1)!}(\mathrm{ad}X)^n\right) \frac{dX(t)}{dt}.
\end{aligned}$$

Hence, we see that

$$\frac{d}{dt}e^{X(t)} = e^{X(t)}Y(1,t) = e^{X(t)}\left(\sum_{n=0}^{\infty} \frac{(-1)^n}{(n+1)!}(\mathrm{ad}X)^n\right)\frac{dX(t)}{dt}, \tag{3.32}$$

Note that at this point the Baker-Campbell-Hausdorff formula can be derived with Equation (3.32) by considering the derivative of

$$e^{Z(t)} = e^{tX}e^{tY},$$

and subsequent integration of the derivative of $Z(t)$ [260, 176].

From Equation (3.32) we see that the partial derivative with respect to a parameter θ_l is given by

$$\frac{\partial}{\partial\theta_l}U(\boldsymbol{\theta}) = U(\boldsymbol{\theta})\sum_{p=0}^{\infty}\frac{(-1)^p}{(p+1)!}(\mathrm{ad}_{A(\boldsymbol{\theta})})^p\frac{\partial}{\partial\theta_l}A(\boldsymbol{\theta}). \tag{3.33}$$

Here, ad_X denotes the adjoint action of the Lie algebra given by the commutator $\mathrm{ad}_X(Y) = [X, Y]$ [176]. Furthermore, we write $(\mathrm{ad}_X)^p(Y) = [X, [X, \ldots [X, Y]]]$, hence $(\mathrm{ad}_X)^p$ denotes a nested commutator of p terms. Note that the term on the right of $U(\boldsymbol{\theta})$ in Equation (3.33) is an element of the Lie algebra, since $\partial/\partial\theta_l A(\boldsymbol{\theta}) = G_l \in \mathfrak{su}(N)$ and so the commutator keeps the entire sum in the algebra. For notational clarity we define

$$\Omega_l(\boldsymbol{\theta}) = \sum_{p=0}^{\infty}\frac{(-1)^p}{(p+1)!}(\mathrm{ad}_{A(\boldsymbol{\theta})})^p\frac{\partial}{\partial\theta_l}A(\boldsymbol{\theta}), \tag{3.34}$$

where $\Omega_l(\boldsymbol{\theta}) \in \mathfrak{su}(N)$ is a skew-Hermitian operator that generates a unitary, which we call the *effective generator*. This gives the equation

$$\frac{\partial}{\partial\theta_l}U(\boldsymbol{\theta}) = U(\boldsymbol{\theta})\Omega_l(\boldsymbol{\theta}) \in T_{U(\boldsymbol{\theta})}\mathrm{SU}(N).$$

Given that Equation (3.34) is an infinite series of nested commutators it is not clear how $\Omega_l(\boldsymbol{\theta})$ can be calculated in practice without truncating the sum.

We make the following observation. One can think of $U(\boldsymbol{\theta})$ as a function $U : \mathbb{R}^{N^2-1} \to \mathrm{SU}(N)$ that we evaluate at the point $\boldsymbol{\theta}$. Since $\mathrm{SU}(N)$ is a differentiable manifold, we can define a set of local coordinates on the group and represent $U(\boldsymbol{x})$ as a vector described by $N^2 - 1$ real numbers. Hence, we can think of our gate as a coordinate transformation between the parameters $\boldsymbol{x}$ and the entries of the matrix representing the unitary. Since $U(\boldsymbol{x})$ depends smoothly on x_l via the matrix exponential, this coordinate transformation comes with a corresponding Jacobian (or more accurately, pushforward) $dU(\boldsymbol{x}) : T_{\boldsymbol{x}}\mathbb{R}^{N^2-1} \to T_{U(\boldsymbol{x})}\mathrm{SU}(N)$ that maps vectors tangential to $\mathbb{R}^{N^2-1}$ to vectors tangential to $\mathrm{SU}(N)$ (see also Equation (3.4). We can obtain this Jacobian by differentiating the

elements $U_{nm}(\boldsymbol{x})$ with respect to x_l:

$$\frac{\partial}{\partial x_l} U_{nm}(\boldsymbol{x}) = \partial_{x_l} \mathfrak{Re}[U_{nm}(\boldsymbol{x})] + i \partial_{x_l} \mathfrak{Im}[U_{nm}(\boldsymbol{x})]. \tag{3.35}$$

To obtain the above matrix function numerically, we rely on the fact that the matrix exponential and its derivative are implemented in differentiable programming frameworks such as JAX [261], PyTorch [262] and Tensorflow [263] through automatic differentiation. Here we make use of the JAX implementation, which provides the matrix exponential through a differentiable Padé approximation [264, 265].

Continuing, we note that evaluating $\partial U(\boldsymbol{x})/\partial x_l$ at a point $\boldsymbol{\theta}$ produces an element of the tangent space $T_{U(\boldsymbol{\theta})}\mathrm{SU}(N)$. We can move from the tangent space to the Lie algebra by left (or right depending on the chosen convention) multiplying the elementwise derivative of Equation (3.35) in Equation (3.33) with $U^\dagger(\boldsymbol{\theta})$ (see Section 3.1.7),

$$U^\dagger(\boldsymbol{\theta}) \left(\frac{\partial}{\partial x_l} U(\boldsymbol{x}) \bigg|_{\boldsymbol{\theta}} \right) = U^\dagger(\boldsymbol{\theta}) U(\boldsymbol{\theta}) \Omega_l(\boldsymbol{\theta}) = \Omega_l(\boldsymbol{\theta}), \tag{3.36}$$

which allows us to obtain $\Omega_l(\boldsymbol{\theta})$ exactly, up to machine precision. We emphasize that these steps can be performed on a classical computer, with a cost that is only dependent on the number of qubits the gate acts on, not the number of qubits in the circuit.

We now make the following observation: $\Omega_l(\boldsymbol{\theta})$ corresponds to a tangent vector on $\mathrm{SU}(N)$ and generates the one-parameter subgroup $V(t) = \exp\{t\Omega_l(\boldsymbol{\theta})\}$ such that

$$\Omega_l(\boldsymbol{\theta}) = \frac{d}{dt} \exp\{t\Omega_l(\boldsymbol{\theta})\}\big|_{t=0},$$

and

$$\frac{\partial}{\partial \theta_l} U(\boldsymbol{\theta}) = U(\boldsymbol{\theta}) \frac{d}{dt} \exp\{t\Omega_l(\boldsymbol{\theta})\}\big|_{t=0}. \tag{3.37}$$

We sketch this procedure schematically in Figure 3.7.

Next, consider a typical variational setting, where we are interested in minimizing the following cost function:

$$E(\boldsymbol{\theta}) = \mathrm{Tr}\left\{ U(\boldsymbol{\theta}) \rho U^\dagger(\boldsymbol{\theta}) H \right\}, \tag{3.38}$$

where H is some Hermitian operator and ρ the initial state of the system. For simplicity, we consider a circuit consisting of a single $\mathbb{SU}(N)$ gate. Differentiating the cost function with respect to θ_l gives

$$\frac{\partial}{\partial \theta_l} E(\boldsymbol{\theta}) = \mathrm{Tr}\left\{ \left(\frac{\partial}{\partial \theta_l} U(\boldsymbol{\theta}) \right) \rho U^\dagger(\boldsymbol{\theta}) H \right\} + \text{h.c.}.$$

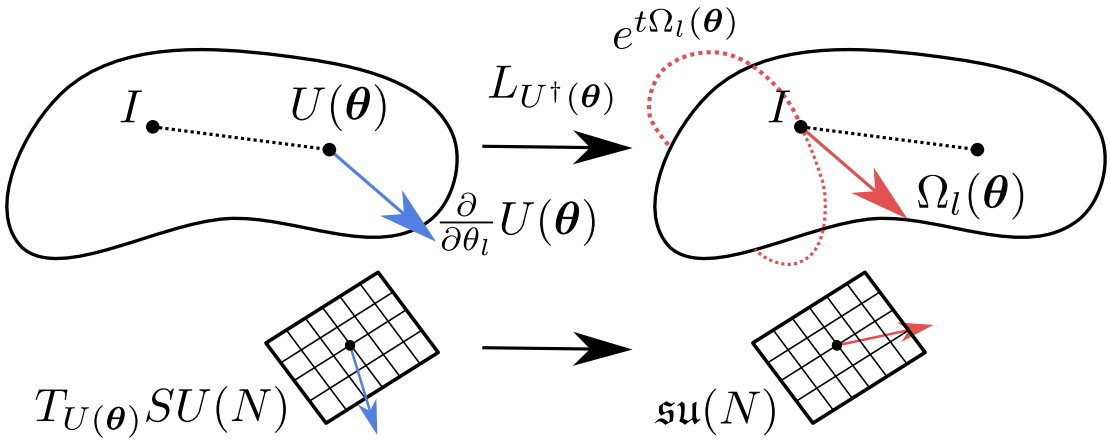


Figure 3.7: Schematic depiction of our gradient method. We move to the Lie algebra from the tangent space by left multiplication with $U^\dagger(\boldsymbol{\theta})$ and obtain $\Omega_l(\boldsymbol{\theta})$. The orbit generated by $\Omega_l(\boldsymbol{\theta})$ corresponds to the gate we have to insert in the circuit to compute the gradient.

Then, plugging in Equation (3.37) we find,

$$\frac{\partial}{\partial\theta_l}E(\boldsymbol{\theta}) = \frac{d}{dt}\mathrm{Tr}\left\{\left(U(\boldsymbol{\theta})e^{t\Omega_l(\boldsymbol{\theta})}\rho e^{-t\Omega_l(\boldsymbol{\theta})}U^\dagger(\boldsymbol{\theta})\right)H\right\}\Big|_{t=0} \equiv E(t), \tag{3.39}$$

where we used the skew-Hermitian property of the tangent vector $\Omega_l^\dagger(\boldsymbol{\theta}) = -\Omega_l(\boldsymbol{\theta})$. Note that Equation (3.39) corresponds to a new circuit with the gate $\exp\{t\Omega_l(\boldsymbol{\theta})\}$ inserted before $U(\boldsymbol{\theta})$ (see Figure 3.8).

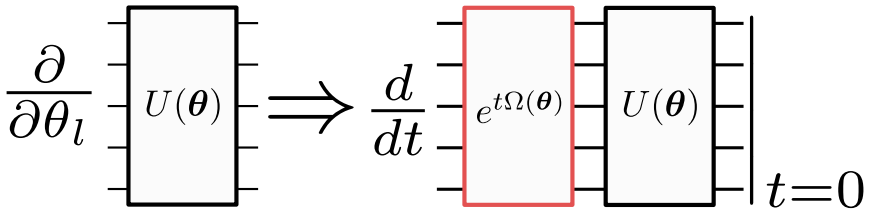


Figure 3.8: **Rewriting the gradient calculation.** The partial derivative with respect to the gate parameter θ_l can be obtained by adding a gate to the circuit that is generated by $\Omega_l(\boldsymbol{\theta})$. Calculating the derivative with respect to t and evaluating at $t = 0$ then provides one with the correct gradient.

The gradient of this new circuit can be computed on quantum hardware with a generalized parameter-shift rule (GPSR) [116, 118, 119]. In Algorithm 2, we outline the entire

algorithm for our gradient estimation, where we denote the GPSR subroutine with `gpsr`. An alternative to the generalized shift rule is to decompose the effective generators and apply the original two-term parameter-shift rule to the constituents (see Appendix F.5 for details). In [121], the authors proposed the so-called stochastic parameter-shift rule for multivariate gates, which is based on the Monte Carlo approximation of an operator identity.

Algorithm 2: SU(N) gradients.

Input: $U(\boldsymbol{x})$, ρ, H, $\boldsymbol{\theta}$
Obtain the Jacobian function:
for $l \in (1, \ldots, N^2 - 1)$ **do**
  $\partial_{x_l} U_l(\boldsymbol{x}) = \partial_{x_l} \mathfrak{Re}[U(\boldsymbol{x})] + i \partial_{x_l} \mathfrak{Im}[U(\boldsymbol{x})]$
For each gradient step:
for $l \in (1, \ldots, N^2 - 1)$ **do**
  $\Omega_l(\boldsymbol{\theta}) \leftarrow U^\dagger(\boldsymbol{\theta}) dU_l(\boldsymbol{x})|_{\boldsymbol{\theta}}$
  $E(t) \leftarrow \mathrm{Tr}\left\{U(\boldsymbol{\theta}) e^{t\Omega_l(\boldsymbol{\theta})} \rho e^{-t\Omega_l(\boldsymbol{\theta})} U^\dagger(\boldsymbol{\theta}) H\right\}$
  $\frac{\partial}{\partial \theta_l} E(\boldsymbol{\theta}) \leftarrow \texttt{gpsr}(\Omega_l(\boldsymbol{\theta}))$

In Figure 3.9 we consider a toy example using a random Hamiltonian on a single qubit and compare the exact derivative of an $\mathbb{SU}(2)$ gate with our generalized parameter-shift method (Algorithm 2), the stochastic parameter-shift rule and the central finite difference derivative with shifts $\pm\frac{\delta}{2}$. In particular, we consider the gate $U(\boldsymbol{\theta}) = \exp(iaX + ibY)$ with $\boldsymbol{\theta} = (a, b)$ and compute the partial derivative with respect to a over the range $a \in [0, \pi]$ for three fixed values of b on a state vector simulator (without shot noise). For the finite difference recipe we use $\delta = 0.75$, which we found to be a reasonable choice for a shot budget of 100 shots per cost function evaluation (see Appendix F.4). We observe that the generalized SU(N) derivative reproduces the exact value while the finite difference derivative is slightly biased. This is to be expected because the latter is an approximate method. While decreasing the shift size δ reduces the deterministic approximation error, it leads to larger overall estimation errors in shot-based computations like on quantum computers (see Appendix F.4 and e.g., [266]). Finally, the stochastic parameter-shift rule yields an unbiased estimator for the exact derivative but has a finite variance, which we estimated using 100 samples (see Appendix F.3). We stress that this variance is a property of the differentiation method itself and not due to sampling on the quantum computer. All methods require two unique circuits per derivative, but the stochastic shift rule needs additional circuits in order to suppress the variance. We provide the code for all our numerical experiments at [267].

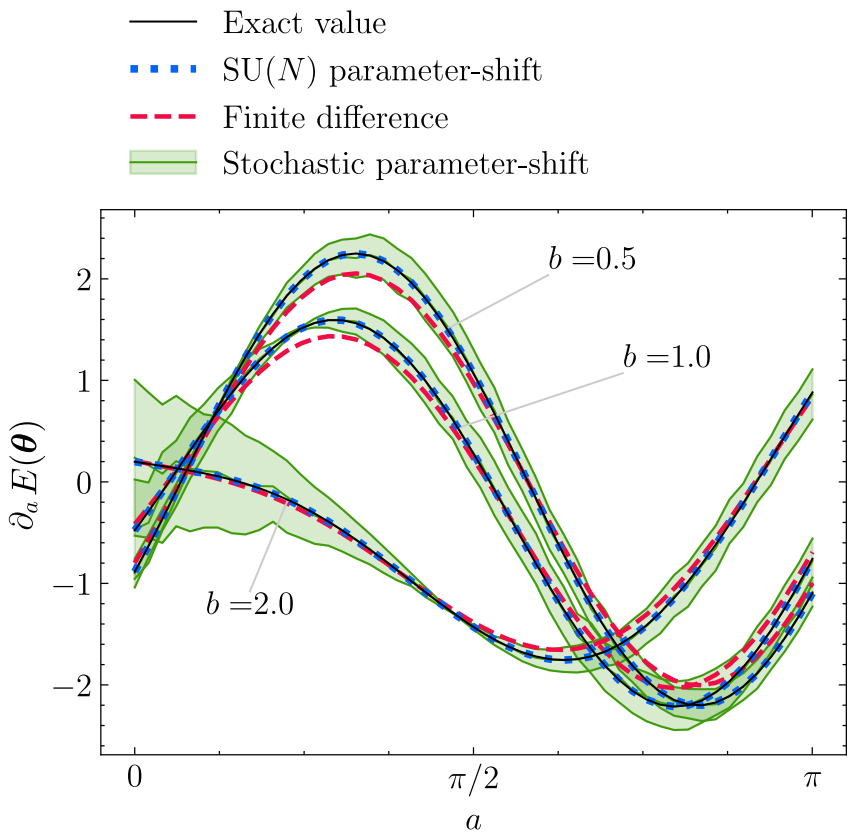


Figure 3.9: **Gradients of $E(\boldsymbol{\theta})$ for a single $\mathbb{SU}(2)$ gate and a random single-qubit Hamiltonian, in the limit of infinitely many shots on quantum hardware.** We take $A(\boldsymbol{\theta}) = iaX + ibY$ where $\boldsymbol{\theta} = (a, b)$ and consider the fixed values $b = 0.5, 1.0, 2.0$ together with $a \in [0, \pi]$. Our generalized shift rule (dotted) reproduces the exact value (solid), whereas the central finite difference (dashed) is biased and the stochastic shift rule (solid, shaded) comes with a finite statistical error even without shot noise from the quantum measurements. Since we look at a single-qubit operation, $\Omega_a(\boldsymbol{\theta})$ has a single spectral gap, so we require two shifted circuits to calculate the gradient entry (see Appendix F.1 for details). The finite difference and the stochastic shift rule require two circuits as well, but additional executions are need for the latter to reduce the shown single-sample error.

In addition, we compare the three methods in the presence of shot noise in Figure 3.10. We show the means and *single-shot* errors estimated with 1000 shots, which we split over 100 samples for the stochastic shift rule. We observe that the generalized $\mathrm{SU}(N)$ shift rule

systematically performs best. It is not only unbiased but also has the smallest variance. Note that for smaller parameters b, the SU(N) shift rule and the stochastic shift rule show very similar variances. This is because $U(\boldsymbol{\theta})$ approaches the gate $R_X(a) = \exp(iaX)$, which can be differentiated with the original parameter-shift rule, and both rules indeed reduce to the two-term shift rule for R_X.

3.3.3 Comparison with Riemannian gradient flow

We note that Equation (3.39) is closely related to the Riemannian gradient on SU(N) we discussed in Section 3.2.2 [185, 4]. However, instead of a gradient flow on a Lie group, we have defined a flow on the Lie algebra $\mathfrak{su}(N)$, which we retract back to the manifold via the exponential map. This subtle difference induces a different flow from the SU(N) one. Next, we follow the results of [196] in our notation. Consider the cost function

$$E(A) = \mathrm{Tr}\left\{e^A \rho e^{-A} H\right\},$$

where $A \in \mathfrak{su}(N)$. Note that although the minimum of the function is unchanged, the parameterization of a unitary via the Lie algebra changes the resulting gradient flow. To see this, we consider again the differential,

$$d(E \circ \exp A)(\Omega) = \mathrm{Tr}\left\{d(e^A)\rho e^{-A} H + e^A \rho d(e^{-A}) H\right\}(\Omega),$$

where now $d(E \circ \exp A) : \mathfrak{su}(N) \to \mathbb{R}$ and $\Omega \in \mathfrak{su}(N)$. We can now make use of the result in Equation (3.32) (in the right multiplication convention of the tangent space),

$$\begin{aligned} d(e^A)(\Omega) &= \left(\sum_{n=0}^{\infty} \frac{(-1)^n}{(n+1)!} (\mathrm{ad}A)^n\right) e^A(\Omega) \\ &= \Phi_X(\Omega) e^A, \end{aligned}$$

to obtain

$$d(E \circ \exp A)(\Omega) = \mathrm{Tr}\left\{\Phi_X(\Omega) e^A \rho e^{-A} H + e^A \rho e^{-A} \Phi_X^\dagger(\Omega) H\right\}.$$

Using that $\Phi_X^\dagger(\Omega) = -\Phi_X(\Omega)$ and $\Phi_X^\dagger(\Phi_X(\Omega)) = \Omega$, we then have

$$\begin{aligned} d(E \circ \exp A)(\Omega) &= \mathrm{Tr}\{\left[e^{-A}\rho e^A, H\right]\Phi_X(\Omega)\} \\ &= \mathrm{Tr}\{\Phi_X(\left[e^{-A}\rho e^A, H\right])\Omega\}, \end{aligned}$$

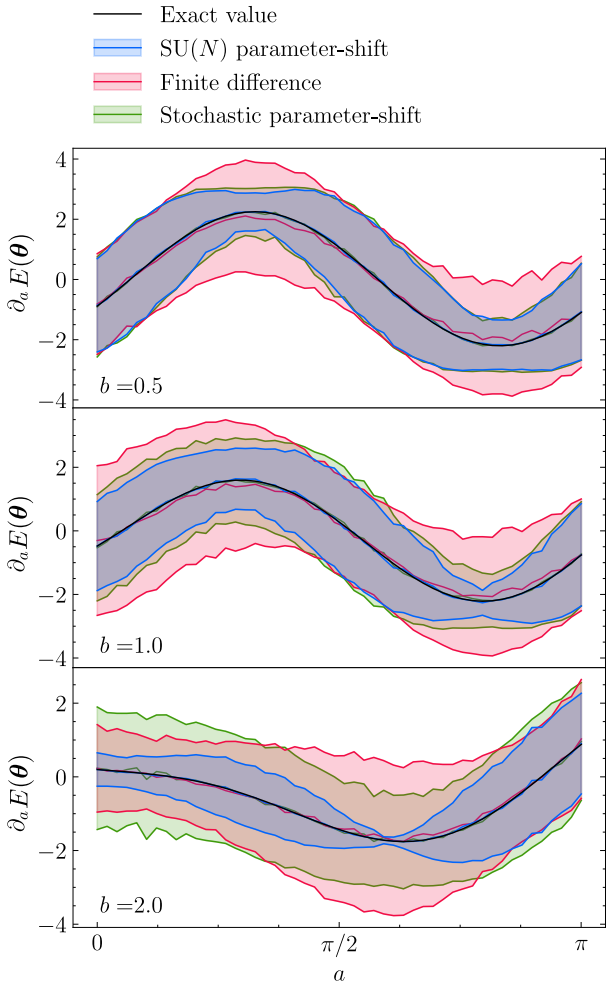


Figure 3.10: **Gradients of $E(\boldsymbol{\theta})$ as in Figure 3.9 but for finitely many shots on quantum hardware.** We show the single-shot error for each method, estimated with 1000 shots, which varies with the gate parameters as noted e.g., in [119]. Our generalized SU(N) shift rule systematically outperforms the other methods. For small b, the SU(N) and the stochastic shift rule approach the single-parameter shift rule and hence behave similarly. The finite difference shift $\delta = 0.75$ is chosen such that the bias and variance are traded off reasonably for 100 shots (see Appendix F.3 and e.g., [266]). For other shot numbers, δ needs to be optimized anew, whereas the parameter-shift rules are known to perform optimally at fixed shifts.

With the metric

$$g(\Omega U, \zeta U) = g(\Omega, \zeta) = \mathrm{Tr}\left\{\Omega^\dagger \zeta\right\}, \quad \forall \Omega, \zeta \in T_U \mathrm{SU}(N), \tag{3.40}$$

we find through the compatibility condition of Equation (3.10),

$$d(E \circ \exp A)(\Omega) = g(\mathrm{grad}\, E(A), \Omega)$$

from which we conclude

$$\begin{aligned}(\mathrm{grad}\, E(A))^\dagger &= \Phi_X(\left[e^{-A}\rho e^A, H\right]) \\ \mathrm{grad}\, E(A)) &= -\Phi_X(\left[e^{-A}\rho e^A, H\right]).\end{aligned}$$

Hence, with a similar argument as in Section 3.2.2, the gradient on $\mathfrak{su}(N)$ is

$$\mathrm{grad}\, E(A) = -\Phi_X(\left[e^{-A}\rho e^A, H\right]).$$

Compare this with the Riemmannian gradient on $\mathrm{SU}(N)$ of section 3.2.2

$$\mathrm{grad}\, E(U) = -\left[U\rho_0 U^\dagger, H\right]U.$$

We see that only when $\Phi_X = I$ do we obtain the Riemannian gradient on $\mathrm{SU}(N)$; hence only if $A \in \mathfrak{g}$ where $\mathfrak{g}$ is Abelian. An example would be the case where $A = Z$, i.e. the generator is a single Pauli. The optimization path followed by optimizing the parameters of an $\mathbb{SU}(N)$ gate is thus different from the one following a Riemannian gradient descent on $\mathrm{SU}(N)$ for non-Abelian Lie algebras.

3.3.4 Comparison with decomposed unitaries

Previous parameterizations of $\mathrm{SU}(N)$ unitaries consist of products of single-qubit gates and CNOTs [211, 237, 238, 239, 240, 213]. We refer to this parameterization as *decomposed* $\mathbb{SU}(N)$ gates. On the other hand, Equation (3.31) describes a general $\mathrm{SU}(N)$ unitary by exponentiating a parameterization of the Lie algebra $\mathrm{SU}(N)$. Here, we investigate the effects of this alternative parameterization.

Gate speed limit

First, we investigate a speed limit in terms of the gate time. We slightly modify the definition of Equation (3.31) for a unitary evolution of the system, $U(\boldsymbol{\theta}; t) \in \mathrm{SU}(N)$, to include a time $t \in \mathbb{R}^+$,

$$U(\boldsymbol{\theta}; t) = \exp\{\bar{A}(\boldsymbol{\theta})t\}, \tag{3.41}$$

where $\bar{A}(\boldsymbol{\theta}) = A(\boldsymbol{\theta})/\sqrt{\mathrm{Tr}\left\{A(\boldsymbol{\theta})^\dagger A(\boldsymbol{\theta})\right\}}$ is a normalized time-independent Hamiltonian (the imaginary unit i is included in $A(\boldsymbol{\theta})$). The normalization of $\bar{A}(\boldsymbol{\theta})$ is equivalent to the normalization of $\boldsymbol{\theta}$ in Euclidean norm, see Lemma F.6.1 in Appendix F.6. The normalization of the Hamiltonian (or, equivalently, $\boldsymbol{\theta}$) means that the total path length covered by the evolution is directly proportional to the evolution time t, since we are effectively setting the speed of the evolution to 1.

The Lie group $\mathrm{SU}(N)$ can be turned into a Riemannian manifold by equipping it with the Hilbert-Schmidt inner product of Equation (3.40). The unitary evolution $U(\boldsymbol{\theta}; t)$, parameterized by t, is a one-parameter subgroup that gives the geodesic [185, Theorem III.6] from the identity element at time $t = 0$. Geodesics can be defined as generalizations of straight lines in Euclidean geometry. Using Lemma F.6.2 (Appendix F.6), the length of the path (see Equation (3.8)) after time t is constant for time-independent normalized Hamiltonians with $|\boldsymbol{\theta}| = 1$,

$$L[U(\boldsymbol{\theta}; t), t] = \sqrt{N}t.$$

In general, there is more than one geodesic between two points on the manifold. For example, two points on the Bloch sphere can be connected by rotations about the same axis moving in opposite directions. Using Lemma F.6.3 (Appendix F.6), one of these geodesics must be the curve of the minimal path length. Hence, the minimum time to generate the evolution $U(\boldsymbol{\theta}; t_g)$ is t_g along the geodesic of the minimal path. For an initial state ρ and final state ρ_f, the Fubini-Study metric is used to find a minimum evolution time

$$t_g = \frac{1}{\sqrt{N}} \arccos\left(\sqrt{\mathrm{Tr}\left\{\rho\rho_f\right\}}\right),$$

giving the Mandelstam-Tamm bound for time-independent normalized Hamiltonians.

In practice, we may only have access to a restricted family of gates within $\mathrm{SU}(N)$, for example due to hardware limitations, in which case we require a decomposition of a desired gate in $\mathrm{SU}(N)$ into gates from this family. Here we want to compute the additional evolution time required by such a decomposition. The simplest gate decomposition is to break the unitary into two terms, $U(\boldsymbol{\theta}; t_g) = U(\boldsymbol{\phi}^{(2)}; t_2)U(\boldsymbol{\phi}^{(1)}; t_1)$. The parameters $\boldsymbol{\phi}^{(1)}$ and $\boldsymbol{\phi}^{(2)}$ are also normalized Hamiltonians, i.e., they have the norm $|\boldsymbol{\phi}^{(1)}| = |\boldsymbol{\phi}^{(2)}| = 1$. The following theorem shows that using a decomposed circuit over an $\mathbb{SU}(N)$ gate gives an additional evolution time, which corresponds to longer circuit run times.

Theorem 3.3.1. *For unitary gates generated by normalized time-independent Hamiltonians, consider a general circuit decomposition of two gates $U(\boldsymbol{\phi}^{(2)}; t_2)U(\boldsymbol{\phi}^{(1)}; t_1)$. There*

exists an equivalent evolution with an $\mathbb{SU}(N)$ *gate* $U(\boldsymbol{\theta}; t_g) = U(\boldsymbol{\phi}^{(2)}; t_2)U(\boldsymbol{\phi}^{(1)}; t_1)$*, with evolution time* t_g*, such that*

$$t_g \leq t_1 + t_2,$$

with equality if $\boldsymbol{\phi}^{(1)} + \boldsymbol{\phi}^{(2)} = \boldsymbol{\theta}$*.*

The proof of the theorem is in Appendix F.6. As expected, a decomposition into two gates gives a longer total evolution time than is possible with an $\mathbb{SU}(N)$ gate due to the normalizations of $\boldsymbol{\phi}^{(1)}$, $\boldsymbol{\phi}^{(2)}$, and $\boldsymbol{\theta}$. A decomposition into more gates would generally lead to an even greater evolution time. A corollary of Theorem 3.3.1 is that any circuit with multiple non-commuting layers of gates cannot be optimal in total time.

Unbiased cost landscapes

An additional advantage of the $\mathbb{SU}(N)$ gate is that it weighs all optimization directions equally. In contrast, a parameterization of $\mathrm{SU}(N)$ in terms of a product of gates will create a bias in the parameter space. We illustrate this point with the following example. Consider the decomposed $\mathbb{SU}(2)$ gate $V(\boldsymbol{\theta}) = R_Z(\theta_3)R_Y(\theta_2)R_Z(\theta_1)$ where $R_A(\theta) = \exp\{i\theta A\}$ and $A = X, Y, Z$. This is the ZYZ decomposition. Using similar techniques as in Appendix F.6, we can rewrite $V(\boldsymbol{\theta})$ to be parameterized in terms of the Lie algebra:

$$V(\boldsymbol{\theta}) = \exp\{i\boldsymbol{\phi} \cdot \boldsymbol{\sigma}\},$$

where $\boldsymbol{\sigma} = (X, Y, Z)$ and

$$\boldsymbol{\phi} = \frac{\arccos(\cos(\theta_2)\cos(\theta_1 + \theta_3))}{\sqrt{1 - \cos^2(\theta_2)\cos^2(\theta_1 + \theta_3)}} \times \begin{pmatrix} \sin(\theta_2)\sin(\theta_1 - \theta_3) \\ \sin(\theta_2)\cos(\theta_1 - \theta_3) \\ \cos(\theta_2)\sin(\theta_1 + \theta_3) \end{pmatrix}.$$

If we look at the components of $\boldsymbol{\phi}$, we see that the different directions in the Lie algebra are stretched or compressed as a result of the particular choice of parameterization. Consider the normalization $|\theta_1| + |\theta_2| + |\theta_3| = 1$ for the ZYZ decomposition and $|\boldsymbol{\theta}| = 1$ for the $\mathbb{SU}(N)$ gate. With each Hamiltonian term normalized to 1, the prefactor gives the evolution time.

These choices of norm give equal total evolution times for the ZYZ decomposition and $\mathbb{SU}(2)$ gate, $T_{\mathrm{ZYZ}} = T_{\mathrm{SU}(N)} = \sqrt{2}$, irrespective of the specific parameters chosen. In

Figure 3.11, we graphically illustrate the Lie algebra deformation by showing the $\boldsymbol{\phi}$ surface for both the ZYZ decomposition and $\mathbb{SU}(2)$ gate. Note that we have not considered any cost function here; the bias occurs at the level of the parameterization of a specific unitary.

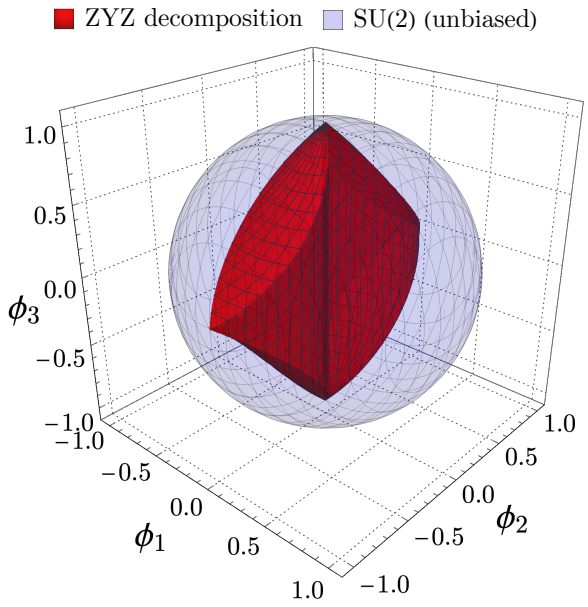


Figure 3.11: **Comparing the decomposed unitary with the** $\mathrm{SU}(N)$ **parameterization.** The total unitary evolution for the ZYZ decomposition (red) and the $\mathbb{SU}(2)$ gate (blue) can be expressed in the form $\exp\{i\boldsymbol{\phi}\cdot\boldsymbol{\sigma}\}$. The components $\boldsymbol{\phi}=(\phi_1,\phi_2,\phi_3)$ give the magnitude of the respective basis generators $\boldsymbol{\sigma}=(X,Y,Z)$. The original parameterization in $\boldsymbol{\theta}$ with norm $|\theta_1|+|\theta_2|+|\theta_3|=1$ gives a surface of possible values of $\boldsymbol{\phi}$ and therefore possible unitary evolutions. The $\mathbb{SU}(2)$ gate (blue) is unbiased because its parameterization gives the correspondence $\boldsymbol{\theta}=\boldsymbol{\phi}$ with normalization $\boldsymbol{\phi}_1^2+\boldsymbol{\phi}_2^2+\boldsymbol{\phi}_3^2=1$. The unitary evolution for the ZYZ decomposition (red) is biased because the surface in the $\boldsymbol{\phi}$ coordinates does not maintain an equal magnitude in all directions.

The effect of this bias is demonstrated in Figure 3.12 for the simplest case of a single-qubit system with an $\mathbb{SU}(2)$ gate. The optimal parameters of the circuit are those that produce the state that gives the minimum of the cost function $E(\boldsymbol{\theta})=-\langle Y\rangle$ (green star). We consider various initial parameters acting on the reference state $\rho=|0\rangle\langle 0|$. The corresponding training paths are shown for each initial parameter vector. The training paths for the decomposed ZYZ circuit are depicted in Figure 3.12(a). As the initial parameter $\boldsymbol{\theta}_0$

acting on the reference state ρ (purple dots) moves closer to an unstable equilibrium point (orange diamond) the training path becomes increasingly suboptimal. At the unstable equilibrium the only gradient information is directly away from the instability rather than providing information about the direction towards the global minimum. This behavior is further illustrated by the gradient vector field on the Bloch sphere in Figure 3.12(c). For the $\mathbb{SU}(N)$ gate, we see in Figure 3.12(b) that the optimization trajectories follow a direct path to the minimum.

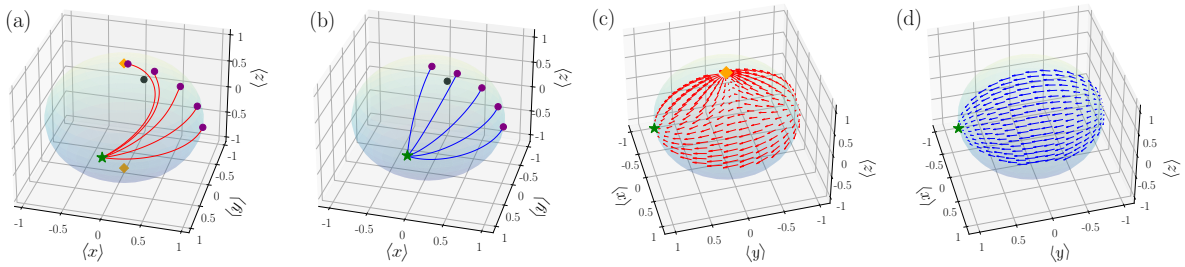


Figure 3.12: **Comparison of the update of circuit parameters from various initial parameters acting on the initial state $\rho = |0\rangle\langle 0|$.** The training paths are depicted on the Bloch sphere for: (a) parameterized single-qubit rotations for the ZYZ ansatz; and (b) using the $\mathbb{SU}(N)$ gate. The purple dots represent initial states generated by applying $U(\boldsymbol{\theta}_0)$ with $\boldsymbol{\theta}_0 = (0, a, 0)$ where $a \in \left\{\frac{\pi}{64}, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \frac{\pi}{2}\right\}$ to ρ. Note that for this choice of initial parameters, $U(\boldsymbol{\theta}_0) = V(\boldsymbol{\theta}_0)$. The objective function is $E(\boldsymbol{\theta}) = -\langle Y \rangle$, giving the target final state at the green star—the state that gives the global minimum of $E(\boldsymbol{\theta})$. The unstable equilibrium points are given by orange diamonds, at $(0, 0, 1)$ and $(0, 0, -1)$, and the black point is at the maximum of the cost function, $(0, 1, 0)$. (c) shows the gradient vector field of the decomposed ZYZ ansatz. The vector field for the $\mathbb{SU}(2)$ gate, shown in (d), coincides with the geodesic flow towards the target final state at all points which satisfies the gate speed limit.

Numerical experiments

To investigate the effect on the performance of a typical optimization, we study how an $\mathbb{SU}(N)$ gate compares with a decomposed gate in a larger circuit. In Figure 3.13a we provide a non-trivial example, where we incorporate our gates into a circuit and show that it performs better than a decomposed $\mathbb{SU}(4)$ gate on a set of random problem instances. We show the individual optimization trajectories in Figure 3.13b which illustrate the faster optimization of $\mathbb{SU}(N)$ gates compared to decomposed gates. Like for the examples in

Figure 3.9 and Figure 3.10, we assume that there is no gate or measurement noise. Additionally, we assume that we can always implement the gate generated by $\Omega_l(\boldsymbol{\theta})$, and have control over all Pauli operators G_m. In practice, we typically only have access to a fixed set of generators $\text{span}(\{G_m\}) < \text{span}(\mathfrak{su}(N))$. If this is the case, then we require a decomposition of $\exp\{t\Omega_l(\boldsymbol{\theta})\}$ in terms of the available operations on the device [211, 213]. All numerical results were obtained with `PennyLane` [225], and the $\mathbb{SU}(N)$ gates can be accessed via the `qml.SpecialUnitary` class. Although we do not explore this here, one could make use of sparse optimization methods such as stochastic optimization [268, 269] and frugal optimization [270] for the GPSR subroutine in our algorithm.

3.3.5 Resource estimation

To obtain the partial derivative in Equation (3.39) in practice we need to estimate the gradient of a circuit that contains a gate generated by $\Omega_l(\boldsymbol{\theta})$. As noted in recent works on GPSR rules [116, 118, 119], the computational cost of estimating this gradient is related to the spectral gaps of $\Omega_l(\boldsymbol{\theta})$. In particular, if $\{\lambda_j\}$ is the set of (possibly degenerate) eigenvalues of $\Omega_l(\boldsymbol{\theta})$, we define the set of unique spectral gaps as $\Gamma = \{|\lambda_j - \lambda_{j'}|\}$ where $j' > j$. Note that for d distinct eigenvalues, the number of unique spectral gaps R is at most $R \leq d(d-1)/2$. The total number of parameter-shifted circuits is then $2R$ for a single partial derivative $\partial_{\theta_l} E(\boldsymbol{\theta})$

Depending on the generator $\Omega_l(\boldsymbol{\theta})$, this complexity can be improved. For instance, in [118], a Cartan decomposition is used to improve the number of circuits required from polynomial to linear or even logarithmic in N. Additionally, in [116], the different costs for first- and second-order gradients are determined for specific varational quantum algorithms like QAOA [49] and RotoSolve [271, 272, 217, 216]. Finally, in [119], the computational cost of a variety of different gates is investigated in detail and the variance across the parameter regime is studied.

Instead of focusing on specific instances of the generator $\Omega_l(\boldsymbol{\theta})$, we make a more general observation about the computational complexity of parameter-shift gradient rules. In general, $\Omega_l(\boldsymbol{\theta})$ has full support on $\mathfrak{su}(N)$, since the consecutive applications of $\text{ad}_{A(\boldsymbol{\theta})}$ in Equation (3.33) typically generate all of $\mathfrak{su}(N)$ [7]. However, for specific choices of $A(\boldsymbol{\theta})$, the application of $\text{ad}_{A(\boldsymbol{\theta})^p}$ to $\partial_{\theta_l} A(\boldsymbol{\theta})$ closes to form a subalgebra, called the dynamical Lie algebra of $A(\boldsymbol{\theta})$, that is contained in $\mathfrak{su}(N)$. These algebras are well-known in the context of quantum optimal control [219, 27], and have recently been studied in the context of variational quantum algorithms [227, 230].

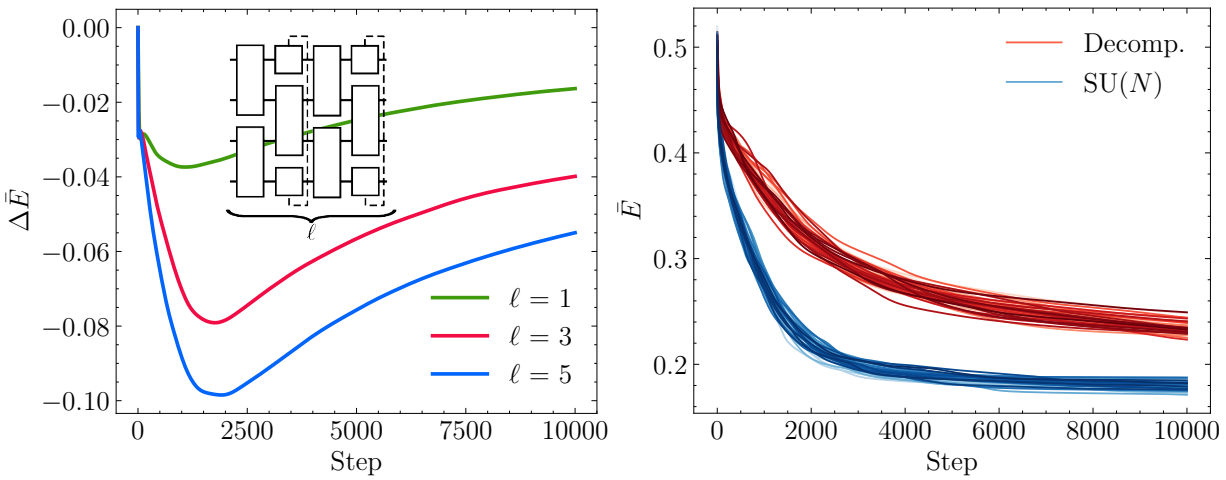


Figure 3.13: **Comparison of decomposed gates versus $\mathbb{SU}(N)$ gates in bricklayer circuits for random 10-qubit Hamiltonians and various depths.** (a) We consider the bricklayer circuit indicated with $\ell = 2$ in the inset, with general two-qubit gates acting on the even and odd qubits in each layer. The decomposed gate is the SU(4) parameterization of [238], which is optimal in the number of CNOTs required. For each instance, we sample a Hamiltonian from the Gaussian unitary ensemble and minimize the cost in Equation (3.38) via standard gradient descent. We show the difference of the relative errors in energy $\bar{E} = (E - E_{\text{min}})/(E_{\text{max}} - E_{\text{min}})$ between the decomposed gates and the $\mathbb{SU}(N)$ gates, that is $\Delta\bar{E} = \bar{E}_{\text{SU}(N)} - \bar{E}_{\text{Decomp.}}$. The plotted lines are the mean $\bar{E}$, averaged over 50 random Hamiltonians for each circuit depth ℓ. We see that for all depths $\Delta\bar{E} < 0$ at all points during the optimization, hence the bricklayer circuit with the $\mathbb{SU}(N)$ gates outperforms the circuit where the two-qubit interactions are parametrized as a gate composition. (b) We show the trajectories from the optimizations in Figure 3.13a for 50 random 10-qubit Hamiltonians sampled from the Gaussian unitary ensemble and an $\ell = 5$ bricklayer circuit of 2-qubit building blocks. We compare the relative error energy (see Figure 3.13a for the definition of $\bar{E}$) when using a standard gate composition to that when using $\mathbb{SU}(4)$ gates as building blocks. The optimization is performed with vanilla gradient descent using a learning rate of $\eta = 10^{-3}$. The $\mathbb{SU}(4)$ gate consistently leads to faster optimization and better approximations of the ground state energy throughout all 10^5 optimization steps.

Here, we define the dynamical Lie algebra (DLA) $\langle A(\boldsymbol{\theta})\rangle_{\text{Lie}}$ as the subalgebra formed under the closure of the non-zero terms in $A(\boldsymbol{\theta})$ under the commutator:

$$\langle A(\boldsymbol{\theta})\rangle_{\text{Lie}} := \text{span}\big\{\text{ad}_{G_{i_1}} \cdots \text{ad}_{G_{i_r}}(G_j) \,\big|\, G_{i_1}, \ldots, G_{i_r}, G_i \in \{G_m\} \text{ if } \theta_m \neq 0\big\}.$$

Ignoring global phases, this will always result in a subalgebra $\mathfrak{a} \subseteq \mathfrak{su}(N)$. For example, given $A(\boldsymbol{\theta}) = i(aX + bY)$, $\forall a, b \in \mathbb{R}$, we have $\langle\{iX, iY\}\rangle_{\text{Lie}} = \text{span}\{iX, iY, iZ\}$, since $\text{ad}_X(Y) = [X, Y] = iZ$ and successive commutators generate no new contributions. Note that for this example the DLA equals the full Lie algebra $\mathfrak{su}(2)$. Explicit constructions of DLAs that span $\mathfrak{so}(N)$ and $\mathfrak{sp}(N)$ are given in [220]. In a more recent work, the DLAs of several typical quantum many-body Hamiltonians are studied and their properties are used to prepare efficient time-evolution circuits [228]. In one dimension, the DLAs that are generated by Pauli strings have recently been classified [6], and we will discuss this classification in detail in Chapter 4.

Interestingly, if the DLA is maximal, i.e., there exists no smaller non-trivial subalgebra within $\mathfrak{a}$, then the *roots* of the Lie algebra can be related directly to the computational cost of estimating the gradients in Equation (3.39). We formally establish this connection with the following theorem:

Theorem 3.3.2. *The number of unique spectral gaps R of $\Omega_l(\boldsymbol{\theta})$ is upper bounded by the number of roots $|\Phi|$ of any maximal semisimple DLA,*

$$R \leq |\Phi|/2.$$

We provide the proof of Theorem 3.3.2 in Appendix F.8. We make use of the fact that any semisimple Lie algebra can be written as a direct sum of its weight spaces, which can be identified with its root system [273]. The number of roots $|\Phi|$ can then be used to bound the total number of unique spectral gaps of $\Omega_l(\boldsymbol{\theta})$. We can thus use Theorem 3.3.2 to assess the run time of Algorithm 2. We give several examples of $\mathbb{SU}(N)$ gates in Appendix F.8 together with the corresponding values of R. Depending on the physical system or hardware that we are working with, we have to choose a representation for $\mathfrak{su}(N)$, which is a map $\mathfrak{su}(N) \to \mathfrak{gl}(N, \mathbb{C})$. We chose this representation to be the tensor product of the fundamental representation, i.e., Pauli monomials $\mathcal{P}_n$ times i. Note however, that Equation (3.39) and Theorem 3.3.2 hold for any irreducible representation of $\mathfrak{su}(N)$.

Additionally, by connecting the spectral gaps to the root system of the DLA, we can make use of a beautiful result in representation theory: the classification of all maximal subalgebras of the classical Lie algebras [274]. Each root system can be uniquely identified

with a particular subalgebra of a Lie algebra and it can be shown that there exist a finite number of root systems. Since a DLA is a subalgebra of $\mathfrak{su}(N)$, we can identify all possible DLAs and by extension all possible families of $\mathbb{SU}(N)$ gates. We provide examples of this procedure in Appendix F.8.

3.3.6 Conclusion

We have proposed an alternative parameterization of general $\mathbb{SU}(N)$ gates and a method of optimizing these gates in prototypical variational quantum algorithms. We have shown that in toy example settings, our gates are more powerful, and motivated why we believe this is the case based on quantum speed-limit arguments. A natural extension of our work would be to test our method in experimental settings, both on gate-based quantum computers or quantum simulators [12, 275, 276]. With regard to the latter, several methods have been investigated that could provide pulse-level optimization of energy cost functions [277, 278]. This would obviate the need for a gate-based model of quantum computing to prepare specific states on quantum hardware. Instead, we work on the Hamiltonian level and control the system directly. Our algorithm could be applied to this setting as well, since we're effectively learning the parameters of some fixed Hamiltonian. An interesting future direction would be to see how truncating the support of the gradient affects the accuracy of the final gradient. We briefly explore this in Appendix F.9.

Additionally, our approach could be used for the purpose of unitary design. In [279], we study the problem of finding a time-independent Hamiltonian that implements some unitary operation U. To avoid getting stuck in local minima, we use the gradient method presented in Section 3.3.2 to follow the geodesic to the optimal gate Hamiltonian given the experimental constraints.

We have shown that the $\mathbb{SU}(N)$ gate in a circuit outperforms a decomposed gate. The number of parameters in our proposed gate equals 4^n, hence $\mathbb{SU}(N)$ gates acting on a large number of qubits will be impractical. Additionally, it is not clear for which problems one would rather have a deeper circuit with simple gates as opposed to a shallow circuit with more powerful gates. This also begs another question: will our gates suffer from barren plateaus [35]? It is likely that a circuit of 2-qubit $\mathbb{SU}(N)$ gates that has linear depth in N will lead to a circuit that forms an approximate 2-design, which will suffer from vanishing gradients. However, appropriate choices of the generators $A(\boldsymbol{\theta})$ of our gate could keep the circuit in a polynomially scaling DLA of the entire circuit, which can avoid barren plateaus [227, 230]. Additionally, we can consider parameter initialization strategies that can improve the optimization [68, 123].

With regard to the connections with DLAs, recent work has explored the use of the adjoint representation of a Lie algebra to efficiently simulate polynomially-sized DLAs [280, 281]. With this in mind, it should be possible to perform the gradient method of Section 3.3.2 in the adjoint representation, which is explored in a recent work [282].

Finally, we believe that the connections between variational quantum circuits and representation theory merit further investigation. We connected the classification of all $\mathbb{SU}(N)$ gates with the classification of semisimple Lie algebras. However, this could possibly be extended to a classification of all variational quantum circuits based on the DLA of an ansatz. It seems that the tools to provide such a classification are available and could provide one with a method to assess the trainability and expressivity of variational circuits without explicitly referring to specific ansätze. We study this problem in Chapter 4.

Chapter 4

A Lie algebra perspective of Variational Quantum Computing

We have already sporadically encountered DLA in previous sections. After our work in [1], subsequent investigation revealed that overparameterization phenomena and barren plateaus were intricately related to the DLA of a variational quantum circuit [227, 230]. In Section 3.2, we referred to the idea of a Riemannian gradient flow in the dynamical Lie algebra of a circuit (see Appendix E.1) and in Section 3.3 we encountered the DLA in our resource estimation of the gradient estimation procedure of the $\mathbb{SU}(N)$ gate. In this section, we dive deeper into the world of DLAs by classifying these objects for a large class of Hamiltonians and quantum circuits. Although we have mentioned Lie algebras and its properties before, we will here discuss them from more mathematical point of view, since this will be required to understand the results in Section 4.2.

4.1 Lie algebras

The purpose of this section is to review some standard terminology concerning Lie algebras, especially $\mathfrak{su}(N)$, $\mathfrak{so}(N)$, and $\mathfrak{sp}(N)$.

The set $\mathbb{C}^{N\times N}$ of $N \times N$ matrices with complex entries is a vector space over $\mathbb{C}$, as matrices can be added or multiplied by complex scalars. If we equip $\mathbb{C}^{N\times N}$ with a matrix product ab, it becomes an *associative algebra*, i.e., ab is bilinear (depends linearly on both a or b) and associative: $a(bc) = (ab)c$. Under the *commutator bracket* $[a, b] = ab - ba$, we

get a complex Lie algebra denoted as $\mathfrak{gl}(N,\mathbb{C})$, which is known as the general linear Lie algebra.

In general, a *Lie algebra* is defined as a vector space $\mathfrak{g}$ (over $\mathbb{C}$), equipped with a bilinear operation $[a,b] \in \mathfrak{g}$ for $a,b \in \mathfrak{g}$, which satisfies the following skew-symmetry and Jacobi identity:

$$[a,b] = -[b,a] \tag{4.1}$$
$$[a,[b,c]] = [[a,b],c] + [b,[a,c]]. \tag{4.2}$$

It is convenient to use the notation $\mathrm{ad}_a(b) := [a,b]$ for $a,b \in \mathfrak{g}$. Then ad_a is a linear operator on $\mathfrak{g}$ for every $a \in \mathfrak{g}$. This is straightforward to see, since

$$\mathrm{ad}_a = [a,.] \tag{4.3}$$
$$\mathrm{ad}_a(b+c) = [a,b+c] = \mathrm{ad}_a(b) + \mathrm{ad}_a(c). \tag{4.4}$$

In fact, since the Lie algebra is closed under the commutator, ad_a is an *endomorphism* on $\mathfrak{g}$: a linear map from a vector space onto itself. Remember that we encountered this map (and its group equivalent) already in Section 3.1.8 when we were studying the Baker-Campbell-Hausdorff formula as the adjoint operator. A trivial example of a Lie algebra is any vector space $\mathfrak{g}$ with the zero bracket $[a,b] = 0$ for all $a,b \in \mathfrak{g}$; such Lie algebras are called *Abelian*.

A *subalgebra* $\mathfrak{s}$ of a Lie algebra $\mathfrak{g}$ is a subspace (i.e., closed under vector addition and scalar multiplication), which is also closed under the bracket: $a,b \in \mathfrak{s} \Rightarrow [a,b] \in \mathfrak{s}$. For example, the set $\mathfrak{sl}(N,\mathbb{C})$ of all $N \times N$ complex matrices with trace 0 is a subalgebra of $\mathfrak{gl}(N,\mathbb{C})$. This is straightforward to show if we take $c = [a,b]$ with $a,b \in \mathfrak{sl}(N,\mathbb{C})$

$$\mathrm{Tr}\{[a,b]\} = \mathrm{Tr}\{ab - ba\} = \mathrm{Tr}\{ab\} - \mathrm{Tr}\{ba\} = 0, \tag{4.5}$$

by linearity and cyclicity of the trace. Hence, $\mathfrak{sl}(N,\mathbb{C})$ is itself a Lie algebra. We could have also realized that the commutator is skew-symmetric, hence it must always produce a traceless object.

An *ideal* in a Lie algebra $\mathfrak{g}$ is a subspace $\mathfrak{s}$ such that $a \in \mathfrak{g}$, $b \in \mathfrak{s} \Rightarrow [a,b] \in \mathfrak{s}$. Hence, any element in $\mathfrak{g}$ gets projected back into $\mathfrak{s}$ under the commutation with an element in $\mathfrak{s}$. For example, both $\mathfrak{sl}(N,\mathbb{C})$ and $\mathbb{C}I_N$ (where I_N is the $N \times N$ identity matrix) and $z \in \mathbb{C}$ are ideals of $\mathfrak{gl}(N,\mathbb{C})$. For $\mathbb{C}I_N$, every element in $\mathfrak{g}$ gets sent to 0 under commutation:

$$[a,b] = 0, \forall a \in \mathfrak{g}, b \in \mathbb{C}I_N, \tag{4.6}$$

and since a Lie algebra is a vector space it always contains the zero element, hence $0 \in \mathfrak{s}$. The space $\mathbb{C}I_N$ is the *center* $Z(\mathfrak{g})$ of $\mathfrak{g} = \mathfrak{gl}(N, \mathbb{C})$, i.e., the set of all $c \in \mathfrak{g}$ such that $[c, a] = 0$ for all $a \in \mathfrak{g}$. We can think of the center as the kernel of the adjoint map ad.

From Equation (4.5) it should be clear that $[a, b] \in \mathfrak{sl}(N, \mathbb{C})$ for all $a \in \mathfrak{g}$ and all $b \in \mathfrak{sl}(N, \mathbb{C})$. Furthermore, we note that

$$\mathfrak{gl}(N, \mathbb{C}) = \mathfrak{sl}(N, \mathbb{C}) \oplus \mathbb{C}I_N$$

is a direct sum of not just subspaces but of commuting subalgebras and ideals. When we write a direct sum of Lie algebras, we will always mean that the summands are subalgebras that commute with each other.

4.1.1 The Lie algebras $\mathfrak{su}(N)$, $\mathfrak{so}(N)$, and $\mathfrak{sp}(N)$

The Lie algebra $\mathfrak{sl}(N, \mathbb{C})$ is *simple*, which means that it is not Abelian and has no ideals other than the trivial $\{0\}$ and the whole algebra. The absence of ideals turns simple Lie algebras into the elementary objects of study, since they contain no subspaces that act in a non-trivial way on elements of the algebra. If we take a direct sum of simple Lie algebras, we obtain a *semisimple* Lie algebra.

Other examples of simple Lie algebras over $\mathbb{C}$ are provided by the *orthogonal* Lie algebras $\mathfrak{so}(N, \mathbb{C})$ and the *symplectic* Lie algebras $\mathfrak{sp}(2N, \mathbb{C})$. Let us recall that $\mathfrak{so}(N, \mathbb{C})$ is defined as the set of all complex skew-symmetric matrices (i.e., such that $a^T = -a$), and it is a subalgebra of $\mathfrak{sl}(N, \mathbb{C})$. Consider the $2N \times 2N$ matrix

$$J_{2N} := \begin{pmatrix} 0 & I_N \\ -I_N & 0 \end{pmatrix}.$$

Then $\mathfrak{sp}(2N, \mathbb{C})$ is defined as the set of all $a \in \mathfrak{gl}(2N, \mathbb{C})$ such that $a^T J_{2N} = -J_{2N} a$; this is a subalgebra of $\mathfrak{sl}(2N, \mathbb{C})$.

Up to this point, we have considered complex Lie algebras, and we will now consider their real counterparts; Lie algebras over $\mathbb{R}$.

The set of all skew-Hermitian matrices (i.e., satisfying $a^\dagger = -a$) is a real vector space, and is closed under the commutator; hence, it is a real Lie algebra, denoted $\mathfrak{u}(N)$. Imposing that the trace of the matrix is 0, we get the Lie algebra $\mathfrak{su}(N)$.

$$\mathfrak{su}(N) := \big\{ a \in \mathfrak{sl}(N, \mathbb{C}) \,\big|\, a^\dagger = -a \big\}. \tag{4.7}$$

If a matrix has real entries, then it is skew-Hermitian if and only if it is skew-symmetric. Thus, we have the subalgebra

$$\mathfrak{so}(N) := \mathfrak{so}(N, \mathbb{R}) = \left\{ a \in \mathfrak{su}(N) \,\middle|\, a^T = -a \right\} \subset \mathfrak{su}(N). \tag{4.8}$$

On the other hand, the Lie algebra of real symplectic matrices $\mathfrak{sp}(2N, \mathbb{R})$ is *not* a subalgebra of $\mathfrak{su}(2N)$. Instead of it, the relevant subalgebra is

$$\mathfrak{sp}(N) := \mathfrak{sp}(2N, \mathbb{C}) \cap \mathfrak{su}(2N) = \left\{ a \in \mathfrak{su}(2N) \,\middle|\, a^T J_{2N} = -J_{2N} a \right\} \subset \mathfrak{su}(2N). \tag{4.9}$$

The Lie algebras $\mathfrak{su}(N)$, $\mathfrak{so}(N)$, $\mathfrak{sp}(N)$ are simple and *compact* (they are Lie algebras of compact Lie groups). One way to understand compactness is through the corresponding Lie group it generates. When a Lie algebra $\mathfrak{g}$ is compact and simple Lie algebra, one can associate to it a compact Lie group G via the exponential map $G = e^{\mathfrak{g}}$. For our purposes, the specifics of compactness are not that important, we can understand it as a topological property of the generated Lie group that makes it "nice".

The dimensions of these Lie algebras over $\mathbb{R}$ are given by:

$$\dim \mathfrak{su}(N) = N^2 - 1, \qquad \dim \mathfrak{so}(N) = \frac{1}{2} N(N-1), \qquad \dim \mathfrak{sp}(N) = N(2N+1). \tag{4.10}$$

There are five other Lie algebras called the *exceptional* Lie algebras that will be of little importance to us, since they will not arise naturally in the setting of Section 4.2. It is known (see e.g. [283, 27]) that any subalgebra of $\mathfrak{u}(N)$ is either Abelian or a direct sum of a center (which could be $\{0\}$) and Lie algebras isomorphic to one of $\mathfrak{su}$, $\mathfrak{so}$, $\mathfrak{sp}$ or to one of five exceptional compact simple Lie algebras. For completeness, we provide the proof of this important fact here.

Proposition 4.1.1. *Any subalgebra of $\mathfrak{u}(N)$ is either Abelian or a direct sum of compact simple Lie algebras and a center.*

Proof. First, recall that the trace form $\mathrm{Tr}\{ab\}$ is negative definite on $\mathfrak{u}(N)$. Indeed, one can see that

$$\mathrm{Tr}\left\{H^2\right\} = \sum_{j=1}^{N} \lambda_j^2 > 0$$

for any nonzero Hermitian matrix $H \in i\mathfrak{u}(N)$ with eigenvalues $\lambda_1, \ldots, \lambda_N$, because all λ_j are real. Second, the trace is bilinear, symmetric and invariant; the latter meaning that

$$\mathrm{Tr}\{[a,b],c\} = -\mathrm{Tr}\{b,[a,c]\}, \qquad a, b, c \in \mathfrak{u}(N).$$

All of these follow easily from the properties of the trace.

Now let $\mathfrak{g}$ be a subalgebra of $\mathfrak{u}(N)$. Then the same proof as in [284], Corollary 4.25, shows that $\mathfrak{g}$ is *reductive*, i.e., it is a direct sum of simple or Abelian ideals. Indeed, for any ideal $\mathfrak{s}$ of $\mathfrak{g}$, we have an orthogonal direct sum

$$\mathfrak{g} = \mathfrak{s} \oplus \mathfrak{s}^{\perp},$$

due to the definiteness of the trace form. Furthermore, the invariance of the trace form implies that $\mathfrak{s}^{\perp}$ is itself an ideal of $\mathfrak{g}$. As both $\mathfrak{s}$ and $\mathfrak{s}^{\perp}$ are ideals, they must commute: $[\mathfrak{s}, \mathfrak{s}^{\perp}] \subseteq \mathfrak{s} \cap \mathfrak{s}^{\perp} = \{0\}$. Thus, $\mathfrak{g}$ is a direct sum of commuting ideals. Suppose that $\dim \mathfrak{s} > 1$ and $\mathfrak{s}$ is not simple as a Lie algebra. Then $\mathfrak{s}$ has a nonzero proper ideal $\mathfrak{t}$. From $[\mathfrak{t}, \mathfrak{s}] \subseteq \mathfrak{t}$ and $[\mathfrak{t}, \mathfrak{s}^{\perp}] = \{0\}$, we get that $[\mathfrak{t}, \mathfrak{g}] \subseteq \mathfrak{t}$, so $\mathfrak{t}$ is an ideal of $\mathfrak{g}$. Hence, we can proceed by induction on $\dim \mathfrak{g}$ and break $\mathfrak{g}$ into a direct sum of commuting ideals, each of which is either simple or 1-dimensional (Abelian).

Finally, we note that any simple subalgebra $\mathfrak{g}$ of $\mathfrak{u}(N)$ is compact, i.e., the Lie group $e^{\mathfrak{g}}$ is compact as a closed subgroup of the compact Lie group $\mathrm{U}(N)$ (see e.g. [284], Proposition 4.27, whose proof still applies). The classification of all compact simple Lie algebras is due to Cartan, and can be found in standard textbooks; for example in [284], Chapter VI. □

Let now $N = 2^n$ where n is the number of qubits. Then a basis for $\mathfrak{u}(2^n)$ is given by the Pauli strings $i\mathcal{P}_n$ (see Appendix A.1). To get a basis for $\mathfrak{su}(2^n)$, we just have to remove $iI^{\otimes n}$, since $I^{\otimes n} = I_N$ is the identity matrix.

Similarly, a basis for $\mathfrak{so}(2^n)$ consists of all ia where $a \in \mathcal{P}_n$ is a Pauli string containing an odd number of Y's, since this ensures that they are skew-symmetric instead of skew-Hermitian.

Finally, to describe the subalgebra $\mathfrak{sp}(2^{n-1}) \subset \mathfrak{su}(2^n)$, we observe that

$$Y_1 = Y \otimes I^{\otimes(n-1)} = -i \begin{pmatrix} 0 & I_{\frac{N}{2}} \\ -I_{\frac{N}{2}} & 0 \end{pmatrix} = -iJ_N.$$

Therefore,

$$\mathfrak{sp}(2^{n-1}) = \left\{ a \in \mathfrak{su}(2^n) \,\middle|\, a^T \cdot Y_1 = -Y_1 \cdot a \right\}. \tag{4.11}$$

4.1.2 Dynamical Lie Algebras

We now give a formal definition of a dynamical Lie algebra that we will use throughout the rest of this thesis.

Definition 4.1.1. *For a Lie algebra $\mathfrak{g}$ and a subset $\mathcal{A} \subset \mathfrak{g}$, we define $\langle \mathcal{A} \rangle_{\text{Lie}}$ to be the minimal (under inclusion) subalgebra of $\mathfrak{g}$ that contains $\mathcal{A}$. We say that $\langle \mathcal{A} \rangle_{\text{Lie}}$ is the subalgebra* generated *by $\mathcal{A}$, and that $\mathcal{A}$ is a set of* generators *of $\langle \mathcal{A} \rangle_{\text{Lie}}$. In the case when $\mathcal{A} \subset \mathcal{P}_n$ is a set of Pauli strings, we will slightly abuse the notation and write $\langle \mathcal{A} \rangle_{\text{Lie}}$ for the subalgebra of $\mathfrak{su}(2^n)$ generated by the subset $i\mathcal{A} \subset \mathfrak{su}(2^n)$.*

More explicitly, it follows from the Jacobi identity that $\langle \mathcal{A} \rangle_{\text{Lie}}$ is the set of all linear combinations of all nested commutators of the form

$$\text{ad}_{a_1}\text{ad}_{a_2}\cdots\text{ad}_{a_r}(a_{r+1}) = [a_1, [a_2, [\cdots [a_r, a_{r+1}]\cdots]]], \qquad a_j \in \mathcal{A}, \;\; r \geq 0, \tag{4.12}$$

where $r = 0$ corresponds to an empty commutator $= a_1$. In the context of Proposition 4.1.1, this means that any dynamical Lie algebra will be a subalgebra of $\mathfrak{u}(2^n)$ that is either Abelian or a direct sum of compact simple Lie algebras and a center. Since these objects have been classified, we can therefore always express a DLA in terms of these known constituents.

The following observation will be useful.

Lemma 4.1.1. *For any subset $\mathcal{A} \subset \mathcal{P}_n$, the Lie algebra $\langle \mathcal{A} \rangle_{\text{Lie}} \subseteq \mathfrak{su}(2^n)$ generated by $i\mathcal{A}$ has a basis consisting of Pauli strings times i. In other words,*

$$\langle \mathcal{A} \rangle_{\text{Lie}} = \text{span}_{\mathbb{R}}\big(i\mathcal{P}_n \cap \langle \mathcal{A} \rangle_{\text{Lie}}\big).$$

Proof. By definition, $\langle \mathcal{A} \rangle_{\text{Lie}}$ is linearly spanned over $\mathbb{R}$ by all elements of the form Equation (4.12) with $a_j \in i\mathcal{A} \subset i\mathcal{P}_n$. All such elements are scalar multiples of Pauli strings, i.e., lie in $i\mathbb{R}\mathcal{P}_n$. From any spanning set, one can choose a subset that forms a basis. □

This Lemma will enable us to work with Pauli strings, which have useful algebraic properties that we can exploit.

4.2 Classification of dynamical Lie algebras in one dimension

Mathematical classifications of the fundamental symmetries of physical systems date back to the work of Wigner, who proposed three symmetry classes of non-interacting fermionic Hamiltonians depending on their time-reversal and spin-rotation properties [285]. Three decades later, Dyson would mathematically solidify this theory and connect the spectral properties of these different types of Hamiltonians with random matrix theory [286] (see [287] for a modern treatment). It would take another thirty years before Altland and Zinbauer extended these results to ten symmetry classes [288], each of which correspond to a symmetric space in Cartan's original classification of these spaces [289, 290]. Further extensions of these results were made in recent years with regard to topological phases of matter [291, 292, 293].

The above mathematical classifications in quantum physics rest on the powerful theory of Lie groups, which provides a framework for describing the continuous symmetries and transformations that characterize the behavior of quantum systems. The study of Lie groups, and by extension physical symmetries, can often be simplified by considering the corresponding Lie algebra of the group. The commutation relations of operators in the Lie algebra capture the essential features of the underlying symmetries and can be used to analyze the spectrum, eigenstates, and dynamics of quantum systems.

A Hamiltonian of a finite-dimensional system can be understood as (i times) an element of some Lie algebra $\mathfrak{g} \subseteq \mathfrak{su}(N)$. Here, $\mathfrak{su}(N)$ is the vector space of all skew-Hermitian $N \times N$ matrices equipped with the standard commutator. Typically, a Hamiltonian is described by a linear combination of terms that correspond to a certain physical interaction. The algebra formed by taking all (finite) products and linear combinations of the terms in a Hamiltonian is called the *bond algebra* of the Hamiltonian [294, 295, 296]. Bond algebras have been studied extensively to understand the symmetries and spectra of different classes of Hamiltonians. More recently, they have been used to study thermalization phenomena in quantum many-body systems [297, 298]. Instead of studying the algebra formed by taking products of individual interactions in the Hamiltonian, we can study the Lie algebra generated by these terms under commutation. The result is called the Hamiltonian algebra or *dynamical Lie algebra* (DLA) [299, 300, 301, 302, 303], which is intricately linked to the unitary dynamics of a quantum system.

Since each DLA is a subalgebra of $\mathfrak{su}(N)$, a classification of DLAs can be phrased as a classification of all subalgebras of $\mathfrak{su}(N)$. Such a classification is intractable, except when specific constraints are placed on the subalgebras one considers. For example, in the original

works of Killing and Cartan, all *simple* Lie algebras were classified [304]; similarly, Dynkin provided a classification of the *maximal* subalgebras of simple Lie algebras [305, 306]. We follow a different approach: instead of adding algebraic constraints such as simplicity or maximality, we make use of the fact that any Lie algebra can be described by a set of generators, and we consider the Lie algebras that arise by using the terms of specific Hamiltonians as the generators. In contrast with the previously-mentioned classifications of [285, 286, 288], this approach covers interacting quantum many-body systems.

Specifically, we consider the class of Hamiltonians that correspond to 1-dimensional 2-local spin chains, and provide a classification of the Lie algebras that are generated under commutation by the individual Pauli terms of the Hamiltonian. Much about these systems is well-understood, from their entanglement properties [76], their phases [307] and their integrability [308, 309]. Additionally, they show up in the study of bond algebras [294, 298]. However, to the best of our knowledge, the Lie algebraic properties of these Hamiltonians have not yet been explored in full. It is thus reasonable to ask, given our comprehensive knowledge of the physics governing these systems, what more can be learned from the Lie algebra? We provide an answer to this question by classifying all unique dynamical Lie algebras of linear, circular and permutation invariant spin systems generated by 2-local Pauli operators. We discuss how our classification has bearing on areas of quantum control, variational quantum computing, and quantum dynamics and thermodynamics.

For *variational quantum computing*, one is not interested in representing the whole unitary group, but in using a parameterized subgroup in order to generate a state that maximizes a given objective function. Understanding what subgroup a particular quantum circuit parameterizes can give insight into its representational power. For example, one can connect the dimension of the DLA to a phenomenon called overparameterization [74, 1, 231, 230]. Additionally, DLAs can be used to understand barren plateaus [227] — flat areas in the cost landscape of a variational quantum algorithm that hinder optimization [35, 38]. Finally, a recent work uses knowledge of the DLA to perform efficient classical simulations of several quantum algorithms [281]. We discuss these results by showing that barren plateau and overparameterization phenomena show for several examples found in our classification as expected, which highlights the usefulness of classifying sets of quantum circuits by their Lie algebraic properties.

In *quantum control*, the DLA of a dynamical quantum system can be related to the set of reachable states of that system. In particular, DLAs can be used to define a notion of controllability of a quantum system [220, 310, 311], which is highly relevant when it comes to designing unitary operations for quantum simulators and quantum computers. One is typically interested in Hamiltonians that can generate an arbitrary unitary operator, while the existence of symmetries can inhibit the control of a physical system [283].

Finally, one can use the knowledge of the DLA to provide insights into the *dynamics* of physical systems. Here, it should be noted that the physical properties of models described by the Hamiltonian whose terms generate the DLA strongly depend on the coefficients of said terms. Our classification, which neglects the coefficients, is thus limited to the properties that belong to the entire class of models described by the same terms. While this is a coarse classification, we can nevertheless make quite general observations from just the study of the DLA. For example, one can construct constant-depth quantum circuits for the dynamical simulation of a specific quantum system [228, 312, 313, 312], or state preparation via adiabatic state preparation [312, 312], or implement Hartree–Fock [99]. The dimension of the DLA is directly related to the quantum circuit depth needed to capture the full dynamics [228]. Additionally, Non-Abelian commutants describe non-Abelian symmetries.

This section is structured as follows. We end the introduction with a summary of our main mathematical results. Then, in Section 4.3, we establish our notation and introduce the necessary mathematical preliminaries. We discuss the method of our classification in Section 4.4 and present the main results in Section 4.5. Finally, we discuss the implications of our results in Section 4.6. In the Appendix we review preliminaries present the full details of the proofs of the main results.

4.2.1 Summary of the main results

Here, we give a brief summary of our main results, which include the classification of all DLAs generated by 2-local spin Hamiltonians of length n in one dimension. We emphasize that our method is not limited to one-dimensonal topologies, but can be extend to other graphs, which we will explore in a follow-up work [314]. Recall that a Lie algebra can be constructed by a set of generators so that it is closed under linear combinations and under the Lie bracket. In our case, the Lie bracket is the standard commutator $[A, B] = AB - BA$. We now choose the generators of our Lie algebra to be (i times) the terms of a 2-local spin chain Hamiltonian generated by Pauli strings. Since a Hamiltonian is always a Hermitian operator, we can understand it as (i times) an element of the Lie algebra $\mathfrak{u}(2^n)$. Therefore, we can limit ourselves to the study of DLAs that are subalgebras of $\mathfrak{u}(2^n)$, for which we have the following useful fact. Although this result is known (see e.g. [283, 27]), for completeness, we provide its proof and review the necessary definitions in Section 4.1.1.

Proposition 4.2.1. *Any DLA must be either Abelian, isomorphic to* $\mathfrak{su}(N')$, $\mathfrak{so}(N')$, $\mathfrak{sp}(N'')$ *(with* $N' \leq 2^n$, $N'' \leq 2^{n-1}$*), an exceptional compact simple Lie algebra, or a direct sum of such Lie algebras. Indeed, any subalgebra of* $\mathfrak{u}(N)$ *is either Abelian or a direct sum of compact simple Lie algebras and a center.*

Note that all simple Lie algebras over the complex and real numbers have been classified by Killing and Cartan [250]. The above proposition forms the backbone of our classification, as we know that any DLA generated by our class of Hamiltonians must be of the described form.

To obtain our classification, we first calculate all DLAs that can be generated by Pauli strings of length 2. Then we identify the orbits under the symmetries of the Pauli group and the swap of the two sites, thus reducing the number of unique Lie algebras to 27. Next, we find several isomorphisms between some of the sets of generators, reducing the set of unique structures even further. Finally, we determine how these Lie algebras scale with system size as the number of spins grows beyond 2 sites. In this final step, we take the boundaries of the spin chain into account, since the Lie algebra will behave differently for open or periodic boundary conditions of the chain. The following is our main result.

Result 4.2.1 (Classification of spin chain DLAs)**.** *We provide a classification of all dynamical Lie algebras of 2-local spin Hamiltonians in one dimension. For both open and closed spin chains, there are 17 unique Lie algebras that can be generated by a spin chain Hamiltonian.*

The formal statement of this result is presented in the main text with Theorems 4.5.1 and 4.5.2 along with a sketch of the proof.

The dimension of a DLA can be related to the trainability of variational quantum circuits, and may therefore be of high interest. Since we know the dimensions of all simple Lie algebras, a direct corollary of our result is the following.

Result 4.2.2 (Dimension scaling of DLAs)**.** *The dimension of any dynamical Lie algebra of a 2-local spin chain Hamiltonian of length n will scale as either $O(4^n)$, $O(n^2)$ or $O(n)$.*

To illustrate this, we plot the dimensions of the open DLAs in our classification in Figure 4.1.

Our proof technique also applies to the case of a permutation-invariant graph, where each site is interacting with every other site via at most 2-local interactions; in other words, all-to-all connected. We therefore also include this topology in our classification.

Result 4.2.3 (Classification of permutation invariant DLAs)**.** *We provide a classification of all dynamical Lie algebras of 2-local permutation-invariant spin Hamiltonians. There are 8 unique Lie algebras that can be generated by such a spin chain Hamiltonian.*

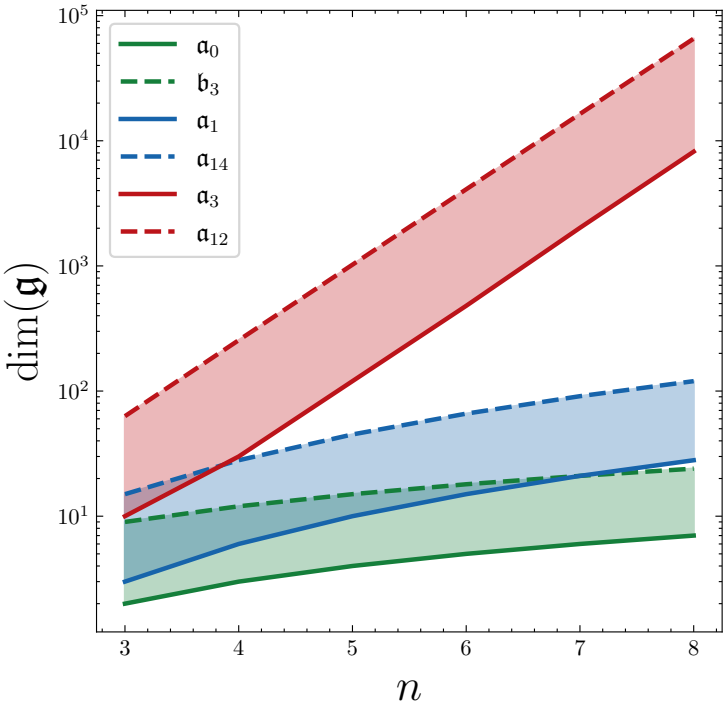


Figure 4.1: **Scaling of the DLAs of spin chains with open boundary conditions.** The exponentially scaling DLAs are denoted in red, the quadratically scaling ones in blue, and the linearly scaling algebras are denoted in green. The full and dashed line denote the smallest and largest scaling algebra in our classification, respectively.

We present the formal statement of this result in Theorem 4.5.3. Similarly to Result 4.2.2, we find DLAs with linear, quadratic and exponentially scaling dimensions.

In addition to the classification of unique Lie algebraic structures, we also provide an explicit list of Hamiltonians that generate them in Table G.1. Some of the generators in our classification show up in well-known models such as the transverse-field Ising model or the Heisenberg model, whereas other Hamiltonians we find are perhaps not realizable in nature. However, some of these more exotic models may be of interest due to their properties. For instance, we find a class of Hamiltonians with globally non-commuting charges.

4.3 Background

4.3.1 Preliminaries

We assume knowledge of finite-dimensional Lie algebras (for a formal treatment, see e.g. Refs. [290, 250]), but will review some essential concepts here. A Lie algebra $\mathfrak{g}$ is a vector space equipped with a Lie bracket $[\cdot,\cdot]\colon \mathfrak{g}\times\mathfrak{g}\to\mathfrak{g}$ satisfying certain axioms (which are reviewed in Section 4.1.1). The Lie bracket defines the adjoint endomorphism $\mathrm{ad}_a\colon \mathfrak{g}\to\mathfrak{g}$ where $\mathrm{ad}_a(b)=[a,b]$. For our purposes, the Lie bracket is the standard commutator of linear operators on a vector space: $[a,b]=ab-ba$. When $\mathfrak{g}$ is a compact simple Lie algebra (cf. Section 4.1.1 and Proposition 4.2.1), one can associate to it a compact Lie group G via the exponential map $G=e^{\mathfrak{g}}$.

Consider a set of generators $\mathcal{A}=\{a_1,a_2,\ldots,a_M\}$ with $a_k\in\mathfrak{g}$. We first define the nested commutator,

$$\mathrm{ad}_{a_{i_1}}\cdots\mathrm{ad}_{a_{i_r}}(a_j)=[a_{i_1},[a_{i_2},[\cdots[a_{i_r},a_j]\cdots]]], \tag{4.13}$$

which is just a_j in the special case $r=0$. The linear span of all nested commutators

$$\langle\mathcal{A}\rangle_{\mathrm{Lie}}:=\mathrm{span}_{\mathbb{R}}\big\{\mathrm{ad}_{a_{i_1}}\cdots\mathrm{ad}_{a_{i_r}}(a_j)\,\big|\,a_{i_1},\ldots,a_{i_r},a_j\in\mathcal{A}\big\}$$

is then called a *dynamical Lie algebra* (DLA) [27, 299]. This is the minimal (under inclusion) subalgebra of $\mathfrak{g}$ that contains the set $\mathcal{A}$. The depth r of the nested commutator is finite and will depend on the size of the DLA, which we typically do not know beforehand. In practice, the DLA of a given set of generators $\mathcal{A}$ can be obtained recursively with Algorithm 3.

Algorithm 3: Calculating the DLA

```
Input: Set of generators A
for a_i ∈ A do
    for a_j ∈ A do
        a_k = [a_i, a_j]
        if a_k ∉ span(A) then
            A ← A ∪ {a_k}
⟨A⟩_Lie ← span(A)
```

Remark 4.3.1. *We are interested in the case where the generators $\mathcal{A}$ are Pauli strings. Then, since the commutator of two Pauli strings is up to a scalar again a Pauli string, Algorithm 3 can be simplified by replacing the condition $a_k \notin \operatorname{span}(\mathcal{A})$ with $a_k \notin \mathcal{A} \cup \{0\}$. Moreover, as different Pauli strings are linearly independent, the final set $\mathcal{A}$ will be a basis for the DLA $\langle \mathcal{A} \rangle_{\text{Lie}}$.*

4.3.2 2-local spin systems in one dimension

Due to Proposition 4.2.1, we know what form the subalgebras of $\mathfrak{su}(N)$ must take. Our goal is to find which of these direct sums of simple or Abelian Lie algebras can be generated by a physically inspired set of generators.

In particular, we are interested in the subalgebras of $\mathfrak{su}(2^n)$ that are generated by the terms of 1-dimensional 2-local Hamiltonians. By this we mean that we consider operators that couple up to two neighboring sites and where the interactions between neighboring sites is the same, but the interaction strength may vary per interaction, and could potentially depend on time due to an external field allowing control over the system. We consider a spin system with a complex Hilbert space $(\mathbb{C}^2)^{\otimes n}$ Hamiltonian H, which is a Hermitian operator on $(\mathbb{C}^2)^{\otimes n}$ of the form

$$H = \sum_{k=1}^{n-1} \sum_{a \otimes a' \in \mathcal{A}} J_{k,a,a'} a_k \otimes a'_{k+1}, \tag{4.14}$$

where $J_{k,a,a'}$ are arbitrary real coefficients, and

$$a_k \otimes a'_{k+1} := I^{\otimes(k-1)} \otimes a \otimes a' \otimes I^{\otimes(n-k-1)}, \tag{4.15}$$

with $a \otimes a' \in \mathcal{A}$ and I is the 2×2 identity matrix. We consider $a, a' \in \mathcal{P}_1 := \{I, X, Y, Z\}$ (Pauli matrices), and

The generating set $\mathcal{A} \subseteq \mathcal{P}_2 := \mathcal{P}_1^{\otimes 2}$ defines a specific set of 2-local operators that make up the Hamiltonian H; in the parlance of quantum computing and physics, this is a 2-local Hamiltonian corresponding to a *spin chain*. Note that physical models come with coefficients $J_{k,a,a'}$ in front of each $a_k \otimes a'_{k+1}$ term that determine the equilibrium and non-equilibrium physics of the model. Here, we are only concerned with the algebraic properties of the Hamiltonian H on the Lie algebra level, and we will not consider any spectral properties of H. Additionally, we note that we are considering a *directed* interaction graph, since we allow terms like $X \otimes Y$, which do not act symmetrically between neighboring qubits.

Continuing, we note that Pauli matrices $i\mathcal{P}_1 \setminus \{iI\}$ form a basis of $\mathfrak{su}(2)$, and the tensor products $i\mathcal{P}_2 \setminus \{iI^{\otimes 2}\}$ form a basis of $\mathfrak{su}(4)$. Hence, $\mathrm{span}(\mathcal{A}) \subseteq \mathfrak{su}(4)$ (recall that $\mathrm{span} = i\,\mathrm{span}_{\mathbb{R}}$). In the following, we will suppress the tensor product between Pauli operators and identities for clarity, and we denote $a \otimes a' = aa'$. We now give some examples to illustrate how several well-known spin chains can be written in this notation.

Example 4.3.1. *Random field Ising model .* The Hamiltonian of the TFIM in one dimension with open boundary conditions and random transverse field is given by

$$H_{\mathrm{TFIM}} = J \sum_{i=1}^{n-1} Z_i Z_{i+1} + \sum_{i=1}^{n} g_i X_i,$$

where $J, g_i \in \mathbb{R}$. As it was stated before, we only consider the Pauli strings in this Hamiltonian. Thus, we consider its generating set of Pauli strings, which is the following

$$\mathcal{A}_{\mathrm{TFIM}} := \{ZZ, XI, IX\}.$$

Example 4.3.2. *XXZ chain.* For the 1-dimensional XXZ chain with open boundary conditions, the Hamiltonian is given by

$$H_{\mathrm{XXZ}} = \sum_{i=1}^{n-1} (X_i X_{i+1} + Y_i Y_{i+1} + \Delta Z_i Z_{i+1}),$$

which has generators

$$\mathcal{A}_{\mathrm{XXZ}} := \{XX, YY, ZZ\}.$$

Example 4.3.3. *Spinless fermionic Gaussian state.* A free fermion Hamiltonian chain in one dimension with periodic boundary conditions can be built from the generators on two sites:

$$c_1^\dagger c_2^\dagger, c_1^\dagger c_1, c_2^\dagger c_2, c_1^\dagger c_2, c_2^\dagger c_1, c_1 c_2,$$

where $c^\dagger$ and c are fermionic raising and lowering operators, respectively. The corresponding Hamiltonian is

$$H_{\mathrm{FF}} = \sum_{i=1}^{n-1} (c_i^\dagger c_{i+1}^\dagger + c_i^\dagger c_{i+1} + c_i c_{i+1}^\dagger + c_i c_{i+1}).$$

The fermionic raising and lowering operators may be translated to Pauli string form via a number of transformations. If we use the common Jordan–Wigner transformation, the resulting set of Pauli generators is

$$\mathcal{A}_{\mathrm{FF}} := \{XX, ZI, IZ, YY, XY, YX\}.$$

4.3.3 Growing the dynamical Lie algebras

Since we are only interested in the algebraic structure of the class of Hamiltonians, we will focus on the structure of the generators and ignore the coefficients in front of the individual Pauli terms. On a 2-qubit system, these terms will generate a Lie algebra $\langle \mathcal{A} \rangle_{\text{Lie}} = \mathfrak{a}$ that is a subalgebra of $\mathfrak{su}(4)$. We now investigate the structure of these algebras as we add terms that have been translated by one site. Starting from a subalgebra $\mathfrak{a} \subseteq \mathfrak{su}(4)$, let $\mathfrak{a}(n)$ be the subalgebra of $\mathfrak{su}(2^n)$ generated by the set

$$\bigcup_{1 \leq k \leq n-1} I^{\otimes(k-1)} \otimes \mathfrak{a} \otimes I^{\otimes(n-k-1)}.$$

In particular, $\mathfrak{a}(2) = \mathfrak{a}$. By construction, we have a Lie algebra embedding $\mathfrak{a}(n) \hookrightarrow \mathfrak{a}(n+1)$, given by appending I to the last qubit (see Figure 4.2).

Example 4.3.4. Consider the generating set $\mathcal{A} = \{XY\}$ on 2 qubits. The DLA is given by

$$\langle \mathcal{A} \rangle_{\text{Lie}} = \text{span}\{XY\},$$

which is an Abelian Lie algebra isomorphic to $\mathfrak{u}(1)$. Constructing the generators of $\mathfrak{a}(3)$ according to the procedure above gives

$$\mathfrak{a}(3) = \langle XYI, IXY \rangle_{\text{Lie}} = \text{span}\{XYI, IXY, XZY\}.$$

It is easy to confirm that $\mathfrak{a}(3) \cong \mathfrak{so}(3)$. We see that going from $n = 2$ to $n = 3$, the DLA changes from $\mathfrak{u}(1)$ to $\mathfrak{so}(3)$.

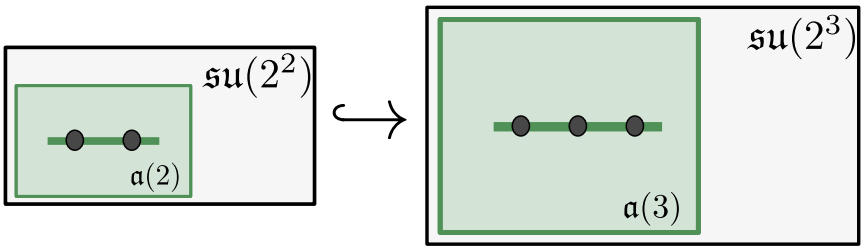


Figure 4.2: **Growing a Lie algebra by adding a site to the chain.** Increasing the system size from 2 sites to 3 sites changes the DLA from $\mathfrak{a}(2)$ to $\mathfrak{a}(3)$ within $\mathfrak{su}(4)$ and $\mathfrak{su}(8)$, respectively. The goal of our classification is to describe the behavior for any $n \geq 3$.

The above example illustrates that the algebraic structure of a DLA can change as we increase the system size. Additionally, when we extend the number of sites to $n > 2$, we need to take into account what happens at the edge of the chain. For 1-dimensional systems, this leads to two cases: open boundary conditions (operators on a line) and periodic boundary conditions (operators on a circle). We will denote the resulting DLAs of these two cases with $\mathfrak{a}(n)$ and $\mathfrak{a}^{\circ}(n)$, respectively.

4.4 Method

We can now state the central question of our work. Given a Hamiltonian of the form Equation (4.14), we seek a classification of all DLAs generated by the terms of the Hamiltonian, for $n \geq 3$ with both open and periodic boundary conditions.

4.4.1 The power sets

First, for the generators of $\mathfrak{a}$-type Lie algebras, we note that there are 9 Pauli strings that consist of two Pauli operators. Hence, the power set of the possible generators $\mathcal{A}$ contains $2^9 - 1 = 511$ elements. Similarly, for the $\mathfrak{b}$-type Lie algebras, there are 15 Pauli strings, which results in a power set of $2^{15} - 1 = 32767$ possible sets of generators. Clearly, the sets of generators of the $\mathfrak{a}$-type are included in the $\mathfrak{b}$-type power set. We thus only report the $\mathfrak{b}$-type Lie algebras that are not also $\mathfrak{a}$-types. Within the $\mathfrak{b}$-type set, there is a third class of Lie algebras, the $\mathfrak{c}$-type Lie algebras, which are an edge case where there are generators which Pauli strings of the form $a \otimes I$, but not the corresponding term $I \otimes a$. The structure of these Lie algebras is captured by the $\mathfrak{b}$-type Lie algebras, except for a small boundary effect at the last site in the chain. We therefore exclude the $\mathfrak{c}$-type Lie algebras from our classification (see Figure 4.3).

We proceed by going through all sets of generators $\mathcal{A}$ (of either $\mathfrak{a}$ or $\mathfrak{b}$-type) and use Algorithm 3 (in the simplified form from Remark 4.3.1) to perform the nested commutators in Equation (4.13). We then store only the unique subalgebras generated by this procedure, and we obtain only 127 and 19 generating sets for $\mathfrak{a}$-type and $\mathfrak{b}$-type, respectively. Since the largest power set we consider has only 32767 elements, this procedure can be performed numerically with ease. We provide the code to reproduce this at [267].

Example 4.4.1. Consider the generating sets:

$$\mathcal{A}_1 = \{XY, XZ\},$$
$$\mathcal{A}_2 = \{IX, XY\}.$$

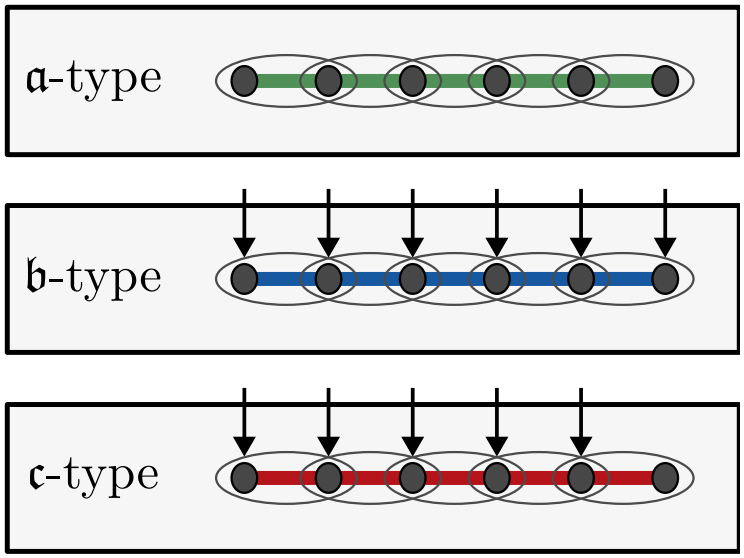


Figure 4.3: **The three types of Lie algebras in the classification.** The $\mathfrak{a}$-type Lie algebras can be generated only by nearest neighbor 2-Pauli interactions. The $\mathfrak{b}$-type Lie algebras are generated by nearest neighbor 2-Pauli interactions and 1-Pauli operators acting on every qubit; for $\mathfrak{c}$-type Lie algebras the 1-Pauli operators act differently on the boundary.

Note that $\mathcal{A}_1$ is of $\mathfrak{a}$-type and $\mathcal{A}_2$ is of $\mathfrak{c}$-type. After running Algorithm 3, we find:

$$\langle \mathcal{A}_1 \rangle_{\text{Lie}} = \text{span}\{XY, XZ, IX\},$$
$$\langle \mathcal{A}_2 \rangle_{\text{Lie}} = \text{span}\{IX, XY, XZ\}.$$

We see that $\mathcal{A}_1$ and $\mathcal{A}_2$ generate the same Lie algebra; hence, this Lie algebra is counted among the $\mathfrak{a}$-type Lie algebras.

Example 4.4.2. Consider the generating sets of $\mathfrak{a}$-type:

$$\mathcal{A}_1 = \{XX, YY\},$$
$$\mathcal{A}_2 = \{XX, YX\}.$$

After running Algorithm 3, we find:

$$\langle \mathcal{A}_1 \rangle_{\text{Lie}} = \text{span}\{XX, YY\},$$
$$\langle \mathcal{A}_2 \rangle_{\text{Lie}} = \text{span}\{XX, YX, ZI\}.$$

Hence, $\mathcal{A}_1$ and $\mathcal{A}_2$ generate distinct Lie algebras, both of $\mathfrak{a}$-type.

Example 4.4.3. The set of $\mathfrak{b}$-type

$$\mathcal{A} = \{XY, XI, IX\}$$

generates the Lie algebra

$$\langle \mathcal{A} \rangle_{\text{Lie}} = \text{span}\{XY, XZ, XI, IX\}.$$

If we try to generate it from the $\mathfrak{a}$-type subset $\{XY, XZ\}$, we find the strictly smaller subalgebra

$$\langle XY, XZ \rangle_{\text{Lie}} = \text{span}\{XY, XZ, IX\}.$$

4.4.2 Symmetries of the power sets

There are certain symmetries that can be exploited to reduce the number of subalgebras of $\mathfrak{su}(4)$ in the above power sets. To start, we note that the Pauli matrices satisfy the following algebraic relations:

$$[\sigma^\alpha, \sigma^\beta] = 2i \sum_{\gamma=1}^{3} \epsilon_{\alpha\beta\gamma} \sigma^\gamma,$$

where $\epsilon_{\alpha\beta\gamma}$ is the Levi-Civita tensor and $\alpha, \beta, \gamma \in \{1, 2, 3\}$, respectively (see Appendix A.1 for more details). We will ignore the factor $2i$, since we only care about the linear span of nested commutators. Note that the above relation is independent of how we assign X, Y, Z to σ^α. In other words, we can relabel the Paulis and retain the algebraic structure of the subalgebras, which together with ignoring the prefactors formally corresponds to an S_3 permutation symmetry.

In addition to relabelling, we consider the exchange of location of the two Pauli terms, since the order of such terms is an arbitrary choice that does not impact the structure of the resulting Lie algebras. This location exchange corresponds to a $\mathbb{Z}_2$ symmetry. Hence, the symmetry group of the Pauli algebra for $n = 2$ is $S_3 \times \mathbb{Z}_2$. Subalgebras of $\mathfrak{su}(4)$ that are in the same orbit of this symmetry group are considered equivalent, which allows us to reduce the number of subalgebras significantly.

Example 4.4.4. We have that $\{XX, YZ\} \equiv \{YY, ZX\}$ under relabeling $X \to Y \to Z \to X$. On the other hand, $\{XX, YZ\}$ and $\{XX, YY\}$ are not equivalent.

In order to determine the orbits of the symmetry group $S_3 \times \mathbb{Z}_2$ on the set of subalgebras of $\mathfrak{su}(4)$, we introduce their invariants s, p, e, d, defined as follows. These enumerate the number of single Paulis (such as XI) in the basis of the Lie algebra, the number of single Pauli pairs (such as XI, IX), the number of double equal Paulis (such as XX) and the number of double different Paulis (such as XY), respectively. Since all these quantities are invariant under the action of the symmetry group, two subalgebras are not equivalent if they have different invariants.

Example 4.4.5. Consider the following bases of subalgebras and their invariants:

$$\begin{aligned}\mathcal{A}_1 &= \{ZZ, YX, XY\} \to (0,0,1,2),\\ \mathcal{A}_2 &= \{XX, YZ, ZY\} \to (0,0,1,2),\\ \mathcal{A}_3 &= \{YY, ZX, XZ\} \to (0,0,1,2).\end{aligned}$$

We see that $\mathcal{A}_1 \equiv \mathcal{A}_2$ under $Z \rightleftharpoons X$. Similarly, $\mathcal{A}_2 \equiv \mathcal{A}_3$ under $X \rightleftharpoons Y$ and $\mathcal{A}_3 \equiv \mathcal{A}_1$ under $Y \rightleftharpoons Z$.

Example 4.4.6. Consider the following bases of subalgebras and their invariants:

$$\begin{aligned}\mathcal{A}_1 &= \{XX, XZ, IY\} \to (1,0,1,1),\\ \mathcal{A}_2 &= \{XY, XZ, IX\} \to (1,0,0,2).\end{aligned}$$

We see that $\mathcal{A}_1 \not\equiv \mathcal{A}_2$, since they have different invariants.

Example 4.4.7. Even though the two bases

$$\begin{aligned}\mathcal{A}_1 &= \{XY, YX\} \to (0,0,0,2),\\ \mathcal{A}_2 &= \{XY, YZ\} \to (0,0,0,2)\end{aligned}$$

have the same invariants, they are not equivalent under the symmetry group $S_3 \times \mathbb{Z}_2$.

Carrying out this procedure exhaustively for the 127 and 19 subalgebras of $\mathfrak{a}$-type and $\mathfrak{b}$-type gives us 23 and 5 inequivalent Lie algebras, respectively. We denote these subalgebras by $\mathfrak{a}_k$ $(0 \leq k \leq 22)$ and $\mathfrak{b}_l$ $(0 \leq l \leq 4)$. For the full list of invariants, see Table G.1 in the Supplemental Materials. In particular, it turns out that the only case in which the invariants (s, p, e, d) cannot distinguish inequivalent subalgebras is that presented in Example 4.4.7.

By Proposition 4.2.1, we can identify these subalgebras by inspection with direct sums of simple Lie algebras plus a center.

Example 4.4.8. The set

$$\mathcal{A} = \{XX, YY, ZZ, ZY\}$$

generates the Lie algebra

$$\begin{aligned}
&\mathfrak{a}_{20} := \langle \mathcal{A} \rangle_{\text{Lie}} = \text{span}\{XX, YY, ZZ, YZ, ZY, XI, IX\} \\
&= \text{span}\{YY + ZZ, YZ + ZY, XI + IX\} \\
&\oplus \text{span}\{YY - ZZ, YZ - ZY, XI - IX\} \oplus \text{span}\{XX\} \\
&\cong \mathfrak{su}(2) \oplus \mathfrak{su}(2) \oplus \mathfrak{u}(1).
\end{aligned}$$

At this point, we have reduced the number of possible DLAs significantly by taking into account the symmetries of the Pauli group. As a final step, we now generate all $\mathfrak{a}_k(n)$, $\mathfrak{b}_k(n)$, $\mathfrak{a}_k^\circ(n)$ and $\mathfrak{b}_k^\circ(n)$ up to $n = 8$ with Algorithm 3 and Remark 4.3.1. Inspired by this list of Lie algebras we construct formal proofs to determine them for all $n \geq 3$, which is discussed in the next section.

4.5 Results

4.5.1 Main theorem

We state the main theorem of our work below, and tabulate the generators of the Lie algebras of our classification in Table 4.1. For convenience, we recall our definition of dynamical Lie algebras of types $\mathfrak{a}$ and $\mathfrak{b}$. A *DLA of* $\mathfrak{a}$*-type* on an open spin chain of length n is a Lie algebra of the form $\mathfrak{a}(n)$ (cf. Section 4.3.3), where $\mathfrak{a} = \langle \mathcal{A} \rangle_{\text{Lie}}$ for some generating set of 2-site Pauli strings $\mathcal{A} \subseteq \{X, Y, Z\}^{\otimes 2}$. Explicitly, $\mathfrak{a}(n) = \langle \mathcal{A}(n) \rangle_{\text{Lie}}$ where

$$\mathcal{A}(n) = \{A_i B_{i+1} \,|\, AB \in \mathcal{A},\ 1 \leq i \leq n-1\}, \tag{4.16}$$

with AB denoting 2-site Pauli strings. A $\mathfrak{b}$*-type DLA* on an open spin chain of length n is one that cannot be expressed as $\mathfrak{a}$-type and has the form $\mathfrak{b}(n) = \langle \mathcal{A}(n) \cup \mathcal{B}(n) \rangle_{\text{Lie}}$ for some $\mathcal{A} \subseteq \{X, Y, Z\}^{\otimes 2}$, a non-empty set of Paulis $\mathcal{B} \subseteq \{X, Y, Z\}$, and

$$\mathcal{B}(n) = \{B_i \,|\, B \in \mathcal{B},\ 1 \leq i \leq n\}. \tag{4.17}$$

Theorem 4.5.1 (Classification of Open DLAs)**.** *Every dynamical Lie algebra of type* $\mathfrak{a}$ *or* $\mathfrak{b}$ *on an open spin chain of length* $n \geq 3$ *is isomorphic to one of the following:*

$$\mathfrak{a}_0(n) \cong \mathfrak{u}(1)^{\oplus(n-1)},$$

$$
\begin{aligned}
\mathfrak{a}_1(n) &\cong \mathfrak{so}(n),\\
\mathfrak{a}_2(n) &\cong \mathfrak{a}_4(n) \cong \mathfrak{so}(n) \oplus \mathfrak{so}(n),\\
\mathfrak{a}_3(n) &\cong \begin{cases} \mathfrak{so}(2^{n-2})^{\oplus 4}, & n \equiv 0 \bmod 8,\\ \mathfrak{so}(2^{n-1}), & n \equiv \pm 1 \bmod 8,\\ \mathfrak{su}(2^{n-2})^{\oplus 2}, & n \equiv \pm 2 \bmod 8,\\ \mathfrak{sp}(2^{n-2}), & n \equiv \pm 3 \bmod 8,\\ \mathfrak{sp}(2^{n-3})^{\oplus 4}, & n \equiv 4 \bmod 8, \end{cases}\\
\mathfrak{a}_5(n) &\cong \begin{cases} \mathfrak{so}(2^{n-2})^{\oplus 4}, & n \equiv 0 \bmod 6,\\ \mathfrak{so}(2^{n-1}), & n \equiv \pm 1 \bmod 6,\\ \mathfrak{su}(2^{n-2})^{\oplus 2}, & n \equiv \pm 2 \bmod 6,\\ \mathfrak{sp}(2^{n-2}), & n \equiv 3 \bmod 6, \end{cases}\\
\mathfrak{a}_6(n) &\cong \mathfrak{a}_7(n) \cong \mathfrak{a}_{10}(n)\\
&\cong \begin{cases} \mathfrak{su}(2^{n-1}), & n \text{ odd},\\ \mathfrak{su}(2^{n-2})^{\oplus 4}, & n \geq 4 \text{ even}, \end{cases}\\
\mathfrak{a}_8(n) &\cong \mathfrak{so}(2n-1),\\
\mathfrak{a}_9(n) &\cong \mathfrak{sp}(2^{n-2}),\\
\mathfrak{a}_{11}(n) &= \mathfrak{a}_{16}(n) = \mathfrak{so}(2^n), \quad n \geq 4,\\
\mathfrak{a}_k(n) &= \mathfrak{su}(2^n), \quad k = 12, 17, 18, 19, 21, 22, \ n \geq 4,\\
\mathfrak{a}_{13}(n) &= \mathfrak{a}_{20}(n) \cong \mathfrak{a}_{15}(n) \cong \mathfrak{su}(2^{n-1})^{\oplus 2},\\
\mathfrak{a}_{14}(n) &\cong \mathfrak{so}(2n),\\
\mathfrak{b}_0(n) &\cong \mathfrak{u}(1)^{\oplus n},\\
\mathfrak{b}_1(n) &\cong \mathfrak{u}(1)^{\oplus(2n-1)},\\
\mathfrak{b}_2(n) &\cong \mathfrak{sp}(2^{n-2}) \oplus \mathfrak{u}(1),\\
\mathfrak{b}_3(n) &\cong \mathfrak{su}(2)^{\oplus n},\\
\mathfrak{b}_4(n) &\cong \mathfrak{su}(2^{n-1}) \oplus \mathfrak{su}(2^{n-1}) \oplus \mathfrak{u}(1).
\end{aligned}
$$

The following corollary immediately follows from Theorem 4.5.1 and knowledge of the dimensions of $\mathfrak{su}$, $\mathfrak{so}$ and $\mathfrak{sp}$ (see SM Equation (4.10)). As can be seen from Theorem 4.5.2 below, it holds for both open and closed spin chains.

Corollary 4.5.1 (Dimension scaling of DLAs)**.** *The dimensions of all non-trivial dynamical Lie algebras of 2-local spin chains of length n in one dimension scale as either $O(4^n)$, $O(n^2)$ or $O(n)$.*

We thus see that the DLAs can be separated in three classes based on the scaling of their dimensions.

4.5.2 Sketch of the proof

The complete proof of Theorem 4.5.1 is presented in the Supplemental Materials. Here is a brief sketch of the proof; we refer to Appendix G.3.1 for a more detailed outline. We divide the set of Lie algebras $\mathfrak{a}_k(n)$, $\mathfrak{b}_l(n)$ into three classes: linear, quadratic, and exponential, according to the anticipated growth of their dimension. The *linear* class consists of $\mathfrak{a}_0(n)$ and $\mathfrak{b}_l(n)$ with $l = 0, 1, 3$, and their treatment is obvious. The *quadratic* class contains $\mathfrak{a}_k(n)$ with $k = 1, 2, 4, 8, 14$. These Lie algebras are determined by using the frustration graphs of their generators in Appendix G.3.3. For the *exponential* class, we first observe that $\mathfrak{b}_2(n) = \mathfrak{a}_9(n) \oplus \text{span}\{X_1\}$ and $\mathfrak{b}_4(n) = \mathfrak{a}_{15}(n) \oplus \text{span}\{X_1\}$. Next, we identify the cases when $\mathfrak{a}_k(n) = \mathfrak{su}(2^n)$; see Appendix G.2.4 for details. We also find isomorphisms that are obtained by relabeling of the Pauli matrices among some of the algebras (Appendix G.3.2).

The **strategy** in the remaining exponential cases is as follows.

1. For each of our Lie subalgebras $\mathfrak{s} = \mathfrak{a}_k(n) \subseteq \mathfrak{su}(2^n)$, we find its *stabilizer* $\text{St}(\mathfrak{s})$, which is defined as the set of all Pauli strings $\in \mathcal{P}_n$ that commute with every element of $\mathfrak{s}$. This can be done explicitly, because the stabilizer is determined only from the generators of $\mathfrak{s}$ (see Proposition G.3.3).

2. By definition, $\mathfrak{s}$ commutes with all elements of its stabilizer $\text{St}(\mathfrak{s})$; hence, it is contained in the *centralizer* of $\text{St}(\mathfrak{s})$ in $\mathfrak{su}(2^n)$, which we denote $\mathfrak{su}(2^n)^{\text{St}(\mathfrak{s})}$. We can reduce the Lie subalgebra $\mathfrak{su}(2^n)^{\text{St}(\mathfrak{s})}$ further by factoring all elements of the center of $\text{St}(\mathfrak{s})$, which will become central in it, because we have shown that $\mathfrak{s}$ has a trivial center (Lemma G.3.11). This results in a Lie algebra denoted $\mathfrak{g}_k(n)$ when $\mathfrak{s} = \mathfrak{a}_k(n)$.

3. By the above construction, we have $\mathfrak{a}_k(n) \subseteq \mathfrak{g}_k(n)$. In the case of associative algebras, we would get equality due to (a finite-dimensional version of) von Neumann's Double Commutant Theorem (see e.g. [315], Theorem 6.2.5). However, in the Lie case, we might have a strict inclusion. We improve the upper bounds for $\mathfrak{a}_k(n)$ by finding *involutions* θ_k of $\mathfrak{g}_k(n)$ such that all elements of $\mathfrak{a}_k(n)$ are fixed under θ_k. The last condition can be checked only on the generators of $\mathfrak{a}_k(n)$ (see Appendix G.3.5).

4. We prove by induction on n that the upper bound is exact, that is $\mathfrak{a}_k(n) = \mathfrak{g}_k(n)^{\theta_k}$ (see Appendix G.3.6). First we note that both $\mathfrak{a}_k(n)$ and $\mathfrak{g}_k(n)^{\theta_k}$ are linearly spanned by the Pauli strings contained in them. We start with an arbitrary Pauli string

$a \in i\mathcal{P}_n \cap \mathfrak{g}_k(n)^{\theta_k}$ and want to show that it is in $\mathfrak{a}_k(n)$. The main idea is to use suitable commutators of a with elements of $\mathfrak{a}_k(n)$ to produce a Pauli string $b \in i\mathcal{P}_n \cap \mathfrak{g}_k(n)^{\theta_k}$ with I in one of its positions. Erasing the I in b gives an element of $\mathfrak{g}_k(n-1)^{\theta_k}$, which by induction is in $\mathfrak{a}_k(n-1)$.

5. Finally, we identify the Lie algebras $\mathfrak{g}_k(n)^{\theta_k}$ with those from Theorem 4.5.1 (see Appendix G.3.7). This is accomplished by applying in each case a suitable unitary transformation that brings the stabilizer $\mathrm{St}(\mathfrak{s})$ to a more convenient form (cf. Appendix G.1).

4.5.3 Example: $\mathfrak{a}_9(n)$

Consider the example of $\mathfrak{a}_9 = \langle XY, XZ \rangle_{\mathrm{Lie}}$, which produces the subalgebra $\mathfrak{a}_9(n) \subseteq \mathfrak{su}(2^n)$ generated by:

$$X_1Y_2, X_1Z_2, X_2Y_3, X_2Z_3, \ldots, X_{n-1}Y_n, X_{n-1}Z_n. \tag{4.18}$$

Let us sketch the above steps in the strategy of the proof of Theorem 4.5.1 in the case $\mathfrak{s} = \mathfrak{a}_9(n)$.

1. The stabilizer $\mathrm{St}(\mathfrak{s})$ is the set of all Pauli strings $P \in \mathcal{P}_n$ such that $[a, P] = 0$ for every $a \in \mathfrak{s}$. It is enough to check this for all a in the list of generators Equation (4.18), which means that the substring of P in positions $j, j+1$ commutes with XY and XZ for all $1 \leq j \leq n-1$. By inspection, we find $\mathrm{St}(XY, XZ) = \{II, XI, YX, ZX\}$, so these are the only possible such substrings of P. This gives $\mathrm{St}(\mathfrak{s}) = \{I^{\otimes n}, X_1, Y_1X_2, Z_1X_2\}$.

2. The centralizer $\mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}$ is the set of all $a \in \mathfrak{su}(2^n)$ such that $[a, P] = 0$ for every $P \in \mathrm{St}(\mathfrak{s})$; hence it contains $\mathfrak{s}$. As the center of $\mathrm{St}(\mathfrak{s})$ is trivial, we have $\mathfrak{g}_9(n) = \mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}$. To illustrate this last step, we mention that $\mathrm{St}(\mathfrak{a}_{15}(n)) = \{I^{\otimes n}, X_1\}$. In this case, $X_1 \in \mathfrak{su}(2^n)^{X_1}$ is central and we have to quotient by it to obtain $\mathfrak{g}_{15}(n) = \mathfrak{su}(2^n)^{X_1}/\mathrm{span}\{X_1\}$.

3. We saw above that $\mathfrak{s} \subseteq \mathfrak{g}_9(n)$. Now we find an involution θ_9 of $\mathfrak{g}_9(n)$ such that $\mathfrak{s} \subseteq \mathfrak{g}_9(n)^{\theta_9}$, the set of fixed points under θ_9. Since θ_9 respects the Lie brackets, it is enough to check $\theta_9(a) = a$ only for the generators Equation (4.18). Explicitly, we let $\theta_9(a) = -Q_9 a^T Q_9$ where $Q_9 = IYZZ\cdots Z$.

4. We prove by induction on n that $\mathfrak{a}_9(n) = \mathfrak{g}_9(n)^{\theta_9}$. To show that any $a \in i\mathcal{P}_n \cap \mathfrak{g}_9(n)^{\theta_9}$ with $n \geq 4$ is in $\mathfrak{a}_9(n)$, we first take suitable commutators of a with the generators

Equation (4.18) to produce $b \in i\mathcal{P}_n \cap \mathfrak{g}_k(n)^{\theta_k}$ that has I in some position $j \geq 3$. Erasing the I gives an element $c \in \mathfrak{g}_9(n-1)^{\theta_9}$, which by induction is in $\mathfrak{a}_9(n-1)$. Inserting I back in j-th place in c gives that $b \in \mathfrak{a}_9(n)$.

5. As $\mathrm{St}(\mathfrak{s}) \cong \{I^{\otimes n}, X_1, Y_1, Z_1\}$, we can simplify $\mathfrak{g}_9(n)^{\theta_9}$ by applying a unitary transformation $a \mapsto UaU^\dagger$ that takes $\mathrm{St}(\mathfrak{s})$ to $\{I^{\otimes n}, X_1, Y_1, Z_1\}$. Explicitly, we take $U = e^{i\frac{\pi}{4}X_1X_2}$. Then $\mathfrak{g}_9(n) \cong \mathfrak{su}(2^n)^{\{X_1,Y_1,Z_1\}} = I \otimes \mathfrak{su}(2^{n-1}) \cong \mathfrak{su}(2^{n-1})$. The involution $\theta_9(a) = -Q_9 a^T Q_9$ gets transformed to $-\tilde{Q}_9 a^T \tilde{Q}_9$, where $\tilde{Q}_9 = UQ_9U^T$ in this case happens to be $= Q_9$. Restricted to $\mathfrak{su}(2^{n-1})$, this gives the involution $-Qa^TQ$ with $Q = YZZ\cdots Z$, whose fixed points are $\cong \mathfrak{sp}(2^{n-2})$ because $Q^T = -Q$.

We conclude that $\mathfrak{a}_9(n) \cong \mathfrak{sp}(2^{n-2})$.

4.5.4 Periodic boundary conditions

In the periodic case, we define a *DLA of $\mathfrak{a}^\circ$-type* on a closed spin chain of length n as a Lie algebra of the form $\mathfrak{a}^\circ(n) = \langle \mathcal{A}^\circ(n) \rangle_{\mathrm{Lie}}$ for some generating set of 2-site Pauli strings $\mathcal{A} \subseteq \{X, Y, Z\}^{\otimes 2}$, where

$$\mathcal{A}^\circ(n) = \{A_iB_{i+1}, A_nB_1 \,|\, AB \in \mathcal{A},\ 1 \leq i \leq n-1\}. \tag{4.19}$$

A *$\mathfrak{b}^\circ$-type DLA* on a closed spin chain of length n is one that cannot be expressed as $\mathfrak{a}^\circ$-type and has the form $\mathfrak{b}^\circ(n) = \langle \mathcal{A}^\circ(n) \cup \mathcal{B}(n) \rangle_{\mathrm{Lie}}$ for some $\mathcal{A} \subseteq \{X, Y, Z\}^{\otimes 2}$, a non-empty set of Paulis $\mathcal{B} \subseteq \{X, Y, Z\}$, and $\mathcal{B}(n)$ given by Equation (4.17).

Theorem 4.5.2 (Classification of Periodic DLAs)**.** *Every dynamical Lie algebra of type $\mathfrak{a}^\circ$ or $\mathfrak{b}^\circ$ on a closed spin chain of length $n \geq 3$ is isomorphic to one of the following:*

$$\begin{aligned}
\mathfrak{a}_0^\circ(n) &\cong \mathfrak{u}(1)^{\oplus n},\\
\mathfrak{a}_1^\circ(n) &\cong \mathfrak{so}(n)^{\oplus 2},\\
\mathfrak{a}_2^\circ(n) &\cong \mathfrak{so}(n)^{\oplus 4},\\
\mathfrak{a}_3^\circ(n) &= \begin{cases} \mathfrak{a}_{13}(n), & n \text{ odd},\\ \mathfrak{a}_3(n), & n \equiv 0 \mod 4,\\ \mathfrak{a}_6(n), & n \equiv 2 \mod 4, \end{cases}\\
&\cong \begin{cases} \mathfrak{su}(2^{n-1})^{\oplus 2}, & n \text{ odd},\\ \mathfrak{so}(2^{n-2})^{\oplus 4}, & n \equiv 0 \mod 8,\\ \mathfrak{sp}(2^{n-3})^{\oplus 4}, & n \equiv 4 \mod 8,\\ \mathfrak{su}(2^{n-2})^{\oplus 4}, & n \equiv 2 \mod 4, \end{cases}
\end{aligned}$$

$$
\begin{aligned}
\mathfrak{a}_4^\circ(n) &\cong \begin{cases} \mathfrak{so}(2n), & n \text{ odd}, \\ \mathfrak{so}(n)^{\oplus 4}, & n \text{ even}, \end{cases} \\
\mathfrak{a}_5^\circ(n) &= \begin{cases} \mathfrak{a}_{16}(n), & n \equiv \pm 1 \mod 3, \\ \mathfrak{a}_5(n), & n \equiv 0 \mod 3, \end{cases} \\
&\cong \begin{cases} \mathfrak{so}(2^n), & n \equiv \pm 1 \mod 3, \\ \mathfrak{so}(2^{n-2})^{\oplus 4}, & n \equiv 0 \mod 6, \\ \mathfrak{sp}(2^{n-2}), & n \equiv 3 \mod 6, \end{cases} \\
\mathfrak{a}_6^\circ(n) &= \begin{cases} \mathfrak{a}_{13}(n) \cong \mathfrak{su}(2^{n-1})^{\oplus 2}, & n \text{ odd}, \\ \mathfrak{a}_6(n) \cong \mathfrak{su}(2^{n-2})^{\oplus 4}, & n \text{ even}, \end{cases} \\
\mathfrak{a}_k^\circ(n) &= \mathfrak{a}_k(n), \quad k = 7, 13, 16, 20, \\
\mathfrak{a}_8^\circ(n) &\cong \mathfrak{so}(2n)^{\oplus 2}, \\
\mathfrak{a}_9^\circ(n) &\cong \mathfrak{so}(2^n), \quad n \geq 4, \\
\mathfrak{a}_{10}^\circ(n) &= \begin{cases} \mathfrak{su}(2^n), & n \equiv \pm 1 \mod 3, \\ \mathfrak{a}_{10}(n), & n \equiv 0 \mod 3, \end{cases} \\
&\cong \begin{cases} \mathfrak{su}(2^n), & n \equiv \pm 1 \mod 3, \\ \mathfrak{su}(2^{n-2})^{\oplus 4}, & n \equiv 0 \mod 6, \\ \mathfrak{su}(2^{n-1}), & n \equiv 3 \mod 6, \end{cases} \\
\mathfrak{a}_{11}^\circ(n) &= \mathfrak{so}(2^n), \quad n \geq 4, \\
\mathfrak{a}_k^\circ(n) &= \mathfrak{su}(2^n), \quad k = 12, 15, 17, 18, 19, 21, 22, \\
\mathfrak{a}_{14}^\circ(n) &\cong \mathfrak{so}(2n)^{\oplus 2}, \\
\mathfrak{b}_0^\circ(n) &= \mathfrak{b}_0(n) \cong \mathfrak{u}(1)^{\oplus n}, \\
\mathfrak{b}_1^\circ(n) &\cong \mathfrak{u}(1)^{\oplus 2n}, \\
\mathfrak{b}_2^\circ(n) &\cong \mathfrak{so}(2^n), \quad n \geq 4, \\
\mathfrak{b}_3^\circ(n) &= \mathfrak{b}_3(n) \cong \mathfrak{su}(2)^{\oplus n}, \\
\mathfrak{b}_4^\circ(n) &= \mathfrak{su}(2^n).
\end{aligned}
$$

The proof of this theorem is given in Appendix G.3.8. The proof strategy is different from that of Theorem 4.5.1 because, unlike the open case, the periodic Lie algebras $\mathfrak{a}_k^\circ(n)$ are not generated inductively from $\mathfrak{a}_k^\circ(n-1)$. Instead, we use that $\mathfrak{a}_k^\circ(n)$ is generated from $\mathfrak{a}_k(n)$ and its cyclic shifts, and we utilize the explicit description $\mathfrak{a}_k(n) = \mathfrak{g}_k(n)^{\theta_k}$ (see Part 4. in Section 4.5.2, and for more details Appendix G.3.6).

4.5.5 Permutation-invariant subalgebras

The strategies employed for periodic boundary conditions can also be used to classify the DLAs in the case when the Hamiltonian is defined on a complete graph. For this, we have to adapt the definition of Equation (4.14) so that each spin is connected to each other spin via 2-local interactions given by Pauli strings. In more detail, we define a *DLA of $\mathfrak{a}^\pi$-type* as a Lie algebra of the form $\mathfrak{a}^\pi(n) = \langle \mathcal{A}^\pi(n) \rangle_{\text{Lie}}$ for some generating set of 2-site Pauli strings $\mathcal{A} \subseteq \{X, Y, Z\}^{\otimes 2}$, where

$$\mathcal{A}^\pi(n) = \{A_i B_j \,|\, AB \in \mathcal{A},\ 1 \leq i \neq j \leq n\}. \tag{4.20}$$

A *$\mathfrak{b}^\pi$-type DLA* is one that cannot be expressed as $\mathfrak{a}^\pi$-type and has the form $\mathfrak{b}^\pi(n) = \langle \mathcal{A}^\pi(n) \cup \mathcal{B}(n) \rangle_{\text{Lie}}$ for some $\mathcal{A} \subseteq \{X, Y, Z\}^{\otimes 2}$, a non-empty set of Paulis $\mathcal{B} \subseteq \{X, Y, Z\}$, and $\mathcal{B}(n)$ given by Equation (4.17).

Theorem 4.5.3 (Classification of Permutation Invariant DLAs)**.** *For $n \geq 3$, every dynamical Lie algebra of type $\mathfrak{a}^\pi$ or $\mathfrak{b}^\pi$ is isomorphic to one of the following:*

$$\begin{aligned}
&\mathfrak{a}_k^\pi(n) = \mathfrak{a}_k(n), \qquad k = 7, 16, 20, 22,\\
&\mathfrak{a}_0^\pi(n) \cong \mathfrak{u}(1)^{\oplus n(n-1)/2},\\
&\mathfrak{a}_2^\pi(n) = \mathfrak{so}(2^n)^{Z\cdots Z} \cong \mathfrak{so}(2^{n-1})^{\oplus 2},\\
&\mathfrak{a}_4^\pi(n) = \mathfrak{a}_7(n) \cong \begin{cases} \mathfrak{su}(2^{n-1}), & n \text{ odd},\\ \mathfrak{su}(2^{n-2})^{\oplus 4}, & n \geq 4 \text{ even}, \end{cases}\\
&\mathfrak{a}_6^\pi(n) = \mathfrak{a}_{20}(n) \cong \mathfrak{a}_{14}^\pi(n) \cong \mathfrak{su}(2^{n-1})^{\oplus 2},\\
&\mathfrak{b}_0^\pi(n) = \mathfrak{b}_0(n) \cong \mathfrak{u}(1)^{\oplus n},\\
&\mathfrak{b}_1^\pi(n) \cong \mathfrak{u}(1)^{\oplus n(n+1)/2},\\
&\mathfrak{b}_3^\pi(n) = \mathfrak{b}_3(n) \cong \mathfrak{su}(2)^{\oplus n}.
\end{aligned}$$

The proof of this theorem is given in Appendix G.3.9.

4.6 Discussion

In the previous section, we have classified the dynamical Lie algebras generated by the Pauli terms of 2-local spin chain Hamiltonians, both for open and closed boundary conditions and on complete graphs. In this section, we discuss the importance of this classification for various fields in physics.

4.6.1 Relevance for variational quantum computing

As we saw in Equation (2.3), a quantum circuit can be described as a product of unitaries $U = \overleftarrow{\prod_l^L} U_l$. We can write $U_l = e^{a_l}$ where a_l is a 1- or 2-local operator. For a set of generators $\mathcal{A} = \{a_l\}$ with a corresponding DLA $\mathfrak{a} = \langle \mathcal{A} \rangle_{\text{Lie}}$, we have that

$$e^{\mathfrak{a}} = \left\{ e^{a_{l_1} t_1} e^{a_{l_2} t_2} \cdots e^{a_{l_r} t_r} \,\middle|\, t_i \in \mathbb{R},\ a_{l_i} \in \mathcal{A} \right\}. \tag{4.21}$$

In other words, any element in the Lie group $e^{\mathfrak{a}}$ generated by the DLA can be reached by a finite product of unitaries in that group (see [27], Corollary 3.2.6). In quantum computing, if $e^{\mathfrak{a}} = \mathrm{SU}(2^n)$, then the gate set $\{e^{a_k}\}$ is called *universal* [7]. It is known that almost any combination of unitaries is universal [316, 317]. However, we can make specific choices for the generators $\{a_l\}$ that correspond to a non-universal gate set, which instead will generate a proper subgroup of $\mathrm{SU}(2^n)$. This is especially relevant for a class of quantum algorithms called variational quantum algorithms [24, 25].

If limited to 1-dimensional topology, the generators in our classification will produce a circuit that is an element of the Lie group $e^{\mathfrak{a}}$. This notion can be used to construct specific quantum algorithms that always act within a subgroup of $\mathrm{SU}(2^n)$. For example, if we consider a variational circuit of the form

$$U(\boldsymbol{\theta}) = U_1(\theta_1) U_2(\theta_2) \cdots U_L(\theta_L),$$

with

$$U_k(\theta_k) = e^{\theta_k a_k}. \tag{4.22}$$

The gate parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_L)$ are real parameters that we can optimize with VQE (see Section 2.1.1). We restate the cost function for clarity,

$$E(\boldsymbol{\theta}) = \mathrm{Tr}\left\{ U(\boldsymbol{\theta}) \rho_0 U^\dagger(\boldsymbol{\theta}) H_c \right\}, \tag{4.23}$$

where H_c is a Hermitian operator and $\rho_0 = |\psi_0\rangle\langle\psi_0|$ is the initial state of the system. Crucial to the success of these algorithms is the choice of a circuit ansatz $U(\boldsymbol{\theta})$ and the properties of the cost function Equation (4.23).

VQE ansätze

A large class of variational circuits consist of L repeating layers of unitary blocks [32, 49, 39, 40, 318, 228, 319, 320, 321] and we have seen these used in Chapter 2. In this section, we will give some examples of these circuits and how our classification relates to them.

Example 4.6.1. *Hamiltonian Variational Ansatz.* The Hamiltonian Variational Ansatz circuit is obtained by Trotterizing the exponential of a Hamiltonian [39, 40]. Consider the Hamiltonian $H_{XY} = \sum_{i=1}^{n-1} X_i Y_{i+1}$, which has $\mathfrak{a}_1(n)$ as its DLA. Exponentiation of H and the application of the Trotter–Suzuki formula then gives:

$$U(\boldsymbol{\theta}) = \prod_{l=1}^{\overleftarrow{L}} \left(\prod_{\text{even } k}^{\longleftarrow} e^{i\theta_k^{(l)} X_k Y_{k+1}} \prod_{\text{odd } k}^{\longleftarrow} e^{i\theta_k^{(l)} X_k Y_{k+1}} \right),$$

where we grouped the odd and even terms together due to the structure imposed by the 1- and 2-qubit gates available on the quantum computer. Due to Equation (4.21) and the knowledge that $\mathfrak{a}_1(n) \cong \mathfrak{so}(n)$, we know that the above circuit must be a parameterization of a unitary operator $U(\boldsymbol{\theta}) \in \mathrm{SO}(n)$.

Similarly, we can take the DLA $\mathfrak{a}_9(n)$ with generators $\{XY, XZ\}$, which gives a circuit within $\mathrm{Sp}(2^{n-2})$:

$$U(\boldsymbol{\theta}, \boldsymbol{\phi}) = \prod_{l=1}^{\overleftarrow{L}} \left(\prod_{\text{even } k}^{\longleftarrow} e^{i\theta_k^{(l)} X_k Y_{k+1}} \prod_{\text{odd } k}^{\longleftarrow} e^{i\theta_k^{(l)} X_k Y_{k+1}} \prod_{\text{even } k}^{\longleftarrow} e^{i\phi_k^{(l)} X_k Z_{k+1}} \prod_{\text{odd } k}^{\longleftarrow} e^{i\phi_k^{(l)} X_k Z_{k+1}} \right).$$

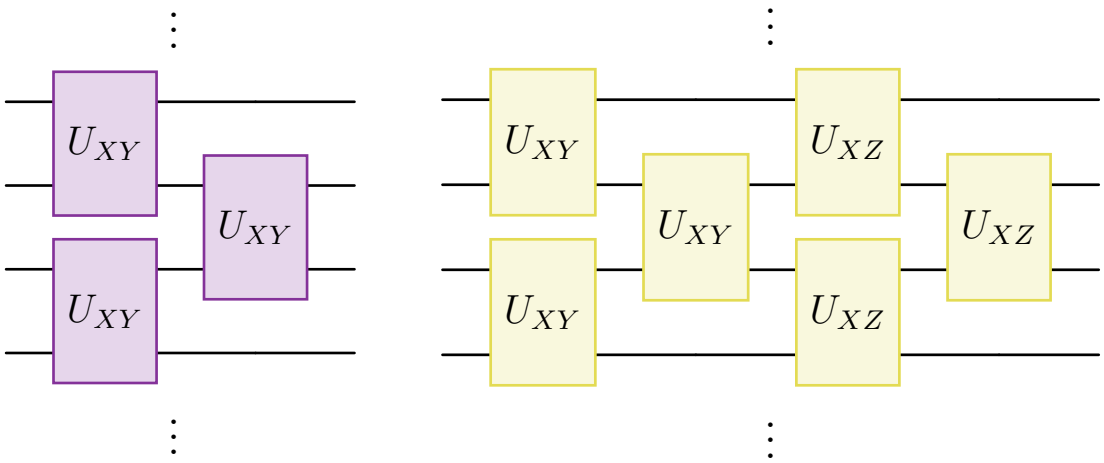
We illustrate these circuits schematically in Figure 4.4.

Example 4.6.2. *ADAPT-VQE.* In ADAPT-VQE, one dynamically grows the circuit using a predetermined operator pool, so that each gate lowers the cost function by the largest amount [113]. This class of dynamical circuit ansätze can be understood as a Riemannian gradient flow over a specific subgroup [4]. This heuristic is popular in quantum chemistry for circuit design, where specific operator pools are considered that are tailored to fermionic Hamiltonians [324, 223, 221]. The operator pool can be seen as a set of generators, with a corresponding DLA. In the context of our classification, we can thus determine the resulting subgroup of the dynamically grown circuit ansatz based on the generators in the operator pool.

Example 4.6.3. *Permutation-invariant circuits.* Instead of a 1-dimensional topology, one can consider a Hamiltonian with a fully connected topology (see Figure 4.6):

$$H = \sum_{1 \le i \ne j \le n} A_i B_j.$$

This topology is common in ion trap quantum computers [325] and also shows up in the context of quantum Boltzmann machines [326, 327], which are the quantum equivalent



(a) $\mathfrak{a}_1(n)$ (b) $\mathfrak{a}_9(n)$

Figure 4.4: **Examples of brick layer circuits that fall within our classification.** (a) Hamiltonian Variational Ansatz circuit for the Hamiltonian $H_{XY} = \sum_{i=1}^{n-1} X_i Y_{i+1}$, which parameterizes an element of the group $\mathrm{SO}(n)$. (b) Variational ansatz that parameterizes a unitary in $\mathrm{Sp}(2^{n-2})$ via products of unitaries generated by terms in $\mathfrak{a}_9(n)$. We note that these types of brick-layer circuits often show up in the tensor network literature on quantum compilation [322, 323].

of the Sherrington–Kirkpatrick model with tunable parameters [328]. Closely related are the so-called permutation-equivariant circuits, which consist of parameterized blocks of unitaries that are permutation invariant [329]. These circuit ansätze were shown to be powerful quantum machine learning models for permutation-invariant data sets. Our classification of permutation-invariant 2-site Hamiltonians in one dimension thus provides a classification of DLAs for these types of ansätze.

Barren plateaus

A hurdle in minimizing a cost function of the form Equation (4.23) are so-called *barren plateaus* we explored in Chapter 2, which are flat areas in the cost landscape of a variational quantum algorithm. When barren plateaus are present, the variance of gradients with respect to the gate parameters will decay, on average, exponentially as a function of system size. Hence, obtaining accurate estimates quickly becomes intractable due to the large number of shots required. There is a variety of different setups in which barren plateaus

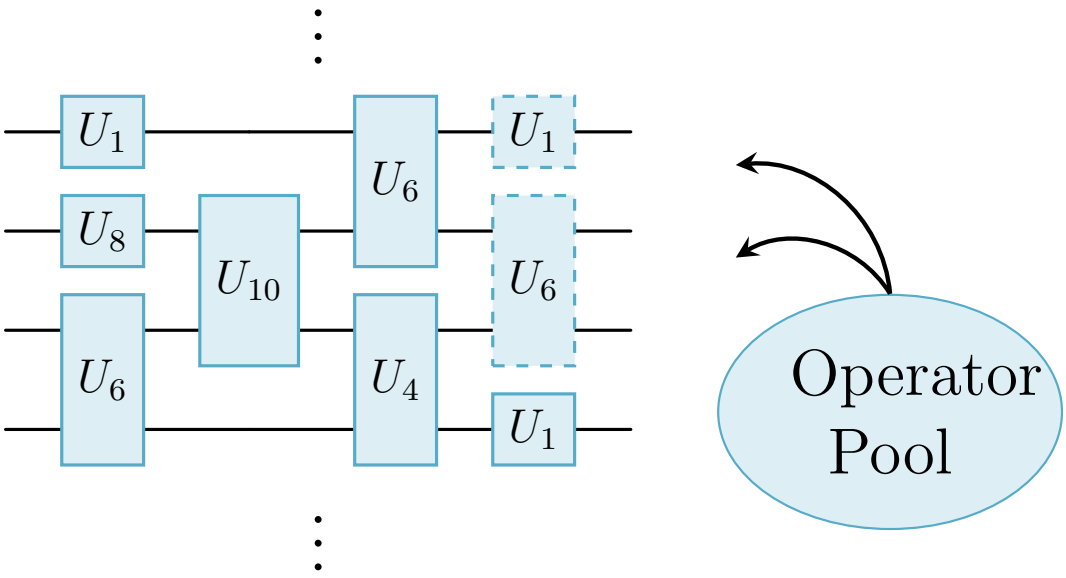


Figure 4.5: **ADAPT-VQE circuit growing heuristic.** We consider a generator pool in our classification and dynamically grow the circuit.

occur [230, 35, 38, 107, 83, 109, 330]. To mitigate this problem, several recent works are aimed at finding ways to avoid the regions where optimization is hard [108, 37, 68, 122, 123, 124, 2].

The relevance of our classification for barren plateaus stems from the conjecture of [227], which states that the variance of the gradients of gate parameters is inversely proportional to the dimension of the DLA $\mathfrak{g}$ of the circuit:

$$\mathrm{Var}[\partial_k E(\boldsymbol{\theta})] \propto \frac{1}{\dim \mathfrak{g}}.$$

There are some subtleties involved in this conjecture, such as the locality of the cost function and the choice of initial state, which are discussed in [227, 331]. In the common case where $H_c \in i\mathfrak{g}$, an exact formula for the variance was obtained independently in Refs. [280, 331], which in particular refines and proves the above conjecture. This formula was interpreted in Ref. [331] in terms of the $\mathfrak{g}$-*purity* [332, 333]

$$P_{\mathfrak{g}}(A) = \mathrm{Tr}\left\{(A|_{\mathfrak{g}})^2\right\}$$

where $A|_{\mathfrak{g}}$ denotes the projection of a linear operator A onto the Lie algebra $\mathfrak{g}$. The $\mathfrak{g}$-purity

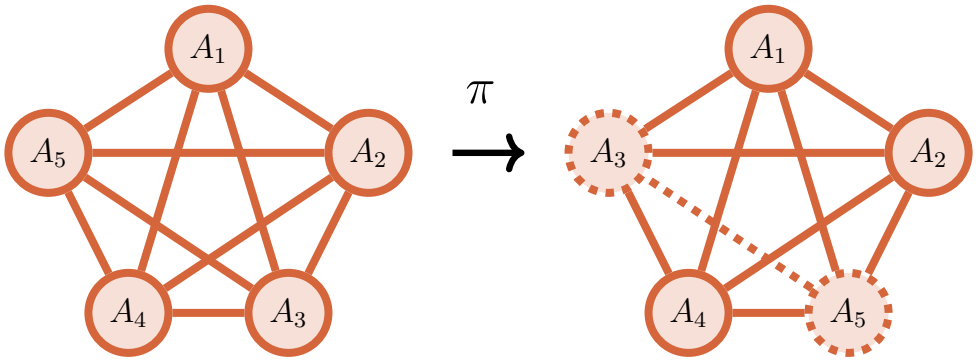
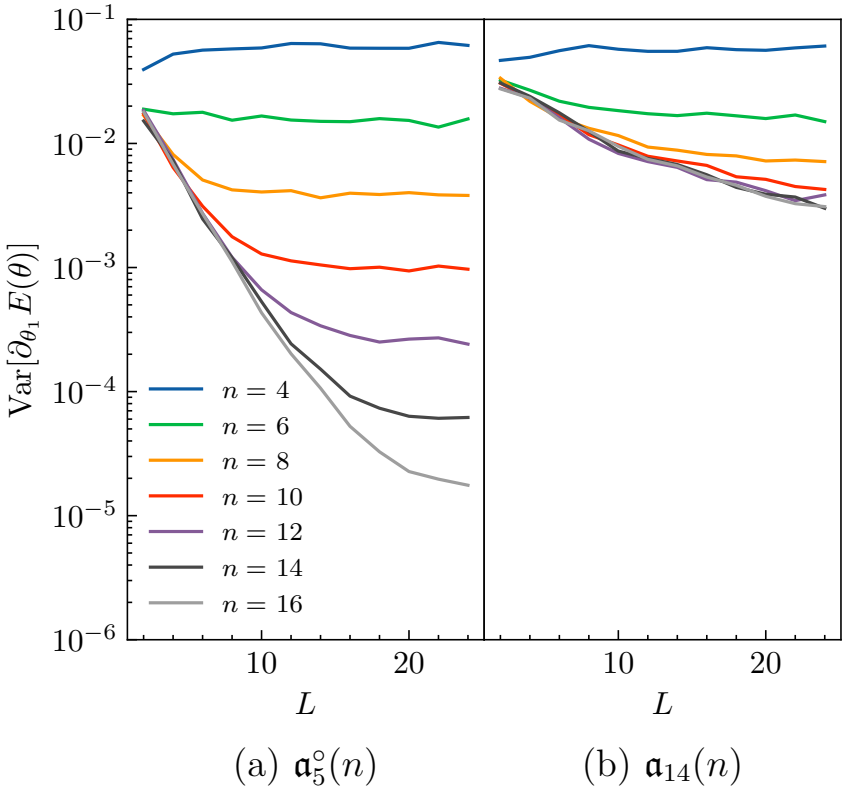


Figure 4.6: **Permutation-invariant topology.** Permuting sites leaves the Hamiltonian invariant.

of the initial state ρ_0 and the observable H_c are intricately connected to barren plateaus, underscoring again the crucial role of the DLA.

As an illustration, we compare the barren plateau behavior for two of our Lie algebras, $\mathfrak{a}_5^\circ(n)$ and $\mathfrak{a}_{14}(n)$, whose dimensions scale exponentially and polynomially with n, respectively. We consider the cost function Equation (4.23) with $H_c = Z_1 Z_2$ and $\rho_0 = |0\rangle\langle 0|^{\otimes n}$. The circuit ansatz $U(\boldsymbol{\theta})$ consists of unitaries generated by generators in our classification. The parameters of the circuit are then uniformly sampled between 0 and π. To observe the barren plateau effect, we take the derivative of the cost function with respect to the first parameter in the first layer of the circuit, $\theta_1^{(1)}$. In Figure 4.7, we then observe the expected gradient decay as a function of the system size for an exponentially scaling DLA and a polynomially scaling DLA. In particular, in Figure 4.7(a) we consider the circuit generated by $H = \sum_{i=1}^{n-1}(X_iY_{i+1} + Y_iZ_{i+1}) + X_nY_1 + Y_nZ_1$ with periodic boundary conditions, whose DLA $\mathfrak{a}_5^\circ(n)$ is isomorphic to $\mathfrak{so}(2^n)$, $\mathfrak{so}(2^{n-2})^{\oplus 4}$ or $\mathfrak{sp}(2^{n-2})$ depending on n (see Theorem 4.5.2). Since $\dim \mathfrak{a}_5^\circ(n) = O(4^n)$, we expect the gradients to decay exponentially. Similarly, in Figure 4.7(b) we consider the circuit generated by $\mathfrak{a}_{14}(n) \cong \mathfrak{so}(2n)$, which is described by the Hamiltonian $H = \sum_{i=1}^{n-1}(X_iX_{i+1} + Y_iY_{i+1} + X_iY_{i+1})$. Here, we have $\dim \mathfrak{a}_{14}(n) = O(n^2)$; hence, we expect the decay of gradients to be polynomial with respect to the system size.

According to Corollary 4.5.1, the only circuits free from barren plateaus generated by Hamiltonians in our classification, which are not composed of only 1-qubit gates, have to be composed of $\mathfrak{so}(n)$-type, since these are the only polynomially scaling DLAs in our classification.

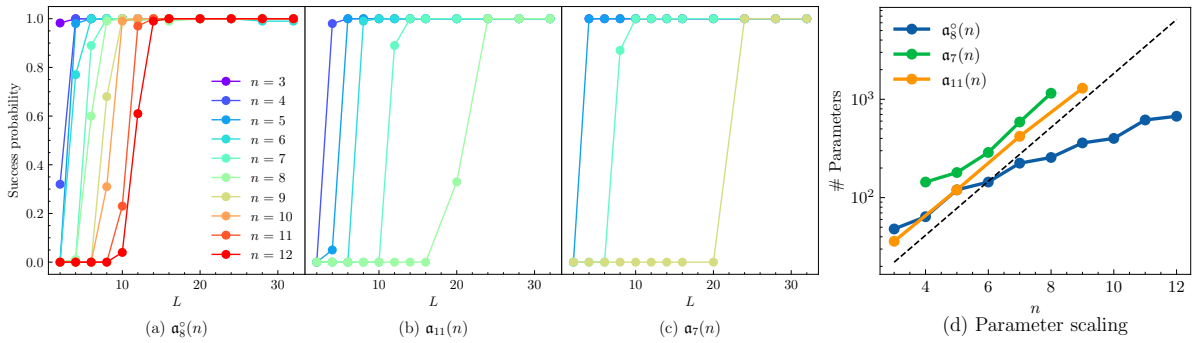


(a) $\mathfrak{a}_5^\circ(n)$ (b) $\mathfrak{a}_{14}(n)$

Figure 4.7: **Barren plateaus in variational quantum circuits.** We calculate the variance of 1000 randomly initialized circuits with $\boldsymbol{\theta}$ sampled uniformly in $[0, \pi]$. (a) The Lie algebra $\mathfrak{a}_5^\circ(n)$ is isomorphic to $\mathfrak{so}(2^n)$, $\mathfrak{so}(2^{n-2})^{\oplus 4}$ or $\mathfrak{sp}(2^{n-2})$ depending on n; hence we expect exponentially decaying gradients for all n. This is confirmed in the figure above, since for a linear increase in n, we see an order of magnitude decrease in the gradient variances. (b) Since $\mathfrak{a}_{14}(n) \cong \mathfrak{so}(2n)$, we find polynomially decaying gradients as a function of system size.

Overparameterization

Similar to the barren plateaus, we can place our classification in the context of the overparameterization phenomenon we discovered in Section 2.2.5. Recent works that have made progress in theoretically understanding this effect in quantum circuits can be connected to the DLA generated by the circuit ansätze used [230, 231]. In particular, in [230], the dimension of the DLA can be used to analyze the Hessian around the global minimum of a typical variational quantum eigensolver cost function [29]. Additionally, the authors



(a) $\mathfrak{a}_8^\circ(n)$ (b) $\mathfrak{a}_{11}(n)$ (c) $\mathfrak{a}_7(n)$ (d) Parameter scaling

Figure 4.8: **Overparameterization of variational quantum circuits.** We plot the success probability of reaching a state with $|E_0 - E(\boldsymbol{\theta})| < 5 \times 10^{-4}$ as a function of the circuit depth L, where E_0 is the lowest energy of the cost Hamiltonian H_c. These results were obtained by averaging 100 random instances for each L and N. The error bars indicate one standard deviation σ. A single random instance consists of 3000 optimization steps of the Adam optimizer [334] with learning rate $\eta = 10^{-2}$. A handful of instances converge to solutions that are further than 5σ from the mean and these outliers are therefore not included in the final plot. (a) The TFIM on a ring has DLA $\mathfrak{a}_8^\circ(n) \cong \mathfrak{so}(2n)^{\oplus 2}$, whose dimension scales quadratically. We see that for a moderate circuit depth, the probability of success goes to 1. (b) Since the DLA $\mathfrak{a}_{11}(n) \cong \mathfrak{so}(2^n)$, we expect that overparameterization occurs at depths that are exponential in the system size. Although this is not immediately clear in here, we see in (d) that the number of parameters indeed scales exponentially in n. (c) For the Heisenberg chain, which has an exponentially-scaling DLA $\mathfrak{a}_7(n)$, the choice of initial state $\rho_0 = |0\rangle\langle 0|^{\otimes n}$ prevents overparameterization from occurring for odd n. (d) If we set the threshold for overparameterization to be a success probability of 0.99, we can plot the required number of parameters to reach this threshold. We see that $\mathfrak{a}_{11}(n)$ and $\mathfrak{a}_7(n)$ require an exponentially scaling number of parameters, whereas $\mathfrak{a}_8^\circ(n)$ only requires a polynomial number. The dashed line is a guiding line that indicates $O(4^n)$ scaling.

find that the critical number of parameters needed to overparameterize a variational quantum circuit can be directly linked to the dimension of the associated DLA. In [231], the authors study the optimization dynamics of overparameterized quantum circuit as perturbations of Riemannian gradient flows [185]. The size of the DLA (defined as the effective dimension in [231]) allows one to bound the number of parameters required to reach the overparameterization regime.

Corollary 4.5.1 tells us that for quantum circuits constructed from the generators of 1-dimensional spin chains, there are only DLAs whose dimension scales as $O(n)$, $O(n^2)$ and $O(4^n)$. Consequently, the linearly and quadratically scaling DLAs are expected to overparameterize with a non-exponential number of parameters. Additionally, the quadratically scaling DLAs in our classification correspond to free fermion models, whose dynamics can be simulated efficiently if ρ_0 is an eigenstate of H_c. However, choosing ρ_0 to be an arbitrary quantum state will still be intractable to simulate classically.

As discussed in [231], a requirement for overparameterization is that the initial state has non-vanishing overlap with the ground state. Similarly, in [335], it is shown that choosing the initial state in the right symmetry sector is crucial for the quality of the optimization. We highlight this importance in one of the numerical examples, where we choose an initial state that prevents overparameterization from occurring for an odd number of sites.

In Figure 4.8, we illustrate the overparameterization phenomenon for three examples in our classification. In particular, in Figure 4.8(a), we consider the TFIM on a ring, which is given by the Hamiltonian $H_c = \sum_{i=1}^{n}(Z_iZ_{i+1} + X_i)$ where $Z_{n+1} := Z_1$. The corresponding DLA is given by $\mathfrak{a}_8^{\circ}(n) \cong \mathfrak{so}(2n)^{\oplus 2}$, whose dimension scales quadratically in n. We take the Hamiltonian Variational Ansatz of H_c on even and odd qubits as a circuit ansatz, and take the initial state to be $\rho_0 = |+\rangle\langle+|^{\otimes n}$. We observe that the cost landscape quickly becomes favorable, resulting in almost guaranteed convergence to the lowest energy state.

In Figure 4.8(b), we take the DLA $\mathfrak{a}_{11}(n) \cong \mathfrak{so}(2^n)$, and a Hamiltonian $H_c \in \mathfrak{so}(2^n)$ given by a random orthogonal $2^n \times 2^n$ matrix. The circuit consists of unitaries generated by the generators of $\mathfrak{a}_{11}$ on even and odd qubits, and we take $\rho_0 = |0\rangle\langle 0|^{\otimes n}$. It now takes much deeper circuits to reach the same success probabilities as in Figure 4.8(a), which is due to the exponential scaling of the DLA.

Finally, in Figure 4.8(c), we consider $\mathfrak{a}_7(n)$, which corresponds to the Heisenberg chain with $H_c = \sum_{i=1}^{n-1}(X_iX_{i+1} + Y_iY_{i+1} + Z_iZ_{i+1})$. The circuit is again the Hamiltonian Variational Ansatz of H_c, and $\rho_0 = |0\rangle\langle 0|^{\otimes n}$. This choice of an initial state only works for an odd number n of qubits, while it fails to produce the overparameterization phenomenon for even n, leading to a success probability of 0 (not plotted). Instead, for even n, the optimization of deep circuits gets stuck in a local minimum. We still observe the exponential

scaling of the number of parameters, in accordance with the scaling of the dimension of $\mathfrak{a}_7(n)$, which is $O(4^n)$.

4.6.2 Relevance for quantum control

In quantum control, one is interested in performing a specific unitary evolution by controlling individual terms in a Hamiltonian acting on a small number of qubits. In order to create any unitary, one requires the system to be controllable, which corresponds to the DLA being equal to $\mathfrak{su}(2^n)$. The conditions for complete controllability are known [220], and it is in principle easy to create a controllable system since any real simple Lie algebra can be generated from 2 elements [336]. If the DLA is a proper subalgebra of $\mathfrak{su}(2^n)$, we say that the system is *uncontrollable*. Note that this includes simple Lie algebras like $\mathfrak{so}$ and $\mathfrak{sp}$ [310]. Uncontrollable systems can arise when there are conserved quantities or symmetries in the physical system one is trying to control. Note that, due to Proposition 4.2.1, the DLA must split into a direct sum of simple Lie algebras and a center. If this decomposition has the form $\mathfrak{su}(N_1) \oplus \cdots \oplus \mathfrak{su}(N_k)$ where $N_1 + \cdots + N_k = 2^n$, then we say that the system is *subspace controllable* [27, Section 4.3.3].

We can contextualize our classification in terms of these definitions. For example, we know that $\mathfrak{a}_{12}(n)$ will produce a controllable quantum system for $n \geq 4$ since this DLA is equal to $\mathfrak{su}(2^n)$. Similarly, since $\mathfrak{a}_1(n) \cong \mathfrak{so}(n)$, we know that it is uncontrollable. Finally, there are many examples of uncontrollable systems that consist of direct sums of $\mathfrak{su}$ blocks and are subspace controllable. For instance, $\mathfrak{a}_3^\circ(n)$ for odd n produces a DLA of the form $\mathfrak{su}(2^{n-1})^{\oplus 2}$. In addition to the notion of controllability of spin systems, we can ask what other types of systems we can simulate with our spin chains, e.g., fermionic or bosonic systems. This question was originally explored for Hamiltonians on cubic lattices with translation symmetry [337, 338]. In particular, the dynamics of quadratic fermionic Hamiltonians is described by DLAs of the $\mathfrak{so}(2n)$ or $\mathfrak{so}(2n+1)$ type, which show up in our classification as $\mathfrak{a}_{14}(n)$ and $\mathfrak{a}_8(n)$. Similarly, the dynamics of a bosonic quadratic Hamiltonian with n modes is related to a symplectic DLA [283], which we can identify with $\mathfrak{a}_5(n)$ for $n \equiv 3 \bmod 6$.

4.6.3 Relevance for spin systems

Our classification of Lie algebras arising in one dimension has bearing on other areas of physics and quantum simulation. The most direct connection is that we have established a set of models, some of which are traditional spin models [339, 340] studied in physics,

while others are new (cf. Table G.2). The integrability [307, 53, 308], dynamical Lie algebra, and symmetry of 1-dimensional spin systems remains an active area of research, and our result provides a database of models where desired properties can be selected or different hypotheses tested.

The various properties of physical systems typically arise from the exact Hamiltonian. As a straightforward example, we can consider the transverse field Ising model; whether this model is in the paramagnetic or ferromagnetic phase depends on the relative strengths of the nearest neighbor coupling and applied magnetic field. Here, we have neglected these parameters in the Hamiltonian, and solely focused on the presence (or absence) of terms in the Hamiltonian. This coarsening of the problem hides much of the critical detail. However, here is still a remarkable amount of information that can be gleaned from this admittedly coarse view.

One example of this is the observation that Hamiltonians with polynomially-scaling dynamical Lie algebras belong to a special class of integrable models which can be simulated efficiently with Lie algebra-based algorithms [333, 341]. Additionally, the dynamics of such models can be simulated in less time than the duration of the desired evolution through a process called fast-forwarding [342]. These algorithms have recently been rediscovered for the purposes of simulating specific classes of quantum circuits [281, 280]. Interestingly, the number of polynomially-scaling algebras in our classification is relatively small ($\mathfrak{a}_1, \mathfrak{a}_2, \mathfrak{a}_4, \mathfrak{a}_8, \mathfrak{a}_{14}, \mathfrak{a}_1^\circ, \mathfrak{a}_2^\circ, \mathfrak{a}_4^\circ, \mathfrak{a}_8^\circ, \mathfrak{a}_{14}^\circ$), and they are all of the $\mathfrak{so}$ type. For example, the TFIM model, which is known to be integrable, arises in our classification as $\mathfrak{a}_7(n)$, which is polynomial in size.

The polynomially scaling algebras in principle come with a "maximal set of independently commuting quantum operators" [343], which enables the integration in the first place. Unfortunately, our method does not capture these because the conserved quantities are not single Pauli strings. However, global symmetries are preserved for some of the models; these include $\mathbb{Z}_2$ (spin flip), SU(2) (global spin rotation) and U(1) (global phase rotation).

One particular property of note is the presence of *non-commuting charges* which describe *non-Abelian symmetries* — that is, elements of the stabilizer that do not commute. These are found in $\mathfrak{a}_8(n), \mathfrak{a}_9(n)$ for all n, and in $\mathfrak{a}_2(n)-\mathfrak{a}_7(n), \mathfrak{a}_{10}(n)$ for odd n only (see Appendix G.3.4). We emphasize that the non-commuting charges we find are intensive, as opposed to extensive. The latter consist of a sum of terms that grows with the system size, which can be related to a wide range of quantum effects in thermodynamics (see Ref. [344] for a review). Extensive non-commuting charges have been studied in the context of bond algebras to understand thermalization and quantum many body scars [345, 298],

hence our classification could potentially be used in this context. Notably, the presence of non-commuting charges complicates questions regarding thermalization. Depending on the context, they either help thermalization (e.g. by increasing entanglement entropy [346]) or hinder it (e.g. by invalidating the Eigenstate Thermalization Hypothesis [347]). Perhaps more interestingly within the context of quantum computing, non-commuting charges couple the dynamics between different irreducible representations of the charges, severely limiting the unitaries that can be implemented [348]. The DLA can also be used in the path integral formulation of quantum mechanics to study many-body systems [349].

A final point is the appearance of symplectic Lie algebras, which are not as common as the orthogonal or unitary types. Here they appear from an AIII Cartan decomposition of a larger Lie algebra; in applications, they come up in the preparation of bosonic quantum states [338], photonics [350] and Clifford circuits/error correction [351].

4.7 Conclusion

We have provided a classification of the dynamical Lie algebras of 2-local spin systems on a linear, circular or permutation invariant topology, and have discussed the relevance of this result in a variety of contexts. We have discovered several new examples beyond the standard Ising and Heisenberg models; thus increasing dramatically the number of explicit Hamiltonians available for theoretical investigations. It would be interesting to study in more detail the thermodynamic properties of these new Hamiltonians, and in particular to determine all of their symmetries, including the extensive non-commuting charges. This would require taking the coefficients of the Hamiltonian into account to refine the classification from the general form we consider here to a specific physical system. We hope that our classification can be used to inspire new quantum algorithms and allow researchers to identify the circuits that they use in practice with the Lie algebras in our classification to assess their optimization properties. Moreover, the methods that we have developed can be used to identify the DLA even in cases that fall outside of our classification.

In fact, we have been able to extend the classification presented in here from 2-local one-dimensional topologies to arbitrary graphs. At the basis of these results are the results on permutation-invariant graphs, which serve as an upper bound for the DLA of an arbitrary graph [314]. Further extensions to 3-local Hamiltonians may be possible, but the initial set of Hamiltonians ($2^{63}-1$) would have to be reduced beforehand to make a full enumeration tractable. Additionally, we would like to investigate how the coefficients of the Hamiltonian affect the DLA, which would require extending our method to the case where the generators are linear combinations of Pauli strings.

Another future direction would be to consider other types of systems. For example, instead of spin systems, we could consider fermionic or bosonic Hamiltonians. Such a classification already exists for nearest-neighbor, quadratic Hamiltonians on cubic lattices [338, 352], so this question would have to be explored in the context of non-cubic graphs.

Label	Generating set	Example Model
$\mathfrak{a}_0$	XX	Ising model
$\mathfrak{a}_1$	XY	Kitaev chain
$\mathfrak{a}_2$	XY, YX	Massless free fermion + magnetic field
$\mathfrak{a}_3$	XX, YZ	Kitaev chain + Coulomb
$\mathfrak{a}_4$	XX, YY	XY-model
$\mathfrak{a}_5$	XY, YZ	
$\mathfrak{a}_6$	XX, YZ, ZY	Massless free fermion + magnetic field + Coulomb
$\mathfrak{a}_7$	XX, YY, ZZ	Heisenberg chain
$\mathfrak{a}_8$	XX, XZ	Ising model + transverse field
$\mathfrak{a}_9$	XY, XZ	Kitaev chain + longitudinal field
$\mathfrak{a}_{10}$	XY, YZ, ZX	Heisenberg
$\mathfrak{a}_{11}$	XY, YX, YZ	XY-model + longitudinal field
$\mathfrak{a}_{12}$	XX, XY, YZ	
$\mathfrak{a}_{13}$	XX, YY, YZ	XY-model + longitudinal field
$\mathfrak{a}_{14}$	XX, YY, XY	XY-model + transverse field
$\mathfrak{a}_{15}$	XX, XY, XZ	Ising model + arbitrary field
$\mathfrak{a}_{16}$	XY, YX, YZ, ZY	Kitaev chain + longitudinal field
$\mathfrak{a}_{17}$	XX, XY, ZX	Ising model + arbitrary field
$\mathfrak{a}_{18}$	XX, XZ, YY, ZY	XY-model + arbitrary field
$\mathfrak{a}_{19}$	XX, XY, ZX, YZ	
$\mathfrak{a}_{20}$	XX, YY, ZZ, ZY	Heisenberg chain + magnetic field
$\mathfrak{a}_{21}$	XX, YY, XY, ZX	XY-model + arbitrary field
$\mathfrak{a}_{22}$	XX, XY, XZ, YX, ZX	Ising model + arbitrary field
$\mathfrak{b}_0$	XI, IX	Uncoupled spins
$\mathfrak{b}_1$	XX, XI, IX	Ising model
$\mathfrak{b}_2$	XY, XI, IX	Kitaev chain + longitudinal field
$\mathfrak{b}_3$	XI, YI, IX, IY	Uncoupled spins
$\mathfrak{b}_4$	XX, XY, XI, IX	Ising model + arbitrary field

Table 4.1: **List of generators of the DLAs in Theorem 4.5.1 and examples of conventional spin models that have the equivalent DLA.**

Chapter 5

Conclusion

In this thesis we explored the use of variational quantum algorithms from three different perspectives. Then, in Chapter 2, we numerically investigated VQE optimization schemes and ways to improve them. In Chapter 3, we took a more geometric to variational optimization and incorporated ideas from differential geometry and Riemannian optimization into the current VQE paradigm. We concluded with a more mathematical approach in Chapter 4 to classify objects called dynamical Lie algebras, which are related to the trainability of variational quantum circuits.

At the moment, the field of variational quantum computing is in an interesting state. From a theoretical point of view, the results in Chapter 4 and [314], combined with the most recent iteration of barren plateau results in [331] paint a bleak picture: Most circuits one will have exponentially-sized DLAs which will have barren plateaus and can therefore not be optimized. Furthermore, the small known set of variational quantum circuits with polynomially-sized DLAs are known to be efficiently simulatable [333, 341], and fast-forwardable [342], leaving little room for overparameterization phenomena of Section 2.2 that makes optimizing circuits efficient. Additionally, it seems that the current levels of noise in quantum hardware further reduce the trainability of variational quantum circuits [83], which will likely make applying the ideas of Section 2.3 and Chapter 3 challenging in practice.

The problem is that it is hard to challenge these theoretical results due to the absence of quantum computers (with reasonable error rates). Since one can only classically simulate systems around 20 qubits, where barren plateau issues might not be that prohibitive, there is no real way of testing the wide variety of training heuristics and initialization strategies to reliably train variational quantum circuits [113, 108, 37, 335, 2]. The field is

therefore in the opposite state of modern Deep Learning, where the neural networks which have effectively solved computer vision and natural language rely on many heuristics for training [353, 354, 355], while their training dynamics are not well understood. In fact, from a complexity theory point of view, training neural networks can be NP-Complete or NP-Hard [356, 357, 358], depending on the choice of neural network. Additionally, vanishing gradients also plagued machine learning approaches for a while, until we found suitable initialization strategies[359] and architectures to circumvent those [360]. Similarly, tailored initialization strategies can help with the convergence of DMRG algorithms [361].

These may be false equivalences. Perhaps the optimization problems that plague variational quantum algorithms are indeed as bad as the theory makes them out to be, and we should not compare them to heuristics in other numerical methods that are crucial to get stable performance. However, until we can test variational quantum algorithms at scale they merit further investigation. Luckily, quantum hardware is continuously improving, and we are getting closer to meso-scale devices with low enough error rates that we may be able to test these algorithms in the wild.

References

[1] **Wiersema**, **Roeland**, Cunlu Zhou, Yvette de Sereville, Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen. Exploring entanglement and optimization within the hamiltonian variational ansatz. *PRX Quantum*, 1(2):020319, 12 2020.

[2] **Wiersema**, **Roeland**, Cunlu Zhou, Juan Felipe Carrasquilla, and Yong Baek Kim. Measurement-induced entanglement phase transitions in variational quantum circuits. *SciPost Phys.*, 14(6):147, 6 2023.

[3] **Wiersema**, **Roeland**, Leonardo Guerini, Juan Felipe Carrasquilla, and Leandro Aolita. Circuit connectivity boosts by quantum-classical-quantum interfaces. *Phys. Rev. Res.*, 4:043221, Dec 2022.

[4] **Roeland Wiersema** and Nathan Killoran. Optimizing quantum circuits with Riemannian gradient flow. *Phys. Rev. A*, 107(6):062421, Jun 2023.

[5] **Wiersema**, **Roeland**, Dylan Lewis, David Wierichs, Juan Carrasquilla, and Nathan Killoran. Here comes the SU(N): multivariate quantum gates and gradients. *Quantum*, 8:1275, March 2024.

[6] **Wiersema**, **Roeland**, Efekan Kökcü, Alexander F Kemper, and Bojko N Bakalov. Classification of dynamical Lie algebras for translation-invariant 2-local spin systems in one dimension. *arXiv:2203.05690*, 9 2023.

[7] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 8 1996.

[8] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[9] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery.

[10] Ewin Tang. Dequantizing algorithms to understand quantum advantage in machine learning. *Nature Reviews Physics*, 4(11):692–693, 2022.

[11] Richard P. Feynman. Quantum mechanical computers. *Foundations of Physics*, 16(6):507–531, 6 1986.

[12] I. M. Georgescu, S. Ashhab, and Franco Nori. Quantum simulation. *Rev. Mod. Phys.*, 86(1):153–185, Mar 2014.

[13] Alexander M Dalzell, Sam McArdle, Mario Berta, Przemyslaw Bienias, Chi-Fang Chen, András Gilyén, Connor T Hann, Michael J Kastoryano, Emil T Khabiboulline, Aleksander Kubica, et al. Quantum algorithms: A survey of applications and end-to-end complexities. *arXiv preprint arXiv:2310.03011*, 2023.

[14] Dennis Willsch, Madita Willsch, Carlos D Gonzalez Calaza, Fengping Jin, Hans De Raedt, Marika Svensson, and Kristel Michielsen. Benchmarking advantage and d-wave 2000q quantum annealers with exact cover problems. *Quantum Information Processing*, 21(4):141, 2022.

[15] Sepehr Ebadi, Tout T. Wang, Harry Levine, Alexander Keesling, Giulia Semeghini, Ahmed Omran, Dolev Bluvstein, Rhine Samajdar, Hannes Pichler, Wen Wei Ho, Soonwon Choi, Subir Sachdev, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. Quantum phases of matter on a 256-atom programmable quantum simulator. *Nature*, 595(7866):227–232, 2021.

[16] Joschka Roffe. Quantum error correction: an introductory guide. *Contemporary Physics*, 60(3):226–245, 2019.

[17] Iris Cong, Harry Levine, Alexander Keesling, Dolev Bluvstein, Sheng-Tao Wang, and Mikhail D. Lukin. Hardware-efficient, fault-tolerant quantum computation with rydberg atoms. *Phys. Rev. X*, 12:021049, Jun 2022.

[18] Lukas Postler, Sascha Heussen, Ivan Pogorelov, Manuel Rispler, Thomas Feldker, Michael Meth, Christian D. Marciniak, Roman Stricker, Martin Ringbauer, Rainer Blatt, Philipp Schindler, Markus Müller, and Thomas Monz. Demonstration of fault-tolerant universal quantum gate operations. *Nature*, 605(7911):675–680, 2022.

[19] Rajeev Acharya, Igor Aleiner, Richard Allen, Trond I. Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham Asfaw, Juan Atalaya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Joao Basso, Andreas Bengtsson, Sergio Boixo, Gina Bortoli, Alexandre Bourassa, Jenna Bovaird, Leon Brill, Michael Broughton, Bob B. Buckley, David A. Buell, Tim Burger, Brian Burkett, Nicholas Bushnell, Yu Chen, Zijun Chen, Ben Chiaro, Josh Cogan, Roberto Collins, Paul Conner, William Courtney, Alexander L. Crook, Ben Curtin, Dripto M. Debroy, Alexander Del Toro Barba, Sean Demura, Andrew Dunsworth, Daniel Eppens, Catherine Erickson, Lara Faoro, Edward Farhi, Reza Fatemi, Leslie Flores Burgos, Ebrahim Forati, Austin G. Fowler, Brooks Foxen, William Giang, Craig Gidney, Dar Gilboa, Marissa Giustina, Alejandro Grajales Dau, Jonathan A. Gross, Steve Habegger, Michael C. Hamilton, Matthew P. Harrigan, Sean D. Harrington, Oscar Higgott, Jeremy Hilton, Markus Hoffmann, Sabrina Hong, Trent Huang, Ashley Huff, William J. Huggins, Lev B. Ioffe, Sergei V. Isakov, Justin Iveland, Evan Jeffrey, Zhang Jiang, Cody Jones, Pavol Juhas, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Tanuj Khattar, Mostafa Khezri, Mária Kieferová, Seon Kim, Alexei Kitaev, Paul V. Klimov, Andrey R. Klots, Alexander N. Korotkov, Fedor Kostritsa, John Mark Kreikebaum, David Landhuis, Pavel Laptev, Kim-Ming Lau, Lily Laws, Joonho Lee, Kenny Lee, Brian J. Lester, Alexander Lill, Wayne Liu, Aditya Locharla, Erik Lucero, Fionn D. Malone, Jeffrey Marshall, Orion Martin, Jarrod R. McClean, Trevor McCourt, Matt McEwen, Anthony Megrant, Bernardo Meurer Costa, Xiao Mi, Kevin C. Miao, Masoud Mohseni, Shirin Montazeri, Alexis Morvan, Emily Mount, Wojciech Mruczkiewicz, Ofer Naaman, Matthew Neeley, Charles Neill, Ani Nersisyan, Hartmut Neven, Michael Newman, Jiun How Ng, Anthony Nguyen, Murray Nguyen, Murphy Yuezhen Niu, Thomas E. O'Brien, Alex Opremcak, John Platt, Andre Petukhov, Rebecca Potter, Leonid P. Pryadko, Chris Quintana, Pedram Roushan, Nicholas C. Rubin, Negar Saei, Daniel Sank, Kannan Sankaragomathi, Kevin J. Satzinger, Henry F. Schurkus, Christopher Schuster, Michael J. Shearn, Aaron Shorter, Vladimir Shvarts, Jindra Skruzny, Vadim Smelyanskiy, W. Clarke Smith, George Sterling, Doug Strain, Marco Szalay, Alfredo Torres, Guifre Vidal, Benjamin Villalonga, Catherine Vollgraff Heidweiller, Theodore White, Cheng Xing, Z. Jamie Yao, Ping Yeh, Juhwan Yoo, Grayson Young, Adam Zalcman, Yaxing Zhang, Ningfeng Zhu, and Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. *Nature*, 614(7949):676–681, 2023.

[20] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, 8 2018.

[21] Yong (Alexander) Liu, Xin (Lucy) Liu, Fang (Nancy) Li, Haohuan Fu, Yuling Yang, Jiawei Song, Pengpeng Zhao, Zhen Wang, Dajia Peng, Huarong Chen, Chu Guo, Heliang Huang, Wenzhao Wu, and Dexun Chen. Closing the "quantum supremacy" gap: achieving real-time simulation of a random quantum circuit using a new sunway supercomputer. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, New York, NY, USA, 2021. Association for Computing Machinery.

[22] Jacob F. F. Bulmer, Bryn A. Bell, Rachel S. Chadwick, Alex E. Jones, Diana Moise, Alessandro Rigazzi, Jan Thorbecke, Utz-Uwe Haus, Thomas Van Vaerenbergh, Raj B. Patel, Ian A. Walmsley, and Anthony Laing. The boundary for quantum advantage in gaussian boson sampling. *Science Advances*, 8(4):eabl9236, 2022.

[23] Joseph Tindall, Matthew Fishman, E. Miles Stoudenmire, and Dries Sels. Efficient tensor network simulation of ibm's eagle kicked ising experiment. *PRX Quantum*, 5:010308, Jan 2024.

[24] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 8 2021.

[25] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H Booth, et al. The variational quantum eigensolver: a review of methods and best practices. *Physics Reports*, 986:1–128, 11 2022.

[26] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, 5 2020.

[27] Domenico d'Alessandro. *Introduction to quantum control and dynamics*. Chapman and hall/CRC, 2nd edition, 7 2021.

[28] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th edition, 2011.

[29] 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, Jul 2014.

[30] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. Quantum circuit learning. *Phys. Rev. A*, 98(3):032309, 9 2018.

[31] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A*, 99(3):032331, 3 2019.

[32] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, 9 2017.

[33] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 10 2019.

[34] Carlos Bravo-Prieto, Josep Lumbreras-Zarapico, Luca Tagliacozzo, and José I. Latorre. Scaling of variational quantum circuit depth for condensed matter systems. *Quantum*, 4:272, May 2020.

[35] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):4812, 11 2018.

[36] Kunal Sharma, M. Cerezo, Lukasz Cincio, and Patrick J. Coles. Trainability of dissipative perceptron-based quantum neural networks. *Phys. Rev. Lett.*, 128:180505, May 2022.

[37] Tyler Volkoff and Patrick J Coles. Large gradients via correlation in random parameterized quantum circuits. *Quantum Science and Technology*, 6(2):025008, jan 2021.

[38] Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature communications*, 12(1):1791, 2021.

[39] Dave Wecker, Matthew B Hastings, and Matthias Troyer. Progress towards practical quantum variational algorithms. *Phys. Rev. A*, 92(4):042303, 10 2015.

[40] Wen Wei Ho and Timothy H. Hsieh. Efficient variational simulation of non-trivial quantum states. *SciPost Phys.*, 6(3):29, 3 2019.

[41] Chris Cade, Lana Mineh, Ashley Montanaro, and Stasja Stanisic. Strategies for solving the Fermi-Hubbard model on near-term quantum computers. *Phys. Rev. B*, 102(23):235122, Dec 2020.

[42] J. Eisert, M. Cramer, and M. B. Plenio. Colloquium: Area laws for the entanglement entropy. *Rev. Mod. Phys.*, 82(1):277–306, 2 2010.

[43] Jens Eisert. Entanglement and Tensor Network States. In Eva Pavarini, Erik Koch, and Ulrich Schollwöck, editors, *Emergent Phenomena in Correlated Matter*, volume 3 of *Modeling and Simulation*, page 520 p., Jülich, Sep 2013. Autumn School on Correlated Electrons, Jülich (Germany), 23 Sep 2013 - 27 Sep 2013, Forschungszentrum Jülich Zentralbibliothek, Verlag.

[44] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2 2017.

[45] Or Sharir, Yoav Levine, Noam Wies, Giuseppe Carleo, and Amnon Shashua. Deep Autoregressive Models for the Efficient Variational Simulation of Many-Body Quantum Systems. *Phys. Rev. Lett.*, 124(2):020503, 1 2020.

[46] Juan Carrasquilla, Giacomo Torlai, Roger G. Melko, and Leandro Aolita. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, 1(3):155–161, 3 2019.

[47] Mohamed Hibat-Allah, Martin Ganahl, Lauren E Hayward, Roger G Melko, and Juan Carrasquilla. Recurrent neural network wave functions. *Phys. Rev. Res.*, 2(2):023358, 6 2020.

[48] Dong-Ling Deng, Xiaopeng Li, and S. Das Sarma. Quantum Entanglement in Neural Network States. *Phys. Rev. X*, 7(2):021021, May 2017.

[49] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm. *arXiv preprint arxiv:1411.4028*, 11 2014.

[50] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum Computation by Adiabatic Evolution. *arXiv preprint arxiv:quant-ph/0001106*, 1 2000.

[51] Glen Bigan Mbeng, Rosario Fazio, and Giuseppe Santoro. Quantum Annealing: a journey through Digitalization, Control, and hybrid Quantum Variational schemes. *arXiv preprint arXiv:1906.08948*, 6 2019.

[52] David Wierichs, Christian Gogolin, and Michael Kastoryano. Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer. *Phys. Rev. Research*, 2(4):043246, Nov 2020.

[53] Fabio Franchini. *An Introduction to Integrable Techniques for One-Dimensional Quantum Systems*, volume 940. Springer International Publishing, 2017.

[54] Elliott Lieb and Daniel Mattis. Ordering Energy Levels of Interacting Spin Systems. *Journal of Mathematical Physics*, 3(4):749–751, 07 1962.

[55] Román Orús. Tensor Networks for Complex Quantum Systems. *Nature Reviews Physics*, 1(9):538–550, September 2019.

[56] Matthew J. S. Beach, Roger G. Melko, Tarun Grover, and Timothy H. Hsieh. Making trotters sprint: A variational imaginary time ansatz for quantum many-body systems. *Phys. Rev. B*, 100(9):094434, 9 2019.

[57] Hui Li and F. D. M. Haldane. Entanglement Spectrum as a Generalization of Entanglement Entropy: Identification of Topological Order in Non-Abelian Fractional Quantum Hall Effect States. *Phys. Rev. Lett.*, 101(1):010504, 7 2008.

[58] Alexei Kitaev and John Preskill. Topological Entanglement Entropy. *Phys. Rev. Lett.*, 96(11):110404, 3 2006.

[59] Sergei V. Isakov, Matthew B. Hastings, and Roger G. Melko. Topological entanglement entropy of a Bose–Hubbard spin liquid. *Nature Physics*, 7(10):772–775, 7 2011.

[60] Yi Zhang, Tarun Grover, and Ashvin Vishwanath. Entanglement Entropy of Critical Spin Liquids. *Phys. Rev. Lett.*, 107(6):067202, 8 2011.

[61] Zhi-Cheng Yang, Claudio Chamon, Alioscia Hamma, and Eduardo R. Mucciolo. Two-Component Structure in the Entanglement Spectrum of Highly Excited States. *Phys. Rev. Lett.*, 115(26):267206, 12 2015.

[62] Daniel Shaffer, Claudio Chamon, Alioscia Hamma, and Eduardo R Mucciolo. Irreversibility and entanglement spectrum statistics in quantum circuits. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(12):P12007, 12 2014.

[63] Scott D. Geraedts, Rahul Nandkishore, and Nicolas Regnault. Many-body localization and thermalization: Insights from the entanglement spectrum. *Phys. Rev. B*, 93(17):174202, 5 2016.

[64] Gabriel Matos, Sonika Johri, and Zlatko Papić. Quantifying the efficiency of state preparation via quantum variational eigensolvers. *PRX Quantum*, 2:010309, Jan 2021.

[65] Marko Žnidarič. Entanglement of random vectors. *Journal of Physics A: Mathematical and Theoretical*, 40(3):F105–F111, 12 2006.

[66] V A Marčenko and L A Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, 4 1967.

[67] Don N. Page. Average entropy of a subsystem. *Phys. Rev. Lett.*, 71(9):1291–1294, 8 1993.

[68] Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum*, 3:214, December 2019.

[69] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. *Learning and generalization in overparameterized neural networks, going beyond two layers*, page 6158–6169. Curran Associates Inc., Red Hook, NY, USA, 2019.

[70] Zixiang Chen, Yuan Cao, Difan Zou, and Quanquan Gu. How much over-parameterization is sufficient to learn deep re{lu} networks? In *International Conference on Learning Representations*, 2021.

[71] Herschel A. Rabitz, Michael M. Hsieh, and Carey M. Rosenthal. Quantum Optimally Controlled Transition Landscapes. *Science*, 303(5666):1998–2001, 3 2004.

[72] Herschel Rabitz, Michael Hsieh, and Carey Rosenthal. Landscape for optimal control of quantum-mechanical unitary transformations. *Phys. Rev. A*, 72(5):052337, 11 2005.

[73] Benjamin Russell, Herschel Rabitz, and Re-Bing Wu. Control landscapes are almost always trap free: a geometric assessment. *Journal of Physics A: Mathematical and Theoretical*, 50(20):205302, 4 2017.

[74] Bobak Toussi Kiani, Seth Lloyd, and Reevu Maity. Learning Unitaries by Gradient Descent. *arXiv preprint arxiv:2001.11897*, 1 2020.

[75] Madan Lal Mehta. *Random Matrices*. Pure and Applied Mathematics. Academic Press, 3rd edition, 2004.

[76] M B Hastings. An area law for one-dimensional quantum systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(08):P08024, aug 2007.

[77] Itai Arad, Zeph Landau, and Umesh Vazirani. Improved one-dimensional area law for frustration-free systems. *Phys. Rev. B*, 85(19):195145, May 2012.

[78] Itai Arad, Alexei Kitaev, Zeph Landau, and Umesh Vazirani. An area law and sub-exponential algorithm for 1D systems. *arXiv preprint arXiv.1301.1162*, 1 2013.

[79] Pasquale Calabrese and John Cardy. Entanglement entropy and quantum field theory. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(06):P06002, 6 2004.

[80] Sevag Gharibian, Yichen Huang, Zeph Landau, and Seung Woo Shin. Quantum Hamiltonian Complexity. *Foundations and Trends® in Theoretical Computer Science*, 10(3):159–282, 2015.

[81] F. D. M. Haldane. Exact Jastrow-Gutzwiller resonating-valence-bond ground state of the spin-$\frac{1}{2}$ antiferromagnetic Heisenberg chain with $1/r^2$ exchange. *Phys. Rev. Lett.*, 60(7):635–638, 2 1988.

[82] B. Sriram Shastry. Exact solution of an S1/2 Heisenberg antiferromagnetic chain with long-ranged interactions. *Phys. Rev. Lett.*, 60(7):639–642, 2 1988.

[83] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature communications*, 12(1):6961, 11 2021.

[84] Amos Chan, Rahul M. Nandkishore, Michael Pretko, and Graeme Smith. Unitary-projective entanglement dynamics. *Phys. Rev. B*, 99(22):224307, 6 2019.

[85] Brian Skinner, Jonathan Ruhman, and Adam Nahum. Measurement-Induced Phase Transitions in the Dynamics of Entanglement. *Phys. Rev. X*, 9(3):031009, 07 2019.

[86] Yaodong Li, Xiao Chen, and Matthew P. A. Fisher. Measurement-driven entanglement transition in hybrid quantum circuits. *Phys. Rev. B*, 100(13):134306, Oct 2019.

[87] Yaodong Li, Xiao Chen, and Matthew P. A. Fisher. Quantum Zeno effect and the many-body entanglement transition. *Phys. Rev. B*, 98(20):205136, 11 2018.

[88] Xiangyu Cao, Antoine Tilloy, and Andrea De Luca. Entanglement in a fermion chain under continuous monitoring. *SciPost Phys.*, 7(2):24, 8 2019.

[89] Yimu Bao, Soonwon Choi, and Ehud Altman. Theory of the phase transition in random unitary circuits with measurements. *Phys. Rev. B*, 101(10):104301, 3 2020.

[90] Stefanie Czischek, Giacomo Torlai, Sayonee Ray, Rajibul Islam, and Roger G. Melko. Simulating a measurement-induced phase transition for trapped-ion circuits. *Phys. Rev. A*, 104(6):062405, Dec 2021.

[91] Maxwell Block, Yimu Bao, Soonwon Choi, Ehud Altman, and Norman Y. Yao. Measurement-Induced Transition in Long-Range Interacting Quantum Circuits. *Phys. Rev. Lett.*, 128(1):010604, Jan 2022.

[92] Michele Coppola, Emanuele Tirrito, Dragi Karevski, and Mario Collura. Growth of entanglement entropy under local projective measurements. *Phys. Rev. B*, 105(9):094303, Mar 2022.

[93] Alberto Biella and Marco Schiró. Many-Body Quantum Zeno Effect and Measurement-Induced Subradiance Transition. *Quantum*, 5:528, August 2021.

[94] Xhek Turkeshi, Alberto Biella, Rosario Fazio, Marcello Dalmonte, and Marco Schiró. Measurement-induced entanglement transitions in the quantum Ising chain: From infinite to zero clicks. *Phys. Rev. B*, 103(22):224210, Jun 2021.

[95] Chao-Ming Jian, Yi-Zhuang You, Romain Vasseur, and Andreas W. W. Ludwig. Measurement-induced criticality in random quantum circuits. *Phys. Rev. B*, 101(10):104302, 3 2020.

[96] Joris Kattemölle and Jasper van Wezel. Variational quantum eigensolver for the Heisenberg antiferromagnet on the kagome lattice. *Phys. Rev. B*, 106(21):214429, 12 2022.

[97] Cornelius Hempel, Christine Maier, Jonathan Romero, Jarrod McClean, Thomas Monz, Heng Shen, Petar Jurcevic, Ben P. Lanyon, Peter Love, Ryan Babbush, Alán Aspuru-Guzik, Rainer Blatt, and Christian F. Roos. Quantum Chemistry Calculations on a Trapped-Ion Quantum Simulator. *Phys. Rev. X*, 8(3):031022, Jul 2018.

[98] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. R. McClean, J. Carter, W. A. de Jong, and I. Siddiqi. Computation of Molecular Spectra on a Quantum Processor with an Error-Resilient Algorithm. *Phys. Rev. X*, 8(1):011021, Feb 2018.

[99] Google AI Quantum, Collaborators*†, Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Sergio Boixo, Michael Broughton, Bob B Buckley, et al. Hartree-Fock on a superconducting qubit quantum computer. *Science*, 369(6507):1084–1089, 8 2020.

[100] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik, and J. M. Martinis. Scalable Quantum Simulation of Molecular Energies. *Phys. Rev. X*, 6(3):031007, Jul 2016.

[101] Edward Farhi and Hartmut Neven. Classification with Quantum Neural Networks on Near Term Processors. *arXiv preprint arXiv:1802.06002*, 2 2018. arXiv:1802.06002.

[102] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 3 2019.

[103] Maria Schuld and Nathan Killoran. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.*, 122(4):040504, Feb 2019.

[104] Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Phys. Rev. A*, 101(3):032308, Mar 2020.

[105] Hyungwon Kim and David A. Huse. Ballistic Spreading of Entanglement in a Diffusive Nonintegrable System. *Phys. Rev. Lett.*, 111(12):127205, Sep 2013.

[106] Adam Nahum, Jonathan Ruhman, Sagar Vijay, and Jeongwan Haah. Quantum Entanglement Growth under Random Unitary Dynamics. *Phys. Rev. X*, 7(3):031016, Jul 2017.

[107] Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe. Entanglement-Induced Barren Plateaus. *PRX Quantum*, 2:040316, Oct 2021.

[108] Taylor L. Patti, Khadijeh Najafi, Xun Gao, and Susanne F. Yelin. Entanglement devised barren plateau mitigation. *Phys. Rev. Res.*, 3(3):033090, Jul 2021.

[109] Zoë Holmes, Kunal Sharma, M. Cerezo, and Patrick J. Coles. Connecting Ansatz Expressibility to Gradient Magnitudes and Barren Plateaus. *PRX Quantum*, 3(1):010313, Jan 2022.

[110] Honeywell Quantum. Mid-circuit measurements on the System Model H1. https://www.honeywell.com/us/en/company/quantum/quantum-computer/, 2021. [Online; accessed 07/18/2021].

[111] IBM Quantum. Mid-circuit Measurements Tutorial. https://quantum-computing.ibm.com/lab/docs/iql/manage/systems/midcircuit-measurement/, 2021. [Online; accessed 07/18/2021].

[112] Zijun Chen, Kevin J. Satzinger, Juan Atalaya, Alexander N. Korotkov, Andrew Dunsworth, Daniel Sank, Chris Quintana, Matt McEwen, Rami Barends, Paul V. Klimov, Sabrina Hong, Cody Jones, Andre Petukhov, Dvir Kafri, Sean Demura, Brian Burkett, Craig Gidney, Austin G. Fowler, Alexandru Paler, Harald Putterman, Igor Aleiner, Frank Arute, Kunal Arya, Ryan Babbush, Joseph C. Bardin, Andreas Bengtsson, Alexandre Bourassa, Michael Broughton, Bob B. Buckley, David A. Buell, Nicholas Bushnell, Benjamin Chiaro, Roberto Collins, William Courtney, Alan R. Derk, Daniel Eppens, Catherine Erickson, Edward Farhi, Brooks Foxen, Marissa Giustina, Ami Greene, Jonathan A. Gross, Matthew P. Harrigan, Sean D. Harrington, Jeremy Hilton, Alan Ho, Trent Huang, William J. Huggins, L. B. Ioffe, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Kostyantyn Kechedzhi, Seon Kim, Alexei Kitaev, Fedor Kostritsa, David Landhuis, Pavel Laptev, Erik Lucero, Orion Martin, Jarrod R. McClean, Trevor McCourt, Xiao Mi, Kevin C. Miao, Masoud Mohseni, Shirin Montazeri, Wojciech Mruczkiewicz, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Michael Newman, Murphy Yuezhen Niu, Thomas E. O'Brien, Alex Opremcak, Eric Ostby, Bálint Pató, Nicholas Redd, Pedram Roushan, Nicholas C. Rubin, Vladimir Shvarts, Doug Strain, Marco Szalay, Matthew D. Trevithick, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Juhwan Yoo, Adam Zalcman, Hartmut Neven, Sergio Boixo, Vadim Smelyanskiy, Yu Chen, Anthony Megrant, Julian Kelly, and Google Quantum AI. Exponential suppression of bit or phase errors with cyclic error correction. *Nature*, 595(7867):383–387, 2021.

[113] Harper R. Grimsley, Sophia E. Economou, Edwin Barnes, and Nicholas J. Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature Communications*, 10(1):3007, Jul 2019.

[114] Vincenzo Alba and Pasquale Calabrese. Entanglement and thermodynamics after a quantum quench in integrable systems. *Proceedings of the National Academy of Sciences*, 114(30):7947–7951, 7 2017.

[115] M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1):1791, Mar 2021.

[116] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. General parameter-shift rules for quantum gradients. *Quantum*, 6:677, 3 2022.

[117] Andrea Mari, Thomas R. Bromley, and Nathan Killoran. Estimating the gradient and higher-order derivatives on quantum hardware. *Phys. Rev. A*, 103(1):012405, Jan 2021.

[118] Artur F. Izmaylov, Robert A. Lang, and Tzu-Ching Yen. Analytic gradients in variational quantum algorithms: Algebraic extensions of the parameter-shift rule to general unitary transformations. *Phys. Rev. A*, 104(6):062443, 12 2021.

[119] Oleksandr Kyriienko and Vincent E. Elfving. Generalized quantum circuit differentiation rules. *Phys. Rev. A*, 104(5):052417, Nov 2021.

[120] Johannes Jakob Meyer, Johannes Borregaard, and Jens Eisert. A variational toolbox for quantum multi-parameter estimation. *npj Quantum Information*, 7(1):89, Jun 2021.

[121] Leonardo Banchi and Gavin E. Crooks. Measuring Analytic Gradients of General Quantum Evolution with the Stochastic Parameter Shift Rule. *Quantum*, 5:386, January 2021.

[122] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin. Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices. *Phys. Rev. X*, 10(2):021067, Jun 2020.

[123] Andrea Skolik, Jarrod R McClean, Masoud Mohseni, Patrick van der Smagt, and Martin Leib. Layerwise learning for quantum neural networks. *Quantum Machine Intelligence*, 3(1):1–11, 1 2021.

[124] Arthur Pesah, M. Cerezo, Samson Wang, Tyler Volkoff, Andrew T. Sornborger, and Patrick J. Coles. Absence of Barren Plateaus in Quantum Convolutional Neural Networks. *Phys. Rev. X*, 11(4):041011, Oct 2021.

[125] Bálint Koczor and Simon C. Benjamin. Quantum natural gradient generalized to noisy and nonunitary circuits. *Phys. Rev. A*, 106(6):062416, Dec 2022.

[126] R. R. Ferguson, L. Dellantonio, A. Al Balushi, K. Jansen, W. Dür, and C. A. Muschik. Measurement-Based Variational Quantum Eigensolver. *Phys. Rev. Lett.*, 126(22):220501, Jun 2021.

[127] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, Jan 2009.

[128] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 11 2019.

[129] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.*, 94:015004, Feb 2022.

[130] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. Error Mitigation for Short-Depth Quantum Circuits. *Phys. Rev. Lett.*, 119(18):180509, Nov 2017.

[131] Suguru Endo, Simon C. Benjamin, and Ying Li. Practical Quantum Error Mitigation for Near-Future Applications. *Phys. Rev. X*, 8(3):031027, Jul 2018.

[132] Ying Li and Simon C. Benjamin. Efficient Variational Quantum Simulator Incorporating Active Error Minimization. *Phys. Rev. X*, 7(2):021050, Jun 2017.

[133] Abhinav Kandala, Kristan Temme, Antonio D. Córcoles, Antonio Mezzacapo, Jerry M. Chow, and Jay M. Gambetta. Error mitigation extends the computational reach of a noisy quantum processor. *Nature*, 567(7749):491–495, Mar 2019.

[134] Sergey Bravyi, Graeme Smith, and John A. Smolin. Trading Classical and Quantum Computational Resources. *Phys. Rev. X*, 6(2):021043, Jun 2016.

[135] V. Dunjko, Y. Ge, and J. I. Cirac. Computational speedups using small quantum devices. *Phys. Rev. Lett.*, 121(25):250501, 12 2018.

[136] Tianyi Peng, Aram W. Harrow, Maris Ozols, and Xiaodi Wu. Simulating Large Quantum Circuits on a Small Quantum Computer. *Phys. Rev. Lett.*, 125(15):150504, Oct 2020.

[137] Wei Tang, Teague Tomesh, Martin Suchara, Jeffrey Larson, and Margaret Martonosi. CutQC: Using Small Quantum Computers for Large Quantum Circuit Evaluations.

In Tim Sherwood, Emery Berger, and Christos Kozyrakis, editors, *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS 2021, pages 473–486, New York, NY, USA, 4 2021. Association for Computing Machinery.

[138] Michael A. Perlin, Zain H. Saleem, Martin Suchara, and James C. Osborn. Quantum circuit cutting with maximum-likelihood tomography. *NPJ Quantum Information*, 7(1):64, Apr 2021.

[139] Kosuke Mitarai and Keisuke Fujii. Constructing a virtual two-qubit gate by sampling single-qubit operations. *New Journal of Physics*, 23(2):023021, 2 2021.

[140] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, Oct 2020.

[141] Christopher Ferrie and Joseph Emerson. Frame representations of quantum mechanics and the necessity of negativity in quasi-probability representations. *Journal of Physics A: Mathematical and Theoretical*, 41(35):352001, Jul 2008.

[142] Christopher Ferrie. Quasi-probability representations of quantum theory with applications to quantum information science. *Reports on Progress in Physics*, 74(11):24, 10 2011.

[143] A. Mari and J. Eisert. Positive Wigner functions render classical simulation of quantum computation efficient. *Phys. Rev. Lett.*, 109(23):230503, 12 2012.

[144] V. Veitch, N. Wiebe, C. Ferrie, and J. Emerson. Efficient simulation scheme for a class of quantum optics experiments with non-negative Wigner representation. *New Journal of Physics*, 15(1):013037, 1 2013.

[145] Hakop Pashayan, Joel J. Wallman, and Stephen D. Bartlett. Estimating Outcome Probabilities of Quantum Circuits Using Quasiprobabilities. *Phys. Rev. Lett.*, 115(7):070501, Aug 2015.

[146] Naomichi Hatano and Masuo Suzuki. Representation basis in quantum Monte Carlo calculations and the negative-sign problem. *Physics Letters A*, 163(4):246–249, 3 1992.

[147] E.Y Loh, J.E Gubernatis, R.T Scalettar, S.R White, D.J Scalapino, and R.L Sugar. Sign problem in the numerical simulation of many-electron systems. *Phys. Rev. B*, 41(13):9301–9307, 5 1990.

[148] Matthias Troyer and Uwe-Jens Wiese. Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations. *Phys. Rev. Lett.*, 94(17):170201, May 2005.

[149] C. Radhakrishna Rao. Calculus of Generalized Inverses of Matrices Part I: General Theory. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 29(3):317–342, 1967.

[150] Milad Marvian, Daniel A. Lidar, and Itay Hen. On the computational complexity of curing non-stoquastic Hamiltonians. *Nature Communications*, 10(1):1571, Apr 2019.

[151] D. Hangleiter, I. Roth, D. Nagaj, and J. Eisert. Easing the Monte Carlo sign problem. *Science advances*, 6(33):eabb8341, 8 2020.

[152] Asher Peres. *Quantum theory : concepts and methods*. Fundamental theories of physics. Kluwer Academic, Dordrecht, 1993.

[153] Scott Aaronson. Shadow Tomography of Quantum States. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 325–338, New York, NY, USA, 6 2018. Association for Computing Machinery.

[154] R. Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 7 1955.

[155] Giacomo Torlai, Christopher J. Wood, Atithi Acharya, Giuseppe Carleo, Juan Carrasquilla, and Leandro Aolita. Quantum process tomography with unsupervised learning and tensor networks. *Nature Communications*, 14(1):2858, 2023.

[156] Leonardo Guerini, **Roeland Wiersema**, Juan Felipe Carrasquilla, and Leandro Aolita. Quasiprobabilistic state-overlap estimator for NISQ devices. *ArXiv preprint: 2112.11618*, 2022.

[157] Juan Carrasquilla, Di Luo, Felipe Pérez, Ashley Milsted, Bryan K. Clark, Maksims Volkovs, and Leandro Aolita. Probabilistic simulation of quantum circuits using a deep-learning architecture. *Phys. Rev. A*, 104(3):032610, Sep 2021.

[158] Christophe Piveteau and David Sutter. Circuit knitting with classical communication. *IEEE Transactions on Information Theory*, 70(4):2734–2745, 2024.

[159] A. H. Werner, D. Jaschke, P. Silvi, M. Kliesch, T. Calarco, J. Eisert, and S. Montangero. Positive Tensor Network Approach for Simulating Open Quantum Many-Body Systems. *Phys. Rev. Lett.*, 116(23):237201, Jun 2016.

[160] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, Oct 2019.

[161] Eliakim Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*, 26(9):394 – 395, 1920.

[162] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 6 1953.

[163] Chris Sherlock, Paul Fearnhead, and Gareth O. Roberts. The Random Walk Metropolis: Linking Theory and Practice Through a Case Study. *Statistical Science*, 25(2):172 – 190, 5 2010.

[164] A. Gelman, W. R. Gilks, and G. O. Roberts. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110 – 120, 2 1997.

[165] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed Experiment to Test Local Hidden-Variable Theories. *Phys. Rev. Lett.*, 23(15):880–884, Oct 1969.

[166] NJ Cerf and SE Koonin. Monte Carlo simulation of quantum computation. *Mathematics and Computers in Simulation*, 47(2-5):143–152, 8 1998.

[167] J. Baez and J. P. Muniain. *Gauge fields, knots and gravity*, volume 4 of *Series on Knots and Everything*. World Scientific Publishing, 1995.

[168] M. Nakahara. *Geometry, topology and physics*. Graduate Student Series in Physics. Taylor and Francis group, 2003.

[169] M.P. do Carmo. *Riemannian Geometry*. Mathematics. Birkhäuser, 1992.

[170] J.M. Lee and J.M. Lee. *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. Springer, 2003.

[171] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[172] Shun-ichi Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276, 2 1998.

[173] HD Doebner and J Tolar. Quantum mechanics on homogeneous spaces. *Journal of Mathematical Physics*, 16(4):975–984, 1975.

[174] JP Provost and G Vallee. Riemannian structure on manifolds of quantum states. *Communications in Mathematical Physics*, 76:289–301, 1980.

[175] Federico Becca and Sandro Sorella. *Quantum Monte Carlo Approaches for Correlated Systems*. Cambridge University Press, 11 2017.

[176] W. Rossmann. *Lie Groups: An Introduction Through Linear Groups*. Oxford graduate texts in mathematics. Oxford University Press, 5th edition, 1 2002.

[177] Kevin J Sung, Jiahao Yao, Matthew P Harrigan, Nicholas C Rubin, Zhang Jiang, Lin Lin, Ryan Babbush, and Jarrod R McClean. Using models to improve optimizers for variational quantum algorithms. *Quantum Science and Technology*, 5(4):044008, 10 2020.

[178] R. A. Fisher. On the Mathematical Foundations of Theoretical Statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222(594-604):309–368, 1 1922.

[179] Razvan Pascanu and Yoshua Bengio. Revisiting Natural Gradient for Deep Networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Cornell University, 2014.

[180] Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy Natural Gradient as Variational Inference. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5852–5861. PMLR, 7 2018.

[181] Sandro Sorella. Green Function Monte Carlo with Stochastic Reconfiguration. *Phys. Rev. Lett.*, 80(20):4558–4561, 5 1998.

[182] Amara Katabarwa, Sukin Sim, Dax Enshan Koh, and Pierre-Luc Dallaire-Demers. Connecting geometry and performance of two-qubit parameterized quantum circuits. *Quantum*, 6:782, August 2022.

[183] Gary Bécigneul and Octavian-Eugen Ganea. Riemannian Adaptive Optimization Methods. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[184] Constantin Udrişte. *Convex Functions and Optimization Methods on Riemannian Manifolds*, volume 297 of *Mathematics and Its Applications*. Springer Science + Business Media, 7 1994.

[185] Thomas Schulte-Herbrüggen, Steffen j. Glaser, Gunther Dirr, and Uwe Helmke. Gradient Flows for Optimization in Quantum Information and Quantum Dynamics: Foundations and Applications. *Reviews in Mathematical Physics*, 22(06):597–667, 7 2010.

[186] Uwe Helmke and John B. Moore. *Optimization and Dynamical Systems*, volume 84 of *Communications and Control Engineering*. Springer-Verlag London, 2 1994.

[187] Anand Oza, Alexander Pechen, Jason Dominy, Vincent Beltrani, Katharine Moore, and Herschel Rabitz. Optimization search effort over the control landscapes for open quantum systems with Kraus-map evolution. *Journal of Physics A: Mathematical and Theoretical*, 42(20):205305, 5 2009.

[188] Ilia A Luchnikov, Mikhail E Krechetov, and Sergey N Filippov. Riemannian geometry and automatic differentiation for optimization problems of quantum physics and quantum technologies. *New Journal of Physics*, 23(7):073006, 7 2021.

[189] U. Helmke, K. Hüper, J. B. Moore, and Th. Schulte-Herbrüggen. Gradient Flows Computing the C-numerical Range with Applications in NMR Spectroscopy. *Journal of Global Optimization*, 23(3-4):283–308, 8 2002.

[190] S. J. Glaser, T. Schulte-Herbrüggen, M. Sieveking, O. Schedletzky, N. C. Nielsen, O. W. Sørensen, and C. Griesinger. Unitary Control in Quantum Ensembles: Maximizing Signal Intensity in Coherent Spectroscopy. *Science*, 280(5362):421–424, 4 1998.

[191] Markus Hauru, Maarten Van Damme, and Jutho Haegeman. Riemannian optimization of isometric tensor networks. *SciPost Phys.*, 10(2):040, 2 2021.

[192] Qiang Miao and Thomas Barthel. A quantum-classical eigensolver using multiscale entanglement renormalization. *arXiv preprint arXiv:2108.13401*, 2021.

[193] Simone Fiori. Quasi-Geodesic Neural Learning Algorithms Over the Orthogonal Group: A Tutorial. *JMLR*, 6(26):743–781, 12 2005.

[194] Simone Fiori. Learning by Natural Gradient on Noncompact Matrix-Type Pseudo-Riemannian Manifolds. *IEEE Transactions on Neural Networks*, 21(5):841–852, 5 2010.

[195] Scott Wisdom, Thomas Powers, John R. Hershey, Jonathan Le Roux, and Les Atlas. Full-Capacity Unitary Recurrent Neural Networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike V. Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 4887–4895, Red Hook, NY, USA, 10 2016. Curran Associates Inc.

[196] Mario Lezcano-Casado and David Martínez-Rubio. Cheap Orthogonal Constraints in Neural Networks: A Simple Parametrization of the Orthogonal and Unitary Group. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3794–3803. PMLR, 6 2019.

[197] Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building Deep Networks on Grassmann Manifolds. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, volume 32 of *AAAI'18/IAAI'18/EAAI'18*. AAAI Press, 4 2018.

[198] Michael A. Nielsen, Mark R. Dowling, Mile Gu, and Andrew C. Doherty. Quantum Computation as Geometry. *Science*, 311(5764):1133–1135, 2 2006.

[199] Mark R. Dowling and Michael A. Nielsen. The geometry of quantum computation. *Quantum Information & Computation*, 8(10):861–899, November 2008.

[200] Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient Descent Only Converges to Minimizers. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1246–1257, Columbia University, New York, New York, USA, 6 2016. PMLR.

[201] Andreas Arvanitogeōrgos. *An introduction to Lie groups and the geometry of homogeneous spaces*, volume 22. American Mathematical Soc., 2003.

[202] Simone Fiori. Lie-group-type neural system learning by manifold retractions. *Neural Networks*, 21(10):1524–1529, 12 2008.

[203] Tin-Yau Tam. Gradient flows and double bracket equations. *Differential Geometry and its Applications*, 20(2):209–224, 3 2004.

[204] Moody T. Chu and Kenneth R. Driessel. The Projected Gradient Method for Least Squares Matrix Approximations with Spectral Constraints. *SIAM Journal on Numerical Analysis*, 27(4):1050–1060, 8 1990.

[205] R.W. Brockett. Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems. *Linear Algebra and its Applications*, 146:79–91, 2 1991.

[206] Anthony M. Bloch, Roger W. Brockett, and Tudor S. Ratiu. Completely integrable gradient flows. *Communications in Mathematical Physics*, 147(1):57–74, 6 1992.

[207] Franz Wegner. Flow equations for Hamiltonians. *Annalen der Physik*, 506(2):77–91, 1 1994.

[208] Stefan Kehrein. *The Flow Equation Approach to Many-Particle Systems*, volume 216 of *Springer Tracts in Modern Physics*. Springer-Verlag Berlin, 7 2006.

[209] Stanisław D. Głazek and Kenneth G. Wilson. Renormalization of Hamiltonians. *Phys. Rev. D*, 48(12):5863–5872, 12 1993.

[210] C. M. Dawson, J. Eisert, and T. J. Osborne. Unifying Variational Methods for Simulating Quantum Many-Body Systems. *Phys. Rev. Lett.*, 100(13):130501, 3 2008.

[211] Navin Khaneja and Steffen J. Glaser. Cartan decomposition of $SU(2^n)$and control of spin systems. *Chemical Physics*, 267(1):11–23, 6 2001.

[212] Henrique N. Sá Earp and Jiannis K. Pachos. A constructive algorithm for the Cartan decomposition of $SU(2^N)$. *Journal of Mathematical Physics*, 46(8):082108, 8 2005.

[213] Domenico D'Alessandro and Raffaele Romano. Decompositions of unitary evolutions and entanglement dynamics of bipartite quantum systems. *Journal of Mathematical Physics*, 47(8):082109, 8 2006. https://doi.org/10.1063/1.2245205.

[214] Silvère Bonnabel. Stochastic Gradient Descent on Riemannian Manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 9 2013.

[215] Kosuke Mitarai, Tennin Yan, and Keisuke Fujii. Generalization of the Output of a Variational Quantum Eigensolver by Parameter Interpolation with a Low-depth Ansatz. *Phys. Rev. Applied*, 11(4):044087, 4 2019.

[216] Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. Structure optimization for parameterized quantum circuits. *Quantum*, 5:391, January 2021.

[217] Ken M. Nakanishi, Keisuke Fujii, and Synge Todo. Sequential minimal optimization for quantum-classical hybrid algorithms. *Phys. Rev. Res.*, 2(4):043158, Oct 2020.

[218] Alicia B. Magann, Kenneth M. Rudinger, Matthew D. Grace, and Mohan Sarovar. Feedback-Based Quantum Optimization. *Phys. Rev. Lett.*, 129(25):250502, Dec 2022.

[219] Francesca Albertini and Domenico D'Alessandro. Notions of controllability for quantum mechanical systems. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*, volume 2, pages 1589–1594. IEEE, IEEE, 6 2001.

[220] Sonia G Schirmer, H Fu, and Allan I Solomon. Complete controllability of quantum systems. *Phys. Rev. A*, 63(6):063410, 5 2001.

[221] Ho Lun Tang, V.O. Shkolnikov, George S. Barron, Harper R. Grimsley, Nicholas J. Mayhall, Edwin Barnes, and Sophia E. Economou. Qubit-ADAPT-VQE: An Adaptive Algorithm for Constructing Hardware-Efficient Ansätze on a Quantum Processor. *PRX Quantum*, 2(2):020310, Apr 2021.

[222] Jie Liu, Zhenyu Li, and Jinlong Yang. An efficient adaptive variational quantum solver of the Schrödinger equation based on reduced density matrices. *The Journal of Chemical Physics*, 154(24), 06 2021. 244112.

[223] Yordan S Yordanov, Vasileios Armaos, Crispin HW Barnes, and David RM Arvidsson-Shukur. Qubit-excitation-based adaptive variational quantum eigensolver. *Communications Physics*, 4(1):228, 10 2021.

[224] Alicia B. Magann, Sophia E. Economou, and Christian Arenz. Randomized adaptive quantum state preparation. *Phys. Rev. Res.*, 5:033227, Sep 2023.

[225] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M. Sohaib Alam, Guillermo Alonso-Linaje, B. AkashNarayanan, Ali Asadi, Juan Miguel Arrazola, Utkarsh Azad, Sam Banning, Carsten Blank, Thomas R Bromley, Benjamin A. Cordier, Jack Ceroni, Alain Delgado, Olivia Di Matteo, Amintor Dusko, Tanya Garg, Diego Guala, Anthony Hayes, Ryan Hill, Aroosa Ijaz, Theodor Isacsson, David Ittah, Soran Jahangiri, Prateek Jain, Edward Jiang, Ankit Khandelwal, Korbinian Kottmann, Robert A. Lang, Christina Lee, Thomas Loke, Angus Lowe, Keri McKiernan, Johannes Jakob Meyer, J. A. Montañez-Barrera, Romain Moyard, Zeyue Niu, Lee James O'Riordan, Steven Oud, Ashish Panigrahi, Chae-Yeun Park, Daniel Polatajko, Nicolás Quesada, Chase Roberts, Nahum Sá, Isidor Schoch, Borun Shi, Shuli Shu, Sukin Sim, Arshpreet Singh, Ingrid Strandberg, Jay Soni, Antal Száva, Slimane Thabet, Rodrigo A. Vargas-Hernández, Trevor Vincent, Nicola Vitucci, Maurice Weber, David Wierichs, **Roeland Wiersema**, Moritz Willmann, Vincent Wong, Shaoming Zhang, and Nathan Killoran. PennyLane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, abs/1811.04968, 11 2018.

[226] 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*, 4 2019. *arXiv:1904.03206*.

[227] Martin Larocca, Piotr Czarnik, Kunal Sharma, Gopikrishnan Muraleedharan, Patrick J. Coles, and M. Cerezo. Diagnosing Barren Plateaus with Tools from Quantum Optimal Control. *Quantum*, 6:824, September 2022.

[228] Efekan Kökcü, Thomas Steckmann, Yan Wang, JK Freericks, Eugene F Dumitrescu, and Alexander F Kemper. Fixed depth Hamiltonian simulation via Cartan decomposition. *Phys. Rev. Lett.*, 129(7):070501, 8 2022.

[229] Joonho Kim, Jaedeok Kim, and Dario Rosa. Universal effectiveness of high-depth circuits in variational eigenproblems. *Phys. Rev. Res.*, 3(2):023203, 6 2021.

[230] Martin Larocca, Nathan Ju, Diego García-Martín, Patrick J Coles, and Marco Cerezo. Theory of overparametrization in quantum neural networks. *Nature Computational Science*, 3(6):542–551, 6 2023.

[231] Xuchen You, Shouvanik Chakrabarti, and Xiaodi Wu. A convergence theory for over-parameterized variational quantum eigensolvers. *arXiv preprint arXiv:2205.12481*, abs/2205.12481, 5 2022.

[232] Jun Li, Xiaodong Yang, Xinhua Peng, and Chang-Pu Sun. Hybrid Quantum-Classical Approach to Quantum Optimal Control. *Phys. Rev. Lett.*, 118(15):150503, 4 2017.

[233] Gavin E. Crooks. Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition. *arXiv preprint arXiv:1905.13311*, 5 2019.

[234] Dirk Oliver Theis. "Proper" Shift Rules for Derivatives of Perturbed-Parametric Quantum Evolutions. *Quantum*, 7:1052, July 2023.

[235] Lucas Slattery, Benjamin Villalonga, and Bryan K. Clark. Unitary block optimization for variational quantum algorithms. *Phys. Rev. Research*, 4(2):023072, Apr 2022.

[236] Jin-Guo Liu, Yi-Hong Zhang, Yuan Wan, and Lei Wang. Variational quantum eigensolver with fewer qubits. *Phys. Rev. Research*, 1(2):023025, Sep 2019.

[237] Barbara Kraus and Juan I Cirac. Optimal creation of entanglement using a two-qubit gate. *Phys. Rev. A*, 63(6):062309, 5 2001.

[238] Farrokh Vatan and Colin Williams. Optimal quantum circuits for general two-qubit gates. *Phys. Rev. A*, 69(3):032315, Mar 2004.

[239] Farrokh Vatan and Colin P Williams. Realization of a general three-qubit quantum gate. *arXiv*, 1 2004.

[240] Juha J. Vartiainen, Mikko Möttönen, and Martti M. Salomaa. Efficient Decomposition of Quantum Gates. *Phys. Rev. Lett.*, 92(17):177902, Apr 2004.

[241] Alwin Zulehner and Robert Wille. Compiling SU(4) Quantum Circuits to IBM QX Architectures. In Toshiyuki Shibuya, editor, *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, ASPDAC '19, pages 185–190, New York, NY, USA, 1 2019. Association for Computing Machinery.

[242] B. Foxen, C. Neill, A. Dunsworth, P. Roushan, B. Chiaro, A. Megrant, J. Kelly, Zijun Chen, K. Satzinger, R. Barends, F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, S. Boixo, D. Buell, B. Burkett, Yu Chen, R. Collins, E. Farhi, A. Fowler, C. Gidney, M. Giustina, R. Graff, M. Harrigan, T. Huang, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, P. Klimov, A. Korotkov, F. Kostritsa, D. Landhuis, E. Lucero, J. McClean, M. McEwen, X. Mi, M. Mohseni, J. Y. Mutus, O. Naaman, M. Neeley, M. Niu, A. Petukhov, C. Quintana, N. Rubin, D. Sank, V. Smelyanskiy, A. Vainsencher, T. C. White, Z. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis. Demonstrating a Continuous Set of Two-Qubit Gates for Near-Term Quantum Algorithms. *Phys. Rev. Lett.*, 125(12):120504, Sep 2020.

[243] E Groeneveld. A reparameterization to improve numerical optimization in multivariate REML (co) variance component estimation. *Genetics Selection Evolution*, 26(6):537–545, 12 1994.

[244] Tapani Raiko, Harri Valpola, and Yann Lecun. Deep Learning Made Easier by Linear Transformations in Perceptrons. In Neil D. Lawrence and Mark Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 924–932, La Palma, Canary Islands, 21–23 Apr 2012. PMLR.

[245] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei, editors, *International conference on machine learning*, volume 37, pages 448–456. PMLR, Cornell University, 2 2015.

[246] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike V. Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in neural information processing systems*, volume 29, pages 901–909, 2 2016.

[247] Robert Price. A useful theorem for nonlinear devices having Gaussian inputs. *IRE Transactions on Information Theory*, 4(2):69–72, 6 1958.

[248] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR.

[249] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[250] Brian C Hall. *Lie groups, Lie algebras, and representations*. Springer, 2nd edition, 2013.

[251] William Fulton and Joe Harris. *Representation theory: a first course*, volume 129. Springer Science & Business Media, 2013.

[252] Jean-Pierre Serre. *Lie algebras and Lie groups: 1964 lectures given at Harvard University*. Springer, 2009.

[253] Norbert Schuch and Jens Siewert. Natural two-qubit gate for quantum computation using the XY interaction. *Phys. Rev. A*, 67(3):032301, Mar 2003.

[254] T. P. Orlando, J. E. Mooij, Lin Tian, Caspar H. van der Wal, L. S. Levitov, Seth Lloyd, and J. J. Mazo. Superconducting persistent-current qubit. *Phys. Rev. B*, 60(22):15398–15413, Dec 1999.

[255] B. E. Kane. A silicon-based nuclear spin quantum computer. *Nature*, 393(6681):133–137, 5 1998.

[256] A. Imamog¯lu, D. D. Awschalom, G. Burkard, D. P. DiVincenzo, D. Loss, M. Sherwin, and A. Small. Quantum Information Processing Using Quantum Dot Spins and Cavity QED. *Phys. Rev. Lett.*, 83(20):4204–4207, Nov 1999.

[257] Jiaqi Leng, Yuxiang Peng, Yi-Ling Qiao, Ming Lin, and Xiaodi Wu. Differentiable analog quantum computing for optimization and control. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[258] Ralph M Wilcox. Exponential operators and parameter differentiation in quantum physics. *Journal of Mathematical Physics*, 8(4):962–982, 4 1967.

[259] E. T. Whittaker. XVIII.—On the Functions which are represented by the Expansions of the Interpolation-Theory. *Proceedings of the Royal Society of Edinburgh*, 35:181–194, 1915.

[260] Rüdiger Achilles and Andrea Bonfiglioli. The early proofs of the theorem of Campbell, Baker, Hausdorff, and Dynkin. *Archive for history of exact sciences*, 66(3):295–358, 4 2012.

[261] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs. `http://github.com/google/jax`, 2018.

[262] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché Buc, Edward A. Fox, and Roman Garnett, editors, *Advances in neural information processing systems*, volume 32. Cornell University, 12 2019.

[263] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015.

[264] A JAX implementation of the matrix exponential that can be differentiated via automatic differentiation: `https://jax.readthedocs.io/en/latest/_autosummary/jax.scipy.linalg.expm.html`.

[265] Awad H Al-Mohy and Nicholas J Higham. A new scaling and squaring algorithm for the matrix exponential. *SIAM Journal on Matrix Analysis and Applications*, 31(3):970–989, 1 2010.

[266] Lennart Bittel, Jens Watty, and Martin Kliesch. Fast gradient estimation for variational quantum algorithms. *arXiv preprint arXiv:2210.06484*, 10 2022.

[267] **Wiersema**, **Roeland**, Dylan Lewis, David Wierichs, Juan Carrasquilla, and Nathan Killoran. Here comes the SU(N): multivariate quantum gates and gradients. `https://github.com/dwierichs/Here-comes-the-SUN`, 3 2023.

[268] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K. Faehrmann, Barthélémy Meynard-Piganeau, and Jens Eisert. Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum*, 4:314, August 2020.

[269] Aram W. Harrow and John C. Napp. Low-Depth Gradient Measurements Can Improve Convergence in Variational Hybrid Quantum-Classical Algorithms. *Phys. Rev. Lett.*, 126(14):140502, Apr 2021.

[270] Andrew Arrasmith, Lukasz Cincio, Rolando D Somma, and Patrick J Coles. Operator sampling for shot-frugal optimization in variational algorithms. *arXiv preprint arXiv:2004.06252*, 4 2020.

[271] Javier Gil Vidal and Dirk Oliver Theis. Calculus on parameterized quantum circuits. *arXiv preprint arXiv:1812.06323*, 12 2018.

[272] 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*, 4 2019.

[273] Jean-Pierre Serre. *Complex semisimple Lie algebras*. Springer Science & Business Media, 1st edition, 2000.

[274] Eugene Borisovich Dynkin. *American Mathematical Society Translations: Five Papers on Algebra and Group Theory*. American Mathematical Society, 1957.

[275] 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, 7 2021.

[276] P. Scholl, H. J. Williams, G. Bornet, F. Wallner, D. Barredo, L. Henriet, A. Signoles, C. Hainaut, T. Franz, S. Geier, A. Tebben, A. Salzinger, G. Zürn, T. Lahaye, M. Weidemüller, and A. Browaeys. Microwave Engineering of Programmable XXZ Hamiltonians in Arrays of Rydberg Atoms. *PRX Quantum*, 3(2):020303, Apr 2022.

[277] Mohannad M. Ibrahim, Hamed Mohammadbagherpoor, Cynthia Rios, Nicholas T. Bronn, and Gregory T. Byrd. Evaluation of parameterized quantum circuits with cross-resonance pulse-driven entanglers. *IEEE Transactions on Quantum Engineering*, 3:1–13, 2022.

[278] Oinam Romesh Meitei, Bryan T. Gard, George S. Barron, David P. Pappas, Sophia E. Economou, Edwin Barnes, and Nicholas J. Mayhall. Gate-free state preparation for fast variational quantum eigensolver simulations. *npj Quantum Information*, 7(1):155, 10 2021.

[279] Dylan Lewis, **Roeland Wiersema**, Juan Carrasquilla, and Sougato Bose. Geodesic Algorithm for Unitary Gate Design with Time-Independent Hamiltonians. *arXiv preprint arXiv:2401.05973*, 2024.

[280] Enrico Fontana, Dylan Herman, Shouvanik Chakrabarti, Niraj Kumar, Romina Yalovetzky, Jamie Heredge, Shree Hari Sureshbabu, and Marco Pistoia. The Adjoint Is All You Need: Characterizing Barren Plateaus in Quantum Ansätze. *arXiv preprint arXiv:2309.07902*, 9 2023.

[281] Matthew L Goh, Martin Larocca, Lukasz Cincio, M Cerezo, and Frédéric Sauvage. Lie-algebraic classical simulations for variational quantum computing. *arXiv preprint arXiv:2308.01432*, 8 2023.

[282] Mohsen Heidari, Masih Mozakka, and Wojciech Szpankowski. Efficient gradient estimation of variational quantum circuits with lie algebraic symmetries. *arXiv preprint arXiv:2404.05108*, 2024.

[283] Robert Zeier and Thomas Schulte-Herbrüggen. Symmetry principles in quantum systems theory. *Journal of mathematical physics*, 52(11):113510, 11 2011.

[284] Anthony W Knapp. *Lie Groups Beyond an Introduction*, volume 140. Springer Science & Business Media, 2013.

[285] Eugene P. Wigner. *Group Theory - And its Application to the Quantum Mechanics of Atomic Spectra*. Elsevier, 1959.

[286] Freeman J Dyson. The threefold way. Algebraic structure of symmetry groups and ensembles in quantum mechanics. *Journal of Mathematical Physics*, 3(6):1199–1215, 11 1962.

[287] Alan Edelman and Sungwoo Jeong. On the Cartan decomposition for classical random matrix ensembles. *Journal of Mathematical Physics*, 63(6), 6 2022.

[288] Alexander Altland and Martin R. Zirnbauer. Nonstandard symmetry classes in mesoscopic normal-superconducting hybrid structures. *Phys. Rev. B*, 55(2):1142–1161, Jan 1997.

[289] Élie Cartan. Sur une classe remarquable d'espaces de Riemann. *Bulletin de la Société mathématique de France*, 54:214–264, 1926.

[290] Sigurdur Helgason. *Differential geometry, Lie groups, and symmetric spaces*. Academic press, 1979.

[291] Andreas P. Schnyder, Shinsei Ryu, Akira Furusaki, and Andreas W. W. Ludwig. Classification of topological insulators and superconductors in three spatial dimensions. *Phys. Rev. B*, 78(19):195125, Nov 2008.

[292] Shinsei Ryu, Andreas P Schnyder, Akira Furusaki, and Andreas WW Ludwig. Topological insulators and superconductors: tenfold way and dimensional hierarchy. *New Journal of Physics*, 12(6):065010, 6 2010.

[293] Maissam Barkeshli, Chao-Ming Jian, and Xiao-Liang Qi. Classification of topological defects in Abelian topological states. *Phys. Rev. B*, 88(24):241103, Dec 2013.

[294] Zohar Nussinov and Gerardo Ortiz. Bond algebras and exact solvability of Hamiltonians: Spin $s = \frac{1}{2}$ multilayer systems. *Phys. Rev. B*, 79:214440, Jun 2009.

[295] E. Cobanera, G. Ortiz, and Z. Nussinov. Unified approach to quantum and classical dualities. *Phys. Rev. Lett.*, 104:020402, Jan 2010.

[296] Gerardo Ortiz Emilio Cobanera and Zohar Nussinov. The bond-algebraic approach to dualities. *Advances in Physics*, 60(5):679–798, 2011.

[297] Sanjay Moudgalya and Olexei I. Motrunich. Hilbert space fragmentation and commutant algebras. *Phys. Rev. X*, 12:011050, Mar 2022.

[298] Sanjay Moudgalya and Olexei I. Motrunich. From symmetries to commutant algebras in standard Hamiltonians. *Annals of Physics*, 455:169384, 2023.

[299] Francesca Albertini and Domenico D'Alessandro. The Lie algebra structure and controllability of spin systems. *Linear algebra and its applications*, 350(1-3):213–235, 7 2002.

[300] Francesca Albertini and Domenico D'Alessandro. Subspace controllability of multipartite spin networks. *Systems & Control Letters*, 151:104913, 5 2021.

[301] Jiahui Chen, Hui Zhou, Changkui Duan, and Xinhua Peng. Preparing Greenberger-Horne-Zeilinger and W states on a long-range Ising spin model by global controls. *Phys. Rev. A*, 95(3):032340, 3 2017.

[302] Xiaoting Wang, Daniel Burgarth, and S Schirmer. Subspace controllability of spin-1 2 chains with symmetries. *Phys. Rev. A*, 94(5):052319, 11 2016.

[303] D'Alessandro, Domenico and Hartwig, Jonas T. Dynamical decomposition of bilinear control systems subject to symmetries. *Journal of Dynamical and Control Systems*, 27(1):1–30, 1 2021.

[304] É. Cartan. Les groupes réels simples, finis et continus. *Ann. Sci. Éc. Norm. Supér. (3)*, 31:263–356, 1914.

[305] Evgenii Borisovich Dynkin. Maximal subgroups of classical groups. *Uspekhi Matematicheskikh Nauk*, 7(6):226–229, 1952.

[306] Jacques Tits. Sous–algèbres des algèbres de Lie semi–simples (d'après V. Morozov, A. Malcev, E. Dynkin et F. Karpelevitch). In *Séminaire Bourbaki, 2*, pages 1–18. Secrétariat mathématique, 1959.

[307] Xie Chen, Zheng-Cheng Gu, and Xiao-Gang Wen. Classification of gapped symmetric phases in one-dimensional spin systems. *Phys. Rev. B*, 83(3):035107, 1 2011.

[308] Marius De Leeuw, Anton Pribytok, and Paul Ryan. Classifying integrable spin-1/2 chains with nearest neighbour interactions. *Journal of Physics A: Mathematical and Theoretical*, 52(50):505201, 11 2019.

[309] Nick G Jones and Noah Linden. Integrable spin chains and the Clifford group. *Journal of Mathematical Physics*, 63(10), 10 2022.

[310] SG Schirmer, ICH Pullen, and AI Solomon. Identification of dynamical Lie algebras for finite-level quantum control systems. *Journal of Physics A: Mathematical and General*, 35(9):2327, 3 2002.

[311] Xiaoting Wang, Peter Pemberton-Ross, and Sophie G Schirmer. Symmetry and subspace controllability for spin networks with a single-node control. *IEEE transactions on automatic control*, 57(8):1945–1956, 8 2012.

[312] Efekan Kökcü, Daan Camps, Lindsay Bassman Oftelie, Wibe A de Jong, Roel Van Beeumen, and AF Kemper. Algebraic Compression of Free Fermionic Quantum Circuits: Particle Creation, Arbitrary Lattices and Controlled Evolution. *arXiv preprint arXiv:2303.09538*, 3 2023.

[313] Daan Camps, Efekan Kökcü, Lindsay Bassman Oftelie, Wibe A De Jong, Alexander F Kemper, and Roel Van Beeumen. An algebraic quantum circuit compression algorithm for hamiltonian simulation. *SIAM Journal on Matrix Analysis and Applications*, 43(3):1084–1108, 7 2022.

[314] Efekan Kökcü, **Roeland Wiersema**, Alexander F Kemper, and Bojko N Bakalov. Classification of dynamical Lie algebras generated by spin systems on undirected graphs. *(in preparation)*, 2024.

[315] Claudio Procesi. *Lie groups: an approach through invariants and representations*, volume 115. Springer, 2007.

[316] Seth Lloyd. Almost Any Quantum Logic Gate is Universal. *Phys. Rev. Lett.*, 75(2):346–349, Jul 1995.

[317] David Elieser Deutsch, Adriano Barenco, and Artur Ekert. Universality in quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 449(1937):669–677, 6 1995.

[318] Alexandre Choquette, Agustin Di Paolo, Panagiotis Kl Barkoutsos, David Sénéchal, Ivano Tavernelli, and Alexandre Blais. Quantum-optimal-control-inspired ansatz for variational quantum algorithms. *Phys. Rev. Res.*, 3(2):023092, 5 2021.

[319] Gabriel Matos, Chris N Self, Zlatko Papić, Konstantinos Meichanetzidis, and Henrik Dreyer. Characterization of variational quantum algorithms using free fermions. *Quantum*, 7:966, 3 2023.

[320] Pierre-Luc Dallaire-Demers, Jonathan Romero, Libor Veis, Sukin Sim, and Alán Aspuru-Guzik. Low-depth circuit ansatz for preparing correlated fermionic states on a quantum computer. *Quantum Science and Technology*, 4(4):045005, 9 2019.

[321] Abhinav Anand, Philipp Schleich, Sumner Alperin-Lea, Phillip WK Jensen, Sukin Sim, Manuel Díaz-Tinoco, Jakob S Kottmann, Matthias Degroote, Artur F Izmaylov, and Alán Aspuru-Guzik. A quantum computing view on unitary coupled cluster theory. *Chemical Society Reviews*, 51(5):1659–1684, 2022.

[322] 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.

[323] Conor Mc Keever and Michael Lubasch. Classically optimized Hamiltonian simulation. *Phys. Rev. Res.*, 5:023146, Jun 2023.

[324] John S. Van Dyke, Karunya Shirali, George S. Barron, Nicholas J. Mayhall, Edwin Barnes, and Sophia E. Economou. Scaling adaptive quantum simulation algorithms via operator pool tiling. *Phys. Rev. Res.*, 6:L012030, Feb 2024.

[325] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J. S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, K. M. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Ducore, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim. Benchmarking an 11-qubit quantum computer. *Nature Communications*, 10(1):5464, 11 2019.

[326] Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko. Quantum Boltzmann Machine. *Phys. Rev. X*, 8(2):021050, May 2018.

[327] Hilbert J Kappen. Learning quantum models from quantum or classical data. *Journal of Physics A: Mathematical and Theoretical*, 53(21):214001, 5 2020.

[328] David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Phys. Rev. Lett.*, 35(26):1792, 12 1975.

[329] Louis Schatzki, Martín Larocca, Quynh T. Nguyen, Frédéric Sauvage, and M. Cerezo. Theoretical guarantees for permutation-equivariant quantum neural networks. *npj Quantum Information*, 10(1):12, 2024.

[330] Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Phys. Rev. A*, 80(1):012304, 7 2009.

[331] Michael Ragone, Bojko N. Bakalov, Fr/'ed/'eric Sauvage, Alexander F. Kemper, Carlos Ortiz Marrero, Martín Larocca, and M. Cerezo. A Lie algebraic theory of barren plateaus for deep parametrized quantum circuits. *arXiv preprint arXiv:2309.09342, to appear in Nature Comm.*, 2023.

[332] Rolando Somma, Gerardo Ortiz, Howard Barnum, Emanuel Knill, and Lorenza Viola. Nature and measure of entanglement in quantum phase transitions. *Phys. Rev. A*, 70(4):042311, 10 2004.

[333] Rolando D Somma. Quantum computation, complexity, and many-body physics. *arXiv preprint quant-ph/0512209*, 12 2005.

[334] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[335] Antonio A. Mele, Glen B. Mbeng, Giuseppe E. Santoro, Mario Collura, and Pietro Torta. Avoiding barren plateaus via transferability of smooth solutions in a Hamiltonian variational ansatz. *Phys. Rev. A*, 106(6):L060401, Dec 2022.

[336] Tudor Ionescu. On the generators of semisimple Lie algebras. *Linear Algebra and Its Applications*, 15(3):271–292, 1976.

[337] Norbert Schuch, J. Ignacio Cirac, and Michael M. Wolf. Quantum States on Harmonic Lattices. *Communications in Mathematical Physics*, 267(1):65–92, 7 2006.

[338] Christina V Kraus, Michael M Wolf, and J Ignacio Cirac. Quantum simulations under translational symmetry. *Phys. Rev. A*, 75(2):022303, 2 2007.

[339] Pedro de M Rios and Eldar Straume. *Symbol Correspondences for Spin Systems*. Springer, 2 2014.

[340] John B Parkinson and Damian JJ Farnell. *An introduction to quantum spin systems*, volume 816. Springer, 8 2010.

[341] Rolando Somma, Howard Barnum, Gerardo Ortiz, and Emanuel Knill. Efficient solvability of hamiltonians and limits on the power of some quantum computational models. *Phys. Rev. Lett.*, 97:190501, Nov 2006.

[342] Shouzhen Gu, Rolanfdo D Somma, and Burak Şahinoğlu. Fast-forwarding quantum evolution. *Quantum*, 5:577, 11 2021.

[343] Jean-Sébastien Caux and Jorn Mossel. Remarks on the notion of quantum integrability. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(02):P02023, 2 2011.

[344] Shayan Majidy, William F. Braasch, Aleksander Lasek, Twesh Upadhyaya, Amir Kalev, and Nicole Yunger Halpern. Noncommuting conserved charges in quantum thermodynamics and beyond. *Nature Reviews Physics*, 5(11):689–698, 2023.

[345] Sanjay Moudgalya and Olexei I Motrunich. Exhaustive characterization of quantum many-body scars using commutant algebras. *arXiv preprint arXiv:2209.03377*, 2022.

[346] Shayan Majidy, Aleksander Lasek, David A. Huse, and Nicole Yunger Halpern. Non-Abelian symmetry can increase entanglement entropy. *Phys. Rev. B*, 107(4):045102, Jan 2023.

[347] Chaitanya Murthy, Arman Babakhani, Fernando Iniguez, Mark Srednicki, and Nicole Yunger Halpern. Non-Abelian Eigenstate Thermalization Hypothesis. *Phys. Rev. Lett.*, 130(14):140402, Apr 2023.

[348] Iman Marvian. (Non-) Universality in symmetric quantum circuits: Why Abelian symmetries are special. *arXiv preprint arXiv:2302.12466*, 2 2023.

[349] Victor Galitski. Quantum-to-classical correspondence and Hubbard-Stratonovich dynamical systems: A Lie-algebraic approach. *Phys. Rev. A*, 84:012118, Jul 2011.

[350] Yuan Yao, Filippo Miatto, and Nicolás Quesada. The recursive representation of Gaussian quantum mechanics. *arXiv preprint arXiv:2209.06069*, 9 2022.

[351] Narayanan Rengaswamy, Robert Calderbank, Henry D Pfister, and Swanand Kadhe. Synthesis of logical Clifford operators via symplectic geometry. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 791–795. IEEE, IEEE, 6 2018.

[352] Zoltán Zimborás, Robert Zeier, Michael Keyl, and Thomas Schulte-Herbrüggen. A dynamic systems approach to fermions and their relation to spins. *EPJ Quantum Technology*, 1(1):11, 9 2014.

[353] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[354] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022. Survey Certification.

[355] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[356] Avrim L. Blum and Ronald L. Rivest. Training a 3-node neural network is np-complete. *Neural Networks*, 5(1):117–127, 1992.

[357] Surbhi Goel, Adam Klivans, Pasin Manurangsi, and Daniel Reichman. Tight Hardness Results for Training Depth-2 ReLU Networks. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:14, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[358] Vincent Froese and Christoph Hertrich. Training neural networks is NP-hard in fixed dimension. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[359] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 3 2010. PMLR.

[360] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[361] Steven R. White. Spin gaps in a frustrated heisenberg model for cav_4O_9. *Phys. Rev. Lett.*, 77:3633–3636, Oct 1996.

[362] **Roeland Wiersema**. Zyglrox. `https://github.com/therooler/zyglrox`, 2021.

[363] M. E. J. Newman and G. T. Barkema. *Monte Carlo methods in statistical physics*. Clarendon Press, Oxford, 2 1999.

[364] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, 01 1965.

[365] Carlton M. Caves. Quantum Error Correction and Reversible Operations. *Journal of Superconductivity*, 12(6):707–718, Dec 1999.

[366] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac. Matrix Product Density Operators: Simulation of Finite-Temperature and Dissipative Systems. *Phys. Rev. Lett.*, 93(20):207204, Nov 2004.

[367] R. Blankenbecler, D. J. Scalapino, and R. L. Sugar. Monte carlo calculations of coupled boson-fermion systems. i. *Phys. Rev. D*, 24:2278–2286, Oct 1981.

[368] Benjamin Russell and Susan Stepney. Geometric Methods for Analysing Quantum Speed Limits: Time-Dependent Controlled Quantum Systems with Constrained Control Functions. In Giancarlo Mauri, Alberto Dennunzio, Luca Manzoni, and Antonio E. Porreca, editors, *Unconventional Computation and Natural Computation*, volume 7956 of *Lecture Notes in Computer Science*, pages 198–208, Berlin, Heidelberg, 7 2013. Springer.

[369] James E Humphreys. *Introduction to Lie algebras and representation theory*, volume 9. Springer Science & Business Media, 2012.

[370] Shalin Parekh. The KPZ limit of ASEP with boundary. *Communications in Mathematical Physics*, 365:569–649, 1 2019.

[371] Rodney J Baxter. *Exactly solved models in statistical mechanics*. Elsevier, 1982.

[372] Adrian Chapman and Steven T Flammia. Characterization of solvable spin models via graph invariants. *Quantum*, 4:278, 6 2020.

[373] Yosi Atia and Dorit Aharonov. Fast-forwarding of Hamiltonians and exponentially precise measurements. *Nature communications*, 8(1):1–9, 11 2017.

[374] Lindsay Bassman Oftelie, Roel Van Beeumen, Ed Younis, Ethan Smith, Costin Iancu, and Wibe A. de Jong. Constant-depth circuits for dynamic simulations of materials on quantum computers. *Materials Theory*, 6(1):13, 3 2022.

APPENDICES

Appendix A

Quantum computing preliminaries

A.1 Pauli strings and $\mathfrak{su}(2^n)$

Length-n Pauli strings, when multiplied with the imaginary unit i, form a natural basis for the Lie algebra $\mathfrak{u}(2^n)$ of skew-Hermitian matrices.

In this section, we review the notation and basic properties of Pauli strings.

Throughout the paper, we work with the *Pauli matrices*

$$\sigma^0 = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \sigma^1 = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma^2 = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma^3 = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

including the identity matrix I, which form a basis for the real vector space of 2×2 Hermitian matrices. We will denote by A^T the transpose of a matrix, and by $A^\dagger$ its Hermitian conjugate (which is obtained from A^T by taking complex conjugates of all entries). Thus, $A^\dagger = A$ for all $A \in \mathcal{P}_1 := \{I, X, Y, Z\}$. On the other hand, we have

$$Y^T = -Y, \qquad A^T = A \quad \text{for} \quad A = I, X, Z.$$

Fix a positive number n. Length-n *Pauli strings* are tensor products of n Pauli matrices of the form

$$a = A^1 \otimes A^2 \otimes \cdots \otimes A^n, \qquad A^j \in \mathcal{P}_1. \tag{A.1}$$

We denote the set of all such Pauli strings by $\mathcal{P}_n := \{I, X, Y, Z\}^{\otimes n}$. Every $a \in \mathcal{P}_n$ is a linear operator on the Hilbert space $(\mathbb{C}^2)^{\otimes n}$ of n qubits, so a can be represented as a matrix

of size $2^n \times 2^n$ (by the Kronecker product). In particular, $I^{\otimes n}$ is the $2^n \times 2^n$ identity matrix. The Hermitian conjugate and transpose of a Pauli string are done component-wise:

$$a^\dagger = (A^1)^\dagger \otimes (A^2)^\dagger \otimes \cdots \otimes (A^n)^\dagger = a,$$
$$a^T = (A^1)^T \otimes (A^2)^T \otimes \cdots \otimes (A^n)^T = (-1)^{\#\{A^j = Y\}} a.$$

All Pauli strings are Hermitian, and $\mathcal{P}_n$ is a basis (over $\mathbb{R}$) of the vector space of $2^n \times 2^n$ Hermitian matrices.

To shorten the notation, we will often omit the tensor product signs in Pauli strings, so Equation (2.6) will be written as $a = A^1 A^2 \cdots A^n$. For example, we will write

$$XX = X \otimes X, \qquad XY = X \otimes Y, \qquad Z \cdots Z = Z^{\otimes n}, \quad \text{etc.}$$

For $A \in \mathcal{P}_1$ and $1 \le j \le n$, we will denote by

$$A_j := I^{\otimes(j-1)} \otimes A \otimes I^{\otimes(n-j)} \tag{A.2}$$

the linear operator A acting on the j-th qubit. For example, for $n = 3$,

$$X_1 = XII = X \otimes I \otimes I, \qquad Z_2 = IZI = I \otimes Z \otimes I, \qquad X_1 Z_2 Y_3 = XZY = X \otimes Z \otimes Y, \quad \text{etc.}$$

With this notation, we distinguish

$$A_1 A_2 \cdots A_n = AA \cdots A = A \otimes A \otimes \cdots \otimes A = A^{\otimes n}$$

from Equation (2.6), where in the latter the tensor factors $A^1, \ldots, A^n$ are allowed to be different.

When there is a danger to confuse the tensor product and the matrix product, we will use $\cdot$ for the product of matrices. We have:

$$X \cdot Y = iZ = -Y \cdot X, \qquad Y \cdot Z = iX = -Z \cdot Y, \qquad Z \cdot X = iY = -X \cdot Z,$$

and each Pauli matrix squares to the identity:

$$X \cdot X = Y \cdot Y = Z \cdot Z = I.$$

The matrix product of Pauli strings is done component-wise:

$$(A^1 \otimes \cdots \otimes A^n) \cdot (B^1 \otimes \cdots \otimes B^n) = (A^1 \cdot B^1) \otimes \cdots \otimes (A^n \cdot B^n).$$

From here, it is easy to deduce the following important properties of Pauli strings.

Lemma A.1.1. *For any $a, b \in \mathcal{P}_n$, we have $a \cdot a = I^{\otimes n}$ and $a \cdot b = \pm b \cdot a$. Hence, any two Pauli strings either commute or anticommute.*

Notice that the product of two Pauli strings is again a Pauli string, up to a multiple of $\pm 1, \pm i$. Thus, the set $\{\pm a, \pm ia \,|\, a \in \mathcal{P}_n\}$ is a group under the matrix product, called the *Pauli group*. The following corollary of Lemma A.1.1 will be useful in the future.

Corollary A.1.1. *For any $a, b \in \mathcal{P}_n$, if $[a, b] := a \cdot b - b \cdot a \neq 0$, then $[a, [a, b]] = 4b$.*

Proof. When $[a, b] \neq 0$, we have $[a, b] = 2a \cdot b$ and $[a, [a, b]] = 4a \cdot a \cdot b = 4b$. □

Another important consequence of Lemma A.1.1 is *Euler's formula*

$$e^{i\theta a} = (\cos\theta) I^{\otimes n} + i(\sin\theta) a, \qquad a \in \mathcal{P}_n, \;\; \theta \in \mathbb{R}. \tag{A.3}$$

A useful special case is $\theta = \pi/2$; then

$$e^{i\frac{\pi}{2}a} = ia, \qquad a \in \mathcal{P}_n, \tag{A.4}$$

which we also saw more generally defined in Equation (2.11) for idempotent matrices. Note that any $a \in \mathcal{P}_n$ is Hermitian (i.e., $a^\dagger = a$), ia is skew-Hermitian (i.e., $(ia)^\dagger = -ia$), and $U = e^{i\theta a}$ is unitary (i.e., $UU^\dagger = I^{\otimes n}$). We will also need the following corollary of Euler's formulas Equation (A.3), Equation (A.4).

Corollary A.1.2. *For any anticommuting $a, b \in \mathcal{P}_n$ and a real number θ, we have*

$$e^{i\theta a} \cdot b = b \cdot e^{-i\theta a}. \tag{A.5}$$

In particular,

$$e^{i\frac{\pi}{4}a}\, b\, e^{-i\frac{\pi}{4}a} = ia \cdot b. \tag{A.6}$$

A.2 Projective measurements

Measurable quantities in quantum physics are given by observables, which are represented by linear, self-adjoint operators acting on a complex Hilbert space $\mathcal{H}$. The spectral theorem states that a self-adjoint operator O can be decomposed into a linear combination of eigenspace projectors and eigenvalues (we restrict ourselves to the case where $\mathcal{H}$ is finite).

Let $\{|\psi_n\rangle\}$ be an eigenbasis of O such that $O|\psi_n\rangle = r_n|\psi_n\rangle$ with $r_n \in \mathbb{R}$. We can write down orthonormal projectors onto the eigenspace of O

$$\Pi_n = |\psi_n\rangle\langle\psi_n|,$$

so that

$$O = \sum_n r_n \Pi_n,$$

where we assume O to have a non-degenerate spectrum. Note that Π_n is idempotent, i.e. $\Pi^2 = \Pi$, and we have

$$\mathrm{Tr}\{\Pi_n \Pi_m\} = \delta_{mn}, \tag{A.7}$$

hence the projectors form an orthonormal basis.

Our interaction with a quantum system as observers come from the action of measurement. Formally, this means that if we want to measure observable O on a state $|\phi\rangle$, the probability of seeing eigenvalue

$$r_m = \mathrm{Tr}\{\Pi_m O\}, \tag{A.8}$$

given a quantum state ϕ on our detector is given by the Born rule

$$p(r_m) = \langle\phi|\Pi_m|\phi\rangle = |\langle\psi_m|\phi\rangle|^2,$$

or in terms of a density matrix

$$p(r_m) = \mathrm{Tr}\{\Pi\rho\}.$$

A projective measurement transforms a state as

$$|\psi\rangle \mapsto \frac{\Pi|\psi\rangle}{\langle\psi|\Pi|\psi\rangle},$$

or

$$\rho \mapsto \frac{\Pi\rho\Pi}{\mathrm{Tr}\{\Pi\rho\}},$$

Given a measurement, we can estimate the average value of an observable under the distribution $p(\boldsymbol{r})$

$$\begin{aligned}\langle O\rangle &= \langle\phi|O|\phi\rangle \\ &= \sum_n r_n \langle\phi|\Pi_n|\phi\rangle = \sum_n r_n p(r_n).\end{aligned}$$

We can also estimate the variance,

$$\mathrm{Var}[O] = \langle\psi| O^2 |\psi\rangle - \langle\psi| O |\psi\rangle^2 .$$

To estimate observables we define an estimator for the expectation value based on the measurements we obtain. The sample mean estimator

$$f(O) = \frac{1}{N_s}\sum_i^{N_s} r^{(i)}, \quad r^{(i)} \sim p(r),$$

is an unbiased estimator of $\mathbb{E}[X]$ with $p(r)$ the probability distribution over all eigenvalues of O.

On a quantum computer, we typically only have access to a specific set of measurements, which are given by the projectors $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ onto the eigenvectors of the Pauli $Z \in \mathcal{P}_1$ operator, with eigenvalues $r_n = \pm 1$ respectively. A finite-size estimator would then be

$$\langle Z_1\rangle \approx \frac{1}{N_s}\sum_i^{N_s} r^{(i)},$$

with $p(r) = \{|\langle\phi|0\rangle|^2, |\langle\phi|1\rangle|^2\}$.

Multi-qubit observables are often described with tensor products of operators, such as Pauli strings (see Equation (A.2)). Consider $\langle Z_1 Z_2\rangle$. We can decompose the operator $Z_1 Z_2$ as

$$\begin{aligned} Z_1 \otimes Z_2 &= \left(\sum_n r_n \Pi_n\right) \otimes \left(\sum_m r'_m \Pi_m\right) \\ &= \sum_{n,m} r_n r'_m \Pi_n \otimes \Pi_m, \end{aligned}$$

which gives for the expectation value

$$\langle Z_1 Z_2\rangle = \sum_{n,m} r_n r'_m \langle\phi| \Pi_n \otimes \Pi_m |\phi\rangle = \sum_{n,m} r_n r'_m p(r_n, r_m).$$

Hence, the final estimator is an estimator for the product of eigenvalues $r_n r_m$

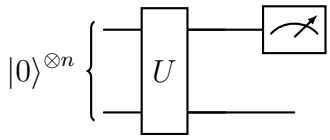
$$\langle Z_1 Z_2\rangle \approx \frac{1}{N_s}\sum_i^{N_s} r^{(i)} r'^{(i)} \cdot m_{j,2}.$$

If we look at the variance of this estimator

$$\begin{aligned}\mathrm{Var}(Z_1Z_2) &= \langle\psi|\,(Z_1Z_2)^2\,|\psi\rangle - \langle\psi|\,Z_1Z_2\,|\psi\rangle^2 \\ &= 1 - \langle Z_1Z_2\rangle^2\,,\end{aligned}$$

because $(Z_1Z_2)^2 = \mathrm{I}$ and $\langle\psi|\psi\rangle = 1$.

In order to measure in a basis different from the computational basis, we need to appropriately transform the states prior to measurement to gather statistics. For measuring Z_1 we do not need to do anything:



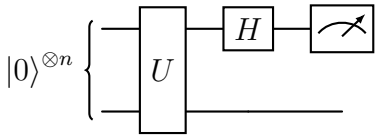
To measure X and Y, we need to transform the basis vectors of Z, to the eigenbases of X and Y, respectively. These are given by

$$\begin{aligned}v_1^X &= \left\{\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right\} \equiv \{|+\rangle\,, |-\rangle\} \\ v_1^Y &= \left(\frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle), \quad v_2^Y = \frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle)\right).\end{aligned}$$

We see that applying the Hadamard gate H to the basis of Z gives

$$\begin{aligned}H\,|0\rangle &= |+\rangle \\ H\,|1\rangle &= |-\rangle\,,\end{aligned}$$

which produces the eigenbasis vectors of X



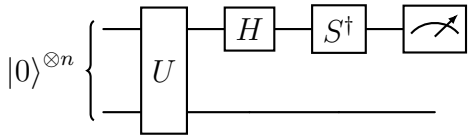
For Y, we need to apply a phase gate $S^\dagger$ in addition to Hadamard gate:

$$\begin{aligned}S^\dagger H\,|0\rangle &= S^\dagger\,|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle) \\ S^\dagger H\,|1\rangle &= S^\dagger\,|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle),\end{aligned}$$

where

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.$$

This gives the circuit



A.3 Informationally Complete POVMs

Determining what quantum state one has prepared on a device, a process known as quantum tomography, is a crucial task in building a functional quantum computer. In order to reconstruct a state, we need to measure it and one can use a particular set of measurements called positive operator valued measure to fully reconstruct a state.

A POVM is a set of operators $\{M_{\boldsymbol{a}}\}_{\boldsymbol{a}}$ with $M_{\boldsymbol{a}} \geq 0$ that satisfies the condition

$$\sum_{\boldsymbol{a}} M_{\boldsymbol{a}} = I.$$

A POVM is informationally complete if $\{M_{\boldsymbol{a}}\}_{\boldsymbol{a}}$ spans $\mathcal{L}(\mathcal{H}_{\mathcal{S}})$. As opposed to the projective measurements in Appendix A.2, the measurements $M_{\boldsymbol{a}}$ are not necessarily orthogonal,

$$\sum_{\boldsymbol{a},\boldsymbol{a}'} \mathrm{Tr}\left\{M_{\boldsymbol{a}} M_{\boldsymbol{a}'}\right\} \neq \delta_{\boldsymbol{a}\boldsymbol{a}'} I,$$

in contrast with Equation (A.7).

Let $\{M_{a_i}\}_{a_i}$ be a POVM that acts on a single-qubit Hilbert space. We can define a factorable POVM as a tensor product of single-qubit POVM elements as

$$M_{\boldsymbol{a}} = M_{a_1} \otimes \ldots \otimes M_{a_N},$$

for $\boldsymbol{a} := (a_1, \ldots a_N)$. Clearly, if all M_{a_i} are informationally complete, then so is $M_{\boldsymbol{a}}$. An example of an informationally complete POVM is the Pauli-6 POVM, which is defined as

$$\{M_{\boldsymbol{a}}\}_{\boldsymbol{a}}^{\text{Pauli-6}} := \bigcup_{i=x.y,z} \left\{\frac{1}{3}\left|\uparrow_i\right\rangle\!\left\langle\uparrow_i\right|, \frac{1}{3}\left|\downarrow_i\right\rangle\!\left\langle\downarrow_i\right|\right\},$$

where the vectors $|\uparrow_i\rangle, |\downarrow_i\rangle$ correspond to the eigenvectors of the Pauli operators with eigenvalue ± 1 respectively. We can implement this POVM by rotating to the Pauli basis with probability 1/3. For the $\{X, Y, Z\}$ Paulis this means applying the gates $\{H, HS, I\}$ where H is a Hadamard, S is a Z-phase and I the identity gate, respectively (see Appendix A.2). Measuring in the computational basis, then produces outcomes $\boldsymbol{a}$ according to the Pauli-6 POVM.

Appendix B

Additional details for the Hamiltonian Variational Ansatz

B.1 Computational Details

For the implementation of our quantum circuits, we use `Zyglrox` [362], a powerful TensorFlow-based quantum simulator. For the classical optimization process, we use Adam (adaptive moment estimation) [334], a gradient descent-based optimizer, which is widely used in the machine learning community. Compared to vanilla gradient descent and its other variants, Adam updates the learning rates adaptively on a per-parameter basis by using estimates of the first and second moments of the gradients. In our own investigation for solving the ground energy problem with HVA, Adam outperformed all the other optimizers available in TensorFlow, with respect to fidelity and convergence times.

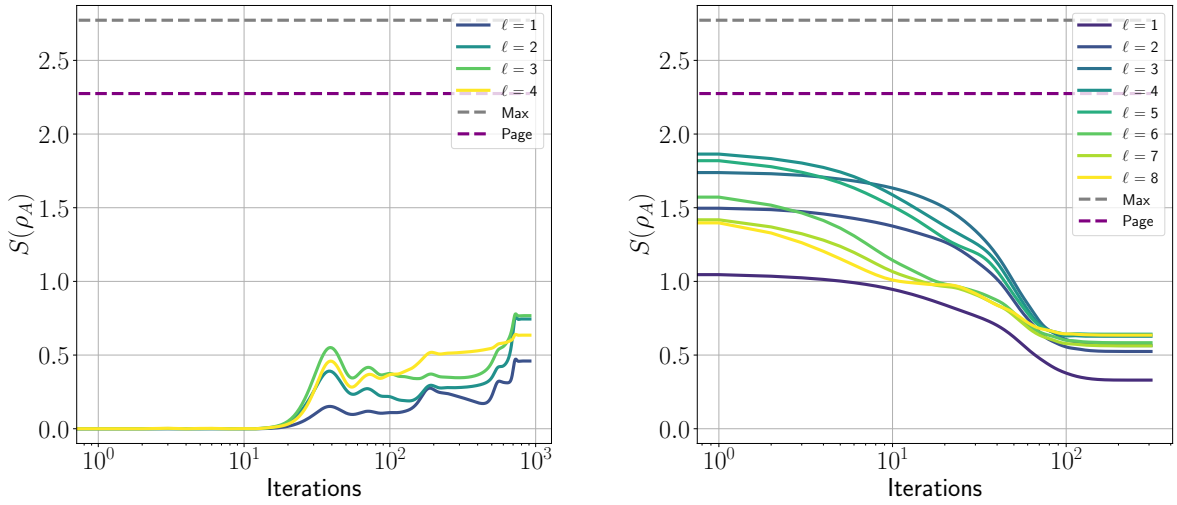
Unless stated otherwise, the stopping criterion for our optimization is defined as

$$|E(\boldsymbol{\theta}_t) - E(\boldsymbol{\theta}_{t+1})| < 10^{-13},$$

where t is the iteration number. The maximum number of iterations is set to 15000. We use an initial learning rate $r = 0.01$ for Adam which gives reasonably consistent results across all the models. Through our own investigation into initial Adam learning rates, we found a learning rate $10^{-3} \leq r \leq 4 \times 10^{-2}$ to be a good choice for the optimization for both the TFIM and the XXZ models, as it balances optimization accuracy and convergence speed.

B.2 Dynamics of Entanglement Entropy during Optimization

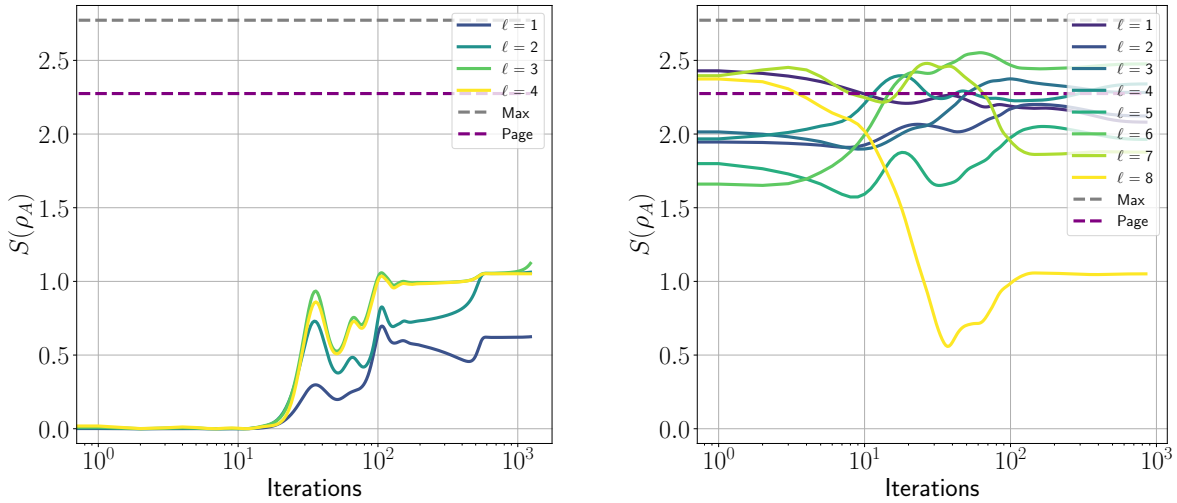
To further elucidate the difference in initialization strategies we qualitatively study the dynamics of the entanglement entropy during optimization. In Figures B.1 and B.2, we calculate the entanglement entropy of ρ_A at each layer of the circuit during the optimization. Although not much can be said about the intermediate states for the random state initialization, except that they are highly entangled, the entanglement entropy dynamics for the identity initialization have a distinct structure that is consistent as we increase the system size. In Figure B.3, we compare the scaling of the entanglement entropy for the identity start halfway through the circuit for different system sizes.



(a) TFIM $L = 4$

(b) TFIM $L = 4$

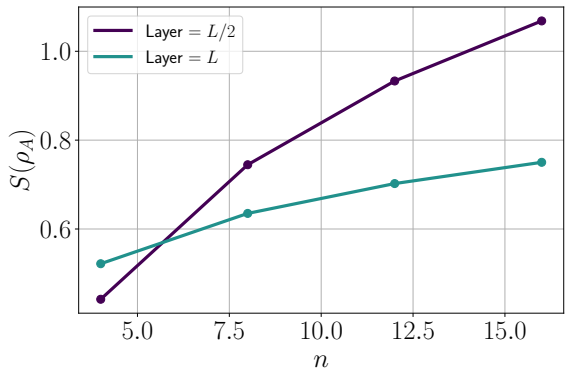
Figure B.1: **Dynamics of entanglement entropy at each layer during optimization (TFIM).** Each separate line indicates the entanglement entropy of the state in layer l. The gray dashed line denotes the maximum possible entanglement and the purple line gives the Page entropy. For all figures, the final state is a $> 99.9\%$ fidelity state. (a) Identity initialization for an 8-qubit TFIM with $g = 1.0$ and $L = 4$. (b) Same TFIM with a random-state initialization and overparameterization $L = 8$.



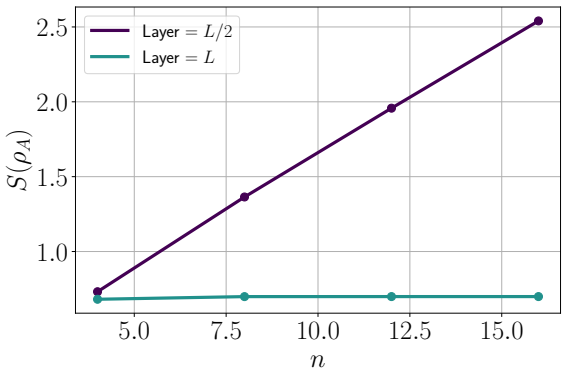
(a) XXZ model $L = 8$

(b) XXZ model $L = 8$

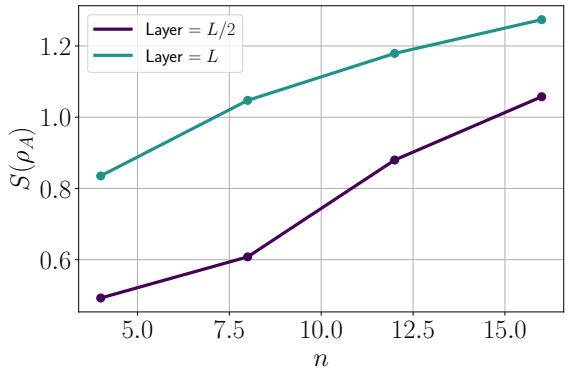
Figure B.2: **Dynamics of entanglement entropy at each layer during optimization (XXZ).** Each separate line indicates the entanglement entropy of the state in layer l. The gray dashed line denotes the maximum possible entanglement and the purple line gives the Page entropy. For all figures, the final state is a $> 99.9\%$ fidelity state. (a) Random-state initialization for an 8-qubit XXZ model with $\Delta = 1.0$ and $L = 4$. (b) Typical XXZ model dynamics for a random-state initialization and overparameterization $L = 8$.



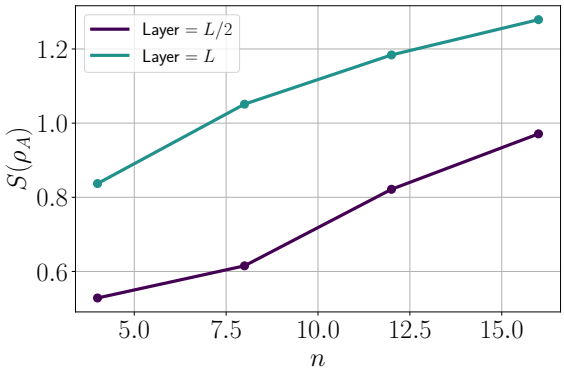
(a) TFIM $g = 1.0$



(b) TFIM $g = 0.5$



(c) XXZ model $\Delta = 0.5$



(d) XXZ model $\Delta = 1.0$

Figure B.3: **Scaling of the entanglement entropy of the converged state after $L/2$ and L layers.** (a) For the TFIM at the critical point, the ground state entanglement entropy has a logarithmic correction with increasing n. The entanglement halfway through the circuit is larger than in the final layer. (b) For a non-critical point, the ground state entanglement entropy is constant, but the entanglement entropy halfway through the circuit scales linearly with system size. (c)-(d) For the XXZ model, in addition to the logarithmic scaling of the entanglement entropy, the final layer entanglement is consistently higher than in the $L/2$ depth layer.

Appendix C

Additional details for measurement-induced entanglement phase transitions in quantum gradients

C.1 Finite-scaling analysis and data collapse

The correlation length ξ of a system quantifies the length scale over which parts of a system are correlated. When a system undergoes a continuous phase transition, the correlation length diverges. Phase transitions only occur in the thermodynamic limit, and hence simulations of finite-sized systems will contain artifacts that have to be accounted for in order to capture the correct behavior [363]. In particular, for a finite system the correlation length ξ cannot become infinite and is cut off at l^d, the volume of a finite d-dimensional hypercube with edges of length l. To account for this effect, we can perform a finite-scaling analysis.

The entanglement entropy as a function of measurement rate is conjectured to follow a volume law for $p < p_c$, a constant plus logarithmic correction at $p = p_c$ and area law for $p > p_c$ [85, 86, 89]. We can therefore construct a scaling form of the entanglement entropy as

$$S(n, p, \nu) = S(n, p_c, \nu) + f(n^{1/\nu}(p - p_c)), \tag{C.1}$$

where $S(n, p, \nu)$ denotes the von Neumann entropy at measurement rate p and f is a scaling function. The critical exponent ν determines the scaling of the entanglement entropy near p_c. If this scaling form is correct, we should be able to account for finite-size effects and all the data can be appropriately rescaled to match a single curve representing f with a proper choice of ν.

To determine the critical exponents, we fit a 5th-degree polynomial g to our data using a Nelder-Mead optimization [364] and minimize the χ^2-statistic

$$\chi^2 = \sum_i \frac{(S(n_i, p_i, \nu) - \tilde{S}(n_i, p_i, \nu))^2}{\Delta S}.$$

Here, $\tilde{S}(n_i, p_i, \nu)$ is estimated from the data and $S(n_i, p_i, \nu)$ is the proposed scaling form from Equation (C.1). ΔS is the standard deviation of the von Neumann entropies which arises due to the fluctuations induced by the randomized measurements and their outcomes. From the unscaled data, we determine a set of potential critical points p_c and fit the above χ^2-statistic to determine ν. We then report the values of p_c and ν that provided the best fit.

To verify the stability of the fit, we perform a statistical bootstrapping procedure to estimate the error bars on the fitted critical exponent ν. We take $K_{\text{boot}} = 100$, where each data set consists of K samples obtained by sampling from the entire data set of 3×10^3 data points with replacement. The final obtained error bars on ν are ≈ 0.01.

We can extrapolate our result to the thermodynamic limit by fitting the data for $n' = n_{\max}/2$ to $n' = n_{\max}$ and plotting the resulting values for ν against $1/n'$ [85]. By doing a linear fit on the resulting data, we obtain

$$\tilde{\nu}(n') = a\frac{1}{n'} + b,$$

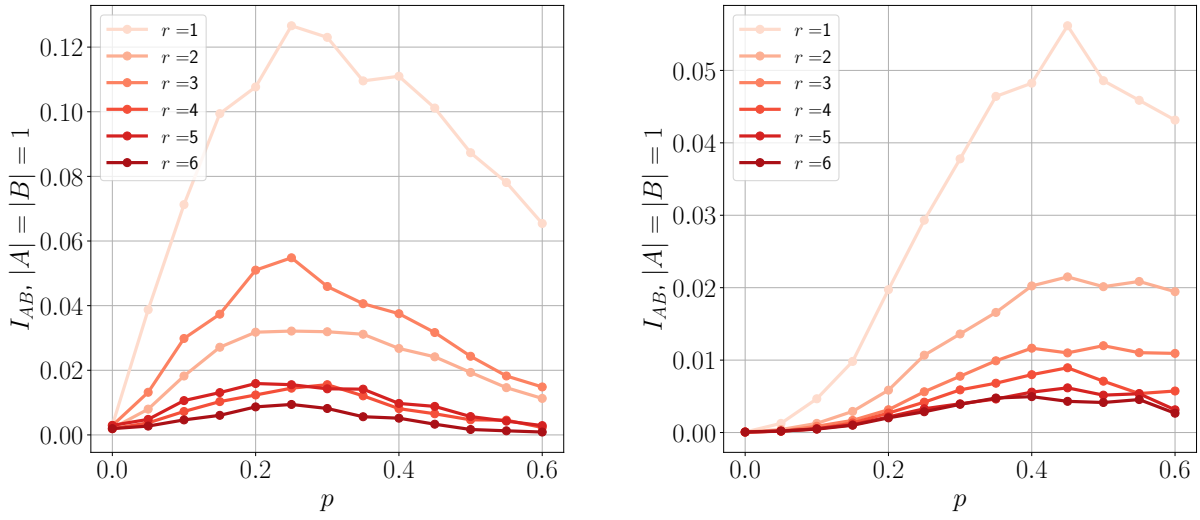
and so the intercept b corresponds to the value of ν in the thermodynamic limit, since $\lim_{n' \to \infty} 1/n' = 0$. When fitting the data, we weigh the errors by the standard errors obtained in the statistical bootstrap described above.

C.2 Mutual information

The quantum mutual information can be used to quantify subsystem correlations, and subsequently detect phase transitions since we expect correlations to divergence at criticality [85, 86, 88]. As additional confirmation that the critical values p_c estimated from the prior analysis are correct, we calculate the quantum mutual information as,

$$I(A, B) = S_A(n, p) + S_B(n, p) - S_{A\cup B}(n, p).$$

Here, we take the same approach as in [86], and take A and B to be two single qubit subsystems $|A| = 1$ and $|B| = 1$. We then vary the distance r between qubit A and B, to determine the effect of the distance on the subsystem correlations. In Figure C.1, we observe two broad peaks around the previously found values $p_c \approx 0.25$ and $p_c \approx 0.5$ for the XXZ-HVA and HAA, respectively.



(a) XXZ-HVA (b) HAE

Figure C.1: **Quantum mutual information between two qubits A and B separated by a distance r on a chain of length** 16. The mutual information is averaged over 3×10^3 samples, where each sample corresponds to a random circuit realization, as described in the main text.

C.3 Projective gradients

We consider the parameterized quantum circuit of Equation (2.5),

$$|\psi(\boldsymbol{\theta})\rangle = \prod_{l=1}^{\overleftarrow{L}} U_l(\theta_l)\,|0\rangle\,,$$

where we take $\theta_l \in \mathbb{R}$ without loss of generality. Additionally, we consider the case where after each layer $U_l(\theta_k)$ we apply a projective measurement Π_l, where $\Pi_l = \Pi_{l,0} \otimes \ldots \otimes \Pi_{l,n}$ with $\Pi_{l,i} \in \{|0\rangle\langle 0|\,, |1\rangle\langle 1|\,, I\}$. We denote by ρ_L the state resulting from applying L projectors Π to the circuit. Similarly, we denote by p_L the probability of obtaining the state ρ_L (see Appendix A.2).

Consider an initial state $\rho_0 = |0\rangle\langle 0|^{\otimes n}$, to which we apply the unitary $U_1(\theta_1)$ followed by a projective measurement Π_1,

$$\begin{aligned}\rho_1(\theta_1) &= \frac{\Pi_1 U_1(\theta_1)\rho_0 U_1^\dagger(\theta_1)\Pi_1}{\mathrm{Tr}\left\{\Pi_1 U_1(\theta_1)\rho_0 U_1^\dagger(\theta_1)\Pi_1\right\}} \\ &= \frac{\Pi_1 U_1(\theta_1)\rho_0 U_1^\dagger(\theta_1)\Pi_1}{p_1(\theta_1)}.\end{aligned}$$

Next, we add a unitary and measurement,

$$\begin{aligned}\rho_2(\theta_1,\theta_2) &= \frac{\Pi_2 U_2(\theta_2)\rho_1(\theta_1) U_2^\dagger(\theta_2)\Pi_2}{\mathrm{Tr}\left\{\Pi_2 U_2(\theta_2)\rho_1(\theta_1) U_2^\dagger(\theta_2)\Pi_2\right\}} \\ &= \frac{\Pi_2 U_2(\theta_2)\Pi_1 U_1(\theta_1)\rho_0 U_1^\dagger(\theta_1)\Pi_1 U_2^\dagger(\theta_2)\Pi_2}{\mathrm{Tr}\left\{\Pi_2 U_2(\theta_2)\Pi_1 U_1(\theta_1)\rho_0 U_1^\dagger(\theta_1)\Pi_1 U_2^\dagger(\theta_2)\Pi_2\right\}} \times \frac{p_1(\theta_1)}{p_1(\theta_1)} \\ &= \frac{\Pi_2 U_2(\theta_2)\Pi_1 U_1(\theta_1)\rho_0 U_1^\dagger(\theta_1)\Pi_1 U_2^\dagger(\theta_2)\Pi_2}{p_2(\theta_1,\theta_2)}.\end{aligned}$$

Note how the normalization constant of $\rho_1(\theta_1)$ cancels. Generalizing this to L projectors, we get the general form

$$\begin{aligned}\rho_L(\theta_1,\ldots,\theta_L) &= \left(\prod_{l=1}^{\overleftarrow{L}} \Pi_l U_l(\theta_l)\right)\rho_0\left(\prod_{l=1}^{\overrightarrow{L}} U_l^\dagger(\theta_l)\Pi_l\right) p_L^{-1}(\theta_1,\ldots,\theta_L) \\ &= \tilde{\rho}_L(\theta_1,\ldots,\theta_L) p_L^{-1}(\theta_1,\ldots,\theta_L),\end{aligned}$$

where

$$\tilde{\rho}_L(\theta_1, \ldots, \theta_L) = \left(\prod_{l=1}^{\overleftarrow{L}} \Pi_l U_l(\theta_l) \right) \rho_0 \left(\prod_{l=1}^{\overrightarrow{L}} U_l^\dagger(\theta_l) \Pi_l \right),$$
$$p_L(\theta_1, \ldots, \theta_L) = \mathrm{Tr}\left\{ \tilde{\rho_L}(\theta_1, \ldots, \theta_L) \right\},$$

are the unnormalized state and its normalization constant, respectively. To simplify the notation, we will write $\boldsymbol{\theta} \equiv (\theta_1, \ldots, \theta_L)$.

We are interested in the derivative of the variational energy of Equation (2.9)

$$E(\boldsymbol{\theta}) = \mathrm{Tr}\left\{ \rho_L(\boldsymbol{\theta}) H \right\},$$

where H is a Hermitian operator. We write the state as the product of an unnormalized state and its normalization constant

$$\mathrm{Tr}\left\{ \rho_L(\boldsymbol{\theta}) H \right\} = \mathrm{Tr}\left\{ \tilde{\rho}_L(\boldsymbol{\theta}) p_L^{-1}(\boldsymbol{\theta}) H \right\}.$$

Hence, the derivative consists of two parts via the product rule

$$\partial_{\theta_k} E(\boldsymbol{\theta}) = \overbrace{\mathrm{Tr}\left\{ \partial_{\theta_k} \left(\tilde{\rho}_L(\boldsymbol{\theta}) \right) p_L^{-1}(\boldsymbol{\theta}) H \right\}}^{(i)} + \overbrace{\mathrm{Tr}\left\{ \tilde{\rho}_L(\boldsymbol{\theta}) \partial_{\theta_k} \left(p_L^{-1}(\boldsymbol{\theta}) \right) H \right\}}^{(ii)}.$$

(i) For the derivative of the unnormalized state, we get

$$\begin{aligned}
\mathrm{Tr}\left\{ \left(\partial_{\theta_k} \tilde{\rho}_L(\boldsymbol{\theta}) \right) H \right\} &= \langle 0 | \left(\prod_{l=1}^{\overrightarrow{L}} U_l^\dagger(\theta_l) \Pi_l \right) H \left(\prod_{l=k+1}^{\overleftarrow{L}} \Pi_l U_l(\theta_l) \right) \Pi_k \partial_{\theta_k} U_k(\theta_k) \left(\prod_{l=1}^{\overleftarrow{k-1}} \Pi_l U_l(\theta_l) \right) | 0 \rangle \\
&+ \langle 0 | \left(\prod_{l=1}^{\overrightarrow{k-1}} U_l^\dagger(\theta_l) \Pi_l \right) \partial_{\theta_k} U_k^\dagger(\theta_k) \Pi_k \left(\prod_{l=k+1}^{\overrightarrow{L}} U_l^\dagger(\theta_l) \Pi_l \right) H \left(\prod_{l=1}^{\overleftarrow{L}} \Pi_l U_l(\theta_l) \right) | 0 \rangle \\
&= \left\langle \tilde{\psi}_0 \right| U_k^\dagger(\theta_k) \tilde{H} \partial_{\theta_k} U_k(\theta_k) \left| \tilde{\psi}_0 \right\rangle + \left\langle \tilde{\psi}_0 \right| \partial_{\theta_k} U_k^\dagger(\theta_k) \tilde{H} U_k(\theta_k) \left| \tilde{\psi}_0 \right\rangle,
\end{aligned}$$

where

$$\left| \tilde{\psi}_0 \right\rangle = \left(\prod_{l=1}^{\overleftarrow{k-1}} \Pi_l U_l(\theta_l) \right) | 0 \rangle, \tag{C.2}$$

is an unnormalized state and

$$\tilde{H} = \Pi_k \left(\prod_{l=k+1}^{\overrightarrow{L}} U_l^\dagger(\theta_l)\Pi_l \right) H \left(\prod_{l=k+1}^{\overleftarrow{L}} \Pi_l U_l(\theta_l) \right) \Pi_k. \tag{C.3}$$

If $U(\theta_k)$ is generated by a Pauli string A (see Appendix A.1), then $\partial_{\theta_k} U_k(\theta_k) = -\frac{i}{2} A U_k(\theta_k)$ and so we can use the parameter-shift rule [30, 31]

$$-\frac{i}{2} \left\langle \tilde{\psi}_0 \right| \left[A, U_k^\dagger(\theta_k) \tilde{H} U_k(\theta_k) \right] \left| \tilde{\psi}_0 \right\rangle = \frac{1}{2} \left\langle \tilde{\psi}_0 \right| U^\dagger(\theta_k + \frac{\pi}{2}) \tilde{H} U(\theta_k + \frac{\pi}{2}) - U^\dagger(\theta_k - \frac{\pi}{2}) \tilde{H} U(\theta_k - \frac{\pi}{2}) \left| \tilde{\psi}_0 \right\rangle, \tag{C.4}$$

which we derived in Equation (2.12).

If the expectation values in Equation (C.4) were with respect to properly normalized states, then this would provide a strategy for measuring the projective gradient. Hence, we need to first normalize the state in order to be able to perform the gradient calculation on the device. The normalization constants for the plus and minus shifted circuits are given by

$$p_L^{\pm,k} \equiv \left\langle \tilde{\psi}_0 \right| U^\dagger(\theta_k \pm \frac{\pi}{2}) \tilde{\Pi} U(\theta_k \pm \frac{\pi}{2}) \left| \tilde{\psi}_0 \right\rangle, \tag{C.5}$$

where

$$\tilde{\Pi} = \Pi_k \left(\prod_{l=k+1}^{\overrightarrow{k-1}} U_l^\dagger(\theta_l)\Pi_l \right) U_L^\dagger(\theta_L)\Pi_L U_L(\theta_L) \left(\prod_{l=k+1}^{\overleftarrow{k-1}} \Pi_l U_l(\theta_l) \right) \Pi_k,$$

is obtained by setting $H \to I$ in Equation (C.3). If we multiply with the identity

$$\begin{aligned} \mathrm{Tr}\left\{ (\partial_{\theta_k} \tilde{\rho}_L(\boldsymbol{\theta}))\, H \right\} &= \frac{1}{2} \left\langle \tilde{\psi}_0 \right| \left(U^\dagger(\theta_k + \frac{\pi}{2}) \tilde{H} U(\theta_k + \frac{\pi}{2}) \times \frac{p_L^{+,k}}{p_L^{+,k}} \right. \\ &\quad \left. - U^\dagger(\theta_k - \frac{\pi}{2}) \tilde{H} U(\theta_k - \frac{\pi}{2}) \times \frac{p_L^{-,k}}{p_L^{-,k}} \right) \left| \tilde{\psi}_0 \right\rangle \qquad \text{(C.6)} \\ &= \frac{1}{2} \left(E(\boldsymbol{\theta})^{+,k} p_L^{+,k} - E(\boldsymbol{\theta})^{-,k} p_L^{-,k} \right). \qquad \text{(C.7)} \end{aligned}$$

Here, $E(\boldsymbol{\theta})^{\pm,k}$ is the expectation value of the observable H after the measurements $\{\Pi_1, \ldots, \Pi_L\}$ have been applied and parameter θ_k has been shifted by $\pm\pi/2$.

(ii) For the gradient of the inverse of the normalization constant, we get

$$\mathrm{Tr}\left\{\tilde{\rho}_L(\boldsymbol{\theta})\left(\partial_{\theta_k} p_L^{-1}(\boldsymbol{\theta})\right) H\right\} = -E(\boldsymbol{\theta}) p_L^{-1}(\boldsymbol{\theta}) \partial_{\theta_k} p_L(\boldsymbol{\theta}),$$

where we used the normalization constant to write $\mathrm{Tr}\left\{\tilde{\rho}_L(\boldsymbol{\theta}) p_L^{-1}(\boldsymbol{\theta}) H\right\} = E(\boldsymbol{\theta})$, the expectation value of H with respect to the measured circuit. The final step is to calculate $\partial_{\theta_k} p_L(\boldsymbol{\theta})$:

$$\begin{aligned}
\partial_{\theta_k} p_L(\boldsymbol{\theta}) &= \mathrm{Tr}\left\{\partial_{\theta_k} \tilde{\rho}_L(\boldsymbol{\theta})\right\} \\
&= \langle 0| \left(\prod_{l=1}^{\overrightarrow{L-1}} U_l^\dagger(\theta_l)\Pi_l\right) U_L^\dagger(\theta_L)\Pi_L U_L(\theta_L) \left(\prod_{l=k+1}^{\overleftarrow{L-1}} \Pi_l U_l(\theta_l)\right) \Pi_k \partial_{\theta_k} U_k(\theta_k) \\
&\times \left(\prod_{l=1}^{\overleftarrow{k-1}} \Pi_l U_l(\theta_l)\right) |0\rangle + \langle 0| \left(\prod_{l=1}^{\overrightarrow{k-1}} U_l^\dagger(\theta_l)\Pi_l\right) \partial_{\theta_k} U_k^\dagger(\theta_k)\Pi_k \left(\prod_{l=k+1}^{\overrightarrow{L-1}} U_l^\dagger(\theta_l)\Pi_l\right) \\
&\times U_L^\dagger(\theta_L)\Pi_L U_L(\theta_L) \left(\prod_{l=1}^{\overleftarrow{L-1}} \Pi_l U_l(\theta_l)\right) |0\rangle \\
&= \left\langle \tilde{\psi}_0 \right| U_k^\dagger(\theta_k)\tilde{\Pi}_L \partial_{\theta_k} U_k(\theta_k) \left|\tilde{\psi}_0\right\rangle + \left\langle \tilde{\psi}_0 \right| \partial_{\theta_k} U_k^\dagger(\theta_k)\tilde{\Pi}_L U_k(\theta_k) \left|\tilde{\psi}_0\right\rangle,
\end{aligned}$$

where

$$\tilde{\Pi}_L = \Pi_k \left(\prod_{l=k+1}^{\overrightarrow{k-1}} U_l^\dagger(\theta_l)\Pi_l\right) U_L^\dagger(\theta_L)\Pi_L U_L(\theta_L) \left(\prod_{l=k+1}^{\overleftarrow{k-1}} \Pi_l U_l(\theta_l)\right) \Pi_k,$$

and $\left|\tilde{\psi}_0\right\rangle$ is the same as in Equation (C.2). Again, we can apply the parameter-shift rule to obtain

$$\partial_{\theta_k} p_L(\boldsymbol{\theta}) = \frac{1}{2}\left(\left\langle \tilde{\psi}_0 \right| U^\dagger(\theta_k + \frac{\pi}{2})\tilde{\Pi}_L U(\theta_k + \frac{\pi}{2}) - U^\dagger(\theta_k - \frac{\pi}{2})\tilde{\Pi}_L U(\theta_k - \frac{\pi}{2}) \left|\tilde{\psi}_0\right\rangle\right).$$

But these expectation values are simply the normalization constants $p_L^{\pm,k}$ of Equation (C.5), hence the final result becomes

$$\mathrm{Tr}\left\{\tilde{\rho}_L(\boldsymbol{\theta})\left(\partial_{\theta_k} p_L^{-1}(\boldsymbol{\theta})\right) H\right\} = -E(\boldsymbol{\theta})\frac{1}{2}\left(\frac{p_L^{+,k}}{p_L} - \frac{p_L^{-,k}}{p_L}\right).$$

Combining (i) and (ii) we finally obtain the projective gradient:

$$\partial_{\theta_k} E(\boldsymbol{\theta}) = \frac{1}{2}\left(\left(E(\boldsymbol{\theta})^{+,k} - E(\boldsymbol{\theta})\right) \frac{p_L^{+,k}}{p_L} - \left(E(\boldsymbol{\theta})^{-,k} - E(\boldsymbol{\theta})\right) \frac{p_L^{-,k}}{p_L} \right).$$

To calculate these projective gradients and produce Figure 2.13 in the main text, we use the TensorFlow-based quantum simulator `Zyglrox` [362]. Note that in practice, estimating these gradients will be exponentially difficult due to the ratio $p_L^{-,k}/p_L$.

C.4 A practical optimization algorithm with projective measurements

Any state $\rho_L(\boldsymbol{\theta})$ is weighted by two probabilities: a classical and quantum probability. The former is the result of flipping a coin with probability p after each layer for each qubit, which results in a measurement configuration. The latter is the quantum probability (obtained via the Born rule) of measuring an outcome of the particular measurement configuration, which we've denoted by $p_L(\boldsymbol{\theta})$.

We can denote the classical probability of a measurement configuration in layer l with

$$s_l^{(\boldsymbol{c}_l)} = \prod_{j=1}^{n} p^{\mathbb{I}(c_{l,j}=0)} \, (1-p)^{\mathbb{I}(c_{l,j}=1)},$$

where $c_{l,j} = 0$ indicates that we perform a measurement and $c_{l,j} = 1$ indicates that we do not. The tuple $\boldsymbol{c}_l = (c_{l,1}, \ldots, c_{l,n})$ thus labels a measurement setting in layer l. The total probability over all layers is then given by the product of these individual layer-wise probabilities:

$$s(\boldsymbol{c}) = \prod_{l=1}^{L} s_l^{(\boldsymbol{c}_l)}.$$

The tuple $\boldsymbol{c} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L)$ then labels a possible measurement setting.

After choosing a measurement setting, we run the circuit and perform the measurements. This results in a set of outcomes $\boldsymbol{i} = (\boldsymbol{i}_1, \ldots, \boldsymbol{i}_L)$, where $\boldsymbol{i}_l = (i_{l,1}, \ldots, i_{l,n})$ indicates the outcomes per layer. The integer $i_{l,j} \in \{0, 1, 2\}$ with $j = 1, \ldots, n$ indicates the measurement of $|0\rangle\langle 0|$, $|1\rangle\langle 1|$ and the identity operator, respectively. We now explicitly denote with

$\rho_L(\boldsymbol{i}, \boldsymbol{c}, \boldsymbol{\theta})$ the state resulting from a particular measurement setting, and with $p_L(\boldsymbol{i}|\boldsymbol{\theta}; \boldsymbol{c})$ the probability of obtaining a particular outcome $\boldsymbol{i}$, given a measurement setting $\boldsymbol{c}$.

If we remix the resulting pure states $\rho_L(\boldsymbol{i}, \boldsymbol{c}, \boldsymbol{\theta})$ according to the classical probability $s(\boldsymbol{c})$ and quantum probability $p_L(\boldsymbol{i}|\boldsymbol{\theta}; \boldsymbol{c})$ into a single density matrix, we obtain

$$\rho = \sum_{\boldsymbol{i},\boldsymbol{c}} s(\boldsymbol{c}) p_L(\boldsymbol{i}|\boldsymbol{\theta}; \boldsymbol{c})(\boldsymbol{\theta}) \rho_L(\boldsymbol{i}, \boldsymbol{c}, \boldsymbol{\theta}).$$

We can calculate a variational energy with respect to this density matrix as

$$E_{\text{int}}(\boldsymbol{\theta}) = \sum_{\boldsymbol{i},\boldsymbol{c}} s(\boldsymbol{c}) p_L(\boldsymbol{i}|\boldsymbol{\theta}; \boldsymbol{c})(\boldsymbol{\theta}) \text{Tr}\left\{\rho_L(\boldsymbol{i}, \boldsymbol{c}, \boldsymbol{\theta}) H\right\}, \tag{C.8}$$

where H is a Hermitian operator. Clearly $E_{\text{ground}} \leq E_{\text{int}}(\boldsymbol{\theta})$. Calculating the gradient of Equation (C.8) involves calculating the gradient for all individual states in the mixture. Note that the mixture in Equation (C.8) can be written as a sum of unnormalized states

$$\begin{aligned} E_{\text{int}}(\boldsymbol{\theta}) &= \sum_{\boldsymbol{i},\boldsymbol{c}} s(\boldsymbol{c}) p_L(\boldsymbol{i}|\boldsymbol{\theta}; \boldsymbol{c})(\boldsymbol{\theta}) \text{Tr}\left\{\tilde{\rho}_L(\boldsymbol{i}, \boldsymbol{c}, \boldsymbol{\theta}) p_L^{-1}(\boldsymbol{i}|\boldsymbol{\theta}; \boldsymbol{c}) H\right\} \\ &= \sum_{\boldsymbol{i},\boldsymbol{c}} s(\boldsymbol{c}) \text{Tr}\left\{\tilde{\rho}(\boldsymbol{i}, \boldsymbol{c}, \boldsymbol{\theta}) H\right\}. \end{aligned}$$

From Equation (C.7) we then see immediately that the gradient of the mixed state is then given by

$$\text{Tr}\left\{(\partial_{\theta_k} \rho) H\right\} = \sum_{\boldsymbol{i},\boldsymbol{c}} \frac{s(\boldsymbol{c})}{2} \left(E(\boldsymbol{\theta})^{+,l} p_L^{+,l}(\boldsymbol{i}|\boldsymbol{\theta}; \boldsymbol{c}) - E(\boldsymbol{\theta})^{-,l} p_L^{-,l}(\boldsymbol{i}|\boldsymbol{\theta}; \boldsymbol{c}) \right). \tag{C.9}$$

Hence, the estimator for the gradient corresponds to the average expectation value over intermediate measurements done on parameter-shifted circuits weighted by $p_L^{\pm,l}(\boldsymbol{i}|\boldsymbol{\theta}; \boldsymbol{c})$ and the classical probability $s(\boldsymbol{c})$. Therefore, the projective gradients can be estimated by obtaining statistics from the measurements done on the parameter-shifted circuits. Given a number of shots N_s, the gradient of Equation (C.9) can be obtained with Algorithm 4.

Algorithm 4: Algorithm to obtain the gradient of Equation (C.9)

Input: ρ_0, $U(\boldsymbol{\theta})$,H, p, N_s, θ_k

$h^{+,k} \leftarrow 0$

$h^{-,k} \leftarrow 0$

for $n \in (1, \ldots, N_s)$ **do**

  Create measurement configuration **for** $l \in (1, \ldots, L)$ **do**

    for $j \in (1, \ldots, n)$ **do**

      $\boldsymbol{c}_{l,j} \sim \mathrm{Ber}(p)$

  $\theta_k \leftarrow \theta_k + \pi/2$

  Run $U(\boldsymbol{\theta})$ with measurement setting $\boldsymbol{c}$, obtain outcomes $\boldsymbol{i}$ and $\rho_L^{+,l}(\boldsymbol{i}|\boldsymbol{\theta};\boldsymbol{c})$.

  Measure H and obtain eigenvalue h

  $h^{+,k} \leftarrow h^{+,k} + h$

  $\theta_k \leftarrow \theta_k - \pi$

  Run $U(\boldsymbol{\theta})$ with measurement setting $\boldsymbol{c}$, obtain outcomes $\boldsymbol{i}'$ and $\rho_L^{-,k}(\boldsymbol{i}'|\boldsymbol{\theta};\boldsymbol{c})$.

  Measure H and obtain eigenvalue h'

  $h^{-,k} \leftarrow h^{-,k} + h'$

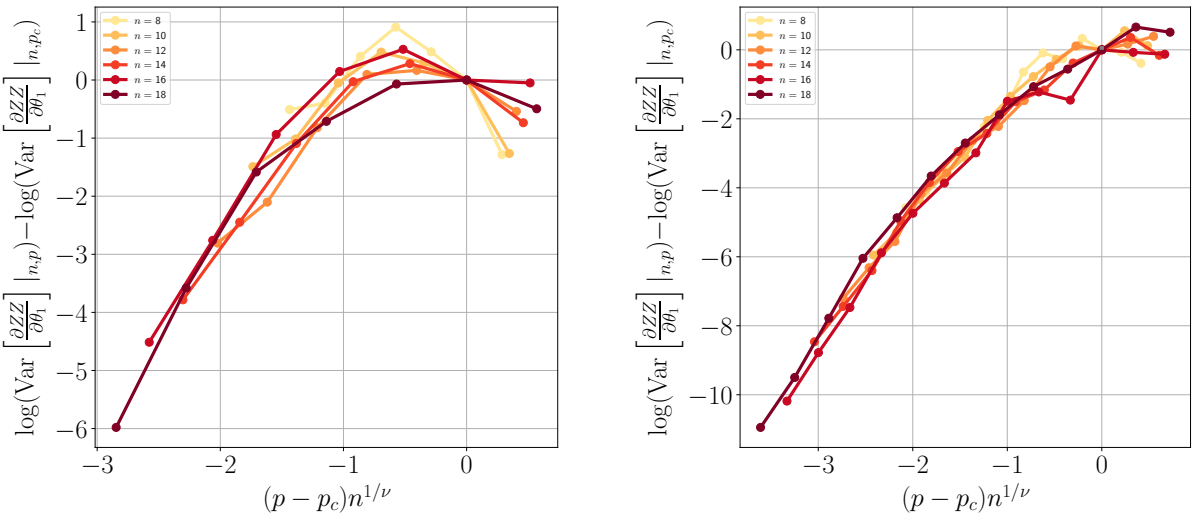
Output: $\frac{1}{2}(h^{+,k} - h^{-,k})$

C.5 Data collapse of the projective gradients

To observe the phase transition in the variance of the projective gradients of Figure C.2, we perform a data collapse of the following quantity:

$$\log\left(\mathrm{Var}\left[\frac{\partial ZZ}{\partial\theta_0}\right]\Big|_{n,p}\right) = \log\left(\mathrm{Var}\left[\frac{\partial ZZ}{\partial\theta_0}\right]\Big|_{n,p_c}\right) + g(n^{1/\nu}(p-p_c)).$$

We use the same method as in Appendix C.1. The resulting data collapse can be seen in Figure 2.13.



(a) XXZ-HVA

(b) HAE

Figure C.2: **Data collapse for the projective gradients at $p_c = 0.25$ and $p_c = 0.5$ for the XXZ-HVA and HAE circuits, respectively.** Since the data shown in Figure 2.13 is noisy, the data collapse is not as clean, especially for the XXZ-HVA circuit. However, we still find critical exponents that are close to the ones obtained from the entanglement entropy scaling collapse, with $\nu \approx 1.31$ and $\nu \approx 1.5$, respectively.

Appendix D

Additional details for Quantum-Classical-Quantum interfaces

D.1 Interfaces for hybrid classical-quantum circuits with Frames

Here we give a more formal presentation of the mathematical background of our algorithm in the language of frames.

Preliminaries

We consider an n-qubit system $\mathcal{S}$ of Hilbert space $\mathcal{H}_{\mathcal{S}}$, and denote the space of bounded, linear operators on $\mathcal{H}_{\mathcal{S}}$ by $\mathcal{L}(\mathcal{H}_{\mathcal{S}})$. We now consider the notion of a frame, which generalizes the notion of basis [141, 142]. For our purposes, a frame $\mathcal{F}_{\mathcal{S}}$ for $\mathcal{L}(\mathcal{H}_{\mathcal{S}}$ is any set $\mathcal{F}_{\mathcal{S}} := \{M_{\boldsymbol{a}}\}_{\boldsymbol{a}}$ of Hermitian operators $M_{\boldsymbol{a}}$ that spans $\mathcal{L}(\mathcal{H}_{\mathcal{S}})$. Such a (in general linearly-dependent) spanning set is sometimes referred to as over-complete basis of $\mathcal{L}(\mathcal{H}_{\mathcal{S}})$. In turn, a frame $\mathcal{D}_{\mathcal{S}} := \{\tilde{M}_{\boldsymbol{a}}\}_{\boldsymbol{a}}$ s.t.

$$\mathcal{I} = \sum_{\boldsymbol{a}} \left|\tilde{M}_{\boldsymbol{a}}\right\rangle\!\left(M_{\boldsymbol{a}}\right|, \tag{D.1}$$

where $\mathcal{I}$ is the identity map on $\mathcal{L}(\mathcal{H}_\mathcal{S})$, is called dual to $\mathcal{F}_\mathcal{S}$ (and we then refer to $\mathcal{F}_\mathcal{S}$ as the primal to $\mathcal{D}_\mathcal{S}$). In Equation (D.1), the identity channel is written in the so-called Liouville or transfer matrix representation. That is, the round kets and bras denote 2^{2N}-dimensional column and row vectors, respectively, representing operators in $\mathcal{L}(\mathcal{H}_\mathcal{S})$ and their Hermitian adjoints. Accordingly, $(A|\ B)$ denotes the Hilbert-Schmidt inner product $\mathrm{Tr}[A^\dagger B]$ in $\mathcal{L}(\mathcal{H}_\mathcal{S})$. This is a popular notation in quantum information [365, 141, 142] that will be used here interchangeably with the (more usual) operator notation upon convenience.

We take throughout $M_{\boldsymbol{a}} \geq 0$ for all $\boldsymbol{a}$ and $\sum_{\boldsymbol{a}} M_{\boldsymbol{a}} = I_\mathcal{S}$, with $I_\mathcal{S}$ the identity operator on $\mathcal{H}_\mathcal{S}$, so that $\mathcal{F}_\mathcal{S}$ is a positive operator-valued measure (POVM) on $\mathcal{H}_\mathcal{S}$. POVMs define generalized (i.e. beyond von Neumann) measurements [28, 152]. This, together with Equation (D.1), allows us to express any density operator $\rho \in \mathcal{L}(\mathcal{H}_\mathcal{S})$ as

$$|\rho) = \sum_{\boldsymbol{a}} P_\rho(\boldsymbol{a}) \left|\tilde{M}_{\boldsymbol{a}}\right), \tag{D.2}$$

where $P_\rho(\boldsymbol{a}) := (M_{\boldsymbol{a}}|\ \rho)$ is the probability of measurement outcome $\boldsymbol{a}$ on ρ. Equation (D.2) is the basis of classical-shadow tomography, a powerful technique to get compact classical representations of states from measurements [153, 140].

Note that $M_{\boldsymbol{a}} \geq 0$ for all $\boldsymbol{a}$ implies $\tilde{M}_{\boldsymbol{a}} \ngeq 0$ in general [141, 142]. In addition, it will be useful to express the dual frame elements as affine combination of elements of $\mathcal{F}_\mathcal{S}$,

$$\left|\tilde{M}_{\boldsymbol{a}}\right) = \sum_{\boldsymbol{a}'} \mathfrak{T}_{\boldsymbol{a},\boldsymbol{a}'}\, |M_{\boldsymbol{a}'})\,, \forall\, \boldsymbol{a}, \tag{D.3}$$

for some adequately chosen $\mathfrak{T}$. With this parametrization, the primal- and dual-frame overlap matrices T and $\tilde{T}$, respectively defined as $T_{\boldsymbol{a},\boldsymbol{a}'} := (M_{\boldsymbol{a}}|\ M_{\boldsymbol{a}'})$ and $\tilde{T}_{\boldsymbol{a},\boldsymbol{a}'} := \left(\tilde{M}_{\boldsymbol{a}}\right|\ \tilde{M}_{\boldsymbol{a}'}\big)$, are related as $\tilde{T} = \mathfrak{T}\, T\, \mathfrak{T}$.

An experimentally convenient choice of $\mathcal{F}_\mathcal{S}$ and $\mathcal{D}_\mathcal{S}$ is $M_{\boldsymbol{a}} = M_{a_1} \otimes \ldots \otimes M_{a_n}$ and $\tilde{M}_{\boldsymbol{a}} = \tilde{M}_{a_1} \otimes \ldots \otimes \tilde{M}_{a_n}$, for $\boldsymbol{a} := (a_1, \ldots a_n)$. Here, M_{a_j} is the j-th element of a single-qubit POVM frame and $\tilde{M}_{a_j}$ that of the corresponding dual frame. We refer to these as factorable frames. By virtue of Eqs. (D.2) and (D.3), these allow one to express any ρ as an affine combination of product states $\sigma_{\boldsymbol{a}} := M_{\boldsymbol{a}}/t_{\boldsymbol{a}}$, where $t_{\boldsymbol{a}} := \mathrm{Tr}[M_{\boldsymbol{a}}]$ [46]. This fact has been used to reconstruct quantum states [46], processes [155], and overlaps [156] from single-qubit measurements. Additionally, this has been used to simulate quantum circuits [157] with generative machine learning models, where $\mathfrak{T}$ was taken as the canonical pseudo-inverse of T. However, other choices of $\mathfrak{T}$ are possible. It can be seen (see App. Appendix D.1.1) that Equation (D.3) defines a dual to $\mathcal{F}_\mathcal{S}$ iff $\mathfrak{T}_{\boldsymbol{a},\boldsymbol{a}'} \in \mathbb{R}$, $\sum_a \mathfrak{T}_{\boldsymbol{a},\boldsymbol{a}'} = 1$, and

$$T = T\, \mathfrak{T}\, T. \tag{D.4}$$

In general, the elements of $\mathfrak{T}$ can be positive or negative. As shown below, the negativity of $\mathfrak{T}$ governs the sample complexity of Monte Carlo estimations of expectation values of observables. Finally, note also that if $\mathfrak{T}$ fulfills Equation (D.4), necessarily so does $\tilde{T} = \mathfrak{T}\, T\, \mathfrak{T}$ ($\tilde{T}$ and $\mathfrak{T}$ collapsing to each other for the canonical choice of $\mathfrak{T}$ being a pseudo-inverse of T).

Interfaces for hybrid classical-quantum circuits

Our goal is to simulate quantum circuits using hybrid classical-quantum ones. More precisely, we are given an observable O, an n-qubit input state $\rho_0 := |0\rangle\langle 0|$, and a target circuit $C := \{U_k\}_{k\in[f]}$, with $f \in \mathbb{N}$ single- or two-qubit unitary gates U_k. We denote by $s_k \subset \mathcal{S}$ the subset of qubits on which U_k acts, and by $\boldsymbol{a}_{s_k}$ a corresponding sub-string of measurement outcomes on s_k. In addition, we use the shorthand notations $\overline{s}_k := \mathcal{S} \setminus s_k$ for the qubits on which U_k does not act and $I_{\overline{s}_k}$ for the identity on $\mathcal{H}_{\overline{s}_k}$. From the f gates, $l < f$ are particularly experimentally demanding for NISQ implementations, and they are marked by the set of labels $L := \{k_1, k_2, \dots k_l\}$. The case we explicitly study below is that of two-qubit gates on qubits far apart in the connectivity graph in question. However, other relevant cases may be due to error mitigation convenience or other hardware-specific limitations, e.g. Either way, our goal is to estimate the expectation value $\mathrm{Tr}[\rho_K\, O]$ of O on the output state $\rho_K := U_f \dots U_1\, \rho_0\, U_1^\dagger \dots U_f^\dagger$ by substituting every U_k with $k \in L$ by a classical simulation of it.

Our main tool to achieve this are interfaces between quantum objects and their (classical) frame representations. The first one is based on Equation (D.2).

Definition D.1.1 (*Quantum-classical interfaces*)**.** *We refer as a QC interface on s_k to the assignment of a classical snapshot $\tilde{M}_{\boldsymbol{a}_{s_k}}$ to s_k according to the measurement outcome $\boldsymbol{a}_{s_k}$ of a factorable POVM frame $\mathcal{F}_{s_k}$ on a state $\rho \in \mathcal{H}_{\mathcal{S}}$, occurring with probability $P_\rho(\boldsymbol{a}_{s_k}) = (I_{\overline{s}_k}|(M_{\boldsymbol{a}_{s_k}}|\rho)$.*

The second one is the reverse interface, which simulates $\tilde{M}_{\boldsymbol{a}_{s_k}}$ as a linear combination of states $\sigma_{\boldsymbol{b}_{s_k}} := M_{\boldsymbol{b}_{s_k}}/t_{\boldsymbol{b}_{s_k}}$. This is done by importance-sampling $\boldsymbol{b}_{s_k}$ from $\tilde{T}^{(\mathcal{I}_{s_k})}$, given $\boldsymbol{a}_{s_k}$, with $\tilde{T}^{(\mathcal{I}_{s_k})}$ the dual-frame overlap matrix on s_k. To see this, we apply on $|\tilde{M}_{\boldsymbol{a}_{s_k}})$ the Hermitian conjugate of Equation (D.1) and get $|\tilde{M}_{\boldsymbol{a}_{s_k}}) = \sum_{\boldsymbol{b}_{s_k}} \tilde{T}^{(\mathcal{I}_{s_k})}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\, t_{\boldsymbol{b}_{s_k}}\, |\sigma_{\boldsymbol{b}_{s_k}})$. Then, using a standard trick, we rewrite

$$\tilde{T}^{(\mathcal{I}_{s_k})}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}} =: \left\|\tilde{T}^{(\mathcal{I}_{s_k})}_{\boldsymbol{a}_{s_k}}\right\|_1 P_{\mathcal{I}_{s_k}}(\boldsymbol{b}_{s_k}|\boldsymbol{a}_{s_k})\, \mathrm{sgn}\big(\tilde{T}^{(\mathcal{I}_{s_k})}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\big), \tag{D.5}$$

where $\tilde{T}^{(\mathcal{I}_{s_k})}_{\boldsymbol{a}_{s_k}}$ is a shorthand notation for the vector given by the $\boldsymbol{a}_{s_k}$-th row of $\tilde{T}^{(\mathcal{I}_{s_k})}$, $\left\|\tilde{T}^{(\mathcal{I}_{s_k})}_{\boldsymbol{a}_{s_k}}\right\|_1 := \sum_{\boldsymbol{b}_{s_k}} \left|\tilde{T}^{(\mathcal{I}_{s_k})}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\right|$ its l_1-norm, and $P_{\mathcal{I}_{s_k}}(\boldsymbol{b}_{s_k}|\boldsymbol{a}_{s_k}) := \left|\tilde{T}^{(\mathcal{I}_{s_k})}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\right| / \left\|\tilde{T}^{(\mathcal{I}_{s_k})}_{\boldsymbol{a}_{s_k}}\right\|_1$.

By construction, $P_{\mathcal{I}_{s_k}}(\circ|\boldsymbol{a}_{s_k})$ is a valid probability distribution, from which $\boldsymbol{b}_{s_k}$ can be sampled. This can be used to quantum Monte Carlo simulate $\tilde{M}_{\boldsymbol{a}_{s_k}}$ [145].

Definition D.1.2 (*Classical-quantum interface*)**.** *We refer as CQ interface on s_k to the repreparation of s_k in the state $\sigma_{\boldsymbol{b}_{s_k}}$, with probability $P_{\mathcal{I}_{s_k}}(\boldsymbol{b}_{s_k}|\boldsymbol{a}_{s_k})$, given a classical snapshot $\tilde{M}_{\boldsymbol{a}_{s_k}}$. Each sampled duple $(\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k})$ is assigned the value $\left\|\tilde{T}^{(\mathcal{I}_{s_k})}_{\boldsymbol{a}_{s_k}}\right\|_1 t_{\boldsymbol{b}_{s_k}}\, sgn\big(\tilde{T}^{(\mathcal{I}_{s_k})}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\big)$.*

The third and final ingredient integrates QC and CQ interfaces with a classical simulation of U_k. We denote by $\mathcal{U}_k$ the superoperator representing the action of the unitary U_k on $\mathcal{L}(\mathcal{H}_{\mathcal{S}})$. Multiplying $\mathcal{U}_k$ from the right by Equation (D.1) and from the left by the Hermitian conjugate of Equation (D.1), we get $\mathcal{U}_k = \sum_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}} |M_{\boldsymbol{b}_{s_k}})\, \tilde{T}^{(\mathcal{U}_k)}_{\boldsymbol{b}_{s_k},\boldsymbol{a}_{s_k}} (M_{\boldsymbol{a}_{s_k}}|$, where $\tilde{T}^{(\mathcal{U}_k)}_{\boldsymbol{b}_{s_k},\boldsymbol{a}_{s_k}} := (\tilde{M}_{\boldsymbol{b}_{s_k}}|\,\mathcal{U}_k\,|\tilde{M}_{\boldsymbol{a}_{s_k}})$. With this, we get

$$\mathcal{U}_k\,|\rho_{k-1}) = \sum_{\boldsymbol{a},\boldsymbol{a}'} \tilde{T}^{(\mathcal{U}_k)}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\, t_{\boldsymbol{b}_{s_k}}\, |\sigma_{\boldsymbol{b}_{s_k}})\, (M_{\boldsymbol{a}_{s_k}}|\rho_{k-1}), \tag{D.6}$$

where $\rho_{k-1} = U_{k-1} \ldots U_1\, \rho_0\, U_1^\dagger \ldots U_{k-1}^\dagger$. That is, the action of $\mathcal{U}_k$ is absorbed into the repreparation by sampling from $\tilde{T}^{(\mathcal{U}_k)}$ instead of $\tilde{T}^{(\mathcal{I}_{s_k})}$. This leads to:

Definition D.1.3 (*Quantum-classical-quantum interface*)**.** *We refer as a QCQ interface for U_k on s_k to the measurement of $\mathcal{F}_{s_k}$, with outcome $\boldsymbol{a}_{s_k}$, followed by the repreparation of $\sigma_{\boldsymbol{b}_{s_k}}$ with probability $P_{\mathcal{U}_k}(\boldsymbol{b}_{s_k}|\boldsymbol{a}_{s_k}) := \left|\tilde{T}^{(\mathcal{U}_k)}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\right| / \left\|\tilde{T}^{(\mathcal{U}_k)}_{\boldsymbol{a}_{s_k}}\right\|_1$. Each sampled duple $(\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k})$ is assigned the value $v_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}} := \left\|\tilde{T}^{(\mathcal{U}_k)}_{\boldsymbol{a}_{s_k}}\right\|_1 t_{\boldsymbol{b}_{s_k}}\, sgn\big(\tilde{T}^{(\mathcal{U}_k)}_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\big)$; and the corresponding interface realized in such experimental run is thus mathematically represented by the operator $\mathcal{V}_k(\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k}) \coloneqq v_{\boldsymbol{a}_{s_k},\boldsymbol{b}_{s_k}}\, |\sigma_{\boldsymbol{b}_{s_k}})(M_{\boldsymbol{a}_{s_k}}|$.*

Our hybrid-circuit simulation then applies on ρ_{k-1} the gate U_k if $k \notin L$, but a QCQ interface for U_k instead if $k \in L$. Introducing the terminology

$$\mathcal{W}_k(\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k}) = \begin{cases} \mathcal{U}_k, & \text{if } k \notin L, \\ \mathcal{V}_k(\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k}), & \text{if } k \in L, \end{cases} \tag{D.7}$$

and using the fact that O is Hermitian, we can express the target expectation value $\mathrm{Tr}[\rho_K\, O]$ as

$$(O|\ \rho_K) = \sum_{\boldsymbol{\alpha}_{s_L}} (O| \prod_{k=1}^{f} \mathcal{W}_k(\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k})|\rho_0), \tag{D.8}$$

with the shorthand notation $\boldsymbol{\alpha}_{s_L} := (\boldsymbol{a}_{s_{k_l}}, \boldsymbol{b}_{s_{k_l}}, \dots\ \boldsymbol{a}_{s_{k_1}}, \boldsymbol{b}_{s_{k_1}})$.

Equation (D.8) can be experimentally estimated through an average $O^*_{M_s}$ over $M_s \in \mathbb{N}$ runs. M_s is chosen to guarantee that the statistical error and significance level (failure probability) of the estimation are respectively given by target values ε and δ. We refer to M_s as *sample complexity* of the protocol and its explicit value is given in Theorem 2.4.1 below.

The procedure is sketched by the following pseudocode.

Algorithm 5: Hybrid classical-quantum simulation with QCQ interfaces.

Input: ρ_0, C, O, ε, δ
Output: $O^*_{M_s}$ s.t. $\left|O^*_{M_s} - \mathrm{Tr}\left[O\,\rho_K\right]\right| \leq \varepsilon$ with probability at least $1-\delta$.
Initialize $O^*_{M_s} = 0$, $v = 1$, and M_s as in Equation (D.10).
for $m \in (1, \dots, M_s)$ **do**
  for $k \in (1, \dots, f)$ **do**
    if $k \in L$ **then**
      Apply a QCQ interface for U_k on qubits s_k, obtaining the duple $(\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k})$;
      $v \leftarrow v \times v_{\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k}}$, with $v_{\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k}}$ as in Definition D.1.3.
    else
      Apply the gate U_k on qubits s_k.
  Measure O, obtaining the measurement outcome (eigenvalue of O) o;
  $O^*_{M_s} \leftarrow O^*_{M_s} + o \times v$.
$O^*_{M_s} \leftarrow \frac{O^*_{M_s}}{M_s}$.

To quantify the runtime of the algorithm, we define the *interface negativity* of the gate $\mathcal{U}_k$ and the *total forward interface negativity* of the entire circuit C respectively as

$$\mathcal{N}_{U_k} := \max_{\boldsymbol{a}_{s_k}, \boldsymbol{b}_{s_k}} \left\| \tilde{T}^{(\mathcal{U}_{s_k})}_{\boldsymbol{a}_{s_k}} \right\|_1 t_{\boldsymbol{b}_{s_k}} \quad \text{and} \quad \mathcal{N}_{\rightarrow} := \prod_{k \in L} \mathcal{N}_{U_k}. \tag{D.9}$$

This allows us to state the following theorem.

Theorem D.1.1. *[Correctness and sample complexity] The finite-statistics average $O^*_{M_s}$ of Algorithm 5 is an unbiased estimator of* $\mathrm{Tr}[\rho_K O]$ *(See Appendix D.2). Moreover, if*

$$M \geq \mathcal{N}_{\rightarrow}^2 \times \frac{2\,\|O\|^2\,\log(2/\delta)}{\varepsilon^2}, \tag{D.10}$$

with $\|O\|$ the operator norm of O, then, with probability at least $1-\delta$, the statistical error of $O^*_{M_s}$ is at most ε.

The proof follows straightforwardly from the Hoeffding bound. We note that the factor $\frac{2\,\|O\|^2\,\log(2/\delta)}{\varepsilon^2}$ in Equation (D.10) is the equivalent sample complexity bound one would obtain if $\mathrm{Tr}[\rho_K O]$ was estimated from measurements on the actual state ρ_K. Hence, $\mathcal{N}_{\rightarrow}^2$ quantifies the runtime overhead introduced by the interfaces. In that regard, the interface negativities play the same role in our hybrid classical-quantum simulation as the negativities of Ref. [145] in fully classical simulations with quasi-probability representations. An innovative and advantageous feature of Equation (D.9) is the presence of the POVM-element trace $t_{\boldsymbol{b}_{s_k}}$ in $\mathcal{N}_{U_k}$, which comes from the state repreparation. Indeed, since $t_{\boldsymbol{b}_{s_k}} < 1$, the $\mathcal{N}_{U_k}$'s (and therefore also $\mathcal{N}_{\rightarrow}$) are significantly smaller than their counterparts for fully classical simulations [145]. This is consistent with the intuition that hybrid classical-quantum Monte Carlo simulations should cause lower sample-complexity increases than fully classical ones.

Either way, the most relevant property for our purposes is that $\mathcal{N}_{\rightarrow}^2$ (and therefore also M_s) is independent not only of the numbers of gates f or qubits n but also, and most importantly, of the connectivity-graph distance between the qubits on which the interfaces act. In other words, for a fixed budget of measurement runs, simulating a gate U_k with a QCQ interface increases the statistical error at most by a constant factor $\mathcal{N}_{U_k}$, regardless how far apart in the circuit the qubits s_k are. In contrast, experimentally synthesizing U_k with noisy nearest-neighbor gates would give a systematic error due to infidelity accumulation that grows linearly with the distance between those qubits. Clearly, the drawback is that $\mathcal{N}_{\rightarrow}^2$ grows exponentially with the number l of interfaces used. However, for a many circuits, Algorithm 5 constitutes a better alternative than the bare NISQ implementation. We study relevant exemplary circuits with such trade-offs in the next sections.

Finally, note that $\mathcal{N}_{\rightarrow}^2$ is frame-dependent. This is crucial to the efficiency of classical simulations [146, 147, 148]. For instance, in quantum Monte Carlo, it is known that the statistical overhead due to negative (quasi-)probabilities can be ameliorated [151] or even removed [150] by local base changes. Something similar applies here: the interface negativities depend not only on the primal frame but also on the choice of dual to it.

D.1.1 Dual frame decomposition

Here, we show that Equation (D.3) defines a dual frame with respect to $\mathcal{F}_{\mathcal{S}}$ if Equation (D.4) holds. For the forward direction of this statement, we start with Equation (D.1) and plug in Equation (D.3) to obtain

$$\mathcal{I} = \sum_{\boldsymbol{a},\boldsymbol{b}} \mathfrak{T}_{\boldsymbol{a},\boldsymbol{b}} \, |M_{\boldsymbol{b}}) \, (M_{\boldsymbol{a}}| \, . \tag{D.11}$$

Applying $(M_{\boldsymbol{c}}|$ and $|M_{\boldsymbol{d}})$ to the left and right of Equation (D.11) then gives

$$(M_{\boldsymbol{c}}| \, M_{\boldsymbol{d}}) = \sum_{\boldsymbol{a},\boldsymbol{b}} \mathfrak{T}_{\boldsymbol{a},\boldsymbol{b}} \, (M_{\boldsymbol{c}}| \, M_{\boldsymbol{b}}) \, (M_{\boldsymbol{a}}| \, M_{\boldsymbol{d}}) \, ,$$

therefore we see that $T = T\mathfrak{T}T$ as required.

For the converse direction, we start with a map $\mathcal{J}$ on $\mathcal{L}(\mathcal{H}_{\mathcal{S}})$,

$$\mathcal{J} = \sum_{\boldsymbol{a},\boldsymbol{b}} \mathfrak{T}_{\boldsymbol{a},\boldsymbol{b}} \, |M_{\boldsymbol{b}}) \, (M_{\boldsymbol{a}}| \, . \tag{D.12}$$

Applying $(M_{\boldsymbol{c}}|$ and $|M_{\boldsymbol{d}})$ to the left and right of Equation (D.12) then gives

$$(M_{\boldsymbol{c}}| \, \mathcal{J} \, |M_{\boldsymbol{d}}) = \sum_{\boldsymbol{a},\boldsymbol{b}} \mathfrak{T}_{\boldsymbol{a},\boldsymbol{b}} \, (M_{\boldsymbol{c}}| \, M_{\boldsymbol{b}}) \, (M_{\boldsymbol{a}}| \, M_{\boldsymbol{d}}) \, .$$

If we then plug in Equation (D.4) we find

$$(M_{\boldsymbol{c}}| \, \mathcal{J} \, |M_{\boldsymbol{d}}) = (M_{\boldsymbol{c}}| \, M_{\boldsymbol{d}}) \, ,$$

from which we conclude that $\mathcal{J} \equiv \mathcal{I}$, i.e. $\mathcal{J}$ equals the identity map and so Equation (D.1) holds.

D.2 Finite statistics estimator

Let O be a generic observable we wish to measure, with support on an arbitrary subset of $\mathcal{S}$ and with arbitrary spectral norm $\|O\|_{\text{sp}} \coloneqq o_{\max}$. Hence, it admits a spectral decomposition as $|O) = \sum_{\lambda} o_{\lambda} \, |\lambda)$, where o_{λ} and $|\lambda)$ are respectively its λ-th eigenvalue and eigenvector

projector, with $|o_\lambda| \leq o_{\max}$ for all λ. Using Equation (D.8), we write the finite statistics estimator of the expectation value $\langle O \rangle \coloneqq \mathrm{Tr}[O\,\rho_\mathrm{f}]$ of O as

$$O_M^* \coloneqq \frac{1}{M} \sum_{i=1}^{M} o_{\lambda^{(i)}, \boldsymbol{\alpha}_{s_L}^{(i)}} \prod_{k \in L} v_{\boldsymbol{a}_{s_k}^{(i)}, \boldsymbol{b}_{s_k}^{(i)}}, \tag{D.13}$$

where $o_{\lambda^{(i)}, \boldsymbol{\alpha}_{s_L}^{(i)}}$ is the eigenvalue obtained from the single-shot i obtained from a state that is measured and reprepared according to $\boldsymbol{\alpha}_{s_L}^{(i)}$. The probability of observing $o_{\lambda^{(i)}, \boldsymbol{\alpha}_{s_L}^{(i)}}$ is given by

$$P(o_{\lambda^{(i)}, \boldsymbol{\alpha}_{s_L}^{(i)}}) = (\lambda^{(i)}| \prod_{k=1}^{f} \mathcal{W}_k(\boldsymbol{a}_{s_k}^{(i)}, \boldsymbol{b}_{s_k}^{(i)}) |\rho_0), \tag{D.14}$$

with $\boldsymbol{a}_{s_k}^{(i)} \sim P_{\rho_{k-1}}(\boldsymbol{a}_{s_k})$ and $\boldsymbol{b}_{s_k}^{(i)} \sim P_{\mathcal{U}_k}(\boldsymbol{b}_{s_k} | \boldsymbol{a}_{s_k})$ where

$$|\rho_{k-1}) = \prod_{k=1}^{l-1} \mathcal{W}_l(\boldsymbol{a}_{s_l}^{(i)}, \boldsymbol{b}_{s_l}^{(i)}) |\rho_0). \tag{D.15}$$

Importantly, O_M^* is an unbiased estimator.

D.3 Locally Purified Density Operators

Numerical simulations with the full density matrix of size $2^n \times 2^n$ quickly become prohibitive due to the large memory requirements. Hence, we have to resort to tensor networks to find efficient representations of mixed quantum states. The canonical choice for representing operators with tensor networks are matrix product operators (MPO) [366]. A drawback of this approach is that applying completely positive maps to the state can still lead to the MPO becoming non-positive due to truncation errors. The Locally Purified Density Operator tensor network solves this issue by representing the state as $\rho = \chi\chi^\dagger$, where the purification operator χ is given by a tensor network

$$[\chi]_{\kappa_1, \ldots, \kappa_n}^{p_1, \ldots, p_n} = \sum_{b_1, \ldots, b_{n-1}} A_{b_1}^{[1] p_1, \kappa_1} A_{b_1, b_2}^{[2] p_2, \kappa_2} \ldots A_{b_{n-1}}^{[n] p_n \kappa_n},$$

with $1 \leq p_l \leq P$, $1 \leq \kappa_l \leq \kappa$ and $1 \leq b_l \leq D$ [159]. Here, P is called the physical dimension, κ is the Kraus dimension and D is the bond dimension.

Analogous to the bond dimension truncation in MPOs, truncating the Kraus dimension after applying a channel leads to errors in our state representation that can affect the accuracy of numerical simulations. However, we can control the accuracy of the simulation by increasing D and κ and keeping track of a runtime lower bound estimate of the state fidelity. Let $\rho = \chi^\dagger \chi$, $\sigma = \eta^\dagger \eta$, then the fidelity is given by

$$F(\rho, \sigma) = \mathrm{Tr}\sqrt{\sqrt{\sigma}\rho\sqrt{\sigma}}.$$

From Lemma 1 in [159] we know that,

$$F(\rho, \sigma) \geq \frac{1}{2}\left(2 - \|\chi - \eta\|_2^2\right).$$

Let χ be a locally purified description of a quantum state with local tensors $\{A^{[n]}\}$ that is in mixed canonical form with respect to a local tensor $A^{[l_{cp}]}$. If a single tensor $A^{[l]}$ is compressed by discarding singular values in either the Kraus or bond dimensions, then by Lemma 6 of [159] we know that

$$\delta := \left(\sum_{i,\text{discarded}} s_i^2\right)^{\frac{1}{2}},$$

and subsequently

$$\|\chi - \chi'\|_2^2 = 2(1 - \sqrt{1 - \delta^2}),$$

where χ' is the compressed tensor. By the triangle inequality, the two norm errors introduced by the discarded weights can at most sum up. Hence, the true operator norm is lower bounded by the sum of all discarded weight errors

$$\|\rho_{\text{exact}} - \rho_{\text{truncated}}\|_2 \leq \sum_d \sqrt{2(1 - \sqrt{1 - \delta_d^2})}.$$

With d the number of truncations and δ_k the discarded weights. This brings the final runtime fidelity estimate to

$$F(\rho, \sigma) \geq \frac{1}{2}\left(2 - \|\chi - \eta\|_2^2\right) \tag{D.16}$$

$$\geq \frac{1}{2}\left(2 - \left(\sum_d \sqrt{2(1 - \sqrt{1 - \delta_d^2})}\right)^2\right). \tag{D.17}$$

In all our experiments, we apply depolarizing channels to both qubits only after applying a two qubit gate, since single qubit gate noise tends to be small in experimental settings. The single-qubit depolarizing channel is given by

$$\rho = \sum_{m=1}^{M} K_m \rho K_m^\dagger,$$

where $\{K_m\}$ is a set of Kraus operators with

$$K_1 = \sqrt{\frac{(4-3\lambda)}{4}} I, \quad K_2 = \sqrt{\frac{\lambda}{4}} X$$
$$K_3 = \sqrt{\frac{\lambda}{4}} Y, \quad K_4 = \sqrt{\frac{\lambda}{4}} Z.$$

Here, $\{X, Y, Z\} \in \mathcal{P}_1$ are the Pauli matrices and I is the identity. The scalar $\lambda \in [0, 1]$ controls the strength of the depolarization. With these channels, illustrate the bound of Equation (D.17) by comparing the final state overlap of an exact full density matrix simulation and a LPDO simulation for a random 4 qubit circuit with a varying number of CNOT gates. In Figure D.1, we see that the runtime estimate of the fidelity is about two orders of magnitude above the true fidelity.

D.4 Circuit cutting via the Hubbard-Stratonovich transformation

In [166], a method is proposed to simulate a sequence of unitaries via Auxiliary MCMC. Consider a quantum gate

$$U(\alpha) = \exp\{-i\alpha AB\}, \quad [A, B] = 0, \tag{D.18}$$

In addition, we require that $A = A_1 \otimes I$ and $B = I \otimes B_2$. For example, a CNOT gate can be constructed from

$$A_1^{\text{CNOT}} = I - Z, \quad B_2^{\text{CNOT}} = I - X, \quad \alpha = \pi/4,$$

where $I_2 = I \otimes I$.

The Hubbard-Stratonovich transformation is used in field theory to linearize a quadratic field in an exponent. This is achieved by completing the square, integrating and shifting

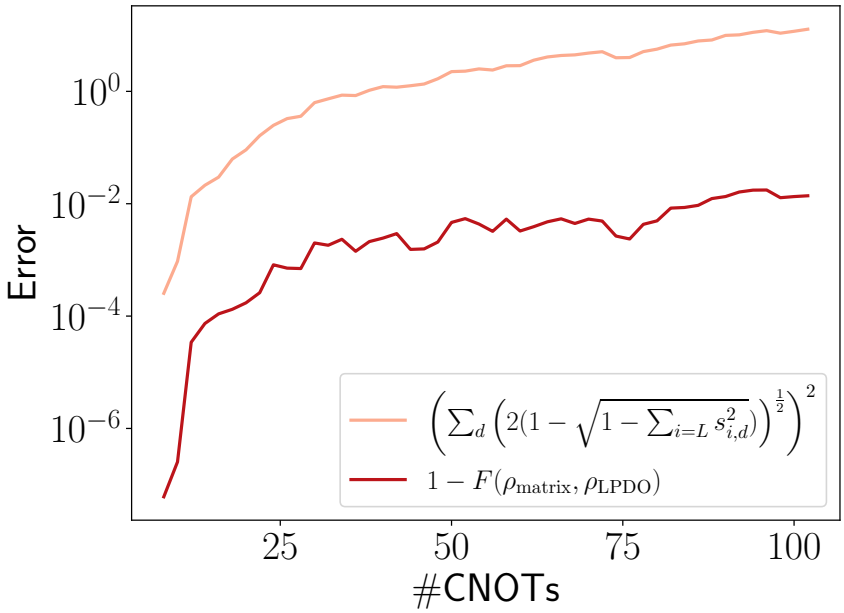


Figure D.1: **Illustration of the lower bound of Equation (D.17).** The circuit consists of an initial state $|+\rangle^{\otimes 4}$ to which we apply a varying number of CNOT gates with random control and target qubits. We set the noise to $\lambda = 0.005$ and take $D = 4$ and $\kappa = 16$. The red line indicates the true accuracy of the LPDO simulation by comparing with the exact full density matrix simulation. The orange line gives the runtime fidelity estimate. We see that the accuracy of the simulation degrades as we add more two-qubit gates and depolarizing channels. The runtime fidelity gives an estimate two orders of magnitude above the exact error, indicating that for this example, the bound is a conservative estimate of the simulation error.

the integration variables. Let, x be a scalar field, then by completing the square and introducing the auxiliary field μ, we obtain

$$\begin{aligned} -ax^2 &= -a(x-\mu)^2 + a\mu^2 - 2a\mu x \\ -ax^2 + a(x-\mu)^2 &= a\mu^2 - 2a\mu x \\ \exp\{-ax^2 + a(x-\mu)^2\} &= \exp\{a\mu^2 - 2a\mu x\} \\ \int_{-\infty}^{\infty} d\mu \exp\{-ax^2 + a(x-\mu)^2\} &= \int_{-\infty}^{\infty} d\mu \exp\{a\mu^2 - 2a\mu x\}. \end{aligned}$$

By shifting the integration on the right side with $\mu \to \mu' + x$, $d\mu \to d\mu'$, we find

$$\exp\{-ax^2\} \int_{-\infty}^{\infty} d\mu' \exp\{a(-\mu')^2\} = \int_{-\infty}^{\infty} d\mu \exp\{a\mu^2 - 2a\mu x\}$$
$$\exp\{-ax^2\} = \sqrt{\frac{a}{\pi}} \int_{-\infty}^{\infty} d\mu \exp\{a\mu^2 - 2a\mu x\},$$

where in the last line we used the Gaussian integral

$$\int_{-\infty}^{\infty} dx \exp\{-ax^2\} = \sqrt{\frac{\pi}{a}}.$$

We now apply the same idea to Equation (D.18). We have the identity,

$$i\alpha AB = i\alpha((A-\tau)(B-\varphi) + \varphi A + \tau B - \tau\varphi)$$
$$i\alpha(AB - (A-\tau)(B-\varphi)) = i\alpha(\varphi A + \tau B - \tau\varphi),$$

where τ and φ are real auxiliary fields. Integrating and shifting the integration variables then gives

$$\exp\{-i\alpha AB\} = \frac{|\alpha|}{\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} d\tau d\varphi \exp\{-i\alpha(\varphi A + \tau B - \tau\varphi)\}.$$

We introduce the measure

$$D\sigma = \frac{|\alpha|}{\pi} d\tau d\varphi,$$

and the gates

$$V(\tau) = \exp\{-i\alpha\tau A\}$$
$$W(\varphi) = \exp\{-i\alpha\varphi B\}$$
$$P[\sigma] = V(\varphi)W(\tau),$$

where $\sigma = (\tau, \varphi)$. We can therefore represent the gate $U(\alpha)$ as

$$U(\alpha) = \int_{-\infty}^{\infty} D\sigma \; P[\sigma] e^{iS(\alpha,\tau,\varphi)},$$

where $iS(\alpha, \tau, \varphi) = i\alpha\tau\varphi$ is a complex action.

Consider the evolution of a state ρ under given this gate representation,

$$\begin{aligned} U(\alpha)\rho U^{\dagger}(\alpha) &= \int_{-\infty}^{\infty} D\sigma \int_{-\infty}^{\infty} D\sigma' e^{iS(\alpha,\tau,\varphi)+iS(\alpha,\tau',\varphi')}\, P[\sigma]\rho P^{\dagger}[\sigma'] \\ &= \int_{-\infty}^{\infty} D\sigma \int_{-\infty}^{\infty} D\sigma' e^{i\tilde{S}(\alpha,\sigma,\sigma')}\, P[\sigma]\rho P^{\dagger}[\sigma'], \end{aligned}$$

where we defined $i\tilde{S}(\alpha,\sigma,\sigma') = iS(\alpha,\tau,\varphi) + iS(\alpha,\tau',\varphi')$. We can calculate an expectation value given some Hermitian operator H as

$$\mathrm{Tr}U(\alpha)\rho U^{\dagger}(\alpha)H = \int_{-\infty}^{\infty} D\sigma \int_{-\infty}^{\infty} D\sigma' e^{i\tilde{S}(\alpha,\sigma,\sigma')}\, \mathrm{Tr}P[\sigma]\rho P^{\dagger}[\sigma']H.$$

This quantity can be estimated via the auxiliary quantum Monte Carlo method [367].

Appendix E

Additional details Riemannian gradient flow

E.1 Dynamical Lie algebra gradient flow

Given a Hamiltonian $H = \sum_n O_n$, let $\mathfrak{g}$ denote the set of operators spanned by consecutive applications of the Lie bracket to the set $\{O_n\}$, i.e. the closure of $\{O_n\}$ under commutation (see Equation (4.1) and Chapter 4). The resulting dynamical Lie algebra is a subalgebra of $\mathfrak{su}(2^n)$ and determines the set of states that can be reached by applying unitaries generated by elements of $\mathfrak{g}$ [219, 220, 228]. Let $H \in \mathfrak{g}$ and

$$\rho_0 = \frac{1}{2^n} I + \sum_l P_l, \quad \forall P_l \in \mathfrak{g}.$$

We then find that the commutator

$$\begin{aligned}[\rho_0, H] &= \frac{1}{2^n}[I, H] + \sum_l [P_n, H] \\ &= \sum_i [P_n, H],\end{aligned}$$

is also an element of $\mathfrak{g}$ because $\mathfrak{g}$ is closed under commutation. Since $\mathfrak{g}$ is a subalgebra, there is a corresponding subgroup G whose elements are generated by exponentiation of elements in $\mathfrak{g}$. Hence,

$$U = \exp\{\epsilon[\rho_0, H]\},$$

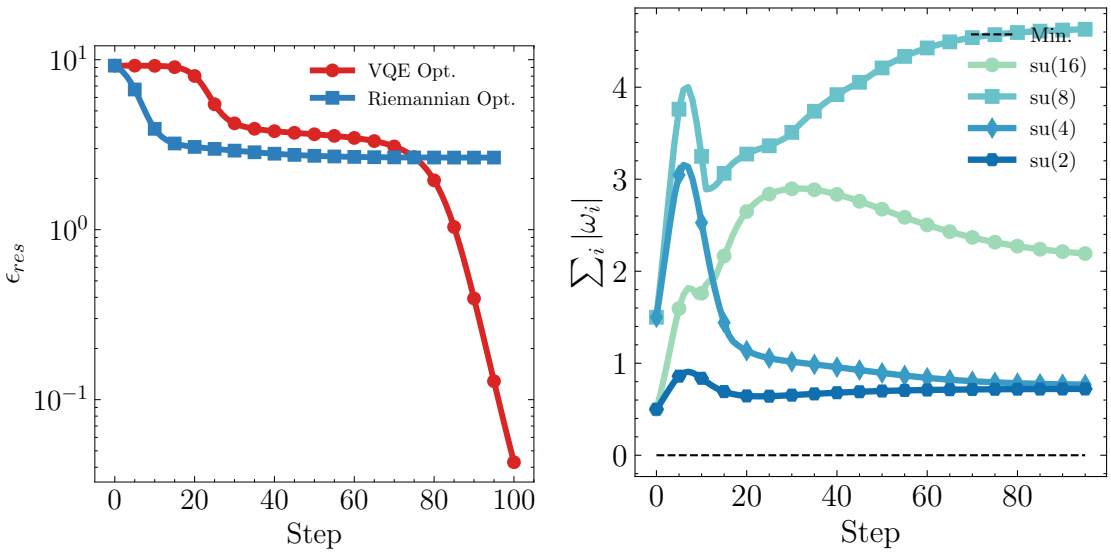


Figure E.1: **Comparison of Riemannian gradient optimization versus gradient-based VQE for the 4-qubit transverse field Ising-model.** The Riemannian gradient circuit is initialized the with the identity. The VQE circuit is initialized in the same way as in Figure 3.6 in the main text. The available directions for the Riemannian optimizer are the Pauli strings in the dynamical Lie algebra: $\mathfrak{g} := \{X_i, Z_iZ_j, Y_iZ_j, Z_iY_j, Y_iY_j | 1 \leq i, j \leq N, i < j\}$, where X_i, Y_i, Z_i are Paulis on location i [228]. Even though the gradient flow stays within the dynamical Lie algebra at every step during the optimization, we see that the ground state is still unreachable and the optimization gets stuck in a local minimum.

is an element of the subgroup G. But then

$$U_{k+1} = \exp\{\epsilon[\rho_0, H]\}U_k,$$

will stay in the group G as long as U_0 is an element of G. We therefore see that an appropriate choice of ρ_0 and U_0 will keep the Riemannian gradient flow within the subgroup G. Unfortunately, it is possible that H cannot be diagonalized by elements of G and so the ground state may be unreachable for a flow that stays in the dynamical Lie algebra, see Figure E.1.

Appendix F

Additional details here comes the SU(N)

F.1 The generalized parameter-shift rule

When using our $\mathbb{SU}(N)$ gate in an application that involves gradient-based optimization, like demonstrated in the numerical experiments in this work, we require calculating the partial derivatives in Equation (3.39). Here we provide the details for how this can be achieved in practice via the generalized parameter-shift rule (GPSR) [116, 118, 119]. Without loss of generality, we rewrite the cost function in Equation (3.38) as

$$E(t) = \mathrm{Tr}\left\{\tilde{H}e^{t\Omega}\tilde{\rho}e^{-t\Omega}\right\}, \tag{F.1}$$

where we absorbed the rest of the circuit into $\tilde{\rho}$, $\tilde{H}$ and fixed any other parameters in the circuit. Computing the derivative of Equation (F.1) with respect to t is equivalent to the problem of finding the gradient in Equation (3.39) at $t = 0$. For the numerical experiments in this paper we make use of the particular implementation of the GPSR in [119] as well as the alternative method outlined in Appendix F.5.

The skew-Hermitian operator Ω in Equation (F.1) has (possibly degenerate) eigenvalues $\{i\lambda_j\}$. We define the set of unique spectral gaps as $\Gamma = \{|\lambda_j - \lambda_{j'}|\}$ where $j' > j$. Note that for d distinct eigenvalues, the number of unique spectral gaps R is bounded via $R \leq d(d-1)/2$. We relabel every unique spectral gap with an integer, i.e. we write $\Delta_r \in \Gamma$, and define the corresponding vector $\boldsymbol{\Delta} = (\Delta_1, \ldots, \Delta_R)$. We pick a set of parameter shifts

that are equidistant and create a vector of R shifts $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_R)$ where

$$\delta_n = \frac{(2n-1)\pi}{4R}, \quad n = 1, \ldots, R.$$

Next, we create the length R cost vector $\boldsymbol{c}$ and the $R \times R$ matrix M

$$\boldsymbol{c} = \begin{pmatrix} E(\delta_1) - E(-\delta_1) \\ E(\delta_2) - E(-\delta_2) \\ \vdots \\ E(\delta_R) - E(-\delta_R) \end{pmatrix}, \quad \mathrm{M}(\boldsymbol{\delta}) = \begin{pmatrix} 2\sin(\delta_1\Delta_1) & \ldots & 2\sin(\delta_1\Delta_R) \\ 2\sin(\delta_2\Delta_1) & \ldots & 2\sin(\delta_2\Delta_R) \\ \vdots & \vdots & \vdots \\ 2\sin(\delta_R\Delta_1) & \ldots & 2\sin(\delta_R\Delta_R) \end{pmatrix}.$$

We then calculate the coefficient vector as

$$\boldsymbol{r} = (\mathrm{M}(\boldsymbol{\delta}))^{-1} \cdot \boldsymbol{c},$$

which finally gives the gradient

$$\left.\frac{dE(x)}{dt}\right|_{t=0} = \boldsymbol{\Delta} \cdot \boldsymbol{r}.$$

Since the final gradient is exact, finite shot estimates of all $c(\delta_n)$'s will produce an unbiased estimate of $dC(x)/dt$,

$$\left.\frac{dE(x)}{dt}\right|_{t=0} = \boldsymbol{\Delta} \cdot (\mathrm{M}(\boldsymbol{\delta}))^{-1} \cdot \mathbb{E}\left[\boldsymbol{c}\right],$$

where we pulled out $\boldsymbol{\Delta}$ and $(\mathrm{M}(\boldsymbol{\delta}))^{-1}$ since they are constant. The difficulty of obtaining an accurate estimate of the gradient is determined by the variance of this estimator, which is given by

$$\mathrm{Var}\left[\left.\frac{dE(x)}{dt}\right|_{t=0}\right] = (\boldsymbol{\Delta} \cdot (\mathrm{M}(\boldsymbol{\delta}))^{-1})^{\odot 2} \cdot \left(\mathbb{E}\left[\boldsymbol{c}^{\odot 2}\right] - \mathbb{E}\left[\boldsymbol{c}\right]^{\odot 2}\right),$$

where we used $\odot 2$ to emphasize that the squares are taken elementwise. We assume that the estimates for each shifted circuit obey normal statistics and so since these are independent, we can write

$$\mathbb{E}\left[\boldsymbol{c}^{\odot 2}\right] - \mathbb{E}\left[\boldsymbol{c}\right]^{\odot 2} \approx \frac{1}{N_{\mathrm{shots}}} \begin{pmatrix} \sigma^2(\delta_1) + \sigma^2(-\delta_1) \\ \sigma^2(\delta_2) + \sigma^2(-\delta_2) \\ \vdots \\ \sigma^2(\delta_R) + \sigma^2(-\delta_R) \end{pmatrix},$$

where $\sigma^2(\pm\delta_n)$ is the variance of the cost for each shifted circuit. If we assume that the dependence of σ on the shifts is mild, i.e., $\sigma(\delta) \approx \sigma_0$ then the total variance will only depend on the prefactor. Setting the estimate $\mathbb{E}\left[\boldsymbol{c}^{\odot 2}\right] - \mathbb{E}\left[\boldsymbol{c}\right]^{\odot 2} = (\sigma_0^2, \sigma_0^2, \ldots, \sigma_0^2)$ then finally gives

$$\mathrm{Var}\left[\left.\frac{dE(x)}{dt}\right|_{t=0}\right] \approx 2\sigma_0^2 \left(\sum_n \Delta_n \mathrm{M}^{-1}_{nm}(\boldsymbol{\delta})\right)^2 .$$

One can minimize this quantity with respect to $\boldsymbol{\delta}$ to find the optimal set of shifts for the gradient estimation [119].

F.2 Alternative differentiation of $\mathrm{SU}(N)$ gates

In this section we summarize a number of alternative differentiation techniques that may be applied to the presented $\mathbb{SU}(N)$ gates. In particular, we discuss the stochastic parameter-shift rule, which was created for multi-parameter gates, finite differences as a standard tool in numerical differentiation, as well as an alternative to the general parameter-shift rule above which also exploits the notion of effective generators.

F.3 The Stochastic parameter-shift rule

The stochastic parameter-shift rule [233] relies on the following operator identity [258]

$$\frac{\partial e^{Z(x)}}{\partial x} = \int_0^1 ds\, e^{sZ(x)} \frac{\partial Z(x)}{\partial x} e^{(1-s)Z(x)},$$

for any bounded operator $Z(x)$. We now fix all parameters θ_m for $m \neq l$ and rewrite the cost in Equation (3.38) as

$$c(x) = \mathrm{Tr}\left\{H e^{i(xG_l + A')} \rho e^{-i(xG_l + A')}\right\}, \quad A' \equiv \sum_{m \neq l} \theta_m G_m. \tag{F.2}$$

Then, if we take $Z(x)$ to be the operator $Z(x) = i(xG_l + A')$ we can construct the gradient of Equation (F.2) as

$$\frac{\partial c(x)}{\partial x} = \int_0^1 ds\, \left(C_+(x,s) - C_-(x,s)\right), \tag{F.3}$$

where

$$C_{\pm}(x,s) = \mathrm{Tr}\left\{HV_{\pm}(x)\rho V_{\pm}^{\dagger}(x)\right\}$$
$$V_{\pm}(x) = e^{is(xG_l+A')}e^{\pm\frac{\pi}{4}G_l}e^{i(1-s)(xG_l+A')}.$$

Hence, similar to our method, the gradient evaluation requires adding gates to the circuit and evaluating the new circuit. However, the stochastic parameter-shift rule comes at a significant cost: the evaluation of the integral in Equation (F.3). In practice, one approximates this integral by sampling values of s uniformly in the interval $(0,1)$ and then calculating the costs $C_{\pm}(x,s)$ with a finite-shot estimate. Although this produces an unbiased estimator, we find that the variance of this estimator is larger than ours, see Figure 3.10. In addition, this method leads to a bigger number of unique circuits to compute the derivative, increasing the compilation overhead for both hardware and simulator implementations.

F.4 Finite differences

Finite differences are widely used to differentiate functions numerically. We briefly discuss this method in the context of variational quantum computation (VQC) and refer the reader to recent works comparing and optimizing differentiation techniques for VQC [117, 266].

In particular, we consider the central difference recipe

$$\partial_{\mathrm{FD},\theta_j}E(\boldsymbol{\theta}) = \frac{1}{\delta}\left[E\left(\boldsymbol{\theta}+\frac{\delta}{2}\boldsymbol{e}_j\right) - E\left(\boldsymbol{\theta}-\frac{\delta}{2}\boldsymbol{e}_j\right)\right], \tag{F.4}$$

where δ is a freely chosen shift parameter and $\boldsymbol{e}_j$ is the jth canonical basis vector. This recipe is an approximation of $\partial_{\theta_j}E(\boldsymbol{\theta})$, making the corresponding estimator on a shot-based quantum computer biased. This bias, which depends on δ, has to be traded off against the variance of the estimator, which grows approximately with δ^{-2}.

In classical computations, the numerical precision cutoff plays the role of the variance. Due to the high precision in classical computers, this leads to optimal shifts $\delta \ll 1$, which allows treating the bias to leading order in δ and thus enables rough estimates of the optimal δ^* in advance. On a quantum computer, however, the variance typically is more than ten orders of magnitude larger, leading to a very different δ^*, which furthermore depends on the function and derivative values. As a consequence, shifts of $\mathcal{O}(1)$ become a reasonable choice, highlighting the similarity of the central difference recipe to the two-term parameter-shift rule [117].

As a demonstration of the above, and in preparation for the numerical experiments shown in Figures 3.9 and 3.10, we compute the central difference gradient for a random single-qubit Hamiltonian, a single $\mathbb{SU}(2)$ gate $U(\boldsymbol{\theta}) = \exp(iaX + ibY)$ and $\delta \in \{0.5, 0.75, 1.0\}$. For this, we evaluate the mean and standard error 50 times and show the difference to the exact derivative in Figure F.1. As expected, we observe that the bias increases with δ and that the variance is suppressed with larger δ. We determine $\delta = 0.75$ to be a reasonable choice for the purpose of the demonstration in Figures 3.9 and 3.10, but stress that for any other circuit, qubit count, Hamiltonian, and even for a different parameter position $\boldsymbol{\theta}$ for this circuit, the optimal shift size needs to be determined anew.

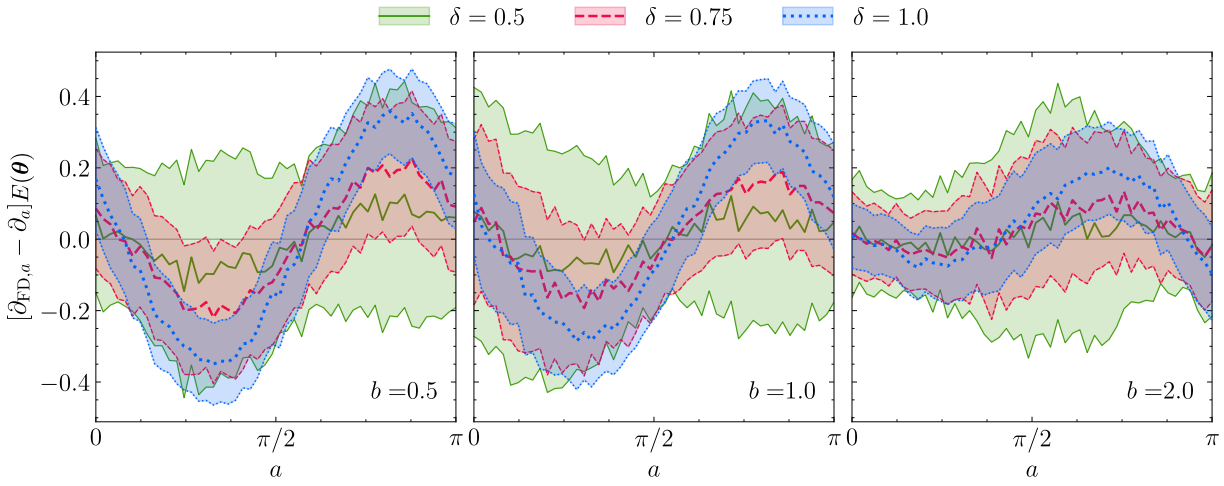


Figure F.1: **Error of the central difference gradients with $\delta = 0.5, 0.75, 1.0$ for the single-qubit example from Figures 3.9 and 3.10.** The value of the second parameter again is fixed to $b = 0.5, 1.0, 2.0$ in the panels (from left to right). The shift parameter δ influences the strengths of bias and variance, leading to a trade-off. For smaller δ, the variance is enhanced due to the coefficients in Equation (F.4) that scale with δ^{-1}. For larger δ, the bias based on the approximate nature of Equation (F.4) is increased. We find $\delta = 0.75$ to be a reasonable choice for this particular circuit, Hamiltonian and parameter position $\boldsymbol{\theta}$.

F.5 Decomposing effective generators for differentiation

In Algorithm 2 we suggested using the generalized parameter-shift rule [118, 116, 119] in order to compute the partial derivatives $\frac{\partial}{\partial\theta_l}E(\boldsymbol{\theta})$ independently. In addition, Theorem 3.3.2 bounds the number of frequencies occurring in the univariate auxiliary cost function $E(t) = \mathrm{Tr}\left\{U(\boldsymbol{\theta})e^{t\Omega_l(\boldsymbol{\theta})}\rho e^{-t\Omega_l(\boldsymbol{\theta})}U^\dagger(\boldsymbol{\theta})H\right\}$ and the corresponding number of parameter shifts required during differentiation.

Realizing the shift rule requires us to implement not only $U(\boldsymbol{\theta})$—which is necessary to compute $E(\boldsymbol{\theta})$ itself—but also the gate $e^{t_r\Omega_l(\boldsymbol{\theta})}$ for $2R$ shift values t_r and each Ω_l separately. Alternatively, we may follow the approach to decompose all effective generators Ω_l and compute the derivative as a linear combination of the derivatives for simpler auxiliary gates, similar to [118]. In particular, we again choose the Pauli basis $\mathcal{P}_n$ of $\mathfrak{su}(N)$ for this decomposition.

Decompose the effective generators $\Omega_l(\boldsymbol{\theta})$ as

$$\Omega_l(\boldsymbol{\theta}) = \sum_m \omega_{lm}(\boldsymbol{\theta})G_m, \qquad \omega_{lm}(\boldsymbol{\theta}) = \frac{1}{N}\mathrm{Tr}\left\{G_m\Omega_l(\boldsymbol{\theta})\right\}.$$

Note that the coefficients are purely imaginary due to the skew-Hermiticity of $\Omega_l(\boldsymbol{\theta})$. The partial derivative we are interested in can then be written as

$$\begin{aligned}
\frac{\partial}{\partial\theta_l}E(\boldsymbol{\theta}) &= \mathrm{Tr}\left\{HU(\boldsymbol{\theta})\left[\sum_{m=1}^{d}\omega_{lm}(\boldsymbol{\theta})G_m, \rho\right]U^\dagger(\boldsymbol{\theta})\right\} \\
&= \sum_m \omega_{lm}(\boldsymbol{\theta})\mathrm{Tr}\left\{HU(\boldsymbol{\theta})[G_m,\rho]U^\dagger(\boldsymbol{\theta})\right\} \\
&= \sum_m \omega_{lm}(\boldsymbol{\theta})2i\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Tr}\left\{HU(\boldsymbol{\theta})\left[\exp\left\{-i\frac{t}{2}G_m\right\}, \rho\right]U^\dagger(\boldsymbol{\theta})\right\}\Big|_{t=0} \\
&= \sum_m \widetilde{\omega}_{lm}(\boldsymbol{\theta})\frac{\mathrm{d}}{\mathrm{d}t}C_{G_m}(\boldsymbol{\theta},t)\big|_{t=0}.
\end{aligned}$$

Here we abbreviated $\widetilde{\omega}_{lm}(\boldsymbol{\theta}) = 2i\omega_{lm}(\boldsymbol{\theta})$ and wrote $C_{G_m}(\boldsymbol{\theta},t)$ for the cost function with a rotation gate with parameter $-t/2$ about G_m inserted before $U(\boldsymbol{\theta})$. This modified cost function can be differentiated with respect to t using the original two-term parameter-shift rule, as the inserted gate is generated by (the multiple of) a Pauli string.

The above linear combination of Pauli rotation derivatives can be reused for all partial derivatives, so that the full gradient for one $\mathbb{SU}(N)$ gate is given by

$$\nabla E(\boldsymbol{\theta}) = \widetilde{\omega}(\boldsymbol{\theta}) \cdot \mathbf{d}\boldsymbol{E},$$
$$\mathbf{d}\boldsymbol{E} = \begin{pmatrix} \frac{\mathrm{d}}{\mathrm{d}t} C_{G_1}(\boldsymbol{\theta}, t)\big|_{t=0} \\ \vdots \\ \frac{\mathrm{d}}{\mathrm{d}t} C_{G_d}(\boldsymbol{\theta}, t)\big|_{t=0} \end{pmatrix}.$$

So far we did not discuss the number of Pauli strings occurring in the decomposition of the generators Ω_l. As can be seen from Equation (3.36) and our definition of the DLA in Section 3.3.5, this number is bounded by the size of the DLA, and we again remark that this bound will be saturated for most values of $\boldsymbol{\theta}$. As two shifts are required for each Pauli rotation, the gradient $\nabla E(\boldsymbol{\theta})$ can thus be computed using $2\dim\langle A(\boldsymbol{\theta})\rangle_{\mathrm{Lie}}$ circuits, using Pauli rotations from the DLA and, e.g., shift angles $\pm\frac{\pi}{2}$.

Since we only required a linear decomposition of Ω_l, any other basis for the DLA may be used as well, potentially allowing for fewer shifted circuits or different inserted gates that may be more convenient to realize on hardware.

F.6 Gate speed limit

The following Lemmas are used in Section 3.3.4.

Lemma F.6.1. *For Hamiltonians of the form $H = \sum_m \theta_m G_m$, where G_m are strings of $\log_2 N$ Pauli operators, $Tr\{H^2\} = N\sum_m \theta_m^2$.*

Proof. All Pauli strings $G_m \in \mathcal{P}_n$ are orthonormal with respect to the trace inner product, $\mathrm{Tr}(G_m^\dagger G_n) = \delta_{n,m} N$. Using this gives

$$\begin{aligned} \mathrm{Tr}\left\{H^2\right\} &= \sum_{m,n} \theta_m \theta_n \mathrm{Tr}\{G_m G_n\} \\ &= N \sum_{m,n} \theta_m \theta_n \delta_{n,m} \\ &= N \sum_m \theta_m^2. \end{aligned}$$

□

Lemma F.6.2. *The length of a smooth curve on the Riemannian manifold* $\mathrm{SU}(N)$*, with metric* $g(x,y) = \mathrm{Tr}\{x^\dagger y\}$*, for a time-independent Hamiltonian* H *after a fixed time* τ *only depends on the norm of* H *and* τ*.*

Proof. The unitary evolution of $U(\boldsymbol{\theta};t)$, parameterized by t, corresponds to a smooth curve on $\mathrm{SU}(N)$ with length according to the Riemannian metric, $g(x,y) = \mathrm{Tr}\{x^\dagger y\}$ [368]. Integrating over the metric norm through the tangent spaces from $t = 0$ to final time τ gives the path length,

$$\begin{aligned} L[U(\boldsymbol{\theta};t),\tau] &= \int_{t=0}^{\tau} ds \\ &= \int_0^\tau \sqrt{g(\dot{U}(\boldsymbol{\theta};t),\dot{U}(\boldsymbol{\theta};t))}dt \\ &= \int_0^\tau \sqrt{\mathrm{Tr}\left\{\dot{U}^\dagger(\boldsymbol{\theta};t)\dot{U}(\boldsymbol{\theta};t)\right\}}dt, \end{aligned}$$

where $\dot{U} = \frac{dU}{dt}$. From Schrödinger evolution,

$$\frac{dU(\boldsymbol{\theta};t)}{dt} = -iHU(\boldsymbol{\theta};t), \tag{F.5}$$

we find

$$L[U(\boldsymbol{\theta};t),\tau] = \tau\sqrt{\mathrm{Tr}\left\{H^2\right\}}, \tag{F.6}$$

for time-independent Hamiltonians. Therefore, for all Hamiltonians with fixed norm $\mathrm{Tr}\{H^2\}$, the path distance travelled after time τ is the same regardless of the specific unitary evolution. □

Lemma F.6.3. *The minimal path between the identity element* I *and a point* V *on the Riemannian manifold of* $\mathrm{SU}(N)$*, with metric* $g(x,y) = \mathrm{Tr}\{x^\dagger y\}$*, is a geodesic curve* $\gamma(t) = e^{Xt}$ *with* $X \in \mathfrak{su}(N)$*.*

Proof. From Proposition 3.10 [201], the geodesics of $\mathrm{SU}(N)$ are the one-parameter subgroups, given by $\gamma(t) = e^{Xt}$. In general, multiple geodesics curves can give V from the identity. The minimal path is the curve with the minimum length. Since geodesic curves are the extrema of the path length functional [290, Lemma 9.3], there must exist a geodesic, which is of the form $\gamma(t) = e^{Xt}$, that is the minimal path to V. □

F.7 Proof of Theorem 3.3.1

We restate Theorem 3.3.1 again for convenience.

Theorem F.7.1. *For unitary gates generated by normalized time-independent Hamiltonians, consider a general circuit decomposition of two gates $U(\boldsymbol{\phi}^{(2)};t_2)U(\boldsymbol{\phi}^{(1)};t_1)$. There exists an equivalent evolution with an $\mathbb{SU}(N)$ gate $U(\boldsymbol{\theta};t_g) = U(\boldsymbol{\phi}^{(2)};t_2)U(\boldsymbol{\phi}^{(1)};t_1)$, with evolution time t_g, such that*

$$t_g \leq t_1 + t_2,$$

with equality if $\boldsymbol{\phi}^{(1)} + \boldsymbol{\phi}^{(1)} = \boldsymbol{\theta}$.

Proof. The product $U(\boldsymbol{\phi}^{(2)};t_2)U(\boldsymbol{\phi}^{(1)};t_1)$ corresponds to a specific point V on the manifold $\mathrm{SU}(N)$. By Lemma F.6.3, there exists a geodesic between the identity I and V, given by the curve e^{Xt} that is of minimal length. We can parameterize this geodesic as $U(\boldsymbol{\phi}^{(2)};t_g) = \exp\{\bar{A}(\boldsymbol{\theta})t_g\}$, which is always possible since $A(\boldsymbol{\theta})$ parameterizes an arbitrary point in $\mathfrak{su}(N)$ and is an $\mathbb{SU}(N)$ gate. By Lemma F.6.2, the length of this path only depends on the norm of $\bar{A}(\boldsymbol{\theta})$, which is 1, and on t_g, which gives

$$t_g = L[U(\boldsymbol{\theta};t), t_g)].$$

Since this path is minimal, we have

$$t_g \leq t_1 + t_2$$

with equality if $\boldsymbol{\phi}^{(1)} + \boldsymbol{\phi}^{(1)} = \boldsymbol{\theta}$.

□

F.7.1 Special case of $\mathrm{SU}(2)$

In the following we give the additional time for decomposing an optimal $\mathbb{SU}(2)$ gate into two gates. By *optimal*, we refer to the geodesic along the minimal path length curve – see Lemma F.6.3.

We consider the optimal $\mathbb{SU}(2)$ gate $U(\boldsymbol{\theta};t_g)$ with geodesic evolution time t_g together with a decomposition $U(\boldsymbol{\phi}^{(2)};t_2)U(\boldsymbol{\phi}^{(1)};t_1) = U(\boldsymbol{\theta};t_g)$. The decomposed circuit is given by two unitary evolutions. Each individual evolution $U(\tilde{\boldsymbol{\phi}}^{(\nu)};t_\nu)$ is a U(1) rotation such

that only a single basis element is required. With two rotations, the overall evolution is an element of SU(2). The corresponding $\mathfrak{su}(2)$ algebra is spanned by three basis elements—the three Pauli matrices for example. The two rotations can be represented as lying on a Bloch sphere. A unitary transformation, $K \in \mathrm{SU}(N)$, can therefore be applied to the evolution such that $U(\tilde{\boldsymbol{\phi}}^{(\nu)}; t_\nu) = KU(\boldsymbol{\phi}^{(\nu)}; t_\nu)K^\dagger$ for

$$\tilde{\boldsymbol{\phi}}^{(\nu)} = \begin{pmatrix} \sin(\alpha_\nu)\cos(\beta_\nu) \\ \sin(\alpha_\nu)\sin(\beta_\nu) \\ \cos(\alpha_\nu) \end{pmatrix},$$

where $\nu = 1, 2$, with α_ν and β_ν parameterizing the rotations. By construction $\tilde{\boldsymbol{\phi}}^{(\nu)} \cdot \tilde{\boldsymbol{\phi}}^{(\nu)} = 1$ is normalized for all parameters α_ν and β_ν. The same transformation K defines $U(\tilde{\boldsymbol{\theta}}; t_g) = KU(\boldsymbol{\theta}; t_g)K^\dagger$ and gives the same relationship $U(\tilde{\boldsymbol{\theta}}; t_g) = U(\tilde{\boldsymbol{\phi}}^{(2)}; t_2)U(\tilde{\boldsymbol{\phi}}^{(1)}; t_1)$. This is straightforward to show

$$\begin{aligned} U(\tilde{\boldsymbol{\theta}}; t_g) &= KU(\boldsymbol{\theta}; t_g)K^\dagger \\ &= KU(\boldsymbol{\phi}^{(2)}; t_2)U(\boldsymbol{\phi}^{(1)}; t_2)K^\dagger \\ &= KU(\boldsymbol{\phi}^{(2)}; t_2)K^\dagger KU(\boldsymbol{\phi}^{(1)}; t_2)K^\dagger \\ &= U(\tilde{\boldsymbol{\phi}}^{(2)}; t_2)U(\tilde{\boldsymbol{\phi}}^{(1)}; t_1). \end{aligned}$$

We have

$$\bar{A}(\tilde{\boldsymbol{\phi}}^{(\nu)}) = \sin(\alpha_\nu)\cos(\beta_\nu)G_1 + \sin(\alpha_\nu)\sin(\beta_\nu)G_2 + \cos(\alpha_\nu)G_3,$$

where we choose $G_1 = iX$, $G_2 = iY$, and $G_3 = iZ$. These basis elements of $\mathfrak{su}(2)$ generate the group SU(2). We also define the basis vector $\boldsymbol{G} = (G_1, G_2, G_3)$. Exponentiation therefore gives the closed-form expression

$$\begin{aligned} \exp\Big\{\bar{A}(\tilde{\boldsymbol{\phi}}^{(\nu)})t_\nu\Big\} &= \exp\Big\{\Big(\tilde{\boldsymbol{\phi}}^{(\nu)} \cdot \boldsymbol{G}\Big)\, t_\nu\Big\} \\ &= \cos(t_\nu)I^{\otimes N_{\mathrm{qubits}}} + \sin(t_\nu)\tilde{\boldsymbol{\phi}}^{(\nu)} \cdot \boldsymbol{G}. \end{aligned}$$

By the group composition law of SU(2), the product of two exponentials in SU(2) also gives a closed-form expression,

$$\exp\Big\{\bar{A}(\tilde{\boldsymbol{\phi}}^{(2)})t_2\Big\}\exp\Big\{\bar{A}(\tilde{\boldsymbol{\phi}}^{(1)})t_1\Big\} = \Big(\cos(t_2)I^{\otimes N_{\mathrm{qubits}}} + \sin(t_2)\tilde{\boldsymbol{\phi}}^{(2)} \cdot \boldsymbol{G}\Big)\Big(\cos(t_1)I^{\otimes N_{\mathrm{qubits}}} + \sin(t_1)\tilde{\boldsymbol{\phi}}^{(1)} \cdot \boldsymbol{G}\Big)$$

Collecting terms gives

$$\exp\Big\{\bar{A}(\tilde{\boldsymbol{\phi}}^{(2)})t_2\Big\}\exp\Big\{\bar{A}(\tilde{\boldsymbol{\phi}}^{(1)})t_1\Big\} = \Big(\cos(t_1)\cos(t_2) - \tilde{\boldsymbol{\phi}}^{(1)}\cdot\tilde{\boldsymbol{\phi}}^{(2)}\sin(t_1)\sin(t_2)\Big)\, I^{\otimes N_{\text{qubits}}} + \Big(\cos(t_2)\sin(t_1)\tilde{\boldsymbol{\phi}}^{(1)} + \cos(t_1)\sin(t_2)\tilde{\boldsymbol{\phi}}^{(2)} + i\sin(t_1)\sin(t_2)\tilde{\boldsymbol{\phi}}^{(1)}\times\tilde{\boldsymbol{\phi}}^{(2)}\Big)\cdot\boldsymbol{G}. \quad \text{(F.7)}$$

The total evolution is

$$\exp\Big\{\bar{A}(\tilde{\boldsymbol{\phi}}^{(2)})t_2\Big\}\exp\Big\{\bar{A}(\tilde{\boldsymbol{\phi}}^{(1)})t_1\Big\} = \exp\Big\{\bar{A}(\tilde{\boldsymbol{\theta}})t_g\Big\} \quad \text{(F.8)}$$

$$= \cos(t_g)I^{\otimes N_{\text{qubits}}} + \sin(t_g)\tilde{\boldsymbol{\theta}}\cdot\boldsymbol{G}. \quad \text{(F.9)}$$

By comparison of Equations F.7 and F.9, we find

$$\tilde{\boldsymbol{\theta}} = \frac{1}{\sin(t_g)}\Big(\cos(t_2)\sin(t_1)\tilde{\boldsymbol{\phi}}^{(1)} + \cos(t_1)\sin(t_2)\tilde{\boldsymbol{\phi}}^{(2)} + i\sin(t_1)\sin(t_2)\tilde{\boldsymbol{\phi}}^{(1)}\times\tilde{\boldsymbol{\phi}}^{(2)}\Big),$$

and

$$\cos(t_g) = \cos(t_1)\cos(t_2) - \tilde{\boldsymbol{\phi}}^{(1)}\cdot\tilde{\boldsymbol{\phi}}^{(2)}\sin(t_1)\sin(t_2).$$

The additional evolution time is $\Delta t = t_d - t_g$, with $t_d = t_1 + t_2$ the total decomposed unitary evolution time. Due to the invariance of the scalar product, $\tilde{\boldsymbol{\phi}}^{(1)}\cdot\tilde{\boldsymbol{\phi}}^{(2)} = \boldsymbol{\phi}^{(1)}\cdot\boldsymbol{\phi}^{(2)}$, the additional time $\Delta t = t_d - t_g$ required by the decomposition is then given by

$$\Delta t = t_d - \arccos\big(\cos(t_1)\cos(t_2) - \boldsymbol{\phi}^{(1)}\cdot\boldsymbol{\phi}^{(2)}\sin(t_1)\sin(t_2)\big) \geq 0.$$

F.8 Unique spectral gaps of Dynamical Lie Algebras

F.8.1 Proof of Theorem 3.3.2

We restate Theorem 3.3.2 here for convenience.

Theorem F.8.1. *The number of unique spectral gaps R of $\Omega_l(\boldsymbol{\theta})$ is upper bounded by the number of roots $|\Phi|$ of any maximal semisimple DLA,*

$$R \leq |\Phi|/2.$$

In the following we set $\mathfrak{g}$ to be a semisimple Lie algebra. A subspace $\mathfrak{a} \subseteq \mathfrak{g}$ is called a subalgebra if it is closed under the Lie bracket, i.e., if $[a_1, a_2] \in \mathfrak{a}$, $\forall a_1, a_2 \in \mathfrak{a}$. Since $\mathfrak{g}$ is a semisimple Lie algebra, it always contains a subalgebra called a Cartan subalgebra [250] (Chapter 7, Definition 7.10).

Definition F.8.1. *A Cartan subalgebra $\mathfrak{h}$ of $\mathfrak{g}$ is a subalgebra that satisfies the following conditions:*

1. *For all $h_1, h_2 \in \mathfrak{h}$, $[h_1, h_2] = 0$.*
2. *For all $x \in \mathfrak{g}$, if $[h, x] = 0$ for all $h \in \mathfrak{h}$, then $x \in \mathfrak{h}$.*

The first condition tells us that $\mathfrak{h}$ is a commutative subalgebra of $\mathfrak{g}$, while the second condition says that $\mathfrak{h}$ is maximal, i.e., there is no larger commutative subalgebra. The first step in proving Theorem 3.3.2 is to make use of the following result:

Theorem F.8.2. *[273, Chapter VI, Theorem 1]. If $\mathfrak{g}$ is a semisimple Lie algebra, we can write $\mathfrak{g}$ as a direct sum of the root spaces $\mathfrak{g}_\alpha$:*

$$\mathfrak{g} = \bigoplus_\alpha \mathfrak{g}_\alpha,$$

where

$$\mathfrak{g}_{\alpha \in \mathfrak{h}^*} = \{x \in \mathfrak{g} | \mathrm{ad}_h(x) = \alpha(h)x, \, \forall h \in \mathfrak{h}\},$$

and $\alpha \in \mathfrak{h}^*$ are functionals on $\mathfrak{h}$. That is, a root space is a subspace of $\mathfrak{g}$ on which the action of the adjoint representation of $\mathfrak{h}$ is described by a functional (and scalar multiplication).

The above decomposition is called a root space decomposition, which is an essential tool in classifications of Lie algebras [274, 273]. Since

$$\mathfrak{g}_0 = \{x \in \mathfrak{g} | \mathrm{ad}_h(x) = 0, \forall h \in \mathfrak{h}\},$$

we find that $\mathfrak{h} = \mathfrak{g}_0$ and hence

$$\mathfrak{g} = \mathfrak{h} \oplus \bigoplus_{\alpha \neq 0} \mathfrak{g}_\alpha.$$

We then immediately see that

$$\dim \mathfrak{g} = \dim \mathfrak{h} + \sum_{\alpha \neq 0} \dim \mathfrak{g}_\alpha.$$

We can thus relate the dimensionality of a Lie algebra to the dimensionality of its Cartan subalgebra and its weight spaces. The second step of the proof relies on identifying the unique spectral gaps of $\Omega_l(\boldsymbol{\theta})$ with the weight spaces $\mathfrak{g}_\alpha$. To achieve this, we will construct the linear operator ad_h and apply it to the eigenbasis of $\Omega_l(\boldsymbol{\theta})$ to show that the maps α can be identified with the spectral gaps of $\Omega_l(\boldsymbol{\theta})$.

Consider an element $\Omega \in \mathfrak{g}$, where $\mathfrak{g} \subseteq \mathfrak{su}(N)$ is a non-trivial subalgebra and $\mathfrak{h}$ is a Cartan subalgebra of $\mathfrak{g}$. Since $\mathfrak{h}$ is the Lie algebra of a maximally Abelian group, we can represent elements of $\mathfrak{h}$ by diagonal matrices. Since Ω is skew-Hermitian, there exists a unitary $V \in \mathrm{SU}(N)$ that diagonalizes Ω, i.e., $V^\dagger \Omega V = h$ with $h \in \mathfrak{h}$. Here, V is the matrix with columns equal to the eigenvectors v_k of Ω with corresponding eigenvalues λ_k. We can thus always choose a basis for $\mathfrak{g}$ such that Ω is diagonal. If Ω is non-degenerate, then it must be full rank, and therefore an element of $\mathfrak{h}$. All Cartan algebras are equivalent up to conjugacy, hence we can choose the matrix h to be the diagonal matrix containing the eigenvalues of Ω to represent the Cartan subalgebra $\mathfrak{h}$. We now take E_{nm} to be the matrix with entries (n, m) equal to 1 and all other entries to 0. Define the operator

$$e_{nm} = V^\dagger E_{nm} V,$$

and apply ad_h to it:

$$\begin{aligned}
\mathrm{ad}_h(e_{nm}) &= h e_{nm} - e_{nm} h \\
&= V^\dagger \Omega E_{nm} V - V^\dagger E_{nm} \Omega V \\
&= V^\dagger h E_{nm} V - V^\dagger E_{nm} h V \\
&= (\lambda_n - \lambda_m) e_{nm}.
\end{aligned}$$

This means that ad_h has the eigenvectors e_{nm} with corresponding eigenvalues $\alpha_{nm}(h) = \lambda_n - \lambda_m$ [369], and so we have identified the eigenvalue differences with the roots of the Lie algebra. We define the set of all roots as

$$\Phi = \{\lambda_n - \lambda_m, n \neq m = 1, \ldots, N\}.$$

Since the dimensionality of each weight space is one [273, Chapter VI, Theorem 2(a)], we can see that

$$\sum_{\alpha \neq 0} \dim \mathfrak{g}_\alpha = |\Phi|.$$

Therefore,

$$\dim \mathfrak{g} = \dim \mathfrak{h} + |\Phi|.$$

We now set $\mathfrak{g} = \langle A(\boldsymbol{\theta})\rangle_{\text{Lie}}$. If we take the absolute value of the elements of Φ, we can identify $R = |\Phi|/2$, where the factor $1/2$ is to account for double the counting of the spectral gaps. Since Ω can be degenerate in general, we obtain the inequality $R \leq |\Phi|/2$. With this, the proof of Theorem 3.3.2 is completed.

F.8.2 Examples

Here, we give several examples of maximal DLAs and their corresponding value of $|\Phi|/2$. Analogous to the main text, we choose the Pauli representation, but these results should hold for any irreducible representation of $\mathfrak{su}(N)$.

1. $\mathfrak{su}(2)$: For a 1-qubit system, there are no non-trivial subalgebras, hence we can only look at the full special unitary Lie algebra $\mathfrak{su}(2)$. Any $A(\boldsymbol{\theta})$ that consists of two Pauli operators will generate this algebra, e.g.,

   $$A(\boldsymbol{\theta}) = i(\theta_1 X + \theta_2 Y)$$

   will give $\langle A(\boldsymbol{\theta})\rangle_{\text{Lie}} = \mathfrak{su}(2)$. A Cartan subalgebra of $\mathfrak{su}(2)$ is given by $\mathfrak{h} = \text{span}Z$. We therefore find that $\dim \mathfrak{g} = 3$ and $\dim \mathfrak{h} = 1$ and so $|\Phi| = 2$. Hence, we have $R \leq 1$ and need $2R \leq 2$ shifts. This matches the result in [31], where the parameter-shift rule was generalized from single Pauli matrices to Hermitian operators with two unique eigenvalues.

2. TFIM: A DLA that has been studied before [227, 228] is the 1D transverse field Ising-Model (TFIM) Hamiltonian:

   $$A(\boldsymbol{\theta}) = i(\theta_1 X \otimes I + \theta_2 I \otimes X + \theta_3 Z \otimes Z),$$

   with $\langle A(\boldsymbol{\theta})\rangle_{\text{Lie}} = \text{span}X \otimes I, I \otimes X, Y \otimes Y, Z \otimes Z, Z \otimes Y, Y \otimes Z$. We can take $\mathfrak{h} = \text{span}X \otimes I, I \otimes X$ as a Cartan subalgebra and so $\dim \mathfrak{g} = 6$ and $\dim \mathfrak{h} = 2$, which gives $|\Phi| = 4$. Hence, we need (at most) 4 shifts to obtain the gradient of an operator in the DLA of the TFIM, which corresponds to $\mathfrak{so}(4)$.

3. $\mathfrak{su}(4)$: The full Lie algebra of $\mathfrak{su}(4)$ is spanned by

   $$A(\boldsymbol{\theta}) = \sum_m \theta_m G_m,$$

   where the G_m are the tensor products of Pauli $\mathcal{P}_2$ multiplied by i. A Cartan subalgebra of $\mathfrak{su}(4)$ is given by $\mathfrak{h} = \{Z \otimes I, I \otimes Z, Z \otimes Z\}$. This means that $\dim \mathfrak{g} = 15$ and $\dim \mathfrak{h} = 3$, which gives $|\Phi| = 12$. Hence, we have $R = 6$ and need 12 shifts to obtain the gradient for a general operator in $\mathfrak{su}(4)$.

In the above examples, we have only been concerned with the dimensionality of the root system. We could go one step further and look at the structure of the root systems. It turns out that there exists only a finite set of root systems, which leads to the classification of all semisimple Lie algebras (such a program was originally carried out by Dynkin [274] and is explained in most textbooks on Lie algebras [176, 369, 250]). This allows us to make the following observation about DLAs and the $\mathbb{SU}(N)$ gates in our work: there is a finite number of families of $\mathbb{SU}(N)$ gates for each N, given by the possible DLAs. Again, we emphasize that this is independent of the representation of the algebra. We summarize the above results together with the identification of the corresponding classical group in Table F.1 [176] (Chapter 3, Table 3.4).

Name	$\dim(\mathfrak{g})$	$\dim(\mathfrak{h})$	$\lvert\Phi\rvert$	Classical group
$\mathfrak{su}(2)$	3	1	2	A_1
$\mathfrak{so}(4)$	6	2	4	$A_1 \times A_1 \cong D_2$
$\mathfrak{su}(4)$	15	3	12	A_2

Table F.1: **Examples of DLAs and the size of the root spaces.** Each root system Φ can be identified with a Lie algebra of one of the classical groups A_n, B_n, C_n, D_n. The classical group D_2 corresponds to SO(4), with the corresponding Lie algebra $\mathfrak{so}(4)$ which has dimension $N(N-1)/2$.

F.9 Errors due to truncation

Let $\boldsymbol{\theta}$, $\boldsymbol{\theta}' \in \mathbb{R}^{N^2-1}$ and $\boldsymbol{\theta} - \boldsymbol{\theta}' = \boldsymbol{\delta}$. We can consider the case where the difference (due to truncating the support) $\boldsymbol{\delta}$ is small. This could save the number of gradient evaluations by discarding irrelevant directions in $\Omega_l(\boldsymbol{\theta})$.

Take $\{G_m\}$ to be a basis for $\mathfrak{su}(N)$ and $\boldsymbol{G} = (G_1, \ldots, G_N)$. Define $A \equiv \boldsymbol{\theta} \cdot \boldsymbol{G}$. Then,

$$\mathrm{Tr}\{A^2\} = \frac{1}{N}\mathrm{Tr}\left\{\sum_{i,j=1}^{N^2-1} \theta_i\theta_j G_i G_j\right\} \tag{F.10}$$

$$= \frac{1}{N}\sum_{i,j=1}^{N^2-1} \theta_i\theta_j \mathrm{Tr}\left\{G_i G_j\right\} \tag{F.11}$$

$$= \sum_{i,j=1}^{N^2-1} \theta_i\theta_j\delta_{ij} = \|\boldsymbol{\theta}\|_2^2 \tag{F.12}$$

Given the above parameterization, we are interested in bounding the difference between the two generated unitaries as a function of $\|\boldsymbol{\delta}\|$. The following is due to [370]. Let $\gamma : [0,1] \to \mathbb{C}^{n\times n}$ be a smooth curve and f a function that has a derivative. Then

$$f(\gamma(1)) - f(\gamma(0)) = \int_0^1 Df(\gamma(t))\gamma'(t)dt$$

where $Df(x)$ is the Frechet derivative. Given the curve $\gamma(t) = (1-t)a + tb$ for $t \in [0,1]$ we find

$$\|f(b) - f(a)\| = \left(\sup_{x\in[a,b]} Df(x)\right)\|b-a\|$$

We now consider the function $f(X) = \exp\{X\} \in \mathrm{SU}(N)$. We consider the curve $\gamma(t) = (1-t)X + tY$. Clearly we have

$$\gamma(0) = X, \quad \gamma(1) = X, \quad \dot{\gamma}(t) = -X + Y \equiv H$$

Hence we see that $\gamma(t) = X + tH$ Using the fact that

$$Df(X)(H) = \frac{d}{dt}(f\circ\gamma)(t)\Big|_{t=0}$$

for $\gamma(0) = X$ and $\dot{\gamma}(0) = H$ we find

$$Df(X)(H) = \frac{d}{dt}e^{X+tH}\Big|_{t=0}$$

Let $H \in T_X\mathfrak{su}(N)$. We use the integral identity of [258] (see also Appendix F.3). Consider the function

$$g(s) = e^{-sX}\frac{\partial}{\partial t}e^{s(X+tH)}\Big|_{t=0}$$

Taking the partial derivative with respect to s, we find

$$\begin{aligned}
g'(s) &= -Xe^{-sX}\frac{\partial}{\partial t}e^{s(X+tH)}\Big|_{t=0} + e^{-sX}\frac{\partial}{\partial t}(X+tH)e^{s(X+tH)} \\
&= -Xe^{-sX}\frac{\partial}{\partial t}e^{s(X+tH)}\Big|_{t=0} + Xe^{-sX}\frac{\partial}{\partial t}e^{s(X+tH)}\Big|_{t=0} \\
&+ e^{-sX}He^{sX} + e^{-sX}tH\frac{\partial}{\partial t}e^{s(X+tH)}\Big|_{t=0} \\
&= e^{-sX}He^{sX}
\end{aligned}$$

Using the fundamental theorem of calculus,

$$g(1) - g(0) = \int_0^1 ds\, g'(s)$$

where

$$g(0) = 0, \quad g(1) = e^{-X} \frac{\partial}{\partial t} e^{X+tH}$$

We then find

$$e^{-X} \frac{d}{dt} e^{X+tH} \bigg|_{t=0} = \int_0^1 ds\, e^{-sX} H e^{sX}$$

so

$$Df(X)(H) = \int_0^1 ds\, e^{(1-s)X} H e^{sX}$$

If we take the velocity $H = I$ and use the Cauchy-Schwarz inequality, we get

$$\|Df(X)(I)\| \leq \int_0^1 ds\, \left\|e^{(1-s)X}\right\| \left\|e^{sX}\right\|$$

by bringing the norm into the integral. Using the fact that iX is self-adjoint, we find

$$\left\|e^{sX}\right\| \leq e^{s\alpha}$$

where α is the largest eigenvalue of X and Y (Note that α is maximal at the end points of the curve $\gamma(t)$). Plugging this into the integral, we get

$$\|Df(X)(I)\| \leq e^{\alpha}$$

Hence

$$\left\|e^{X} - e^{Y}\right\| \leq e^{\alpha} \|X - Y\|$$

Plugging in our equations then gives

$$\left\|e^{\boldsymbol{\theta G} + \boldsymbol{\delta G}} - e^{\boldsymbol{\theta G}}\right\| \leq e^{\alpha} \|\boldsymbol{\delta}\|$$

Hence for small perturbations $\boldsymbol{\delta}$ the loss of unitary fidelity is manageable.

Appendix G

Additional details DLA classification

G.1 Involutions of $\mathfrak{su}(2^n)$

In this subsection, we explain how we can describe subalgebras of $\mathfrak{su}(2^n)$ as fixed points of involutions. We start more generally by recalling that an *isomorphism* from a Lie algebra $\mathfrak{g}$ to another Lie algebra $\mathfrak{g}_1$ is an invertible linear transformation $\varphi\colon \mathfrak{g} \to \mathfrak{g}_1$ that is compatible with the bracket, i.e., $\varphi([a,b]) = [\varphi(a), \varphi(b)]$ for all $a, b \in \mathfrak{g}$. We write $\mathfrak{g} \cong \mathfrak{g}_1$ to indicate that $\mathfrak{g}$ is isomorphic to $\mathfrak{g}_1$. An *automorphism* of $\mathfrak{g}$ is an isomorphism $\varphi\colon \mathfrak{g} \to \mathfrak{g}$. The set of fixed points of φ is defined as:

$$\mathfrak{g}^{\varphi} = \big\{a \in \mathfrak{g} \,\big|\, \varphi(a) = a\big\}. \tag{G.1}$$

It is easy to check that $\mathfrak{g}^{\varphi}$ is a subalgebra of $\mathfrak{g}$, called the *fixed-point subalgebra*. An *involution* on $\mathfrak{g}$ is an automorphism $\theta\colon \mathfrak{g} \to \mathfrak{g}$ with the property that $\theta(\theta(a)) = a$ for all $a \in \mathfrak{g}$, i.e., $\theta^{-1} = \theta$.

Later, we will need to understand how, for an involution θ, the fixed-point subalgebra $\mathfrak{g}^{\theta}$ transforms under another automorphism φ of $\mathfrak{g}$. The answer is given in the following lemma.

Lemma G.1.1. *Let φ be an automorphism of a Lie algebra $\mathfrak{g}$, and θ be an involution of $\mathfrak{g}$. Then $\varphi\theta\varphi^{-1}$ is an involution of $\mathfrak{g}$, and we have $\varphi(\mathfrak{g}^{\theta}) = \mathfrak{g}^{\varphi\theta\varphi^{-1}}$.*

Proof. Note that $\varphi\theta\varphi^{-1}$ is an automorphism of $\mathfrak{g}$, since the composition of isomorphisms is again an isomorphism. It is an involution of $\mathfrak{g}$, because $(\varphi\theta\varphi^{-1})^{-1} = \varphi\theta^{-1}\varphi^{-1} = \varphi\theta\varphi^{-1}$.

To check the claim about the fixed points, suppose that $a \in \mathfrak{g}^\theta$, i.e., $\theta(a) = a$. Then

$$(\varphi\theta\varphi^{-1})\varphi(a) = \varphi(\theta(a)) = \varphi(a) \quad \Rightarrow \quad \varphi(a) \in \mathfrak{g}^{\varphi\theta\varphi^{-1}}.$$

Conversely, if $b \in \mathfrak{g}^{\varphi\theta\varphi^{-1}}$, then the above calculation shows that $a = \varphi^{-1}(b) \in \mathfrak{g}^\theta$. □

Now we will discuss how to construct automorphisms and involutions of $\mathfrak{su}(N)$.

Lemma G.1.2. *Suppose that U and Q are unitary $N \times N$ matrices, and $Q^T = \pm Q$. Then the formulas*

$$\varphi(a) = UaU^\dagger, \qquad \theta(a) = -Qa^TQ^\dagger, \qquad a \in \mathfrak{su}(N),$$

define an automorphism φ and an involution θ of $\mathfrak{su}(N)$. Moreover, we have

$$(\varphi\theta\varphi^{-1})(a) = -(UQU^T)a^T(UQU^T)^\dagger, \qquad a \in \mathfrak{su}(N).$$

Proof. First, note that φ is invertible with $\varphi^{-1}(a) = U^\dagger aU$. It is clear that $\varphi([a,b]) = [\varphi(a), \varphi(b)]$, because

$$\varphi(a)\varphi(b) = UaU^\dagger UbU^\dagger = UabU^\dagger = \varphi(ab)$$

for any two matrices a, b. (This means that φ is an automorphism of the associative algebra $\mathbb{C}^{N\times N}$.)

To check that θ is an involution, we calculate

$$\theta(\theta(a)) = \theta(-Qa^TQ^\dagger) = Q(Q^\dagger)^T(a^T)^TQ^TQ^\dagger = Q(\pm Q^\dagger)a(\pm Q)Q^\dagger = a,$$

where we used that $(Q^\dagger)^T = (Q^T)^\dagger$. Next, we have

$$\theta(ab) = -Q(ab)^TQ^\dagger = -Qb^Ta^TQ^\dagger = -Qb^TQ^\dagger Qa^TQ^\dagger = -\theta(b)\theta(a).$$

This implies that $\theta([a,b]) = [\theta(a), \theta(b)]$ and proves that θ is an involution of $\mathfrak{su}(N)$.

Finally, we find

$$\begin{aligned}(\varphi\theta\varphi^{-1})(a) &= \varphi\big(\theta(U^\dagger aU)\big) \\ &= -\varphi\big(QU^Ta^T(U^\dagger)^TQ^\dagger\big) \\ &= -UQU^Ta^T(U^\dagger)^TQ^\dagger U^\dagger \\ &= -(UQU^T)a^T(UQU^T)^\dagger,\end{aligned}$$

as claimed. □

Example G.1.1. The subalgebra $\mathfrak{so}(N) \subset \mathfrak{su}(N)$ is the fixed-point subalgebra of the involution $a \mapsto -a^T$ (see Equation (4.8)). Similarly, we see from Equation (4.9) that $\mathfrak{sp}(N) = \mathfrak{su}(2N)^\theta$, where $\theta(a) = -Qa^TQ^\dagger$ as in Lemma G.1.2, with $Q = -iJ_{2N}$.

It is well known, in general, that for an involution $\theta(a) = -Qa^TQ^\dagger$ as in Lemma G.1.2, the fixed-point subalgebra $\mathfrak{su}(N)^\theta \cong \mathfrak{so}(N)$ when $Q^T = Q$, and $\mathfrak{su}(N)^\theta \cong \mathfrak{sp}(N/2)$ when $Q^T = -Q$ (in which case N must be even). For completeness, we will present the proof of these facts in the special case of interest to us: when $N = 2^n$ and $Q \in \mathcal{P}_n$ is a length-n Pauli string. Note that all Pauli strings Q satisfy $Q = Q^\dagger = Q^{-1}$ and $Q^T = \pm Q$.

Lemma G.1.3. *For any Pauli string $Q \in \mathcal{P}_n$, there exists a unitary $2^n \times 2^n$ matrix U such that*

$$UQU^T = \begin{cases} I^{\otimes n}, & \text{if } Q^T = Q, \\ Y_1, & \text{if } Q^T = -Q. \end{cases}$$

Proof. In the case $Q^T = Q$, we let $U = e^{i\frac{\pi}{4}Q}$. Then

$$UQU^T = UQU = U^2Q = e^{i\frac{\pi}{2}Q}Q = iQ \cdot Q = iI^{\otimes n},$$

where we used Euler's formula Equation (A.4). The superfluous phase i can be eliminated by applying the unitary transformation $V = e^{-i\frac{\pi}{4}I^{\otimes n}}$, which satisfies $V^T = V$ and $V^2 = -iI^{\otimes n}$.

Suppose now that $Q^T = -Q$, which means that Q contains an odd number of Y's. If Q has a Y in j-th position, let $P = Y_j \cdot Q$ and $U = e^{i\frac{\pi}{4}P}$. Note that

$$P = Y_j \cdot Q = Q \cdot Y_j \quad \Rightarrow \quad P \cdot Q = Q \cdot P = Y_j$$

and

$$P^T = Q^T \cdot Y_j^T = (-Q) \cdot (-Y_j) = Q \cdot Y_j = P.$$

Hence, as above, we find:

$$UQU^T = UQU = U^2Q = e^{i\frac{\pi}{2}P}Q = iP \cdot Q = iY_j.$$

If $j = 1$, we are done (after eliminating the phase i). If $j \neq 1$, we apply the unitary transformation $e^{-i\frac{\pi}{4}Y_1Y_j}$ and obtain:

$$e^{-i\frac{\pi}{4}Y_1Y_j}(iY_j)(e^{-i\frac{\pi}{4}Y_1Y_j})^T = e^{-i\frac{\pi}{2}Y_1Y_j}(iY_j) = -iY_1Y_j \cdot (iY_j) = Y_1,$$

completing the proof. □

Corollary G.1.1. *Any Pauli string $Q \in \mathcal{P}_n$ defines an involution θ of $\mathfrak{su}(2^n)$ given by $\theta(a) = -Qa^TQ$. The fixed-point subalgebra of this involution is:*

$$\mathfrak{su}(2^n)^\theta \cong \begin{cases} \mathfrak{so}(2^n), & \text{if } Q^T = Q, \\ \mathfrak{sp}(2^{n-1}), & \text{if } Q^T = -Q. \end{cases}$$

G.2 Statement of Results

G.2.1 Subalgebras of $\mathfrak{su}(4)$ up to symmetry

Recall that $\mathfrak{su}(4)$ has a basis over $\mathbb{R}$ consisting of all possible tensor products iAB, where $A, B \in \{I, X, Y, Z\}$, $AB \neq II$. As a first step, we found that there are 202 subalgebras of $\mathfrak{su}(4)$, which are generated by subsets of this basis. The symmetry group $S_3 \times \mathbb{Z}_2$ acts on $\mathfrak{su}(4)$ as follows: the symmetric group S_3 permutes simultaneously all $\{X, Y, Z\}$, while the non-identity element of $\mathbb{Z}_2$ acts as the flip $AB \rightleftharpoons BA$. We examined the orbits of the action of $S_3 \times \mathbb{Z}_2$ on the 202 subalgebras, and found that there are 36 orbits, which are listed in Table G.1 below. The full list of all 202 subalgebras is presented in the next subsection.

The numbers s, p, e, d are equal to the numbers of: single Paulis (such as XI), single Pauli pairs (such as XI, IX), double equal Paulis (such as XX), and double different Paulis (such as XY), respectively, in the basis of the subalgebra. These are invariant under the action of the symmetry group; hence, subalgebras of $\mathfrak{su}(4)$ with different invariants are not equivalent to each other. It turns out that the only two non-equivalent subalgebras with the same invariants are $\mathfrak{a}_2$ and $\mathfrak{a}_5$.

We distinguish between three types of subalgebras. The $\mathfrak{a}$-type Lie algebras are generated by products of two Paulis both different from the identity, whereas the $\mathfrak{b}$-type can be generated by Pauli strings that contain the identity. The $\mathfrak{c}$-type Lie algebras are an edge case where the generators contain some Pauli strings of the form $a \otimes I$ without the corresponding term $I \otimes a$. These Lie algebras will behave like the $\mathfrak{b}$-type Lie algebras but include a boundary effect at the last site in the chain. In particular, note that all single Pauli generators come in pairs such as XI, IX or YI, IY or ZI, IZ due to the translation invariance; hence we exclude the Lie algebras $\mathfrak{c}_0, \ldots, \mathfrak{c}_7$ from our classification. However, we include them in the following tables for completeness.

Label	Basis	dim	Stabilizer	Orbit	(s, p, e, d)
$\mathfrak{a}_0$	XX	1	4	3	(0,0,1,0)
$\mathfrak{a}_1$	XY	1	2	6	(0,0,0,1)
$\mathfrak{a}_2$	XY, YX	2	4	3	(0,0,0,2)
$\mathfrak{a}_3$	XX, YZ	2	2	6	(0,0,1,1)
$\mathfrak{a}_4$	XX, YY	2	4	3	(0,0,2,0)
$\mathfrak{a}_5$	XY, YZ	2	2	6	(0,0,0,2)
$\mathfrak{a}_6$	XX, YZ, ZY	3	4	3	(0,0,1,2)
$\mathfrak{a}_7$	XX, YY, ZZ	3	12	1	(0,0,3,0)
$\mathfrak{a}_8$	XX, XZ, IY	3	1	12	(1,0,1,1)
$\mathfrak{a}_9$	XY, XZ, IX	3	2	6	(1,0,0,2)
$\mathfrak{a}_{10}$	XY, YZ, ZX	3	6	2	(0,0,0,3)
$\mathfrak{a}_{11}$	XY, YX, YZ, IY	4	1	12	(1,0,0,3)
$\mathfrak{a}_{12}$	XX, XY, YZ, IZ	4	1	12	(1,0,1,2)
$\mathfrak{a}_{13}$	XX, YY, YZ, IX	4	1	12	(1,0,2,1)
$\mathfrak{a}_{14}$	XX, YY, XY, YX, ZI, IZ	6	4	3	(2,1,2,2)
$\mathfrak{a}_{15}$	XX, XY, XZ, IX, IY, IZ	6	2	6	(3,0,1,2)
$\mathfrak{a}_{16}$	XY, YX, YZ, ZY, YI, IY	6	4	3	(2,1,0,4)
$\mathfrak{a}_{17}$	XX, XY, ZX, ZY, YI, IZ	6	2	6	(2,0,1,3)
$\mathfrak{a}_{18}$	XX, YY, XZ, ZY, XI, IY	6	2	6	(2,0,2,2)
$\mathfrak{a}_{19}$	$XX, XY, ZX, ZY, YZ, YI, IZ$	7	2	6	(2,0,1,4)
$\mathfrak{a}_{20}$	$XX, YY, ZZ, YZ, ZY, XI, IX$	7	4	3	(2,1,3,2)
$\mathfrak{a}_{21}$	$XX, YY, XY, YX, ZX, ZY, XI, YI, ZI, IZ$	10	2	6	(4,1,2,4)
$\mathfrak{a}_{22}$	all Paulis except II	15	12	1	(6,3,3,6)
$\mathfrak{b}_0$	XI, IX	2	4	3	(2,1,0,0)
$\mathfrak{b}_1$	XX, XI, IX	3	4	3	(2,1,1,0)
$\mathfrak{b}_2$	XY, XZ, XI, IX	4	2	6	(2,1,0,2)
$\mathfrak{b}_3$	XI, YI, ZI, IX, IY, IZ	6	12	1	(6,3,0,0)
$\mathfrak{b}_4$	$XX, XY, XZ, XI, IX, IY, IZ$	7	2	6	(4,1,1,2)
$\mathfrak{c}_0$	XI	1	2	6	(1,0,0,0)
$\mathfrak{c}_1$	XY, XI	2	1	12	(1,0,0,1)
$\mathfrak{c}_2$	XX, XI	2	2	6	(1,0,1,0)
$\mathfrak{c}_3$	XI, IY	2	2	6	(2,0,0,0)
$\mathfrak{c}_4$	XY, XI, IY	3	2	6	(2,0,0,1)
$\mathfrak{c}_5$	XI, YI, ZI	3	6	2	(3,0,0,0)
$\mathfrak{c}_6$	XX, XY, XI, IZ	4	1	12	(2,0,1,1)
$\mathfrak{c}_7$	XI, IX, YI, ZI	4	2	6	(4,1,0,0)

Table G.1: **List of all subalgebras of $\mathfrak{su}(4)$ up to symmetry $S_3 \times \mathbb{Z}_2$.** For each subalgebra, we have listed: a basis (over $\mathbb{R}$, after multiplication by i), its dimension, the order of the stabilizer, the order of the orbit under the action of $S_3 \times \mathbb{Z}_2$, and the invariants s, p, e, d. Note that the orders of all orbits add up to 202.

For each subalgebra of $\mathfrak{su}(4)$, we list below a minimal set of generators and the isomorphism class of the Lie algebra. Commuting direct summands are denoted with $\oplus$. Note also that $\mathfrak{so}(4) \cong \mathfrak{su}(2) \oplus \mathfrak{su}(2)$.

$$
\begin{aligned}
\mathfrak{a}_0 &= \langle XX \rangle_{\text{Lie}} \cong \mathfrak{u}(1), \\
\mathfrak{a}_1 &= \langle XY \rangle_{\text{Lie}} \cong \mathfrak{u}(1), \\
\mathfrak{a}_2 &= \langle XY, YX \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{a}_3 &= \langle XX, YZ \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{a}_4 &= \langle XX, YY \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{a}_5 &= \langle XY, YZ \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{a}_6 &= \langle XX, YZ, ZY \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{a}_7 &= \langle XX, YY, ZZ \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{a}_8 &= \langle XX, XZ \rangle_{\text{Lie}} \cong \mathfrak{su}(2), \\
\mathfrak{a}_9 &= \langle XY, XZ \rangle_{\text{Lie}} \cong \mathfrak{su}(2), \\
\mathfrak{a}_{10} &= \langle XY, YZ, ZX \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{a}_{11} &= \langle XY, YX, YZ \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{u}(1), \\
\mathfrak{a}_{12} &= \langle XX, XY, YZ \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{u}(1), \\
\mathfrak{a}_{13} &= \langle YY, YZ, XX \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{u}(1), \\
\mathfrak{a}_{14} &= \langle XX, YY, XY \rangle_{\text{Lie}} = \mathfrak{so}(4), \\
\mathfrak{a}_{15} &= \langle XX, XY, XZ \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{su}(2), \\
\mathfrak{a}_{16} &= \langle XY, YX, YZ, ZY \rangle_{\text{Lie}} = \mathfrak{so}(4), \\
\mathfrak{a}_{17} &= \langle XX, XY, ZX \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{su}(2), \\
\mathfrak{a}_{18} &= \langle XX, XZ, YY, ZY \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{su}(2), \\
\mathfrak{a}_{19} &= \langle XX, XY, ZX, YZ \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{su}(2) \oplus \mathfrak{u}(1), \\
\mathfrak{a}_{20} &= \langle XX, YY, ZZ, ZY \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{su}(2) \oplus \mathfrak{u}(1), \\
\mathfrak{a}_{21} &= \langle XX, YY, XY, ZX \rangle_{\text{Lie}} \cong \mathfrak{sp}(2), \\
\mathfrak{a}_{22} &= \langle XX, XY, XZ, YX, ZX \rangle_{\text{Lie}} = \mathfrak{su}(4), \\
\mathfrak{b}_0 &= \langle XI, IX \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{b}_1 &= \langle XX, XI, IX \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{b}_2 &= \langle XY, XI, IX \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{u}(1), \\
\mathfrak{b}_3 &= \langle XI, YI, IX, IY \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{su}(2),
\end{aligned}
$$

$$
\begin{aligned}
\mathfrak{b}_4 &= \langle XX, XY, XI, IX \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{su}(2) \oplus \mathfrak{u}(1), \\
\mathfrak{c}_0 &= \langle XI \rangle_{\text{Lie}} \cong \mathfrak{u}(1), \\
\mathfrak{c}_1 &= \langle XY, XI \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{c}_2 &= \langle XX, XI \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{c}_3 &= \langle XI, IY \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{c}_4 &= \langle XY, XI, IY \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{c}_5 &= \langle XI, YI, ZI \rangle_{\text{Lie}} \cong \mathfrak{u}(1) \oplus \mathfrak{u}(1) \oplus \mathfrak{u}(1), \\
\mathfrak{c}_6 &= \langle XX, XY, XI \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{u}(1), \\
\mathfrak{c}_7 &= \langle XI, YI, IX \rangle_{\text{Lie}} \cong \mathfrak{su}(2) \oplus \mathfrak{u}(1).
\end{aligned}
$$

Finally, note that

$$
\begin{aligned}
\mathfrak{b}_2 &= \mathfrak{a}_9 \oplus \text{span}\{XI\}, \\
\mathfrak{b}_4 &= \mathfrak{a}_{15} \oplus \text{span}\{XI\},
\end{aligned}
$$

are central extensions of $\mathfrak{a}$-type subalgebras.

G.2.2 List of all 202 subalgebras of $\mathfrak{su}(4)$

Here, we group the 202 subalgebras of $\mathfrak{su}(4)$ into orbits of the symmetry group $S_3 \times \mathbb{Z}_2$. For each orbit, we provide its label and its size.

$\mathfrak{a}_0 : 3,\ \{XX\},\{YY\},\{ZZ\};$

$\mathfrak{a}_1 : 6,\ \{XY\},\{XZ\},\{YX\},\{YZ\},\{ZX\},\{ZY\};$

$\mathfrak{a}_2 : 3,\ \{XY,YX\},\{XZ,ZX\},\{YZ,ZY\};$

$\mathfrak{a}_3 : 6,\ \{XX,YZ\},\{XX,ZY\},\{YY,XZ\},\{YY,ZX\},\{ZZ,XY\},\{ZZ,YX\};$

$\mathfrak{a}_4 : 3,\ \{XX,YY\},\{XX,ZZ\},\{YY,ZZ\};$

$\mathfrak{a}_5 : 6,\ \{XY,YZ\},\{XZ,ZY\},\{YX,XZ\},\{YZ,ZX\},\{ZX,XY\},\{ZY,YX\};$

$\mathfrak{a}_6 : 3,\ \{XY,YX,ZZ\},\{XZ,ZX,YY\},\{YZ,ZY,XX\};$

$\mathfrak{a}_7 : 1,\ \{XX,YY,ZZ\};$

$\mathfrak{a}_8 : 12,\ \{XX,XY,IZ\},\{XX,XZ,IY\},\{XX,YX,ZI\},\{XX,ZX,YI\},$
$\{YY,YX,IZ\},\{YY,YZ,IX\},\{YY,XY,ZI\},\{YY,ZY,XI\},$
$\{ZZ,ZX,IY\},\{ZZ,ZY,IX\},\{ZZ,XZ,YI\},\{ZZ,YZ,XI\};$

$\mathfrak{a}_9 : 6,\ \{XY,XZ,IX\},\{YX,YZ,IY\},\{ZX,ZY,IZ\},$
$\{YX,ZX,XI\},\{XY,ZY,YI\},\{XZ,YZ,ZI\};$

$\mathfrak{a}_{10} : 2,\ \{XY,YZ,ZX\},\{XZ,ZY,YX\};$

$\mathfrak{a}_{11} : 12,\ \{XY,YX,XZ,IX\},\{XZ,ZX,XY,IX\},\{XY,YX,ZX,XI\},\{XZ,ZX,YX,XI\},$
$\{YX,XY,YZ,IY\},\{YZ,ZY,YX,IY\},\{YX,XY,ZY,YI\},\{YZ,ZY,XY,YI\},$
$\{ZX,XZ,ZY,IZ\},\{ZY,YZ,ZX,IZ\},\{ZX,XZ,YZ,ZI\},\{ZY,YZ,XZ,ZI\};$

$\mathfrak{a}_{12} : 12,\ \{XX,XY,YZ,IZ\},\{XX,XZ,ZY,IY\},\{XX,YX,ZY,ZI\},\{XX,ZX,YZ,YI\},$
$\{YY,YX,XZ,IZ\},\{YY,YZ,ZX,IX\},\{YY,XY,ZX,ZI\},\{YY,ZY,XZ,XI\},$
$\{ZZ,ZX,XY,IY\},\{ZZ,ZY,YX,IX\},\{ZZ,XZ,YX,YI\},\{ZZ,YZ,XY,XI\};$

$\mathfrak{a}_{13} : 12,\ \{XX,YY,YZ,IX\},\{XX,YY,XZ,IY\},\{XX,YY,ZY,XI\},\{XX,YY,ZX,YI\},$
$\{XX,ZZ,ZY,IX\},\{XX,ZZ,XY,IZ\},\{XX,ZZ,YZ,XI\},\{XX,ZZ,YX,ZI\},$
$\{YY,ZZ,ZX,IY\},\{YY,ZZ,YX,IZ\},\{YY,ZZ,XZ,YI\},\{YY,ZZ,XY,ZI\};$

$\mathfrak{a}_{14} : 3,\ \{XX,YY,XY,YX,ZI,IZ\},\{XX,ZZ,XZ,ZX,YI,IY\},\{YY,ZZ,YZ,ZY,XI,IX\};$

$\mathfrak{a}_{15} : 6,\ \{XX,XY,XZ,IX,IY,IZ\},\{YY,YX,YZ,IX,IY,IZ\},\{ZZ,ZX,ZY,IX,IY,IZ\},$
$\{XX,YX,ZX,XI,YI,ZI\},\{YY,XY,ZY,XI,YI,ZI\},\{ZZ,XZ,YZ,XI,YI,ZI\};$

$\mathfrak{a}_{16} : 3,\ \{XY,YX,XZ,ZX,XI,IX\},\{YX,XY,YZ,ZY,YI,IY\},\{ZX,XZ,ZY,YZ,ZI,IZ\};$

$\mathfrak{a}_{17} : 6,\ \{XX, XY, ZX, ZY, YI, IZ\}, \{YY, YX, ZY, ZX, XI, IZ\}, \{ZZ, ZX, YZ, YX, XI, IY\},$
$\{XX, YX, XZ, YZ, IY, ZI\}, \{YY, XY, YZ, XZ, IX, ZI\}, \{ZZ, XZ, ZY, XY, IX, YI\};$

$\mathfrak{a}_{18} : 6,\ \{XX, YY, XZ, ZY, XI, IY\}, \{XX, ZZ, XY, YZ, XI, IZ\}, \{YY, ZZ, YX, XZ, YI, IZ\},$
$\{XX, YY, ZX, YZ, IX, YI\}, \{XX, ZZ, YX, ZY, IX, ZI\}, \{YY, ZZ, XY, ZX, IY, ZI\};$

$\mathfrak{a}_{19} : 6,\ \{XX, XY, ZX, ZY, YZ, YI, IZ\}, \{XX, YX, XZ, YZ, ZY, IY, ZI\},$
$\{YY, YX, ZY, ZX, XZ, XI, IZ\}, \{YY, XY, YZ, XZ, ZX, IX, ZI\},$
$\{ZZ, ZX, YZ, YX, XY, XI, IY\}, \{ZZ, XZ, ZY, XY, YX, IX, YI\};$

$\mathfrak{a}_{20} : 3,\ \{XX, YY, ZZ, XY, YX, ZI, IZ\}, \{XX, YY, ZZ, XZ, ZX, YI, IY\},$
$\{XX, YY, ZZ, YZ, ZY, XI, IX\};$

$\mathfrak{a}_{21} : 6,\ \{XX, YY, XY, YX, ZX, ZY, XI, YI, ZI, IZ\}, \{XX, YY, XY, YX, XZ, YZ, IX, IY, ZI, IZ\},$
$\{XX, ZZ, XZ, ZX, YX, YZ, XI, ZI, YI, IY\}, \{XX, ZZ, XZ, ZX, XY, ZY, IX, IZ, YI, IY\},$
$\{YY, ZZ, YZ, ZY, XY, XZ, YI, ZI, XI, IX\}, \{YY, ZZ, YZ, ZY, YX, ZX, IY, IZ, XI, IX\};$

$\mathfrak{a}_{22} : 1,\ \{XX, YY, ZZ, XY, YX, XZ, ZX, YZ, ZY, XI, IX, YI, IY, ZI, IZ\};$

$\mathfrak{b}_0 : 3,\ \{XI, IX\}, \{YI, IY\}, \{ZI, IZ\};$

$\mathfrak{b}_1 : 3,\ \{XX, XI, IX\}, \{YY, YI, IY\}, \{ZZ, ZI, IZ\};$

$\mathfrak{b}_2 : 6,\ \{XY, XZ, XI, IX\}, \{YX, YZ, YI, IY\}, \{ZX, ZY, ZI, IZ\},$
$\{YX, ZX, XI, IX\}, \{XY, ZY, YI, IY\}, \{XZ, YZ, ZI, IZ\};$

$\mathfrak{b}_3 : 1,\ \{XI, YI, ZI, IX, IY, IZ\};$

$\mathfrak{b}_4 : 6,\ \{XX, XY, XZ, XI, IX, IY, IZ\}, \{XX, YX, ZX, IX, XI, YI, ZI\},$
$\{YY, YX, YZ, YI, IY, IX, IZ\}, \{YY, XY, ZY, IY, YI, XI, ZI\},$
$\{ZZ, ZX, ZY, ZI, IZ, IX, IY\}, \{ZZ, XZ, YZ, IZ, ZI, XI, YI\};$

$\mathfrak{c}_0 : 6,\ \{XI\}, \{YI\}, \{ZI\}, \{IX\}, \{IY\}, \{IZ\};$

$\mathfrak{c}_1 : 12,\ \{XY, XI\}, \{XZ, XI\}, \{YX, YI\}, \{YZ, YI\}, \{ZX, ZI\}, \{ZY, ZI\},$
$\{YX, IX\}, \{ZX, IX\}, \{XY, IY\}, \{ZY, IY\}, \{XZ, IZ\}, \{YZ, IZ\};$

$\mathfrak{c}_2 : 6,\ \{XX, XI\}, \{XX, IX\}, \{YY, YI\}, \{YY, IY\}, \{ZZ, ZI\}, \{ZZ, IZ\};$

$\mathfrak{c}_3 : 6,\ \{XI, IY\}, \{XI, IZ\}, \{YI, IX\}, \{YI, IZ\}, \{ZI, IX\}, \{ZI, IY\};$

$\mathfrak{c}_4 : 6,\ \{XY, XI, IY\}, \{YZ, YI, IZ\}, \{ZX, ZI, IX\},$
$\{YX, YI, IX\}, \{ZY, ZI, IY\}, \{XZ, XI, IZ\};$

$\mathfrak{c}_5 : 2,\ \{XI, YI, ZI\}, \{IX, IY, IZ\};$

$\mathfrak{c}_6 : 12,\ \{XX, XY, XI, IZ\}, \{XX, XZ, XI, IY\}, \{XX, YX, IX, ZI\}, \{XX, ZX, IX, YI\},$
$\{YY, YX, YI, IZ\}, \{YY, YZ, YI, IX\}, \{YY, XY, IY, ZI\}, \{YY, ZY, IY, XI\},$

$$\{ZZ, ZX, ZI, IY\}, \{ZZ, ZY, ZI, IX\}, \{ZZ, XZ, IZ, YI\}, \{ZZ, YZ, IZ, XI\};$$
$\mathfrak{c}_7 : 6,\ \{XI, IX, YI, ZI\}, \{YI, IY, XI, ZI\}, \{ZI, IZ, XI, YI\},$
$$\{XI, IX, IY, IZ\}, \{YI, IY, IX, IZ\}, \{ZI, IZ, IX, IY\}.$$

Adding up the orders of the orbits, we obtain a total of 127 Lie algebras of type $\mathfrak{a}$, 19 of type $\mathfrak{b}$, and 56 of type $\mathfrak{c}$.

G.2.3 Identifying the subalgebras with known spin systems

In Table G.2 below, we identify some of our Lie algebras with the DLAs of known spin models. The listed generating sets are from Appendix G.2.1. However, these generating sets are not unique, and they can be replaced with alternative generators that are used to generate the terms of the Hamiltonian; we call those conventional generators.

Label	Generating set	Conventional generators	Example Model
$\mathfrak{a}_0$	XX	ZZ	Ising model [371]
$\mathfrak{a}_1$	XY	XY	Kitaev chain
$\mathfrak{a}_2$	XY, YX		Massless free fermion in a magnetic field
$\mathfrak{a}_3$	XX, YZ	ZZ, XY	Kitaev chain with nearest neighbor Coulomb interaction
$\mathfrak{a}_4$	XX, YY		XY-model [53] / Massless free fermion
$\mathfrak{a}_5$	XY, YZ		
$\mathfrak{a}_6$	XX, YZ, ZY	ZZ, XY, YX	Massless free fermion in a magnetic field with nearest neighbor Coulomb interaction
$\mathfrak{a}_7$	XX, YY, ZZ		Heisenberg model / XXZ Chain [53]
$\mathfrak{a}_8$	XX, XZ	ZZ, IX	Transverse-field Ising model [53]
$\mathfrak{a}_9$	XY, XZ	XY, IX	Kitaev chain in an X field
$\mathfrak{a}_{10}$	XY, YZ, ZX		Heisenberg model
$\mathfrak{a}_{11}$	XY, YX, YZ	XX, YY, IY	XY model in a Y field
$\mathfrak{a}_{12}$	XX, XY, YZ		
$\mathfrak{a}_{13}$	XX, YY, YZ	XX, YY, IX	XY-model in a longitudinal field [53]
$\mathfrak{a}_{14}$	XX, YY, XY		Transverse-field XY / Ising model [53]
$\mathfrak{a}_{15}$	XX, XY, XZ	$ZZ, IX, IY, (IZ)$	Ising model in an arbitrary magnetic field
$\mathfrak{a}_{16}$	XY, YX, YZ, ZY	XY, YX, IY, YI	Kitaev chain in a Y field
$\mathfrak{a}_{17}$	XX, XY, ZX	$ZZ, IX, IY, (IZ)$	Ising model in an arbitrary magnetic field
$\mathfrak{a}_{18}$	XX, XZ, YY, ZY	$XX, YY, IY, XI, (ZI)$	XY model in an arbitrary field
$\mathfrak{a}_{19}$	XX, XY, ZX, YZ		
$\mathfrak{a}_{20}$	XX, YY, ZZ, ZY	XX, YY, ZZ, IX, XI	XXZ chain in an X field [53]
$\mathfrak{a}_{21}$	XX, YY, XY, ZX	$XX, YY, IZ, YI, (IX)$	XY model in an arbitrary field
$\mathfrak{a}_{22}$	XX, XY, XZ, YX, ZX	ZZ, XI, IY, IZ, YI	Ising model in an arbitrary field
$\mathfrak{b}_0$	XI, IX	ZI, IZ	Uncoupled spins
$\mathfrak{b}_1$	XX, XI, IX	ZZ, ZI, IZ	Ising model [371]
$\mathfrak{b}_2$	XY, XI, IX		Kitaev chain in an X field
$\mathfrak{b}_3$	XI, YI, IX, IY		Uncoupled spins
$\mathfrak{b}_4$	$XX, XY, XZ, XI, IX, IY, IZ$	ZZ, IX, IY, IZ, ZI	Ising model in an arbitrary field

Table G.2: **Conventional spin models corresponding to the dynamical Lie algebras discussed in the main text.** Terms in parentheses do not appear explicitly in the set of generators, but are generated from them.

G.2.4 Extending subalgebras of $\mathfrak{su}(4)$ to $\mathfrak{su}(2^n)$

Starting from a subalgebra $\mathfrak{a} \subseteq \mathfrak{su}(4)$, let $\mathfrak{a}(n)$ be the subalgebra of $\mathfrak{su}(2^n)$ generated by the set

$$\bigcup_{0 \le k \le n-2} I^{\otimes k} \otimes \mathfrak{a} \otimes I^{\otimes(n-2-k)}. \tag{G.2}$$

In particular, $\mathfrak{a}(2) = \mathfrak{a}$. By construction, we have two Lie algebra embeddings $\mathfrak{a}(n) \to \mathfrak{a}(n+1)$, given by appending I in the last or first qubit, and $\mathfrak{a}(n+1)$ is generated as a Lie algebra by the union of the two images:

$$\mathfrak{a}(n+1) = \langle (\mathfrak{a}(n) \otimes I) \cup (I \otimes \mathfrak{a}(n)) \rangle_{\text{Lie}}. \tag{G.3}$$

This allows us to determine the sequence of Lie algebras $\mathfrak{a}(2), \mathfrak{a}(3), \ldots$ inductively. In particular, if $\mathfrak{a}(n) = \mathfrak{su}(2^n)$ for some $n = n_0$, then this is true for all $n \ge n_0$. For instance, since

$$\mathfrak{a}_{18}(3) = \mathfrak{a}_{19}(3) = \mathfrak{a}_{21}(3) = \mathfrak{a}_{22}(3) = \mathfrak{su}(8), \quad \mathfrak{a}_{12}(4) = \mathfrak{a}_{17}(4) = \mathfrak{su}(16), \tag{G.4}$$

we obtain that

$$\mathfrak{a}_k(n) = \mathfrak{su}(2^n), \qquad k = 12, 17, 18, 19, 21, 22, \quad n \ge 4. \tag{G.5}$$

We also consider *periodic boundary conditions*. For each $n \ge 2$, define the subalgebra of $\mathfrak{su}(2^n)$:

$$\mathfrak{a}^\circ(n) = \langle \{ A_i B_{i+1}, B_1 A_n \,|\, AB \in \mathfrak{a},\ 1 \le i \le n-1 \} \rangle_{\text{Lie}}. \tag{G.6}$$

Since

$$\mathfrak{a}(n) = \langle \{ A_i B_{i+1} \,|\, AB \in \mathfrak{a},\ 1 \le i \le n-1 \} \rangle_{\text{Lie}}, \tag{G.7}$$

it is obvious that $\mathfrak{a}(n) \subseteq \mathfrak{a}^\circ(n)$. Introduce the cyclic shift operator $\tau_n \colon \mathfrak{su}(2^n) \to \mathfrak{su}(2^n)$, which acts on Pauli strings as

$$\tau_n(P^1 \otimes P^2 \otimes \cdots \otimes P^n) = P^2 \otimes \cdots \otimes P^n \otimes P^1, \qquad P^j \in \{I, X, Y, Z\} \tag{G.8}$$

(where the superscipts are indices not powers). Then τ_n is a Lie algebra automorphism, and $\tau_n \mathfrak{a}^\circ(n) = \mathfrak{a}^\circ(n)$. In particular, this implies that $\tau_n \mathfrak{a}(n) \subseteq \mathfrak{a}^\circ(n)$. Note that, by definition, $\mathfrak{a}^\circ(n)$ is generated as a Lie algebra by the union of $\mathfrak{a}(n)$ and $\tau_n \mathfrak{a}(n)$:

$$\mathfrak{a}^\circ(n) = \langle \mathfrak{a}(n) \cup \tau_n \mathfrak{a}(n) \rangle_{\text{Lie}}. \tag{G.9}$$

Another case we will investigate is when our subalgebras of $\mathfrak{su}(2^n)$ are *permutation invariant*, i.e., invariant under the action of the symmetric group S_n that permutes the n qubits. We define

$$\mathfrak{a}^\pi(n) = \langle \{A_i B_j \,|\, AB \in \mathfrak{a},\ 1 \leq i \neq j \leq n\} \rangle_{\text{Lie}}. \tag{G.10}$$

Note that, in particular, $\mathfrak{a}^\circ(n) \subseteq \mathfrak{a}^\pi(n)$. Moreover, without loss of generality, we can assume that the generating subalgebra $\mathfrak{a} \subseteq \mathfrak{su}(4)$ is itself invariant under S_2, i.e., under the flip of the two qubits. In other words, we have:

$$\begin{aligned}
\mathfrak{a}_1^\pi(n) &= \mathfrak{a}_2^\pi(n), \\
\mathfrak{a}_3^\pi(n) &= \mathfrak{a}_6^\pi(n), \\
\mathfrak{a}_5^\pi(n) &= \mathfrak{a}_{11}^\pi(n) = \mathfrak{a}_{16}^\pi(n), \\
\mathfrak{a}_8^\pi(n) &\cong \mathfrak{a}_{14}^\pi(n), \\
\mathfrak{a}_9^\pi(n) &= \mathfrak{b}_2^\pi(n) \cong \mathfrak{a}_{16}^\pi(n), \\
\mathfrak{a}_{13}^\pi(n) &= \mathfrak{a}_{20}^\pi(n), \\
\mathfrak{a}_k^\pi(n) &= \mathfrak{b}_4^\pi(n) = \mathfrak{su}(2^n), \qquad k = 10, 12, 15, 17, 18, 19, 21, 22.
\end{aligned}$$

Thus, we only need to determine $\mathfrak{a}_k^\pi(n)$ for $k = 0, 2, 4, 6, 7, 14, 16, 20$ and $\mathfrak{b}_l^\pi(n)$ for $l = 0, 1, 3$.

G.2.5 Subalgebras of $\mathfrak{su}(8)$

For completeness and for later use, we list a linear basis (over $\mathbb{R}$, after multiplication by i) for each of the following subalgebras of $\mathfrak{su}(8)$.

Open case:

$\mathfrak{a}_0(3) : \{IXX, XXI\}$,
$\mathfrak{a}_1(3) : \{IXY, XYI, XZY\}$,
$\mathfrak{a}_2(3) : \{IXY, IYX, XYI, XZY, YXI, YZX\}$,
$\mathfrak{a}_3(3) : \{IXX, IYZ, XXI, XZZ, YIY, YXZ, YYX, YZI, ZIZ, ZXY\}$,
$\mathfrak{a}_4(3) : \{IXX, IYY, XXI, XZY, YYI, YZX\}$,
$\mathfrak{a}_5(3) : \{IXY, IYZ, XYI, XZY, YIX, YXZ, YYY, YZI, ZIY, ZYX\}$,
$\mathfrak{a}_6(3) : \{IXY, IYX, IZZ, XIX, XXZ, XYI, XZY, YIY, YXI, YYZ, YZX, ZIZ, ZXX, ZYY, ZZI\}$,
$\mathfrak{a}_7(3) : \{IXX, IYY, IZZ, XIX, XXI, XYZ, XZY, YIY, YXZ, YYI, YZX, ZIZ, ZXY, ZYX, ZZI\}$,
$\mathfrak{a}_8(3) : \{IIY, IXX, IXZ, IYI, IZX, IZZ, XXI, XYX, XYZ, XZI\}$,
$\mathfrak{a}_9(3) : \{IIX, IXI, IXY, IXZ, XYI, XYY, XYZ, XZI, XZY, XZZ\}$,

$\mathfrak{a}_{10}(3) : \{IXY, IYZ, IZX, XIZ, XXX, XYI, XZY, YIX, YXZ, YYY, YZI, ZIY, ZXI, ZYX, ZZZ\},$
$\mathfrak{a}_{11}(3) : \{IIX, IXI, IXY, IXZ, IYX, IZX, XIY, XIZ, XXX, XYI, XYY, XYZ, XZI, XZY,$
$XZZ, YXI, YYX, YZX, ZIX, ZXY, ZXZ\},$
$\mathfrak{a}_{12}(3) : \{IIX, IIY, IIZ, IXX, IXY, IXZ, IYX, IYY, IYZ, IZI, XII, XXI, XYI, XZX, XZY,$
$XZZ, YIX, YIY, YIZ, YXX, YXY, YXZ, YYX, YYY, YYZ, YZI, ZIX, ZIY, ZIZ, ZXX,$
$ZXY, ZXZ, ZYX, ZYY, ZYZ, ZZI\},$
$\mathfrak{a}_{13}(3) : \{IIX, IXI, IXX, IYY, IYZ, IZY, IZZ, XII, XIX, XXI, XYY, XYZ, XZY, XZZ, YIY,$
$YIZ, YXY, YXZ, YYI, YYX, YZI, YZX, ZIY, ZIZ, ZXY, ZXZ, ZYI, ZYX, ZZI, ZZX\},$
$\mathfrak{a}_{14}(3) : \{IIZ, IXX, IXY, IYX, IYY, IZI, XXI, XYI, XZX, XZY, YXI, YYI, YZX, YZY, ZII\},$
$\mathfrak{a}_{15}(3) : \{IIX, IIY, IIZ, IXI, IXX, IXY, IXZ, IYI, IYX, IYY, IYZ, IZI, IZX, IZY, IZZ, XIX,$
$XIY, XIZ, XXI, XXX, XXY, XXZ, XYI, XYX, XYY, XYZ, XZI, XZX, XZY, XZZ\},$
$\mathfrak{a}_{16}(3) : \{IIX, IXI, IXY, IXZ, IYX, IZX, XII, XIY, XIZ, XXX, XYI, XYY, XYZ, XZI,$
$XZY, XZZ, YIX, YXI, YXY, YXZ, YYX, YZX, ZIX, ZXI, ZXY, ZXZ, ZYX, ZZX\},$
$\mathfrak{a}_{17}(3) : \{IIZ, IXI, IXX, IXY, IYI, IYX, IYY, IZI, IZX, IZY, XIZ, XXI, XXX, XXY,$
$XYI, XYX, XYY, XZI, XZX, XZY, YII, YIX, YIY, YXZ, YYZ, YZZ, ZIZ, ZXI,$
$ZXX, ZXY, ZYI, ZYX, ZYY, ZZI, ZZX, ZZY\},$
$\mathfrak{a}_{20}(3) : \{IIZ, IXX, IXY, IYX, IYY, IZI, IZZ, XIX, XIY, XXI, XXZ, XYI, XYZ, XZX, XZY,$
$YIX, YIY, YXI, YXZ, YYI, YYZ, YZX, YZY, ZII, ZIZ, ZXX, ZXY, ZYX, ZYY, ZZI\}.$

Periodic case:

$\mathfrak{a}_0^\circ(3) : \{IXX, XIX, XXI\},$
$\mathfrak{a}_1^\circ(3) : \{IXY, XYI, XZY, YIX, YXZ, ZYX\},$
$\mathfrak{a}_2^\circ(3) : \{IXY, IYX, XIY, XYI, XYZ, XZY, YIX, YXI, YXZ, YZX, ZXY, ZYX\},$
$\mathfrak{a}_3^\circ(3) : \{IIX, IXI, IXX, IYY, IYZ, IZY, IZZ, XII, XIX, XXI, XYY, XYZ, XZY, XZZ, YIY,$
$YIZ, YXY, YXZ, YYI, YYX, YZI, YZX, ZIY, ZIZ, ZXY, ZXZ, ZYI, ZYX, ZZI, ZZX\},$
$\mathfrak{a}_4^\circ(3) : \{IXX, IYY, IZZ, XIX, XXI, XYZ, XZY, YIY, YXZ, YYI, YZX, ZIZ, ZXY, ZYX, ZZI\},$
$\mathfrak{a}_6^\circ(3) : \{IIZ, IXX, IXY, IYX, IYY, IZI, IZZ, XIX, XIY, XXI, XXZ, XYI, XYZ, XZX, XZY,$
$YIX, YIY, YXI, YXZ, YYI, YYZ, YZX, YZY, ZII, ZIZ, ZXX, ZXY, ZYX, ZYY, ZZI\},$
$\mathfrak{a}_8^\circ(3) : \{IIY, IXX, IXZ, IYI, IYY, IZX, IZZ, XIX, XIZ, XXI, XXY, XYX, XYZ, XZI, XZY,$
$YII, YIY, YXX, YXZ, YYI, YZX, YZZ, ZIX, ZIZ, ZXI, ZXY, ZYX, ZYZ, ZZI, ZZY\},$
$\mathfrak{a}_9^\circ(3) : \{IIX, IXI, IXY, IXZ, XII, XYI, XYY, XYZ, XZI, XZY, XZZ, YIX, YXY, YXZ, YYX,$

$YZX, ZIX, ZXY, ZXZ, ZYX, ZZX\}$,

$\mathfrak{a}_{11}^{\circ}(3)$: $\{IIX, IXI, IXY, IXZ, IYX, IZX, XII, XIY, XIZ, XXX, XYI, XYY, XYZ, XZI, XZY,$
$XZZ, YIX, YXI, YXY, YXZ, YYX, YZX, ZIX, ZXI, ZXY, ZXZ, ZYX, ZZX\}$,

$\mathfrak{a}_{14}^{\circ}(3)$: $\{IIZ, IXX, IXY, IYX, IYY, IZI, IZZ, XIX, XIY, XXI, XXZ, XYI, XYZ, XZX, XZY,$
$YIX, YIY, YXI, YXZ, YYI, YYZ, YZX, YZY, ZII, ZIZ, ZXX, ZXY, ZYX, ZYY, ZZI\}$.

Moreover,

$$\mathfrak{a}_k^{\circ}(3) = \mathfrak{a}_k(3), \qquad k = 5, 7, 10, 13, 16, 20,$$

and

$$\mathfrak{a}_{12}^{\circ}(3) = \mathfrak{a}_{15}^{\circ}(3) = \mathfrak{a}_{17}^{\circ}(3) = \mathfrak{su}(8).$$

G.2.6 Subalgebras of $\mathfrak{su}(2^n)$

For convenience, here we repeat the statement of Theorem 4.5.1, with the additional information of the dimensions of the Lie algebras (cf. Equation (4.10)). The proof of the theorem is given in Appendix G.3 below.

$$\begin{aligned}
\mathfrak{a}_0(n) &= \operatorname{span}\{X_j X_{j+1}\}_{1\le j\le n-1} \cong \mathfrak{u}(1)^{\oplus(n-1)}, \quad \dim = n-1,\\
\mathfrak{a}_1(n) &= \operatorname{span}\{X_i Z_{i+1}\cdots Z_{j-1} Y_j\}_{1\le i<j\le n} \cong \mathfrak{so}(n), \quad \dim = \frac{n(n-1)}{2},\\
\mathfrak{a}_2(n) &= \operatorname{span}\{X_i Z_{i+1}\cdots Z_{j-1} Y_j\}_{1\le i<j\le n} \oplus \operatorname{span}\{Y_i Z_{i+1}\cdots Z_{j-1} X_j\}_{1\le i<j\le n}\\
&\cong \mathfrak{so}(n)\oplus\mathfrak{so}(n), \quad \dim = n(n-1),\\
\mathfrak{a}_3(n) &\cong \begin{cases}
\mathfrak{so}(2^{n-2})^{\oplus 4}, & \dim = 2^{n-1}(2^{n-2}-1), \quad n\equiv 0 \mod 8,\\
\mathfrak{so}(2^{n-1}), & \dim = 2^{n-2}(2^{n-1}-1), \quad n\equiv \pm 1 \mod 8,\\
\mathfrak{su}(2^{n-2})^{\oplus 2}, & \dim = 2^{2n-3}-2, \quad n\equiv \pm 2 \mod 8,\\
\mathfrak{sp}(2^{n-2}), & \dim = 2^{n-2}(2^{n-1}+1), \quad n\equiv \pm 3 \mod 8,\\
\mathfrak{sp}(2^{n-3})^{\oplus 4}, & \dim = 2^{n-1}(2^{n-2}+1), \quad n\equiv 4 \mod 8,
\end{cases}\\
\mathfrak{a}_4(n) &\cong \mathfrak{a}_2(n),\\
\mathfrak{a}_5(n) &\cong \begin{cases}
\mathfrak{so}(2^{n-2})^{\oplus 4}, & \dim = 2^{n-1}(2^{n-2}-1), \quad n\equiv 0 \mod 6,\\
\mathfrak{so}(2^{n-1}), & \dim = 2^{n-2}(2^{n-1}-1), \quad n\equiv \pm 1 \mod 6,\\
\mathfrak{su}(2^{n-2})^{\oplus 2}, & \dim = 2^{2n-3}-2, \quad n\equiv \pm 2 \mod 6,\\
\mathfrak{sp}(2^{n-2}), & \dim = 2^{n-2}(2^{n-1}+1), \quad n\equiv 3 \mod 6,
\end{cases}
\end{aligned}$$

$$\mathfrak{a}_6(n) \cong \mathfrak{a}_7(n) \cong \mathfrak{a}_{10}(n) \cong \begin{cases} \mathfrak{su}(2^{n-1}), & \dim = 2^{2n-2}-1, \quad n \text{ odd}, \\ \mathfrak{su}(2^{n-2})^{\oplus 4}, & \dim = 2^{2n-2}-4, \quad n \geq 4 \text{ even}, \end{cases}$$

$$\mathfrak{a}_8(n) \cong \mathfrak{so}(2n-1), \quad \dim = (n-1)(2n-1),$$

$$\mathfrak{a}_9(n) \cong \mathfrak{sp}(2^{n-2}), \quad \dim = 2^{n-2}(2^{n-1}+1),$$

$$\mathfrak{a}_{11}(n) = \mathfrak{a}_{16}(n) = \mathfrak{so}(2^n), \quad \dim = 2^{n-1}(2^n-1), \quad n \geq 4,$$

$$\mathfrak{a}_k(n) = \mathfrak{su}(2^n), \quad \dim = 2^{2n}-1, \quad k = 12, 17, 18, 19, 21, 22, \ \ n \geq 4,$$

$$\mathfrak{a}_{13}(n) = \mathfrak{a}_{20}(n) \cong \mathfrak{a}_{15}(n) \cong \mathfrak{su}(2^{n-1}) \oplus \mathfrak{su}(2^{n-1}), \quad \dim = 2^{2n-1}-2,$$

$$\mathfrak{a}_{14}(n) \cong \mathfrak{so}(2n), \quad \dim = n(2n-1),$$

$$\mathfrak{b}_0(n) = \operatorname{span}\{X_i\}_{1 \leq i \leq n} \cong \mathfrak{u}(1)^{\oplus n}, \quad \dim = n,$$

$$\mathfrak{b}_1(n) = \operatorname{span}\{X_i, X_j X_{j+1}\}_{1 \leq i \leq n,\ 1 \leq j \leq n-1} \cong \mathfrak{u}(1)^{\oplus (2n-1)}, \quad \dim = 2n-1,$$

$$\mathfrak{b}_2(n) = \mathfrak{a}_9(n) \oplus \operatorname{span}\{X_1\} \cong \mathfrak{sp}(2^{n-2}) \oplus \mathfrak{u}(1), \quad \dim = 2^{n-2}(2^{n-1}+1)+1,$$

$$\mathfrak{b}_3(n) = \operatorname{span}\{X_i, Y_i, Z_i\}_{1 \leq i \leq n} \cong \mathfrak{su}(2)^{\oplus n}, \quad \dim = 3n,$$

$$\mathfrak{b}_4(n) = \mathfrak{a}_{15}(n) \oplus \operatorname{span}\{X_1\} \cong \mathfrak{su}(2^{n-1}) \oplus \mathfrak{su}(2^{n-1}) \oplus \mathfrak{u}(1), \quad \dim = 2^{2n-1}-1.$$

G.3 Proofs

This section contains detailed proofs of Theorems 4.5.1, 4.5.2, and 4.5.3. The proof of Theorem 4.5.1 occupies Appendix G.3.2–G.3.7; its plan is outlined in Appendix G.3.1 below. The proofs of Theorems 4.5.2 and 4.5.3 utilize the results of Theorem 4.5.1, and are given in Appendix G.3.8 and G.3.9, respectively. For an index of each proof for each algebra, see Table G.3.

Label	Generators	Scaling	Isomorphism	Reason
$\mathfrak{a}_0$	XX	$O(n)$	$\mathfrak{u}(1)^{\oplus(n-1)}$	Appendix G.3.1
$\mathfrak{a}_1$	XY	$O(n^2)$	$\mathfrak{so}(n)$	Appendix G.3.3, frustration graph
$\mathfrak{a}_2$	XY, YX	$O(n^2)$	$\mathfrak{so}(n) \oplus \mathfrak{so}(n)$	Appendix G.3.3, frustration graph
$\mathfrak{a}_3$	XX, YZ	$O(n)$	n-dependent	Lemmas G.3.29, G.3.31
$\mathfrak{a}_4$	XX, YY	$O(n^2)$	$\mathfrak{a}_2$	Appendix G.3.2, inclusion
$\mathfrak{a}_5$	XY, YZ	$O(4^n)$	n-dependent	Lemmas G.3.30, G.3.32
$\mathfrak{a}_6$	XX, YZ, ZY	$O(4^n)$	$\mathfrak{a}_7$	Appendix G.3.2, inclusion
$\mathfrak{a}_7$	XX, YY, ZZ	$O(4^n)$	$\mathfrak{a}_6$	Lemma G.3.28
$\mathfrak{a}_8$	XX, XZ	$O(n^2)$	$\mathfrak{so}(2n-1)$	Appendix G.3.3, frustration graph
$\mathfrak{a}_9$	XY, XZ	$O(4^n)$	$\mathfrak{sp}(2^{n-2})$	Lemma G.3.27, fixed point under involution
$\mathfrak{a}_{10}$	XY, YZ, ZX	$O(4^n)$	$\mathfrak{a}_6$	Appendix G.3.2, inclusion
$\mathfrak{a}_{11}$	XY, YX, YZ	$O(4^n)$	$\mathfrak{a}_{16}$	Appendix G.3.2, inclusion
$\mathfrak{a}_{12}$	XX, XY, YZ	$O(4^n)$	$\mathfrak{su}(2^n)$	Appendix G.2.4, explicit for $n = 4$
$\mathfrak{a}_{13}$	XX, YY, YZ	$O(4^n)$	$\mathfrak{a}_{20}$	Lemma G.3.26, fixed point under involution
$\mathfrak{a}_{14}$	XX, YY, XY	$O(n^2)$	$\mathfrak{so}(2n)$	Appendix G.3.3, frustration graph
$\mathfrak{a}_{15}$	XX, XY, XZ	$O(4^n)$	$\mathfrak{a}_{13}$	Lemma G.3.26, fixed point under involution
$\mathfrak{a}_{16}$	XY, YX, YZ, ZY	$O(4^n)$	$\mathfrak{so}(2^n)$	Lemma G.3.24, fixed point under involution
$\mathfrak{a}_{17}$	XX, XY, ZX	$O(4^n)$	$\mathfrak{su}(2^n)$	Appendix G.2.4, explicit for $n = 4$
$\mathfrak{a}_{18}$	XX, XZ, YY, ZY	$O(4^n)$	$\mathfrak{su}(2^n)$	Appendix G.2.4, explicit for $n = 3$
$\mathfrak{a}_{19}$	XX, XY, ZX, YZ	$O(4^n)$	$\mathfrak{su}(2^n)$	Appendix G.2.4, explicit for $n = 3$
$\mathfrak{a}_{20}$	XX, YY, ZZ, ZY	$O(4^n)$	$\mathfrak{a}_{13}$	Appendix G.3.2, inclusion
$\mathfrak{a}_{21}$	XX, YY, XY, ZX	$O(4^n)$	$\mathfrak{su}(2^n)$	Appendix G.2.4, explicit for $n = 3$
$\mathfrak{a}_{22}$	XX, XY, XZ, YX	$O(4^n)$	$\mathfrak{su}(2^n)$	Appendix G.2.4, explicit for $n = 3$
$\mathfrak{b}_0$	XI, IX	$O(n)$	$\mathfrak{u}(1)^{\oplus n}$	Appendix G.3.1
$\mathfrak{b}_1$	XX, XI, IX	$O(n)$	$\mathfrak{u}(1)^{\oplus(2n-1)}$	Appendix G.3.1
$\mathfrak{b}_2$	XY, XI, IX	$O(4^n)$	$\mathfrak{sp}(2^{n-2}) \oplus \mathfrak{u}(1)$	Appendix G.3.1, central extension
$\mathfrak{b}_3$	XI, YI, IX, IY	$O(n)$	$\mathfrak{su}(2)^{\oplus n}$	Appendix G.3.1
$\mathfrak{b}_4$	$XX, XY, XZ, XI, IX, IY, IZ$	$O(4^n)$	$\mathfrak{su}(2^{n-1}) \oplus \mathfrak{su}(2^{n-1}) \oplus \mathfrak{u}(1)$	Appendix G.3.1, central extension

Table G.3: Proofs and where to find them.

G.3.1 Plan of the proof of Theorem 4.5.1

Our starting point is the list of subalgebras of $\mathfrak{su}(4)$ from Appendix G.2.1:

$$\mathfrak{a}_k, \mathfrak{b}_l \subseteq \mathfrak{su}(4), \qquad 0 \le k \le 22, \;\; 0 \le l \le 4, \tag{G.11}$$

and the goal is to determine (up to isomorphism) their extensions as subalgebras of $\mathfrak{su}(2^n)$:

$$\mathfrak{a}_k(n), \mathfrak{b}_l(n) \subseteq \mathfrak{su}(2^n), \qquad 3 \leq n, \;\; 0 \leq k \leq 22, \;\; 0 \leq l \leq 4, \tag{G.12}$$

which are defined in Appendix G.2.4. The answer is presented in Theorem 4.5.1, and in more detail, in Appendix G.2.6 above. Here we outline the proof, which consists of multiple parts. We divide the set of Lie algebras Equation (G.12) into three classes: linear, quadratic, and exponential, according to the anticipated growth of their dimension.

Linear case: $\mathfrak{a}_0(n)$, $\mathfrak{b}_l(n)$ $(l = 0, 1, 3)$. The linear case is obvious. Indeed, the Lie algebras $\mathfrak{a}_0(n)$, $\mathfrak{b}_0(n)$, and $\mathfrak{b}_1(n)$ are Abelian (i.e., have identically zero Lie brackets), because all of their generators commute with each other. The Lie algebra $\mathfrak{b}_3(n) \cong \mathfrak{su}(2)^{\oplus n}$ is a direct sum of n commuting copies of $\mathfrak{su}(2)$, since its generators split into n groups commuting with each other (acting independently on each qubit).

Quadratic case: $\mathfrak{a}_k(n)$ $(k = 1, 2, 4, 8, 14)$. These Lie algebras are determined by using the *frustration graphs* of their generators; see Appendix G.3.3 and especially Lemma G.3.6. Note that $\mathfrak{a}_2(n) \cong \mathfrak{a}_4(n)$ by Lemma G.3.3.

Exponential case: $\mathfrak{a}_k(n)$, $\mathfrak{b}_l(n)$ $(k = 3, 5, 6, 7, 9{-}13, 15{-}22, \; l = 2, 4)$. First, recall that in Appendix G.2.4, we have already found that

$$\mathfrak{a}_k(n) = \mathfrak{su}(2^n), \qquad k = 12, 17, 18, 19, 21, 22, \;\; n \geq 4. \tag{G.13}$$

Second, we observe that $\mathfrak{b}_2(n) = \mathfrak{a}_9(n) \oplus \mathrm{span}\{X_1\}$ and $\mathfrak{b}_4(n) = \mathfrak{a}_{15}(n) \oplus \mathrm{span}\{X_1\}$, because their generators consist of a central element X_1 (commuting with all other generators) and the generators of $\mathfrak{a}_9(n)$ or $\mathfrak{a}_{15}(n)$, respectively. Third, in Appendix G.3.2, we find equalities and isomorphisms among some of the Lie algebras $\mathfrak{a}_k(n)$. Namely,

$$\mathfrak{a}_6(n) \cong \mathfrak{a}_7(n) \cong \mathfrak{a}_{10}(n), \qquad \mathfrak{a}_{11}(n) = \mathfrak{a}_{16}(n), \qquad \mathfrak{a}_{13}(n) = \mathfrak{a}_{20}(n), \qquad n \geq 4. \tag{G.14}$$

Thus, we are left to investigate the Lie algebras $\mathfrak{a}_k(n)$ for $k = 3, 5, 7, 9, 13, 15, 16$.

Strategy for $\mathfrak{a}_k(n)$ $(k = 3, 5, 7, 9, 13, 15, 16)$. The strategy in the remaining exponential cases is as follows.

1. For each of our Lie algebras $\mathfrak{s} = \mathfrak{a}_k(n)$, we find its *stabilizer* $\mathrm{St}(\mathfrak{s})$, which is defined as the set of all Pauli strings $\in \mathcal{P}_n$ that commute with every element of $\mathfrak{s}$. This can be done explicitly, because the stabilizer is determined only from the generators of $\mathfrak{s}$; see Proposition G.3.3 in Appendix G.3.4.

2. By definition, $\mathfrak{s}$ commutes with all elements of its stabilizer $\mathrm{St}(\mathfrak{s})$; hence, it is contained in the *centralizer* of $\mathrm{St}(\mathfrak{s})$ in $\mathfrak{su}(2^n)$, which we denote $\mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}$. We can reduce the Lie subalgebra $\mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}$ further by factoring all elements of the center of $\mathrm{St}(\mathfrak{s})$, which will become central in it, because $\mathfrak{s}$ has a trivial center by Lemma G.3.11. This results in a Lie algebra denoted $\mathfrak{g}_k(n)$ when $\mathfrak{s} = \mathfrak{a}_k(n)$; these are listed explicitly in Equation (G.35)–Equation (G.41).

3. By the above construction, we have $\mathfrak{s} \subseteq \mathfrak{g}_k(n)$. However, equality does not hold in all cases. We improve the upper bound for $\mathfrak{s}$ by finding an *involution* θ_k of $\mathfrak{g}_k(n)$ such that all elements of $\mathfrak{s}$ are fixed under θ_k, i.e., $\theta_k(a) = a$ for all $a \in \mathfrak{s}$ (see Appendix G.1 for a refresher on involutions). The last condition can be checked only on the generators of $\mathfrak{s}$, and the details are carried out in Appendix G.3.5 (Theorem G.3.1 and Lemmas G.3.13, G.3.14, G.3.15). Thus, we have the upper bound $\mathfrak{s} \subseteq \mathfrak{g}_k(n)^{\theta_k}$, where the superscript θ_k indicates fixed points under θ_k.

4. Then, in Appendix G.3.6, we establish a lower bound for $\mathfrak{s}$, i.e., we prove that the upper bound is exact: $\mathfrak{a}_k(n) = \mathfrak{g}_k(n)^{\theta_k}$. The main idea is to start with an arbitrary Pauli string $\in i\mathcal{P}_n \cap \mathfrak{g}_k(n)^{\theta_k}$ and use suitable commutators with elements of $\mathfrak{a}_k(n)$ to produce a Pauli string $\in i\mathcal{P}_n \cap \mathfrak{g}_k(n)^{\theta_k}$ with I in one of its positions. Erasing the I gives an element of $\mathfrak{g}_k(n-1)^{\theta_k}$, which by induction is in $\mathfrak{a}_k(n-1)$. The specific details are broken into a sequence of lemmas. The cases $k = 3, 5, 7$, $k = 9$, $k = 13$, $k = 15$, and $k = 16$ are treated in Lemmas G.3.20, G.3.22, G.3.23, G.3.25, and G.3.24, respectively. As a consequence, since $\mathfrak{g}_{16}(n) = \mathfrak{su}(2^n)$ by Equation (G.40), and $\theta_{16}(g) = -g^T$ by Equation (G.46), we obtain that $\mathfrak{a}_{16}(n) = \mathfrak{so}(2^n)$.

5. Finally, in Appendix G.3.7, we identify the Lie algebras $\mathfrak{g}_k(n)^{\theta_k}$ with those from Theorem 4.5.1. The idea is to apply a suitable unitary transformation that brings the stabilizer $\mathrm{St}(\mathfrak{s})$ to a more convenient form (such transformations and their effect on the fixed points of involutions are reviewed in Appendix G.1). For example, $\mathrm{St}(\mathfrak{a}_{13}(n)) = \{I^{\otimes n}, X^{\otimes n}\}$ and we can bring $X^{\otimes n}$ to X_1 with a unitary transformation, after which it is easy to determine the centralizer $\mathfrak{su}(2^n)^{X_1}$. This is carried out in Lemmas G.3.26, G.3.27, and G.3.28 for $k = 13, 15$, $k = 9$, and $k = 7$, respectively. The more complicated cases $k = 3$ and $k = 5$ are further broken down to n odd and n even; see Lemmas G.3.29, G.3.31, G.3.30, G.3.32. Taken all together, this completes the proof of Theorem 4.5.1.

G.3.2 Inclusions and isomorphisms

There are several obvious inclusions among the Lie algebras $\mathfrak{a}_k$ ($k = 0{-}11, 13{-}16, 20$), which extend to the corresponding subalgebras of $\mathfrak{su}(2^n)$ due to the following trivial observation.

Lemma G.3.1. *If $\mathfrak{a} \subset \mathfrak{b} \subseteq \mathfrak{su}(4)$, then $\mathfrak{a}(n) \subseteq \mathfrak{b}(n)$, $\mathfrak{a}^\circ(n) \subseteq \mathfrak{b}^\circ(n)$ and $\mathfrak{a}^\pi(n) \subseteq \mathfrak{b}^\pi(n)$ for all $n \geq 2$. Furthermore, if $\mathfrak{a}(n_0) = \mathfrak{b}(n_0)$ for some n_0, then $\mathfrak{a}(n) = \mathfrak{b}(n)$ for all $n \geq n_0$.*

Proof. The first claim follows from the definitions of the Lie algebras $\mathfrak{a}(n)$, $\mathfrak{a}^\circ(n)$, and $\mathfrak{a}^\pi(n)$; see Equation (G.7), Equation (G.6), Equation (G.10), respectively. The second claim follows from the inductive construction of the Lie algebras $\mathfrak{a}(n)$; see Equation (G.3). □

For the rest of this subsection, we focus on the open case; we will treat the periodic (closed) case and the permutation-invariant case later. By comparing the bases of the subalgebras $\mathfrak{a}_k \subset \mathfrak{su}(4)$ listed in Appendix G.2.1, we note the following inclusions:

$$\begin{aligned}
&\mathfrak{a}_0 \subset \mathfrak{a}_3 \subset \mathfrak{a}_6 \subset \mathfrak{a}_{20}, && \mathfrak{a}_1 \subset \mathfrak{a}_2 \subset \mathfrak{a}_{11} \subset \mathfrak{a}_{16},\\
&\mathfrak{a}_0 \subset \mathfrak{a}_4 \subset \mathfrak{a}_7, && \mathfrak{a}_1 \subset \mathfrak{a}_2 \subset \mathfrak{a}_{14},\\
&\mathfrak{a}_0 \subset \mathfrak{a}_4 \subset \mathfrak{a}_{13} \subset \mathfrak{a}_{20}, && \mathfrak{a}_1 \subset \mathfrak{a}_5 \subset \mathfrak{a}_{10}, \quad \text{and} \quad \mathfrak{a}_1 \subset \mathfrak{a}_5 \subset \mathfrak{a}_{16},\\
&\mathfrak{a}_0 \subset \mathfrak{a}_8 \subset \mathfrak{a}_{15}, && \mathfrak{a}_1 \subset \mathfrak{a}_9 \subset \mathfrak{a}_{15}.
\end{aligned}$$

The above chains of inclusions are maximal, i.e., cannot be extended further. For any pair $\mathfrak{a}_k \subset \mathfrak{a}_l$, we get an inclusion $\mathfrak{a}_k(n) \subseteq \mathfrak{a}_l(n)$ for all $n \geq 3$. However, even though $\mathfrak{a}_5 = \mathfrak{a}_{10} \cap \mathfrak{a}_{16}$, we only have $\mathfrak{a}_5(n) \subseteq \mathfrak{a}_{10}(n) \cap \mathfrak{a}_{16}(n)$ and not necessarily an equality. In fact, one checks that $IZX \in \mathfrak{a}_{10}(3) \cap \mathfrak{a}_{16}(3)$ but $IZX \notin \mathfrak{a}_5(3)$. In the next lemma, we present two consequences of the above inclusions.

Lemma G.3.2. *We have:*

$$\mathfrak{a}_{11}(n) = \mathfrak{a}_{16}(n), \qquad n \geq 4, \tag{G.15}$$

$$\mathfrak{a}_{13}(n) = \mathfrak{a}_{20}(n), \qquad n \geq 3. \tag{G.16}$$

Proof. Since $\mathfrak{a}_{11} \subset \mathfrak{a}_{16}$, we have $\mathfrak{a}_{11}(n) \subseteq \mathfrak{a}_{16}(n)$ for all n. But because $\dim \mathfrak{a}_{11}(4) = \dim \mathfrak{a}_{16}(4) = 120$, we obtain that $\mathfrak{a}_{11}(4) = \mathfrak{a}_{16}(4)$, which implies Equation (G.15). Similarly, from $\mathfrak{a}_{13} \subset \mathfrak{a}_{20}$, we get $\mathfrak{a}_{13}(n) \subseteq \mathfrak{a}_{20}(n)$ for all n. But because $\dim \mathfrak{a}_{13}(3) = \dim \mathfrak{a}_{20}(3) = 30$, we have $\mathfrak{a}_{13}(3) = \mathfrak{a}_{20}(3)$. □

There are other inclusions among the Lie algebras $\mathfrak{a}_k$ after we relabel some of the Paulis. Such relabelings act as automorphisms of the Lie algebra $\mathfrak{su}(2^n)$, i.e., they are invertible linear operator that respects the Lie bracket (see Appendix G.1). We will express them in the form $\varphi(a) = UaU^\dagger$ ($a \in \mathfrak{su}(2^n)$) for some fixed unitary matrix U, as in Lemma G.1.2.

As a first example, consider the linear operator ψ on $\mathbb{C}^{2\times 2}$, defined by $\psi(A) = VAV^\dagger$ where $V = e^{i\frac{\pi}{4}Z}$. Using Equation (A.6), we find:

$$\psi(I) = I, \qquad \psi(X) = iZ \cdot X = -Y, \qquad \psi(Y) = iZ \cdot Y = X, \qquad \psi(Z) = Z.$$

We extend it to an automorphism ψ_n of $\mathfrak{su}(2^n)$ by:

$$\psi_n := \psi^0 \otimes \psi^1 \otimes \cdots \otimes \psi^{n-1},$$

where ψ^j denotes the j-th power of ψ. Note that, up to an overall sign, ψ_n swaps $X \rightleftharpoons Y$ on all even qubits. It can be represented as a unitary transformation:

$$\psi_n(a) = UaU^\dagger, \qquad \text{with} \quad U = V^1 \otimes V^2 \otimes \cdots \otimes V^n = \exp\Big(i\frac{\pi}{4}\sum_{j=1}^{n}(j-1)Z_j\Big). \tag{G.17}$$

Lemma G.3.3. *The map ψ_n, defined by Equation* (G.17)*, restricts to an isomorphism $\mathfrak{a}_2(n) \cong \mathfrak{a}_4(n)$.*

Proof. Since ψ_n is an automorphism of $\mathfrak{su}(2^n)$, it is in particular injective and respects the Lie bracket. The same is true for the restriction of ψ_n to $\mathfrak{a}_2(n)$. In order to prove that ψ_n is an isomorphism from $\mathfrak{a}_2(n)$ to $\mathfrak{a}_4(n)$, it remains to show that it is surjective, i.e., $\psi_n\mathfrak{a}_2(n) = \mathfrak{a}_4(n)$. Note that $\psi_n\mathfrak{a}_2(n)$ is a subalgebra of $\mathfrak{a}_4(n)$. We will show that $\psi_n\mathfrak{a}_2(n)$ contains all generators of $\mathfrak{a}_4(n)$, which would imply that it is equal to it.

Indeed, ψ_n acts as follows on the generators of $\mathfrak{a}_2(n)$:

$$\begin{aligned}\psi_n(X_iY_{i+1}) &= (\psi^{i-1}(X))_i(\psi^i(Y))_{i+1} = (\psi^{i-1}(X))_i(\psi^{i-1}(X))_{i+1},\\ \psi_n(Y_iX_{i+1}) &= (\psi^{i-1}(Y))_i(\psi^i(X))_{i+1} = -(\psi^{i-1}(Y))_i(\psi^{i-1}(Y))_{i+1}.\end{aligned}$$

Hence, up to a sign, ψ_n sends the generators of $\mathfrak{a}_2(n)$ to the generators X_iX_{i+1}, Y_iY_{i+1} of $\mathfrak{a}_4(n)$. Therefore, $\psi_n\mathfrak{a}_2(n) = \mathfrak{a}_4(n)$, which completes the proof of the lemma. □

As another similar example, consider the linear operator φ on $\mathbb{C}^{2\times 2}$, defined by

$$\varphi(A) := e^{i\frac{\pi}{4}X}Ae^{-i\frac{\pi}{4}X} \;\Rightarrow\; \varphi(I) = I, \quad \varphi(X) = X, \quad \varphi(Y) = -Z, \quad \varphi(Z) = Y. \tag{G.18}$$

We extend it to an automorphism of $\mathfrak{su}(2^n)$ by

$$\varphi_n := \varphi^0 \otimes \varphi^1 \otimes \cdots \otimes \varphi^{n-1}, \tag{G.19}$$

which, up to a sign, swaps $Y \rightleftharpoons Z$ on all even qubits. As in Equation (G.17), we have

$$\varphi_n(a) = UaU^\dagger, \qquad \text{with} \quad U = \exp\Big(i\frac{\pi}{4}\sum_{j=1}^{n}(j-1)X_j\Big). \tag{G.20}$$

Lemma G.3.4. *The map φ_n, defined by Equation* (G.18)*, Equation* (G.19)*, restricts to an isomorphism $\mathfrak{a}_6(n) \cong \mathfrak{a}_7(n)$.*

Proof. As in the proof of Lemma G.3.3, we find that φ_n acts on the generators of $\mathfrak{a}_6(n)$ as follows:

$$\begin{aligned}
\varphi_n(X_iX_{i+1}) &= X_iX_{i+1},\\
\varphi_n(Y_iZ_{i+1}) &= (\varphi^{i-1}(Y))_i(\varphi^{i-1}(Y))_{i+1},\\
\varphi_n(Z_iY_{i+1}) &= -(\varphi^{i-1}(Z))_i(\varphi^{i-1}(Z))_{i+1}.
\end{aligned}$$

Up to a sign, the images are exactly the generators X_iX_{i+1}, Y_iY_{i+1}, Z_iZ_{i+1} of $\mathfrak{a}_7(n)$. Hence, $\varphi_n\mathfrak{a}_6(n) = \mathfrak{a}_7(n)$. □

Now consider the composition $\gamma := \varphi\psi$, which acts as a cyclic rotation $X \mapsto Z \mapsto Y \mapsto X$:

$$\gamma(I) = I, \qquad \gamma(X) = Z, \qquad \gamma(Y) = X, \qquad \gamma(Z) = Y. \tag{G.21}$$

We extend it to automorphism of $\mathfrak{su}(2^n)$ as follows:

$$\gamma_n := \gamma^1 \otimes \gamma^2 \otimes \gamma^3 \otimes \cdots \otimes \gamma^n. \tag{G.22}$$

Since $S := (X+Y+Z)/\sqrt{3}$ satisfies $S \cdot S = I$, we can apply Euler's formula Equation (A.3) to show that

$$\gamma(A) = e^{i\frac{\pi}{4}X}e^{i\frac{\pi}{4}Z}Ae^{-i\frac{\pi}{4}Z}e^{-i\frac{\pi}{4}X} = e^{i\frac{\pi}{3}S}Ae^{-i\frac{\pi}{3}S}.$$

Hence, similarly to Equation (G.17), Equation (G.20), we can express γ_n as

$$\gamma_n(a) = UaU^\dagger, \qquad \text{with} \quad U = \exp\Big(i\frac{\pi}{3\sqrt{3}}\sum_{j=1}^{n} j(X_j + Y_j + Z_j)\Big). \tag{G.23}$$

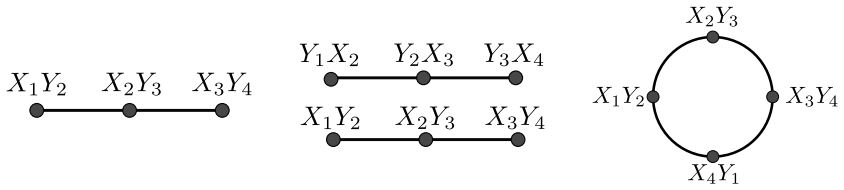
Lemma G.3.5. *The map γ_n, defined by Equation* (G.21)*, Equation* (G.22)*, restricts to an isomorphism $\mathfrak{a}_{10}(n) \cong \mathfrak{a}_7(n)$.*

Proof. We find that γ_n acts on the generators of $\mathfrak{a}_{10}(n)$ as follows:

$$\begin{aligned}
\gamma_n(X_iY_{i+1}) &= (\gamma^i(X))_i(\gamma^{i+1}(Y))_{i+1} = (\gamma^i(X))_i(\gamma^i(X))_{i+1}, \\
\gamma_n(Y_iZ_{i+1}) &= (\gamma^i(Y))_i(\gamma^{i+1}(Z))_{i+1} = (\gamma^i(Y))_i(\gamma^i(Y))_{i+1}, \\
\gamma_n(Z_iX_{i+1}) &= (\gamma^i(Z))_i(\gamma^{i+1}(X))_{i+1} = (\gamma^i(Z))_i(\gamma^i(Z))_{i+1}.
\end{aligned}$$

The images are exactly the generators X_iX_{i+1}, Y_iY_{i+1}, Z_iZ_{i+1} of $\mathfrak{a}_7(n)$; hence, $\gamma_n\mathfrak{a}_{10}(n) = \mathfrak{a}_7(n)$. □

G.3.3 Frustration graphs



(a) $\mathfrak{a}_1(4)$ (b) $\mathfrak{a}_2(4)$ (c) $\mathfrak{a}_2^\circ(4)$

Figure G.1: Frustration graphs for several examples in our classification for $n = 4$. (a) A frustration graph given by a line. (b) A frustration graph consisting of two disjoint lines. (c) A circular frustration graph.

In this subsection, we review the notion of frustration graph, which is a useful visualization tool; see e.g. [372, 373, 342]. We determine the DLA in the cases when the frustration graph is a line or a circle, and apply these results to identify several of our Lie algebras, namely, $\mathfrak{a}_k(n)$ and $\mathfrak{a}_k^\circ(n)$ for $k = 1, 2, 4, 8, 14$.

Definition G.3.1. *Given a set of Pauli strings $\mathcal{A} \subset \mathcal{P}_n$, its* frustration graph *is the graph with a set of vertices $\mathcal{A}$ and edges connecting all pairs of vertices $a, b \in \mathcal{A}$ such that $[a, b] \neq 0$.*

The frustration graph makes it especially easy to determine when two subsets of the generating set $\mathcal{A}$ commute with each other: it means that they are disconnected from each other in the frustration graph. Suppose that $\mathcal{A} = \mathcal{A}_1 \sqcup \mathcal{A}_2$ is a disjoint union of subsets with disconnected frustration graphs. Then

$$\langle \mathcal{A}_1 \sqcup \mathcal{A}_2 \rangle_{\mathrm{Lie}} = \langle \mathcal{A}_1 \rangle_{\mathrm{Lie}} \oplus \langle \mathcal{A}_2 \rangle_{\mathrm{Lie}} \tag{G.24}$$

is a direct sum of commuting subalgebras. This is illustrated in Figure G.1.

In the next proposition, we determine the DLA of a line frustration graph.

Proposition G.3.1. *Suppose that the frustration graph of $\mathcal{A} = \{a_1, \ldots, a_N\} \subset \mathcal{P}_n$ is a line, so that $[a_j, a_k] \neq 0$ for $1 \leq j < k \leq N$ if and only if $k = j + 1$. Then*

$$\langle \mathcal{A} \rangle_{\mathrm{Lie}} \cong \mathfrak{so}(N+1),$$

and a basis for it is given by $\{i^{k-j} L_{j,k}\}_{1 \leq j < k \leq N+1}$, where i is the imaginary unit and

$$L_{j,k} := a_j \cdot a_{j+1} \cdots a_{k-1} \qquad (1 \leq j < k \leq N+1) \tag{G.25}$$

are products over line segments.

Proof. Recall that $\langle \mathcal{A} \rangle_{\mathrm{Lie}}$ is the subalgebra of $\mathfrak{su}(2^n)$ generated by the subset $i\mathcal{A} \subset \mathfrak{su}(2^n)$ (see Definition 4.1.1). First, let us prove that all $i^{k-j} L_{j,k}$ are in $\langle \mathcal{A} \rangle_{\mathrm{Lie}}$. For $k = j + 1$, we have $iL_{j,j+1} := ia_j \in i\mathcal{A} \subseteq \langle \mathcal{A} \rangle_{\mathrm{Lie}}$. Suppose, by induction on $k - j$, that $i^{k-j} L_{j,k} \in \langle \mathcal{A} \rangle_{\mathrm{Lie}}$ for some $1 < j < k \leq N + 1$; then we will show that $i^{k-j+1} L_{j-1,k} \in \langle \mathcal{A} \rangle_{\mathrm{Lie}}$. By definition,

$$L_{j-1,k} = a_{j-1} \cdot a_j \cdot a_{j+1} \cdots a_{k-1} = a_{j-1} \cdot L_{j,k},$$

and by assumption, a_{j-1} anticommutes with a_j and commutes with $a_{j+1}, \ldots, a_{k-1}$. Hence, a_{j-1} anticommutes with $L_{j,k}$, which implies that

$$2i^{k-j+1} L_{j-1,k} = 2i^{k-j+1} a_{j-1} \cdot L_{j,k} = i^{k-j+1} [a_{j-1}, L_{j,k}] = [ia_{j-1}, i^{k-j} L_{j,k}] \in \langle \mathcal{A} \rangle_{\mathrm{Lie}}.$$

This proves the claim that $i^{k-j} L_{j,k} \in \langle \mathcal{A} \rangle_{\mathrm{Lie}}$ for all $1 \leq j < k \leq N + 1$.

Similarly to above, one can check that ($1 \leq j < k < l \leq N + 1$):

$$[i^{k-j} L_{j,k}, i^{l-k} L_{k,l}] = 2i^{l-j} L_{j,l}, \quad [i^{l-k} L_{k,l}, i^{l-j} L_{j,l}] = 2i^{k-j} L_{j,k}, \quad [i^{l-j} L_{j,l}, i^{k-j} L_{j,k}] = 2i^{l-k} L_{k,l}, \tag{G.26}$$

and all other commutators (not following from skewsymmetry) are zero. In particular, the real linear span of all $i^{k-j} L_{j,k}$ is closed under the bracket, i.e., is a subalgebra of $\mathfrak{su}(2^n)$.

Since $\langle \mathcal{A} \rangle_{\text{Lie}}$ is the minimal (under inclusion) subalgebra of $\mathfrak{su}(2^n)$ containing $i\mathcal{A}$, it follows that

$$\langle \mathcal{A} \rangle_{\text{Lie}} = \text{span}\{i^{k-j} L_{j,k}\}_{1 \le j < k \le N+1}.$$

Recall that $\mathfrak{so}(N+1) = \mathfrak{so}(N+1, \mathbb{R})$ is the Lie algebra of all skewsymmetric $(N+1) \times (N+1)$ real matrices; see Equation (4.8). Consider the standard basis $\{E_{j,k}\}_{1 \le j,k \le N+1}$ of $\mathfrak{gl}(N+1, \mathbb{R})$, where $E_{j,k}$ is the matrix with (j,k)-entry $= 1$ and all other entries $= 0$. Then a basis for $\mathfrak{so}(N+1)$ is $\{F_{j,k} := E_{j,k} - E_{k,j}\}_{1 \le j < k \le N+1}$. Using that

$$[E_{j,k}, E_{l,m}] = \delta_{k,l} E_{j,m} - \delta_{j,m} E_{l,k},$$

it is easy to see that

$$[F_{j,k}, F_{k,l}] = F_{j,l}, \quad [F_{k,l}, F_{j,l}] = F_{j,k}, \quad [F_{j,l}, F_{j,k}] = F_{k,l}, \quad \text{for} \quad 1 \le j < k < l \le N+1.$$

Hence, the matrices $2F_{j,k}$ satisfy the same commutation relations as $i^{k-j} L_{j,k}$ given in Equation (G.26). This means that the map $\mathfrak{so}(N+1) \to \langle \mathcal{A} \rangle_{\text{Lie}}$ that sends $2F_{j,k}$ to $i^{k-j} L_{j,k}$ is a Lie algebra homomorphism. Its kernel is an ideal in $\mathfrak{so}(N+1)$, but since $\mathfrak{so}(N+1)$ is simple, it has no non-zero proper ideals. Therefore, this map is an isomorphism. □

Remark G.3.1. *One can see from the above proposition that, for linear frustration graphs, the dimension of the DLA scales quadratically with the number of generators. This was observed for free fermionic models in [374, 228, 312, 313, 312], where the number of generators is proportional to the system size and the circuit gate complexity is quadratic with respect to the system size. These models are fast-forwardable along with the other Hamiltonians given in [372, 373, 342], and the fundamental reason for this is the polynomial scaling of the DLA.*

After a Jordan–Wigner transformation, it can be shown that the algebra of free fermions on n sites can be generated by $\{Z_1, X_1X_2, Z_2, X_2X_3, Z_3, \ldots, X_{n-1}X_n, Z_n\}$, which will be shown to be equivalent to $\mathfrak{a}_{14}(n)$ in Lemma G.3.6. These generators have a linear frustration graph with $2n-1$ vertices; hence, its DLA is $\mathfrak{so}(2n)$.

Next, we consider the case when the frustration graph is a circle.

Proposition G.3.2. *Suppose that the frustration graph of $\mathcal{A} = \{a_1, \ldots, a_N\} \subset \mathcal{P}_n$ is a circle with $N \ge 3$, so that $[a_j, a_k] \ne 0$ for $1 \le j < k \le N$ if and only if $k = j+1$ or $j = 1$, $k = N$. Then*

$$\langle \mathcal{A} \rangle_{\text{Lie}} \cong \mathfrak{so}(N) \oplus \mathfrak{so}(N),$$

and it has a basis $\{i^{k-j} L_{j,k}, i^{N+k-j} C \cdot L_{j,k}\}_{1 \le j < k \le N}$, where i is the imaginary unit, $L_{j,k}$ are defined in Equation (G.25)*, and*

$$C := a_1 \cdot a_2 \cdots a_{N-1} \cdot a_N. \tag{G.27}$$

Proof. First, notice that $[C, a_j] = 0$ for all $1 \leq j \leq N$, because a_j does not commute only with its two neighboring vertices in the circle frustration graph. Moreover, using that $a_j \cdot a_j = I^{\otimes n}$ (cf. Lemma A.1.1), we get $C \cdot C = (-1)^N I^{\otimes n}$. From here, we deduce that

$$[i^N C, i^{k-j} L_{j,k}] = 0, \qquad (i^N C) \cdot (i^N C) = I^{\otimes n}. \tag{G.28}$$

If we remove any vertex from the frustration graph of $\mathcal{A}$, we obtain a line frustration graph. By Proposition G.3.1, we know that

$$i^{k-j} L_{j,k} \in \langle a_1, \ldots, a_{N-1} \rangle_{\mathrm{Lie}} \subseteq \langle \mathcal{A} \rangle_{\mathrm{Lie}}, \qquad 1 \leq j < k \leq N,$$

and these elements form a basis for the subalgebra $\langle a_1, \ldots, a_{N-1} \rangle_{\mathrm{Lie}} \cong \mathfrak{so}(N)$. In particular, we have

$$\{ia_1, \ldots, ia_{N-1}\} \subset \langle a_1, \ldots, a_{N-1} \rangle_{\mathrm{Lie}} = \mathrm{span}_{\mathbb{R}} \{i^{k-j} L_{j,k}\}_{1 \leq j < k \leq N}.$$

Similarly, the set $\mathcal{A} \setminus \{a_{k-1}\} = \{a_k, a_{k+1}, \ldots, a_N, a_1, \ldots, a_{k-2}\}$ has a line frustration graph and its subset $\{a_k, a_{k+1}, \ldots, a_N, a_1, \ldots, a_{j-1}\}$ is a line segment for $1 \leq j < k \leq N$. Hence, again by Proposition G.3.1,

$$i^{N+k-j} C \cdot L_{j,k} = \pm i^{N-k+j} a_k \cdot a_{k+1} \cdots a_N \cdot a_1 \cdots a_{j-1} \in \langle \mathcal{A} \setminus \{a_{k-1}\} \rangle_{\mathrm{Lie}} \subseteq \langle \mathcal{A} \rangle_{\mathrm{Lie}}, \qquad 1 \leq j < k \leq N.$$

In particular, the choice $j = 1$, $k = N$ gives

$$i^{N+N-1} C \cdot L_{1,N} = \pm i a_N.$$

The above discussion implies that

$$i\mathcal{A} \subset \mathcal{L} := \mathrm{span}_{\mathbb{R}} \{i^{k-j} L_{j,k}, i^{N+k-j} C \cdot L_{j,k}\}_{1 \leq j < k \leq N} \subseteq \langle \mathcal{A} \rangle_{\mathrm{Lie}}.$$

We claim that the vector space $\mathcal{L}$ is closed under the Lie bracket. Indeed, we already know that $\mathrm{span}_{\mathbb{R}} \{i^{k-j} L_{j,k}\}$ is closed. For the other brackets, we use that from Equation (G.28), we have:

$$\begin{aligned} [i^{k-j} L_{j,k}, i^{N+m-l} C \cdot L_{l,m}] &= i^N C \cdot [i^{k-j} L_{j,k}, i^{m-l} L_{l,m}], \\ [i^{N+k-j} C \cdot L_{j,k}, i^{N+m-l} C \cdot L_{l,m}] &= [i^{k-j} L_{j,k}, i^{m-l} L_{l,m}]. \end{aligned}$$

As the Lie algebra $\mathcal{L}$ contains $i\mathcal{A}$, it must contain $\langle \mathcal{A} \rangle_{\mathrm{Lie}}$. Therefore, $\mathcal{L} = \langle \mathcal{A} \rangle_{\mathrm{Lie}}$.

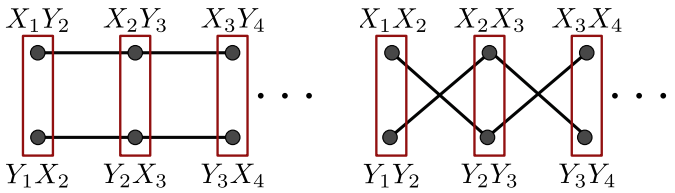
Using Equation (G.28) again (or from the above brackets), we see that

$$\langle \mathcal{A} \rangle_{\mathrm{Lie}} = \mathrm{span}_{\mathbb{R}} \{(I^{\otimes n} + i^N C) \cdot i^{k-j} L_{j,k}\}_{1 \leq j < k \leq N} \oplus \mathrm{span}_{\mathbb{R}} \{(I^{\otimes n} - i^N C) \cdot i^{k-j} L_{j,k}\}_{1 \leq j < k \leq N}$$

is isomorphic as a Lie algebra to a direct sum of two copies of $\mathrm{span}_{\mathbb{R}} \{i^{k-j} L_{j,k}\}_{1 \leq j < k \leq N} \cong \mathfrak{so}(N)$. Therefore, $\langle \mathcal{A} \rangle_{\mathrm{Lie}} \cong \mathfrak{so}(N) \oplus \mathfrak{so}(N)$. □

Remark G.3.2. *A circular frustration graph corresponds to free fermionic evolution controlled with one ancilla, where the ancilla degree of freedom can be readily found as the operator C defined in Equation* (G.27)*. As expected, this is not the only example. Some periodic* 1*-dimensional spin systems such as TFXY, XY and Kitaev models also have DLAs that are generated from Pauli strings with a circular frustration graph. For those spin models, $C = ZZ \cdots Z$.*

Applying the results of Propositions G.3.1, G.3.2, in the following lemmas we determine the Lie algebras $\mathfrak{a}_k(n)$ and $\mathfrak{a}_k^\circ(n)$ for $k = 1, 2, 4, 8, 14$. Examples are presented in Figures G.2 and G.3.



(a) $\mathfrak{a}_2(n)$ (b) $\mathfrak{a}_4(n)$

Figure G.2: **Frustration graphs for $\mathfrak{a}_2(n)$ and $\mathfrak{a}_4(n)$.** The red boxes denote a generator acting on the site $(i,\ i+1)$. Both frustrations graphs are given by two disjoint lines for any n; hence we can conclude that $\mathfrak{a}_2(n) \cong \mathfrak{a}_4(n)$ (cf. Lemma G.3.3).

Lemma G.3.6. *We have:*

$$\begin{aligned}
\mathfrak{a}_1(n) &\cong \mathfrak{so}(n),\\
\mathfrak{a}_2(n) \cong \mathfrak{a}_4(n) &\cong \mathfrak{so}(n) \oplus \mathfrak{so}(n),\\
\mathfrak{a}_8(n) &\cong \mathfrak{so}(2n-1),\\
\mathfrak{a}_{14}(n) &\cong \mathfrak{so}(2n).
\end{aligned}$$

Proof. The proof is based on the frustration graphs of the generating sets of these Lie algebras (see Appendix G.2.1, G.2.4).

Generators of $\mathfrak{a}_1(n)$ are XY on each adjacent pair of qubits:

$$X_1Y_2, X_2Y_3, X_3Y_4, \ldots, X_{n-1}Y_n.$$

These form a linear frustration graph with $n-1$ vertices, leading to $\mathfrak{a}_1(n) \cong \mathfrak{so}(n)$ (see Figure G.3(a)).

The Lie algebra $\mathfrak{a}_2(n)$ is generated by XY and YX on adjacent pairs of qubits:

$$X_1Y_2, X_2Y_3, X_3Y_4, \ldots, X_{n-1}Y_n \qquad \text{and} \qquad Y_1X_2, Y_2X_3, Y_3X_4, \ldots, Y_{n-1}X_n.$$

Both of these form linear frustration graphs with $n-1$ vertices, and commute with each other (see Figure G.2). Thus, $\mathfrak{a}_2(n) \cong \mathfrak{so}(n) \oplus \mathfrak{so}(n)$. Note that $\mathfrak{a}_4(n) \cong \mathfrak{a}_2(n)$ due to Lemma G.3.3 (see also Figure G.2).

Since $\mathfrak{a}_8 = \text{span}\{XX, XZ, IY\} = \langle XX, IY\rangle_{\text{Lie}}$, we can generate $\mathfrak{a}_8(n)$ by:

$$X_1X_2, Y_2, X_2X_3, Y_3, X_3X_4, Y_4, \ldots, X_{n-1}X_n, Y_n. \tag{G.29}$$

These form a linear frustration graph with $2(n-1)$ vertices; hence $\mathfrak{a}_8(n) \cong \mathfrak{so}(2n-1)$ (see Figure G.3(b)).

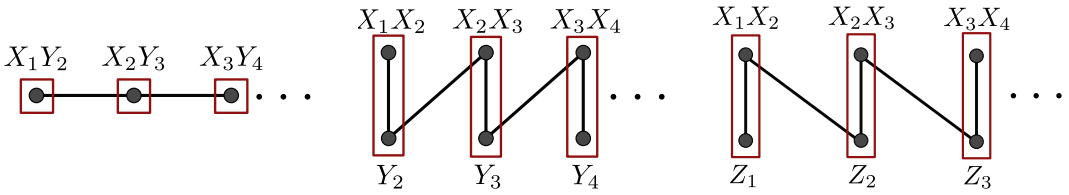
Similarly, note that

$$\mathfrak{a}_{14} = \text{span}\{XX, YY, XY, YX, ZI, IZ\} = \langle XX, ZI, IZ\rangle_{\text{Lie}},$$

because $[XX, ZI] = 2iYX$, $[XX, IZ] = 2iXY$, and $[XY, ZI] = -2iYY$. Thus, $\mathfrak{a}_{14}(n)$ is generated by:

$$Z_1, X_1X_2, Z_2, X_2X_3, Z_3, X_3X_4, Z_4, \ldots, X_{n-1}X_n, Z_n, \tag{G.30}$$

which gives a linear frustration graph with $2n-1$ vertices. Hence, $\mathfrak{a}_{14}(n) \cong \mathfrak{so}(2n)$ (see Figure G.3(c)). □



(a) $\mathfrak{a}_1(n)$ (b) $\mathfrak{a}_8(n)$ (c) $\mathfrak{a}_{14}(n)$

Figure G.3: **Visualization of the frustration graphs of generators of certain Lie algebras.** The red boxes denote a generator acting on the site $(i,\, i+1)$. For the three cases $\mathfrak{a}_1(n)$, $\mathfrak{a}_8(n)$ and $\mathfrak{a}_{14}(n)$, we see that the frustration graph is a line for any n. For (a), the 2-site DLA $\mathfrak{a}_1(2) \cong \mathfrak{u}(1)$, but as n grows we find $\mathfrak{a}_1(n) \cong \mathfrak{so}(n)$. (b) The frustration graph is a line with $2n-2$ vertices; hence $\mathfrak{a}_8(n) \cong \mathfrak{so}(2n-1)$. (c) The frustration graph is a line with $2n-1$ vertices, giving $\mathfrak{a}_{14}(n) \cong \mathfrak{so}(2n)$.

Lemma G.3.7. *We have* $\mathfrak{a}_k^\circ(n) \cong \mathfrak{a}_k(n)^{\oplus 2}$ *for* $k = 1, 2, 14$ *and* $n \geq 3$.

Proof. The Lie algebra $\mathfrak{a}_1^\circ(n)$ is generated by XY applied on adjacent qubits, including periodic boundary conditions:

$$X_1Y_2, X_2Y_3, X_3Y_4, \ldots, X_{n-1}Y_n, X_nY_1.$$

The frustration graph is a circle with n vertices. Therefore, $\mathfrak{a}_1^\circ(n) \cong \mathfrak{so}(n)^{\oplus 2} \cong \mathfrak{a}_1(n)^{\oplus 2}$.

The Lie algebra $\mathfrak{a}_2^\circ(n)$ is generated by XY and YX applied on adjacent qubits with periodic boundary conditions:

$$X_1Y_2, X_2Y_3, X_3Y_4, \ldots, X_{n-1}Y_n, X_nY_1 \qquad \text{and} \qquad Y_1X_2, Y_2X_3, Y_3X_4, \ldots, Y_{n-1}X_n, Y_nX_1.$$

These form two circular frustration graphs with n vertices that are disconnected from each other. Thus, $\mathfrak{a}_2^\circ(n) \cong \mathfrak{so}(n)^{\oplus 2} \oplus \mathfrak{so}(n)^{\oplus 2} \cong \mathfrak{a}_2(n)^{\oplus 2}$.

Using the generating set Equation (G.30) of $\mathfrak{a}_{14}(n)$, we see that $\mathfrak{a}_{14}^\circ(n)$ can be generated by:

$$Z_1, X_1X_2, Z_2, X_2X_3, Z_3, X_3X_4, Z_4, \ldots, X_{n-1}X_n, Z_n, X_nX_1.$$

This leads to a circular frustration graph with $2n$ vertices, so $\mathfrak{a}_{14}^\circ(n) \cong \mathfrak{so}(2n)^{\oplus 2} \cong \mathfrak{a}_{14}(n)^{\oplus 2}$. □

Lemma G.3.8. *We have* $\mathfrak{a}_8^\circ(n) \cong \mathfrak{a}_{14}^\circ(n) \cong \mathfrak{so}(2n)^{\oplus 2}$ *for all* $n \geq 3$.

Proof. The generating sets Equation (G.29) and Equation (G.30) of $\mathfrak{a}_8(n)$ and $\mathfrak{a}_{14}(n)$ are the same after swapping $Y \rightleftharpoons Z$, except that $\mathfrak{a}_8(n)$ does not have Y_1. When the periodic boundary condition is applied, this difference disappears and we obtain that $\mathfrak{a}_8^\circ(n) \cong \mathfrak{a}_{14}^\circ(n)$. □

Lemma G.3.9. *We have* $\mathfrak{a}_4^\circ(n) \cong \begin{cases} \mathfrak{so}(2n)^{\oplus 2}, & n \text{ odd}, \\ \mathfrak{so}(n)^{\oplus 4}, & n \text{ even}. \end{cases}$

Proof. As $\mathfrak{a}_4 = \langle XX, YY \rangle_{\text{Lie}}$, the generators of $\mathfrak{a}_4^\circ(n)$ are:

$$X_1X_2, Y_1Y_2, X_2X_3, Y_2Y_3, \ldots, X_{n-1}X_n, Y_{n-1}Y_n, X_nX_1, Y_nY_1.$$

For odd n, these generators form a circular frustration graph with $2n$ vertices:

$$X_1X_2, Y_2Y_3, X_3X_4, Y_4Y_5, \ldots, Y_{n-1}Y_n, X_nX_1, Y_1Y_2, X_2X_3, Y_3Y_4, X_4X_5, \ldots, X_{n-1}X_n, Y_nY_1.$$

Hence, in this case, $\mathfrak{a}_4^\circ(n) \cong \mathfrak{so}(2n)^{\oplus 2}$.

When n is even, the generators form two disjoint circles with n vertices each:

$$X_1X_2, Y_2Y_3, X_3X_4, \ldots, X_{n-1}X_n, Y_nY_1 \qquad \text{and} \qquad Y_1Y_2, X_2X_3, Y_3Y_4, \ldots, Y_{n-1}Y_n, X_nX_1.$$

In this case, we get $\mathfrak{a}_4^\circ(n) \cong \mathfrak{so}(n)^{\oplus 2} \oplus \mathfrak{so}(n)^{\oplus 2}$. □

Remark G.3.3. *Notice that, although $\mathfrak{a}_2(n) \cong \mathfrak{a}_4(n)$ for all $n \geq 3$, we have $\mathfrak{a}_2^\circ(n) \ncong \mathfrak{a}_4^\circ(n)$ for odd n.*

G.3.4 Stabilizers, commutants, and centralizers

For any Pauli string $A \in \mathcal{P}_k$, we will use the notation $P_A = AAA\cdots \in \mathcal{P}_n$ truncated to the n-th qubit. For example,

$$P_X = XXX\cdots, \qquad P_{YZ} = YZYZ\cdots, \qquad P_{ZY} = ZYZY\cdots,$$
$$P_{XYZ} = XYZXYZ\cdots, \qquad P_{YZX} = YZXYZX\cdots, \qquad P_{ZXY} = ZXYZXY\cdots,$$

where these are viewed as elements of $\mathcal{P}_n$. In particular, $P_I = I\cdots I = I^{\otimes n}$. Recall that $\pm\mathcal{P}_n \cup \pm i\mathcal{P}_n$ is a group under the matrix product, the Pauli group (see Appendix A.1).

For any set of matrices $\mathcal{A} \subseteq \mathbb{C}^{2^n \times 2^n}$, we define its *stabilizer* $\mathrm{St}(\mathcal{A}) \subseteq \mathcal{P}_n$ as the set of all Pauli strings that commute with every element of $\mathcal{A}$. It is clear that $\mathrm{St}(\mathcal{A})$ is closed under multiplication, so after allowing appropriate powers of i it is a group. There are two related, and essentially equivalent, notions called the commutant and centralizer. All of these consist of elements commuting with the given set $\mathcal{A}$ but differ in where such elements are and what structure they form: the stabilizer is a subgroup of the Pauli group (up to factors of $\pm 1, \pm i$); the commutant is a subalgebra of the associative algebra $\mathbb{C}^{2^n \times 2^n}$ of all complex matrices; while the centralizer is a subalgebra of the real Lie algebra $\mathfrak{su}(2^n)$. The precise relations are explained in the following two remarks.

Remark G.3.4. *The* commutant *of a set $\mathcal{A} \subseteq \mathbb{C}^{2^n \times 2^n}$ is defined as the set $\mathcal{A}'$ of all $2^n \times 2^n$ complex matrices that commute with all elements of $\mathcal{A}$. Then $\mathcal{A}'$ is closed under addition, multiplication, and multiplication by any complex scalar, i.e., it is an associative algebra over $\mathbb{C}$. It is easy to see that $\mathcal{A}' = \mathrm{span}_{\mathbb{C}}\mathrm{St}(\mathcal{A})$ is the complex linear span of the stabilizer $\mathrm{St}(\mathcal{A})$.*

Note that $\mathcal{A} \subseteq \mathcal{A}'' := (\mathcal{A}')'$. By (a finite-dimensional version of) von Neumann's Double Commutant Theorem (see e.g. [315], Theorem 6.2.5), $\mathcal{A}''$ is the associative algebra

generated by $\mathcal{A}$. In particular, for a complex vector space $\mathcal{A}$, we have $\mathcal{A}'' = \mathcal{A} + \mathbb{C}I^{\otimes n}$ if and only if $\mathcal{A}$ is closed under multiplication.

Remark G.3.5. *The* centralizer *of a set $\mathcal{A} \subseteq \mathfrak{su}(2^n)$ is the set $\mathfrak{su}(2^n)^{\mathcal{A}} \subseteq \mathfrak{su}(2^n)$ of all traceless skew-Hermitian $2^n \times 2^n$ matrices that commute with all elements of $\mathcal{A}$. Then $\mathfrak{su}(2^n)^{\mathcal{A}}$ is closed under addition, commutator, and multiplication by any real scalar, i.e., it is a Lie subalgebra of $\mathfrak{su}(2^n)$. It is easy to see that $\mathfrak{su}(2^n)^{\mathcal{A}} = \operatorname{span}(\operatorname{St}(\mathcal{A}) \setminus \{I^{\otimes n}\})$ is the real span of $i(\operatorname{St}(\mathcal{A}) \setminus \{I^{\otimes n}\})$.*

When $\mathfrak{s} \subseteq \mathfrak{su}(2^n)$ is a Lie subalgebra, a given Pauli string commutes with all elements of $\mathfrak{s}$ if and only if it commutes with all generators of $\mathfrak{s}$. In other words, we have

$$\operatorname{St}(\langle \mathcal{A} \rangle_{\text{Lie}}) = \operatorname{St}(\mathcal{A}). \tag{G.31}$$

Thus, to determine the stabilizers of our Lie subalgebras of $\mathfrak{su}(2^n)$, it suffices to find the stabilizers of their generating sets. In the case $n = 2$, it is easy to find the answer by inspection, which is given as follows:

$$\begin{aligned}
\operatorname{St}(\mathfrak{a}_0) &= \{II, IX, XI, XX, YY, YZ, ZY, ZZ\},\\
\operatorname{St}(\mathfrak{a}_1) &= \{II, IY, XI, XY, YX, YZ, ZX, ZZ\},\\
\operatorname{St}(\mathfrak{a}_2) &= \{II, XY, YX, ZZ\},\\
\operatorname{St}(\mathfrak{a}_3) &= \operatorname{St}(\mathfrak{a}_6) = \{II, XX, YZ, ZY\},\\
\operatorname{St}(\mathfrak{a}_4) &= \operatorname{St}(\mathfrak{a}_7) = \{II, XX, YY, ZZ\},\\
\operatorname{St}(\mathfrak{a}_5) &= \operatorname{St}(\mathfrak{a}_{10}) = \{II, XY, YZ, ZX\},\\
\operatorname{St}(\mathfrak{a}_8) &= \{II, XI, YY, ZY\},\\
\operatorname{St}(\mathfrak{a}_9) &= \{II, XI, YX, ZX\},\\
\operatorname{St}(\mathfrak{a}_{11}) &= \{II, XY\},\\
\operatorname{St}(\mathfrak{a}_{12}) &= \operatorname{St}(\mathfrak{a}_{17}) = \operatorname{St}(\mathfrak{a}_{19}) = \{II, YZ\},\\
\operatorname{St}(\mathfrak{a}_{13}) &= \operatorname{St}(\mathfrak{a}_{20}) = \{II, XX\},\\
\operatorname{St}(\mathfrak{a}_{14}) &= \{II, ZZ\},\\
\operatorname{St}(\mathfrak{a}_{15}) &= \operatorname{St}(\mathfrak{b}_2) = \operatorname{St}(\mathfrak{b}_4) = \{II, XI\},\\
\operatorname{St}(\mathfrak{b}_0) &= \operatorname{St}(\mathfrak{b}_1) = \{II, XX, XI, IX\},\\
\operatorname{St}(\mathfrak{a}_k) &= \operatorname{St}(\mathfrak{b}_3) = \{II\}, \qquad k = 16, 18, 21, 22.
\end{aligned}$$

Using that, we can find the stabilizers of the subalgebras of $\mathfrak{su}(2^n)$.

Proposition G.3.3. *For $n \geq 3$, we have the following stabilizers:*

$$\begin{aligned}
\mathrm{St}(\mathfrak{a}_0(n)) &= \{I, X\}^{\otimes n} \cup \{Y, Z\}^{\otimes n},\\
\mathrm{St}(\mathfrak{a}_2(n)) &= \{P_I, P_{XY}, P_{YX}, P_Z\},\\
\mathrm{St}(\mathfrak{a}_3(n)) &= \mathrm{St}(\mathfrak{a}_6(n)) = \{P_I, P_X, P_{YZ}, P_{ZY}\},\\
\mathrm{St}(\mathfrak{a}_4(n)) &= \mathrm{St}(\mathfrak{a}_7(n)) = \{P_I, P_X, P_Y, P_Z\},\\
\mathrm{St}(\mathfrak{a}_5(n)) &= \mathrm{St}(\mathfrak{a}_{10}(n)) = \{P_I, P_{XYZ}, P_{YZX}, P_{ZXY}\},\\
\mathrm{St}(\mathfrak{a}_8(n)) &= \{P_I, P_Y, X_1, ZY\cdots Y\},\\
\mathrm{St}(\mathfrak{a}_9(n)) &= \{P_I, X_1, Y_1X_2, Z_1X_2\},\\
\mathrm{St}(\mathfrak{a}_{13}(n)) &= \mathrm{St}(\mathfrak{a}_{20}(n)) = \{P_I, P_X\},\\
\mathrm{St}(\mathfrak{a}_{14}(n)) &= \{P_I, P_Z\},\\
\mathrm{St}(\mathfrak{a}_{15}(n)) &= \mathrm{St}(\mathfrak{b}_2(n)) = \mathrm{St}(\mathfrak{b}_4(n)) = \{P_I, X_1\},\\
\mathrm{St}(\mathfrak{b}_0(n)) &= \mathrm{St}(\mathfrak{b}_1(n)) = \{I, X\}^{\otimes n},\\
\mathrm{St}(\mathfrak{a}_k(n)) &= \mathrm{St}(\mathfrak{b}_3(n)) = \{P_I\}, \qquad k = 11, 12, 16{-}19, 21, 22.
\end{aligned}$$

Proof. Recall that, for any subalgebra $\mathfrak{a} \subseteq \mathfrak{su}(4)$, the subalgebra $\mathfrak{a}(n) \subseteq \mathfrak{su}(2^n)$ is generated by all Pauli strings A_iB_{i+1}, where $AB \in \mathfrak{a}$, $1 \leq i \leq n-1$ (see Equation (G.7)). Thus, a Pauli string $P^1 \otimes \cdots \otimes P^n \in \mathcal{P}_n$ is in $\mathrm{St}(\mathfrak{a}(n))$ if and only if $P^i \otimes P^{i+1} \in \mathrm{St}(\mathfrak{a})$ for all $1 \leq i \leq n-1$. Using this observation and the knowledge of all $\mathrm{St}(\mathfrak{a})$, it is straightforward to determine $\mathrm{St}(\mathfrak{a}(n))$.

Let us consider the case of $\mathfrak{a}_2$ as an illustration. Since $\mathrm{St}(\mathfrak{a}_2) = \{II, XY, YX, ZZ\}$, we want to find all Pauli strings, such that for any two consecutive qubits, we have either II, XY, YX, or ZZ. The only possible such strings are $I\cdots I$, $XYXY\cdots$, $YXYX\cdots$, or $Z\cdots Z$. □

The answer for $\mathrm{St}(\mathfrak{a}_1(n))$ is given without proof in Remark G.3.6 below, as it is more complicated but not needed for the rest of the paper. For future use, we will need the centers of some of the above stabilizers. Let us recall that the *center* $Z(G)$ of a group G consists of all $z \in G$ that commute with every $g \in G$. In particular, a group G is Abelian if and only if $Z(G) = G$.

Lemma G.3.10. *For $n \geq 3$, we have the following centers:*

$$Z\big(\mathrm{St}(\mathfrak{a}_2(n))\big) = \begin{cases} \{P_I, P_{XY}, P_{YX}, P_Z\}, & n \text{ even},\\ \{P_I\}, & n \text{ odd}, \end{cases}$$

$$Z\big(\mathrm{St}(\mathfrak{a}_3(n))\big) = Z\big(\mathrm{St}(\mathfrak{a}_6(n))\big) = \begin{cases} \{P_I, P_X, P_{YZ}, P_{ZY}\}, & n \text{ even}, \\ \{P_I\}, & n \text{ odd}, \end{cases}$$

$$Z\big(\mathrm{St}(\mathfrak{a}_4(n))\big) = Z\big(\mathrm{St}(\mathfrak{a}_7(n))\big) = \begin{cases} \{P_I, P_X, P_Y, P_Z\}, & n \text{ even}, \\ \{P_I\}, & n \text{ odd}, \end{cases}$$

$$Z\big(\mathrm{St}(\mathfrak{a}_5(n))\big) = Z\big(\mathrm{St}(\mathfrak{a}_{10}(n))\big) = \begin{cases} \{P_I, P_{XYZ}, P_{YZX}, P_{ZXY}\}, & n \text{ even}, \\ \{P_I\}, & n \text{ odd}, \end{cases}$$

$$Z\big(\mathrm{St}(\mathfrak{a}_{13}(n))\big) = Z\big(\mathrm{St}(\mathfrak{a}_{20}(n))\big) = \{P_I, P_X\},$$

$$Z\big(\mathrm{St}(\mathfrak{a}_{14}(n))\big) = \{P_I, P_Z\},$$

$$Z\big(\mathrm{St}(\mathfrak{a}_{15}(n))\big) = \{P_I, X_1\},$$

$$Z\big(\mathrm{St}(\mathfrak{a}_k(n))\big) = \{P_I\}, \qquad k = 0, 8, 9, 11, 12, 16{-}19, 21, 22.$$

Next, we determine the stabilizers in the periodic case.

Proposition G.3.4. *For $n \geq 3$, we have the following stabilizers:*

$$\mathrm{St}(\mathfrak{a}_0^\circ(n)) = \{I, X\}^{\otimes n} \cup \{Y, Z\}^{\otimes n},$$

$$\mathrm{St}(\mathfrak{a}_2^\circ(n)) = \begin{cases} \{P_I, P_{XY}, P_{YX}, P_Z\}, & n \text{ even}, \\ \{P_I, P_Z\}, & n \text{ odd}, \end{cases}$$

$$\mathrm{St}(\mathfrak{a}_3^\circ(n)) = \mathrm{St}(\mathfrak{a}_6^\circ(n)) = \begin{cases} \{P_I, P_X, P_{YZ}, P_{ZY}\}, & n \text{ even}, \\ \{P_I, P_X\}, & n \text{ odd}, \end{cases}$$

$$\mathrm{St}(\mathfrak{a}_4^\circ(n)) = \mathrm{St}(\mathfrak{a}_7^\circ(n)) = \{P_I, P_X, P_Y, P_Z\},$$

$$\mathrm{St}(\mathfrak{a}_5^\circ(n)) = \mathrm{St}(\mathfrak{a}_{10}^\circ(n)) = \begin{cases} \{P_I, P_{XYZ}, P_{YZX}, P_{ZXY}\}, & n \equiv 0 \mod 3, \\ \{P_I\}, & n \equiv \pm 1 \mod 3, \end{cases}$$

$$\mathrm{St}(\mathfrak{a}_8^\circ(n)) = \{P_I, P_Y\},$$

$$\mathrm{St}(\mathfrak{a}_{13}^\circ(n)) = \mathrm{St}(\mathfrak{a}_{20}^\circ(n)) = \{P_I, P_X\},$$

$$\mathrm{St}(\mathfrak{a}_{14}^\circ(n)) = \{P_I, P_Z\},$$

$$\mathrm{St}(\mathfrak{b}_0^\circ(n)) = \mathrm{St}(\mathfrak{b}_1^\circ(n)) = \{I, X\}^{\otimes n},$$

$$\mathrm{St}(\mathfrak{a}_k^\circ(n)) = \mathrm{St}(\mathfrak{b}_l^\circ(n)) = \{P_I\}, \qquad k = 9, 11, 12, 15{-}19, 21, 22, \quad l = 2, 3, 4.$$

Proof. For any subalgebra $\mathfrak{a} \subseteq \mathfrak{su}(4)$, comparing the definitions of $\mathfrak{a}(n)$ and $\mathfrak{a}^\circ(n)$ (see Equation (G.6), Equation (G.7)), we see that $\mathrm{St}(\mathfrak{a}^\circ(n))$ consists of all Pauli strings $P^1 \otimes \cdots \otimes P^n \in \mathrm{St}(\mathfrak{a}(n))$ such that $P^n \otimes P^1 \in \mathrm{St}(\mathfrak{a})$. Thus, we determine $\mathrm{St}(\mathfrak{a}^\circ(n))$ by inspecting all elements of $\mathrm{St}(\mathfrak{a}(n))$. □

Remark G.3.6. *One can show that* $\mathrm{St}(\mathfrak{a}_1(n)) = \mathrm{St}(\mathfrak{a}_1^\circ(n))$*, and as a group it is generated by the elements* P_Z*,* Y_1X_2*,* Y_2X_3*, …,* $Y_{n-1}X_n$*. This means that* $\mathrm{St}(\mathfrak{a}_1(n))$ *consists of all possible matrix products of these generators. For the center, we have* $Z\big(\mathrm{St}(\mathfrak{a}_1(n))\big) = \{P_I, P_Z\}$*.*

We also find the stabilizers in the permutation-invariant case.

Proposition G.3.5. *For* $n \geq 3$*, we have the following stabilizers:*

$$\begin{aligned}
\mathrm{St}(\mathfrak{a}_0^\pi(n)) &= \{I, X\}^{\otimes n} \cup \{Y, Z\}^{\otimes n},\\
\mathrm{St}(\mathfrak{a}_2^\pi(n)) &= \mathrm{St}(\mathfrak{a}_{14}^\pi(n)) = \{P_I, P_Z\},\\
\mathrm{St}(\mathfrak{a}_4^\pi(n)) &= \mathrm{St}(\mathfrak{a}_7^\pi(n)) = \{P_I, P_X, P_Y, P_Z\},\\
\mathrm{St}(\mathfrak{a}_6^\pi(n)) &= \mathrm{St}(\mathfrak{a}_{20}^\pi(n)) = \{P_I, P_X\},\\
\mathrm{St}(\mathfrak{b}_0^\pi(n)) &= \mathrm{St}(\mathfrak{b}_1^\pi(n)) = \{I, X\}^{\otimes n},\\
\mathrm{St}(\mathfrak{a}_{16}^\pi(n)) &= \mathrm{St}(\mathfrak{b}_3^\pi(n)) = \{P_I\}.
\end{aligned}$$

Proof. For any subalgebra $\mathfrak{a} \subseteq \mathfrak{su}(4)$, from the definition of $\mathfrak{a}^\pi(n)$ (see Equation (G.10)), we see that $\mathrm{St}(\mathfrak{a}^\pi(n))$ consists of all Pauli strings $P^1 \otimes \cdots \otimes P^n \in \mathcal{P}_n$ such that $P^i \otimes P^j \in \mathrm{St}(\mathfrak{a})$ for all $i \neq j$. Moreover, as we explained in Appendix G.2.4, $\mathfrak{a}$ can be assumed itself invariant under the flip of the two qubits; so we only need to consider $\mathfrak{a}_k^\pi(n)$ for $k = 0, 2, 4, 6, 7, 14, 16, 20$ and $\mathfrak{b}_l^\pi(n)$ for $l = 0, 1, 3$. □

We finish this subsection with an important lemma.

Lemma G.3.11. *The Lie algebras* $\mathfrak{a}_k(n)$ *have trivial centers for* $1 \leq k \leq 22$ *and* $n \geq 3$*.*

Proof. Due to Lemma 4.1.1, $\mathfrak{a}_k(n)$ has a basis $\mathcal{B} \subseteq i\mathcal{P}_n \cap \mathfrak{a}_k(n)$ consisting of Pauli strings. Suppose that $\mathfrak{a}_k(n)$ has a central element $c \neq 0$, and write c as a linear combination of basis vectors:

$$c = \sum \alpha_j c_j, \qquad \alpha_j \in \mathbb{R},\ \ \alpha_j \neq 0,\ \ \ c_j \in \mathcal{B}.$$

We claim that all c_j in this expression are central too.

Indeed, suppose that $[b, c_j] \neq 0$ for some index j and a basis vector $b \in \mathcal{B}$. Since $b, c_j \in i\mathcal{P}_n$, we have $[b, [b, c_j]] = -4c_j$ whenever $[b, c_j] \neq 0$, by Corollary A.1.1. Hence,

$$0 = [b, [b, c]] = \sum \alpha_j [b, [b, c_j]] = -4 \sum{}' \alpha_j c_j,$$

where $\sum'$ denotes the sum over all indices j such that $[b, c_j] \neq 0$. This contradicts the fact that all $\alpha_j \neq 0$ and the vectors c_j are linearly independent.

Therefore, without loss of generality, we can take $c \in \mathcal{B}$ to be itself one of the basis vectors. One can verify by inspection that the generators of $\mathfrak{a}_k(n)$ are not central for $1 \leq k \leq 22$ and $n \geq 3$; for example, by checking that $Z\big(\mathrm{St}(\mathfrak{a}_k(n))\big)$ does not contain any of the generators of $\mathfrak{a}_k(n)$. Thus, we can write c in the form

$$c = \mathrm{ad}_{a_1}\mathrm{ad}_{a_2}\cdots\mathrm{ad}_{a_r}(a_{r+1}),$$

for some $r \geq 1$ and generators $a_1, \ldots, a_{r+1}$. Since all generators $a_j \in i\mathcal{P}_n$, we have $a := \mathrm{ad}_{a_2}\cdots\mathrm{ad}_{a_r}(a_{r+1}) \in i\mathcal{P}_n$. Then $c = [a_1, a] \neq 0$, and from Corollary A.1.1, we get $-4a = [a_1, [a_1, a]] = [a_1, c] = 0$, which implies that $c = 0$, a contradiction. □

G.3.5 Upper bounds for $\mathfrak{a}_k(n)$

In this subsection, we establish upper bounds for the Lie algebras $\mathfrak{a}_k(n)$, i.e., we find certain subalgebras $\mathfrak{g}_k(n)^{\theta_k} \subseteq \mathfrak{su}(2^n)$ that contain $\mathfrak{a}_k(n)$. Then, in the next subsection G.3.6, we will prove that these bounds are exact, that is $\mathfrak{a}_k(n) = \mathfrak{g}_k(n)^{\theta_k}$. While $\mathfrak{a}_k(n)$ is defined in terms of its generators, $\mathfrak{g}_k(n)^{\theta_k}$ is defined as the set of elements of $\mathfrak{su}(2^n)$ that are fixed under certain automorphisms and involutions. This will allow us, in the following subsection G.3.7, to identify the Lie algebras $\mathfrak{a}_k(n)$ as direct sums of $\mathfrak{su}$, $\mathfrak{so}$, and $\mathfrak{sp}$ Lie algebras.

We start by recalling that any Pauli string $P \in \mathcal{P}_n$ defines an automorphism of $\mathfrak{su}(2^n)$ by conjugation $a \mapsto PaP$ (recall that $P = P^\dagger = P^{-1}$); see Lemma G.1.2. We will denote by $\mathfrak{su}(2^n)^P$ the set of *fixed points* under this automorphism, i.e., the set of all $a \in \mathfrak{su}(2^n)$ such that $PaP = a$. The latter is equivalent to $Pa = aP$; hence $\mathfrak{su}(2^n)^P$ is equal to the *centralizer* of P, i.e., the set of all $a \in \mathfrak{su}(2^n)$ that commute with P (see Remark G.3.5). More generally, for a set Φ of automorphisms of a Lie algebra $\mathfrak{g}$, we will denote by $\mathfrak{g}^\Phi$ the set of fixed points $a \in \mathfrak{g}$ such that $\phi(a) = a$ for all $\phi \in \Phi$.

Given a subalgebra $\mathfrak{s} \subseteq \mathfrak{su}(2^n)$, recall from Appendix G.3.4, that its stabilizer $\mathrm{St}(\mathfrak{s})$ consists of all Pauli strings $P \in \mathcal{P}_n$ such that $[a, P] = 0$ for every $a \in \mathfrak{s}$. On the other hand, the centralizer $\mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}$ of $\mathrm{St}(\mathfrak{s})$ in $\mathfrak{su}(2^n)$ consists of all $a \in \mathfrak{su}(2^n)$ such that $[a, P] = 0$ for every $P \in \mathrm{St}(\mathfrak{s})$. Hence, by definition,

$$\mathfrak{s} \subseteq \mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}. \tag{G.32}$$

This simple observation will be the key to finding upper bounds for our subalgebras $\mathfrak{a}_k(n)$, because we have already determined their stabilizers in Proposition G.3.3. Here is another observation, which in some cases will allow us to further reduce the upper bound.

Lemma G.3.12. *For any subalgebra $\mathfrak{s} \subseteq \mathfrak{su}(2^n)$, we have*

$$\mathrm{St}(\mathfrak{s}) \cap \mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})} = Z\big(\mathrm{St}(\mathfrak{s})\big) \setminus \{I^{\otimes n}\} \subseteq Z\big(\mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}\big),$$

where $Z(G)$ denotes the center of a group or an algebra G.

Proof. Elements $z \in \mathrm{St}(\mathfrak{s}) \cap \mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}$ satisfy $[z, a] = 0$ for every $a \in \mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}$ since $z \in \mathrm{St}(\mathfrak{s})$, and $[z, P] = 0$ for every $P \in \mathrm{St}(\mathfrak{s})$ since $z \in \mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}$. Hence, such z are central in both $\mathrm{St}(\mathfrak{s})$ and $\mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}$. However, $I^{\otimes n}$ is excluded, because $I^{\otimes n} \notin \mathfrak{su}(2^n)$. □

As the Lie algebras $\mathfrak{s} = \mathfrak{a}_k(n)$ for $1 \leq k \leq 22$, $n \geq 3$ have trivial centers (Lemma G.3.11), for them we can reduce the upper bound $\mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}$ if we quotient by the central elements $Z\big(\mathrm{St}(\mathfrak{s})\big) \setminus \{I^{\otimes n}\}$. Thus, we introduce the notation

$$\mathfrak{g}_k(n) := \mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{a}_k(n))} / \mathrm{span}\big(Z\big(\mathrm{St}(\mathfrak{a}_k(n))\big) \setminus \{I^{\otimes n}\}\big), \tag{G.33}$$

and from the above discussion, we have

$$\mathfrak{a}_k(n) \subseteq \mathfrak{g}_k(n), \qquad 1 \leq k \leq 22, \quad n \geq 3. \tag{G.34}$$

Using Proposition G.3.3 and Lemma G.3.10, we can write explicitly:

$$\mathfrak{g}_3(n) = \mathfrak{g}_6(n) = \begin{cases} \mathfrak{su}(2^n)^{\{P_X, P_{YZ}, P_{ZY}\}} / \mathrm{span}\{P_X, P_{YZ}, P_{ZY}\}, & n \text{ even}, \\ \mathfrak{su}(2^n)^{\{P_X, P_{YZ}, P_{ZY}\}}, & n \text{ odd}, \end{cases} \tag{G.35}$$

$$\mathfrak{g}_5(n) = \mathfrak{g}_{10}(n) = \begin{cases} \mathfrak{su}(2^n)^{\{P_{XYZ}, P_{YZX}, P_{ZXY}\}} / \mathrm{span}\{P_{XYZ}, P_{YZX}, P_{ZXY}\}, & n \text{ even}, \\ \mathfrak{su}(2^n)^{\{P_{XYZ}, P_{YZX}, P_{ZXY}\}}, & n \text{ odd}, \end{cases} \tag{G.36}$$

$$\mathfrak{g}_7(n) = \begin{cases} \mathfrak{su}(2^n)^{\{P_X, P_Y, P_Z\}} / \mathrm{span}\{P_X, P_Y, P_Z\}, & n \text{ even}, \\ \mathfrak{su}(2^n)^{\{P_X, P_Y, P_Z\}}, & n \text{ odd}, \end{cases} \tag{G.37}$$

$$\mathfrak{g}_9(n) = \mathfrak{su}(2^n)^{\{X_1, Y_1X_2, Z_1X_2\}}, \tag{G.38}$$

$$\mathfrak{g}_{11}(n) = \mathfrak{g}_{16}(n) = \mathfrak{su}(2^n), \tag{G.39}$$

$$\mathfrak{g}_{13}(n) = \mathfrak{g}_{20}(n) = \mathfrak{su}(2^n)^{P_X} / \mathrm{span}\{P_X\}, \tag{G.40}$$

$$\mathfrak{g}_{15}(n) = \mathfrak{su}(2^n)^{X_1} / \mathrm{span}\{X_1\}. \tag{G.41}$$

It turns out that in some cases the inclusions Equation (G.34) are strict, and we need to reduce the Lie algebras $\mathfrak{g}_k(n)$ further to smaller subalgebras. We do that by finding suitable involutions and then taking their fixed points (see Appendix G.1).

Theorem G.3.1. *We have*

$$\mathfrak{a}_k(n) = \mathfrak{g}_k(n), \qquad k = 6, 7, 10, 13, 15, 20, \quad n \geq 3. \tag{G.42}$$

In the remaining cases, there exists an involution θ_k of $\mathfrak{g}_k(n)$, such that

$$\mathfrak{a}_k(n) = \mathfrak{g}_k(n)^{\theta_k}, \qquad k = 3, 5, 9, 11, 16, \quad n \geq 3, \tag{G.43}$$

is the set of fixed points under θ_k.

In the remainder of this subsection, we will construct the involution θ_k explicitly, and will check that $\mathfrak{a}_k(n) \subseteq \mathfrak{g}_k(n)^{\theta_k}$. The opposite inclusion will be proved in the next subsection. Then, in Appendix G.3.7, we will identify the Lie algebras $\mathfrak{g}_k(n)^{\theta_k}$ with those from Theorem 4.5.1. For $k = 9, 11, 16$, the involution θ_k will have the form (cf. Lemma G.1.2):

$$\theta(g) = -Qg^TQ \tag{G.44}$$

for some given Pauli string $Q \in \mathcal{P}_n$.

Lemma G.3.13. *For any fixed Pauli string $Q \in \mathcal{P}_n$, Equation* (G.44) *defines an involution of $\mathfrak{su}(2^n)$, which restricts to an involution of $\mathfrak{g}_k(n)$ for all k.*

Proof. We already know from Lemma G.1.2 that θ is an involution of $\mathfrak{su}(2^n)$, so we only need to check that $\theta(g) \in \mathfrak{g}_k(n)$ for all $g \in \mathfrak{g}_k(n)$. As before, write $\mathfrak{s} = \mathfrak{a}_k(n)$ for short. Consider an element $g \in \mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}$, which means that $[g, P] = 0$ for all $P \in \mathrm{St}(\mathfrak{s})$. Then

$$\theta(P) = -QP^TQ = \pm P,$$

because $P^T = \pm P$ and $PQ = \pm QP$ for any two Pauli strings $P, Q \in \mathcal{P}_n$ (the signs here are not coordinated). Hence,

$$[\theta(g), P] = \pm[\theta(g), \theta(P)] = \pm\theta([g, P]) = 0,$$

which implies that $\theta(g) \in \mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{s})}$. Furthermore,

$$\theta(P) = \pm P \in \mathrm{span}\big(Z\big(\mathrm{St}(\mathfrak{s})\big) \setminus \{I^{\otimes n}\}\big) \quad \text{for all} \quad P \in Z\big(\mathrm{St}(\mathfrak{s})\big) \setminus \{I^{\otimes n}\}.$$

Therefore, $\theta(g) \in \mathfrak{g}_k(n)$ for $g \in \mathfrak{g}_k(n)$. □

Let us record the following consequence of the proof of Lemma G.3.13, which will be useful later.

Corollary G.3.1. *Every element of $\mathfrak{g}_k(n)^{\theta_k}$ is a linear combination of Pauli strings that are themselves in $\mathfrak{g}_k(n)^{\theta_k}$, i.e.,*

$$\mathfrak{g}_k(n)^{\theta_k} = \mathrm{span}_{\mathbb{R}}\big(i\mathcal{P}_n \cap \mathfrak{g}_k(n)^{\theta_k}\big).$$

Proof. We saw in the proof of Lemma G.3.13 that $\theta_k(P) = \pm P$ for any Pauli string $P \in \mathcal{P}_n$. Similarly, for any given $S \in \mathrm{St}(\mathfrak{a}_k(n))$, we have $SPS = \pm P$. Elements g of $\mathfrak{g}_k(n)^{\theta_k}$ are determined by the conditions

$$SgS = g = \theta_k(g) \qquad \text{for all} \quad S \in \mathrm{St}(\mathfrak{a}_k(n)).$$

Writing $g \in \mathfrak{su}(2^n)$ as i times a real linear combination of Pauli strings, we see that g satisfies these conditions if and only if every summand does. □

Now we go back to the construction of the involutions θ_k.

Lemma G.3.14. *For $k = 9, 11, 16$, we define $\theta_k(g) = -Q_k g^T Q_k$, where the Pauli strings Q_k are given as follows:*

$$Q_9 = IYZ\cdots Z, \tag{G.45}$$

$$Q_{11} = Q_{16} = I\cdots I \quad \Rightarrow \quad \theta_{11}(g) = \theta_{16}(g) = -g^T. \tag{G.46}$$

Then $\mathfrak{a}_k(n) \subseteq \mathfrak{g}_k(n)^{\theta_k}$.

Proof. We already know that $\mathfrak{a}_k(n) \subseteq \mathfrak{g}_k(n)$, so we only need to check that $\theta_k(g) = g$ for all $g \in \mathfrak{a}_k(n)$. It is enough to check this only for the generators g of $\mathfrak{a}_k(n)$, because $\theta_k([a,b]) = [\theta_k(a), \theta_k(b)]$.

For $k = 9$, we take $g = X_iY_{i+1}$ or $g = X_iZ_{i+1}$. In the first case, $g^T = -g$ and $gQ_9 = Q_9g$; while in the second case, $g^T = g$ and $gQ_9 = -Q_9g$. Hence, in both cases we have $\theta_9(g) = g$.

For $k = 11$, the generators are $g = X_iY_{i+1}, Y_iX_{i+1}, Y_iZ_{i+1}$; while for $k = 16$, the generators are $g = X_iY_{i+1}, Y_iX_{i+1}, Y_iZ_{i+1}, Z_iY_{i+1}$. All of them satisfy $g^T = -g$. □

The remaining cases $k = 3$ and $k = 5$ are a little more complicated. The trick is to first embed $\mathfrak{a}_3(n)$ and $\mathfrak{a}_5(n)$ as subalgebras of $\mathfrak{a}_7(n)$. Recall from Appendix G.3.2 that $\mathfrak{a}_3(n) \subseteq \mathfrak{a}_6(n)$ and $\mathfrak{a}_6(n) \cong \mathfrak{a}_7(n)$ under the automorphism φ_n of $\mathfrak{su}(2^n)$ that swaps (up to a sign) $Y \rightleftharpoons Z$ on all even qubits (see Equation (G.18), Equation (G.19)). Then $\tilde{\mathfrak{a}}_3(n) := \varphi_n\mathfrak{a}_3(n) \subset \mathfrak{a}_7(n)$. Likewise, we have $\mathfrak{a}_5(n) \subseteq \mathfrak{a}_{10}(n)$ and $\mathfrak{a}_{10}(n) \cong \mathfrak{a}_7(n)$ under the

automorphism γ_n of $\mathfrak{su}(2^n)$ that applies on the j-th qubit γ^j, where γ is the cycle $X \mapsto Z \mapsto Y \mapsto X$ (see Equation (G.21), Equation (G.22)). Then $\tilde{\mathfrak{a}}_5(n) := \gamma_n \mathfrak{a}_5(n) \subset \mathfrak{a}_7(n)$.

We consider the involutions

$$\tilde{\theta}_k(g) = -Q_k g^T Q_k, \qquad k = 3, 5, \tag{G.47}$$

where

$$Q_3 = P_{ZIYX} = (Z_1 Y_3 X_4)(Z_5 Y_7 X_8)(Z_9 Y_{11} X_{12}) \cdots , \tag{G.48}$$
$$Q_5 = P_{IYZ} = (Y_2 Z_3)(Y_5 Z_6)(Y_8 Z_9)(Y_{11} Z_{12}) \cdots . \tag{G.49}$$

Then we define

$$\theta_3 := \varphi_n^{-1} \tilde{\theta}_3 \varphi_n, \qquad \theta_5 := \gamma_n^{-1} \tilde{\theta}_5 \gamma_n. \tag{G.50}$$

Lemma G.3.15. *For $k = 3, 5$, and θ_k defined as above, we have $\mathfrak{a}_k(n) \subseteq \mathfrak{g}_k(n)^{\theta_k}$.*

Proof. As we already know that $\mathfrak{a}_k(n) \subseteq \mathfrak{g}_k(n)$, we only need to check that $\theta_k(g) = g$ for all generators g of $\mathfrak{a}_k(n)$. By conjugation, it is equivalent to check that $\tilde{\theta}_k(g) = g$ for the generators g of $\tilde{\mathfrak{a}}_k(n)$. Applying φ_n to the generators of $\mathfrak{a}_3(n)$, we find that the generators of $\tilde{\mathfrak{a}}_3(n)$ are:

$$X_1X_2, X_2X_3, X_3X_4, X_4X_5, X_5X_6, X_6X_7, X_7X_8, \ldots,$$
$$Y_1Y_2, Z_2Z_3, Y_3Y_4, Z_4Z_5, Y_5Y_6, Z_6Z_7, Y_7Y_8, \ldots .$$

Similarly, applying γ_n to the generators of $\mathfrak{a}_5(n)$, we find the generators of $\tilde{\mathfrak{a}}_5(n)$:

$$Z_1Z_2, Y_2Y_3, X_3X_4, Z_4Z_5, Y_5Y_6, X_6X_7, Z_7Z_8, Y_8Y_9, X_9X_{10}, \ldots,$$
$$X_1X_2, Z_2Z_3, Y_3Y_4, X_4X_5, Z_5Z_6, Y_6Y_7, X_7X_8, Z_8Z_9, Y_9Y_{10}, \ldots .$$

We observe that all generators g above satisfy $g^T = g$ and $gQ_k = -Q_k g$; hence, $\tilde{\theta}_k(g) = g$. □

G.3.6 Lower bounds for $\mathfrak{a}_k(n)$

In this subsection, we prove that the upper bounds $\mathfrak{a}_k(n) \subseteq \mathfrak{g}_k(n)^{\theta_k}$ established in Appendix G.3.5 are exact. The proof will rely on the next lemma.

Lemma G.3.16. *Let $\mathfrak{s}$ be a Lie subalgebra of $\mathfrak{su}(2^n)$. If $\mathrm{ad}_{a_1} \cdots \mathrm{ad}_{a_r}(b) \in \mathfrak{s} \setminus \{0\}$ for some Pauli strings $a_1, \ldots, a_r \in i\mathcal{P}_n \cap \mathfrak{s}$ and $b \in i\mathcal{P}_n$, then $b \in \mathfrak{s}$.*

Proof. Using induction on r, it is enough to prove the statement for $r = 1$. In this case, it follows from Corollary A.1.1: $[a_1, b] \in \mathfrak{s} \setminus \{0\}$ implies that $b = -4[a_1, [a_1, b]] \in \mathfrak{s}$. □

In order to prove that $\mathfrak{a}_k(n) = \mathfrak{g}_k(n)^{\theta_k}$, we want to show that every element $b \in \mathfrak{g}_k(n)^{\theta_k}$ is in $\mathfrak{a}_k(n)$. Since, by Corollary G.3.1, b is a linear combination of Pauli strings that are themselves in $\mathfrak{g}_k(n)^{\theta_k}$, we can assume without loss of generality that $b \in i\mathcal{P}_n \cap \mathfrak{g}_k(n)^{\theta_k}$. Then the strategy of the proof is to take suitable commutators of b with elements of $i\mathcal{P}_n \cap \mathfrak{a}_k(n)$ to produce a Pauli string $c \in i\mathcal{P}_n \cap \mathfrak{g}_k(n)^{\theta_k}$ that has I in one of its positions. Erasing the I will give an element of $\mathfrak{g}_k(n-1)^{\theta_k}$, which by induction will be in $\mathfrak{a}_k(n-1)$. From here, we will obtain that $c \in \mathfrak{a}_k(n)$, and then we can conclude that $b \in \mathfrak{a}_k(n)$ due to Lemma G.3.16.

In order to realize the above strategy, we will have to do a detailed case-by-case analysis. We start with the cases $k = 3, 5, 7$, for which we need the following lemmas.

Lemma G.3.17. *We have vector space decompositions:*

$$\begin{aligned}
\mathfrak{a}_3(4) &= (I \otimes \mathfrak{a}_3(3)) + P_X \cdot (I \otimes \mathfrak{a}_3(3)) + P_{YZ} \cdot (I \otimes \mathfrak{a}_3(3)) + P_{ZY} \cdot (I \otimes \mathfrak{a}_3(3)),\\
\mathfrak{a}_5(6) &= (I \otimes \mathfrak{a}_5(5)) + P_{XYZ} \cdot (I \otimes \mathfrak{a}_5(5)) + P_{YZX} \cdot (I \otimes \mathfrak{a}_5(5)) + P_{ZXY} \cdot (I \otimes \mathfrak{a}_5(5)),\\
\mathfrak{a}_7(4) &= (I \otimes \mathfrak{a}_7(3)) + P_X \cdot (I \otimes \mathfrak{a}_7(3)) + P_Y \cdot (I \otimes \mathfrak{a}_7(3)) + P_Z \cdot (I \otimes \mathfrak{a}_7(3)),
\end{aligned}$$

where $\cdot$ denotes the component-wise matrix product.

Proof. By inspection. Here are all Pauli strings in $\mathfrak{a}_3(4)$ (after multiplication by i):

$$\begin{array}{cccccccccc}
IIXX, & IIYZ, & IXXI, & IXZZ, & IYIY, & IYXZ, & IYYX, & IYZI, & IZIZ, & IZXY,\\
XIIX, & XIYY, & XXII, & XXZY, & XYIZ, & XYXY, & XZIY, & XZXZ, & XZYX, & XZZI,\\
YIYI, & YIZX, & YXIY, & YXXZ, & YXYX, & YXZI, & YYXI, & YYZZ, & YZII, & YZZY,\\
ZIIY, & ZIXZ, & ZIYX, & ZIZI, & ZXYI, & ZXZX, & ZYXX, & ZYYZ, & ZZIX, & ZZYY.
\end{array}$$

The Pauli strings in $\mathfrak{a}_7(4)$ (after multiplication by i) are:

$$\begin{array}{cccccccc}
IIXX, & IIYY, & IIZZ, & IXIX, & IXXI, & IXYZ, & IXZY, & IYIY,\\
IYXZ, & IYYI, & IYZX, & IZIZ, & IZXY, & IZYX, & IZZI, & \\
XIIX, & XIXI, & XIYZ, & XIZY, & XXII, & XXYY, & XXZZ, & XYIZ,\\
XYXY, & XYYX, & XYZI, & XZIY, & XZXZ, & XZYI, & XZZX, & \\
YIIY, & YIXZ, & YIYI, & YIZX, & YXIZ, & YXXY, & YXYX, & YXZI,\\
YYII, & YYXX, & YYZZ, & YZIX, & YZXI, & YZYZ, & YZZY, & \\
ZIIZ, & ZIXY, & ZIYX, & ZIZI, & ZXIY, & ZXXZ, & ZXYI, & ZXZX,
\end{array}$$

$$ZYIX, \quad ZYXI, \quad ZYYZ, \quad ZYZY, \quad ZZII, \quad ZZXX, \quad ZZYY.$$

We have: $|\mathfrak{a}_5(5)| = 120$, $|\mathfrak{a}_5(6)| = 480$, and there are 120 elements in $\mathfrak{a}_5(6)$ starting with each of the letters I, X, Y, or Z. The remaining claims were verified using Excel. □

For each pair $(k, n) = (3, 4), (5, 6), (7, 4)$, consider the subalgebra $\mathfrak{s} = \mathfrak{a}_k(n) \subset \mathfrak{su}(2^n)$. Recall from Proposition G.3.3 that the stabilizer $\mathrm{St}(\mathfrak{s})$ is given by:

$$\begin{aligned}
\mathrm{St}(\mathfrak{a}_3(4)) &= \{P_I, P_X, P_{YZ}, P_{ZY}\},\\
\mathrm{St}(\mathfrak{a}_5(6)) &= \{P_I, P_{XYZ}, P_{YZX}, P_{ZXY}\},\\
\mathrm{St}(\mathfrak{a}_7(4)) &= \{P_I, P_X, P_Y, P_Z\},
\end{aligned}$$

and $\mathrm{St}(\mathfrak{s})$ is an Abelian group under the matrix product $\cdot$. We can state Lemma G.3.17 succinctly as

$$\mathfrak{a}_k(n) = \mathrm{St}(\mathfrak{a}_k(n)) \cdot (I \otimes \mathfrak{a}_k(n-1)), \qquad (k, n) = (3, 4), (5, 6), (7, 4). \tag{G.51}$$

Lemma G.3.18. *Let $\mathfrak{s} = \mathfrak{a}_k(n)$ for $(k, n) = (3, 4), (5, 6), (7, 4)$. Consider any Pauli string $a \in i\mathcal{P}_n \cap \mathfrak{g}_k(n)$ not starting with I in the first qubit. Then there exists a basis vector $b \in \mathfrak{s}$ such that $[a, b] \neq 0$ and $[a, b] \in I \otimes \mathfrak{su}(2^{n-1})$.*

Proof. Let us write all Pauli strings up to a suitable multiple of i that makes them skew-Hermitian. Consider the case when $a = XA$ starts with X; the cases when it starts with Y or Z are similar. If there is $b = XB \in \mathfrak{s}$ such that $[A, B] \neq 0$, then $[a, b] = I[A, B] \neq 0$ and we are done. By Lemma G.3.17, any $b = XB \in \mathfrak{s}$ can be written in the form $b = C \cdot (ID)$, where $D \in \mathfrak{a}_k(n-1)$ and $C \in \mathrm{St}(\mathfrak{s})$; explicitly, $C = P_X, P_{XYZ}, P_X$ for $\mathfrak{s} = \mathfrak{a}_3(4), \mathfrak{a}_5(6), \mathfrak{a}_7(4)$, respectively. Similarly, as $C \cdot C = P_I$, we can write $a = C \cdot (IE)$ for some $E \in \mathcal{P}_{n-1}$. Suppose that $[a, b] = 0$. Then $[C, a] = [C, b] = 0$ imply that $[E, D] = 0$. Since this is true for all $D \in \mathfrak{a}_k(n-1)$, it follows that $E \in \mathrm{St}(\mathfrak{a}_k(n-1))$, from where $a = C \cdot (IE) \in \mathrm{St}(\mathfrak{s})$. This is a contradiction, because such elements are factored out from $\mathfrak{g}_k(n)$; see Equation (G.33) and Lemma G.3.12. □

Lemma G.3.19. *Consider $(k, m) = (3, 4), (5, 6), (7, 4)$, and let $n \geq m$. Then for any Pauli string $a \in i\mathcal{P}_n \cap \mathfrak{g}_k(n)$, there exist basis vectors $b_1, \ldots, b_r \in \mathfrak{a}_k(n)$, $r \geq 0$, such that $\mathrm{ad}_{b_1} \cdots \mathrm{ad}_{b_r}(a) \in (I^{\otimes(n-m+1)} \otimes \mathfrak{su}(2^{m-1})) \setminus \{0\}$ (with $r = 0$ corresponding to a).*

Proof. The proof is by induction on n, the base $n = m$ being Lemma G.3.18. For the step of the induction, suppose that $n > m$ and the statement holds for $\mathfrak{g}_k(n-1)$. Again, let us write all Pauli strings up to a suitable multiple of i. Take any Pauli string $a \in \mathfrak{g}_k(n)$,

and write it as $a = AD$ where A is the substring consisting of the first m Paulis. Then $A \in \mathfrak{su}(2^m)^{\mathrm{St}(\mathfrak{a}_k(m))}$.

If $A \notin \mathrm{St}(\mathfrak{a}_k(m))$, we can use Lemma G.3.18 to find $B \in \mathfrak{a}_k(m)$ such that $[B, A]$ starts with I. Then $[b, a] = IC$ starts with I for $b = BI \cdots I \in \mathfrak{a}_k(n)$. After that, we can apply the inductive assumption for $C \in \mathfrak{g}_k(n-1)$.

If $A \in \mathrm{St}(\mathfrak{a}_k(m))$, we repeat the same argument for the substring E of a corresponding to positions $2, \ldots, m+1$. When $E \notin \mathrm{St}(\mathfrak{a}_k(m))$, we can make it to start with I, which will force the substring $A \notin \mathrm{St}(\mathfrak{a}_k(m))$. If both $A, E \in \mathrm{St}(\mathfrak{a}_k(m))$, putting them together we get that the first $m+1$ positions of a are in $\mathrm{St}(\mathfrak{a}_k(m+1))$. Continuing this way will give us $a \in \mathrm{St}(\mathfrak{a}_k(n))$, which is a contradiction, because such elements are factored out from $\mathfrak{g}_k(n)$ (cf. Equation (G.33) and Lemma G.3.12). □

Recall that, in Appendix G.3.2, we constructed certain automorphisms φ_n and γ_n of $\mathfrak{su}(2^n)$ such that the images $\tilde{\mathfrak{a}}_3(n) := \varphi_n \mathfrak{a}_3(n)$ and $\tilde{\mathfrak{a}}_5(n) := \gamma_n \mathfrak{a}_5(n)$ are subalgebras of $\mathfrak{a}_7(n)$. Note that, after these transformations, their stabilizers become equal:

$$\mathrm{St}(\tilde{\mathfrak{a}}_3(n)) = \mathrm{St}(\tilde{\mathfrak{a}}_5(n)) = \mathrm{St}(\mathfrak{a}_7(n)) = \{P_I, P_X, P_Y, P_Z\}. \tag{G.52}$$

Hence, we have (recall Equation (G.47)–Equation (G.50)):

$$\tilde{\mathfrak{a}}_k(n) \subseteq \mathfrak{g}_7(n)^{\tilde{\theta}_k}, \qquad k = 3, 5. \tag{G.53}$$

Lemma G.3.20. *We have $\mathfrak{a}_7(n) = \mathfrak{g}_7(n)$ and equalities in Equation* (G.53)*. Consequently, $\mathfrak{a}_k(n) = \mathfrak{g}_k(n)^{\theta_k}$ for $k = 3, 5$.*

Proof. As before, let $(k, m) = (3, 4), (5, 6), (7, 4)$. In order to include the case $k = 7$ in Equation (G.53), we let $\tilde{\mathfrak{a}}_7(n) = \mathfrak{a}_7(n)$ and $\tilde{\theta}_7$ be the identity. The statement is true for all $2 \leq n \leq m$ by inspection. For $n \geq m$, we prove it by induction on n. Consider any $a \in \mathfrak{g}_7(n)^{\tilde{\theta}_k}$. By Lemma G.3.19, we can find $b_1, \ldots, b_r \in \tilde{\mathfrak{a}}_k(n)$ such that $\mathrm{ad}_{b_1} \cdots \mathrm{ad}_{b_r}(a) = I \cdots ID$ for some $D \in \mathfrak{su}(2^{m-1}) \setminus \{0\}$. Since $b_i \in \tilde{\mathfrak{a}}_k(n) \subseteq \mathfrak{g}_7(n)^{\tilde{\theta}_k}$, we get that $D \in \mathfrak{g}_7(m)^{\tilde{\theta}_k} = \tilde{\mathfrak{a}}_k(m)$. Therefore, $a \in \tilde{\mathfrak{a}}_k(n)$ due to Lemma G.3.16. □

Remark G.3.7. *It follows from Lemma G.3.20 that Equation* (G.51) *holds for all (k, n) such that: $k = 3$, $n \equiv 0 \mod 4$; $k = 5$, $n \equiv 0 \mod 6$; $k = 7$, $n \equiv 0 \mod 2$. As a consequence, for such (k, n), we have $\mathfrak{a}_k(n) \cong \mathfrak{a}_k(n-1)^{\oplus 4}$ as a Lie algebra.*

Now that we are done with the cases $k = 3, 5, 7$, we derive the cases $k = 6, 10$ from $k = 7$ and the isomorphisms $\mathfrak{a}_6(n) \cong \mathfrak{a}_{10}(n) \cong \mathfrak{a}_7(n)$ obtained in Lemmas G.3.4 and G.3.5.

Lemma G.3.21. *We have* $\mathfrak{a}_k(n) = \mathfrak{g}_k(n)$ *for* $k = 6, 10$.

Proof. Recall from Appendix G.3.2 that we have an isomorphism $\varphi_n \colon \mathfrak{a}_6(n) \cong \mathfrak{a}_7(n)$ that up to a sign swaps $Y \rightleftharpoons Z$ on every even qubit (see Equation (G.18), Equation (G.19)). Under φ_n the stabilizers

$$\begin{aligned} \mathrm{St}(\mathfrak{a}_6(n)) &= \{P_I, P_X, P_{YZ}, P_{ZY}\}, \\ \mathrm{St}(\mathfrak{a}_7(n)) &= \{P_I, P_X, P_Y, P_Z\} \end{aligned}$$

are sent to each other; hence $\mathfrak{g}_6(n) \cong \mathfrak{g}_7(n)$. Since $\mathfrak{a}_7(n) = \mathfrak{g}_7(n)$ by Lemma G.3.20, it follows that $\mathfrak{a}_6(n) = \mathfrak{g}_6(n)$.

Similarly, we have an isomorphism $\gamma_n \colon \mathfrak{a}_{10}(n) \cong \mathfrak{a}_7(n)$ given by applying on the j-th qubit ($j = 1, \ldots, n$) the permutation γ^j, where γ is the cycle $X \mapsto Z \mapsto Y \mapsto X$ (see Equation (G.21), Equation (G.22)). Then γ_n sends

$$\mathrm{St}(\mathfrak{a}_{10}(n)) = \{P_I, P_{XYZ}, P_{YZX}, P_{ZXY}\}$$

to $\mathrm{St}(\mathfrak{a}_7(n))$; hence $\mathfrak{g}_{10}(n) \cong \mathfrak{g}_7(n)$ and $\mathfrak{a}_{10}(n) = \mathfrak{g}_{10}(n)$. □

Now we consider the subalgebra $\mathfrak{a}_9(n)$. In this case, we have the involution $\theta_9(g) = -Q_9 g^T Q_9$, where Q_9 is given by Equation (G.45).

Lemma G.3.22. *We have* $\mathfrak{a}_9(n) = \mathfrak{g}_9(n)^{\theta_9}$.

Proof. The claim is true for $n = 2$ and 3 by comparing the dimensions. Suppose by induction that the statement is true for $\mathfrak{a}_9(n-1)$, and consider a Pauli string $a \in \mathfrak{g}_9(n)^{\theta_9}$ for $n \geq 4$. Again, we will omit the multiples of i that make Pauli strings skew-Hermitian.

If a ends with I, we can write $a = AI$ for some $A \in \mathfrak{g}_9(n-1)^{\theta_9}$ and apply the inductive assumption. Similarly, if $a = AIB$ has an I in the j-th position for some $j \geq 3$, we can delete it and get an element $AB \in \mathfrak{g}_9(n-1)^{\theta_9}$, which by induction is in $\mathfrak{a}_9(n-1)$. Then $a \in \mathfrak{a}_9(n)$, because $\mathfrak{a}_9(4)$ contains $IXIY$ and $IXIZ$, which generate elements of $\mathfrak{a}_9(n)$ with I in the middle.

Suppose that a has no I in positions $3, \ldots, n$. If $a = AXXB$ contains XX in positions $j, j+1$, then $[X_j Y_{j+1}, a] = -2iAIZB \in \mathfrak{a}_9(n)$. Since $X_j Y_{j+1} \in \mathfrak{a}_9(n)$, by Lemma G.3.16, we get that $a \in \mathfrak{a}_9(n)$. So, if $a \notin \mathfrak{a}_9(n)$ contains an X, then on the left of it must have a Y or Z. Then we can use $[XZ, XY] = -2iIX$, $[XY, YY] = 2iZI$, $[XY, ZY] = -2iYI$ when a contains a Y, and $[XY, XZ] = 2iIX$, $[XZ, YZ] = 2iZI$, $[XZ, ZZ] = -2iYI$ when a contains a Z. □

Finally, let us briefly discuss the remaining easier cases, $k = 11, 13, 15, 16, 20$. Recall from Equation (G.15), Equation (G.16) that $\mathfrak{a}_{11}(n) = \mathfrak{a}_{16}(n)$ for $n \geq 4$ and $\mathfrak{a}_{13}(n) = \mathfrak{a}_{20}(n)$ for $n \geq 3$. Moreover, $\mathfrak{g}_{11}(n) = \mathfrak{g}_{16}(n)$ and $\mathfrak{g}_{13}(n) = \mathfrak{g}_{20}(n)$, because they have equal stabilizers by Proposition G.3.3. Thus, we are left to consider only $k = 13, 15, 16$.

Lemma G.3.23. *We have* $\mathfrak{a}_{13}(n) = \mathfrak{g}_{13}(n) = \mathfrak{su}(2^n)^{P_X}/\mathrm{span}\{P_X\}$ *for* $n \geq 3$.

Proof. We know that $\mathrm{St}(\mathfrak{a}_{13}(n)) = \{P_I, P_X\}$ and $\mathfrak{a}_{13}(n) \subseteq \mathfrak{g}_{13}(n)$. The proof of the opposite inclusion is similar to the proof of Lemma G.3.22. Consider a Pauli string $a \in \mathfrak{g}_{13}(n)$ for $n \geq 4$. If $a = AIB$ has an I in the j-th position for some $1 \leq j \leq n$, we can delete it and get an element $AB \in \mathfrak{g}_{13}(n-1)$, which by induction is in $\mathfrak{a}_{13}(n-1)$. Then $a \in \mathfrak{a}_{13}(n)$, because $\mathfrak{a}_{13}(3)$ contains XIX, YIY, YIZ, and these generate elements of $\mathfrak{a}_{13}(n)$ with I in the middle. If a has no I's, we can use commutators with the generators of $\mathfrak{a}_{13}(n)$ to produce one. Then again we can apply Lemma G.3.16. □

Lemma G.3.24. *We have* $\mathfrak{a}_{16}(n) = \mathfrak{su}(2^n)^{\theta_{16}} = \mathfrak{so}(2^n)$, *where* $\theta_{16}(g) = -g^T$.

Proof. The same as the proof of Lemma G.3.23, using that $AIB \in \mathfrak{a}_{16}(3)$ for every generator AB of $\mathfrak{a}_{16}$. Indeed, one checks that $\mathfrak{a}_{16} = \langle XY, YX, YZ, ZY\rangle_{\mathrm{Lie}}$ and $XIY, YIX, YIZ, ZIY \in \mathfrak{a}_{16}(3)$; see Appendix G.2.5. □

Lemma G.3.25. *We have* $\mathfrak{a}_{15}(n) = \mathfrak{g}_{15}(n) = \mathfrak{su}(2^n)^{X_1}/\mathrm{span}\{X_1\}$.

Proof. Note that $\mathfrak{su}(2^n)^{X_1}$ is the span of all Pauli strings $\neq I^{\otimes n}$ that start with I or X. As in the proof of Lemma G.3.23, pick any Pauli string $a \in \mathfrak{g}_{15}(n)$ for $n \geq 3$. If $a = AIB$ has an I in the j-th position for some $2 \leq j \leq n$, we can delete it and get an element $AB \in \mathfrak{g}_{15}(n-1)$, which by induction is in $\mathfrak{a}_{15}(n-1)$. The rest of the proof is the same, using that $\mathfrak{a}_{15} = \langle XX, XY, XZ\rangle_{\mathrm{Lie}}$ and $XIX, XIY, XIZ \in \mathfrak{a}_{15}(3)$; see Appendix G.2.5. □

Combining the results of Appendix G.3.5 and G.3.6 completes the proof of Theorem G.3.1.

G.3.7 Identifying the Lie algebras $\mathfrak{g}_k(n)^{\theta_k}$

In this subsection, we finish the proof of Theorem 4.5.1, by identifying the Lie algebras $\mathfrak{g}_k(n)^{\theta_k}$ from Theorem G.3.1 with the Lie algebras appearing in the right-hand sides in Theorem 4.5.1. As in Theorem G.3.1, we only consider the cases $k = 3, 5, 6, 7, 9, 10, 11, 13, 15, 16, 20$. Moreover, due to the isomorphisms $\mathfrak{a}_6(n) \cong \mathfrak{a}_7(n) \cong \mathfrak{a}_{10}(n)$ and the equalities $\mathfrak{a}_{11}(n) =$

$\mathfrak{a}_{16}(n)$ and $\mathfrak{a}_{13}(n) = \mathfrak{a}_{20}(n)$ (see Lemmas G.3.2, G.3.4, G.3.5), we can omit the cases $k = 6, 10, 11, 20$.

The case $k = 16$ is obvious, because $\mathfrak{g}_{16}(n) = \mathfrak{su}(2^n)$ and $\theta_{16}(g) = -g^T$, leading to $\mathfrak{a}_{16}(n) = \mathfrak{so}(2^n)$. Two other easy cases, $k = 15$ and $k = 13$, are treated in the next lemma.

Lemma G.3.26. *We have:*

$$\mathfrak{a}_{15}(n) = \mathfrak{g}_{15}(n) = \mathfrak{su}(2^n)^{X_1}/\mathrm{span}\{X_1\} \cong \mathfrak{su}(2^{n-1}) \oplus \mathfrak{su}(2^{n-1}),$$
$$\mathfrak{a}_{13}(n) = \mathfrak{g}_{13}(n) = \mathfrak{su}(2^n)^{P_X}/\mathrm{span}\{P_X\} \cong \mathfrak{su}(2^{n-1}) \oplus \mathfrak{su}(2^{n-1}).$$

Proof. Note that

$$\mathfrak{su}(2^n)^{X_1}/\mathrm{span}\{X_1\} \cong \mathrm{span}_{\mathbb{R}}\{I, X\} \otimes \mathfrak{su}(2^{n-1})$$

has a basis consisting of all Pauli strings $\neq I^{\otimes n}, X_1$ that start with I or X. Consider the projections $P_\pm$ onto the eigenspaces of X, given by $P_\pm := (I \pm X)/2$. They satisfy the identities:

$$P_\pm \cdot P_\pm = P_\pm, \qquad P_+ \cdot P_- = 0, \qquad P_+ + P_- = I.$$

Then the map

$$(a, b) \mapsto P_+ \otimes a + P_- \otimes b$$

is a Lie algebra isomorphism from $\mathfrak{su}(2^{n-1}) \oplus \mathfrak{su}(2^{n-1})$ to $\mathrm{span}_{\mathbb{R}}\{I, X\} \otimes \mathfrak{su}(2^{n-1})$. This proves the claim about $\mathfrak{a}_{15}(n)$.

For the case $\mathfrak{a}_{13}(n)$, we can replace X_1 with P_X because there exists a unitary transformation U such that $P_X = UX_1U^\dagger$. For example, we can take

$$U = e^{-i\frac{\pi}{4}Y_1} e^{i\frac{\pi}{4}Y \otimes X^{\otimes(n-1)}};$$

then using Equation (A.6) we check that indeed $UX_1U^\dagger = X \otimes X^{\otimes(n-1)} = P_X$. The automorphism $a \mapsto UaU^\dagger$ of $\mathfrak{su}(2^n)$ sends $\mathfrak{su}(2^n)^{X_1}$ onto $\mathfrak{su}(2^n)^{P_X}$, and $\mathfrak{a}_{15}(n)$ onto $\mathfrak{a}_{13}(n)$. Therefore, $\mathfrak{a}_{13}(n) \cong \mathfrak{a}_{15}(n)$. □

We are left with the cases $k = 3, 5, 7, 9$, and we consider $k = 9$ next.

Lemma G.3.27. *We have* $\mathfrak{a}_9(n) = \mathfrak{g}_9(n)^{\theta_9} \cong \mathfrak{sp}(2^{n-2})$.

Proof. Recall that $\mathfrak{g}_9(n) = \mathfrak{su}(2^n)^{\{X_1, Y_1X_2, Z_1X_2\}}$. Since $\mathrm{span}\{X_1, Y_1X_2, Z_1X_2\} \cong \mathfrak{su}(2)$, we can find a unitary transformation that takes this Lie algebra to $\mathrm{span}\{X_1, Y_1, Z_1\}$. Explicitly,

similarly to the proof of Lemma G.3.26, let $U = e^{i\frac{\pi}{4}X_1X_2}$. Then using Equation (A.6), one easily checks that

$$UX_1U^\dagger = X_1, \qquad UY_1X_2U^\dagger = -Z_1, \qquad UZ_1X_2U^\dagger = Y_1.$$

Therefore, the map $a \mapsto UaU^\dagger$ restricts to a Lie algebra isomorphism from $\mathfrak{g}_9(n)$ to

$$\mathfrak{su}(2^n)^{\{X_1,Y_1,Z_1\}} = I \otimes \mathfrak{su}(2^{n-1}) \cong \mathfrak{su}(2^{n-1}).$$

According to Lemmas G.1.1 and G.1.2, under the transformation $a \mapsto UaU^\dagger$, the fixed-point subalgebra $\mathfrak{g}_9(n)^{\theta_9}$ is sent to the fixed points of the following involution:

$$a \mapsto -(UQ_9U^T)a^T(UQ_9U^T)^\dagger.$$

Recalling that $Q_9 = Y_2Z_3\cdots Z_n$ (see Equation (G.45)), we find from $U^T = U$ and $e^{i\frac{\pi}{4}X}Ye^{i\frac{\pi}{4}X} = Y$ that

$$UQ_9U^T = Q_9.$$

Hence, the image of $\mathfrak{g}_9(n)^{\theta_9}$ under $a \mapsto UaU^\dagger$ consists of all $b \in I \otimes \mathfrak{su}(2^{n-1})$ that are fixed by θ_9. Writing $b = I \otimes c$ with $c \in \mathfrak{su}(2^{n-1})$, the condition $b = \theta_9(b)$ is equivalent to $c = -Qc^TQ$, where $Q = YZ\cdots Z \in \mathcal{P}_{n-1}$. Since $Q^T = -Q$, this determines the Lie algebra $\mathfrak{sp}(2^{n-2})$, due to Corollary G.1.1. □

Next we consider the case $k = 7$.

Lemma G.3.28. *We have* $\mathfrak{a}_7(n) = \mathfrak{g}_7(n) \cong \begin{cases} \mathfrak{su}(2^{n-1}), & n \text{ odd}, \\ \mathfrak{su}(2^{n-2})^{\oplus 4}, & n \geq 4 \text{ even}. \end{cases}$

Proof. Recall that $\mathrm{St}(\mathfrak{a}_7(n)) = \{P_I, P_X, P_Y, P_Z\}$. Since $P_X \cdot P_Y = i^n P_Z$, elements that commute with P_X and P_Y will commute with P_Z as well. Hence, $\mathfrak{su}(2^n)^{\mathrm{St}(\mathfrak{a}_7(n))} = \mathfrak{su}(2^n)^{\{P_X,P_Y\}}$. Recall also that $[P_X, P_Y] = 0$ if and only if n is even; in that case, $\mathfrak{su}(2^n)^{\{P_X,P_Y\}}$ has a center spanned by P_X, P_Y, P_Z, and we need to quotient by it to obtain $\mathfrak{g}_7(n)$ (cf. Equation (G.37)).

In order to determine the fixed points under P_X and P_Y, we transform them as in the proof of Lemma G.3.26. Consider the unitary operator

$$U = \begin{cases} e^{i\frac{\pi}{4}Z\otimes Y^{\otimes(n-1)}}e^{i\frac{\pi}{4}Y\otimes X^{\otimes(n-1)}}, & n \text{ odd}, \\ e^{i\frac{\pi}{4}X_2}e^{i\frac{\pi}{4}I\otimes X\otimes Z^{\otimes(n-2)}}e^{i\frac{\pi}{4}Y\otimes X^{\otimes(n-1)}}, & n \text{ even}. \end{cases} \tag{G.54}$$

Using Equation (A.6), one checks that

$$UP_XU^\dagger = Z_1, \qquad UP_YU^\dagger = X_1 \qquad \text{for } n \text{ odd}, \tag{G.55}$$

$$UP_XU^\dagger = Z_1, \qquad UP_YU^\dagger = (-1)^{(n+2)/2} Z_2 \qquad \text{for } n \text{ even.} \tag{G.56}$$

Indeed, we have

$$e^{i\frac{\pi}{4}Y\otimes X^{\otimes(n-1)}} P_X e^{-i\frac{\pi}{4}Y\otimes X^{\otimes(n-1)}} = i(Y \otimes X^{\otimes(n-1)}) \cdot P_X = Z_1.$$

Since the other factors of U commute with Z_1, we obtain that $UP_XU^\dagger = Z_1$. The calculation of $UP_YU^\dagger$ is similar. When n is odd, $Y \otimes X^{\otimes(n-1)}$ commutes with P_Y, and we get from Equation (A.6):

$$UP_YU^\dagger = e^{i\frac{\pi}{4}Z\otimes Y^{\otimes(n-1)}} P_Y e^{-i\frac{\pi}{4}Z\otimes Y^{\otimes(n-1)}} = i(Z \otimes Y^{\otimes(n-1)}) \cdot P_Y = X_1.$$

When n is even, after applying Equation (A.6) three times, we obtain:

$$\begin{aligned} UP_YU^\dagger &= i^3 X_2 \cdot (I \otimes X \otimes Z^{\otimes(n-2)}) \cdot (Y \otimes X^{\otimes(n-1)}) \cdot P_Y \\ &= i^{n+2} X_2 \cdot (I \otimes X \otimes Z^{\otimes(n-2)}) \cdot (I \otimes Z^{\otimes(n-1)}) \\ &= i^{n+1} X_2 \cdot Y_2 = i^{n+2} Z_2. \end{aligned}$$

This proves Equation (G.55) and Equation (G.56).

It follows from Equation (G.55) that, for n odd, the map $a \mapsto UaU^\dagger$ gives a Lie algebra isomorphism

$$\mathfrak{g}_7(n) = \mathfrak{su}(2^n)^{\{P_X,P_Y\}} \to \mathfrak{su}(2^n)^{\{X_1,Z_1\}} = I \otimes \mathfrak{su}(2^{n-1}) \cong \mathfrak{su}(2^{n-1}).$$

Now suppose that n is even. Then, by Equation (G.56), the map $a \mapsto UaU^\dagger$ gives an isomorphism

$$\mathfrak{su}(2^n)^{\{P_X,P_Y\}} \to \mathfrak{su}(2^n)^{\{Z_1,Z_2\}} = \big(\mathrm{span}_{\mathbb{R}}\{I,Z\} \otimes \mathrm{span}_{\mathbb{R}}\{I,Z\} \otimes \mathfrak{su}(2^{n-2})\big) \oplus \mathrm{span}\{Z_1, Z_2, Z_1Z_2\}.$$

After we quotient by the center $\mathrm{span}\{Z_1, Z_2, Z_1Z_2\}$, we obtain

$$\mathfrak{g}_7(n) = \mathfrak{su}(2^n)^{\{P_X,P_Y\}} / \mathrm{span}\{P_X, P_Y, P_Z\} \cong \mathrm{span}_{\mathbb{R}}\{I,Z\} \otimes \mathrm{span}_{\mathbb{R}}\{I,Z\} \otimes \mathfrak{su}(2^{n-2}).$$

Again as in the proof of Lemma G.3.26, let $P_\pm := (I \pm Z)/2$, and consider the four projections

$$P_1 := P_+ \otimes P_+, \qquad P_2 := P_+ \otimes P_-, \qquad P_3 := P_- \otimes P_+, \qquad P_4 := P_- \otimes P_-,$$

which satisfy

$$P_i \cdot P_i = P_i, \qquad P_i \cdot P_j = 0 \quad (i \neq j), \qquad \sum_{i=1}^{4} P_i = I \otimes I.$$

Then the linear map

$$(a_1, a_2, a_3, a_4) \mapsto \sum_{j=1}^{4} P_j \otimes a_j \tag{G.57}$$

is an isomorphism from $\mathfrak{su}(2^{n-2})^{\oplus 4}$ to $\mathrm{span}_{\mathbb{R}}\{I, Z\} \otimes \mathrm{span}_{\mathbb{R}}\{I, Z\} \otimes \mathfrak{su}(2^{n-2})$. □

In the remaining two cases $k = 3, 5$, as before we embed $\mathfrak{a}_3(n)$ and $\mathfrak{a}_5(n)$ as subalgebras of $\mathfrak{a}_7(n)$. We continue to use the notation from Appendix G.3.6 and, as in Lemma G.3.28, we consider separately the cases when n is odd or even.

Lemma G.3.29. *We have* $\mathfrak{a}_3(n) \cong \tilde{\mathfrak{a}}_3(n) = \mathfrak{g}_7(n)^{\tilde{\theta}_3} \cong \begin{cases} \mathfrak{so}(2^{n-1}), & n \equiv \pm 1 \bmod 8, \\ \mathfrak{sp}(2^{n-2}), & n \equiv \pm 3 \bmod 8. \end{cases}$

Proof. We apply the transformation $a \mapsto UaU^\dagger$ from the proof of Lemma G.3.28 that gives a Lie algebra isomorphism $\mathfrak{g}_7(n) \to I \otimes \mathfrak{su}(2^{n-1}) \cong \mathfrak{su}(2^{n-1})$, where U is defined by Equation (G.54) for odd n. Then, by Lemmas G.1.1, G.1.2, the fixed points of $\tilde{\theta}_k$ (see Equation (G.47)) are sent to the fixed points of the involution

$$g \mapsto -(UQ_kU^T)g^T(UQ_kU^T)^\dagger, \qquad k = 3, 5. \tag{G.58}$$

Recall that $Q_3 = P_{ZIYX}$ is given by Equation (G.48), and compute

$$\tilde{Q}_3 := UQ_3U^T = e^{i\frac{\pi}{4}Z\otimes Y^{\otimes(n-1)}} e^{i\frac{\pi}{4}Y\otimes X^{\otimes(n-1)}} P_{ZIYX} e^{-i\frac{\pi}{4}Y\otimes X^{\otimes(n-1)}} e^{i\frac{\pi}{4}Z\otimes Y^{\otimes(n-1)}}.$$

Note that, when $n \equiv 1 \bmod 4$, P_{ZIYX} anticommutes with $Y \otimes X^{\otimes(n-1)}$. By Equation (A.6), we have:

$$\begin{aligned} e^{i\frac{\pi}{4}Y\otimes X^{\otimes(n-1)}} & P_{ZIYX} e^{-i\frac{\pi}{4}Y\otimes X^{\otimes(n-1)}} \\ &= i(Y_1X_2X_3X_4\cdots X_{n-1}X_n)\cdot(Z_1Y_3X_4Z_5Y_7X_8\cdots X_{n-1}Z_n) \\ &= -X_1(X_2Z_3Y_5)(X_6Z_7Y_9)\cdots(X_{n-3}Z_{n-2}Y_n). \end{aligned}$$

As this anticommutes with $e^{i\frac{\pi}{4}Z\otimes Y^{\otimes(n-1)}}$, we obtain

$$\tilde{Q}_3 = -X_1(X_2Z_3Y_5)(X_6Z_7Y_9)\cdots(X_{n-3}Z_{n-2}Y_n), \qquad n \equiv 1 \bmod 4.$$

Hence, restricted to elements $g = I \otimes c$ with $c \in \mathfrak{su}(2^{n-1})$, the involution Equation (G.58) becomes:

$$c \mapsto -P_{XZIY}c^TP_{XZIY}, \qquad \text{for} \quad n \equiv 1 \bmod 4.$$

For the fixed-point subalgebra, we obtain from Corollary G.1.1:

$$(P_{XZIY})^T = \begin{cases} P_{XZIY}, & n \equiv 1 \bmod 8, \\ -P_{XZIY}, & n \equiv 5 \bmod 8 \end{cases} \quad \Rightarrow \quad \tilde{\mathfrak{a}}_3(n) \cong \begin{cases} \mathfrak{so}(2^{n-1}), & n \equiv 1 \bmod 8, \\ \mathfrak{sp}(2^{n-2}), & n \equiv 5 \bmod 8. \end{cases}$$

Alternatively, when $n \equiv 3 \bmod 4$, P_{ZIYX} commutes with both $Y \otimes X^{\otimes(n-1)}$ and $Z \otimes Y^{\otimes(n-1)}$. Hence, in this case,

$$\begin{aligned} \tilde{Q}_3 &= e^{i\frac{\pi}{4}Z\otimes Y^{\otimes(n-1)}} P_{ZIYX} e^{i\frac{\pi}{4}Z\otimes Y^{\otimes(n-1)}} \\ &= i(Z_1Y_2Y_3Y_4\cdots Y_{n-1}Y_n)\cdot(Z_1Y_3X_4Z_5Y_7X_8\cdots X_{n-3}Z_{n-2}Y_n) \\ &= iY_2(Z_4X_5Y_6)(Z_8X_9Y_{10})\cdots(Z_{n-3}X_{n-2}Y_{n-1}). \end{aligned}$$

Thus, restricted to elements $g = I \otimes c$ with $c \in \mathfrak{su}(2^{n-1})$, the involution Equation (G.58) simplifies to:

$$c \mapsto -P_{YIZX} c^T P_{YIZX}, \qquad \text{for} \quad n \equiv 3 \bmod 4.$$

Corollary G.1.1 gives for the fixed-point subalgebra:

$$(P_{YIZX})^T = \begin{cases} P_{YIZX}, & n \equiv 7 \bmod 8, \\ -P_{YIZX}, & n \equiv 3 \bmod 8 \end{cases} \quad \Rightarrow \quad \tilde{\mathfrak{a}}_3(n) \cong \begin{cases} \mathfrak{so}(2^{n-1}), & n \equiv 7 \bmod 8, \\ \mathfrak{sp}(2^{n-2}), & n \equiv 3 \bmod 8. \end{cases}$$

This completes the proof of the lemma. □

Lemma G.3.30. *We have* $\mathfrak{a}_5(n) \cong \tilde{\mathfrak{a}}_5(n) = \mathfrak{g}_7(n)^{\tilde{\theta}_5} \cong \begin{cases} \mathfrak{so}(2^{n-1}), & n \equiv \pm 1 \bmod 6, \\ \mathfrak{sp}(2^{n-2}), & n \equiv 3 \bmod 6. \end{cases}$

Proof. The proof is analogous to that of Lemma G.3.29. Recall that U and $Q_5 = P_{IYZ}$ are given by Equation (G.54), Equation (G.49), and compute

$$\tilde{Q}_5 := UQ_5U^T = e^{i\frac{\pi}{4}Z\otimes Y^{\otimes(n-1)}} e^{i\frac{\pi}{4}Y\otimes X^{\otimes(n-1)}} P_{IYZ} e^{-i\frac{\pi}{4}Y\otimes X^{\otimes(n-1)}} e^{i\frac{\pi}{4}Z\otimes Y^{\otimes(n-1)}}.$$

When $n \equiv 3 \bmod 6$, $P_{IYZ} = Y_2Z_3Y_5Z_6Y_8Z_9\cdots Y_{n-1}Z_n$ commutes with $Y \otimes X^{\otimes(n-1)}$ and anticommutes with $Z \otimes Y^{\otimes(n-1)}$. Hence, by Equation (A.5), $\tilde{Q}_5 = P_{IYZ}$. Restricted to elements $g = I \otimes c$ with $c \in \mathfrak{su}(2^{n-1})$, the involution Equation (G.58) simplifies to:

$$c \mapsto -P_{YZI} c^T P_{YZI}, \qquad \text{for} \quad n \equiv 3 \bmod 6.$$

Since $(P_{YZI})^T = -P_{YZI}$, the fixed-point subalgebra is isomorphic to $\mathfrak{sp}(2^{n-2})$, by Corollary G.1.1.

For $n \equiv 1 \bmod 6$, $P_{IYZ} = Y_2 Z_3 Y_5 Z_6 \cdots Y_{n-2} Z_{n-1}$ commutes with both $Y \otimes X^{\otimes(n-1)}$ and $Z \otimes Y^{\otimes(n-1)}$. Hence, in this case,

$$\begin{aligned}\tilde{Q}_5 &= i(Z_1 Y_2 Y_3 Y_4 \cdots Y_{n-1} Y_n) \cdot (Y_2 Z_3 Y_5 Z_6 \cdots Y_{n-2} Z_{n-1}) \\ &= i^{(n+2)/3} Z_1 (X_3 Y_4)(X_6 Y_7) \cdots (X_{n-1} Y_n).\end{aligned}$$

The involution induced by Equation (G.58) on $c \in \mathfrak{su}(2^{n-1})$ is given by

$$c \mapsto -P_{IXY} c^T P_{IXY}, \qquad \text{for} \quad n \equiv 1 \bmod 6,$$

and the fixed-point subalgebra is isomorphic to $\mathfrak{so}(2^{n-1})$, because $(P_{IXY})^T = P_{IXY}$.

Finally, for $n \equiv 5 \bmod 6$, using Equation (A.6), we find

$$\begin{aligned}e^{i\frac{\pi}{4} Y \otimes X^{\otimes(n-1)}} & P_{IYZ} e^{-i\frac{\pi}{4} Y \otimes X^{\otimes(n-1)}} \\ &= i(Y_1 X_2 X_3 X_4 X_5 \cdots X_n)(Y_2 Z_3 Y_5 \cdots Z_{n-2} Y_n) \\ &= -Y_1 Z_2 Y_3 X_4 Z_5 \cdots Y_{n-2} X_{n-1} Z_n.\end{aligned}$$

Then applying Equation (A.5), we get

$$\begin{aligned}\tilde{Q}_5 &= -i(Z_1 Y_2 Y_3 Y_4 Y_5 \cdots Y_n) \cdot (Y_1 Z_2 Y_3 X_4 Z_5 \cdots Y_{n-2} X_{n-1} Z_n) \\ &= -i X_1 X_2 (Z_4 X_5) \cdots (Z_{n-1} X_n).\end{aligned}$$

This induces the involution on $\mathfrak{su}(2^{n-1})$ given by

$$c \mapsto -P_{XIZ} c^T P_{XIZ}, \qquad \text{for} \quad n \equiv 5 \bmod 6,$$

and the fixed-point subalgebra is isomorphic again to $\mathfrak{so}(2^{n-1})$. □

Lemma G.3.31. *We have* $\mathfrak{a}_3(n) \cong \tilde{\mathfrak{a}}_3(n) = \mathfrak{g}_7(n)^{\tilde{\theta}_3} \cong \begin{cases} \mathfrak{so}(2^{n-2})^{\oplus 4}, & n \equiv 0 \bmod 8, \\ \mathfrak{su}(2^{n-2})^{\oplus 2}, & n \equiv \pm 2 \bmod 8, \\ \mathfrak{sp}(2^{n-3})^{\oplus 4}, & n \equiv 4 \bmod 8. \end{cases}$

Proof. As in the proof of Lemma G.3.29, we need to compute

$$\tilde{Q}_3 := U Q_3 U^T = e^{i\frac{\pi}{4} X_2} e^{i\frac{\pi}{4} I \otimes X \otimes Z^{\otimes(n-2)}} e^{i\frac{\pi}{4} Y \otimes X^{\otimes(n-1)}} P_{ZIYX} e^{-i\frac{\pi}{4} Y \otimes X^{\otimes(n-1)}} e^{i\frac{\pi}{4} I \otimes X \otimes Z^{\otimes(n-2)}} e^{i\frac{\pi}{4} X_2}.$$

Using Equation (A.5), Equation (A.6), we find for $n \equiv 0 \bmod 4$:

$$\tilde{Q}_3 = -Z_1 (X_3 Y_4 Z_6)(X_7 Y_8 Z_{10}) \cdots X_{n-1} Y_n.$$

Via the isomorphism $\mathfrak{g}_7(n) \cong \mathfrak{su}(2^{n-2})^{\oplus 4}$ from the proof of Lemma G.3.28 (see Equation (G.57)), the involution induced from Equation (G.58) on each copy of $\mathfrak{su}(2^{n-2})$ is given by

$$a_j \mapsto -P_{XYIZ} a_j^T P_{XYIZ}, \qquad \text{for} \quad 1 \leq j \leq 4, \quad n \equiv 0 \bmod 4.$$

For the fixed-point subalgebra, we get from Corollary G.1.1:

$$(P_{XYIZ})^T = \begin{cases} P_{XYIZ}, & n \equiv 0 \bmod 8, \\ -P_{XYIZ}, & n \equiv 4 \bmod 8 \end{cases} \quad \Rightarrow \quad \tilde{\mathfrak{a}}_3(n) \cong \begin{cases} \mathfrak{so}(2^{n-2})^{\oplus 4}, & n \equiv 0 \bmod 8, \\ \mathfrak{sp}(2^{n-3})^{\oplus 4}, & n \equiv 4 \bmod 8. \end{cases}$$

Using Equation (A.5), Equation (A.6), we find for $n \equiv 0 \bmod 4$:

$$\tilde{Q}_3 = X_1 X_2 (Z_4 X_5 Y_6)(Z_8 X_9 Y_{10}) \cdots (Z_{n-2} X_{n-1} Y_n).$$

Consider the unitary operator

$$V = \begin{cases} e^{i\frac{\pi}{4} Z_4 X_5 Y_6 \cdots Z_{n-2} X_{n-1} Y_n}, & n \equiv 2 \bmod 8, \\ e^{i\frac{\pi}{4} Z_2 Z_4 X_5 Y_6 \cdots Z_{n-2} X_{n-1} Y_n}, & n \equiv 6 \bmod 8, \end{cases}$$

and perform the transformation $a \mapsto V a V^\dagger$ on $\mathfrak{g}_7(n)$. Since V commutes with Z_1 and Z_2, this transformation preserves the decomposition $\mathfrak{g}_7(n) \cong \mathfrak{su}(2^{n-2})^{\oplus 4}$ given by Equation (G.57). For $n \equiv 2 \bmod 8$, we have $V^T = V$ and V commutes with $\tilde{Q}_3$. Thus, $\tilde{Q}_3$ gets transformed to

$$V \tilde{Q}_3 V^T = V^2 \tilde{Q}_3 = i(Z_4 X_5 Y_6 \cdots Z_{n-2} X_{n-1} Y_n) \cdot \tilde{Q}_3 = i X_1 X_2.$$

The involution induced on $\mathfrak{g}_7(n)$ is given by

$$a \mapsto -X_1 X_2 a^T X_1 X_2.$$

Writing a as in Equation (G.57), we note that

$$X_1 X_2 \cdot P_1 \cdot X_1 X_2 = P_4, \qquad X_1 X_2 \cdot P_2 \cdot X_1 X_2 = P_3.$$

Hence, $a = (a_1, a_2, a_3, a_4)$ is a fixed point of the above involution if and only if $a_4 = -a_1^T$, $a_3 = -a_2^T$. Sending such a to (a_1, a_2) gives an isomorphism from the fixed-point subalgebra to $\mathfrak{su}(2^{n-2})^{\oplus 2}$.

When $n \equiv 6 \bmod 8$, we have $V^T = V^{-1}$ and V anticommutes with $\tilde{Q}_3$. Thus, $\tilde{Q}_3$ gets transformed to

$$V \tilde{Q}_3 V^T = V \tilde{Q}_3 V^{-1} = V^2 \tilde{Q}_3 = i(Z_2 Z_4 X_5 Y_6 \cdots Z_{n-2} X_{n-1} Y_n) \cdot \tilde{Q}_3 = -X_1 Y_2.$$

The rest of the proof is similar to the case $n \equiv 2 \bmod 8$ above. □

Lemma G.3.32. *We have* $\mathfrak{a}_5(n) \cong \tilde{\mathfrak{a}}_5(n) = \mathfrak{g}_7(n)^{\tilde{\theta}_5} \cong \begin{cases} \mathfrak{so}(2^{n-2})^{\oplus 4}, & n \equiv 0 \bmod 6, \\ \mathfrak{su}(2^{n-2})^{\oplus 2}, & n \equiv \pm 2 \bmod 6. \end{cases}$

Proof. The proof is very similar to Lemma G.3.31, so we only indicate the differences. We compute

$$\tilde{Q}_5 := UQ_5U^T = e^{i\frac{\pi}{4}X_2} e^{i\frac{\pi}{4}I\otimes X\otimes Z^{\otimes(n-2)}} e^{i\frac{\pi}{4}Y\otimes X^{\otimes(n-1)}} P_{IYZ} e^{-i\frac{\pi}{4}Y\otimes X^{\otimes(n-1)}} e^{i\frac{\pi}{4}I\otimes X\otimes Z^{\otimes(n-2)}} e^{i\frac{\pi}{4}X_2},$$

and find that

$$\tilde{Q}_5 = \begin{cases} i^{-n/3} Z_2(Z_4X_5)(Z_7X_8)\cdots X_{n-1}, & n \equiv 0 \bmod 6, \\ -Y_1Z_2(Y_3X_4Z_5)(Y_6X_7Z_8)\cdots Z_n, & n \equiv 2 \bmod 6, \\ P_{IYZ} = (Y_2Z_3)(Y_5Z_6)(Y_8Z_9)\cdots Z_{n-1}, & n \equiv 4 \bmod 6. \end{cases}$$

For $n \equiv 0 \bmod 6$, the involution induced from Equation (G.58) on each copy of $\mathfrak{su}(2^{n-2})$ from the decomposition Equation (G.57) is given by

$$a_j \mapsto -P_{IZX} a_j^T P_{IZX}, \qquad \text{for} \quad 1 \le j \le 4, \quad n \equiv 0 \bmod 6.$$

For $n \equiv \pm 2 \bmod 6$, we use the transformation $a \mapsto VaV^\dagger$, where

$$V = \begin{cases} e^{i\frac{\pi}{4}Z_2Y_3X_4Z_5Y_6X_7Z_8\cdots Z_n}, & n \equiv 2 \bmod 6, \\ e^{i\frac{\pi}{4}Z_3Y_5Z_6Y_8Z_9\cdots Z_{n-1}}, & n \equiv 4 \bmod 6, \end{cases}$$

which allows us to replace $\tilde{Q}_5$ with $V\tilde{Q}_5V^T$. This gives the involutions $a \mapsto -Y_1a^TY_1$ and $a \mapsto -Y_2a^TY_2$ for $n \equiv 2$ and $n \equiv 4 \bmod 6$, respectively. □

G.3.8 Periodic boundary conditions

Recall that the subalgebras $\mathfrak{a}_k^\circ(n), \mathfrak{b}_l^\circ(n) \subseteq \mathfrak{su}(2^n)$ ($0 \le k \le 22$, $0 \le l \le 4$) are defined by Equation (G.6). In this subsection, we prove Theorem 4.5.2, which we reproduce here for convenience:

$$\begin{aligned} \mathfrak{a}_0^\circ(n) &\cong \mathfrak{u}(1)^{\oplus n}, \\ \mathfrak{a}_1^\circ(n) &\cong \mathfrak{so}(n)^{\oplus 2}, \\ \mathfrak{a}_2^\circ(n) &\cong \mathfrak{so}(n)^{\oplus 4}, \end{aligned}$$

$$\mathfrak{a}_3^\circ(n) = \begin{cases} \mathfrak{a}_{13}(n), & n \text{ odd}, \\ \mathfrak{a}_3(n), & n \equiv 0 \mod 4, \\ \mathfrak{a}_6(n), & n \equiv 2 \mod 4, \end{cases}$$

$$\mathfrak{a}_4^\circ(n) \cong \begin{cases} \mathfrak{so}(2n), & n \text{ odd}, \\ \mathfrak{so}(n)^{\oplus 4}, & n \text{ even}, \end{cases}$$

$$\mathfrak{a}_5^\circ(n) = \begin{cases} \mathfrak{a}_{16}(n), & n \equiv \pm 1 \mod 3, \\ \mathfrak{a}_5(n), & n \equiv 0 \mod 3, \end{cases}$$

$$\mathfrak{a}_6^\circ(n) = \begin{cases} \mathfrak{a}_{13}(n), & n \text{ odd}, \\ \mathfrak{a}_6(n), & n \text{ even}, \end{cases}$$

$$\mathfrak{a}_k^\circ(n) = \mathfrak{a}_k(n), \quad k = 7, 13, 16, 20,$$

$$\mathfrak{a}_8^\circ(n) \cong \mathfrak{so}(2n)^{\oplus 2},$$

$$\mathfrak{a}_9^\circ(n) = \mathfrak{b}_2^\circ(n) \cong \mathfrak{so}(2^n), \quad n \geq 4,$$

$$\mathfrak{a}_{10}^\circ(n) = \begin{cases} \mathfrak{su}(2^n), & n \equiv \pm 1 \mod 3, \\ \mathfrak{a}_{10}(n), & n \equiv 0 \mod 3, \end{cases}$$

$$\mathfrak{a}_{11}^\circ(n) = \mathfrak{so}(2^n), \quad n \geq 4,$$

$$\mathfrak{a}_k^\circ(n) = \mathfrak{b}_4^\circ(n) = \mathfrak{su}(2^n), \quad k = 12, 15, 17, 18, 19, 21, 22,$$

$$\mathfrak{a}_{14}^\circ(n) \cong \mathfrak{so}(2n)^{\oplus 2},$$

$$\mathfrak{b}_0^\circ(n) = \mathfrak{b}_0(n) \cong \mathfrak{u}(1)^{\oplus n},$$

$$\mathfrak{b}_1^\circ(n) \cong \mathfrak{u}(1)^{\oplus 2n},$$

$$\mathfrak{b}_3^\circ(n) = \mathfrak{b}_3(n) \cong \mathfrak{su}(2)^{\oplus n}.$$

We start the **proof** by observing that due to Equation (G.4), Equation (G.5) and from $\dim \mathfrak{a}_{12}^\circ(3) = \dim \mathfrak{a}_{17}^\circ(3) = 63$, we have:

$$\mathfrak{a}_k^\circ(n) = \mathfrak{su}(2^n), \qquad k = 12, 17, 18, 19, 21, 22, \quad n \geq 3.$$

Moreover,

$$\mathfrak{a}_{15}^\circ(n) = \mathfrak{su}(2^n), \qquad n \geq 3,$$

because $\mathfrak{a}_{15}(n)$ contains (i times) all Pauli strings that start with X or I, except $I^{\otimes n}$. Then, applying the cyclic shift τ_n defined in Equation (G.8), we can generate all Pauli strings $\neq I^{\otimes n}$.

We also note that

$$\mathfrak{a}_k^\circ(n) = \mathfrak{a}_k(n), \qquad k = 7, 13, 16, 20, \quad n \geq 3; \qquad \mathfrak{a}_{11}^\circ(n) = \mathfrak{a}_{11}(n), \qquad n \geq 4,$$

due to Equation (G.15), Equation (G.16) and Lemmas G.3.20, G.3.23, G.3.24, because in this case $\tau_n \mathfrak{a}_k(n) \subseteq \mathfrak{a}_k(n)$.

In Appendix G.3.3, using frustration graphs, we determined the Lie algebras $\mathfrak{a}_k^\circ(n)$ for $k = 1, 2, 4, 8, 14$ (see Lemmas G.3.7, G.3.8, G.3.9). It is also obvious that

$$\begin{aligned}
&\mathfrak{b}_0^\circ(n) = \mathfrak{b}_0(n), && \mathfrak{b}_2^\circ(n) = \mathfrak{a}_9^\circ(n),\\
&\mathfrak{b}_3^\circ(n) = \mathfrak{b}_3(n), && \mathfrak{b}_4^\circ(n) = \mathfrak{a}_{15}^\circ(n),\\
&\mathfrak{b}_1^\circ(n) = \operatorname{span}\{X_i, X_1X_n, X_jX_{j+1}\}_{1\le i\le n,\ 1\le j\le n-1} \cong \mathfrak{u}(1)^{\oplus 2n},\\
&\mathfrak{a}_0^\circ(n) = \operatorname{span}\{X_1X_n, X_jX_{j+1}\}_{1\le j\le n-1} \cong \mathfrak{u}(1)^{\oplus n}.
\end{aligned}$$

We discuss the remaining cases $\mathfrak{a}_k^\circ(n)$ $(k = 3, 5, 6, 9, 10)$ in a sequence of lemmas.

Lemma G.3.33. *We have* $\mathfrak{a}_{10}^\circ(n) = \begin{cases} \mathfrak{su}(2^n), & n \equiv \pm 1 \mod 3,\\ \mathfrak{a}_{10}(n), & n \equiv 0 \mod 3. \end{cases}$

Proof. Recall from Theorem G.3.1 that $\mathfrak{a}_{10}(n) = \mathfrak{g}_{10}(n)$ where $\mathfrak{g}_{10}(n)$ is given by Equation (G.36). When $n \equiv 0 \mod 3$, we have:

$$\tau_n P_{XYZ} = P_{YZX}, \qquad \tau_n P_{YZX} = P_{ZXY}, \qquad \tau_n P_{ZXY} = P_{XYZ},$$

which imply that $\tau_n \mathfrak{a}_{10}(n) \subseteq \mathfrak{a}_{10}(n)$, and hence $\mathfrak{a}_{10}^\circ(n) = \mathfrak{a}_{10}(n)$.

On the other hand, for $n \equiv 1 \mod 3$, we have:

$$\begin{aligned}
\tau_n^{-1} P_{XYZ} &= X \otimes P_{XYZ} = XXYZXYZ \cdots XYZ,\\
\tau_n^{-1} P_{YZX} &= Y \otimes P_{YZX} = YYZXYZX \cdots YZX,\\
\tau_n^{-1} P_{ZXY} &= Z \otimes P_{ZXY} = ZZXYZXY \cdots ZXY.
\end{aligned}$$

In particular, their centralizer contains the elements

$$X_1X_2, Y_1Y_2, Z_1Z_2 \in \tau_n^{-1}\mathfrak{a}_{10}(n) \subset \mathfrak{a}_{10}^\circ(n).$$

From these elements and

$$X_1Y_2, Y_1Z_2, Z_1X_2 \in \mathfrak{a}_{10}(n) \subset \mathfrak{a}_{10}^\circ(n),$$

we can generate all 2-qubit gates: $\mathfrak{su}(4) \otimes I^{\otimes(n-2)} \subset \mathfrak{a}_{10}^\circ(n)$. Therefore, $\mathfrak{a}_{10}^\circ(n) = \mathfrak{su}(2^n)$.

The case $n \equiv -1 \mod 3$ is similar. □

Lemma G.3.34. *We have* $\mathfrak{a}_5^\circ(n) = \begin{cases} \mathfrak{a}_{16}(n), & n \equiv \pm 1 \mod 3, \\ \mathfrak{a}_5(n), & n \equiv 0 \mod 3. \end{cases}$

Proof. Recall the automorphism γ_n of $\mathfrak{su}(2^n)$ defined by Equation (G.21), Equation (G.22). Then, by Lemma G.3.20, $\gamma_n\mathfrak{a}_5(n) = \mathfrak{g}_7(n)^{\tilde{\theta}_5}$, where $\tilde{\theta}_5$ is given by Equation (G.47), Equation (G.49), and $\mathfrak{g}_7(n)$ is given by Equation (G.37). From this, we get

$$\gamma_n\tau_n\mathfrak{a}_5(n) = (\gamma_n\tau_n\gamma_n^{-1})\gamma_n\mathfrak{a}_5(n) = (\gamma_n\tau_n\gamma_n^{-1})\mathfrak{g}_7(n)^{\tilde{\theta}_5}.$$

When $n \equiv 0 \mod 3$, we have

$$(\gamma_n\tau_n\gamma_n^{-1})(P_X) = P_Y, \qquad (\gamma_n\tau_n\gamma_n^{-1})(P_Y) = P_Z, \qquad (\gamma_n\tau_n\gamma_n^{-1})(P_Z) = P_X,$$

which imply that $(\gamma_n\tau_n\gamma_n^{-1})\mathfrak{g}_7(n) \subseteq \mathfrak{g}_7(n)$. Next, we compute (cf. Equation (G.49)):

$$\begin{aligned}(\gamma_n\tau_n\gamma_n^{-1})Q_5 &= (\gamma_n\tau_n\gamma_n^{-1})(P_{IYZ}) \\ &= Z_1X_2Z_4X_5\cdots Z_{n-2}X_{n-1} \\ &= i^{n/3}P_Z\cdot Q_5 = i^{-n/3}Q_5\cdot P_Z.\end{aligned}$$

From this, we deduce that $\tilde{\theta}_5$ commutes with $\gamma_n\tau_n\gamma_n^{-1}$. Indeed, as it commutes with the trace, we find for $g \in \mathfrak{g}_7(n)$:

$$(\gamma_n\tau_n\gamma_n^{-1})\tilde{\theta}_5(g) = -(\gamma_n\tau_n\gamma_n^{-1}Q_5)h^T(\gamma_n\tau_n\gamma_n^{-1}Q_5) = -Q_5\cdot P_Zh^TP_Z\cdot Q_5 = -Qh^TQ = \tilde{\theta}_5(h),$$

where we set $h := (\gamma_n\tau_n\gamma_n^{-1})g$ and use that $h, h^T \in \mathfrak{g}_7(n)$. Therefore, $\gamma_n\tau_n\mathfrak{a}_5(n) \subseteq \gamma_n\mathfrak{a}_5(n)$, and hence $\mathfrak{a}_5^\circ(n) = \mathfrak{a}_5(n)$ for $n \equiv 0 \mod 3$.

Suppose now that $n \equiv 1 \mod 3$. Observe that $\mathfrak{a}_5^\circ(n) \subseteq \mathfrak{so}(2^n) = \mathfrak{a}_{16}(n)$ for all $n \geq 2$, because all generators of $\mathfrak{a}_5^\circ(n)$ have an odd number of Y's. On the other hand, we have

$$X_2X_n \in \mathfrak{g}_7(n)^{\tilde{\theta}_5} = \gamma_n\mathfrak{a}_5(n) \;\Rightarrow\; \gamma_n^{-1}(X_2X_n) = Z_2Y_n \in \mathfrak{a}_5(n) \;\Rightarrow\; \tau_n^{-1}(Z_2Y_n) = Y_1Z_3 \in \mathfrak{a}_5^\circ(n).$$

Since $Y_1X_3 \in \mathfrak{a}_5(3)$ (see Appendix G.2.5), we get that $Y_1X_3 \in \mathfrak{a}_5^\circ(n)$. Hence, $[Y_1Z_3, Y_1X_3] = 2iY_3 \in \mathfrak{a}_5^\circ(n)$, and cyclic shifts give $Y_1, Y_2 \in \mathfrak{a}_5^\circ(n)$. Together with $\mathfrak{a}_5 = \langle XY, YZ\rangle_{\text{Lie}}$, the elements YI, IY can generate $\mathfrak{a}_{16} = \langle XY, YX, YZ, ZY\rangle_{\text{Lie}}$. Therefore, $\mathfrak{a}_5^\circ(n) \supseteq \mathfrak{a}_{16}(n)$, which proves that $\mathfrak{a}_5^\circ(n) = \mathfrak{a}_{16}(n)$.

Similarly, in the case $n \equiv -1 \mod 3$, we have:

$$X_1X_n \in \mathfrak{g}_7(n)^{\tilde{\theta}_5} = \gamma_n\mathfrak{a}_5(n) \;\Rightarrow\; \gamma_n^{-1}(X_1X_n) = Y_1Z_n \in \mathfrak{a}_5(n) \;\Rightarrow\; \tau_n^{-1}(Y_1Z_n) = Z_1Y_2 \in \mathfrak{a}_5^\circ(n),$$
$$Z_1Z_n \in \mathfrak{g}_7(n)^{\tilde{\theta}_5} = \gamma_n\mathfrak{a}_5(n) \;\Rightarrow\; \gamma_n^{-1}(Z_1Z_n) = X_1Y_n \in \mathfrak{a}_5(n) \;\Rightarrow\; \tau_n^{-1}(X_1Y_n) = Y_1X_2 \in \mathfrak{a}_5^\circ(n).$$

Hence, $\mathfrak{a}_5^\circ(n)$ contains $\mathfrak{a}_{16}(n)$, so it must be equal to it. □

Lemma G.3.35. *We have* $\mathfrak{a}_9^\circ(n) \cong \mathfrak{so}(2^n)$ *for* $n \geq 4$.

Proof. First, recall from Lemma G.3.22 that $\mathfrak{a}_9(n) = \mathfrak{g}_9(n)^{\theta_9}$, where $\mathfrak{g}_9(n)$ is given by Equation (G.38), $\theta_9(g) = -Q_9 g^T Q_9$, and $Q_9 = IYZ\cdots Z$ is given by Equation (G.45). For example, $g = Y_3X_4 \in \mathfrak{a}_9(n)$, as $g^T = -g$ and g commutes with X_1, Y_1X_2, Z_1X_2 and Q_9 (or one can check directly that $IIYX \in \mathfrak{a}_9(4)$). Similarly, we check that $Z_3X_4 \in \mathfrak{a}_9(n)$.

Now let us relabel $X \rightleftharpoons Y$, so that $\mathfrak{a}_9 = \langle YX, YZ\rangle_{\text{Lie}}$. Then $\mathfrak{a}_9^\circ(n) \subseteq \mathfrak{so}(2^n) = \mathfrak{a}_{16}(n)$, because all generators of $\mathfrak{a}_9^\circ(n)$ contain an odd number of Y's. From above after relabeling, we have $X_3Y_4, Z_3Y_4 \in \mathfrak{a}_9(n)$, which after a cyclic shift gives $X_1Y_2, Z_1Y_2 \in \mathfrak{a}_9^\circ(n)$. Since $\mathfrak{a}_{16} = \langle XY, YX, YZ, ZY\rangle_{\text{Lie}}$, we obtain that $\mathfrak{a}_9^\circ(n) \supseteq \mathfrak{a}_{16}(n)$. □

Lemma G.3.36. *We have:*

$$\begin{aligned}
&\mathfrak{a}_6^\circ(n) = \mathfrak{a}_6(n), \quad n \text{ even}, && \mathfrak{a}_3^\circ(n) = \mathfrak{a}_6^\circ(n) = \mathfrak{a}_{13}(n), \quad n \text{ odd},\\
&\mathfrak{a}_3^\circ(n) = \mathfrak{a}_6(n), \quad n \equiv 2 \mod 4, && \mathfrak{a}_3^\circ(n) = \mathfrak{a}_3(n), \quad n \equiv 0 \mod 4.
\end{aligned}$$

Proof. First of all, note that $\mathfrak{a}_3^\circ(n) \subseteq \mathfrak{a}_6^\circ(n)$ for all n, because $\mathfrak{a}_3 \subset \mathfrak{a}_6$. By Lemma G.3.21, we have for even n:

$$\mathfrak{a}_6(n) = \mathfrak{g}_6(n) = \mathfrak{su}(2^n)^{\{P_X, P_{YZ}, P_{ZY}\}}/\text{span}\{P_X, P_{YZ}, P_{ZY}\}.$$

In this case,

$$\tau_n(P_X) = P_X, \qquad \tau_n(P_{YZ}) = P_{ZY}, \qquad \tau_n(P_{ZY}) = P_{YZ},$$

which implies that $\tau_n\mathfrak{a}_6(n) \subseteq \mathfrak{a}_6(n)$, and hence $\mathfrak{a}_6^\circ(n) = \mathfrak{a}_6(n)$.

Recall the automorphism φ_n of $\mathfrak{su}(2^n)$ that up to a sign swaps Y and Z on all even qubits; see Equation (G.18), Equation (G.19). By Lemma G.3.20, we have $\varphi_n\mathfrak{a}_3(n) = \mathfrak{g}_7(n)^{\tilde{\theta}_3}$, where $\tilde{\theta}_3$ is given by Equation (G.47), Equation (G.48), and $\mathfrak{g}_7(n)$ is given by Equation (G.37). Hence,

$$\varphi_n\tau_n\mathfrak{a}_3(n) = (\varphi_n\tau_n\varphi_n^{-1})\varphi_n\mathfrak{a}_3(n) = (\varphi_n\tau_n\varphi_n^{-1})\mathfrak{g}_7(n)^{\tilde{\theta}_3}.$$

When n is even, we have:

$$(\varphi_n\tau_n\varphi_n^{-1})(P_X) = P_X, \qquad (\varphi_n\tau_n\varphi_n^{-1})(P_Y) = P_Z, \qquad (\varphi_n\tau_n\varphi_n^{-1})(P_Z) = P_Y,$$

which implies that $(\varphi_n\tau_n\varphi_n^{-1})\mathfrak{g}_7(n) \subseteq \mathfrak{g}_7(n)$. For $n \equiv 0 \mod 4$, we find

$$(\varphi_n\tau_n\varphi_n^{-1})Q_3 = (\varphi_n\tau_n\varphi_n^{-1})(P_{ZIYX}) = P_{IZXY} = P_Z \cdot Q_3 = Q_3 \cdot P_Z.$$

Then, as in the proof of Lemma G.3.34, we conclude that in this case $\mathfrak{a}_3^\circ(n) = \mathfrak{a}_3(n)$.

Next, in the case $n \equiv 2 \bmod 4$, one checks that

$$Y_1Y_n \in \mathfrak{g}_7(n)^{\tilde{\theta}_3} = \varphi_n\mathfrak{a}_3(n) \;\Rightarrow\; \varphi_n^{-1}(Y_1Y_n) = Y_1Z_n \in \mathfrak{a}_3(n) \;\Rightarrow\; \tau_n^{-1}(Y_1Z_n) = Z_1Y_2 \in \mathfrak{a}_3^\circ(n).$$

Hence, $\mathfrak{a}_3^\circ(n)$ contains all generators of $\mathfrak{a}_6(n)$, proving that $\mathfrak{a}_3^\circ(n) = \mathfrak{a}_6(n)$.

Finally, consider the case when n is odd. Recall that, by Lemma G.3.23,

$$\mathfrak{a}_3^\circ(n) \subseteq \mathfrak{a}_6^\circ(n) \subseteq \mathfrak{a}_{13}(n) = \mathfrak{su}(2^n)^{P_X}/\mathrm{span}\{P_X\}.$$

In order to prove that these are equalities, it is enough to show that $\mathfrak{a}_3^\circ(n)$ contains the generators of $\mathfrak{a}_{13}(n)$. When $n \equiv 1 \bmod 4$, we have

$$Z_1Z_{n-1} \in \mathfrak{g}_7(n)^{\tilde{\theta}_3} = \varphi_n\mathfrak{a}_3(n) \;\Rightarrow\; \varphi_n^{-1}(Z_1Z_{n-1}) = -Z_1Y_{n-1} \in \mathfrak{a}_3(n) \;\Rightarrow\; \tau_n^{-2}(Z_1Y_{n-1}) = Y_1Z_3 \in \mathfrak{a}_3^\circ(n).$$

Since $ZIZ \in \mathfrak{a}_3(3)$, we get that $Z_1Z_3 \in \mathfrak{a}_3^\circ(n)$ and hence $X_1 = -\frac{i}{2}[Y_1Z_3, Z_1Z_3] \in \mathfrak{a}_3^\circ(n)$. Similarly, when $n \equiv 3 \bmod 4$, we have

$$Z_1Z_n \in \mathfrak{g}_7(n)^{\tilde{\theta}_3} = \varphi_n\mathfrak{a}_3(n) \;\Rightarrow\; \varphi_n^{-1}(Z_1Z_n) = Z_1Z_n \in \mathfrak{a}_3(n) \;\Rightarrow\; \tau_n^{-1}(Z_1Z_n) = Z_1Z_2 \in \mathfrak{a}_3^\circ(n).$$

Then from $Y_1Z_2 \in \mathfrak{a}_3^\circ(n)$, we get again that $X_1 \in \mathfrak{a}_3^\circ(n)$. Therefore, all $X_i \in \mathfrak{a}_3^\circ(n)$, and we can generate $\mathfrak{a}_{13}(n)$ from them and the generators X_iX_{i+1}, Y_iZ_{i+1} of $\mathfrak{a}_3(n)$. □

The above lemmas complete the proof of Theorem 4.5.2.

G.3.9 Permutation-invariant subalgebras

In this subsection, we classify all permutation-invariant subalgebras of $\mathfrak{su}(2^n)$ that are generated by single Paulis and products of two Paulis, thus proving Theorem 4.5.3. Recall that, starting from a subalgebra $\mathfrak{a} \subseteq \mathfrak{su}(4)$, we generate the subalgebra $\mathfrak{a}^\pi(n) \subseteq \mathfrak{su}(2^n)$, given by Equation (G.10). Moreover, in Appendix G.2.4, we explained that $\mathfrak{a}$ can be assumed itself invariant under the flip of the two qubits; so we only need to consider $\mathfrak{a}_k^\pi(n)$ for $k = 0, 2, 4, 6, 7, 14, 16, 20$ and $\mathfrak{b}_l^\pi(n)$ for $l = 0, 1, 3$. The complete list of such Lie algebras is presented in Theorem 4.5.3 and reproduced here as follows:

$$\begin{aligned}
&\mathfrak{a}_k^\pi(n) = \mathfrak{a}_k(n), \qquad k = 7, 16, 20, 22,\\
&\mathfrak{a}_0^\pi(n) \cong \mathfrak{u}(1)^{\oplus n(n-1)/2},
\end{aligned}$$

$$
\begin{aligned}
\mathfrak{a}_2^\pi(n) &= \mathfrak{so}(2^n)^{P_Z} \cong \mathfrak{so}(2^{n-1})^{\oplus 2},\\
\mathfrak{a}_4^\pi(n) &= \mathfrak{a}_7(n),\\
\mathfrak{a}_{14}^\pi(n) &\cong \mathfrak{a}_6^\pi(n) = \mathfrak{a}_{20}(n),\\
\mathfrak{b}_l^\pi(n) &= \mathfrak{b}_l(n), \qquad l = 0, 3,\\
\mathfrak{b}_1^\pi(n) &\cong \mathfrak{u}(1)^{\oplus n(n+1)/2}.
\end{aligned}
$$

To start the **proof** of the theorem, we first observe that the following subalgebras of $\mathfrak{su}(2^n)$ are permutation invariant, due to their explicit descriptions (cf. Theorem G.3.1):

$$
\begin{aligned}
\mathfrak{a}_7(n) &= \begin{cases} \mathfrak{su}(2^n)^{\{P_X,P_Y,P_Z\}}/\mathrm{span}\{P_X,P_Y,P_Z\}, & n \text{ even},\\ \mathfrak{su}(2^n)^{\{P_X,P_Y,P_Z\}}, & n \text{ odd}, \end{cases}\\
\mathfrak{a}_{16}(n) &= \mathfrak{so}(2^n),\\
\mathfrak{a}_{20}(n) &= \mathfrak{su}(2^n)^{P_X}/\mathrm{span}\{P_X\},\\
\mathfrak{a}_{22}(n) &= \mathfrak{su}(2^n),\\
\mathfrak{b}_0(n) &= \mathrm{span}\{X_i\}_{1\le i\le n},\\
\mathfrak{b}_3(n) &= \mathrm{span}\{X_i, Y_i, Z_i\}_{1\le i\le n}.
\end{aligned}
$$

It is also easy to see that

$$
\begin{aligned}
\mathfrak{a}_0^\pi(n) &= \mathrm{span}\{X_iX_j\}_{1\le i<j\le n},\\
\mathfrak{b}_1^\pi(n) &= \mathrm{span}\{X_k, X_iX_j\}_{1\le i<j\le n,\ 1\le k\le n}.
\end{aligned}
$$

Thus, we are left to determine $\mathfrak{a}_k^\pi(n)$ for $k = 2, 4, 6, 14$. These cases are treated in the next three lemmas.

Lemma G.3.37. *We have $\mathfrak{a}_2^\pi(n) = \mathfrak{so}(2^n)^{P_Z}$ for all $n \ge 2$.*

Proof. Note that all generators X_iY_j $(i \ne j)$ of $\mathfrak{a}_2^\pi(n)$ commute with P_Z and are skew-symmetric, i.e., satisfy $a^T = -a$. Hence, $\mathfrak{a}_2^\pi(n) \subseteq \mathfrak{so}(2^n)^{P_Z}$. For the opposite inclusion, we use the same strategy as in the proof of Lemma G.3.23. Pick an arbitrary Pauli string $a \in \mathfrak{so}(2^n)^{P_Z}$ not containing any I's; then we want to find a Pauli string $b \in \mathfrak{a}_2^\pi(n)$ such that $[a, b] \ne 0$ and $[a, b]$ has an I in some position. Note that a has an odd number of X's and an odd number of Y's. In particular, after a permutation, a must start with XYZ, XXY, or XYY. Then we let $b = X_1Y_3$, $X_1Z_2Y_3$, or $X_1Z_2Y_3$, respectively. Here $b \in \mathfrak{a}_2^\pi(n)$ because $XZY \in \mathfrak{a}_2(3)$; cf. Appendix G.2.5. □

Lemma G.3.38. *We have $\mathfrak{a}_{14}^\pi(n) \cong \mathfrak{a}_6^\pi(n) = \mathfrak{a}_{20}(n)$ for $n \ge 3$.*

Proof. Let us relabel $X \rightleftharpoons Z$ in $\mathfrak{a}_{14}$. Then $\mathfrak{a}_{14} \subset \mathfrak{a}_{20}$, which implies $\mathfrak{a}_{14}^{\pi}(n) \subseteq \mathfrak{a}_{20}^{\pi}(n) = \mathfrak{a}_{20}(n)$ for all n. Similarly, from $\mathfrak{a}_6 \subset \mathfrak{a}_{20}$, we get $\mathfrak{a}_6^{\pi}(n) \subseteq \mathfrak{a}_{20}(n)$. To finish the proof of the lemma, it is enough to show that $\mathfrak{a}_6^{\pi}(3) = \mathfrak{a}_{14}^{\pi}(3) = \mathfrak{a}_{20}(3)$, because $\mathfrak{a}_{20}(n)$ is generated from $\mathfrak{a}_{20}(3)$ using a process similar to Equation (G.2). The claim now follows from $\mathfrak{a}_k^{\circ}(3) \subseteq \mathfrak{a}_k^{\pi}(3)$ and

$$\dim \mathfrak{a}_6^{\circ}(3) = \dim \mathfrak{a}_{14}^{\circ}(3) = \dim \mathfrak{a}_{20}(3) = 30;$$

see Appendix G.2.5. □

Lemma G.3.39. *We have* $\mathfrak{a}_4^{\pi}(n) = \mathfrak{a}_7(n)$ *for* $n \geq 3$.

Proof. Since $\mathfrak{a}_4 \subset \mathfrak{a}_7$, we have $\mathfrak{a}_4^{\pi}(n) \subseteq \mathfrak{a}_7^{\pi}(n) = \mathfrak{a}_7(n)$ for all $n \geq 3$. To prove the opposite inclusion, it is enough to show that $\mathfrak{a}_4^{\pi}(3) = \mathfrak{a}_7(3)$, because $\mathfrak{a}_7(n)$ is generated from $\mathfrak{a}_7(3)$ using a process similar to Equation (G.2). From $IXX, YZX \in \mathfrak{a}_4(3)$, we get $ZYX \in \mathfrak{a}_4^{\pi}(3)$ and $[IXX, ZYX] = 2iZZI \in \mathfrak{a}_4^{\pi}(3)$. Then, by permutation invariance, also $IZZ \in \mathfrak{a}_4^{\pi}(3)$. Hence, $\mathfrak{a}_4^{\pi}(3)$ contains all generators of $\mathfrak{a}_7(3)$, so it must be equal to it. □

The only thing left to finish the proof of Theorem 4.5.3 is to show that $\mathfrak{so}(2^n)^{P_Z} \cong \mathfrak{so}(2^{n-1})^{\oplus 2}$. This follows from the isomorphism $\mathfrak{su}(2^n)^{P_Z}/\mathrm{span}\{P_Z\} \cong \mathfrak{su}(2^{n-1})^{\oplus 2}$ (see Lemma G.3.26), which is compatible with taking matrix transpose.