

Hamiltonian Engineering and Quantum Compilation

Inaugural dissertation

presented to the faculty of Mathematics and Natural Sciences
of Heinrich-Heine-University Düsseldorf
for the degree of
Doctor of Natural Sciences (Dr. rer. nat.)

by
Pascal Baßler
from Kappelrodeck, Germany

Karlsruhe, November 2025

From the Institute of Theoretical Physics III
at the Heinrich-Heine-University Düsseldorf

Published with permission of the
Faculty of Mathematics and Natural Sciences at
Heinrich-Heine-University Düsseldorf

Supervisor: Prof. Dr. Martin Kliesch
Co-supervisor: Dr. Hermann Kampermann

Date of the oral examination: 29.10.2025

Declaration

Ich versichere an Eides Statt, dass die Dissertation von mir selbstständig und ohne unzulässige fremde Hilfe unter Beachtung der „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf“ erstellt worden ist.

Karlsruhe, 24.11.2025

Pascal Baßler

Zusammenfassung

Quantencomputer haben das Potenzial, bestimmte Rechenprobleme wesentlich schneller zu lösen als klassische Computer. Um das volle Potenzial der Quantenhardware auszuschöpfen, müssen wir in der Lage sein, komplexe Quantenalgorithmen effizient in lauffähige Sequenzen von Quantenoperationen zu übersetzen, die den spezifischen Anforderungen einer bestimmten Hardware gerecht werden, eine Aufgabe, die als Quantenkompilierung bekannt ist. Eine der vielversprechendsten Anwendungen ist die Simulation von Quantensystemen, z.B. in der Quantenchemie, wo skalierbare, fehlerresistente und experimentell durchführbare Simulationstechniken unerlässlich sind.

Im ersten Teil dieser Arbeit werden die neuesten Fortschritte auf dem Gebiet der Quantenkompilierung vorgestellt, wobei der Schwerpunkt auf effizienter Gatterzerlegung und Schaltungsoptimierung liegt. Insbesondere analysieren wir Techniken für die Zerlegung von Clifford-Operationen und allgemeine Ein-Qubit-Rotationen. Clifford-Operationen spielen eine zentrale Rolle in der Quantenfehlerkorrektur, der Charakterisierung von Quantenhardware und der Quantenkryptografie. Wir untersuchen asymptotisch optimale Zerlegungsmethoden, einschließlich Konstruktionen, die entweder Zwei-Qubit-Gatter oder globale Verschränkungsoperationen verwenden, wobei letztere besonders gut für Plattformen wie Ionenfallen geeignet sind. Für Ein-Qubit-Rotationen diskutieren wir Approximationsverfahren, die die theoretisch optimale Skalierung der Gatteranzahl in Bezug auf die gewünschte Präzision erreichen. Über die Gattersynthese hinaus untersuchen wir verschiedene hardwarenahe Optimierungsstrategien, die darauf abzielen, die Tiefe der Schaltung und die Anzahl der Gatter zu verringern. Dazu gehören Techniken für das Qubit-Routing in Systemen mit begrenzter Konnektivität sowie Methoden zur Beseitigung redundanter Operationen.

Der zweite und primäre wissenschaftliche Beitrag dieser Arbeit konzentriert sich auf die Implementierung effektiver Vielkörper-Interaktionen auf Quantenhardware durch die Kombination der systemeigenen Dynamik mit Kontrolloperationen, die wir als Hamiltonian Engineering bezeichnen. Motiviert durch die zentrale Rolle der Hamiltonian-Simulation in Quantenwissenschaften, stellen wir einen allgemeinen und effizienten Ansatz für die Synthese beliebiger Hamiltonians unter milden Lokalitätseinschränkungen vor. Wir schlagen einen neuartigen Ansatz der linearen Programmierung (LP) vor, der Pulssequenzen berechnet, die aus π - oder $\pi/2$ -Rotationen auf einem Qubit bestehen, die zu bestimmten Zeitpunkten während der Evolution unter dem nativen Hamiltonian des Systems angewendet werden. Unser LP-basierter Ansatz bietet mehrere Vorteile. Erstens hängt seine klassische Laufzeit nur von der Anzahl und der Lokalität der Interaktionen im System-Hamiltonian ab und nicht direkt von der Gesamtgröße des Systems. Zweitens erweitern wir die Methode um die Robustheit gegenüber häufigen Fehlerquellen wie der Pulsdauer und systematischen Kontrollfehlern, indem wir Werkzeuge aus der "average Hamiltonian" Theorie und robusten zusammengesetzten Pulsen nutzen. Unser Ansatz ermöglicht die effiziente Realisierung der Hamiltonian-Dynamik mit hoher Genauigkeit und minimalem experimentellem Aufwand. Die Skalierbarkeit des Ansatzes wird durch Simulationen demonstriert, einschließlich der Generierung eines zufälligen Zweikörper-Hamiltonians auf einem 225-Qubit-Gitter in weniger als einer Minute klassischer Berechnungszeit.

Zusammenfassend lässt sich sagen, dass diese Arbeit sowohl theoretische Erkenntnisse als auch praktische Werkzeuge liefert, die die Lücke zwischen dem Entwurf von Algorithmen und

der experimentellen Implementierung im Quantencomputing schließen. Die hier entwickelten Techniken sind direkt auf eine Vielzahl von Plattformen anwendbar, darunter Ionenfallen, neutrale Atome und supraleitende Schaltkreise. Indem sie die Grenzen des Hamiltonian Engineering erweitert, ermöglicht diese Arbeit effizientere, robustere und skalierbare Quantensimulationen auf Geräten in naher Zukunft. Schließlich bringt sie uns der Verwirklichung von Quantentechnologien näher, die eventuell in der Lage sein werden, nützliche Probleme zu lösen.

Abstract

Quantum computers hold the promise of solving certain computational problems significantly faster than classical computers. Unlocking the full potential of quantum hardware depends on our ability to efficiently translate high-level quantum algorithms into executable sequences of quantum operations tailored to the specific constraints of a given device, a task known as quantum compilation. Among the most promising applications is the simulation of quantum systems, such as in quantum chemistry, where scalable, error resilient, and experimentally feasible Hamiltonian simulation techniques are essential.

In the first part of this thesis, we present recent advances in the field of quantum compilation, emphasizing on efficient gate decomposition and circuit optimization. In particular, we analyze compilation techniques for Clifford circuits and general single-qubit rotations. Clifford operations play a central role in quantum error correction, the characterization of quantum devices and quantum cryptography. We explore asymptotically optimal decomposition methods, including constructions using either two-qubit gates or global entangling operations, the latter being especially well-suited for platforms like trapped ions. For single-qubit unitaries, we discuss approximation schemes that achieve the theoretically optimal scaling of gate count with respect to the desired precision. Beyond gate synthesis, we examine various hardware-aware optimization strategies aimed at reducing circuit depth and gate overhead. These include techniques for qubit routing in systems with limited connectivity, as well as methods for removing redundant operations.

The second and primary, scientific contribution of this thesis focuses on implementation of effective many-body interactions on quantum hardware by combining the system's native dynamics with control operations, which we call Hamiltonian engineering. Motivated by the central role of Hamiltonian simulation in quantum science, we introduce a general and efficient framework for synthesizing arbitrary target Hamiltonians under mild locality constraints. We propose a novel linear programming (LP) approach that computes pulse sequences consisting of single-qubit π or $\pi/2$ rotations applied at certain times during free evolution under the system's native Hamiltonian. Our LP-based framework offers multiple advantages. First, its classical runtime depends only on the number and locality of the interactions in the system Hamiltonian, not directly on the overall system size. Second, we extend the method to include robustness against common error sources such as finite pulse time effects and systematic control errors, leveraging tools from average Hamiltonian theory and robust composite pulses. Our approach enables the efficient realization of high-fidelity Hamiltonian dynamics with minimal experimental overhead. The scalability of the approach are demonstrated through simulations, including the engineering of a random two-body Hamiltonian on a 225 qubit lattice in under a minute of classical preprocessing.

In conclusion, this thesis contributes both theoretical insights and practical tools that bridge the gap between algorithm design and experimental implementation in quantum computing. The techniques developed here are directly applicable to a variety of platforms, including trapped ions, neutral atoms, and superconducting circuits. By advancing the frontiers of Hamiltonian engineering, this work enables more efficient, robust, and scalable quantum simulations on near-term devices. Finally, it brings us closer to realizing quantum technologies, which are possibly capable of solving meaningful problems.

Acknowledgment

First, I am deeply grateful to my supervisor, Martin Kliesch, for introducing me to the field of quantum information and to scientific research more broadly. His continuous support, insightful discussions, and guidance have been invaluable throughout my doctoral journey. I have greatly benefited from our collaboration and from the many short and long conversations we shared. A special thank you goes to Markus Heinrich, who took on the role of an unofficial co-supervisor. His great ideas and unwavering support have consistently inspired and motivated me.

I would also like to thank my collaborators Martin Kliesch, Markus Heinrich, Matthias Zipper and Christopher Cedzich from our group, as well as Patrick Huber and Michael Johanning, from the university of Siegen, for their fruitful discussions and the productive collaborations. I am particularly grateful to Özgün Kum for his friendship and for the thought-provoking conversations about quantum compiling and beyond. His perspective has often helped me reflect more deeply on my research. I would like to acknowledge Sara Mina Benjadi, my former bachelor student, for her thesis on “Single Qubit Quantum Compiling”, which contributed to this work.

My thanks also go to Raphael Brieger for generously sharing the Latex template used for this thesis. A special note of appreciation goes again to Markus Heinrich for proofreading parts of this thesis and offering helpful feedback.

Finally, and most importantly, I am deeply grateful to my wife, Alena. This dissertation would not have been possible without her unwavering support, patience, and love. Thank you, Alena, for standing by my side through the highs and lows, for encouraging me in difficult moments, and for being my source of strength. I am truly grateful that we have walked this path together.

Contents

Table of contents	x
1 Introduction	1
1.1 Overview of Results	2
1.2 Structure of the Thesis	3
2 Theoretical background	5
2.1 Mathematical preliminaries and notation	5
2.2 Linear programming	7
2.2.1 Sparse solutions	7
2.2.2 Duality	8
2.2.3 Algorithms for solving linear programs	9
2.3 Quantum gates	9
2.3.1 Single-qubit gates	10
2.3.2 Multi-qubit gates	11
2.4 Stabilizer formalism	13
2.5 Product formulae	15
2.6 Average Hamiltonian theory	17
2.7 Robust composite pulses	18
3 Quantum compilation	20
3.1 Decomposition and gate synthesis	21
3.1.1 Decomposing Clifford unitaries	21
3.1.2 Decomposing Single-qubit unitaries	25
3.2 Quantum circuit optimization	28
3.2.1 Quantum hardware and native operations	28
3.2.2 Qubit mapping and routing	30
3.2.3 Minimize redundant operations	32
3.3 Outlook and Challenges	33
4 Hamiltonian engineering	35
4.1 A brief history of Hamiltonian engineering	36
4.2 Time-optimal Ising Hamiltonian engineering	37
4.2.1 Conjugation of Ising Hamiltonians	37
4.2.2 The linear program approach	38
4.2.3 Bounds on the minimal evolution time	39
4.2.4 Complexity and efficient heuristic	40
4.2.5 Applications	41
4.3 General, efficient, and robust Hamiltonian engineering	43
4.3.1 Conjugation of many-body Hamiltonians	43
4.3.2 The efficient linear program approach	46
4.3.3 Error robustness	47

4.3.4	Numerical simulations	49
4.4	Summary of results	51
5	Conclusion	53
6	Appendix	70
A	Paper - Synthesis of and compilation with time-optimal multi-qubit gates	70
B	Paper - Time-optimal multi-qubit gates: Complexity, efficient heuristic and gate-time bounds	106
C	Paper - General, efficient, and robust Hamiltonian engineering	130

Chapter 1

Introduction

The field of quantum computing originated from the intersection of physics and computer science, first formalized during the 1981 conference on the “Physics of Computation”. At this event, Richard Feynman delivered his seminal talk “Simulating physics with computers” [1], proposing the use of controllable quantum systems to simulate other quantum systems, a task that classical computers struggle with due to the exponential scaling of the quantum state space. Building on this perspective, other researchers articulated the limitations of classical simulation for quantum dynamics and suggested using quantum systems themselves to simulate quantum phenomena [2]. Deutsch formalized the first model of universal quantum computation in the late 1980s [3], which led to further breakthroughs, including Shor’s algorithm for factoring large integers [4] and Grover’s search algorithm for unstructured databases [5]. These theoretical advancements demonstrated that quantum computers could offer exponential or quadratic speedups over classical methods for specific classes of problems. Simultaneously, efforts began to transition these theoretical models into physical realizations. Early proposals for implementing quantum gates appeared in 1995 [6], followed closely by experimental demonstrations with trapped ions [7]. This period marked the start of quantum computation.

Today, quantum computers are being developed using a variety of physical platforms, including superconducting circuits, neutral atoms, trapped ions, and photonic systems [8]. Although the ultimate goal remains the development of large-scale, fault-tolerant quantum computers, current devices are constrained in size and accuracy. These so-called noisy and intermediate scale quantum (NISQ) devices [9] typically feature tens to hundreds of qubits, with limited connectivity and coherence times, which restrict the design and duration of quantum operations. Despite these limitations, the field has made remarkable progress. Notably, recent experiments have demonstrated early forms of quantum error correction on physical platforms [10, 11], representing a milestone toward fault-tolerant computation. However, the broader goal of establishing “quantum advantage”, where a quantum computer outperforms any classical computer on a well-defined task, remains elusive [12–14].

A central requirement for quantum computation, and particularly for achieving quantum advantage, is the translation of high-level quantum algorithms into executable sequences of hardware-native operations. A process, known as quantum compilation. Digital quantum computing is typically built around a small, universal set of gates. A gate set is considered universal if any unitary operation can be approximated to arbitrary precision by a finite sequence of gates from the set. The Solovay–Kitaev algorithm [15] was among the first general-purpose methods for compiling arbitrary operations with polylogarithmic overhead. Among universal gate sets, the Clifford+T set holds special importance, particularly in the context of fault-tolerant quantum computation [16]. Compilation strategies targeting this gate set have been widely studied and optimized over the years [17–20]. Beyond gate decomposition, quantum compilation must account for architectural constraints, such as qubit connectivity, coherence time, and available gate primitives. On NISQ devices, where circuit depth and gate fidelity are major limitations, efficient compilation must minimize the overhead due to hardware limitations. This demands

Careful optimization of gate count, routing of quantum information, and resilience to device-specific imperfections.

Alongside the digital model of quantum computation, alternative paradigms have emerged. In 1996, Lloyd demonstrated that quantum systems governed by local Hamiltonians could efficiently simulate the dynamics of other quantum systems [21], laying the foundation for quantum simulation. This observation led to the exploration of using physical quantum systems as simulators, early work in this direction included liquid-state nuclear magnetic resonance (NMR) systems [22], which, along with trapped ions [23], formed some of the first viable experimental quantum platforms. In NMR-based quantum computers, nuclear spin states in molecules serve as qubits. These systems were also responsible for the first experimental demonstrations of quantum algorithms [24–26]. Research in this area continues to yield valuable insights into quantum control and simulation [27, 28]. Out of this line of research emerged the concept of designing effective system dynamics by interleaving native Hamiltonian evolution with tailored single-qubit gates. Dodd et al. [29] showed that any fixed two-body interaction, when supplemented with appropriate single-qubit operations, can simulate the dynamics of any other two-body Hamiltonian. This result was later generalized to higher-dimensional systems [30], establishing a theoretical foundation for programmable analog quantum simulation, which we call Hamiltonian engineering. Around the same time efficient methods to engineer individual interactions terms has been proposed [31]. The central idea of Hamiltonian engineering is to construct an effective dynamic by exploiting naturally available interactions and controlling them via fast and accurate single-qubit operations. This makes it particularly attractive for quantum platforms where entangling gates are often slower or more error-prone than single-qubit gates. By carefully timing and selecting these operations, one can synthesize complex Hamiltonians without resorting to digital decomposition, thereby reducing circuit depth and possibly improving fidelity.

1.1 Overview of Results

This thesis presents a selection of state-of-the-art quantum compilation methods. Central to the investigation is the synthesis of quantum circuits within the universal Clifford+T gate set. The thesis addresses the efficient decomposition of Clifford circuits and arbitrary single-qubit unitaries into this gate set, alongside hardware-aware optimizations aimed at minimizing circuit depth and gate count under constraints such as qubit connectivity, decoherence times, and gate fidelity.

Beyond digital quantum compilation, the main scientific contribution of this work lies in the development of a novel Hamiltonian engineering technique based on linear programming (LP). It is based on our publications [32–34]. The research concentrates on the engineering of arbitrary Ising-type interactions, which naturally arise in various physical platforms, including superconducting qubits [35, 36], neutral atoms [37], and, in particular, trapped ions [38–40]. In this context, we focus on ion-trap devices that implement native Ising couplings through magnetic gradient induced coupling (MAGIC) [39]. To efficiently implement any target Ising Hamiltonian from native Ising-type interactions, we introduce a framework based on linear programming. This approach identifies sequences of single-qubit pulses and associated durations of evolution under the system’s fixed Ising Hamiltonian, such that the overall evolution reproduces a desired Ising Hamiltonian. The approach minimizes the total evolution time required to achieve the target Hamiltonian, which is beneficial for NISQ devices, where limited coherence times pose a significant constraint. We analyze the linear programming approach in detail, considering both its complexity and practical implementation. It is shown that finding the globally optimal solution, i.e., the minimum evolution time, is NP-hard [33]. Despite this, we propose a heuristic method that efficiently generates pulse sequences that still implement the target Hamiltonian exactly, although not necessarily with minimal evolution time. We study the scaling behavior of the minimal evolution time as a function of system size, and provide evidence, supported by both numerical experiments and theoretical arguments, that the required evolution time grows

at most linearly with the number of qubits. This favorable scaling makes the approach attractive for mid-scale quantum processors.

The linear programming framework is then extended to encompass the engineering of general many-body Hamiltonians beyond the Ising model [34], which we present in section 4.3. The generalized scheme employs π and $\pi/2$ pulses, exploiting the commutation relation of these pulses with certain interaction terms. We demonstrate that the classical runtime required to compute the pulse sequence depends not directly on the number of qubits, but only on the number and locality of interaction terms in the native system Hamiltonian. This enables the method to scale efficiently with system size. We further enhance the method to account for and mitigate several experimental imperfections, including the finite duration of control pulses, which can otherwise introduce significant errors in practice. To validate the scalability of the method, we consider a system with 225 qubits arranged in a 2D-lattice with two-body interaction and engineer randomly chosen two-body interactions. The pulse sequence required to engineer the corresponding Hamiltonian is computed in under one minute on a classical computer, demonstrating the computational efficiency. To assess robustness, we simulate the time evolution of the system, confirming that the engineered dynamics retain high fidelity even in the presence of experimental imperfections. These results suggest that the proposed techniques are well-suited for implementation on current quantum devices and provide a practical path toward Hamiltonian engineering for quantum computation and simulation.

In summary, this thesis develops a class of Hamiltonian engineering methods grounded in linear programming, providing a robust and scalable way to synthesize arbitrary Hamiltonians using native system dynamics and local control pulses. These methods are designed to be efficient in terms of both classical runtime and quantum evolution time, flexible in their applicability to different target interactions, and robust against common sources of experimental error. Moreover, the pulse sequences they generate are directly implementable in many physical systems without requiring further compilation steps. The results presented in this thesis bridge the gap between high-level algorithm design and hardware-specific implementation, contributing both to the theory of quantum compilation and to its practical realization on NISQ-era quantum devices.

1.2 Structure of the Thesis

This thesis is structured to guide the reader from the fundamental concepts of quantum computation to advanced methods in quantum compilation and Hamiltonian engineering. The content is organized across three chapters, where chapters 3 and 4 can be read independently of each other.

1. In chapter 2, we begin by introducing the mathematical tools and formalism necessary for the remainder of the thesis. Section 2.1 defines the mathematical notation used throughout the work, along with basic concepts from convex analysis and quantum computing. Section 2.3 covers the most commonly used quantum gates, starting with single-qubit operations and their effect on quantum states, followed by two-qubit controlled gates and global multi-qubit operations. Section 2.4 provides an overview of the stabilizer formalism, which offers an efficient framework for representing and manipulating Clifford circuits. Following that, section 2.2 introduces the fundamentals of linear programming, including duality and standard algorithms for solving linear programs. Section 2.5 discusses well-known product formulae, both deterministic and randomized, that are used for approximating quantum dynamics. Section 2.6 presents average Hamiltonian theory (AHT), a technique for approximating the evolution of a time-dependent Hamiltonian with an effective time-independent one. Finally, section 2.7 introduces robust composite pulses, which enable high-fidelity quantum operations even in the presence of systematic errors.
2. In chapter 3, we turn to the topic of quantum compilation, which encompasses gate decomposition and circuit optimization. This chapter addresses both theoretical and practical

aspects relevant to compiling quantum algorithms on real hardware.

- Section 3.1 begins with the state-of-the-art methods for decomposing Clifford unitaries, as described in section 3.1.1. Subsequently, section 3.1.2 explores optimal decomposition strategies for arbitrary single-qubit unitaries, with a focus on minimizing the number of elementary gates.
 - Section 3.2 surveys various optimization techniques used in modern quantum compilers. Section 3.2.1 introduces typical quantum platforms and their constraints from a compilation perspective. Section 3.2.2 addresses strategies to overcome hardware-imposed connectivity limitations. Finally, section 3.2.3 presents techniques for reducing the overall gate count.
3. In chapter 4, we present Hamiltonian engineering methods that form the core scientific contribution of this thesis.
- Section 4.1 provides an overview of relevant prior literature and the development of Hamiltonian engineering techniques.

Following the background and literature review chapters, we present the principal research results developed during the course of this doctoral work:

- Section 4.2 is based on our publications [32, 33] and investigates time-optimal Hamiltonian engineering using native Ising-type interactions. The section introduces a linear programming framework for synthesizing target Ising Hamiltonians with minimized evolution time. It analyzes the scaling behavior of the approach with system size, establishes the NP-hardness of finding time-optimal solutions, and presents an efficient heuristic that enables exact synthesis with reduced computational cost. Several applications are discussed, including their integration into quantum compilation strategies and algorithm design.
- Section 4.3, based on our recent work [34], extends the previous methodology to support the engineering of arbitrary many-body Hamiltonians. We develop an efficient synthesis scheme using π and $\pi/2$ pulses, whose classical computational cost depends only on the locality and number of interactions rather than the total system size. The section further introduces techniques from average Hamiltonian theory to suppress common experimental errors, particularly those arising from finite pulse durations. Numerical simulations demonstrate the efficiency and robustness of the proposed methods.

Chapter 2

Theoretical background

This chapter provides the theoretical foundations underlying the methods developed throughout the thesis. We begin by introducing notation and mathematical preliminaries from convex analysis, and quantum computing. We then discuss the basics of linear programming, which are key to the optimization techniques employed in our research in chapter 4. We introduce the standard form of a linear program (LP), the sparsity of solutions, the concept of duality and methods to solve a LP. In section 2.3, we introduce the basic quantum gates, which form the building blocks of quantum circuits and algorithms. When appropriately combined, a small set of gates can approximate any quantum operation, constituting what is known as a universal gate set. This universality is the foundation of quantum compilation, the process of decomposing a quantum operation into a sequence of gates and optimizing it. In section 2.4, we introduce the stabilizer formalism, which is a useful framework in quantum information theory to efficiently describe and manipulate a class of quantum states, known as stabilizer states, particularly those relevant to quantum error correction and quantum computing. This formalism simplifies the description of quantum error-correcting codes [41–43], and also underpins the theory of the Clifford group, which is central in many areas in quantum information theory, and allows the development of efficient quantum compilation methods [19, 44]. This is followed by an introduction to product formulae, which form the basis for simulating quantum dynamics. Furthermore, we outline the principles of AHT, an analytical framework for describing time-dependent quantum systems. AHT will be used to achieve the robustness of our Hamiltonian engineering approach against control errors. Finally, we present the design of robust composite pulses, which enable precise quantum operations even in the presence of systematic imperfections.

2.1 Mathematical preliminaries and notation

Throughout this thesis, we adopt the following notational conventions. Vectors are denoted by bold letters, such as \mathbf{x} , while matrices are denoted by uppercase letters unless otherwise specified. The identity matrix or operator is written as $\mathbf{1}$, and the all-zero matrix as $\mathbf{0}$, where a subscript $\mathbf{1}_k$ indicates the dimension. The all-ones vector is denoted by $\mathbf{1} = (1, \dots, 1)$, and similarly the all-zeros vector by $\mathbf{0} = (0, \dots, 0)$. For a positive integer n , we define the index set $[n] := \{1, 2, \dots, n\}$. For a vector $\mathbf{x} \in \mathbb{C}^s$, the ℓ_p -norm is defined as $\|\mathbf{x}\|_{\ell_p} := (\sum_{i=1}^s |x_i|^p)^{1/p}$ for $1 \leq p < \infty$, and as $\|\mathbf{x}\|_{\ell_\infty} := \max_i |x_i|$. The *Schatten 2-norm* or *Frobenius norm* for a matrix $A \in \mathbb{C}^{s \times s}$ is defined as $\|A\|_{\text{Fr}} := \|A\|_2 := \|\sigma(A)\|_{\ell_2}$, where $\sigma(A)$ denotes the vector singular values of A . The finite field with two elements is denoted by $\mathbb{F}_2 = \{0, 1\}$, where addition is performed modulo 2.

Many concepts in this thesis rely on concepts from convex analysis. A set C is said to be *convex* if for every pair of points $x_1, x_2 \in C$ the line segment $\lambda x_1 + (1 - \lambda)x_2$ parameterized by $\lambda \in [0, 1]$ also lies in C . A *convex combination* of elements x_1, \dots, x_k is an expression of the form $\sum_{i=1}^k \lambda_i x_i$, where each $\lambda_i \geq 0$ and the coefficients sum to one, i.e., $\sum_{i=1}^k \lambda_i = 1$. A set C is called

a *cone* if for every $x \in C$ and $\lambda \geq 0$, the point λx belongs to C . If C is both a cone and convex, it is called a *convex cone*. A *conic combination* refers to a sum $\sum_{i=1}^k \lambda_i x_i$ where all $\lambda_i \geq 0$. The *convex hull* of a set C is the set of all convex combinations of its elements

$$\text{conv}(C) := \left\{ \sum_i^k \lambda_i x_i \mid x_i \in C, \lambda_i \geq 0, i \in [k], \sum_i^k \lambda_i = 1 \right\}. \quad (2.1)$$

Similarly, the *conic hull* is the set of all conic combinations

$$\text{cone}(C) := \left\{ \sum_i^k \lambda_i x_i \mid x_i \in C, \lambda_i \geq 0, i \in [k] \right\}. \quad (2.2)$$

A set defined as the solution set of a finite number of linear equations and inequalities,

$$\mathcal{P} := \{ \mathbf{x} \in \mathbb{R}^r \mid W\mathbf{x} = \mathbf{a}, V\mathbf{x} \leq \mathbf{b}, \mathbf{a} \in \mathbb{R}^s, \mathbf{b} \in \mathbb{R}^m \}, \quad (2.3)$$

for matrices $W \in \mathbb{R}^{s \times r}$ and $V \in \mathbb{R}^{m \times r}$, is called a *polyhedron*. If such a set is bounded, it is referred to as a *polytope*.

Quantum information systems are modeled using finite-dimensional complex *Hilbert spaces*. For a system of n subsystems, each with local dimension q , the total Hilbert space is $\mathcal{H} := \mathbb{C}^{q^n}$. In this work, we focus exclusively on qubit systems, where each subsystem has local dimension $q = 2$, and hence $\mathcal{H} = \mathbb{C}^{2^n}$. The space of linear operators acting on \mathcal{H} is denoted by $L(\mathcal{H})$. The subset of unitary operators, those $U \in L(\mathcal{H})$ satisfying $U^\dagger U = \mathbf{1}$, is denoted by $U(\mathcal{H})$. For n qubit systems, we write $U(2^n)$ to emphasize the dimension. The special unitaries acting on a single qubit, denoted $SU(2)$, consists of 2×2 unitary matrices with determinant equal to one. We use the Dirac notation to represent quantum states: a state vector is denoted by a *ket* $|\psi\rangle \in \mathcal{H}$, and its dual by a *bra* $\langle\varphi| \in L(\mathcal{H})$. The *computational basis states* of an n qubit system is the standard orthonormal basis $\{|e\rangle : e \in \mathbb{F}_2^n\}$, where each e is a binary string of length n indexing the standard basis vectors of \mathbb{C}^{2^n} . We use the symbol “ \equiv ” to denote equality up to a global phase. That is, for two operators U and V , we write $U \equiv V$ if there exists a real number θ such that $U = e^{i\theta}V$.

The Pauli operators are defined as follows

$$I := \sigma_0 := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X := \sigma_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y := \sigma_y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z := \sigma_z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (2.4)$$

A *Pauli string* on n qubits is a tensor product of Pauli operators, i.e., $P_1 \otimes \cdots \otimes P_n$ with $P_i \in \{I, X, Y, Z\}$ for each $i \in [n]$. The set of all such strings forms an orthogonal basis of the space of linear operators on \mathcal{H} , namely $L(\mathcal{H})$. The *Pauli group* \mathcal{P}^n on n qubits consists of all Pauli strings including a global phase of the form i^k with $k \in \mathbb{Z}_4$, that is,

$$\mathcal{P}^n := \left\{ i^k \bigotimes_{i=1}^n P_i \mid k \in \mathbb{Z}_4, P_i \in \{I, X, Y, Z\}, i \in [n] \right\}. \quad (2.5)$$

In many scenarios, global phases are physically irrelevant, since it does not affect measurement outcomes, and so the *projective Pauli group* is defined by factoring out the global phase $\mathcal{P}^n := \mathcal{P}^n \setminus \langle i \rangle$.

The *Clifford group* is defined as the unitary normalizer of the Pauli group, that is

$$\text{Cl}^n := \left\{ U \in U(2^n) \mid UPU^\dagger \in \mathcal{P}^n, P \in \mathcal{P}^n \subset \mathcal{P}^n \right\} \setminus U(1), \quad (2.6)$$

which is a finite group without the global phases $U(1)$.

2.2 Linear programming

Linear programs, also known as linear optimization problems, aim to minimize or maximize a given linear objective function subject to linear constraints. Here, we discuss the basic notations and important results from the theory of linear programming [45]. A linear program (LP) in *standard form* is given by:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^T \mathbf{x} \\ & \text{subject to} && W\mathbf{x} = \mathbf{a} \\ & && \mathbf{x} \geq \mathbf{0}, \end{aligned} \quad (\text{standardLP})$$

where $\mathbf{x} \in \mathbb{R}_{\geq 0}^r$ is an entry-wise non-negative variable, and $\mathbf{c} \in \mathbb{R}^r$, $\mathbf{a} \in \mathbb{R}^s$, and $W \in \mathbb{R}^{s \times r}$ are given constants. We focus on the case $s \leq r$, which leads to an underdetermined system of equations. The goal is to determine a vector \mathbf{x} that minimizes the *objective function* $\mathbf{c}^T \mathbf{x}$ while satisfying the constraints. Since maximizing the objective function is equivalent to minimizing its negative, we restrict our discussion to minimization problems without loss of generality. A vector \mathbf{x} that satisfies both constraints $W\mathbf{x} = \mathbf{a}$ and $\mathbf{x} \geq \mathbf{0}$ is called a *feasible solution*. If no such solution for the LP exists, the LP is said to be *infeasible*. Among all feasible solutions \mathbf{x} , those that minimize the objective function are referred to as *optimal solution*. We denote an optimal solution by \mathbf{x}^* . In linear, and more generally, convex, optimization problems, an optimal solution exists and the optimal objective function value is unique if a feasible solution exists.

Any LP can be transformed into the standard form given in (standardLP). To illustrate this, consider the general form of a LP:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^T \mathbf{x} \\ & \text{subject to} && W\mathbf{x} = \mathbf{a} \\ & && V\mathbf{x} \leq \mathbf{b}, \end{aligned} \quad (2.7)$$

where $\mathbf{b} \in \mathbb{R}^m$ and $V \in \mathbb{R}^{m \times r}$ define a set of inequality constraints in addition to the equality constraints. To convert this general LP into standard form, we proceed in two steps. First, introduce the *slack variables* $\mathbf{s} \in \mathbb{R}_{\geq 0}^m$ to replace the inequality constraints with equalities $V\mathbf{x} + \mathbf{s} = \mathbf{b}$. Then, ensure non-negativity of all variables by expressing each component of \mathbf{x} as the difference of two non-negative variables $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-$ with $\mathbf{x}^+, \mathbf{x}^- \in \mathbb{R}_{\geq 0}^r$. Substituting these transformations into the original problem yields the equivalent LP

$$\begin{aligned} & \text{minimize} && \mathbf{c}^T \mathbf{x}^+ - \mathbf{c}^T \mathbf{x}^- \\ & \text{subject to} && W\mathbf{x}^+ - W\mathbf{x}^- = \mathbf{a} \\ & && V\mathbf{x}^+ - V\mathbf{x}^- + \mathbf{s} = \mathbf{b} \\ & && \mathbf{x}^+ \geq \mathbf{0}, \quad \mathbf{x}^- \geq \mathbf{0}, \quad \mathbf{s} \geq \mathbf{0}. \end{aligned} \quad (2.8)$$

To express this in standard form, we define a new variable vector $\tilde{\mathbf{x}} := (\mathbf{x}^+, \mathbf{x}^-, \mathbf{s})$ and updated parameters $\tilde{\mathbf{c}} := (\mathbf{c}, -\mathbf{c}, \mathbf{0})$, $\tilde{\mathbf{a}} := (\mathbf{a}, \mathbf{b})$ and the constraint matrix

$$\tilde{W} := \begin{pmatrix} W & -W & \mathbf{0} \\ V & -V & \mathbf{1} \end{pmatrix}. \quad (2.9)$$

This construction shows that any LP, regardless of the presence of inequality constraints or unconstrained variables, can be recast as a standard form LP. Such a transformation is fundamental in both theoretical analysis and the implementation of general-purpose LP solvers, which typically operate on problems in standard form.

2.2.1 Sparse solutions

A solution to the (standardLP) is called *sparse* if it contains only a few non-zero entries. In many applications, such as compressed sensing, data compression, and machine learning, finding

a sparse solution is highly desirable [46–48]. Note, that the left-hand side of the constraint $W\mathbf{x} = \mathbf{a}$ in the (standardLP) generates a cone in \mathbb{R}^r . In this context, a fundamental result from convex geometry, Carathéodory’s theorem, provides a useful bound on the number of non-zero components in feasible solutions.

Theorem 1 (Carathéodory, [49, 50]). *If $\mathbf{a} \in \text{cone}(C) \subset \mathbb{R}^s$, then \mathbf{a} can be written as a convex combination of at most s points in C .*

Applied to our setting, this implies that if a feasible solution exists, then there is one with at most s non-zero components in $\mathbf{x} \in \mathbb{R}_{\geq 0}^r$. In other words, the feasible region of (standardLP) always contains a sparse solution with support size no greater than the number of equality constraints. However, finding the sparsest optimal solution is generally a computationally hard problem.

To ensure sparsity explicitly, one might consider minimizing the so-called ℓ_0 -norm, defined as $\|\mathbf{x}\|_{\ell_0} := \#\{i \mid x_i \neq 0\}$, which counts the number of non-zero entries in \mathbf{x} . However, this is not a true norm and is not even convex. Unfortunately, optimization problems involving the ℓ_0 -norm are NP-hard in general [51], making them impractical for large-scale problems. A common relaxation replaces the ℓ_0 -norm with the convex ℓ_1 -norm. In particular, if we set the cost vector $\mathbf{c} = \mathbf{1}$, the objective function becomes

$$\mathbf{1}^T \mathbf{x} = \sum_i |x_i| =: \|\mathbf{x}\|_{\ell_1}, \quad (2.10)$$

which is convex and can be minimized efficiently using standard LP solvers, i.e. with the simplex method. This leads to a tractable LP whose solution often closely approximates the sparsest possible solution. In the context of compressed sensing, it has been shown that in many cases ℓ_1 -minimization not only approximates but in fact exactly recovers the sparsest solution [52, 53].

2.2.2 Duality

An important principle in optimization theory is duality, which states that an optimization problem can be examined from two complementary perspectives: the *primal* and the *dual* problem. Duality offers insights into the structure of optimization problems and often leads to more efficient solution methods, as well as theoretical guarantees such as bounds and optimality conditions.

In many cases, the term “dual problem” refers specifically to the *Lagrangian dual problem*. We begin with a general (possibly non-convex) constrained optimization problem

$$\begin{aligned} & \text{minimize} && f_0(\mathbf{x}) \\ & \text{subject to} && f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & && h_i(\mathbf{x}) = 0, \quad i = 1, \dots, s, \end{aligned} \quad (\text{opt})$$

where $\mathbf{x} \in \mathbb{R}^r$ is the optimization variable, $f_i : \mathbb{R}^r \mapsto \mathbb{R}$ are inequality constraint functions, and $h_i : \mathbb{R}^r \mapsto \mathbb{R}$ are equality constraint functions. To analyze this problem using duality, we construct the *Lagrangian* associated with (opt), which incorporates the constraints into the objective function

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) := f_0(\mathbf{x}) + \sum_i^m \lambda_i f_i(\mathbf{x}) + \sum_i^s \nu_i h_i(\mathbf{x}), \quad (2.11)$$

where $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^m$ and $\boldsymbol{\nu} \in \mathbb{R}^s$ are known as the *dual variables* or *Lagrange multipliers* for the inequality and equality constraints, respectively. The intuition behind this approach is closely related to the principle of least action in Lagrangian mechanics, where physical systems evolve to minimize an action functional. Similarly, we seek to minimize the Lagrangian over \mathbf{x} , while adjusting the multipliers to enforce constraint satisfaction. The *Lagrange dual function* is then defined as the pointwise infimum of the Lagrangian over \mathbf{x}

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) := \inf_{\mathbf{x} \in \mathbb{R}^r} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}). \quad (2.12)$$

This dual function provides a lower bound on the optimal value of the primal problem for all $\lambda \geq 0$ and any ν . The Lagrange dual problem is to maximize this bound

$$\text{maximize } g(\lambda, \nu), \quad \text{subject to } \lambda \geq 0. \quad (2.13)$$

Solving the dual problem often yields useful information about the primal problem. Under Slater's condition, such as convexity of the objective function and constraints, *strong duality* holds, meaning that the optimal values of the primal and dual problems are equal [54].

This duality framework not only provides theoretical insight but also plays a critical role in many modern optimization algorithms, including interior-point methods [55], support vector machines [56], and dual decomposition techniques [57].

2.2.3 Algorithms for solving linear programs

Two major classes of algorithms are widely used: the simplex method and interior-point methods. While both solve the same class of problems, they do so in different ways.

The *simplex method* operates on the vertices (or corner points) of the feasible region defined by the constraints. Since the feasible region of a linear program is a convex polyhedron, the optimal solution, if it exists, is guaranteed to occur at one of its vertices. The method begins at an initial vertex and moves along the edges of the polyhedron to neighboring vertices that improve the objective function. This is repeated until no further improvement is possible, indicating that an optimal solution has been found. In practice, the simplex method is highly efficient, even though its worst-case runtime is exponential [58, 59]. It naturally returns sparse solutions, since vertices of the feasible region correspond to solutions where many variables are zero.

In contrast to the simplex method, *interior-point methods* traverse the interior of the feasible region rather than moving along its boundary. These methods have polynomial-time worst-case guarantees and are particularly effective for solving large-scale and sparse problems [55]. However, interior-point methods typically produce dense solutions, with many small but non-zero variable values. This can be a drawback in applications where sparsity is important.

2.3 Quantum gates

Quantum computation in the circuit model, also referred to as digital quantum computation, is structured around the formalism of quantum circuits. A quantum circuit describes the evolution of an initial state of a quantum system composed of multiple qubits on which a finite sequence of unitary operators, referred to as quantum gates, act on. An example quantum circuit is depicted in figure 2.1. These gates manipulating the quantum state, and are designed to exploit quantum mechanical phenomena such as superposition and entanglement. Quantum circuits usually culminate in a measurement stage, where one or more qubits are measured.

In the field of quantum compilation, one of the fundamental results is the notion of universality. A gate set is said to be *universal* for quantum computation if, given any unitary operator $U \in U(2^n)$, there exists a quantum circuit consisting solely of gates from this set which approximates U to within any desired accuracy $\varepsilon > 0$ [60]. This is analogous to the classical case where the NAND gate forms a universal basis for Boolean computation. However, the quantum case involves continuous unitary transformations rather than discrete logical functions, and thus approximation becomes a central concern. A well known universal set of gates, called the Clifford+T gate set, consists of the Hadamard gate H, the controlled-X gate CX, and the T gate.

In what follows, we introduce the most commonly used quantum gates. These include single-qubit gates, which act on individual qubits, and multi-qubit gates, also referred as entangling gates, which involve two or more qubits.

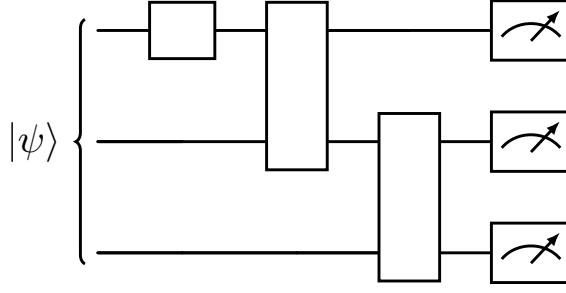


Figure 2.1: Example quantum circuit with an input state $|\psi\rangle$ on $n = 3$ qubits. Each qubit is represented by a horizontal line and time increases from left to right. Unitaries are represented by boxes or special symbols, with input and output qubits. Measurements are depicted on the right end of the qubits.¹

2.3.1 Single-qubit gates

Single-qubit gates are unitary operators belonging to the group $U(2)$. A central tool for visualizing single-qubit operations is the Bloch sphere representation, where single-qubit states correspond to points on the surface of the unit sphere in \mathbb{R}^3 , see figure 2.2. In this framework, single-qubit gates correspond to rotations of the state vector around a chosen axis. The most fundamental single-qubit gates are the Pauli gates X , Y , Z , which correspond to π -rotations around the x -, y -, and z -axes of the Bloch sphere, respectively. These gates are self-inverse, meaning that $P^2 = I$. Another class of gates are the single-qubit Clifford gates, which includes the *Hadamard gate* and the *phase gate* defined as

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{and} \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad (2.14)$$

respectively. The Hadamard gate performs a rotation that maps the basis state to the X -eigenbasis, generating uniform superpositions. Specifically, $H|0\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle)$ and $H|1\rangle = 1/\sqrt{2}(|0\rangle - |1\rangle)$. Therefore, it has the conjugation relations $HXH = Z$ and $HZH = X$, illustrating its role as a basis transformation. When applied to the j -th qubit of a computational basis state $|\mathbf{x}\rangle$, the Hadamard gate produces a local superposition

$$H_j |\mathbf{x}\rangle = \frac{1}{\sqrt{2}} |x_1, \dots, x_{j-1}\rangle (|0\rangle + (-1)^{x_j} |1\rangle) |x_{j+1}, \dots, x_n\rangle. \quad (2.15)$$

The phase gate S , also known as \sqrt{Z} , corresponds to a $\pi/2$ rotation around the z -axis. Its action on a computational basis state introduces an additional phase factor

$$S_j |\mathbf{x}\rangle = i^{x_j} |\mathbf{x}\rangle. \quad (2.16)$$

Similar to the phase gate, the *square-root Pauli* gates are denoted by \sqrt{P} for $P \in \{X, Y, Z\}$, and correspond to $\pi/2$ rotations around different axis. Moreover, all square-root Pauli gates belong to the Clifford group.

In addition to Clifford gates, non-Clifford gates are necessary for universal quantum computation. The most commonly used non-Clifford gate is the T gate, which performs a $\pi/4$ rotation around the z -axis and satisfies $T^2 = S = \sqrt{Z}$. Its matrix form is

$$T = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}. \quad (2.17)$$

¹All quantum circuits in this work are constructed using the Quantikz Latex package [61].

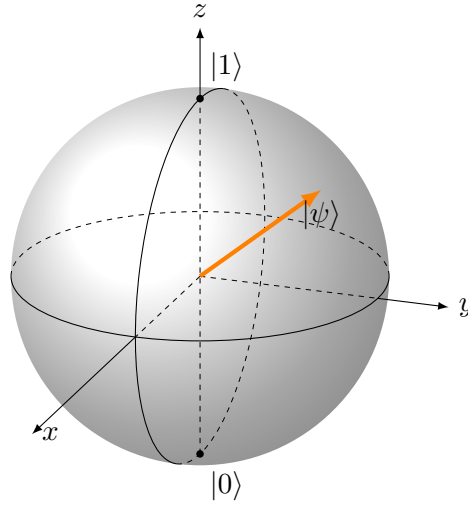


Figure 2.2: Bloch sphere representation of the single-qubit state $|\psi\rangle$. The action of single-qubit gates can be visualized as rotations in the Bloch sphere.

Together, the H and T gates form a universal gate set for single-qubit operations, allowing the approximation of any unitary in $SU(2)$ to arbitrary precision, which we will discuss in section 3.1.2. However, it might be useful to also define the $\pi/4$ rotations around the x -, y - and z -axis and denote these gates as T_x , T_y and T_z respectively. More generally, rotations of arbitrary angle $\theta \in [0, 2\pi)$ around the Bloch sphere axes can be implemented using the continuous families of rotation gates defined as

$$\begin{aligned} R_x(\theta) &:= e^{-i\frac{\theta}{2}X} = \begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix}, \\ R_y(\theta) &:= e^{-i\frac{\theta}{2}Y} = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}, \\ R_z(\theta) &:= e^{-i\frac{\theta}{2}Z} = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}. \end{aligned} \quad (2.18)$$

Finally, the most general single-qubit unitary transformation can be written in exponential form as a rotation by an angle θ around an arbitrary axis $\mathbf{n} \in \mathbb{R}^3$, with $|\mathbf{n}| = 1$,

$$R(\theta, \mathbf{n}) := e^{-i\frac{\theta}{2}\mathbf{n} \cdot \boldsymbol{\sigma}} = \cos(\theta/2)I + i\sin(\theta/2)\mathbf{n} \cdot \boldsymbol{\sigma}, \quad (2.19)$$

where $\boldsymbol{\sigma} = (X, Y, Z)$ denotes the vector of Pauli matrices.

2.3.2 Multi-qubit gates

While single-qubit gates enable rotations in the Bloch sphere and the generation of superposition, another important feature of quantum computing emerges from entangling operations given as multi-qubit gates. Fundamental multi-qubit operations are the controlled Pauli gates, which condition the application of a Pauli operation on the state of another qubit. Two examples within the Clifford gate set are the controlled-X (CX) gate and the controlled-Z (CZ) gate. These gates are defined by

$$CX_{i,j} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \equiv \begin{array}{c} i \text{---} \bullet \\ | \\ j \text{---} \oplus \end{array} \quad \text{and} \quad CZ_{i,j} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \equiv \begin{array}{c} i \text{---} \bullet \\ | \\ j \text{---} \bullet \end{array}, \quad (2.20)$$

where the subscripts on gates indicate on which qubits they act on. The CX gate flips the state of the target qubit j conditional on the control qubit i , effectively acting on the computational basis state as

$$\text{CX}_{i,j} |\mathbf{x}\rangle = |x_1, \dots, x_{j-1}, x_j \oplus x_i, x_{j+1}, \dots, x_n\rangle. \quad (2.21)$$

The CZ gate applies a conditional phase flip, and its action is diagonal in the computational basis

$$\text{CZ}_{i,j} |\mathbf{x}\rangle = (-1)^{x_i x_j} |\mathbf{x}\rangle. \quad (2.22)$$

Notably, CZ gates do mutually commute, where CX gate do not necessarily.

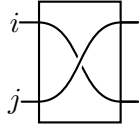
The controlled Pauli gates generalizes naturally to a family of controlled rotations. One such generalization is the controlled- R_z (CR_z) gate, defined for any angle $\theta \in [0, 2\pi)$ by

$$\text{CR}_z(\theta)_{i,j} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{-i\theta/2} & 0 \\ 0 & 0 & 0 & e^{i\theta/2} \end{pmatrix}. \quad (2.23)$$

This gate acts on basis states as $\text{CR}_z(\theta)_{i,j} |\mathbf{x}\rangle = e^{i\theta x_i x_j} |\mathbf{x}\rangle$, and reduces to the CZ gate when $\theta = \pi$. When extended to all pairs of qubits in a system, the global controlled- R_z gate, can be expressed using a symmetric real-valued matrix $A \in \mathbb{R}^{n \times n}$ as

$$\text{GCR}_z(A) := \prod_{i < j} \text{CR}_z(A_{ij})_{i,j}. \quad (2.24)$$

The order of the product does not matter since the CR_z gates mutually commute. Another important two-qubit gate is the SWAP gate, which exchanges the quantum information of the two qubits

$$\text{SWAP}_{i,j} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \equiv \begin{array}{c} i \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ j \end{array} \quad (2.25)$$


While not entangling on its own, the SWAP gate plays a crucial role in “routing” the quantum information, necessary in quantum computing architectures with only short-range interactions, see section 3.2.2.

Gates generated by Ising-type interactions play a central role in our research, as explained in section 4.2. Such gates corresponds to the evolution under the Ising Hamiltonian, and are defined as

$$\text{ZZ}_{i,j}(\theta) := e^{-i\theta Z_i Z_j} = \begin{pmatrix} e^{-i\theta} & 0 & 0 & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & e^{i\theta} & 0 \\ 0 & 0 & 0 & e^{-i\theta} \end{pmatrix}. \quad (2.26)$$

These gates commute with each other and can be applied simultaneously across all qubit pairs. The global ZZ (GZZ) gate, is defined analogously to the global CR_z gate

$$\text{GZZ}(A) := e^{-i \sum_{i < j} A_{ij} Z_i Z_j} \equiv \prod_{i < j} \text{ZZ}(A_{ij})_{i,j}, \quad (2.27)$$

with a symmetric matrix $A \in \mathbb{R}^{n \times n}$. The GZZ gate is a representation of the dynamics under an Ising Hamiltonian. These gates are equivalent to the GCR_z up to single-qubit rotations, and the identity

$$\text{GCR}_z(A) \equiv \text{GZZ}(A/2) \prod_{i=1}^n R_z(b_i/2)_i \quad (2.28)$$

holds, with $b_i := \sum_{j=1}^n A_{ij}$.

2.4 Stabilizer formalism

The stabilizer formalism is a useful mathematical framework that provides a compact and efficient means of describing a class of quantum states, called *stabilizer states*. Originally developed in the context of quantum error correction, it has since become a useful tool in quantum information processing, particularly due to its capacity to efficiently describe stabilizer states, certain quantum error-correcting codes, and operations within the Clifford group [41–43]. The stabilizer formalism enables the manipulation of stabilizer states, not through their exponentially large state vectors, but via the generators of a subgroup of the Pauli group. These states are defined as +1 eigenstates of a set of mutually commuting Pauli operators, which form a subgroup called the *stabilizer group*. This subgroup is Abelian and can be generated by a set of at most n independent generators on a n qubit system. Moreover, the stabilizer formalism underpins the efficient classical simulation of Clifford operations, a fact formalized in the Gottesman-Knill theorem [44]. The theorem states that quantum computations composed solely of Clifford gates, Pauli measurements, and stabilizer state preparation can be simulated efficiently on a classical computer. This result arises from the symplectic structure that underlies the representation of Clifford unitaries, a structure that we now proceed to develop. For a broad and in depth exposition, see the thesis by Heinrich [62].

Let us begin by considering the n qubit Pauli group \mathcal{P}^n , consisting of tensor products of Pauli matrices X , Y , Z , and the identity I , up to a global phase. For an efficient representation and manipulation of these operators, each element of the Pauli group can be mapped to a binary vector in \mathbb{F}_2^{2n} . We define the binary vector representations of Pauli strings as follows. Let $\mathbf{a}_x, \mathbf{a}_z \in \mathbb{F}_2^n$ denote binary vectors indicating the positions of X and Z operators in the tensor product, respectively. Then, the Pauli X and Z strings are defined as

$$X(\mathbf{a}_x) := X^{a_{x,1}} \otimes \cdots \otimes X^{a_{x,n}}, \quad Z(\mathbf{a}_z) := Z^{a_{z,1}} \otimes \cdots \otimes Z^{a_{z,n}}. \quad (2.29)$$

An arbitrary *Pauli string* $P_{\mathbf{a}} \in \mathcal{P}^n$ can then be written as

$$P_{\mathbf{a}} = P_{(\mathbf{a}_x, \mathbf{a}_z)} = i^{\mathbf{a}_x \cdot \mathbf{a}_z} X(\mathbf{a}_x) Z(\mathbf{a}_z), \quad (2.30)$$

where $\mathbf{a} := (\mathbf{a}_x, \mathbf{a}_z) \in \mathbb{F}_2^{2n}$, and the complex phase factor ensures Hermiticity. This encoding defines a group isomorphism (up to global phase) between the projective Pauli group and the additive group \mathbb{F}_2^{2n} . The multiplication of Pauli operators under this map follows the commutation relation

$$P_{\mathbf{a}} P_{\mathbf{b}} = (-1)^{\langle \mathbf{a}, \mathbf{b} \rangle} P_{\mathbf{b}} P_{\mathbf{a}}, \quad (2.31)$$

where the binary symplectic form on \mathbb{F}_2^{2n} is defined as

$$\langle \mathbf{a}, \mathbf{b} \rangle := \mathbf{a}_x \cdot \mathbf{b}_z + \mathbf{a}_z \cdot \mathbf{b}_x = \mathbf{a}^T J \mathbf{b} \quad \text{with } J = \begin{pmatrix} 0 & \mathbb{1}_n \\ \mathbb{1}_n & 0 \end{pmatrix}, \quad (2.32)$$

and the sum is taken in \mathbb{F}_2 , i.e., modulo 2. Similar as the symplectic manifolds in classical mechanics describing the phase space of the system, here, \mathbb{F}_2^{2n} is often referred to as *phase space*.

A *stabilizer state* is a state $|\psi\rangle \in \mathbb{C}^{2^n}$ that is uniquely characterized as the +1 eigenstate of an Abelian subgroup $\mathcal{S} \subset \mathcal{P}^n$, where $-\mathbb{1}_n \notin \mathcal{S}$, i.e. $U|\psi\rangle = |\psi\rangle$ for all $U \in \mathcal{S}$. Instead of storing the full state vector of $|\psi\rangle$, the stabilizer formalism encodes $|\psi\rangle$ using a generating set of the stabilizer group \mathcal{S} . This provides an exponential compression in representation, enabling efficient analysis. There are at most n generators for \mathcal{S} , which follows from the fact that $|\mathcal{S}| = 2^n$ and that the number of generators for a group is at most $\log_2 |\mathcal{S}|$ [44]. Each generator can be represented by $2n + 1$ bits, $2n$ bits for the Pauli string $P_{\mathbf{a}}$ with $\mathbf{a} \in \mathbb{F}_2^{2n}$ and one bit for an additional phase of ± 1 . Stabilizer states can be represented by a binary matrix augmented with phase bits, this structure is known as the *stabilizer tableau* [44]. The tableau contains both

stabilizer and destabilizer generators, which generate the full Pauli group. The stabilizer tableau can be written as

$$\left(\begin{array}{c|c|c} \mathbf{b}_{1,x} & \mathbf{b}_{1,z} & a_1 \\ \vdots & \vdots & \vdots \\ \mathbf{b}_{n,x} & \mathbf{b}_{n,z} & a_n \\ \hline \mathbf{b}_{n+1,x} & \mathbf{b}_{n+1,z} & a_{n+1} \\ \vdots & \vdots & \vdots \\ \mathbf{b}_{2n,x} & \mathbf{b}_{2n,z} & a_{2n} \end{array} \right), \quad (2.33)$$

where rows $1, \dots, n$ represent the destabilizer generators, rows $n+1, \dots, 2n$ represent the stabilizer generators and each $\mathbf{b}_i = (\mathbf{b}_{i,x}, \mathbf{b}_{i,z})$ represents a Pauli string. The pair (S, \mathbf{a}) , with $\mathbf{a} := (a_1, \dots, a_{2n}) \in \mathbb{F}_2^{2n}$ and $S \in \mathbb{F}_2^{2n \times 2n}$ with rows $\mathbf{b}_1, \dots, \mathbf{b}_{2n}$, provides a complete specification of the state.

The Clifford group Cl^n consists of unitaries that normalize the Pauli group, i.e., for every $U \in \text{Cl}^n$ and $P_{\mathbf{b}} \in \mathcal{P}^n$, the conjugation $UP_{\mathbf{b}}U^\dagger$ is again a Pauli operator. Gate representations of Clifford elements include the Hadamard gate H, phase gate S, and the entangling gates CX and CZ. Under conjugation, Clifford unitaries induce an action on the phase space \mathbb{F}_2^{2n}

$$UP_{\mathbf{b}}U^\dagger \propto P_{S\mathbf{b}}, \quad (2.34)$$

where S is an invertible linear map on the phase space. We denote the set of such invertible maps by $GL(\mathbb{F}_2^{2n}) := \{S : \mathbb{F}_2^{2n} \rightarrow \mathbb{F}_2^{2n} \mid S \text{ is invertible}\}$. The group of all such transformations that preserve the symplectic form in eq. (2.32) is the *symplectic group*

$$\text{Sp}^n := \{S \in GL(\mathbb{F}_2^{2n}) \mid S^T J S = J\}. \quad (2.35)$$

The map $S \in \text{Sp}^n$ corresponds to the matrix S in the stabilizer tableau since the induced action of a Clifford unitary also changes the set of stabilizer generators. Each Clifford unitary $U \in \text{Cl}^n$ thus corresponds to a pair (S, \mathbf{a}) , where $\mathbf{a} \in \mathbb{F}_2^{2n}$ tracks the overall phase.

The action of Clifford gates can be explicitly captured through their associated symplectic matrices. We want to provide an intuition behind the structure of the symplectic matrices for specific Clifford gates. The update rules for the phase space element $\mathbf{a} \in \mathbb{F}_2^{2n}$ in the stabilizer tableau are omitted, but can be found in ref. [44]. Let us denote the first n columns of such matrices as the x -part and the latter n columns as the z -part, since these columns represent the Pauli X and Z strings, respectively.

Hadamard gate A Hadamard gate H on qubit k performs a basis change and swaps the k -th column with the $(n+k)$ -th column in S , i.e. exchange the x -part with the z -part. Therefore, the symplectic matrix corresponding to the action of Hadamard gates is a permutation matrix, permuting between the set of columns $1, \dots, n$ and $n+1, \dots, 2n$. Note, that a permutation of the qubits, i.e. a SWAP gate, corresponds to a permutation matrix within each x -/ z -part. For example a Hadamard gate on qubit 2 has the symplectic matrix

$$\left(\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right). \quad (2.36)$$

Phase gate The phase gate leaves Pauli Z operators invariant and maps $X \mapsto Y$. A phase gate on qubit k effectively adds column k , from the x -part, to column $n+k$, from the z -part, modulo 2. For example a phase gate on qubit 2 has the symplectic matrix

$$\left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right). \quad (2.37)$$

CX gate A CX gate with control qubit i and target qubit j updates the tableau by adding column j to column i in the x -part and adding column $n + i$ to column $n + j$ in the z -part. For example a $CX_{1,2}$ gate has the symplectic matrix

$$\left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{array} \right). \quad (2.38)$$

CZ gate Since the $CZ_{i,j}$ gate can be written as $H_j CX_{i,j} H_j$, its symplectic matrix can be obtained by a product of the symplectic matrices of those gates. Therefore, we obtain for a $CZ_{1,2}$ gate the symplectic matrix

$$\left(\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right). \quad (2.39)$$

These gate-specific symplectic matrices provide a practical method for tracking Clifford transformations on stabilizer states, and they will be used in section 3.1.1 to develop systematic decompositions of arbitrary Clifford circuits.

2.5 Product formulae

Simulating quantum systems is one of the most promising applications of quantum computing. It plays a central role in quantum algorithms such as quantum phase estimation (QPE) [63] and solving linear differential equations [64, 65]. Hamiltonian simulation is the task of approximating the unitary time evolution operator generated by a given Hamiltonian H . The challenge in Hamiltonian simulation arises when the Hamiltonian is expressed as a sum of non-commuting terms

$$H = \sum_{i=1}^L h_i H_i, \quad (2.40)$$

with $[H_i, H_j] \neq 0$. In such cases, the exact time evolution operator e^{-itH} cannot be decomposed into a simple product of exponentials e^{-itH_i} . Therefore, approximation methods must be employed. One widely used approach is the Suzuki-Trotter decomposition, which approximates the full evolution by a sequence of short evolutions under the individual terms [66, 67].

More advanced simulation techniques include linear combinations of unitaries (LCU) [68] and quantum signal processing (QSP) [69, 70], which offer improved asymptotic scaling in terms of simulation error and gate complexity. However, for simulating the Heisenberg Hamiltonian the resources required are comparable to product formulae [71]. A detailed discussion of these methods are out of the scope of this work. For a comprehensive treatment of quantum algorithms, specifically Hamiltonian simulation algorithms, we refer the reader to the survey by Dalzell [72].

Hamiltonian simulation also imposes practical constraints on quantum hardware. Many of the simulated interactions do not directly correspond to native gate operations, requiring further decomposition into the native gate sets. Moreover, minimizing circuit depth, i.e. total evolution time, is crucial to mitigate the effects of noise and decoherence on current NISQ devices.

A *product formula* approximates the time evolution under a Hamiltonian of the form as in eq. (2.40) by decomposing the exponential operator into a sequence of exponentials of simpler terms

$$e^{-itH} \approx e^{-i\alpha_q h_{i_q} H_{i_q}} \dots e^{-i\alpha_1 h_{i_1} H_{i_1}}, \quad (2.41)$$

where q is the total number of evolution steps, and the indices $i_1, \dots, i_q \in [L]$ specify which term is used at each step, where each term possibly occurs multiple times. We say that a product

formula is *deterministic* if the sequence of indices i_1, \dots, i_q is computed using a deterministic algorithm. If the sequence involves randomness, we call the formula *non-deterministic*.

The best-known deterministic product formulae are the Suzuki–Trotter formulae. The first-order (also known as Lie–Trotter) formula is

$$e^{-itH} \approx \left(\prod_{i=1}^L e^{-i \frac{t}{n_{\text{Tro}}} H_i} \right)^{n_{\text{Tro}}} =: S_1(t/n_{\text{Tro}})^{n_{\text{Tro}}} \quad (2.42)$$

and the second-order formula is

$$e^{-itH} \approx \left(\prod_{i=L}^1 e^{-i \frac{t}{2n_{\text{Tro}}} H_i} \prod_{i=1}^L e^{-i \frac{t}{2n_{\text{Tro}}} H_i} \right)^{n_{\text{Tro}}} =: S_2(t/n_{\text{Tro}})^{n_{\text{Tro}}}, \quad (2.43)$$

where n_{Tro} is the *number of Trotter cycles*. Increasing n_{Tro} improves the accuracy of the simulation at the cost of a longer sequence of exponentials. We adopt the convention $\prod_{i=1}^L A_i := A_L \dots A_1$ and $\prod_{i=L}^1 A_i := A_1 \dots A_L$ to indicate the order of the products of non-commuting operators. Higher-order approximations are obtained recursively using Suzuki’s method [67, 73]. The $2k$ -th order Suzuki–Trotter formula (for $k > 1$) is defined as

$$S_{2k}(t) := S_{2k-2}(u_k t)^2 S_{2k-2}((1 - 4u_k)t) S_{2k-2}(u_k t)^2, \quad (2.44)$$

with the coefficient

$$u_k := \frac{1}{4 - 4^{1/(2k-1)}}. \quad (2.45)$$

The approximation error of the $2k$ -th order Suzuki–Trotter formula is bounded as

$$\|S_{2k}(t/n_{\text{Tro}})^{n_{\text{Tro}}} - e^{-itH}\| \leq O \left(\left(t \sum_{i=1}^L \|H_i\| \right)^{2k+1} n_{\text{Tro}}^{-2k} \right), \quad (2.46)$$

with the spectral norm $\|H_i\|$ [74].

A prominent non-deterministic product formula for Hamiltonian simulation is the qDrift method [75]. Unlike deterministic methods that apply each Hamiltonian term in a fixed order, qDrift selects Hamiltonian terms randomly according to a probability distribution weighted by the interaction strength h_i . Then, the time evolution under the selected terms are applied in a randomized sequence of length N . Assume, that $h_i \geq 0$, and define the sum over all interaction strengths $\lambda := \sum_i h_i$. In qDrift, each term H_i is selected with probability $p_i = h_i/\lambda$, and evolved for a short time $\tau = t\lambda/N$. The resulting evolution is represented by a quantum channel describing a mixture of unitaries

$$\mathcal{E}(\rho) = \sum_i p_i e^{i\tau H_i} \rho e^{-i\tau H_i}, \quad (2.47)$$

where ρ is the input quantum state. Applying this randomized channel N times approximates the ideal unitary evolution channel

$$\mathcal{U}(\rho) = e^{itH} \rho e^{-itH}. \quad (2.48)$$

The diamond-norm error between the exact and approximate channels is bounded as

$$\|\mathcal{U} - \mathcal{E}^N\|_{\diamond} \leq \frac{2\lambda^2 t^2}{N}. \quad (2.49)$$

This upper bound highlights one of qDrift’s advantages: the error depends on λ , the sum of the coefficients h_i , and not on the number of terms L . As a result, qDrift scales favorably in Hamiltonians with many small terms and can require fewer steps than traditional Trotter-based

methods in such cases. However, qDrift also has limitations. Since qDrift approximates the time evolution channel, rather than the unitary it can be difficult to incorporate qDrift as a subroutine in algorithms that seek to manipulate the unitary directly.

Product formulae are attractive for their simplicity and low circuit overhead. They are especially useful when the individual Hamiltonian terms H_i are local, or otherwise easy to exponentiate. Deterministic product formulae, such as Suzuki–Trotter expansions, are well-suited when deterministic circuit generation is desired. In contrast, non-deterministic formulae, such as qDrift, might scale better for large systems. However, the limitations of product formulae arise in cases where strong non-commutativity leads to large approximation errors, requiring very small evolution time steps [74, 75].

2.6 Average Hamiltonian theory

Given a time-dependent Hamiltonian $H(t)$, it is sometimes advantageous to approximate the dynamics using a time-independent effective Hamiltonian, commonly referred to as the *average Hamiltonian*, denoted by H_{av} [76]. This approximation is particularly useful when analyzing systems under periodic driving or in perturbative regimes [77, 78]. The idea is to approximate the full time evolution over a fixed interval of duration T by a unitary evolution generated by a constant Hamiltonian

$$\mathcal{U}(T) \approx e^{-iT H_{\text{av}}}, \quad (2.50)$$

where $\mathcal{U}(T)$ is the time evolution operator governed by the Schrödinger equation

$$\frac{d\mathcal{U}(t)}{dt} = -iH(t)\mathcal{U}(t), \quad \text{and} \quad \mathcal{U}(0) = \mathbb{1}. \quad (2.51)$$

A method to construct such an average Hamiltonian is provided by the Magnus expansion, which expresses the effective generator of the evolution in a series expansion

$$H_{\text{AV}} = H_{\text{av}}^{(1)} + H_{\text{av}}^{(2)} + \dots, \quad (2.52)$$

where each term captures progressively higher-order corrections due to the non-commutativity of $H(t)$ at different times [79]. An advantage of the Magnus expansion is that exponentiation of the truncated series still yields a unitary operator. The first-order term of the Magnus expansion corresponds to the averaged Hamiltonian

$$H_{\text{av}}^{(1)} := \frac{1}{T} \int_0^T H(\tau) d\tau, \quad (2.53)$$

while the second-order term accounts for the leading-order non-commutativity effects

$$H_{\text{av}}^{(2)} := \frac{1}{2iT} \int_0^T \int_0^\tau [H(\tau), H(\tau')] d\tau' d\tau. \quad (2.54)$$

Higher-order terms involve increasingly complex nested commutators. The convergence of the Magnus expansion is guaranteed under certain conditions, notably if the integral of the operator norm of $H(t)$ satisfies [80]

$$\int_0^T \|H(\tau)\|_2 d\tau < \pi. \quad (2.55)$$

In practice, a useful rule of thumb for ensuring rapid convergence is

$$\max_{\tau \in [0, T]} \|H(\tau)\|_2 T \ll 1, \quad (2.56)$$

where $\|\cdot\|_2$ denotes the Schatten 2-norm, also known as the Frobenius norm [81]. In many applications, only the first order approximation is considered, yielding the average Hamiltonian $H_{\text{av}} = H_{\text{av}}^{(1)}$.

The convergence of the Magnus expansion can often be significantly improved by transforming $H(t)$ into a suitable *interaction frame*, also known as the interaction picture in quantum mechanics textbooks. This transformation effectively removes a dominant part of the Hamiltonian, leaving a remainder that is smaller in norm and therefore more amenable to perturbative treatment. Assume that the Hamiltonian can be split into two parts

$$H(t) = H_A(t) + H_B(t), \quad (2.57)$$

where both $H_A(t)$ and $H_B(t)$ may be time-dependent and non-commuting. We assume that $H_A(t)$ dominates in norm, i.e. $\|H_A(t)\|_2 > \|H_B(t)\|_2$ for all $t \in [0, T]$, and that the propagator $\mathcal{U}_A(t)$ associated with $H_A(t)$ is analytically computable. Moreover, this propagator $\mathcal{U}_A(t)$ satisfies

$$\frac{d\mathcal{U}_A(t)}{dt} = -iH_A(t)\mathcal{U}_A(t), \quad \text{and} \quad \mathcal{U}_A(0) = \mathbb{1}. \quad (2.58)$$

Consider the ansatz to express $\mathcal{U}(t)$ from eq. (2.51) as the product

$$\mathcal{U}(t) = \mathcal{U}_A(t)\tilde{\mathcal{U}}(t), \quad (2.59)$$

where $\tilde{\mathcal{U}}(t)$ is the so-called *interaction frame propagator*. Inserting this into the Schrödinger eq. (2.51) yields

$$\frac{d\tilde{\mathcal{U}}(t)}{dt} = -i \left(\mathcal{U}_A^\dagger(t)H(t)\mathcal{U}_A(t) - i\mathcal{U}_A^\dagger(t) \left(\frac{d\mathcal{U}_A(t)}{dt} \right) \right) \tilde{\mathcal{U}}(t), \quad (2.60)$$

where the *interaction frame Hamiltonian* $\tilde{H}(t)$ is defined as

$$\tilde{H}(t) := \mathcal{U}_A^\dagger(t)H(t)\mathcal{U}_A(t) - i\mathcal{U}_A^\dagger(t) \left(\frac{d\mathcal{U}_A(t)}{dt} \right) = \mathcal{U}_A^\dagger(t)H_B(t)\mathcal{U}_A(t), \quad (2.61)$$

where we used eq. (2.58) to simplify the expression. The total propagator at time T can then be expressed as $\mathcal{U}(T) = \mathcal{U}_A(T)\tilde{\mathcal{U}}(T)$, and the interaction frame propagator $\tilde{\mathcal{U}}(T)$ can itself be approximated using the Magnus expansion

$$\tilde{\mathcal{U}}(T) \approx e^{-iT\tilde{H}_{\text{av}}}. \quad (2.62)$$

Since $\tilde{H}(t)$ is related to $H_B(t)$ by a unitary transformation, and the Schatten 2-norm is invariant under such a transformation, we have $\|\tilde{H}(t)\|_2 = \|H_B(t)\|_2 < \|H(t)\|_2$ for all $t \in [0, T]$. Thus, the interaction frame reduces the effective norm of the Hamiltonian entering the Magnus expansion, leading to faster convergence and better accuracy with fewer terms. The choice of the interaction frame depends on the physical context and the structure of the Hamiltonian. It is often guided by identifying a dominant, analytically tractable part of the Hamiltonian that can be "absorbed" into the frame transformation.

2.7 Robust composite pulses

Robust composite pulses are a class of control techniques specifically designed to perform precise quantum operations even in the presence of systematic errors. These errors, which stem from imperfections in the calibration or stability of control parameters, can significantly reduce the fidelity of quantum gates by introducing coherent deviations in the system's evolution [82, 83]. In the context of quantum computing, where the accurate execution of gate operations is essential for algorithmic reliability, such deviations pose a challenge. Robust composite pulse sequences address this challenge by replacing a single, idealized control pulse with a designed sequence of pulses. Each pulse within the sequence is engineered with specific relative phases and durations, such that the errors introduced by individual pulses interfere destructively, while the desired operation accumulates constructively. More advanced composite pulse schemes offering protection

against a broader range of experimental imperfections. These include not only rotation angle and detuning errors but also phase and control waveform distortions, crosstalk between qubits, and certain classes of non-Markovian noise [84–91].

A variety of composite pulse families exist, each optimized for resilience against a particular subset of errors. Here, we focus on well-established families: BB1 and SCROFULOUS. The BB1 (Broadband 1) sequence is a widely used composite pulse that achieves robustness against rotation angle (pulse length) errors [92]. It suppresses such errors to second-order, making it particularly effective in systems where amplitude fluctuations dominate. However, this comes at the cost of increased pulse sequence duration. In contrast, the SCROFULOUS (Short Composite Rotation For Undoing Length Over and Under Shoot) sequence offers a shorter pulse sequence [93]. It corrects for first-order rotation angle errors while maintaining a shorter total duration compared to BB1. This makes SCROFULOUS attractive in experimental settings where gate speed is a constraint. However, like BB1, it does not provide protection against off-resonance errors. To address this limitation, the SCROBUTUS (Short COMposite Rotation Buffering Two Undesirables with Switchback) sequence was developed as an improved version of SCROFULOUS [85]. It extends the SCROFULOUS scheme by incorporating robustness to both rotation angle and off-resonance errors, while maintaining a relatively short total pulse duration. This makes it particularly suited for scenarios in which both detuning and amplitude fluctuations coexist. Higher-order generalizations of the BB1 and SCROFULOUS sequences have also been proposed. For instance, these extensions can achieve cancellation of higher-order error terms, albeit with increased execution time [94].

Robust composite pulse techniques are important tools in quantum control, enabling precise manipulation of quantum systems in the presence of unavoidable experimental imperfections. By carefully selecting pulse families and tailoring them to the error landscape of a given hardware platform, one can significantly enhance gate fidelity and contribute to the overall reliability of quantum information processing.

Chapter 3

Quantum compilation

Quantum compilation is an important component of the quantum software stack, serving as the translation layer between high-level quantum algorithms and the low-level control instructions that can be physically realized on a quantum processor. This process is analogous to classical compilation in some respects, yet it introduces new challenges due to the fundamental differences between classical and quantum computation, as well as the constraints of current quantum hardware.

In the standard model of digital quantum computation, algorithms are represented as quantum circuits, which consist of sequences of quantum gates acting on qubits, as illustrated in figure 2.1. Each gate corresponds to a unitary operation, and the full circuit implements a global unitary transformation that evolves an initial quantum state into a desired output state. However, quantum hardware does not support the implementation of arbitrary unitary operations. Every physical quantum processor is restricted to a specific set of native quantum gates, known as the *native gate set*. In addition to limitations on the gate set, quantum hardware imposes a range of architectural and operational constraints that significantly complicate the compilation process. These include limited qubit connectivity, which restricts the ability to apply two-qubit gates between arbitrary pairs of qubits, variable gate fidelities, which result in certain operations being noisier than others, and finite coherence times, which constrain the duration over which quantum information can be maintained. We will examine such physical constraints for different quantum architectures in section 3.2.1. Due to these limitations, quantum compilation is necessary, which transforms high-level quantum algorithms to the specific characteristics of the target device.

The compilation process involves several steps. The first step is gate decomposition, in which abstract quantum operations are translated into sequences of gates that belong to the native gate set of the hardware. This is followed by qubit mapping and routing, where logical qubits are assigned to physical qubits in a way that respects the device’s connectivity constraints. Because two-qubit gates are often only feasible between physically adjacent qubits, additional operations, such as SWAP gates or qubit shuttling, may be required. Throughout this process, circuit optimization techniques are applied to reduce gate count and circuit depth, thereby improving the fidelity and efficiency of quantum computations.

This chapter provides an overview of quantum compilation techniques, structured in two main parts. The first part focuses on methods for decomposing quantum circuits into native gate sequences. In section 3.1.1, we explore a decomposition strategy for Clifford unitaries. We then turn to the approximate synthesis of arbitrary single-qubit gates using a discrete gate set in section 3.1.2. The second part of the chapter addresses the challenges posed by hardware constraints and discusses strategies for optimizing quantum circuits. In section 3.2.2, we review approaches for qubit mapping and routing, which are essential for reducing overhead caused by limited connectivity. Finally, in section 3.2.3, we survey techniques for identifying and eliminating redundant or suboptimal operations in quantum circuits, with the goal of minimizing circuit size and improving overall execution performance.

3.1 Decomposition and gate synthesis

A central task in quantum compilation is gate synthesis, which refers to the decomposition of a unitary operator into sequences of native gates. Different quantum hardware platforms, such as superconducting qubits, trapped ions, neutral atoms, and photonic systems, support different gate sets and topologies. For example, superconducting qubit platforms support the controlled-Z (CZ) or cross-resonance (CR) gate as two-qubit entangling gates, but only between qubits that are physically coupled [95–97]. In contrast, trapped-ion systems may allow all-to-all connectivity and have access to global entangling operations such as the Mølmer-Sørensen (MS) gate [98].

When the target operation belongs to a known family, such as Clifford circuits or circuits composed of Clifford and T gates, efficient synthesis algorithms may be available [44, 99, 100]. However, synthesizing general unitaries is substantially more challenging. A generic unitary acting on n qubits is specified by 4^n real parameters, reflecting the exponential growth of the Hilbert space dimension with system size. Indeed, the asymptotic lower bound on the number of CX gates required to decompose a generic n qubit unitary into a sequence of CX and single-qubit gates is $\Omega(4^n)$ [101]. To solve this, compilers exploit structure in the unitary operator, such as sparsity, or tensor product structure, to simplify the decomposition [102, 103]. Techniques from group theory and numerical optimization are frequently employed in this context. For example, the Bruhat decomposition is used to factor Clifford unitaries into simpler components [18], while the ZX-calculus provides a general framework for synthesis of multi-qubit gates [100, 104, 105].

In some cases, approximate synthesis is used to trade off precision for shorter circuits. A common example is the simulation of Hamiltonian dynamics using Trotter-Suzuki product formulae, where the evolution operator is approximated by a sequence of implementable gate layers, see section 2.5. Another prominent method is the Solovay-Kitaev algorithm, which enables the approximation of any single-qubit unitary using a finite gate set with polylogarithmic overhead in the desired precision [15], which has been extended to qudit systems and generalized to multi-qubit operations.

Gate synthesis also includes a variety of specialized tasks. For example, the decomposition of multiple-controlled gates, such as the Toffoli gate, or the transformation of logical gates (e.g., the logical T gate) into a fault-tolerant implementation under a given quantum error correction scheme [106, 107]. While these are important in broader compilation efforts, here we narrow our focus to the synthesis of Clifford and single-qubit gates due to their important role in quantum computation.

3.1.1 Decomposing Clifford unitaries

Clifford unitaries constitutes a central class of operations in quantum computing, both from a theoretical and a practical standpoint. As elements of the Clifford group, defined in eq. (2.6), they map Pauli operators to Pauli operators under conjugation, a property essential to the stabilizer formalism and quantum error correction. Moreover, Clifford circuits often serve as the backbone for more general quantum protocols [16, 108, 109]. In fault-tolerant computing, for example, Clifford operations can typically be implemented transversally, while non-Clifford gates like T or Toffoli gates require more expensive procedures such as magic state injection and distillation [110, 111].¹ Clifford unitaries also possess valuable statistical properties. Notably, the Clifford group forms a unitary 3-design [114], meaning that random Clifford operations approximate the Haar measure up to the third moment. This property makes Cliffords suitable for a variety of randomized protocols, including randomized benchmarking [109], quantum process tomography [108], and applications in quantum cryptography [115].

The efficient decomposition of Clifford unitaries into native gate sets is therefore essential for practical implementations on quantum devices. Unlike general unitaries, Clifford operators

¹However, recent improvements shows that logical non-Clifford gates might require comparable resources as Clifford gates [112, 113].

admit compact circuit representations and benefit from a rich algebraic structure. In particular, the Clifford group is finite up to global phase and can be described using symplectic matrices over finite fields, see section 2.4, enabling fast classical algorithms for simulation, synthesis, and optimization [19, 20, 116]. These structural properties can be exploited to produce efficient decompositions, with a gate count that only scales quadratic with the number of qubits [44]. However, the decomposition strategy must be tailored to the target architecture. On hardware with native two-qubit gates such as CX or CZ and limited qubit connectivity, synthesizing Clifford circuits while minimizing gate count becomes a nontrivial optimization problem [117]. In other settings, such as trapped-ion systems that support global entangling gates, alternative decomposition methods that leverage the global nature of these operations can offer significant resource savings [116, 118].

In this section, we review methods for decomposing Clifford unitaries, with an emphasis on exploiting the underlying algebraic structure of the Clifford group. We also discuss strategies that map these decompositions to native gate sets with minimal overhead, with a focus on utilizing global entangling operations.

The Bruhat decomposition

A powerful tool for synthesizing Clifford unitaries efficiently is the *Bruhat decomposition*, a well-known factorization originating in the theory of Lie groups [119]. This decomposition has been adapted to the symplectic group \mathbf{Sp}^n over the finite field \mathbb{F}_2 [120, 121]. As established in section 2.4, any Clifford unitary on n qubits can be represented by a stabilizer tableau (S, \mathbf{a}) , where $S \in \mathbf{Sp}^n$ is a $2n \times 2n$ symplectic matrix over \mathbb{F}_2 , and $\mathbf{a} \in \mathbb{F}_2^{2n}$ encodes phase information. The goal is to decompose S into a sequence of native gate layers using the Bruhat decomposition. This decomposition is constructive, making it a useful subroutine in many efficient algorithmic approaches to Clifford synthesis [18, 19]. The decomposition expresses any symplectic matrix as a product of elements from two subgroups: the *Weyl group* and *Borel group*. These subgroups correspond to the action of standard Clifford gates. To abstract away from the specific structure of individual quantum circuits, we adopt a notation in which a *layer of G gates*, denoted as -G-, indicating that multiple G gates are applied to any subset of qubits. For example, -CX- refers to any possible quantum circuit consisting of CX gates. A sequence of layers is represented by the concatenation of their labels, such as -CX-H- for a CX layer followed by a H layer.

We begin by defining the Weyl and Borel subgroups of \mathbf{Sp}^n in terms of their generators represented as gates and then state the Bruhat decomposition.

Definition 2 (Weyl group). *The Weyl group W_n on n qubits is a subgroup of \mathbf{Sp}^n and generated by the action of SWAP and H gates on the computational basis state $|0\rangle$ on n qubits.*

Definition 3 (Borel group). *The Borel group B_n on n qubits is a subgroup of \mathbf{Sp}^n and generated by the action of CX and S gates on the computational basis state $|0\rangle$ on n qubits.*

Proposition 4 (Bruhat Decomposition [18]). *Any symplectic matrix $S \in \mathbf{Sp}^n$ can be expressed as a product*

$$S = B_1 W B_2, \quad (3.1)$$

where $B_1, B_2 \in B_n$ and $W \in W_n$.

In the following, we explain in detail how the factors in eq. (3.1) can be decomposed further and represented as a sequence of gate layers from the gate set $\{\text{CX}, \text{CZ}, \text{S}, \text{H}, \text{P}\}$, where P denotes arbitrary Pauli gates. To this end, we use the symplectic representation of the gates as explained in section 2.4. Elements in the Borel group B_n have the matrix representation of the form

$$\left(\begin{array}{c|c} A & AB \\ \hline 0 & (A^{-1})^T \end{array} \right), \quad (3.2)$$

where $A \in GL(\mathbb{F}_2^n)$, and $B \in GL(\mathbb{F}_2^n)$ is symmetric. Note, that the lower left block has to be $(A^{-1})^T$ to satisfy the symplectic relation $S^T JS = J$. The Borel group can be further decomposed into a subgroup generated by CX gates and a subgroup generated by CZ and S gates. We denote the subgroup generated solely by CX gates by $C_n \subset B_n$. The matrix representation of elements in C_n have the form

$$\left(\begin{array}{c|c} A & 0 \\ \hline 0 & (A^{-1})^T \end{array} \right). \quad (3.3)$$

Therefore, a CX gate layer can be represented by a binary invertible matrix $A \in GL(\mathbb{F}_2^n)$. Similarly, we denote the subgroup generated by CZ and S gates by $B_n^0 \subset B_n$. The matrix representation of elements in B_n^0 have the form

$$\left(\begin{array}{c|c} \mathbf{1} & B \\ \hline 0 & \mathbf{1} \end{array} \right), \quad (3.4)$$

where $B \in GL(\mathbb{F}_2^n)$ is symmetric. More Precisely, if $B_{jj} \neq 0$, then there is a S gate on the j -th qubit, and if $B_{ij} = B_{ji} \neq 0$ for a $i \neq j$, then there is a $CZ_{i,j}$ gate on the (i, j) qubit pair. Thus, any element in the Borel group can be written as the product

$$\left(\begin{array}{c|c} A & 0 \\ \hline 0 & (A^{-1})^T \end{array} \right) \left(\begin{array}{c|c} \mathbf{1} & B \\ \hline 0 & \mathbf{1} \end{array} \right), \quad (3.5)$$

leading to a gate sequence -CX-CZ-S- or -CX-S-CZ-. Elements from the Weyl group correspond to a gate sequence -SWAP-H-, implementing qubit permutations and basis transformations.

Combining the above, any Clifford unitary can be decomposed as

$$-P-CX-CZ-S-SWAP-H-CX-CZ-S-, \quad (3.6)$$

where -P- is a Pauli gate layer. Using known gate identities and commutation relations, this sequence can be further simplified. In particular, one -CX- can be absorbed into an adjacent -CZ- at the cost of an additional -H- layer [19]. This yields the more compact decomposition

$$-P-S-CX-CZ-H-CZ-H-S-, \quad (3.7)$$

This reduction is advantageous because single-qubit gates (e.g. H, S) are typically faster and less error-prone than two-qubit gates on most platforms. An information theoretic asymptotic optimal gate count, for the standard Clifford gate set, is $O(n^2/\log(n))$ [44].

In the next section, we explore how access to native multi-qubit entangling gates can reduce execution time even further. We emphasize, however, that circuit depth/gate count is not the best metric for comparing implementations involving two-qubit vs. multi-qubit gates. Execution time and parallelization must also be taken into account for a fair comparison.

GZZ gates as entangling gates

We now discuss how the entangling operations of the gate sequence (3.7) can be consolidated into a small number of multi-qubit GZZ gates. These gates, or equivalently layers of ZZ gates, are generated by Ising interactions, which are ubiquitous in quantum computing platforms. For an overview of their physical realization, see section 3.2.1.

A layer of CZ gates, represented as the off-diagonal entries of a symmetric matrix $B \in GL(\mathbb{F}_2^n)$ from (3.4), can be expressed using a single GZZ gate and single-qubit phase gates. Specifically, one can show that

$$\prod_{i < j} CZ_{i,j}^{B_{ij}} = \prod_{i < j} CR_z(\pi B_{ij})_{i,j} \equiv GZZ(\pi/2 B) \prod_{i=1}^n S_i^{b_i}, \quad (3.8)$$

3.1.2 Decomposing Single-qubit unitaries

Single-qubit unitary operations lie at the center of quantum computation. They correspond to arbitrary rotations on the Bloch sphere and serve as the building blocks from which more complex quantum circuits are assembled. Implementing quantum algorithms such as variational quantum algorithm (VQA), quantum approximate optimization algorithm (QAOA), and QSP, relies on the ability to realize single-qubit unitaries with arbitrary rotation angles [69, 122, 123]. Arbitrary single-qubit gates can be represented continuously, typically parameterized by three rotation angles (e.g., Euler angles). However, in practice, quantum processors can only natively implement a small, finite set of elementary gates due to memory constraints in control electronics. Moreover, it is not possible to implement a continuously parameterized gate in a fault-tolerant way due to small imperfections in any quantum experiment. Therefore, it becomes necessary to approximate arbitrary single-qubit unitaries using finite sequences of gates from a universal set.

A commonly used universal set is the single-qubit Clifford+T gate set, defined as $G := \{H, S, T\}$. While Clifford gates (generated by H and S gates) can be efficiently simulated classically and implemented transversally in many error-correcting codes, they do not suffice for universal quantum computation. The inclusion of the T gate, a $\pi/4$ rotation around the z -axis, makes the Clifford gate set universal. However, from a fault-tolerant perspective, the T gate is considerably more expensive than Clifford gates, primarily because it cannot be implemented transversally and must be injected through magic state preparation and distillation techniques [110, 111]. Consequently, a critical metric in gate decomposition is the T-count, the total number of T gates used in the decomposition of a unitary.

The Solovay-Kitaev algorithm provides a general method for approximating arbitrary unitaries to within precision ε , with gate complexity of $O(\log^c(1/\varepsilon))$ for $c \approx 3.97$ [15]. However, this is suboptimal relative to the information-theoretic lower bound of $O(\log_2(1/\varepsilon))$ [124].² Modern synthesis techniques achieve this lower bound and offer explicit construction algorithms [17, 126, 127], which we survey in this section. Similar techniques have also been generalized to single-qutrit unitaries [128].

The general goal of single-qubit gate decomposition is to approximate a target unitary $U \in SU(2)$ using a finite sequence of gates from G , up to a prescribed accuracy ε . A general $U \in SU(2)$ can be written in matrix form as

$$U = \begin{pmatrix} u & -v^* \\ v & u^* \end{pmatrix}, \quad (3.12)$$

for $u, v \in \mathbb{C}$ with $|u|^2 + |v|^2 = 1$.

Definition 6 (Single-qubit unitary approximation problem). *Given a target unitary $U \in SU(2)$, a gate set G , and the precision $\varepsilon > 0$, find a finite sequence $g_1, \dots, g_k \in G$ such that*

$$\|U - g_1 \dots g_k\| \leq \varepsilon. \quad (3.13)$$

This problem can be simplified using the *Euler angle decomposition* of $SU(2)$ operators. Every $U \in SU(2)$ can be written as

$$U = R_z(\alpha) R_x(\beta) R_z(\gamma) = e^{-i\frac{\alpha}{2}Z} e^{-i\frac{\beta}{2}X} e^{-i\frac{\gamma}{2}Z}, \quad (3.14)$$

where $\alpha, \beta, \gamma \in [0, 2\pi)$. Conjugation of the x -axis rotation $R_x(\beta)$ in eq. (3.14) with Hadamard gates yields

$$U = R_z(\alpha) H R_z(\beta) H R_z(\gamma). \quad (3.15)$$

Thus, the task of approximating arbitrary single-qubit unitaries reduces to approximating three diagonal unitaries R_z up to a precision $\varepsilon/3$. The rest of this section focuses on efficient strategies for approximating $R_z(\theta)$ with arbitrary angle $\theta \in [0, 2\pi)$. Modern decomposition methods divide the single-qubit approximation problem into three subproblems [17, 126, 127]:

²While writing up this thesis, there was a new result implementing single-qubit gates in constant T-depth utilizing a catalyst state [125].

1. Identify a set of unitaries \mathcal{V} that approximate the target U within error ε .
2. From this set, select a V that admits an exact decomposition in the Clifford+T gate set.
3. Find an exact decomposition using k gates such that $V = g_1 \dots g_k$ with $g_1, \dots, g_k \in G$.

We begin by addressing the first subproblem. The following result provides conditions on a unitary operator to approximate a diagonal rotation gate with arbitrary angle.

Lemma 7 ([126, 127]). *Given a diagonal target unitary $R_z(\theta)$ with angle $\theta \in [0, 2\pi)$ and a special unitary*

$$V = \begin{pmatrix} u & -v^* \\ v & u^* \end{pmatrix}, \quad (3.16)$$

then the spectral norm distance is

$$\|R_z(\theta) - V\| = \sqrt{2 - 2\operatorname{Re}(ue^{i\theta/2})}. \quad (3.17)$$

Proof. Recall from section 2.3.1, that

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{+i\theta/2} \end{pmatrix}, \quad (3.18)$$

and define $z := e^{-i\theta/2}$. Then,

$$\begin{aligned} \|R_z(\theta) - V\|^2 &= |u - z|^2 + |v|^2 \\ &= (u - z)^*(u - z) + v^*v \\ &= u^*u - u^*z - z^*u + z^*z + v^*v \\ &= 2 - 2\operatorname{Re}(uz^*), \end{aligned} \quad (3.19)$$

where we used that $R_z(\theta)$ and V are special unitaries, i.e. $u^*u + v^*v = 1$ and $z^*z = 1$. \square

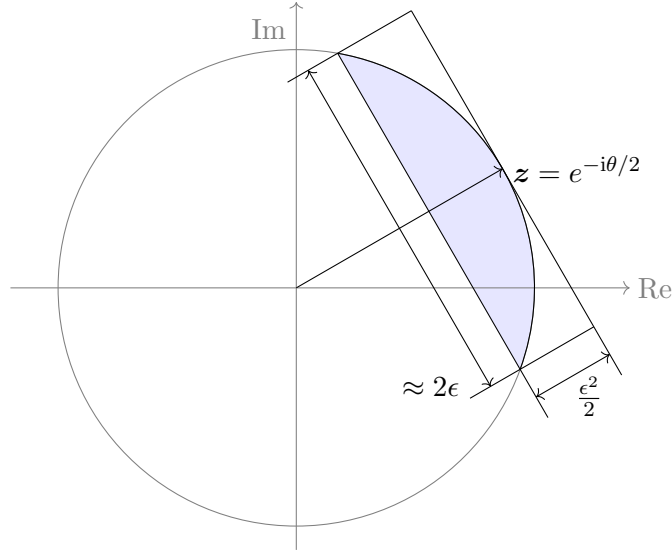


Figure 3.1: Feasible candidates of $u \in \mathbb{C}$ in the colored area yield a unitary matrix V as in lemma 7, approximating the diagonal unitary $R_z(\theta)$ up to precision ε [126].

With lemma 7 a region in the complex plane can be defined for feasible candidates $u \in \mathbb{C}$ as the first entry in the matrix V . This region can be expanded, thereby reducing the required gate count, by incorporating mid-circuit measurements [127]. We represent the complex numbers as

2D vectors in the complex plane, i.e. for a $u \in \mathbb{C}$ we write $\mathbf{u} = (\text{Re}(u), \text{Im}(u))^T$. From lemma 7 we get the condition

$$\|\mathbf{R}_z(\theta) - V\| \leq \varepsilon \iff \text{Re}(ue^{i\theta/2}) \geq 1 - \frac{\varepsilon^2}{2} \iff \mathbf{u} \cdot \mathbf{z} \geq 1 - \frac{\varepsilon^2}{2}, \quad (3.20)$$

which defines a halfspace. The other condition on $u \in \mathbb{C}$ is that it is within the unit circle in the complex plane, i.e. $|u| \leq 1$ or $|\mathbf{u}| \leq 1$.

Next, we tackle the second subproblem: identifying a unitary operator V that can be exactly synthesized from the Clifford+T gate set. To do so, we first define the rings

$$\begin{aligned} \mathbb{Z}[\omega] &= \{a\omega^3 + b\omega^2 + c\omega + d \mid a, b, c, d \in \mathbb{Z}\} \text{ and} \\ \mathbb{Z}[\frac{1}{\sqrt{2}}, i] &= \left\{ \frac{s}{\sqrt{2}^k} \mid k \in \mathbb{N}, s \in \mathbb{Z}[\omega] \right\}, \end{aligned} \quad (3.21)$$

with $\omega := e^{i\pi/4} = (1+i)/\sqrt{2}$. Any unitary with entries in $\mathbb{Z}[1/\sqrt{2}, i]$ can be exactly decomposed into a finite sequence of gates from the Clifford+T gate set [129]. Each value of k corresponds to a grid of points in the complex plane with spacing $1/\sqrt{2}^k$, see figure 3.2. The larger k , the finer the grid, and the more densely packed the candidates u are. Moreover, the gate count can be directly identified as k . Specifically, if an entry in V has the form $u = \hat{u}/\sqrt{2}^k$ for a $\hat{u} \in \mathbb{Z}[\omega]$, then the gate count is k , i.e. $V = g_1 \dots g_k$ with $g_1, \dots, g_k \in G$ [17, 126, 127, 130]. Efficient search techniques, including numerical optimization [127], can identify grid points in the intersection of this ring with the valid region defined by the approximation constraint depicted as colored region in figure 3.1. The minimal such k corresponds to the optimal T-count for the approximation. However, most methods propose a lower bound on k to guarantee (at least with high probability) the existence of a feasible candidate u [17, 126, 127].

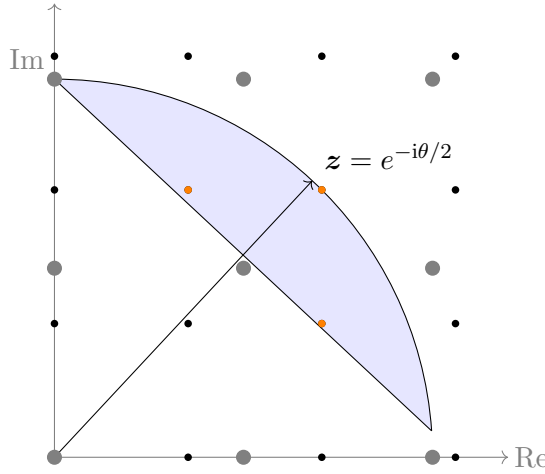


Figure 3.2: Example representation of candidates $u \in \mathbb{Z}[\frac{1}{\sqrt{2}}, i]$, exactly decomposable by a gate sequence of length k (large grey dots) and by a gate sequence of length $k+1$ (small black dots). The three feasible candidates (small orange dots) also fulfill the approximation criterion in lemma 7.

Once a feasible $u = \hat{u}/\sqrt{2}^k$ is identified, the corresponding v must satisfy the normalization condition $u^*u + v^*v = 1$ and lie in the same ring. This corresponds to solving the Diophantine equation in $\mathbb{Z}[\omega]$

$$\hat{v}^*\hat{v} = 2^k - \hat{u}^*\hat{u}, \quad (3.22)$$

with $\hat{u}, \hat{v} \in \mathbb{Z}[\omega]$. The Diophantine equation in $\mathbb{Z}[\omega]$ can be solved efficiently by a randomized algorithm using techniques from number theory [17, 126]. Once both u and v are determined, the matrix V is fully defined.

Finally, the last subproblem is to decompose the unitary V . The goal of exact synthesis is to determine a gate sequence $g_1 \dots g_k$, with $g_i \in G$, such that $V = g_1 \dots g_k$ [127, 129]. The procedure is specific to the chosen gate set but follows the systematic rules:

1. Select a gate g from the gate set such that $g^\dagger V$ has a lower gate count k than V .
2. Set $V \leftarrow g^\dagger V$.
3. Repeat the two previous steps until the gate count of V is zero.

When the gate count is zero, this means the remaining gate is a single-qubit Clifford gate. To find a suitable gate g to reduce the gate count $k \leftarrow k - 1$, one can simply try each gate from the small gate set.

This concludes the decomposition process: starting from an arbitrary target unitary, the algorithm approximates it to within precision ε , constructs a rational approximation in an algebraic ring, and finally translates it into an explicit sequence of gates from the Clifford+T gate set. Moreover, these single-qubit decomposition techniques achieve optimal T-counts, which is important for scalable quantum circuit synthesis.

3.2 Quantum circuit optimization

After decomposition, the resulting circuit can often be further optimized to reduce gate count, depth or increase fidelity. Hardware-aware optimization rewrites circuits to favor gates and patterns that are more efficiently implemented on the specific target device, such as converting sequences of CX gates to CZ gates when these are natively supported with higher fidelity. Moreover, a quantum circuit might also optimize to circumvent the limitations by the hardware architecture, such as limited connectivity. These transformations not only reduce the size of a quantum circuit but are also essential for minimizing the effects of noise, decoherence, and crosstalk on current NISQ devices [131].

Moreover, quantum circuits frequently contain substructures that can be simplified via circuit identities, commutation rules, or template matching [132–135]. These transformations help reduce redundancy, combine gates, and eliminate unnecessary operations. One common optimization involves gate cancellation and merging, where consecutive gates that cancel each other (for instance, $R_z(\theta) R_z(-\theta) = I$) are removed, or multiple gates are fused into a single, equivalent operation. Another method leverages commutation-based optimization, whereby gates are reordered to expose additional simplification opportunities or to increase parallelism, thereby reducing circuit depth. Peephole optimization, which examines small patterns of gates and replaces them with more efficient equivalents based on a library of known simplifications, is also used.

3.2.1 Quantum hardware and native operations

This section provides an abstract overview of some leading quantum computing platforms, with emphasis on their relevance to quantum compilation. Rather than detailing specific experimental realizations, the discussion centers on the native quantum operations, that each platform can directly implement. The most adopted and promising architectures include trapped ions, neutral atoms, and superconducting qubits. Each architecture exhibits a unique combination of capabilities and limitations, which influence both algorithm design and compiler strategies. In the following, we examine these three hardware platforms, highlighting their native gate sets, architectural constraints, and implications for quantum compilation.

Trapped ions

Trapped ion devices use ions confined in an electromagnetic trap, typically using linear Paul traps or surface-electrode traps. Qubits are encoded in internal electronic states of ions, such

as hyperfine Zeeman states. These systems benefit from high-fidelity gate operations and long coherence times, often on the order of seconds [136].

Single-qubit gates are implemented using microwave or laser pulses governed by the Rabi Hamiltonian

$$H_{\text{Rabi}}(\Omega, \phi) = \frac{\hbar\Omega}{2} (\cos(\phi)X + \sin(\phi)Y), \quad (3.23)$$

where Ω is the Rabi frequency, proportional to the pulse amplitude, and ϕ denotes the phase difference between the driving field and the qubit's Larmor precession. The native gates are arbitrary R_x ($\phi = 0$) and R_y ($\phi = \pi/2$) single-qubit rotations. The typical entangling operations are commonly realized using the Mølmer-Sørensen scheme [98], or via native Ising interactions yielding ZZ gates, introduced in section 2.3.2 [32, 137]. Both MS and ZZ gates can be mapped to standard CX or CZ gates via local rotations, making them useful for compilation.

A key advantage of trapped ion systems is their intrinsic all-to-all connectivity, allowing arbitrary pairs of qubits to be entangled without additional overheads. This simplifies the circuit mapping process, as discussed further in section 3.2.2. However, this flexibility comes at the cost of slower gate speeds, typically on the order of tens of microseconds, and increased susceptibility to noise such as crosstalk as system size grows [138].

From a compilation standpoint, the high gate fidelities and all-to-all connectivity enable efficient circuit layouts without qubit routing overhead. Nevertheless, optimizing pulse sequences remains critical to mitigate errors such as crosstalk [131]. Furthermore, scaling to larger numbers of qubits while preserving full connectivity remains challenging [139, 140].

Neutral atoms

Neutral atom platforms encode qubits in internal electronic states of atoms confined in optical tweezers. These architectures are promising due to their long coherence times and scalability [141, 142]. Additional advantages include potential all-to-all connectivity, support for native multi-qubit gates [143, 144], and the ability to rearrange atoms with high fidelity [145].

Single-qubit gates can be implemented with lasers, for ground-Rydberg state transitions [146], or microwaves, for Rydberg-Rydberg state transitions [147], that drive Rabi oscillations, similar as in eq. (3.23). Fast entangling gates rely on the Rydberg blockade effect, which prevents simultaneous excitation of nearby atoms to Rydberg states due to strong dipole-dipole interactions. Common entangling operations are typically arbitrary controlled single-qubit gates, which are used to implement CZ gates utilizing the Rydberg blockade mechanism [141]. Recent developments have also demonstrated the native implementation of fast multi-qubit gates that resemble many-body interactions [143, 144, 148]. While entangling gates operate on microsecond timescales, coherence during these operations is limited by the short lifetimes of the Rydberg states [141]. Nonetheless, computational basis states retain relatively long coherence times, and recent advances have improved entangling gate fidelities to levels comparable to those achieved in trapped ion and superconducting platforms [143, 144]. A key feature of neutral atom platforms is their reconfigurable connectivity, atoms can be dynamically rearranged to enable quasi all-to-all connectivity [145]. This property presents significant opportunities for compilation optimization.

Compilation for neutral atom systems involves optimizing gate placement to reduce interaction distances, managing limited parallelism due to Rydberg blockade constraints, and mitigating errors from atomic motion, dephasing, and Rydberg decay [149]. The architecture's tunable connectivity offers considerable flexibility but also requires careful coordination to fully leverage its potential.

Superconducting qubits

Superconducting qubits are fabricated using Josephson junctions, with quantum information encoded in the discrete energy levels of anharmonic oscillators. Among the most widely used qubit types are the transmon and flux qubits [35, 150]. These systems benefit from fast gate

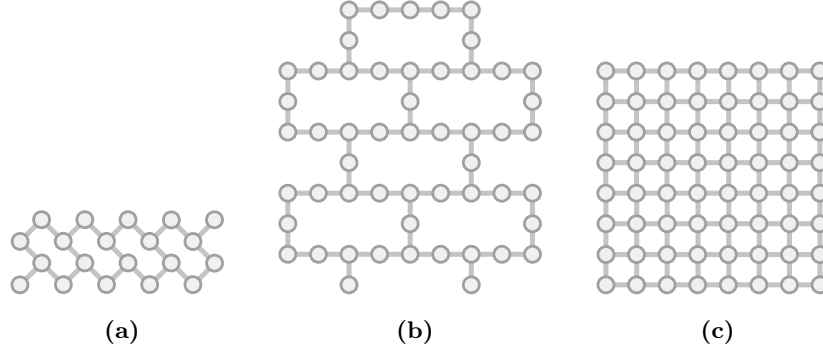


Figure 3.3: Example connectivity graphs of superconducting qubits for the (a) 20-qubit Rigetti Acorn [159], (b) 53-qubit `ibmq_rochester` [160] and (c) 8x8 grid architecture.³

times (on the order of tens of nanoseconds) and compatibility with CMOS-based fabrication techniques [151].

The native gates typically include arbitrary single-qubit rotations implemented via microwave pulses, again described by eq. (3.23). The choice of native entangling gates depends on the specific architecture. In tunable-frequency transmon arrays, the `iSWAP` or controlled-phase gates are directly implemented through parametric frequency modulation [152, 153], while architectures with fixed qubit frequencies support directly the cross-resonance (CR) gate [96, 154]. The `iSWAP` gate swaps the states of two qubits while also introducing a phase factor. It is defined as $\text{iSWAP}_{i,j} := e^{-i\frac{\pi}{4}(X_i X_j + Y_i Y_j)}$, i.e. the unitary operator generated by XY interactions, and serves as a building block for other entangling operations such as `CZ` or `CX` gates. However, the CR gate is more efficient, requiring only one CR gate and two single-qubit $\pi/2$ rotations to implement a `CX` gate [35].

Superconducting qubits are typically arranged in a 2D grid with nearest-neighbor coupling, which imposes significant limitations on connectivity, see figure 3.3. Consequently, implementing long-range entangling gates requires additional `SWAP` operations to bring qubits into proximity. Despite limited connectivity, superconducting platforms are among the most mature and widely deployed, with large-scale systems (e.g., 100+ qubits) already demonstrated [155, 156].

Compilation techniques must account for crosstalk, calibration drift, and device-specific fidelity variations across the chip [131, 157, 158]. Optimizations such as topology-aware qubit mapping are essential for enabling the computational performance on superconducting platforms.

3.2.2 Qubit mapping and routing

In the execution of a quantum program on real hardware, a crucial step involves mapping the logical qubits, those defined by the high-level quantum algorithm, to the physical qubits available on the quantum processor, known as *qubit mapping*. However, the physical constraints of quantum hardware impose limitations: a physical qubit can typically interact directly only with a fixed set of neighboring qubits, as defined by the device’s coupling topology. As a result, the two-qubit operations prescribed by the high-level quantum algorithm can only be executed directly if the corresponding physical qubits are coupled. Consequently, a transformation of the circuit is required to make it executable on the given device. This transformation generally involves modifying the circuit so that two-qubit gates are only applied between physical qubits that are connected in the coupling graph. To achieve this, additional gates, typically `SWAP` gates, are inserted to bring the information stored in the qubits into the appropriate proximity, and the sequence of gates is adjusted accordingly. The resulting transformation preserves the functionality of the original circuit while ensuring compatibility with the device constraints.

This task, although entirely classical, is computationally challenging and has been proven to be NP-complete [117], which limits the scalability of optimal methods for large circuits.

Existing techniques for solving this problem fall into two categories. The first consists of exact methods, which aim to find an optimal solution via a mathematical formulation of the problem, such as an integer linear program or boolean satisfiability problem, and solving it using classical optimization techniques [162, 163]. These approaches yield minimal circuit depth and gate count but are generally restricted to small or moderately sized circuits due to their exponential runtime. Notably, recent advances have demonstrated the feasibility of optimal mapping for circuits with up to 100 qubits in certain configurations [164], improving in the practical applicability of these methods. The second category consists of heuristic approaches, which sacrifice optimality for scalability. One of the most widely adopted heuristic methods is SABRE (SWAP-based **B**idi**R**ectional routing) [165, 166], which globally optimizes the initial mapping using a reverse traverse technique. SABRE has become the de facto standard in many quantum compilers due to its efficiency and robustness. More recently, new heuristics, including divide-and-conquer strategies, have demonstrated improved performance on specific topologies [167].

The formal specification of the qubit mapping problem begins with two inputs. The first is a quantum circuit composed of a sequence of L gates, denoted as g_1, g_2, \dots, g_L , acting on M logical qubits labeled q_1, q_2, \dots, q_M . These gates are assumed to be either single-qubit or two-qubit operations and are native to the target hardware platform, although they might not satisfy the connectivity constraints. The second input is the coupling graph of the quantum device, defined as (V, E) , where the vertex set $V = \{v_1, v_2, \dots, v_N\}$ corresponds to the physical qubits, and the edge set $E = \{e_1, e_2, \dots, e_K\}$ specifies the allowable two-qubit interactions between them, as depicted in figure 3.3. The fidelity of a circuit can also be optimized by considering a weighted graph, where each weight represents the fidelity and incorporating these weights in the objective function of the optimization. The output of the qubit mapping procedure is a transformed circuit composed of L' gates acting on the N physical qubits, where $L' \geq L$ due to the possible insertion of routing operations such as SWAP gates. This new circuit is functionally equivalent to the original but adheres to the connectivity constraints of the coupling graph. In addition, the procedure yields a final permutation that maps each logical qubit to the physical qubit from which its measurement outcome should be extracted. This ensures that, despite the intermediate transformations and remapping, the logical interpretation of the computation remains consistent.

Qubit mapping for the neutral atom platform Qubit mapping takes on a distinct character in neutral atom architectures. In this setting, qubits are encoded into the states of individual atoms confined in optical tweezers, and two-qubit interactions are mediated by proximity, atoms can only interact when brought sufficiently close to one another [149]. A powerful feature of neutral atom systems is the ability to dynamically reposition the optical tweezers during computation without disturbing the quantum state or entanglement between qubits [145].

Typically, atoms are initialized in a static array defined by a regular grid of optical traps generated by a spatial Light Modulator (SLM). During execution, these atoms can be rearranged using auxiliary traps produced by an acousto-Optic Deflector (AOD). This dynamic rearrangement, often referred to as shuttling, enables the physical implementation of qubit routing by relocating atoms rather than by applying SWAP operations as done in other architectures. While shuttling can be performed with high fidelity, it is considerably slower than native quantum gate operations. Consequently, minimizing the number and distance of atom movements is a key consideration in the neutral atom qubit mapping process. Moreover, shuttling via AODs introduces specific hardware-level constraints. Each AOD defines two orthogonal sets of control coordinates: $\{x_1, \dots, x_a\}$ along the horizontal axis and $\{y_1, \dots, y_b\}$ along the vertical. Each intersection (x_i, y_j) corresponds to a potential trapping site where an atom may be confined. Atom

³We used the connectivity graphs from Ref. [161].

movement is achieved by dynamically modifying the values of these coordinates, thereby relocating the associated optical traps. Importantly, changing a single coordinate affects all atoms along that axis simultaneously. For example, adjusting x_i shifts the entire i -th column of atoms in the same direction and by the same amount. This global behavior implies that individual atoms within a single AOD cannot be repositioned independently. A second key constraint arises from the physical limitations of the optical trapping system: adjacent coordinates within each axis must maintain a minimum separation distance. Violating this constraint can cause overlapping trap potentials, leading to undefined behavior or even atom loss. As a consequence, the coordinate sets must preserve their ordering, e.g., $x_1 < x_2 < \dots < x_a$, throughout the process, which prohibits coordinate lines from crossing. These constraints can be partially alleviated by incorporating multiple AODs. Multi-AOD systems offer greater flexibility by enabling overlapping or independent control regions, allowing for more complex and localized rearrangement strategies. However, this comes at the cost of increased system complexity.

In summary, qubit mapping plays a significant role in bridging the gap between abstract quantum algorithms and their physical realizations on constrained quantum hardware. As quantum processors scale and the connectivity topology in the architectures diversify, the efficiency of this compilation stage becomes increasingly important to overall performance.

3.2.3 Minimize redundant operations

Quantum circuits often contain structural redundancies that can be eliminated through a variety of optimization techniques. These include the application of circuit identities, commutation rules, and pattern-based simplifications such as template matching [132–135]. The primary objective of these transformations is to reduce the overall gate count and circuit depth, therefore minimize the number of noisy operations.

An optimization technique involves gate cancellation and merging, where adjacent gates that algebraically cancel each other are removed, or sequences of gates are fused into a single, functionally equivalent operation. For example, two consecutive Pauli X gates on the same qubit can be eliminated, and a series of rotation gates around the same axis may be merged into a single gate with the combined angle. Another strategy is commutation-based optimization, which reorders gates that commute in order to expose further cancellation opportunities, or enable parallel execution. This often contributes to reducing the circuit depth, as seen in eq. (3.7) [19]. Peephole optimization is a local transformation technique where small subcircuits are matched against a library of known patterns and replaced with more efficient ones. This technique can be applied iteratively and at multiple scales to uncover simplifications that may not be immediately apparent.

The success of these techniques relies on the choice of circuit representation, as different representations highlight different structural features of the quantum circuit. One such representation is the directed acyclic graph, where each vertex corresponds to a quantum gate and each edge represents a “qubit wire”. The direction of the edges encodes the temporal ordering of operations. The graph has to be acyclic since there are no closed timelike curves allowed. This structure allows for an easy check if adjacent gates cancel. Another representation is the sum-over-paths formalism, which is a representation for circuits over the non-universal gate set $\{CX, T\}$, which enables efficient classical simulation [168, 169]. In this framework, a unitary operation acting on an n qubit basis state $|\mathbf{x}\rangle$ is represented as

$$U |\mathbf{x}\rangle = e^{i2\pi f(\mathbf{x})} |A\mathbf{x}\rangle, \quad (3.24)$$

where $f : \mathbb{F}_2^n \rightarrow \mathbb{R}$ is called a *phase polynomial*, and $A \in GL(\mathbb{F}_2^n)$ is an invertible boolean matrix. This representation separates the linear component A , representing CX circuits, from the phase polynomial, which can represent arbitrary diagonal unitary operations [32]. This enables algebraic manipulations and optimizations, especially for Clifford+T circuits. A third representation is the ZX-calculus, a category-theoretic and graphical language for quantum computation [170].

In the ZX formalism, quantum processes are represented as diagrams composed of interconnected dots of X- or Z-type, sometimes called “spiders”, with well-defined rewrite rules. ZX-diagrams provide a complete language for reasoning about quantum operations, making structural transformations both intuitive and algebraically rigorous. This formalism enables simplifications via diagrammatic rewrites, many of which are visually apparent but not easily seen in gate-based or matrix-based representations. However, the expressiveness of the ZX-calculus introduces a significant challenge: not all diagrammatic transformations preserve unitarity. This necessitates a non-trivial process known as circuit extraction, whereby a valid quantum circuit is recovered from the optimized diagram. While recent work has made progress in this area, circuit extraction remains computationally hard in the general case [105, 171].

Quantum circuit optimization can be distinct into two primary classes of compilers. The first consists of manually designed optimizers that are based on rewrite rules tailored to specific gate sets and hardware architectures. These compilers are often inspired by traditional compiler design and resemble the use of gate identities. Such optimizations are typically fast, but are limited to the known circuit transformations. The second class of compilers adopts an automated approach. These compilers aim to generate and verify circuit transformations automatically, reducing the need for hand-crafted rules [134]. They regularly employ methods, such as semantics-preserving rewrite rules and symbolic reasoning, to ensure correctness while exploring a broader space of transformations. Recent tools use probabilistic verification techniques to determine equivalence between quantum circuits by transforming them into canonical forms [135]. Furthermore, these compilers have been developed to automatically construct an optimizer tailored to the native gate set of a given quantum device.

Minimizing redundant and potentially noisy operations has direct implications for reducing execution time and improving fidelity on quantum hardware. By exploiting structural features exposed by various circuit representations, these techniques enable the systematic elimination of unnecessary gates and the compression of circuit depth. Future work might focus on unifying these approaches within flexible, device-aware compilers that can dynamically adapt optimization techniques to the characteristics and constraints of specific quantum platforms [172].

3.3 Outlook and Challenges

As quantum computing continues to evolve, the importance of quantum compilation is poised to grow correspondingly. The increasing sophistication of quantum algorithms, coupled with ongoing advancements in hardware platforms, necessitates the development of more refined and efficient compilation strategies. While certain aspects of quantum compilation, such as single-qubit compilation with minimal T-count, seems to be matured, other areas remain at a nascent stage. In particular, compilation techniques for emerging architectures like the neutral atom platform have recently begun to attract more research focus.

With the field progressing toward scalable fault-tolerant quantum computing, the compilation process must adapt to accommodate the complexities of error-corrected quantum architectures. Circuits must be mapped to logical qubits protected by quantum error-correcting codes, and logical operations must be compiled into fault-tolerant gate sequences, often involving costly processes such as magic state distillation. As a result, the compilation stack must combine additional layers for error correction scheduling, logical gate synthesis, and optimization under hardware constraints. Moreover, as error correction schemes are sensitive to different noise models, incorporating noise-aware compilation into the workflow is beneficial [173]. A promising avenue for future research is the development of application-specific fault-tolerant compilation techniques. For instance, quantum algorithms with potential practical relevance, such as quantum simulation, optimization, and the quantum Fourier transform (QFT), could benefit from tailored compilation strategies that exploit their structural properties [174, 175].

Another challenge stems from the growing diversity of quantum hardware platforms, which makes the need for architecture-aware compilation increasingly apparent. In this context, the

neutral atom platform has recently gained interest, driven by its remarkable progress in demonstrating small-scale fault-tolerant protocols and implementing fast, high-fidelity multi-qubit gates [10]. These advances open up new possibilities for quantum compilation, including the exploitation of reconfigurable interactions, and parallelism unique to this platform.

In summary, the road ahead for quantum compilation must navigate a rapidly changing landscape shaped by technological breakthroughs and shifting algorithmic frontiers. Addressing these challenges will require a confluence of insights from quantum information theory, computer architecture, classical compilation, and experimental physics.

Chapter 4

Hamiltonian engineering

Hamiltonian engineering refers to the design and control of quantum dynamics such that a physical system governed by a fixed system Hamiltonian can effectively simulate the evolution of a desired *target Hamiltonian*. This is a central objective in quantum simulation [176–179] and quantum computation [144, 180, 181], where physical interactions are constrained by hardware limitations, but the desired dynamics often require more flexible or non-native interactions. In quantum simulations, it enables the realization of complex spin models beyond the native interactions, and in digital quantum computations, it allows for efficient gate decomposition strategies, see section 3.1.1. The core idea is to exploit a combination of the system’s natural evolution, determined by its *native* Hamiltonian, and applied control pulses, most commonly single-qubit pulses. Through appropriate design of pulse sequences, the joint effect can be tailored to approximate the desired unitary evolution. This concept has its roots in the nuclear magnetic resonance (NMR) community, where early work laid the foundation for modern Hamiltonian engineering approaches [22, 31].

However, such modern approaches vary significantly in the assumptions they make about available control resources. At a minimum, it is typically assumed that the system evolves naturally under a fixed system Hamiltonian, and that certain simple single-qubit pulses can be applied during the evolution. One key distinction between methods lies in the choice of single-qubit pulses. Some methods assume global control, i.e. an identical single-qubit pulse applied to all qubits [182–184]. This assumption simplifies experimental implementation, particularly in platforms like neutral atoms or cold atom clouds with limited addressability [148, 185–187]. On the other hand, assuming local control, i.e. independent single-qubit pulses on each qubit, opens the door to more flexible Hamiltonian engineering methods [29–34, 188–191]. Some frameworks further assume the ability to switch the system Hamiltonian on and off instantaneously at arbitrary times [189, 191]. However, this level of control is not available in current experimental platforms, limiting the accuracy of such methods. Moreover, previous methods relying on local control suffer from poor scalability. In particular, the number of single-qubit pulses required to implement the target evolution often scale poorly with system size, rendering them impractical for near-term devices [29–31, 188, 192]. Furthermore, on current NISQ devices, gate fidelities remain limited, and control pulses are subject to systematic errors such as over-rotation, off-resonance detuning, and finite pulse time effects [9, 193, 194]. These imperfections can significantly degrade the fidelity of engineered evolutions, especially when long sequences are required [195].

In sections 4.2 and 4.3, we give an overview of our work done on the development of time-optimal, efficient and robust Hamiltonian engineering strategies respectively, methods that not only approximate the target evolution in an idealized setting, but also tolerate common error sources and require minimal implementation time. In particular, these sections are based on our publications [32, 33] about time-optimal protocols, and our work [34] about general, robust, and efficient methods.

4.1 A brief history of Hamiltonian engineering

As mentioned before, the idea of Hamiltonian engineering, has deep roots in the field of NMR. Nearly twenty-five years ago, the work in the NMR community laid the theoretical foundation for modern Hamiltonian engineering techniques. Early contributions demonstrated that the time evolution generated by an arbitrary two-local Hamiltonian can be simulated using a different two-local Hamiltonian in conjunction with properly chosen local control pulses [29, 30]. These results were important in proving that the interplay between natural system dynamics and local operations can yield effective interactions that are otherwise not directly available on the physical platform. At about the same time, researchers aimed to develop π pulse (Pauli operator) sequences that could decouple all interactions in a two-local Hamiltonian except for one, a technique referred to as *selective coupling* [31, 188]. The central idea was to interleave the natural time evolution of the system with layers of π pulses, such that undesired interaction terms were effectively canceled out via sign flips, while selected terms accumulated coherently. Specifically, π pulses that anti-commute with a given interaction term flip its sign, and averaging over multiple such flips cancels the term. It was recognized that these sign-flipping patterns follow the combinatorial structure of Hadamard matrices, orthogonal matrices with entries in $\{+1, -1\}$, which enabled a systematic construction of decoupling sequences [188]. However, these early schemes suffered from scalability challenges. In a system of n qubits with $r = O(n^2)$ possible two-local interaction terms, their construction required up to r layers of control pulses to engineer a single desired interaction. This introduces a quadratic overhead for engineering all interaction terms, yielding a total of up to r^2 pulse layers. Moreover, when the system Hamiltonian contains non-commuting terms (as is typical in many-body physics), the resulting sequence must be further broken into short-time product formula (Trotter) approximations.

In 2014, an alternative approach was proposed based on linear programming, which aimed to efficiently determine pulse sequences for Hamiltonian engineering [189]. This method reformulated the engineering task as a linear program, searching for sign-flip sequences that produce the desired effective Hamiltonian. While promising in its efficiency, this method was limited to specific classes of system Hamiltonians, primarily one-dimensional spin chains, and could not be generalized easily.

A renewed interest of research around 2020 focused on Hamiltonian engineering using *global control*, where the same single-qubit gate is applied uniformly across all qubits [182–184]. Though these methods cannot selectively manipulate individual interaction terms, they have proven highly effective in reshaping the global dynamical behavior of the system. Importantly, these techniques have been successfully demonstrated in several experimental settings [187, 196], and their robustness against certain classes of errors (e.g., over-rotations and detuning) further enhances their appeal for near-term experiments.

At the same time, a paradigm known as digital-analog quantum computing (DAQC) was introduced [190]. This approach attempts to engineer an Ising Hamiltonian by determining appropriate π pulse sequences through the solution of a linear equation system, which takes into account the sign flips induced by the pulses. However, DAQC suffers from a critical drawback: the solutions to the linear system often involve *negative evolution times* under the system Hamiltonian.¹ Since negative time evolution cannot be directly implemented, this presents a major obstacle. Despite this limitation, the DAQC approach was further developed in follow-up work, which generalized the framework to engineer arbitrary two-local Hamiltonians [191]. In order to circumvent the problem of negative evolution times, this extended method introduces a classical preprocessing step that involves solving an NP-hard optimization problem to determine the pulse sequences and associated evolution times.

¹In ref. [190, App. A], they try to resolve this issue. But only solved it for qubit numbers $n \in \{3, 5, 6\}$.

4.2 Time-optimal Ising Hamiltonian engineering

In this section, we present our Hamiltonian engineering scheme based on linear programming for modifying arbitrary Ising Hamiltonians with minimal evolution time. It is based on the publications [32, 33], which are listed in appendices A and B. In section 2.3.2, we defined the GZZ(A) gate as the time evolution under an Ising Hamiltonian determined by a symmetric coupling matrix $A \in \mathbb{R}^{n \times n}$. In section 3.1.1 we highlighted the role of GZZ gates in the decomposition of Clifford unitaries. More broadly, implementing the time evolution under a desired Ising Hamiltonian is a key component in various quantum compilation strategies [32, 197–199]. The ability to engineer arbitrary interaction strengths in Ising models is therefore essential, especially given the prominence of Ising-type couplings in quantum hardware platforms such as ion traps [38–40] and superconducting qubits [35, 36].

First, we derive the LP to find the time-optimal single-qubit pulse sequence. Second, we discuss the bounds on the total evolution time required to implement a GZZ gate. Then, we show that solving the time-optimal LP is NP-hard and provide an efficient heuristic. Finally, we present applications of our scheme in quantum compilation methods and quantum algorithms.

4.2.1 Conjugation of Ising Hamiltonians

We consider the Ising Hamiltonian as the native *system Hamiltonian*, defined on n qubits as

$$H_S := - \sum_{i < j}^n J_{ij} Z_i Z_j, \quad (4.1)$$

where $J_{ij} \in \mathbb{R}$ represents the coupling strength between qubits i and j which we collect in a symmetric *coupling matrix* $J \in \mathbb{R}^{n \times n}$, and Z_i denotes the Pauli-Z operator acting on qubit i . We assume to have access to the time evolution under H_S , i.e. the unitary e^{-itH_S} can be implemented natively. To express single-qubit operators, we use the shorthand notation $P_i := I^{\otimes i-1} \otimes P \otimes I^{\otimes n-i}$ for a unitary operator $P \in \text{U}(2)$ acting non-trivially only on the i -th qubit. A Pauli-X gate (also known as a π pulse around the x -axes or just π pulse) is implemented as the unitary operator $e^{-i\frac{\pi}{2}X}$, where X is the generator. A layer of such gates applied in parallel to a subset of qubits is denoted by

$$X^{\mathbf{b}} := e^{-i \sum_i^n (\frac{\pi}{2} X_i)^{b_i}}, \quad (4.2)$$

where $\mathbf{b} = (b_1, \dots, b_n) \in \{0, 1\}$ is a binary vector indicating on which qubits the Pauli-X gate is applied ($b_i = 1$) and where it is skipped ($b_i = 0$). Conjugating the time evolution under H_S by such a layer of Pauli-X gates yields

$$X^{\mathbf{b}} e^{-itH_S} X^{\mathbf{b}} = e^{-itX^{\mathbf{b}} H_S X^{\mathbf{b}}}, \quad (4.3)$$

using the identity $U^\dagger e^{-itH} U = e^{-itU^\dagger H U}$, valid for arbitrary unitary operators U , and the fact that Pauli operators are self-inverse. To evaluate the effect of conjugation more explicitly, we compute

$$\begin{aligned} X^{\mathbf{b}} H_S X^{\mathbf{b}} &= - \sum_{i < j}^n J_{ij} X^{b_i} X^{b_j} (Z_i Z_j) X^{b_i} X^{b_j} \\ &= - \sum_{i < j}^n J_{ij} (-1)^{b_i} (-1)^{b_j} Z_i Z_j \\ &= - \sum_{i < j}^n J_{ij} m_i m_j Z_i Z_j =: H(\mathbf{m}), \end{aligned} \quad (4.4)$$

where we used the anti-commutation relation $XZ = -ZX$, and defined the *encoding* $\mathbf{m} := (-1)^{\mathbf{b}}$ entrywise. Each element $m_i = (-1)^{b_i}$ takes values ± 1 , indicating whether a Pauli-X gate has

flipped the sign of the Z_i operator. We refer to $H(\mathbf{m})$ as the *encoded Hamiltonian*, which differs from the original Hamiltonian H_S by a sign pattern applied to the interaction terms. The modified coupling matrix of $H(\mathbf{m})$ can be written as $J \odot \mathbf{m}\mathbf{m}^T$, where \odot denotes the Hadamard (entry-wise) product, and $\mathbf{m}\mathbf{m}^T$ is the outer product of the encoding vector with itself. The original coupling matrix J is symmetric, and the diagonal entries are zero, which is preserved under the encoding. This scheme allows us to invert the signs of a subset of interaction terms in the Ising Hamiltonian, enabling Hamiltonian engineering via local Pauli-X gates.

4.2.2 The linear program approach

We now describe how encoded Hamiltonians $H(\mathbf{m})$ can be employed to simulate the time evolution under an arbitrary target Ising Hamiltonian. The core idea is to express the desired evolution as a linear combination of evolutions under the encoded Hamiltonians $H(\mathbf{m})$, which can be generated via conjugation by layers of local Pauli-X gates. Let $\lambda_{\mathbf{m}} > 0$ be the *relative evolution time* associated with the encoded Hamiltonian $H(\mathbf{m})$. We consider unitary evolutions of the form

$$\prod_{\mathbf{m}} e^{-it\lambda_{\mathbf{m}}H(\mathbf{m})} = e^{-it\sum_{\mathbf{m}}\lambda_{\mathbf{m}}H(\mathbf{m})} = e^{-itH}, \quad (4.5)$$

where we have used the fact that all encoded Hamiltonians $H(\mathbf{m})$ mutually commute. This holds because they involve only $Z_i Z_j$ terms and are therefore diagonal in the computational basis. Due to the symmetry $(-\mathbf{m})(-\mathbf{m})^T = \mathbf{m}\mathbf{m}^T$, we only have to consider half of the encodings, reducing the total number of encodings from 2^n to 2^{n-1} . We collect all relative evolution times $\lambda_{\mathbf{m}}$ in a vector $\boldsymbol{\lambda} \in \mathbb{R}^{2^{n-1}}$. This aggregates in the time evolution under the effective Hamiltonian $H = -\sum_{i<j}^n A_{ij} Z_i Z_j$ with the modified coupling matrix $A \in \mathbb{R}^{n \times n}$ given by

$$A := J \odot \sum_{\mathbf{m}} \lambda_{\mathbf{m}} \mathbf{m}\mathbf{m}^T. \quad (4.6)$$

The sum accumulates the contributions of all encoded evolutions, each scaled by its respective time coefficient $\lambda_{\mathbf{m}}$. By construction, A is a symmetric matrix with zero diagonal, since these properties are inherited from the original coupling matrix J . To formalize this, we define the space of such matrices as the $\binom{n}{2}$ -dimensional subspace of symmetric real $n \times n$ matrices with vanishing diagonal entries

$$\text{Sym}_0(\mathbb{R}^n) := \{M \in \text{Sym}(\mathbb{R}^n) \mid M_{ii} = 0 \ \forall i \in [n]\}. \quad (4.7)$$

Let $\mathbf{v} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{\binom{n}{2}}$ denote the row-wise vectorization of the upper triangular part of a matrix. We now define the matrix

$$V \in \{-1, +1\}^{\binom{n}{2} \times (2^{n-1})}, \quad (4.8)$$

whose columns are given by $\mathbf{v}(\mathbf{m}\mathbf{m}^T)$ for all 2^{n-1} relevant encodings \mathbf{m} . Each column thus represents how a particular encoding contributes to sign flips in the pairwise coupling terms. To describe only the non-zero coupling terms present in the native Hamiltonian, we define the index set

$$\text{nz}(J) := \{(i, j) \mid J_{ij} \neq 0, i < j\}. \quad (4.9)$$

Using eq. (4.6) and definition (4.8), we formulate the LP that finds the time-optimal decomposition of a target Hamiltonian in terms of encoded evolutions

$$\begin{aligned} & \text{minimize} \quad \mathbf{1}^T \boldsymbol{\lambda} \\ & \text{subject to} \quad V\boldsymbol{\lambda} = \mathbf{v}(M), \ \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^{2^{n-1}}, \end{aligned} \quad (\text{IsingLP})$$

where the matrix $M \in \text{Sym}_0(\mathbb{R}^n)$ is defined as the element-wise division

$$M := A \oslash J, \quad \text{with } (A \oslash J)_{ij} := \begin{cases} A_{ij}/J_{ij}, & \text{if } (i, j) \in \text{nz}(J) \\ 0, & \text{otherwise.} \end{cases} \quad (4.10)$$

The (IsingLP) finds the set of relative evolution times λ that reproduce the desired coupling matrix A , while minimizing the total evolution time.

4.2.3 Bounds on the minimal evolution time

The scaling of the total evolution time required to implement the target Ising evolution is given by the sum of the relative evolution times $\mathbf{1}^T \lambda^* = \sum_{\mathbf{m}} \lambda_{\mathbf{m}}$, where λ^* denotes an optimal solution to the (IsingLP). This total time directly determines the duration of the corresponding GZZ gate, and hence has implications for the overall runtime of quantum circuits containing GZZ gates. Minimizing the gate time is especially critical for NISQ devices, where coherence time impose limitations on the allowable circuit depth [9]. Understanding the bounds on the total evolution time is therefore essential for designing implementable quantum algorithms and compilation strategies.

Theorem 8 (Ref. [33]). *The optimal total gate time of $\text{GZZ}(A)$ with $A \in \text{Sym}_0(\mathbb{R}^n)$ is lower and upper bounded by*

$$\|M\|_{\ell_\infty} \leq \mathbf{1}^T \lambda^* \leq \|M\|_{\ell_1}, \quad (4.11)$$

where $M := A \oslash J$. Equality holds for the lower bound for all matrices $M = C\mathbf{m}\mathbf{m}^T$ for any $\mathbf{m} \in \{-1, +1\}^n$ and $C \geq 0$.

The lower bound $\|M\|_{\ell_\infty}$ is determined by the largest difference between the original and target coupling matrix with $J_{ij} \ll A_{ij}$, while the upper bound $\|M\|_{\ell_1}$ reflects the total magnitude of all entries in M . However, the upper bound can be quite loose, particularly for dense matrices M , and it scales as the number of entries, i.e. $O(n^2)$, in the worst case. This quadratic scaling is often pessimistic and does not reflect the typical instances encountered in practice. Based on numerical results and analytical constructions, we propose a tighter upper bound that scales linearly with the system size:

Conjecture 9 (Ref. [33]). *The optimal gate time of $\text{GZZ}(A)$ with $A \in \text{Sym}_0(\mathbb{R}^n)$ is tightly upper bounded by*

$$\mathbf{1}^T \lambda^* \leq \|M\|_{\ell_\infty} \cdot \begin{cases} n, & \text{for odd } n, \\ n-1, & \text{for even } n. \end{cases} \quad (4.12)$$

This conjectured bound is supported by instances where equality is achieved. Specifically, we proofed, that when $M = C\mathbf{m}\mathbf{m}^T$ with $C \leq 0$, then the upper bound is exactly attained. Such cases mirror the structure of the lower-bound matrices, which also take the form $M = C\mathbf{m}\mathbf{m}^T$, but with $C \geq 0$. These constructions highlight a potential symmetry in the extremal cases depending on the sign of the scalar prefactor C . We now briefly outline the proof strategy for these extremal cases for more details we refer to [33]. Consider the instance $M = -\mathbf{m}\mathbf{m}^T$ with \mathbf{m} set to be the all-ones vector $\mathbf{1}$. In our work, we first explicitly construct feasible solutions for both the primal problem (IsingLP) and its dual (see section 2.2.2). Next, we show that these solutions yield the same objective value, it follows by strong duality that both solutions are optimal [54]. This approach can then be extended to matrices of the form $M = C\mathbf{m}\mathbf{m}^T$ for any $C \leq 0$, establishing that the conjectured upper bound is tight in these cases.

The conjectured linear upper bound, if proven in full generality, would imply that any target Ising coupling matrix can be synthesized with a time overhead that grows at most linearly with the number of qubits, regardless of the complexity of the target interaction pattern. This would mark an important insight in the total runtime of quantum algorithms and scalable quantum compilation strategies, as it directly impacts the feasibility of implementing large-scale Clifford circuits and more general quantum algorithms, see section 4.2.5.

4.2.4 Complexity and efficient heuristic

As quantum devices continue to scale, increasing the number of qubits demands faster and more efficient algorithms for synthesizing GZZ gates. However, a major computational bottleneck arises from the fact that the size of the (IsingLP) grows exponentially with the number of qubits n , due to the 2^{n-1} possible encodings \mathbf{m} . In fact, we showed that solving the decision version of (IsingLP) is NP-complete [33]. The proof leverages a connection to graph theory, specifically the problem of finding cuts in weighted graphs. This correspondence allows one to encode the optimization over Ising couplings as a combinatorial decision problem, highlighting the inherent hardness of the task.

To address this challenge, we develop an efficient heuristic method that circumvents the exponential blow-up by restricting the set of encodings \mathbf{m} used in the (IsingLP). The key idea is to construct a matrix basis that is guaranteed to span the space of all symmetric coupling matrices with zero-valued diagonals $\text{Sym}_0(\mathbb{R}^n)$. In the following, we briefly mention the steps of constructing such a basis. We begin by noting that the sum over outer products can be expressed in terms of a matrix multiplication

$$\sum_{k=1}^d \mathbf{m}_k \mathbf{m}_k^T = H^T H, \quad (4.13)$$

where $H := (\mathbf{m}_1, \dots, \mathbf{m}_d)^T \in \{-1, +1\}^{d \times n}$ is a matrix whose rows are the encoding vectors. Note, that we can realize arbitrary encoding vectors $\mathbf{m} \in \{-1, +1\}^n$ by appropriate positions of the Pauli-X gates. If the matrix H consists of n distinct columns taken from a $d \times d$ Hadamard matrix, then the orthogonality of the rows/columns of a Hadamard matrix ensures that

$$H^T H = d \mathbb{1}_n. \quad (4.14)$$

For each index pair $i, j \in [n]$ with $i < j$, we define a matrix $H_{(i,j)} \in \{-1, +1\}^{d \times n}$ by replacing the i -th column with the j -th column (or vice versa) in H . The resulting matrix satisfies

$$H_{(i,j)}^T H_{(i,j)} = d e_{(i,j)}, \quad (4.15)$$

where $e_{(i,j)} := \mathbb{1}_n + E_{ij} + E_{ij}^T$, and E_{ij} is the standard matrix basis element with a 1 in position (i, j) and zeros elsewhere. By taking a conic combination of such matrices $H_{(i,j)}^T H_{(i,j)}$, we can exactly reach any target matrix $M \in \text{Sym}_0(\mathbb{R}^n)$ with non-negative entries and without the diagonal. We therefore define the restricted set of encodings on n qubits as

$$\mathcal{E}_n^{(2)} := \{\mathbf{m} \mathbf{m}^T \mid \mathbf{m} \in \text{rows}(H_{(i,j)}), \forall i, j \in [n], i < j\}, \quad (4.16)$$

where the superscript (2) refers to the fact that two columns in each matrix $H_{(i,j)}$ are identical. This construction can be generalized. For $k \geq 2$, we define matrices $H_{(i_1, \dots, i_k)}$ obtained from a $d \times d$ dimensional Hadamard matrix in which k selected columns are identical, and use these to build the extended set

$$\mathcal{E}_n^{(k)} := \mathcal{E}_n^{(k-1)} \cup \{\mathbf{m} \mathbf{m}^T \mid \mathbf{m} \in \text{rows}(H_{(i_1, \dots, i_k)}), \forall i_1, \dots, i_k \in [n], i_1 < \dots < i_k\}. \quad (4.17)$$

This yields a hierarchy of sets of increasing size but still tractable subsets of encodings. Correspondingly, we define the restricted LP's

$$\begin{aligned} & \text{minimize} \quad \mathbf{1}^T \boldsymbol{\lambda} \\ & \text{subject to} \quad V^{(k)} \boldsymbol{\lambda} = \mathbf{v}(M), \quad \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^h, \end{aligned} \quad (\text{restrLP})$$

where $V^{(k)}$ contains the vectorized outer products from $\mathcal{E}_n^{(k)}$, similar as in eq. (4.8), and the number of considered encodings $h = |\mathcal{E}_n^{(k)}|$ is polynomial for a fixed k . Even though we previously

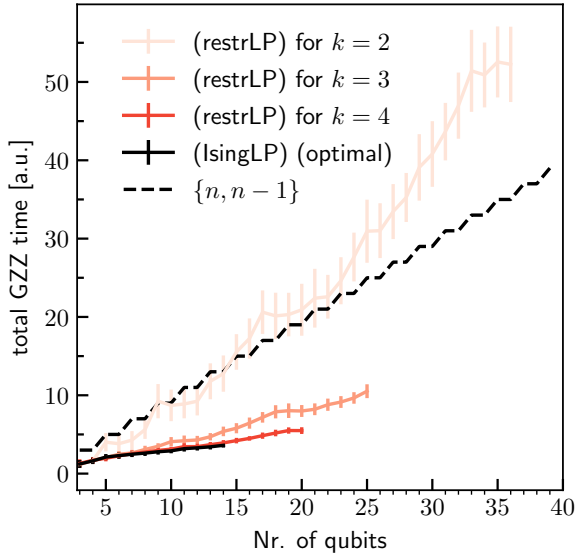


Figure 4.1: Comparing the performance of the original (optimal) (IsingLP) and the (restrLP) for $k = 2, 3, 4$. For each line we let the LP’s run for a fixed time. The black dashed line is the upper bound for the original (IsingLP) from conjecture 9. The reddish lines show the total GZZ times for the (restrLP) for $k = 2, 3, 4$. The data points and error bars show the mean and the standard deviation over 100 uniformly sampled matrices $M_{ij} = M_{ji} \in [-1, 1]$.

showed that (restrLP) with $\mathcal{E}_n^{(2)}$ guarantees feasibility only for non-negative matrices M , we have been proven that the (restrLP) remains feasible for any matrix $M \in \text{Sym}_0(\mathbb{R}^n)$ [33]. This is an essential result, as it shows that despite the exponential nature of the full (IsingLP), efficient heuristics exist that capture all Ising Hamiltonians. Note, that the heuristic still yields an exact decomposition of the target evolution, however the evolution time is no longer time-optimal.

This encoding hierarchy provides a trade-off between the classical runtime required to solve the (restrLP) and the quality of the resulting solutions in terms of evolution time. As illustrated in figure 4.1, increasing the parameter k leads to solutions that rapidly converge toward the optimal evolution time. This demonstrates that even low levels of the hierarchy can yield near-optimal results with significantly reduced computational effort compared to solving the full (IsingLP). It is important to emphasize that, despite the suboptimality in evolution time at lower levels of the hierarchy, all solutions obtained from (restrLP) still correspond to exact decompositions of the target unitary. In other words, the approximation lies solely in the duration of the evolution, not in the fidelity of the implemented operation.

4.2.5 Applications

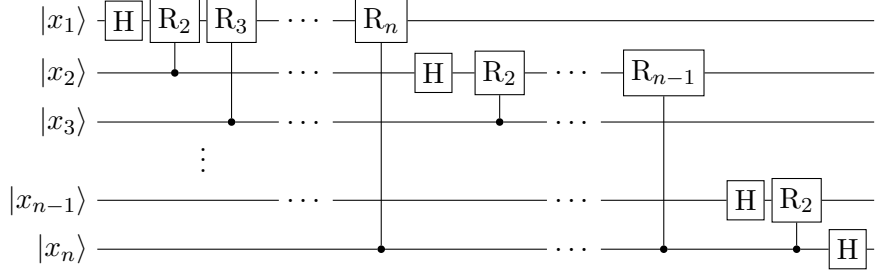
The control and implementation of Ising-type evolutions are not only crucial to quantum compilation strategies [32, 197–199], but also play a central role in several key quantum algorithms. These include variational quantum algorithm (VQA) [123] such as the quantum approximate optimization algorithm (QAOA) [200], implementations of the quantum Fourier transform (QFT) [201, 202], and even methods in quantum chemistry [203]. In this section, we provide an overview of how GZZ gates might be used to implement the latter two applications, following the methodology of our work [32].

The QFT is the quantum analog of the classical discrete Fourier transform, and it maps quantum states from the computational basis into the frequency domain. It plays a pivotal role in many quantum algorithms, most notably as a subroutine of QPE, which in turn is a key component of Shor’s factoring algorithm for efficiently finding the prime factors of an integer on a quantum computer [204]. Beyond factoring, quantum phase estimation (QPE) is also applied in quantum chemistry, particularly for estimating eigenvalues of molecular Hamiltonians [205, 206]. To efficiently implement the QFT using GZZ gates, we start by expressing computational basis states $|x_1, \dots, x_n\rangle$, where $x_j \in \mathbb{F}_2$, as binary-encoded integers, i.e. $x = \sum_{j=1}^n x_j 2^{n-j}$. The action

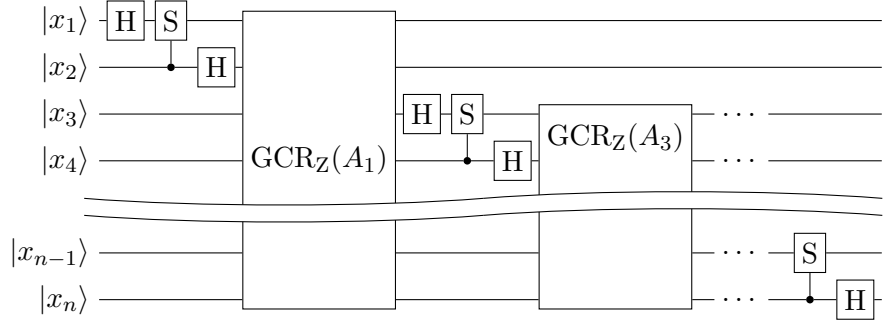
of the QFT on these states can be written as

$$QFT|x\rangle = \frac{1}{2^{n/2}} \bigotimes_{j=1}^n \left(|0\rangle + e^{i2\pi x 2^{-j}} |1\rangle \right). \quad (4.18)$$

This operation is typically realized through a sequence of controlled phase rotations $R_j := R_z(2\pi/2^j)$ interleaved by Hadamard gates, as depicted in the standard QFT circuit diagram:



where the output qubits are in reversed order. Many of the controlled rotations can be commuted past the Hadamard gates and grouped into more efficient multi-qubit gates such as global controlled- R_z gates GCR_z . This yields the compiled QFT circuit



Each GCR_z gate can be implemented using a single GZZ gate and additional single-qubit operations, see eq. (2.28). This transformation reduces the total number of GZZ gates required to implement the QFT to just $\lfloor \frac{n-1}{2} \rfloor$, yielding a significant resource saving for large n .

We now turn our attention to applications of GZZ gates in quantum chemistry beyond the full QPE. One promising approach is the use of quantum filtering techniques, particularly the combination of quantum filter diagonalization (QFD) with double factorization (DF), tailored for electronic structure problems [203]. QFD is a variational algorithm designed to extract spectral features of a Hamiltonian [207], while DF provides an efficient approximation of the electronic structure Hamiltonian [208, 209]. To simulate the time evolution under the compressed electronic structure Hamiltonian given by DF, a Trotter decomposition is employed. This approximates the total unitary evolution by a product of simpler unitaries over $m - 1$ Trotter steps, where m is chosen according to the desired approximation accuracy. The unitary operator can be written as

$$U_t \approx U_{Ext} \hat{G}(\varphi_m) \left[\prod_{k=1}^{m-1} GZZ(A_k) \hat{G}(\varphi_k) \right] \hat{R}_Z(\theta_{0,1}) \hat{G}(\varphi_0), \quad (4.19)$$

where $\hat{G}(\varphi)$ denotes a layer of two-qubit Givens rotations, and U_{Ext} represents the contribution from the nucleus-nucleus Coulomb interaction, which is constant in time. Each Givens rotation layer $\hat{G}(\varphi)$ can be realized using two $GZZ(-\varphi A)$ gates and appropriate single-qubit Clifford gates [32]. Here, the matrix $A \in \text{Sym}_0(\mathbb{F}_2^n)$ encodes nearest-neighbor couplings. Notably, each $GZZ(-\varphi A)$ gate can be implemented in constant depth, independent of n , leveraging our recent results on explicit constructions for time-optimal GZZ gates [33].

4.3 General, efficient, and robust Hamiltonian engineering

This section is based on our publication [34], which is listed in appendix C. Previously, we discussed Hamiltonian engineering techniques tailored specifically for Ising-type interactions. These methods focused on synthesizing target evolutions using only two-body ZZ interactions, which are commonly available on many quantum computing platforms [35, 36, 38–40]. However, quantum devices might possess more complex native dynamics in practice [142, 150, 210]. Each device naturally evolves under a system Hamiltonian H_S , which may contain many-body interaction terms beyond simple Ising-type couplings, such as ZZ , XX or YY . In light of this, we extend the LP framework introduced in section 4.2 to encompass arbitrary many-body Hamiltonians [34]. This generalization allows for broader applicability of the method across diverse quantum architectures, including superconducting qubits, trapped ions, and neutral atoms, each of which may exhibit different native couplings. The generalized method improves upon the efficiency of the original Ising Hamiltonian engineering in classical runtime and in the robustness against typical errors in an experimental setting.

As in the Ising case, we interleave intervals of free evolution under the native system Hamiltonian with single-qubit π or $\pi/2$ pulses, chosen from the set of single-qubit Pauli or Clifford gates. The underlying LP formulation again seeks to minimize the total evolution time required to implement the evolution under the target Hamiltonian. However, as the number of qubits increases, solving the full LP exactly becomes computationally intractable due to the exponential scaling in the number of degrees of freedom. To address this, we introduce a relaxation of the optimization problem that circumvents the exponential overhead. Instead of scaling directly with the number of qubits, the relaxed formulation depends only on the number and locality of interaction terms present in the system Hamiltonian. Consequently, the runtime becomes manageable even for larger quantum systems. Once a decomposition is obtained via the relaxed LP, the actual time evolution under the target Hamiltonian is implemented using established product formula techniques, such as Trotter-Suzuki methods, as reviewed in section 2.5.

In addition to efficiency, robustness is a key consideration in practical implementations. In real devices, single-qubit pulses used to engineer the Hamiltonian are not instantaneous and may overlap non-trivially with the system Hamiltonian. This leads to coherent errors during the pulse, as the system evolves under both the control pulse and the system Hamiltonian simultaneously. To mitigate these effects, we incorporate tools from average Hamiltonian theory (AHT) to analyze and correct these finite pulse time errors. Building on this analysis, we extend our method to support arbitrary robust composite pulses. These specially designed pulse sequences can suppress a broad range of coherent error sources, such as amplitude miscalibrations or off-resonant driving, making them robust in experimental settings [85, 93].

In section 4.3.1, we examine how single-qubit Pauli and Clifford gates affect many-body Hamiltonians through conjugation. Section 4.3.2 introduces an efficient LP formulation for Hamiltonian engineering. We then briefly address cases with incomplete knowledge of the system Hamiltonian in section 4.3.2. In section 4.3.3, we present a robust extension of the method that accounts for coherent control errors. Finally, section 4.3.4 showcases numerical simulations that validate the efficiency and robustness of our proposed techniques.

4.3.1 Conjugation of many-body Hamiltonians

To understand how single-qubit gates can transform and control interactions in a quantum system, we represent many-body Hamiltonians in terms of Pauli operators. Any quantum Hamiltonian acting on n qubits can be written as a weighted sum of Pauli strings, tensor products of single-qubit Pauli operators like I , X , Y , and Z . Each Pauli string specifies a particular type of interaction, and its corresponding weight determines the strength of that interaction. The total

system Hamiltonian is then described by

$$H_S = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n}} J_{\mathbf{a}} P_{\mathbf{a}}, \quad (4.20)$$

where we used the phase space notation from section 2.4 to label tensor products of Pauli operators $P_{\mathbf{a}}$ and their corresponding interaction strength $J_{\mathbf{a}}$ with a binary vector $\mathbf{a} \in \mathbb{F}_2^{2n}$. To describe which qubits are involved in each interaction, we denote the *locality* of a Pauli string as the set of qubits it acts on non-trivially. For example, a Pauli string that involves only two qubits is called 2-local, and the structure of such interactions can be visualized using an interaction graph, where nodes represent qubits and edges represent two-qubit interactions. When more than two qubits are involved in a single term (for example, 3-local or higher), we use interaction hypergraphs, where edges can connect more than two nodes.

Our objective is to simulate the time evolution under a *target Hamiltonian* H_T , using the native evolution of a system Hamiltonian H_S interspersed with layers of single-qubit gates \mathbf{S} . Specifically, we reshape the system dynamics by conjugating the system Hamiltonian, according to the identity

$$\mathbf{S}^\dagger e^{-itH_S} \mathbf{S} = e^{-it\mathbf{S}^\dagger H_S \mathbf{S}}. \quad (4.21)$$

This allows us to effectively synthesize a new Hamiltonian by alternating between periods of free evolution under H_S and applying chosen single-qubit gate layers. Ideally, we want to express H_T as a weighted sum of conjugated versions of H_S

$$H_T := \sum_{\mathbf{a} \in \mathbb{F}_2^{2n}} A_{\mathbf{a}} P_{\mathbf{a}} = \sum_{i=1}^L \lambda_i \mathbf{S}_i^\dagger H_S \mathbf{S}_i, \quad (4.22)$$

where $A_{\mathbf{a}}$ are the target interaction strengths and λ_i controls the relative time spent evolving under the transformed Hamiltonian. When the resulting terms commute with one another, this process allows for an exact implementation of the time evolution:

$$e^{-itH_T} = \prod_{i=1}^L \mathbf{S}_i^\dagger e^{-it\lambda_i H_S} \mathbf{S}_i \quad (4.23)$$

Otherwise, we can use approximation techniques, such as product formulae (discussed in section 2.5), to simulate the desired dynamics.

The key challenge lies in finding a good decomposition, ideally one that uses as little total evolution time as possible and keeps the number of gate layers minimal. As in the Ising case, we can frame this task as a linear programming problem. To this end, we capture the effect on the interaction term in H_S labeled by \mathbf{a} conjugated by the single-qubit gate layer \mathbf{S}_i by

$$W_{\mathbf{a},i} = \frac{1}{2^n} \text{Tr} \left(P_{\mathbf{a}} \left(\mathbf{S}_i^\dagger H_S \mathbf{S}_i \right) \right), \quad (4.24)$$

which can be collected in a matrix $W \in \mathbb{R}^{r \times s}$, where r is the number of interaction terms expressible by the conjugation and s is the number of considered single-qubit gate layers. This matrix takes on different values depending on the single-qubit gates, which will be elaborated in section 4.3.1. With that, we can formulate the LP

$$\begin{aligned} & \text{minimize} && \sum_i \lambda_i \\ & \text{subject to} && W\boldsymbol{\lambda} = \mathbf{A}, \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^s, \end{aligned} \quad (\text{LP})$$

where \mathbf{A} is the vector of the target interaction strengths.

Conjugation with Pauli gates

In this section, we explore how simple layers of Pauli gates (π pulses) can be used to transform the system Hamiltonian into a desired target form. When a Hamiltonian is conjugated by a Pauli string $P_{\mathbf{b}}$, the effect is to flip the sign of certain interaction terms while leaving others unchanged. This behavior follows directly from the Pauli commutation relations, as expressed in eq. (4.25), and changes the system dynamics as

$$P_{\mathbf{b}}^\dagger H_S P_{\mathbf{b}} = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} (-1)^{\langle \mathbf{a}, \mathbf{b} \rangle} J_{\mathbf{a}} P_{\mathbf{a}}. \quad (4.25)$$

This sign-flip pattern is captured by the well-known Walsh-Hadamard transform. Crucially, Pauli conjugation does not alter the type of interaction (i.e., the specific Pauli string involved), but only changes the sign of its coefficient. As a result, the target Hamiltonian must consist of the same set of Pauli interactions as the system Hamiltonian, albeit with possibly different interaction strengths. To describe this transformation compactly, we define the matrix W_P with entries $(-1)^{\langle \mathbf{a}, \mathbf{b} \rangle} J_{\mathbf{a}}$, which records the sign-flips of the coefficients $J_{\mathbf{a}}$ in the system Hamiltonian introduced by the Pauli gate layer labeled by \mathbf{b} . Since only the non-zero terms in H_S can be modified through this procedure, the interactions of the target Hamiltonian must be a subset of that of the system Hamiltonian

$$\text{nz}(\mathbf{A}) \subseteq \text{nz}(\mathbf{J}). \quad (4.26)$$

So, rather than working with the full matrix W_P , we focus on a partial version that includes only the non-zero interaction terms. Let there be r non-zero interaction terms in H_S and s Pauli gate layers considered in the (LP), then we denote the partial matrix capturing the effect of conjugation by $W_P^{(r \times s)} \in \mathbb{R}^{r \times s}$. We use this notation to highlight the number of interactions that can be individually engineered and the number of considered Pauli gate layers. Given this setup, an appropriate sequence of Pauli gates, interleaved with free evolution under the system Hamiltonian, can be determined by solving (LP), using $W_P^{(r \times s)}$ as the constraint matrix. Although the full set of Pauli operators on n qubits consists of $s = 4^n$ possible gate layers, this number can be significantly reduced. In section 4.3.2, we show that it suffices to consider a subset with $s > 2r$, enabling a more scalable and efficient approach.

Conjugation with Clifford gates

In the previous section, we introduced a method for Hamiltonian engineering based on conjugation with Pauli gates. While this approach is general and applies to any system Hamiltonian, it only allows changes in the strength of existing interaction terms. We now extend the framework to include a broader class of single-qubit operations, specifically, single-qubit Clifford gates, which correspond to $\pi/2$ pulses. These gates provide significantly more flexibility: in addition to flipping signs, they can rotate one type of Pauli term into another (e.g., turning an X interaction into a Z interaction). This expansion makes it possible to engineer a wider variety of target Hamiltonians, subject only to the locality constraints imposed by the system Hamiltonians interaction structure.

We consider a specific set of single-qubit Clifford gates

$$\mathcal{C}_{XY} := \mathbf{P} \cup \left\{ \sqrt{X}\sqrt{Y}, \sqrt{X}\sqrt{Y}^\dagger, \sqrt{X}^\dagger\sqrt{Y}, \sqrt{X}^\dagger\sqrt{Y}^\dagger, \sqrt{Y}\sqrt{X}, \sqrt{Y}\sqrt{X}^\dagger, \sqrt{Y}^\dagger\sqrt{X}, \sqrt{Y}^\dagger\sqrt{X}^\dagger \right\}, \quad (4.27)$$

where \mathbf{P} denotes the set of Pauli operators, and the square-root operators represent $\pi/2$ rotations around the x - and y -axes. This gate set is chosen to resemble the sign-flipping structure of Pauli conjugation while introducing the ability to permute interaction types.

We label a Clifford gate layer on n qubits as $S_{(\mathbf{p}, \mathbf{b})} \in \mathcal{C}_{XY}^{\otimes n}$, where \mathbf{p} determines the change in interaction type and \mathbf{b} encodes sign flips. For convenience, we combine these into a single label

$\mathbf{c} := (\mathbf{p}, \mathbf{b})$. When the system Hamiltonian H_S is conjugated by a Clifford layer $S_{\mathbf{c}}$, the resulting transformed Hamiltonian takes the form

$$S_{\mathbf{c}}^\dagger H_S S_{\mathbf{c}} = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} (-1)^{\langle \pi_{\mathbf{p}}(\mathbf{a}), \mathbf{b} \rangle} J_{\pi_{\mathbf{p}}(\mathbf{a})} P_{\mathbf{a}}. \quad (4.28)$$

Here, $\pi_{\mathbf{p}}$ is a permutation that maps each original Pauli term to a new type according to \mathbf{p} , while the symplectic form $\langle \cdot, \cdot \rangle$ with \mathbf{b} determines the sign of each term. This transformation is compactly captured by a matrix W_C with entries $(-1)^{\langle \pi_{\mathbf{p}}(\mathbf{a}), \mathbf{b} \rangle} J_{\pi_{\mathbf{p}}(\mathbf{a})}$, whose structure closely resembles that of the Walsh-Hadamard matrix used in the Pauli conjugation case. As before, we focus on a reduced version of this matrix, denoted $W_C^{(r \times s)} \in \mathbb{R}^{r \times s}$, where r is the number of interaction terms consistent with the locality structure of H_S , and s is the number of Clifford gate layers under consideration. One of the key advantages of Clifford conjugation is that a k -local term in H_S can now be transformed into any other k -local Pauli term acting on the same qubits and with arbitrary interaction strength. For example, a ZZ interaction can be mapped to any of the other 2-local interactions, such as XX , XY , or YZ , on the same pair of qubits. As in the Pauli case, we can determine the Clifford gate layers and the corresponding relative evolution times by solving (LP), but now using $W_C^{(r \times s)}$ as the constraint matrix.

Although the number of all possible Clifford layers scales as $s = 12^n$, we show in the next section how this can be efficiently reduced. In particular, we propose a scalable relaxation of the problem that works with a subset of Clifford layers, requiring only $s > 2r$.

4.3.2 The efficient linear program approach

We present an efficient approach to reduce the number of columns s in the matrices $W_P^{(r \times s)}$ and $W_C^{(r \times s)}$, while still ensuring the feasibility of the (LP). This reduction significantly improves classical runtime without compromising the exactness of the decomposition. A central result relating W to LP feasibility is as follows:

Proposition 10 (Proposition III.2 [34], informal). *Let $W \in \mathbb{R}^{r \times s}$. If $\mathbf{0} \in \text{conv}(W)$, then*

$$\begin{aligned} & \text{minimize} \quad \sum_i \lambda_i \\ & \text{subject to} \quad W\boldsymbol{\lambda} = \mathbf{A}, \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^s \end{aligned} \quad (\text{LP})$$

has a solution for any $\mathbf{A} \in \mathbb{R}^r$.

This result motivates a randomized strategy: instead of using the full matrix, we randomly sample a subset of s columns from a larger matrix to form the reduced matrix $W \in \mathbb{R}^{r \times s}$. The key question is: how large does s need to be to ensure that $\mathbf{0} \in \text{conv}(W)$ holds with high probability? If the columns of W are drawn from a distribution that is spherically symmetric around the origin, Wendel's Theorem [211] states that

$$\mathbb{P}(\mathbf{0} \in \text{conv}(W)) = 1 - \frac{1}{2^{s-1}} \sum_{k=0}^{r-1} \binom{s-1}{k}. \quad (4.29)$$

This distribution shows a sharp transition from ≈ 0 to ≈ 1 at $s = 2r$. Although, the columns of $W_P^{(r \times 4^n)}$ and $W_C^{(r \times 12^n)}$ are not drawn from a spherically symmetric distribution. However, numerical experiments suggest they exhibit similar transition behavior [34]. This transition behavior also occurs for any set of rows in $W_P^{(r \times 4^n)}$ and $W_C^{(r \times 12^n)}$, i.e. any set of interactions in the system Hamiltonian H_S . Based on these observations, we conclude that sampling

$$s > 2r \quad (4.30)$$

columns is sufficient to ensure LP feasibility with high probability. The decomposition of the target Hamiltonian H_T obtained from such a relaxed LP is still exact, however the total relative evolution time is not minimal anymore. The quality of the relaxation can be improved by increasing s , thus expanding the search space. This provides a flexible trade-off between the classical runtime for solving the LP and the “quantum runtime”, i.e. the optimality of $\sum_i \lambda_i$.

Hamiltonian engineering with unknown Hamiltonians

In practice, some system Hamiltonian coupling coefficients in \mathbf{J} may be unknown. For instance, when an experiment targeting two-body interactions unintentionally introduces three-body terms of unknown strength. Nevertheless, such system Hamiltonians with partially known couplings can still be used for Hamiltonian engineering via the Pauli conjugation method from section 4.3.1. Solving (LP) for a vector $\mathbf{M} \in \mathbb{R}^r$ yields the target Hamiltonian

$$\sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} M_{\mathbf{a}} J_{\mathbf{a}} P_{\mathbf{a}} = H_T, \quad (4.31)$$

where $\mathbf{M} \odot \mathbf{J}$ modifies the system couplings element-wise. By choosing $M_{\mathbf{a}} = -1$ or 0 , one can invert or cancel terms $P_{\mathbf{a}}$ without knowing $J_{\mathbf{a}}$. Each term can be treated individually: known interactions can be engineered via $M_{\mathbf{a}} = A_{\mathbf{a}}/J_{\mathbf{a}}$, while unknown ones can be suppressed or inverted. Thus, even with incomplete knowledge of \mathbf{J} , one can design \mathbf{M} to selectively manipulate the effective Hamiltonian. Dummy values for unknown $J_{\mathbf{a}}$ can be used in (LP) to carry out this procedure.

4.3.3 Error robustness

To reliably implement Pauli or Clifford conjugation in practice, it is essential that the resulting pulse sequences are robust against dominant experimental imperfections. In this section, we present low-overhead error mitigation strategies tailored to our efficient conjugation methods, with particular focus on finite pulse time errors, i.e. errors arising from the simultaneous action of the system Hamiltonian and finite duration of single-qubit control pulses.

In addition, we extend our Clifford conjugation method to incorporate robust composite pulses, which enhance resilience against a wide variety of experimental noise sources, including rotation angle errors, off-resonance effects, phase errors, pulse shape distortions, non-Markovian noise, and crosstalk [84–91]. To analyze and mitigate the finite pulse time error, we employ average Hamiltonian theory (AHT) [77, 78], as outlined in section 2.6. A key insight is that the finite pulse time error preserves the locality of the system Hamiltonian, enabling cancellation strategies within our efficient LP framework.

Investigation of finite pulse time errors

We begin by examining the errors that arise when applying sequences of single-qubit pulses to a system governed by an always-on Hamiltonian H_S . Although our main focus is on Clifford conjugation, the similar principles apply to Pauli-based sequences (a detailed analysis of which is given in our work [34]).

We consider single-qubit pulses that applies a rotation around one of the Bloch sphere axes. In practice such a pulse has a finite duration $t_p > 0$. When such pulses are applied in parallel across multiple qubits, we refer to the collective operation as a single-qubit pulse layer. Each layer is associated with a local Hamiltonian H_c and a sequence of layers (indexed by $\ell = 1, \dots, n_L$) defines a pulse block

$$\mathbf{S}_c := \prod_{\ell=1}^{n_L} e^{-it_p^{(\ell)} H_{c^{(\ell)}}}. \quad (4.32)$$

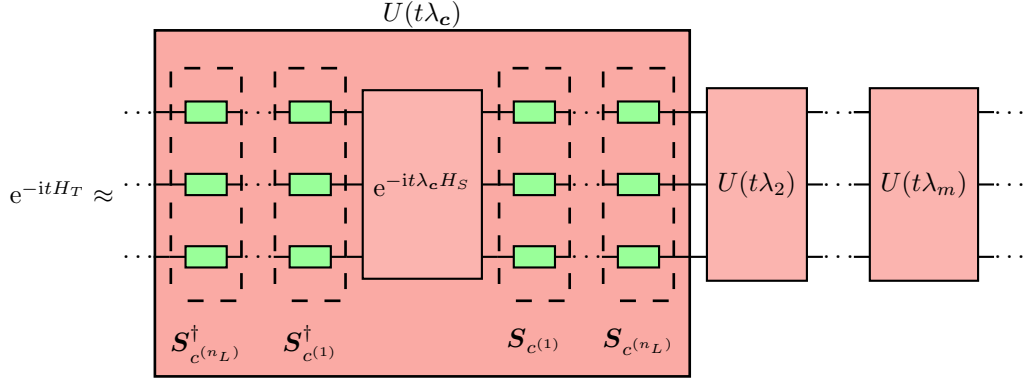


Figure 4.2: Exemplary quantum circuit for approximating the target evolution. We only implement simple single-qubit pulses in the presence of an always-on system Hamiltonian H_S .

Ideally, we would like to treat these pulse blocks as instantaneous transformations, where conjugation of the system Hamiltonian H_S leads to a change in interaction type and sign. Under this assumption, the evolution would be

$$\left(\prod_{\ell=n_L}^1 e^{-i(-t_p^{(\ell)} H_{c(\ell)})} \right) e^{-it\lambda_c H_S} \left(\prod_{\ell=1}^{n_L} e^{-it_p^{(\ell)} H_{c(\ell)}} \right) = \mathbf{S}_c^\dagger e^{-it\lambda_c H_S} \mathbf{S}_c = e^{-it\lambda_c \mathbf{S}_c^\dagger H_S \mathbf{S}_c}, \quad (4.33)$$

where λ_c is the relative evolution time under the transformed Hamiltonian. However, in practice, the system Hamiltonian H_S remains active during the pulse durations. This overlap introduces an unwanted deviation from the ideal behavior. The full evolution now includes the combined effect of the pulses and the background system dynamics, and is described by

$$U(t\lambda_c) := \left(\prod_{\ell=n_L}^1 e^{-it_p^{(\ell)} (H_S - H_{c(\ell)})} \right) e^{-it\lambda_c H_S} \left(\prod_{\ell=1}^{n_L} e^{-it_p^{(\ell)} (H_S + H_{c(\ell)})} \right). \quad (4.34)$$

A visual representation of this circuit is shown in figure 4.2. Using tools from average Hamiltonian theory, specifically the first-order Magnus expansion, we can approximate the resulting evolution as

$$U(t\lambda_c) \approx e^{-it\lambda_c \mathbf{S}_c^\dagger H_S \mathbf{S}_c + H_{\text{err},c}}, \quad (4.35)$$

where $H_{\text{err},c}$ is an effective error Hamiltonian capturing the unwanted impact of finite pulse durations. Crucially, this error term preserves the locality of the original Hamiltonian. This means that we can incorporate it directly into our LP-based optimization scheme to mitigate its influence. To formalize the correction scheme, we define a matrix of error coefficients $E^{(r \times s)} \in \mathbb{R}^{r \times s}$ with the entries

$$E_{\mathbf{a}\mathbf{c}}^{(r \times s)} := \frac{1}{2^n} \text{Tr}(P_{\mathbf{a}} H_{\text{err},c}), \quad (4.36)$$

which captures the finite pulse time error contributions. These coefficients can be computed efficiently for any local system Hamiltonian H_S . We can then efficiently solve a modified linear program

$$\begin{aligned} & \text{minimize} \quad \mathbf{1}^T \boldsymbol{\lambda} \\ & \text{subject to} \quad W_C^{(r \times s)} \boldsymbol{\lambda} + E^{(r \times s)} \mathbf{1} = \mathbf{A}, \\ & \quad \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^s. \end{aligned} \quad (\text{robustLP})$$

This approach enables us to simultaneously approximate the target dynamics and suppress leading-order errors due to finite pulse durations, enhancing the reliability of Hamiltonian engineering in realistic settings.

Robust Clifford conjugation

We now integrate the Clifford conjugation technique described in section 4.3.1 with the general robust framework introduced in section 4.3.3. In that setting, each Clifford operation is realized through two sequential single-qubit $\pi/2$ pulses, corresponding to a total of $n_L = 2$ single-qubit pulse layers per single-qubit pulse block \mathbf{S}_c . To enhance robustness against systematic control errors such as pulse amplitude or detuning errors, each $\pi/2$ rotation can be replaced by a robust composite pulse sequence, as outlined in section 2.7. In this case, n_L is twice the length of the robust composite pulse sequence. Importantly, these robust composite pulses are designed such that they preserve the ideal unitary evolution. Therefore, the ideal conjugation effect on the system Hamiltonian remains unchanged. That is, the coefficient matrix $W_C^{(r \times s)}$, as in (robustLP), remains identical to the one derived for non-robust Clifford conjugation. By constructing a sufficiently large set of single-qubit Clifford pulse blocks \mathbf{S}_c with $s > 2r$, we can ensure with high probability that the (robustLP) is feasible, as detailed in section 4.3.2.

However, one practical challenge arises when seeking sparse solutions to reduce experimental overhead. Each column of the coefficient matrix $E^{(r \times s)}$ corresponds to the finite pulse time error introduced by a specific pulse block \mathbf{S}_c . In cases where the relative evolution time $\lambda_c = 0$, the ideal contribution to the target evolution vanishes, but the physical pulses still need to be implemented. If not, then the term $E^{(r \times s)}\mathbf{1}$ in the constraint of the (robustLP) would take into account too many finite pulse time error terms. To address this, we introduce a sparsity-aware formulation by replacing the all-ones vector $\mathbf{1}$ in the constraint of (robustLP) with a binary indicator vector $\mathbf{z} \in \{0, 1\}^s$, where each entry $z_c = 1$ indicates that the corresponding pulse block is active (i.e. $\lambda_c > 0$). This yields a mixed-integer linear program (MILP) that jointly optimizes for the total evolution time, and the number of activated pulse blocks [212]

$$\begin{aligned} & \text{minimize} && \alpha \mathbf{1}^T \boldsymbol{\lambda} + (1 - \alpha) \mathbf{1}^T \mathbf{z} \\ & \text{subject to} && W_C^{(r \times s)} \boldsymbol{\lambda} + E^{(r \times s)} \mathbf{z} = \mathbf{A}, \\ & && c_l \mathbf{z} \leq \boldsymbol{\lambda} \leq c_u \mathbf{z}, \\ & && \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^s, \quad \mathbf{z} \in \{0, 1\}^s. \end{aligned} \tag{MILP}$$

Here $\alpha \in [0, 1]$ is a user-defined weight that trades off between minimizing total evolution time $\mathbf{1}^T \boldsymbol{\lambda}$ and minimizing the number of pulse blocks $\mathbf{1}^T \mathbf{z}$. The constants $0 \leq c_l < c_u$ provide lower and upper bounds on the non-zero entries of $\boldsymbol{\lambda}$. Although solving MILPs is in general a hard task due to their combinatorial nature, a wide range of powerful heuristics and commercial solvers are available for practical problem sizes [213, 214].

4.3.4 Numerical simulations

To evaluate the performance and robustness of our proposed methods, we perform numerical simulations that model the dominant coherent error sources encountered in experimental implementations. Our simulations focus on the robust Clifford conjugation scheme introduced above.

For a given target Hamiltonian specified by the interaction strength vector $\mathbf{A} \in \mathbb{R}^r$, we first solve the (robustLP) (or the (MILP)) to obtain the relative evolution times $\boldsymbol{\lambda}$. The time evolution of our simulation is then computed by explicitly computing products of matrix exponentials of the form described in eq. (4.34), which include contributions of finite pulse time errors. To approximate the target evolution over time t , we employ a second-order Trotter-Suzuki decomposition

$$U_{\text{sim}} = \left(\overleftarrow{\prod}_c U\left(\frac{t\lambda_c}{2n_{\text{Tro}}}\right) \overrightarrow{\prod}_c U\left(\frac{t\lambda_c}{2n_{\text{Tro}}}\right) \right)^{n_{\text{Tro}}}, \tag{4.37}$$

where n_{Tro} is the number of Trotter cycles and the arrows indicate the order of the product. Note, that if the (MILP) was solved, then only the conjugation blocks $U(t\lambda_c)$ with non-zero λ_c has to be implemented. To capture the effects of control imperfections, we consider a perturbation

of the ideal single-qubit pulse Hamiltonian H_c . Specifically, the imperfect control Hamiltonian takes the form

$$H_{c,(\varepsilon,f)} = \frac{1}{t_p} \sum_{i=1}^n ((1 + \varepsilon_i)h_i + f_i Z_i) , \quad (4.38)$$

where h_i is the ideal generator of the single-qubit pulse, $\varepsilon_i \stackrel{\text{i.i.d.}}{\sim} \text{unif}([0, \varepsilon])$ is the *relative angle error* and $f_i \stackrel{\text{i.i.d.}}{\sim} \text{unif}([0, f])$ is the *off-resonance error* on the i -th qubit. To quantify the accuracy of the simulated implementation U_{sim} relative to the ideal target evolution $U_T = e^{-itH_T}$, we compute the *average gate infidelity*

$$1 - F_{\text{avg}}(U_{\text{sim}}, U_T) = 1 - \frac{\text{Tr}(U_T^\dagger U_{\text{sim}}) + 1}{d + 1} , \quad (4.39)$$

where $d = 2^n$ is the dimension of the Hilbert space.

Simulation of Heisenberg Hamiltonians with an ion trap model

We numerically simulate the implementation of time evolution under Heisenberg Hamiltonians using a model of a trapped-ion quantum processor. Specifically, we consider a linear chain of Ytterbium ions confined in a harmonic trapping potential and subjected to an external magnetic field gradient [202]. The native interaction between the ions is modeled by the Ising Hamiltonian

$$H_S = \sum_{i \neq j}^n J_{ij} Z_i Z_j , \quad (4.40)$$

where the coupling coefficients J_{ij} are computed based on realistic experimental parameters. For further details on the computation of J_{ij} , we refer the reader to ref. [32, App. A]. We assume a finite π pulse duration of $t_p = 2 \mu\text{s}$, rotation angle errors of strength $\varepsilon = 10^{-1}$ and off-resonance errors of strength $f = 10^{-1}$, modeled as described in eq. (4.38). The coherent error parameters are sampled once per simulation instance. This models static, spatially inhomogeneous errors, such as calibration drifts or hardware imperfections that remain approximately constant during a typical experiment. As a target Hamiltonian, we choose the general Heisenberg Hamiltonian

$$H_T = \sum_{i \neq j}^n \left(A_{ij}^x X_i X_j + A_{ij}^y Y_i Y_j + A_{ij}^z Z_i Z_j \right) , \quad (4.41)$$

with coupling coefficients $A_{ij}^x, A_{ij}^y, A_{ij}^z \stackrel{\text{i.i.d.}}{\sim} \text{unif}([10^{-1}, 1] \cdot \text{Hz})$ and all-to-all connectivity.

The results of our numerical simulations are presented in figure 4.3, where we plot the average gate infidelity (as defined in eq. (4.39)) as a function of the number of Trotter cycles n_{Tro} , for a total evolution time of $t = 1 \text{ s}$. The data represent the sample mean and standard deviation over 50 randomly generated Heisenberg Hamiltonians on $n = 8$ qubits. The simulations compare several implementations of the robust Clifford conjugation scheme under different levels of error robustness. The non-robust naive Clifford sequences (red curves) quickly accumulates errors as the number of Trotter cycles increases, particularly due to the finite pulse durations and uncorrected pulse control inaccuracies. In contrast, the robust Clifford method (green curve), which only compensates for finite pulse time effects, significantly improves fidelity and maintains performance even for large n_{Tro} . Further improvements are achieved by combining Clifford conjugation with robust composite pulse sequences: SCROFULOUS (dark blue) and SCROBUTUS (light blue), which additionally suppress rotation angle errors and both rotation angle & off-resonance errors, respectively [85, 93], as outlined in section 2.7. These results highlight the effectiveness of combining our Clifford conjugation method with robust composite pulses. Remarkably, the robust sequences achieve fidelities comparable to, or even slightly exceeding, the ideal Trotterized evolution in the absence of errors. This may be attributed to beneficial averaging effects inherent in the robust design.

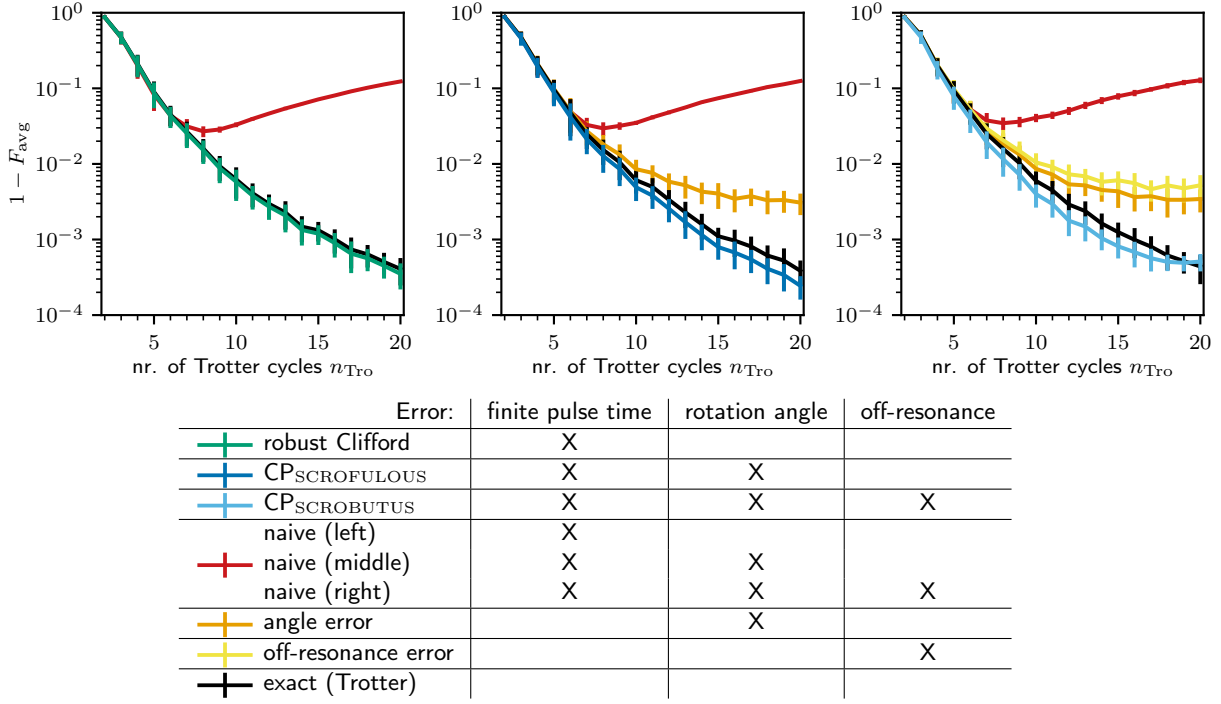


Figure 4.3: **Top Left:** The Clifford conjugation robust against finite pulse time errors (dark green) is compared to the non-robust Clifford conjugation (red). **Top Middle:** The Clifford conjugation in combination with the SCROFULOUS pulse sequence [93] robust against finite pulse time errors and rotation angle errors (dark blue) is compared to the non-robust Clifford conjugation (red). **Top Right:** The Clifford conjugation in combination with the SCROBUTUS pulse sequence [85] robust against finite pulse time errors, rotation angle errors and off-resonance errors (light blue) is compared to the non-robust Clifford conjugation (red). **Bottom:** Table indicating which error types are present for the different simulations.

4.4 Summary of results

This section summarizes the key findings of our publication [34], emphasizing on their required resources and providing a guide on how to apply our method for simulating the dynamics of a desired target Hamiltonian on quantum hardware. We begin by outlining the naive, non-robust method and subsequently extend the discussion to the robust variant developed in the course of this work.

The fundamental requirement for applying our method is knowledge of two ingredients: the system Hamiltonian H_S of the quantum hardware and the desired target Hamiltonian H_T , provided as a sum of interactions as Pauli terms and their corresponding interactions strengths. Our linear programming approach enables an *exact and efficient decomposition* of the target Hamiltonian into the sum

$$H_T = \sum_{i=1}^r \lambda_i \mathbf{S}_i^\dagger H_S \mathbf{S}_i, \quad (4.42)$$

where each $\lambda_i \geq 0$ represents the relative evolution time under H_S , and \mathbf{S}_i denotes parallel single-qubit pulses, either π or $\pi/2$. The number r reflects the number of interaction terms, which can be individually modified, and which are considered in the LP formulation. For the Pauli method (π pulses), r equals the number of interaction terms present in the system Hamiltonian. In contrast, the Clifford method ($\pi/2$ pulses), expands the available basis to include all interactions of the same locality, thereby increasing r accordingly.

Each term in the decomposition of H_T can be implemented on the hardware by the following

procedure: apply π or $\pi/2$ pulses, let the system evolve under H_S for a time given by λ_i , apply inverse π or $\pi/2$ pulses. A product formula can then be used to simulate the complete dynamics governed by H_T . This is the only source of approximation in our non-robust method, as the LP provides an exact decomposition. When evaluating the gate cost of our approach, we only count the number of conjugation terms, i.e., the number of $\mathbf{S}_i^\dagger H_S \mathbf{S}_i$ terms, which have to be implemented and omit the overhead introduced by the product formula, since this overhead applies equally across our methods.

One major error source in an actual implementation is finite pulse time error. To mitigate this, we extend our LP framework to incorporate an additional constraint term that approximately models this error source. The modified LP also allows us to include robust composite pulses, making our method robust against several control errors. The challenge is that the precise impact of the finite pulse time error depends on the specific sequence of pulses used, which are not known a priori. To address this, all s considered pulses in the LP have to be implemented, even if $\lambda_i = 0$. Since we can choose $s > 2r$, the overhead in the number of pulses remains low, especially when compared to the minimal r pulses required for the non-robust version. Again, we want to highlight, that in the robust implementation, the only approximations arise from the approximation in modeling the finite pulse time error via the Magnus expansion, and the use of a product formula to simulate the time evolution.

In terms of computational complexity, for r considered interactions, the LP size and therefore the runtime scales as $O(r^2)$. The number of terms in the decomposition of H_T to be implemented, i.e. the gate cost, is r for the non-robust case and s for the robust case, where $s > 2r$.

Chapter 5

Conclusion

The ability to efficiently compile large-scale quantum algorithms is a foundational requirement for quantum computing. As quantum devices grow in size and complexity, and as quantum algorithms become more sophisticated, the role of quantum compilation becomes ever more important. Compilation serves as the interface between abstract algorithm design and the physical constraints of quantum hardware, determining whether a theoretical advantage can be translated into an actual experiment. Among the most promising applications of quantum computing is the simulation of quantum systems, a task first emphasized by Richard Feynman in 1981, who famously remarked that “if you want to make a simulation of Nature, you’d better make it quantum mechanical” [1]. Simulating the dynamics of a quantum system underpins a wide range of applications. As such, methods that allow for flexible, efficient, and hardware-aware Hamiltonian simulation are of substantial practical and theoretical interest.

In the first part of this thesis, we examined core principles and recent advances in quantum compilation, with particular focus on Clifford circuits and arbitrary single-qubit gate decompositions. These two classes of operations form the backbone of many quantum algorithms [69, 122, 123], and they are integral to fault-tolerant quantum computation [110, 111] and characterization of quantum devices [108, 109]. For Clifford unitaries, we reviewed both asymptotically optimal and practically efficient decomposition strategies, including constructions that require only $O(n^2)$ two-qubit gates [19] or a constant number of global entangling gates [118], which are especially relevant for architectures like trapped ions. For single-qubit gates, we discussed decomposition schemes that achieve optimal scaling in the approximation error ε , requiring $O(\log_2(1/\varepsilon))$ gates [17, 126, 127]. Beyond gate synthesis, we addressed a variety of optimization techniques aimed at minimizing resource overhead in quantum devices. These included strategies for qubit routing on constrained architectures and methods for eliminating redundant gates, thereby reducing the overall circuit depth and improving fidelity under hardware noise.

The second and major contribution of this work lies in the field of Hamiltonian engineering, where we developed a versatile and scalable approach for implementing tailored many-body interactions on quantum hardware. Our method builds on the native time evolution of the quantum system and well-timed, parallel single-qubit pulse sequences. The key innovation is a linear programming LP framework that determines the single-qubit pulses and the associated evolution times under the system Hamiltonian to synthesize an effective target Hamiltonian. A distinguishing feature of our approach is that its classical runtime depends only on the number and locality of the system interactions, and not on the total number of qubits, making it especially well-suited for large quantum systems with few local interactions. In addition, the method inherently supports robustness against common experimental imperfections, such as finite pulse time effects, through the incorporation of techniques from average Hamiltonian theory and robust composite pulses. We demonstrated numerically that this framework can be used to engineer a wide range of Hamiltonians with high fidelity and low experimental overhead, enabling practical quantum simulations on current NISQ devices. The methods developed herein are not only of theoretical interest but are also directly implementable on existing experimental platforms,

including ion traps, neutral atoms, and superconducting qubits.

Looking ahead, several compelling research directions emerge from this work. One natural extension is the generalization of the LP based Hamiltonian engineering approach to systems beyond qubits, such as qudits, higher-dimensional quantum state space that make use of multiple excited states. This generalization could open new possibilities for encoding and manipulating information more compactly. Another promising yet challenging direction is the adaptation of our framework to fermionic systems, which are central to quantum chemistry applications. Incorporating fermionic commutation relations into the synthesis process will likely require novel formulations, but the potential benefits are significant. Beyond these structural extensions, further gains in robustness and error mitigation remain an important goal. Future work may explore the integration of tailored robust pulses or maybe even closed-loop control based on real-time feedback to enhance performance in noisy environments.

Bibliography

- [1] R. P. Feynman, *Simulating physics with computers*, Int. J. Theor. Phys. **21**, 467 (1982).
- [2] P. Benioff, *The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines*, Journal of Statistical Physics **22**, 563 (1980).
- [3] D. Deutsch and R. Penrose, *Quantum theory, the Church–Turing principle and the universal quantum computer*, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences **400**, 97 (1985), <https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1985.0070>.
- [4] P. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings 35th Annual Symposium on Foundations of Computer Science 10.1109/sfcs.1994.365700 (1994).
- [5] L. 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 (Association for Computing Machinery, New York, NY, USA, 1996) p. 212–219, arXiv:quant-ph/9605043 .
- [6] A. Barenco, D. Deutsch, A. Ekert, and R. Jozsa, *Conditional quantum dynamics and logic gates*, Phys. Rev. Lett. **74**, 4083 (1995), arXiv:quant-ph/9503017.
- [7] C. Monroe, D. M. Meekhof, B. E. King, W. M. Itano, and D. J. Wineland, *Demonstration of a fundamental quantum logic gate*, Phys. Rev. Lett. **75**, 4714 (1995).
- [8] B. Cheng, X.-H. Deng, X. Gu, Y. He, G. Hu, P. Huang, J. Li, B.-C. Lin, D. Lu, Y. Lu, C. Qiu, H. Wang, T. Xin, S. Yu, M.-H. Yung, J. Zeng, S. Zhang, Y. Zhong, X. Peng, F. Nori, and D. Yu, *Noisy intermediate-scale quantum computers*, Frontiers of Physics **18**, 21308 (2023), arXiv:2303.04061.
- [9] J. Preskill, *Quantum Computing in the NISQ era and beyond*, Quantum **2**, 79 (2018), arXiv:1801.00862.
- [10] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, J. P. Bonilla Ataides, N. Maskara, I. Cong, X. Gao, P. Sales Rodriguez, T. Karolyshyn, G. Semeghini, M. J. Gullans, M. Greiner, V. Vuletić, and M. D. Lukin, *Logical quantum processor based on reconfigurable atom arrays*, Nature **626**, 58 (2024), arXiv:2312.039822.
- [11] R. Acharya, D. A. Abanin, L. Aghababaie-Beni, I. Aleiner, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, N. Astrakhantsev, J. Atalaya, R. Babbush, D. Bacon, B. Ballard, J. C. Bardin, J. Bausch, A. Bengtsson, A. Bilmes, S. Blackwell, S. Boixo, G. Bortoli, A. Bourassa, J. Bovaird, L. Brill, M. Broughton, D. A. Browne, B. Buchea, B. B. Buckley, D. A. Buell, T. Burger, B. Burkett, N. Bushnell, A. Cabrera, J. Campero, H.-S. Chang, Y. Chen, Z. Chen, B. Chiaro, D. Chik, C. Chou, J. Claes, A. Y. Cleland, J. Cogan, R. Collins, P. Conner, W. Courtney, A. L. Crook, B. Curtin, S. Das, A. Davies, L. De Lorenzo, D. M. Debroy, S. Demura, M. Devoret, A. Di Paolo, P. Donohoe, I. Drozdov, A. Dunsworth, C. Earle, T. Edlich, A. Eickbusch, A. M. Elbag, M. Elzouka, C. Erickson, L. Faoro, E. Farhi, V. S. Ferreira, L. F. Burgos, E. Forati, A. G. Fowler, B. Foxen, S. Ganjam, G. Garcia, R. Gasca, É. Genois, W. Giang, C. Gidney, D. Gilboa, R. Gosula, A. G. Dau, D. Graumann, A. Greene, J. A. Gross, S. Habegger, J. Hall, M. C. Hamilton, M. Hansen, M. P. Harrigan, S. D. Harrington, F. J. H. Heras, S. Heslin, P. Heu, O. Higgott, G. Hill, J. Hilton, G. Holland, S. Hong, H.-Y. Huang, A. Huff, W. J. Huggins,

- L. B. Ioffe, S. V. Isakov, J. Iveland, E. Jeffrey, Z. Jiang, C. Jones, S. Jordan, C. Joshi, P. Juhas, D. Kafri, H. Kang, A. H. Karamlou, K. Kechedzhi, J. Kelly, T. Khairi, T. Khat-tar, M. Khezri, S. Kim, P. V. Klimov, A. R. Klots, B. Kobrin, P. Kohli, A. N. Korotkov, F. Kostitsa, R. Kothari, B. Kozlovskii, J. M. Kreikebaum, V. D. Kurilovich, N. Lacroix, D. Landhuis, T. Lange-Dei, B. W. Langley, P. Laptev, K.-M. Lau, L. Le Guevel, J. Ledford, J. Lee, K. Lee, Y. D. Lensky, S. Leon, B. J. Lester, W. Y. Li, Y. Li, A. T. Lill, W. Liu, W. P. Livingston, A. Locharla, E. Lucero, D. Lundahl, A. Lunt, S. Madhuk, F. D. Malone, A. Maloney, S. Mandrà, J. Manyika, L. S. Martin, O. Martin, S. Martin, C. Maxfield, J. R. McClean, M. McEwen, S. Meeks, A. Megrant, X. Mi, K. C. Miao, A. Mieszala, R. Molavi, S. Molina, S. Montazeri, A. Morvan, R. Movassagh, W. Mruczkiewicz, O. Naaman, M. Nee-ley, C. Neill, A. Nersisyan, H. Neven, M. Newman, J. H. Ng, A. Nguyen, M. Nguyen, C.-H. Ni, M. Y. Niu, T. E. O'Brien, W. D. Oliver, A. Opremcak, K. Ottosson, A. Petukhov, A. Pizzuto, J. Platt, R. Potter, O. Pritchard, L. P. Pryadko, C. Quintana, G. Ramachan-dran, M. J. Reagor, J. Redding, D. M. Rhodes, G. Roberts, E. Rosenberg, E. Rosenfeld, P. Roushan, N. C. Rubin, N. Saei, D. Sank, K. Sankaragomathi, K. J. Satzinger, H. F. Schurkus, C. Schuster, A. W. Senior, M. J. Shearn, A. Shorter, N. Shutty, V. Shvarts, S. Singh, V. Sivak, J. Skrzynny, S. Small, V. Smelyanskiy, W. C. Smith, R. D. Somma, S. Springer, G. Sterling, D. Strain, J. Suchard, A. Szasz, A. Sztein, D. Thor, A. Torres, M. M. Torunbalci, A. Vaishnav, J. Vargas, S. Vdovichev, G. Vidal, B. Villalonga, C. V. Heidweiller, S. Waltman, S. X. Wang, B. Ware, K. Weber, T. Weidel, T. White, K. Wong, B. W. K. Woo, C. Xing, Z. J. Yao, P. Yeh, B. Ying, J. Yoo, N. Yosri, G. Young, A. Zalcman, Y. Zhang, N. Zhu, N. Zobrist, G. Q. AI, and Collaborators, *Quantum error correction below the surface code threshold*, *Nature* **638**, 920 (2024), arXiv:2408.13687.
- [12] C. Huang, F. Zhang, M. Newman, J. Cai, X. Gao, Z. Tian, J. Wu, H. Xu, H. Yu, B. Yuan, M. Szegedy, Y. Shi, and J. Chen, *Classical simulation of quantum supremacy circuits* (2020), arXiv:2005.06787 .
- [13] A. J. Daley, I. Bloch, C. Kokail, S. Flannigan, N. Pearson, M. Troyer, and P. Zoller, *Practical quantum advantage in quantum simulation*, *Nature* **607**, 667 (2022).
- [14] J. Tindall, M. Fishman, E. M. Stoudenmire, and D. Sels, *Efficient tensor network sim-ulation of IBM's Eagle kicked Ising experiment*, *PRX Quantum* **5**, 10.1103/prxquan-tum.5.010308 (2024), arXiv:2306.14887.
- [15] C. M. Dawson and M. A. Nielsen, *The Solovay-Kitaev algorithm*, *Quantum Info. Comput.* **6**, 81–95 (2006), arXiv:quant-ph/0505030.
- [16] S. Bravyi and A. Kitaev, *Universal quantum computation with ideal Clifford gates and noisy ancillas*, *Phys. Rev. A* **71**, 022316 (2005), arXiv:quant-ph/0403025.
- [17] N. J. Ross and P. Selinger, *Optimal ancilla-free Clifford+T approximation of z-rotations*, *Quantum Inform. Compu.* **16**, 901 (2016), arXiv:1403.2975.
- [18] D. Maslov and M. Roetteler, *Shorter stabilizer circuits via Bruhat decomposition and quan-tum circuit transformations*, *IEEE Trans. Inf. Theory* **64**, 4729 (2018), arXiv:1705.09176.
- [19] S. Bravyi and D. Maslov, *Hadamard-free circuits expose the structure of the Clifford group*, *IEEE Trans. Inf. Theory* **67**, 4546 (2021), arXiv:2003.09412.
- [20] S. Bravyi, R. Shaydulin, S. Hu, and D. Maslov, *Clifford circuit optimization with templates and symbolic Pauli gates*, *Quantum* **5**, 580 (2021), arXiv:2105.02291.
- [21] S. Lloyd, *Universal quantum simulators*, *Science* **273**, 1073 (1996).

- [22] D. G. Cory, M. D. Price, and T. F. Havel, *Nuclear magnetic resonance spectroscopy: An experimentally accessible paradigm for quantum computing*, Physica D: Nonlinear Phenomena **120**, 82 (1998), proceedings of the Fourth Workshop on Physics and Consumption, arXiv:quant-ph/9709001.
- [23] J. I. Cirac and P. Zoller, *Quantum computations with cold trapped ions*, Phys. Rev. Lett. **74**, 4091 (1995).
- [24] J. A. Jones and M. Mosca, *Implementation of a quantum algorithm on a nuclear magnetic resonance quantum computer*, The Journal of Chemical Physics **109**, 1648 (1998), arXiv:quant-ph/9801027.
- [25] I. L. Chuang, N. Gershenfeld, and M. Kubinec, *Experimental implementation of fast quantum searching*, Phys. Rev. Lett. **80**, 3408 (1998).
- [26] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, *Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance*, Nature **414**, 883 (2001), arXiv:quant-ph/0112176.
- [27] D. Lu, K. Li, J. Li, H. Katiyar, A. J. Park, G. Feng, T. Xin, H. Li, G. Long, A. Brodutch, J. Baugh, B. Zeng, and R. Laflamme, *Enhancing quantum control by bootstrapping a quantum processor of 12 qubits*, npj Quantum Information **3**, 10.1038/s41534-017-0045-z (2017), arXiv:1701.01198.
- [28] J. Li, Z. Luo, T. Xin, H. Wang, D. Kribs, D. Lu, B. Zeng, and R. Laflamme, *Experimental implementation of efficient quantum pseudorandomness on a 12-spin system*, Physical Review Letters **123**, 10.1103/physrevlett.123.030502 (2019), arXiv:1807.07419.
- [29] J. L. Dodd, M. A. Nielsen, M. J. Bremner, and R. T. Thew, *Universal quantum computation and simulation using any entangling Hamiltonian and local unitaries*, Phys. Rev. A **65**, 040301 (2002), arXiv:quant-ph/0106064.
- [30] M. A. Nielsen, M. J. Bremner, J. L. Dodd, A. M. Childs, and C. M. Dawson, *Universal simulation of Hamiltonian dynamics for quantum systems with finite-dimensional state spaces*, Phys. Rev. A **66**, 022317 (2002), arXiv:quant-ph/0109064.
- [31] D. W. Leung, I. L. Chuang, F. Yamaguchi, and Y. Yamamoto, *Efficient implementation of coupled logic gates for quantum computation*, Phys. Rev. A **61**, 042310 (2000), arXiv:quant-ph/9904100.
- [32] P. Baßler, M. Zipper, C. Cedzich, M. Heinrich, P. H. Huber, M. Johanning, and M. Kliesch, *Synthesis of and compilation with time-optimal multi-qubit gates*, Quantum **7**, 984 (2023), arXiv:2206.06387.
- [33] P. Baßler, M. Heinrich, and M. Kliesch, *Time-optimal multi-qubit gates: Complexity, efficient heuristic and gate-time bounds*, Quantum **8**, 1279 (2024), arXiv:2307.11160.
- [34] P. Baßler, M. Heinrich, and M. Kliesch, *General, efficient, and robust Hamiltonian engineering*, arXiv:2410.19903 (2025).
- [35] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, *A quantum engineer's guide to superconducting qubits*, Applied Physics Reviews **6**, 021318 (2019), arXiv:1904.06560.
- [36] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, *Superconducting qubits: Current state of play*, Annual Review of Condensed Matter Physics **11**, 369 (2020), arXiv:1905.13641.

- [37] P. Scholl, M. Schuler, H. J. Williams, A. A. Eberharter, D. Barredo, K.-N. Schymik, V. Lienhard, L.-P. Henry, T. C. Lang, T. Lahaye, A. M. Läuchli, and A. Browaeys, *Quantum simulation of 2D antiferromagnets with hundreds of Rydberg atoms*, Nature **595**, 233 (2021), arXiv:2012.12268.
- [38] F. Mintert and C. Wunderlich, *Ion-trap quantum logic using long-wavelength radiation*, Phys. Rev. Lett. **87**, 257904 (2001), arXiv:quant-ph/0104041.
- [39] C. Wunderlich, *Conditional spin resonance with trapped ions*, in *Laser Physics at the Limits*, edited by H. Figger, C. Zimmermann, and D. Meschede (Springer, 2002) pp. 261–273, arXiv:quant-ph/0111158.
- [40] T. Monz, P. Schindler, J. T. Barreiro, M. Chwalla, D. Nigg, W. A. Coish, M. Harlander, W. Hänsel, M. Hennrich, and R. Blatt, *14-qubit entanglement: Creation and coherence*, Phys. Rev. Lett. **106**, 130506 (2011), arXiv:1009.6126 [quant-ph].
- [41] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, *Mixed-state entanglement and quantum error correction*, Phys. Rev. A **54**, 3824 (1996), arXiv:quant-ph/9604024.
- [42] A. R. Calderbank and P. W. Shor, *Good quantum error-correcting codes exist*, Phys. Rev. A **54**, 1098 (1996), arXiv:quant-ph/9512032.
- [43] A. Steane, *Multiple-particle interference and quantum error correction*, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **452**, 2551 (1996), arXiv:quant-ph/9601029.
- [44] S. Aaronson and D. Gottesman, *Improved simulation of stabilizer circuits*, Phys. Rev. A **70**, 052328 (2004), arXiv:quant-ph/0406196.
- [45] S. P. Boyd and L. Vandenberghe, *Convex optimization* (Cambridge University Press, Cambridge, UK ; New York, 2004).
- [46] E. J. Candes and M. B. Wakin, *An introduction to compressive sampling*, IEEE Signal Processing Magazine **25**, 21 (2008).
- [47] I. Kozlov and A. Petukhov, *Sparse solutions of underdetermined linear systems*, in *Handbook of Geomathematics*, edited by W. Freeden, M. Z. Nashed, and T. Sonar (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010) pp. 1243–1259.
- [48] Y. Li, *Sparse representation for machine learning*, in *Advances in Artificial Intelligence*, edited by O. R. Zaïane and S. Zilles (Springer Berlin Heidelberg, Berlin, Heidelberg, 2013) pp. 352–357.
- [49] I. Bárány, *A generalization of Carathéodory’s theorem*, Discrete Mathematics **40**, 141 (1982).
- [50] I. E. Leonard and J. E. Lewis, *Convex hulls*, in *Geometry of Convex Sets* (John Wiley & Sons, 2015) pp. 146–155.
- [51] B. K. Natarajan, *Sparse approximate solutions to linear systems*, SIAM J. Comp. **24**, 227 (1995).
- [52] D. L. Donoho, *For most large underdetermined systems of linear equations the minimal ℓ_1 -norm solution is also the sparsest solution*, Communications on Pure and Applied Mathematics **59**, 797 (2006), <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.20132>.

- [53] D. Donoho and J. Tanner, *Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **367**, 4273 (2009), arXiv:<https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.2009.0152>.
- [54] M. Slater, *Lagrange multipliers revisited*, Cowles Foundation Discussion Papers **304** (1959).
- [55] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, in *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84 (Association for Computing Machinery, 1984) p. 302–311.
- [56] M. Awad and R. Khanna, *Support vector machines for classification*, in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers* (Apress, Berkeley, CA, 2015) pp. 39–66.
- [57] A. M. Rush and M. J. Collins, *A tutorial on dual decomposition and Lagrangian relaxation for inference in natural language processing*, J. Artif. Intell. Res. **45**, 305 (2012), arXiv:1405.5208.
- [58] D. A. Spielman and S.-H. Teng, *Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time*, Journal of the ACM **51**, 385 (2004), arXiv:cs/0111050.
- [59] E. Bach and S. Huiberts, *Optimal smoothed analysis of the simplex method*, arXiv:2504.04197 (2025).
- [60] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge University Press, 2010).
- [61] A. Kay, *Tutorial on the quantikz package*, arXiv:1809.03842 (2023).
- [62] M. Heinrich, *On stabiliser techniques and their application to simulation and certification of quantum devices*, Ph.d. thesis, University of Cologne (2021).
- [63] L. Clinton, T. S. Cubitt, R. Garcia-Patron, A. Montanaro, S. Stanisic, and M. Stroeks, *Quantum phase estimation without controlled unitaries*, arXiv:2410.21517 (2024).
- [64] A. Montanaro and S. Pallister, *Quantum algorithms and the finite element method*, Phys. Rev. A **93**, 032324 (2016), arXiv:1512.05903.
- [65] D. Jennings, M. Lostaglio, R. B. Lowrie, S. Pallister, and A. T. Sornborger, *The cost of solving linear differential equations on a quantum computer: fast-forwarding to explicit resource counts*, Quantum **8**, 1553 (2024), arXiv:2309.07881.
- [66] H. F. Trotter, *On the product of semi-groups of operators*, Proceedings of the American Mathematical Society **10**, 545 (1959).
- [67] M. Suzuki, *General theory of fractal path integrals with applications to many-body theories and statistical physics*, Journal of Mathematical Physics **32**, 400 (1991).
- [68] A. M. Childs and N. Wiebe, *Hamiltonian simulation using linear combinations of unitary operations*, Quantum Information and Computation **12**, 10.26421/qic12.11-12 (2012), arXiv:1202.5822.
- [69] G. H. Low, T. J. Yoder, and I. L. Chuang, *Methodology of resonant equiangular composite quantum gates*, Phys. Rev. X **6**, 041067 (2016), arXiv:1603.03996.
- [70] G. H. Low and I. L. Chuang, *Optimal Hamiltonian simulation by quantum signal processing*, Phys. Rev. Lett. **118**, 010501 (2017), arXiv:1606.02685.

- [71] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, *Toward the first quantum simulation with quantum speedup*, Proceedings of the National Academy of Sciences **115**, 9456–9461 (2018), arXiv:1711.10980.
- [72] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, G. Salton, S. Wang, and F. G. S. L. Brandão, *Quantum algorithms: A survey of applications and end-to-end complexities*, arXiv:2310.03011 (2023).
- [73] N. Hatano and M. Suzuki, *Finding exponential product formulas of higher orders*, in *Quantum Annealing and Other Optimization Methods*, edited by A. Das and B. K. Chakrabarti (Springer Berlin Heidelberg, Berlin, Heidelberg, 2005) pp. 37–68, arXiv:math-ph/0506007.
- [74] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, *Theory of Trotter error with commutator scaling*, Phys. Rev. X **11**, 011020 (2021), arXiv:1912.08854.
- [75] E. Campbell, *Random compiler for fast Hamiltonian simulation*, Phys. Rev. Lett. **123**, 070503 (2019), arXiv:1811.08017.
- [76] J. S. Waugh, *Average Hamiltonian theory*, in *eMagRes* (John Wiley & Sons, Ltd, 2007) <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470034590.emrstm0020>.
- [77] U. Haeberlen and J. S. Waugh, *Coherent averaging effects in magnetic resonance*, Phys. Rev. **175**, 453 (1968).
- [78] U. Haeberlen, *High Resolution NMR in Solids* (Academic Press, 1976).
- [79] W. Magnus, *On the exponential solution of differential equations for a linear operator*, Communications on Pure and Applied Mathematics **7**, 649 (1954).
- [80] P. C. Moan and J. Niesen, *Convergence of the Magnus series*, Foundations of Computational Mathematics **8**, 291 (2008), arXiv:0609198 [math].
- [81] A. Brinkmann, *Introduction to average Hamiltonian theory. I. Basics*, Concepts in Magnetic Resonance Part A **45A**, e21414 (2016).
- [82] C. L. Edmunds, C. Hempel, R. J. Harris, V. Frey, T. M. Stace, and M. J. Biercuk, *Dynamically corrected gates suppressing spatiotemporal error correlations as measured by randomized benchmarking*, Phys. Rev. Res. **2**, 013156 (2020), arXiv:1909.10727.
- [83] H.-J. Ding, B. Chu, B. Qi, and R.-B. Wu, *Collaborative learning of high-precision quantum control and tomography*, Phys. Rev. Appl. **16**, 014056 (2021).
- [84] M. Bando, T. Ichikawa, Y. Kondo, and M. Nakahara, *Concatenated composite pulses compensating simultaneous systematic errors*, Journal of the Physical Society of Japan **82**, 014004 (2013), arXiv:1209.4247.
- [85] S. Kukita, H. Kiya, and Y. Kondo, *Short composite quantum gate robust against two common systematic errors*, Journal of the Physical Society of Japan **91**, 104001 (2022), arXiv:2112.12945.
- [86] B. T. Torosov and N. V. Vitanov, *Composite pulses with errant phases*, Phys. Rev. A **100**, 023410 (2019), arXiv:1904.13168.
- [87] G. T. Genov, D. Schraft, T. Halfmann, and N. V. Vitanov, *Correction of arbitrary field errors in population inversion of quantum systems by universal composite pulses*, Phys. Rev. Lett. **113**, 043001 (2014), arXiv:1403.1201.

- [88] G. T. Genov, M. Hain, N. V. Vitanov, and T. Halfmann, *Universal composite pulses for efficient population inversion with an arbitrary excitation profile*, Phys. Rev. A **101**, 013827 (2020), arXiv:2002.08321.
- [89] H.-N. Wu, C. Zhang, J. Song, Y. Xia, and Z.-C. Shi, *Composite pulses for optimal robust control in two-level systems*, Phys. Rev. A **107**, 023103 (2023).
- [90] C. Kabytayev, T. J. Green, K. Khodjasteh, M. J. Biercuk, L. Viola, and K. R. Brown, *Robustness of composite pulses to time-dependent control noise*, Phys. Rev. A **90**, 012316 (2014), arXiv:1402.5174.
- [91] B. T. Torosov, S. S. Ivanov, and N. V. Vitanov, *Narrowband and passband composite pulses for variable rotations*, Phys. Rev. A **102**, 013105 (2020), arXiv:2004.08456.
- [92] S. Wimperis, *Broadband, narrowband, and passband composite pulses for use in advanced nmr experiments*, Journal of Magnetic Resonance, Series A **109**, 221 (1994).
- [93] H. K. Cummins, G. Llewellyn, and J. A. Jones, *Tackling systematic errors in quantum logic gates with composite rotations*, Phys. Rev. A **67**, 042308 (2003), arXiv:quant-ph/0208092.
- [94] H. L. Gevorgyan and N. V. Vitanov, *Ultrahigh-fidelity composite rotational quantum gates*, Phys. Rev. A **104**, 012609 (2021), arXiv:2012.14692.
- [95] J. M. Martinis and M. R. Geller, *Fast adiabatic qubit gates using only σ_z control*, Phys. Rev. A **90**, 022307 (2014), arXiv:1402.5467.
- [96] C. Rigetti and M. Devoret, *Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies*, Phys. Rev. B **81**, 134507 (2010).
- [97] S. Bravyi, O. Dial, J. M. Gambetta, D. Gil, and Z. Nazario, *The future of quantum computing with superconducting qubits*, Journal of Applied Physics **132**, 160902 (2022), arXiv:2209.06841.
- [98] K. Mølmer and A. Sørensen, *Multiparticle entanglement of hot trapped ions*, Phys. Rev. Lett. **82**, 1835 (1999), arXiv:quant-ph/9810040.
- [99] L. Heyfron and E. T. Campbell, *An efficient quantum compiler that reduces T count*, arXiv:1712.01557 (2018).
- [100] A. Kissinger and J. van de Wetering, *Reducing the number of non-Clifford gates in quantum circuits*, Phys. Rev. A **102**, 022406 (2020), arXiv:1903.10477.
- [101] V. V. Shende, I. L. Markov, and S. S. Bullock, *Minimal universal two-qubit controlled-not-based circuits*, Phys. Rev. A **69**, 062321 (2004), arXiv:quant-ph/0308033.
- [102] S. P. Jordan and P. Wocjan, *Efficient quantum circuits for arbitrary sparse unitaries*, Phys. Rev. A **80**, 062301 (2009), arXiv:0904.2211.
- [103] N. Robertson, A. Akhriev, J. Vala, and S. Zhuk, *Approximate quantum compiling for quantum simulation: A tensor network based approach*, ACM Transactions on Quantum Computing **6**, 10.1145/3731251 (2025), arXiv:2301.08609.
- [104] J. van de Wetering, *Zx-calculus for the working quantum computer scientist*, arXiv:2012.13966 (2020).
- [105] R. Duncan, A. Kissinger, S. Perdrix, and J. van de Wetering, *Graph-theoretic simplification of quantum circuits with the ZX-calculus*, Quantum **4**, 279 (2020), arXiv:1902.03178.

- [106] R. Vale, T. M. D. Azevedo, I. C. S. Araújo, I. F. Araujo, and A. J. da Silva, *Decomposition of multi-controlled special unitary single-qubit gates*, arXiv:2302.06377 (2023).
- [107] H. Yamasaki and M. Koashi, *Time-efficient constant-space-overhead fault-tolerant quantum computation*, Nature Physics **20**, 247 (2024), arXiv:2207.08826.
- [108] A. J. Scott, *Optimizing quantum process tomography with unitary 2-designs*, J. Phys. A **41**, 055308 (2008), arXiv:0711.1017 [quant-ph].
- [109] J. Helsen, I. Roth, E. Onorati, A. H. Werner, and J. Eisert, *A general framework for randomized benchmarking*, PRX Quantum **3**, 020357 (2022), arXiv:2010.07974 [quant-ph].
- [110] S. Bravyi and J. Haah, *Magic-state distillation with low overhead*, Phys. Rev. A **86**, 052329 (2012), arXiv:1209.2426.
- [111] J. O’Gorman and E. T. Campbell, *Quantum computation with realistic magic-state factories*, Phys. Rev. A **95**, 032338 (2017), arXiv:1605.07197.
- [112] D. Litinski, *Magic state distillation: Not as costly as you think*, arXiv:1905.06903 (2019).
- [113] C. Gidney, N. Shutty, and C. Jones, *Magic state cultivation: growing T states as cheap as CNOT gates*, arXiv:2409.17595 (2024).
- [114] H. Zhu, *Multiqubit Clifford groups are unitary 3-designs*, Phys. Rev. A **96**, 062336 (2017), arXiv:1510.02619 [quant-ph].
- [115] A. Ambainis, J. Bouda, and A. Winter, *Nonmalleable encryption of quantum information*, Journal of Mathematical Physics **50**, 042106 (2009), arXiv:0808.0353.
- [116] D. Maslov and B. Zindorf, *Depth optimization of CZ, CNOT, and Clifford circuits*, IEEE Transactions on Quantum Engineering **3**, 1 (2022), arxiv:2201.05215.
- [117] M. Y. Siraichi, V. F. d. Santos, C. Collange, and F. M. Q. Pereira, *Qubit allocation*, in *Proceedings of the 2018 International Symposium on Code Generation and Optimization*, CGO ’18 (Association for Computing Machinery, New York, NY, USA, 2018) p. 113–125.
- [118] S. Bravyi, D. Maslov, and Y. Nam, *Constant-cost implementations of Clifford operations and multiply-controlled gates using global interactions*, Phys. Rev. Lett. **129**, 230501 (2022), arXiv:2207.08691.
- [119] N. Bourbaki, *Lie groups and Lie algebras*, in *Elements of Mathematics* (Springer-Verlag, 2002) pp. XII–300.
- [120] J. Tits, *Buildings of Spherical Type and Finite BN-Pairs*, Lecture Notes in Mathematics, Vol. 386 (Springer, 1974).
- [121] R. A. Horn and C. R. Johnson, *Canonical forms for similarity and triangular factorizations*, in *Matrix Analysis* (Cambridge University Press, 1985) Chap. 3, pp. 163–224, <https://www.cambridge.org/core/books/matrix-analysis/9CF2CB491C9E97948B15FAD835EF9A8B>.
- [122] E. Farhi, J. Goldstone, and S. Gutmann, *A quantum approximate optimization algorithm*, arXiv:1411.4028 (2014).
- [123] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, *Variational quantum algorithms*, Nature Reviews Physics **3**, 625 (2021), arXiv:2012.09265.

- [124] K. Matsumoto and K. Amano, *Representation of quantum circuits with Clifford and $\pi/8$ gates*, arXiv:0806.3834 (2008).
- [125] I. H. Kim, *Catalytic z -rotations in constant T -depth*, arXiv:2506.15147 (2025).
- [126] P. Selinger, *Efficient Clifford+ T approximation of single-qubit operators*, arXiv:1212.6253 (2014).
- [127] V. Kliuchnikov, K. Lauter, R. Minko, A. Paetznick, and C. Petit, *Shorter quantum circuits via single-qubit gate approximation*, Quantum **7**, 1208 (2023), arXiv:2203.10064.
- [128] E. J. Gustafson, H. Lamm, D. Liu, E. M. Murairi, and S. Zhu, *Synthesis of single qubit circuits from Clifford+ R* , arXiv:2503.20203 (2025).
- [129] V. Kliuchnikov, D. Maslov, and M. Mosca, *Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and T gates*, Quantum Inform. Compu. **13**, 607 (2013), arXiv:1206.5236.
- [130] V. Kliuchnikov, A. Bocharov, M. Roetteler, and J. Yard, *A framework for approximating qubit unitaries*, arXiv:1510.03888 (2015).
- [131] A. Seif, H. Liao, V. Tripathi, K. Krsulich, M. Malekakhlagh, M. Amico, P. Jurcevic, and A. Javadi-Abhari, *Suppressing correlated noise in quantum computers via context-aware compiling*, in *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)* (2024) pp. 310–324, arXiv:2403.06852 .
- [132] Y. Nam, N. J. Ross, Y. Su, A. M. Childs, and D. Maslov, *Automated optimization of large quantum circuits with continuous parameters*, npj Quantum Information **4**, 23 (2018), arXiv:1710.07345.
- [133] A. Kissinger and J. van de Wetering, *PyZX: Large scale automated diagrammatic reasoning*, Electronic Proceedings in Theoretical Computer Science **318**, 229–241 (2020), arXiv:1904.04735.
- [134] M. Xu, Z. Li, O. Padon, S. Lin, J. Pointing, A. Hirth, H. Ma, J. Palsberg, A. Aiken, U. A. Acar, and Z. Jia, *Quartz: superoptimization of quantum circuits*, in *Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation, PLDI 2022* (Association for Computing Machinery, New York, NY, USA, 2022) p. 625–640, arXiv:2204.09033 .
- [135] A. Xu, A. Molavi, L. Pick, S. Tannu, and A. Albarghouthi, *Synthesizing quantum-circuit optimizers*, Proc. ACM Program. Lang. **7**, 10.1145/3591254 (2023), arXiv:2211.09691.
- [136] C. J. Ballance, T. P. Harty, N. M. Linke, M. A. Sepiol, and D. M. Lucas, *High-fidelity quantum logic gates using trapped-ion hyperfine qubits*, Phys. Rev. Lett. **117**, 060504 (2016), arXiv:1512.04600.
- [137] J. W. Britton, B. C. Sawyer, A. C. Keith, C.-C. J. Wang, J. K. Freericks, H. Uys, M. J. Biercuk, and J. J. Bollinger, *Engineered two-dimensional ising interactions in a trapped-ion quantum simulator with hundreds of spins*, Nature **484**, 489 (2012), arXiv:1204.5789.
- [138] C. Piltz, T. Sriarunothai, A. F. Varón, and C. Wunderlich, *A trapped-ion-based quantum byte with 10^{-5} next-neighbour cross-talk*, Nat. Commun. **5**, 4679 (2014), arXiv:1403.8043.
- [139] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis, *Demonstration of the trapped-ion quantum CCD computer architecture*, Nature **592**, 209–213 (2021), arXiv:2003.01293.

- [140] M. Malinowski, D. Allcock, and C. Ballance, *How to wire a 1000-qubit trapped-ion quantum computer*, PRX Quantum **4**, 040313 (2023), arXiv:2305.12773.
- [141] M. Morgado and S. Whitlock, *Quantum simulation and computing with Rydberg-interacting qubits*, AVS Quantum Science **3**, 10.1116/5.0036562 (2021), arXiv:2011.03031.
- [142] L. Henriët, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, *Quantum computing with neutral atoms*, Quantum **4**, 327 (2020), arXiv:2006.12326.
- [143] H. Levine, A. Keesling, G. Semeghini, A. Omran, T. T. Wang, S. Ebadi, H. Bernien, M. Greiner, V. Vuletić, H. Pichler, and M. D. Lukin, *Parallel implementation of high-fidelity multiqubit gates with neutral atoms*, Physical Review Letters **123**, 10.1103/physrevlett.123.170503 (2019), arXiv:1908.06101.
- [144] S. J. Evered, D. Bluvstein, M. Kalinowski, S. Ebadi, T. Manovitz, H. Zhou, S. H. Li, A. A. Geim, T. T. Wang, N. Maskara, H. Levine, G. Semeghini, M. Greiner, V. Vuletić, and M. D. Lukin, *High-fidelity parallel entangling gates on a neutral-atom quantum computer*, Nature **622**, 268 (2023), arXiv:2304.05420.
- [145] D. Bluvstein, H. Levine, G. Semeghini, T. T. Wang, S. Ebadi, M. Kalinowski, A. Keesling, N. Maskara, H. Pichler, M. Greiner, V. Vuletić, and M. D. Lukin, *A quantum processor based on coherent transport of entangled atom arrays*, Nature **604**, 451 (2022), arXiv:2112.03923.
- [146] H. Levine, A. Keesling, A. Omran, H. Bernien, S. Schwartz, A. S. Zibrov, M. Endres, M. Greiner, V. Vuletić, and M. D. Lukin, *High-fidelity control and entanglement of rydberg-atom qubits*, Phys. Rev. Lett. **121**, 123603 (2018), arXiv:1806.04682.
- [147] S. de Léséleuc, D. Barredo, V. Lienhard, A. Browaeys, and T. Lahaye, *Optical control of the resonant dipole-dipole interaction between Rydberg atoms*, Phys. Rev. Lett. **119**, 053202 (2017), arXiv:1705.03293.
- [148] S. Jandura and G. Pupillo, *Time-Optimal Two- and Three-Qubit Gates for Rydberg Atoms*, Quantum **6**, 712 (2022), arXiv:2202.00903.
- [149] L. Schmid, D. F. Locher, M. Rispler, S. Blatt, J. Zeiher, M. Müller, and R. Wille, *Computational capabilities and compiler development for neutral atom quantum processors—connecting tool developers and hardware experts*, Quantum Sci. Technol. **9**, 10.1088/2058-9565/ad33ac (2024), arXiv:2309.08656.
- [150] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, *Superconducting qubits: Current state of play*, Annual Review of Condensed Matter Physics **11**, 369 (2020), arXiv:1905.13641.
- [151] J. Van Damme, S. Massar, R. Acharya, T. Ivanov, D. Perez Lozano, Y. Canel, M. Demarets, D. Vangoidsenhoven, Y. Hermans, J. G. Lai, A. M. Vadiraj, M. Mongillo, D. Wan, J. De Boeck, A. Potočnik, and K. De Greve, *Advanced cmos manufacturing of superconducting qubits on 300 mm wafers*, Nature **634**, 74 (2024).
- [152] J. Majer, J. M. Chow, J. M. Gambetta, J. Koch, B. R. Johnson, J. A. Schreier, L. Frunzio, D. I. Schuster, A. A. Houck, A. Wallraff, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, *Coupling superconducting qubits via a cavity bus*, Nature **449**, 443–447 (2007), arXiv:0709.2135.
- [153] L. DiCarlo, J. M. Chow, J. M. Gambetta, L. S. Bishop, B. R. Johnson, D. I. Schuster, J. Majer, A. Blais, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf, *Demonstration of two-qubit algorithms with a superconducting quantum processor*, Nature **460**, 240–244 (2009), arXiv:0903.2030.

- [154] E. Magesan and J. M. Gambetta, *Effective Hamiltonian models of the cross-resonance gate*, Physical Review A **101**, 10.1103/physreva.101.052308 (2020), arXiv:1804.04073.
- [155] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, *Evidence for the utility of quantum computing before fault tolerance*, Nature **618**, 500 (2023).
- [156] L. Abdurakhimov, J. Adam, H. Ahmad, O. Ahonen, M. Algaba, G. Alonso, V. Bergholm, R. Beriwal, M. Beuerle, C. Bockstiegel, A. Calzona, C. F. Chan, D. Cucurachi, S. Dahl, R. Davletkaliyev, O. Fedorets, A. G. Frieiro, Z. Gao, J. Guldmyr, A. Guthrie, J. Hassel, H. Heimonen, J. Heinsoo, T. Hiltunen, K. Holland, J. Hotari, H. Hsu, A. Huhtala, E. Hyypä, A. Hämäläinen, J. Ikonen, S. Inel, D. Janzso, T. Jaakkola, M. Jenei, S. Jolin, K. Juliusson, J. Jussila, S. Khalid, S.-G. Kim, M. Koistinen, R. Kokkonen, A. Komlev, C. Ockeloen-Korppi, O. Koskinen, J. Kotilahti, T. Kuisma, V. Kukushkin, K. Kumpulainen, I. Kuronen, J. Kylvälä, N. Lamponen, J. Lamprich, A. Landra, M. Leib, T. Li, P. Liebermann, A. Lintunen, W. Liu, J. Luus, F. Marxer, A. M. van de Griend, K. Mitra, J. K. Moqadam, J. Mrozek, H. Mäkynen, J. Mäntylä, T. Naaranoja, F. Nappi, J. Niemi, L. Ortega, M. Palma, M. Papič, M. Partanen, J. Penttilä, A. Plyushch, W. Qiu, A. Rath, K. Repo, T. Riipinen, J. Ritvas, P. F. Romero, J. Ruoho, J. Rabinä, S. Saarinen, I. Sagar, H. Sargsyan, M. Sarsby, N. Savola, M. Savytskyi, V. Selinmaa, P. Smirnov, M. M. Suárez, L. Sundström, S. Szupinska, E. Takala, I. Takmakov, B. Tarasinski, M. Thapa, J. Tiainen, F. Tosto, J. Tuorila, C. Valenzuela, D. Vasey, E. Vehmaanperä, A. Vepsäläinen, A. Vienamo, P. Vesanen, A. Välimaa, J. Wesdorp, N. Wurz, E. Wybo, L. Yang, and A. Yurtalan, *Technology and performance benchmarks of IQM's 20-qubit quantum computer* (2024), arXiv:2408.12433 .
- [157] J. M. Gambetta, J. M. Chow, and M. Steffen, *Building logical qubits in a superconducting quantum computing system*, npj Quantum Information **3**, 10.1038/s41534-016-0004-0 (2017), arXiv:1510.04375.
- [158] Z. Du, S. Kan, S. Stein, Z. Liang, A. Li, and Y. Mao, *Hardware-aware compilation for chip-to-chip coupler-connected modular quantum systems*, arXiv:2505.09036 (2025).
- [159] J. S. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. S. Fried, S. Hong, P. Karalekas, C. B. Osborn, A. Papageorge, E. C. Peterson, G. Prawiroatmodjo, N. Rubin, C. A. Ryan, D. Scarabelli, M. Scheer, E. A. Sete, P. Sivarajah, R. S. Smith, A. Staley, N. Tezak, W. J. Zeng, A. Hudson, B. R. Johnson, M. Reagor, M. P. da Silva, and C. Rigetti, *Unsupervised machine learning on a hybrid quantum computer*, arXiv:1712.05771 (2017).
- [160] IBM Quantum, *ibmq_rochester* (2021), <https://quantum.cloud.ibm.com/>.
- [161] G. J. Mooney, *A fast and adaptable algorithm for optimal multi-qubit pathfinding in quantum circuit compilation*, arXiv:2405.18785 (2024).
- [162] B. Tan and J. Cong, *Optimal layout synthesis for quantum computing*, in *Proceedings of the 39th International Conference on Computer-Aided Design*, ICCAD '20 (Association for Computing Machinery, New York, NY, USA, 2020) arXiv:2007.15671 .
- [163] C. Zhang, A. B. Hayes, L. Qiu, Y. Jin, Y. Chen, and E. Z. Zhang, *Time-optimal qubit mapping*, in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '21 (Association for Computing Machinery, New York, NY, USA, 2021) p. 360–374.

- [164] I. Shaik and J. van de Pol, *Optimal Layout Synthesis for Deep Quantum Circuits on NISQ Processors with 100+ Qubits*, in *27th International Conference on Theory and Applications of Satisfiability Testing (SAT 2024)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 305, edited by S. Chakraborty and J.-H. R. Jiang (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2024) pp. 26:1–26:18, arXiv:2403.11598 .
- [165] G. Li, Y. Ding, and Y. Xie, *Tackling the qubit mapping problem for NISQ-Era quantum devices*, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '19* (Association for Computing Machinery, New York, NY, USA, 2019) p. 1001–1014, arXiv:1809.02573 .
- [166] H. Zou, M. Treinish, K. Hartman, A. Ivrii, and J. Lishman, *LightSABRE: A lightweight and enhanced SABRE algorithm*, arXiv:2409.08368 (2024).
- [167] Y. Huang, X. Zhou, F. Meng, and S. Li, *Qubit mapping: The adaptive Divide-and-Conquer approach*, arXiv:2409.04752 (2024).
- [168] M. Amy, D. Maslov, and M. Mosca, *Polynomial-time T -depth optimization of Clifford+ T circuits via matroid partitioning*, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **33**, 1476 (2014), arXiv:1303.2042.
- [169] M. Amy, P. Azimzadeh, and M. Mosca, *On the CNOT-complexity of CNOT-PHASE circuits*, Quantum Sci. Technol. **4**, 015002 (2018), arXiv:1712.01859.
- [170] B. Coecke and R. Duncan, *Interacting quantum observables: categorical algebra and diagrammatics*, New Journal of Physics **13**, 043016 (2011), arXiv:0906.4725.
- [171] N. de Beaudrap, A. Kissinger, and J. van de Wetering, *Circuit Extraction for ZX-Diagrams Can Be $\#P$ -Hard*, in *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 229, edited by M. Bojańczyk, E. Merelli, and D. P. Woodruff (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2022) pp. 119:1–119:19, arXiv:2202.09194 .
- [172] A. Xu, A. Molavi, S. Tannu, and A. Albarghouthi, *Optimizing quantum circuits, fast and slow*, arXiv:2411.04104 (2024).
- [173] S. J. Beale and J. J. Wallman, *Randomized compiling in fault-tolerant quantum computation*, arXiv:2306.13752 (2023).
- [174] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, *Encoding electronic spectra in quantum circuits with linear T complexity*, Physical Review X **8**, 10.1103/physrevx.8.041015 (2018), arXiv:1805.03662.
- [175] Y. R. Sanders, D. W. Berry, P. C. Costa, L. W. Tessler, N. Wiebe, C. Gidney, H. Neven, and R. Babbush, *Compilation of fault-tolerant quantum heuristics for combinatorial optimization*, PRX Quantum **1**, 10.1103/prxquantum.1.020312 (2020), arXiv:2007.07391.
- [176] R. Blatt and C. F. Roos, *Quantum simulations with trapped ions*, Nat. Phys. **8**, 277 (2012).
- [177] J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, P. Becker, H. Kaplan, A. V. Gorshkov, Z.-X. Gong, and C. Monroe, *Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator*, Nature **551**, 601 (2017), arXiv:1708.01044.
- [178] N. Friis, O. Marty, C. Maier, C. Hempel, M. Holzäpfel, P. Jurcevic, M. B. Plenio, M. Huber, C. Roos, R. Blatt, and B. Lanyon, *Observation of entangled states of a fully controlled 20-qubit system*, Phys. Rev. X **8**, 021012 (2018), arXiv:1711.11092.

- [179] S. Ebadi, T. T. Wang, H. Levine, A. Keesling, G. Semeghini, A. Omran, D. Bluvstein, R. Samajdar, H. Pichler, W. W. Ho, S. Choi, S. Sachdev, M. Greiner, V. Vuletić, and M. D. Lukin, *Quantum phases of matter on a 256-atom programmable quantum simulator*, Nature **595**, 227 (2021), arXiv:2012.12281.
- [180] H. Hahn, G. Zarantonello, M. Schulte, A. Bautista-Salvador, K. Hammerer, and C. Ospelkaus, *Integrated 9be+ multi-qubit gate device for the ion-trap quantum computer*, npj Quantum Information **5**, 70 (2019).
- [181] I. N. Moskalenko, I. A. Simakov, N. N. Abramov, A. A. Grigorev, D. O. Moskalev, A. A. Pishchimova, N. S. Smirnov, E. V. Zikiy, I. A. Rodionov, and I. S. Besedin, *High fidelity two-qubit gates on fluxoniums using a tunable coupler*, npj Quantum Information **8**, 130 (2022), arXiv:2203.16302.
- [182] J. Choi, H. Zhou, H. S. Knowles, R. Landig, S. Choi, and M. D. Lukin, *Robust dynamic Hamiltonian engineering of many-body spin systems*, Phys. Rev. X **10**, 031002 (2020), arXiv:1907.03771.
- [183] S. Choi, N. Y. Yao, and M. D. Lukin, *Dynamical engineering of interactions in qudit ensembles*, Phys. Rev. Lett. **119**, 183603 (2017), arXiv:1703.09808.
- [184] H. Zhou, H. Gao, N. T. Leitao, O. Makarova, I. Cong, A. M. Douglas, L. S. Martin, and M. D. Lukin, *Robust Hamiltonian engineering for interacting qudit systems*, arXiv:2305.09757 (2023).
- [185] T. Calarco, U. Dorner, P. S. Julienne, C. J. Williams, and P. Zoller, *Quantum computations with atoms in optical lattices: Marker qubits and molecular interactions*, Phys. Rev. A **70**, 012306 (2004), arXiv:quant-ph/0403197.
- [186] C. Weitenberg, M. Endres, J. F. Sherson, M. Cheneau, P. Schauß, T. Fukuhara, I. Bloch, and S. Kuhr, *Single-spin addressing in an atomic Mott insulator*, Nature **471**, 319 (2011), arXiv:1101.2076.
- [187] S. Geier, N. Thaicharoen, C. Hainaut, T. Franz, A. Salzinger, A. Tebben, D. Grimshandl, G. Zürn, and M. Weidemüller, *Floquet Hamiltonian engineering of an isolated many-body spin system*, Science **374**, 1149 (2021), arXiv:2105.01597.
- [188] D. Leung, *Simulation and reversal of n -qubit Hamiltonians using Hadamard matrices*, Journal of Modern Optics **49**, 1199 (2002), arXiv:quant-ph/0107041.
- [189] D. Hayes, S. T. Flammia, and M. J. Biercuk, *Programmable quantum simulation by dynamic Hamiltonian engineering*, New J. Phys. **16**, 10.1088/1367-2630/16/8/083027 (2014), arXiv:1309.6736.
- [190] A. Parra-Rodriguez, P. Lougovski, L. Lamata, E. Solano, and M. Sanz, *Digital-analog quantum computation*, Phys. Rev. A **101**, 10.1103/PhysRevA.101.022305 (2020), arXiv:1812.03637.
- [191] M. Garcia-de Andoin, A. Saiz, P. Pérez-Fernández, L. Lamata, I. Oregi, and M. Sanz, *Digital-analog quantum computation with arbitrary two-body Hamiltonians*, Phys. Rev. Res. **6**, 013280 (2024), arXiv:2307.00966.
- [192] M. Votto, J. Zeiher, and B. Vermersch, *Universal quantum processors in spin systems via robust local pulse sequences*, Quantum **8**, 1513 (2024), arXiv:2311.10600 [quant-ph].
- [193] I. N. Hincks, C. E. Granade, T. W. Borneman, and D. G. Cory, *Controlling quantum devices with nonlinear hardware*, Phys. Rev. Appl. **4**, 024012 (2015).

- [194] M. A. Rol, L. Ciorciaro, F. K. Malinowski, B. M. Tarasinski, R. E. Sagastizabal, C. C. Bultink, Y. Salathe, N. Haandbaek, J. Sedivy, and L. DiCarlo, *Time-domain characterization and correction of on-chip distortion of control pulses in a quantum processor*, Applied Physics Letters **116**, 054001 (2020), arXiv:1907.04818.
- [195] V. P. Canelles, M. G. Algaba, H. Heimonen, M. Papič, M. Ponce, J. Rönkkö, M. J. Thapa, I. de Vega, and A. Auer, *Benchmarking Digital-Analog Quantum Computation*, arXiv:2307.07335 (2023).
- [196] H. Zhou, J. Choi, S. Choi, R. Landig, A. M. Douglas, J. Isoya, F. Jelezko, S. Onoda, H. Sumiya, P. Cappellaro, H. S. Knowles, H. Park, and M. D. Lukin, *Quantum metrology with strongly interacting spin systems*, Phys. Rev. X **10**, 031003 (2020), arXiv:1907.10066.
- [197] E. A. Martinez, T. Monz, D. Nigg, P. Schindler, and R. Blatt, *Compiling quantum algorithms for architectures with multi-qubit gates*, New J. Phys. **18**, 063029 (2016), arXiv:1601.06819.
- [198] D. Maslov and Y. Nam, *Use of global interactions in efficient quantum circuit constructions*, New J. Phys. **20**, 10.1088/1367-2630/aaa398 (2018), arXiv:1707.06356.
- [199] J. van de Wetering, *Constructing quantum circuits with global gates*, New J. Phys. **23**, 043015 (2021), arXiv:2012.09061.
- [200] K. Blekos, D. Brand, A. Ceschini, C.-H. Chou, R.-H. Li, K. Pandya, and A. Summer, *A review on Quantum Approximate Optimization Algorithm and its variants*, Physics Reports **1068**, 1 (2024), a review on Quantum Approximate Optimization Algorithm and its variants, arXiv:2306.09198.
- [201] S. S. Ivanov, M. Johanning, and C. Wunderlich, *Simplified implementation of the quantum Fourier transform with Ising-type Hamiltonians: Example with ion traps*, arXiv:1503.08806 (2015).
- [202] C. Piltz, T. Sriarunothai, S. S. Ivanov, S. Wölk, and C. Wunderlich, *Versatile microwave-driven trapped ion spin system for quantum information processing*, Sci. Adv. **2**, e1600093 (2016), arXiv:1509.01478.
- [203] J. Cohn, M. Motta, and R. M. Parrish, *Quantum filter diagonalization with compressed double-factorized Hamiltonians*, PRX Quantum **2**, 040352 (2021), arXiv:2104.08957.
- [204] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal on Computing **26**, 1484 (1997), arXiv:quant-ph/9508027.
- [205] D. S. Abrams and S. Lloyd, *Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors*, Phys. Rev. Lett. **83**, 5162 (1999).
- [206] Y. Su, D. W. Berry, N. Wiebe, N. Rubin, and R. Babbush, *Fault-tolerant quantum simulations of chemistry in first quantization*, PRX Quantum **2**, 040332 (2021), arXiv:2105.12767.
- [207] R. M. Parrish and P. L. McMahon, *Quantum filter diagonalization: Quantum eigendecomposition without full quantum phase estimation*, arXiv:1909.08925 (2019).
- [208] I. D. Kivlichan, J. McClean, N. Wiebe, C. Gidney, A. Aspuru-Guzik, G. K.-L. Chan, and R. Babbush, *Quantum simulation of electronic structure with linear depth and connectivity*, Phys. Rev. Lett. **120**, 110501 (2018), arXiv:1711.04789.

- [209] M. Motta, E. Ye, J. R. McClean, Z. Li, A. J. Minnich, R. Babbush, and G. K.-L. Chan, *Low rank representations for quantum simulation of electronic structure*, npj Quantum Information **7**, 83 (2021), arXiv:1808.02625.
- [210] S. Nagies, K. T. Geier, J. Akram, J. Okamoto, D. Bantounas, C. Wunderlich, M. Johanning, and P. Hauke, *The role of higher-order terms in trapped-ion quantum computing with magnetic gradient induced coupling*, Quantum Science and Technology **10**, 025051 (2025), arXiv:2409.10498.
- [211] J. G. Wendel, *A problem in geometric probability.*, MATHEMATICA SCANDINAVICA **11**, 109–112 (1962).
- [212] N. B. Karahanoğlu, H. Erdoğan, and Ş. İ. Birbil, *A mixed integer linear programming formulation for the sparse recovery problem in compressed sensing*, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013) pp. 5870–5874.
- [213] MOSEK ApS, *MOSEK Optimizer API for Python 9.3.14* (2022).
- [214] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual* (2024).

Chapter 6

Appendix

A Paper - Synthesis of and compilation with time-optimal multi-qubit gates

Title: Synthesis of and compilation with time-optimal multi-qubit gates

Authors: Pascal Baßler, Matthias Zipper, Christopher Cedzich, Markus Heinrich, Patrick Heinz Huber, Michael Johanning, Martin Kliesch

Journal: Quantum

Publication status: Published

Contribution by PB: Main author (input approx 60%)

A summary of this publication is presented in section 4.2.

I brought forward the initial idea to formulate an optimization problem for modifying the couplings of an Ising Hamiltonian by single-qubit gates. MK provided important insights from frame theory and compressed sensing which helped with the understanding of the initial formulation. CC helped to simplify and restructure this initial formulation. MH and I developed the linear program formulation. MH further wrote Appendix C, providing results from convex optimizations, and the error analysis. MZ, PHH and MJ provided invaluable information about quantum computing with microwave controlled ion traps, which is summarized in chapter 1 and discussed in more detail in Appendix A. I derived the quantum compilation strategies, with suggestions from my co-authors in periodic discussions. MZ developed the code to calculate the couplings in the Ising Hamiltonian, and I developed the rest of the code. I wrote an initial draft of the manuscript, and parts of the introduction and main text were added or rewritten by my co-authors. Finally, the manuscript was proofread by all authors and several paragraphs in the main text were jointly edited.

Synthesis of and compilation with time-optimal multi-qubit gates

P. Baßler¹, M. Zipper¹, C. Cedzich¹, M. Heinrich¹, P. H. Huber², M. Johanning², and M. Kliesch^{1,3}

¹ Institute for Theoretical Physics, Heinrich-Heine-Universität Düsseldorf, Germany

² Department of Physics, School of Science and Technology, University of Siegen, Germany

³ Institute for Quantum and Quantum Inspired Computing, Hamburg University of Technology, Germany

We develop a method to synthesize a class of entangling multi-qubit gates for a quantum computing platform with fixed Ising-type interaction with all-to-all connectivity. The only requirement on the flexibility of the interaction is that it can be switched on and off for individual qubits. Our method yields a time-optimal implementation of the multi-qubit gates. We numerically demonstrate that the total multi-qubit gate time scales approximately linear in the number of qubits. Using this gate synthesis as a subroutine, we provide compilation strategies for important use cases: (i) we show that any Clifford circuit on n qubits can be implemented using at most $2n$ multi-qubit gates without requiring ancilla qubits, (ii) we decompose the quantum Fourier transform in a similar fashion, (iii) we compile a simulation of molecular dynamics, and (iv) we propose a method for the compilation of diagonal unitaries with time-optimal multi-qubit gates, as a step towards general unitaries. As motivation, we provide a detailed discussion on a microwave controlled ion trap architecture with magnetic gradient-induced coupling (MAGIC) for the generation of the Ising-type interactions.

1 Introduction

In order to run a program on any computing platform, it is necessary to decompose its higher-level logical operations into more elementary ones and eventually translate those into the platform's native instruction set. This process is called “compiling.” Both for classical and quantum computers, this is a non-trivial task. The performance of the compiled program depends not only on the optimizations done by the compiler but also on the available instructions and their implementation.

Especially in the era of noisy and intermediate-scale quantum (NISQ) devices, quantum algorithms are limited by the coherence time of the noisy qubits and the number of noisy gates needed to run them [1]. Thus, it is imperative not only to improve the current quantum devices but also to design fast gates and optimized compilers that use the specific architecture's peculiarities to reduce the circuit depth. Moreover, these endeavors help to reduce the noise-levels of physical gates and are thus also important for reducing the overhead in quantum error correction [2].

Quantum compilation is further complicated by the fact that the type and performance of the native instructions depend severely on the available physical interactions and the extent to which they can be controlled. Most of the compiling literature has focused on native instructions given by single and two-qubit gates. Two-qubit gates are arguably the simplest entangling gates that can be experimentally realized and dominate in most quantum computing architectures, such as supercomputing qubits.

In contrast, ion trap quantum computers naturally involve all-to-all interactions and thus allow for the realization of multi-qubit gates which entangle multiple qubits simultaneously [3, 4]. Consequently, there has been a growing interest in studying compilation with multi-qubit gates, and advantages over the use of two-qubit gates have been demonstrated [5–10].

P. Baßler: bassler@hhu.de

M. Kliesch: science@mkliesch.eu

The experimental realization of multi-qubit gates in ion trap quantum computers remains an active field of research. Recent proof-of-principle experiments have demonstrated such gates acting on up to 10 qubits [11–13]. These rely on precomputing and controlling rather complicated laser pulse shapes to physically implement the desired interactions.

In this work, we propose a simple method that uses *some* all-to-all interaction to emulate arbitrary couplings. We use this idea to synthesize time-optimal multi-qubit gates under minimal experimental setup and control hardware assumptions.

Concretely, we consider a quantum computing platform that satisfies the following requirements:

- (I) single-qubit rotations can be executed in parallel, and
- (II) it offers Ising-type interactions with all-to-all connectivity.

We also develop compilation strategies with these gates, for which we additionally require that

- (III) there is a way to exclude specific qubits from participating in the interaction.

This assumption is sufficient to guarantee that unitaries can be compiled in a circuit depth depending only on the size of their support.

The requirements (I)–(III) are satisfied, for example, in ion trap platforms [11–13]. The motivation for our research originates from working with an ion trap where all gate control is based on microwave pulses and where Ising-type interactions with all-to-all connectivity are mediated through magnetic gradient-induced coupling (MAGIC) [14–21], see Section 1.4.

For concreteness, we assume that all Ising interactions are of ZZ type, and we call the multi-qubit gates generated by arbitrary ZZ couplings ‘GZZ gates’. Furthermore, by requirement (II), there is an Ising Hamiltonian H with fixed ZZ interactions. We then present a synthesis method which realizes an arbitrary GZZ gate as a sequence of time evolutions under H , interleaved with suitable X layers. The purpose of these X layers is to temporarily flip the signs of some ZZ coupling terms in H to accumulate the desired coupling over the sequence. We show that such a sequence can always be found and use a linear program (LP) to find a time-optimal realization of the desired GZZ gate. The resulting gate time scales approximately linear with the number of participating qubits n and requires at most $n(n-1)/2$ X layers.

This method may produce very short evolution times that can lead to problematically crammed single-qubit rotations in practice. We address this issue with a variation of our approach that extends the LP to a mixed integer program (MIP).

We proceed by developing several compiling strategies with GZZ gates. We show that any Clifford circuit on n qubits can be implemented using at most $n+1$ GZZ gates, n two-qubit gates and few single-qubit gates. As an example for non-Clifford unitaries, we decompose the quantum Fourier transform (QFT) in a similar fashion into $n/2$ GZZ gates, $n/2$ two-qubit gates and single-qubit gates. An important application of quantum computers is the simulation of molecular dynamics. We present a method to tailor the approximate simulation in Ref. [22] to our setup by compiling layers of Givens rotations into time-optimal GZZ gates. This method significantly reduces the required number of single-qubit rotations with arbitrary small angles, which are challenging to implement in practice. Moreover, we propose a compilation method for diagonal unitaries as a step toward compilation strategies for general unitaries.

1.1 Comparison to previous works

Synthesis of multi-qubit gates. Previous works [11–13] have mainly focused on implementing multi-qubit gates on ion trap quantum computers using the laser-controlled Mølmer-Sørensen (MS) mechanism [23–25]. This setup requires segmented, amplitude-modulated laser pulses, the shape of which can be efficiently precomputed using the efficient, arbitrary, simultaneously entangling (EASE) gate implementation [13].

Here, the novelty of our work is that we only require the engineering of a single, fixed Ising Hamiltonian, which can be calibrated and fine-tuned to high accuracy. This situation can be found in MAGIC ion traps [14–21] but may also serve as a practical design principle for other architectures. With our synthesis method, multi-qubit GZZ gates can be realized using only additional X gates, resulting in a sequential series of simple pulses. Arguably, this requires less

fine-grained control of the pulse shapes than the EASE gate protocol [13] and may thus be more implementation-friendly. However, we leave a detailed comparison of the approaches for future experimental work.

Furthermore, we introduce *gate time*, instead of gate count, as the central metric for our synthesis of multi-qubit gates. This metric is meaningful, especially for NISQ devices, since the execution of circuits is limited by the coherence time of the qubits. As we show, a side effect of our method is that it also produces rather short circuits, but not necessarily the shortest ones. From our numerical studies, we expect that the gate time of our approach scales at most linear with the number of *participating* qubits n . Hence, we expect our method to produce faster multi-qubit gates than the EASE gate protocol, which additionally scales linearly in the *total* number of qubits N .

A conceptually related approach to our synthesis method was presented in Ref. [26] in the context of digital-analog quantum computing (DAQC). There, the gate synthesis is based on solving a system of linear equations and is inherently restricted to X layers acting on at most two qubits. In contrast, our work optimizes for the total gate time of the sequence and, to this end, allows for layers with arbitrary support. In this way, we avoid the problem of negative times encountered in Ref. [26] and observe a gate time scaling approximately linear in n , in contrast to the quadratic scaling in Ref. [26].

Compilation with GZZ gates. A strategy to decompose general unitaries with multi-qubit GZZ gates is presented in Ref. [6]. It is based on maximizing the fidelity while using as few multi-qubit gates as possible. This optimization is computationally costly, so the numerical results in Ref. [6] cover only up to 4 qubits.

Different compiling strategies with multi-qubit GZZ gates have recently been investigated for Clifford unitaries. In Ref. [7], an implementation with $12n - 18$ GZZ gates is reported, which has been improved to $6n - 8$ GZZ gates in Ref. [8]. Subsequently, it was shown that $6 \log(n) + O(1)$ GZZ gates are enough if $n/2$ ancillary qubits are used [9]. Here, our ancilla-free approach reduces the prefactor because it requires at most $n + 1$ multi-qubit GZZ gates and n two-qubit gates. Shortly after publishing the preprint of this work, it was shown in Ref. [10] that any Clifford unitary on n qubits can, in fact, be implemented with at most 26 so-called GCZ gates which are equivalent to GZZ up to single-qubit Z rotations. In Ref. [10], the authors also pointed out that the results in Ref. [27] can be used to obtain an ancilla-free implementation with $2 \log(n) + O(1)$ GZZ gates. The constant-depth scheme of Ref. [10] can be readily combined with the time-optimal synthesis of GZZ gates discussed in Section 2 to show that any Clifford unitary can be realized in linear time on a platform satisfying the requirements (I)–(III). For large n this would further reduce the number of required GZZ gates. However, for small $n \leq 13$ the compilation method presented in Section 3.2 may still be advantageous.

Refs. [19, 28] propose a hand-tailored implementation of the quantum Fourier transform on three qubits that uses simultaneous Ising-type interactions to achieve a speed-up. We use the same interactions, but our scheme can be applied to systems of arbitrary size and employs our time-optimal multi-qubit gates (cf. Section 3.3.)

1.2 Outline

The remainder of the paper is structured as follows: We close this introductory section with a brief introduction to the computational primitives obtained from the requirements (I)–(III) and how these are realized in microwave controlled ion traps with MAGIC. In Section 2, we introduce our GZZ gate and the time-optimal gate synthesis method. Also, we define the MIP to solve the problem of too short Ising-evolution times and support our claim of time-optimality with numerical results. In Section 3 we present compiling strategies with our multi-qubit gate for Clifford circuits, the QFT, molecular dynamics, and general diagonal unitaries. Moreover, we demonstrate the performance of these compilation schemes with numerical results for the Clifford circuits and the QFT.

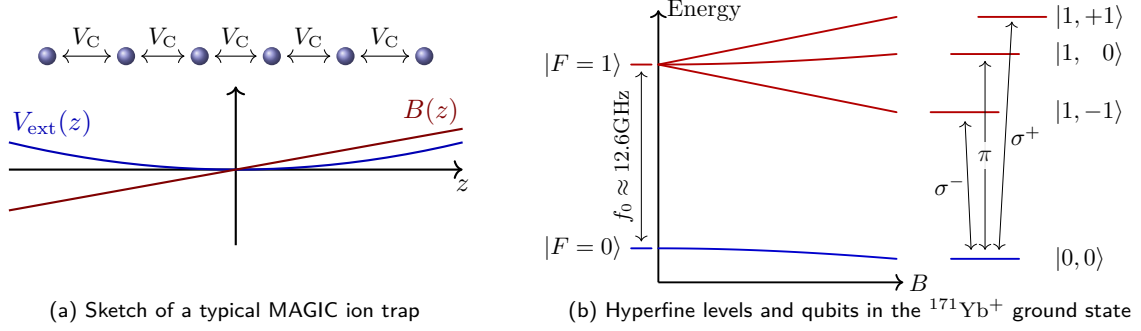


Figure 1: Schematics of a MAGIC ion trap. **Left:** six ions in linear configuration, with the plot below indicating the confining trap potential V_{ext} in axial direction and magnetic gradient field B . There is Coulomb repulsion between any two ions (only indicated for next neighbors). **Right:** the transition frequencies of the hyperfine sublevels of the $^{171}\text{Yb}^+$ ground state depend on the magnetic field strength (Breit-Rabi diagram [29], Zeeman effect exaggerated). The transitions on the right correspond to σ^- , π and σ^+ qubit, respectively.

1.3 Computational primitives

On the abstract quantum computing platform with N qubits specified by the requirements (I)–(III) above, interactions between the qubits are generated by the Ising-type Hamiltonian

$$H_0 := -\frac{1}{2} \mathbf{Z}^T J \mathbf{Z} = -\sum_{i < j}^N J_{ij} Z_i Z_j, \quad (1.1)$$

where the vector $\mathbf{Z} = (Z_1, \dots, Z_N)^T$ collects all local Pauli Z operators. The *coupling matrix* $J \in \mathbb{R}^{N \times N}$ is a symmetric matrix which encapsulates the physical properties of the platform. Since the diagonal entries of J merely generate a global and hence unobservable phase, we henceforth assume that J has vanishing diagonal. By requirement (III), we can assume that H_0 acts only on the $n \leq N$ relevant qubits, and we thus assume w.l.o.g. that J is a $n \times n$ matrix.

Letting the system evolve under the Hamiltonian (1.1) for some time t generates a unitary operation on the qubits. Our gate synthesis method is based on the observation that layers of local X gates can be used to emulate the evolution under a modified Hamiltonian: For any binary vector $\mathbf{s} \in \mathbb{F}_2^n$, set

$$\mathbf{X}^{\mathbf{s}} := \bigotimes_{i=1}^n \mathbf{X}^{s_i}, \quad (1.2)$$

and define the *qubit encoding* $\mathbf{m} = (-1)^{\mathbf{s}} \in \{-1, +1\}^n$ (to be understood entry-wise). We then have the following modified time evolution:

$$\mathbf{X}^{\mathbf{s}} \exp(-itH_0) \mathbf{X}^{\mathbf{s}} = \exp(-itH(\mathbf{m})). \quad (1.3)$$

Here, $H(\mathbf{m}) := -\frac{1}{2} \mathbf{Z}^T J(\mathbf{m}) \mathbf{Z}$ is the modified Hamiltonian with coupling matrix $J(\mathbf{m}) := J \circ \mathbf{m} \mathbf{m}^T$, and \circ denotes the Hadamard (entry-wise) product.

If we apply multiple time evolution operators with different encodings in succession, we can further simplify this scheme. For two encodings $\mathbf{m} = (-1)^{\mathbf{s}}$ and $\mathbf{m}' = (-1)^{\mathbf{s}'}$, we can combine the adjacent X layers in Eq. (1.3) and obtain

$$\mathbf{X}^{\mathbf{s}} e^{-itH_0} \mathbf{X}^{\mathbf{s}} \mathbf{X}^{\mathbf{s}'} e^{-it'H_0} \mathbf{X}^{\mathbf{s}'} = \mathbf{X}^{\mathbf{s}} e^{-itH_0} \mathbf{X}^{\mathbf{s} \oplus \mathbf{s}'} e^{-it'H_0} \mathbf{X}^{\mathbf{s}'}. \quad (1.4)$$

Hence, a change of encoding can be performed with a number of X gates equal to the number of sign flips needed to obtain \mathbf{m}' from \mathbf{m} . The total number of X layers needed to traverse a sequence of encodings is only one more than the length of the sequence.

1.4 Experimental motivation: Ion trap quantum computing with microwaves

Let us give a brief overview of a physical platform on which our computational primitives can be realized. For details, we refer the reader to Appendix A and Ref. [15].

The energy difference between hyperfine sublevels of some atomic state typically falls into the microwave regime of the electromagnetic spectrum, which makes pairs of such hyperfine states natural candidates for microwave-controlled qubits. For example, the “ground state” of ions with nuclear spin $I = \frac{1}{2}$ and total electron angular momentum $J = \frac{1}{2}$ (e.g. Ytterbium-171 ions, $^{171}\text{Yb}^+$) exhibits four hyperfine sublevels. They group into a singlet with $F = 0$ and a triplet with $F = 1$, where F is the quantum number specifying the magnitude of total angular momentum $\mathbf{F} = \mathbf{I} + \mathbf{J}$. The triplet states are energetically degenerate, but can be distinguished by their value of the magnetic quantum number $m_F \in \{-1, 0, +1\}$. The non-degenerate singlet state has $m_F = 0$. There is an energy gap between the multiplets, which for $^{171}\text{Yb}^+$ corresponds to a microwave frequency of about 12.6GHz, see also Figure 1b.

We use one ion to implement a single qubit and choose the singlet state as the computational zero state $|0\rangle := |F = 0, m_F = 0\rangle$. We then have the freedom to encode the computational one state $|1\rangle$ into any of the triplet states, and indicate this by the magnetic quantum number $m_F \in \{-1, 0, +1\}$ of the chosen $|1\rangle := |F = 1, m_F\rangle$.

In ion traps, magnetic fields can be used to lift the degeneracy of the triplet through the Zeeman effect, see Figure 1b. This separates the different qubit encodings in frequency space and enables single-qubit operations through microwave-driven two-level Rabi oscillations. Certain sequences of pulses on different transitions in the multilevel system can also be used to change the qubit encoding coherently (see Appendix A). However, this possibility only plays a minor role in our analysis, as we will explain below.

We now extend our scope to multiple ions in the same trap. They are stored in a linear configuration and form a “Coulomb crystal” due to their mutual repulsion. In the MAGIC scheme, the eponymous magnetic gradient along the crystal axis leads to different field strengths for the different ion positions, see Figure 1a. The consequence are different Zeeman splittings, which make the qubits distinguishable in frequency space. Thus, addressability is achieved, although the microwaves cannot be focused onto single ions. Additionally, the ions experience a “dipole force” in the inhomogeneous field, which couples internal and external degrees of freedom (s. Appendix A). This effect can be interpreted as an Ising-like interaction between the qubits, which we use in this work to generate multi-qubit gates.

To sum it up, the abstract requirements (I)–(III) are realized in microwave-driven ion traps exposed to inhomogeneous magnetic fields as follows: (I) single qubit rotations are realized by microwave-driven Rabi oscillations which can be executed in parallel through digitally generated microwave signals [30]. (II) the Ising-type interaction is the natural interaction in this setup. (III) selected ions can be taken out of the interaction by encoding them into the magnetic insensitive state with $m_F = 0$.

2 Synthesizing multi-qubit gates with Ising-type interactions

In this section, we investigate the set of gates which is generated by all possible time evolution operators of the Hamiltonians $H(\mathbf{m})$ defined in Eq. (1.1). Given time steps $\lambda_{\mathbf{m}} \geq 0$ during which the encoding \mathbf{m} is used, we thus consider unitaries of the form

$$\prod_{\mathbf{m}} e^{-i\lambda_{\mathbf{m}}H(\mathbf{m})} = e^{-i\sum_{\mathbf{m}} \lambda_{\mathbf{m}}H(\mathbf{m})}, \quad (2.1)$$

where we used that the diagonal Hamiltonians $H(\mathbf{m})$ mutually commute. For all possible encodings $\mathbf{m} \in \{-1, +1\}^n$ we collect the time steps $\lambda_{\mathbf{m}}$ in a vector $\boldsymbol{\lambda} \in \mathbb{R}^{2^n}$ and interpret $t = \sum_{\mathbf{m}} \lambda_{\mathbf{m}}$ as the total time of the unitary e^{-iH} .

We interpret the generated unitary as the time evolution operator under the *total Hamiltonian*

$$H := -\frac{1}{2}\mathbf{Z}^T \mathbf{A} \mathbf{Z}, \quad (2.2)$$

where we defined the *total coupling matrix*

$$\mathbf{A} := \sum_{\mathbf{m}} \lambda_{\mathbf{m}} \mathbf{J}(\mathbf{m}) = \mathbf{J} \circ \sum_{\mathbf{m}} \lambda_{\mathbf{m}} \mathbf{m} \mathbf{m}^T \quad (2.3)$$

and used the linearity of the Hadamard product.

Since the $\mathbf{m}\mathbf{m}^T$ are symmetric, A inherits the symmetry and the vanishing of the diagonal of J , see Section 1.3 and Appendix A. We wish to make our description of the coupling matrix independent of the platform dependent details given by J . Therefore, we define the Hadamard quotient M with entries

$$M_{ij} := \begin{cases} A_{ij}/J_{ij}, & i \neq j, \\ 0, & i = j. \end{cases} \quad (2.4)$$

The implicit assumption that J has no vanishing non-diagonal entries is commonly met by experiments. Our objective is to minimize the total gate time and the amount of \mathbf{m} 's needed to express the matrix M . To this end we formulate the following linear program (LP):

$$\begin{aligned} & \text{minimize} && \mathbf{1}^T \boldsymbol{\lambda} \\ & \text{subject to} && M = \sum_{\mathbf{m}} \lambda_{\mathbf{m}} \mathbf{m}\mathbf{m}^T, \\ & && \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^{2^n-1}, \\ & && \mathbf{m} \in \{-1, +1\}^n. \end{aligned} \quad (2.5)$$

As above, $\mathbf{1} = (1, 1, \dots, 1)$ is the all-ones vector such that $\mathbf{1}^T \boldsymbol{\lambda} = \sum_{\mathbf{m}} \lambda_{\mathbf{m}}$. Moreover, we use the symmetry $(-\mathbf{m})(-\mathbf{m})^T = \mathbf{m}\mathbf{m}^T$ to reduce the degree of freedom in $\boldsymbol{\lambda}$ from 2^n to 2^n-1 .

This LP has the form of a ℓ_1 -norm minimization over the non-negative vector $\boldsymbol{\lambda}$. As such, it is a convex relaxation of minimizing the number of non-zero entries of $\boldsymbol{\lambda}$, sometimes called the ℓ_0 -“norm”. Heuristically, it is thus expected that the LP (2.5) favors sparse solutions. As we shall see shortly in Theorem 2.2, the LP (2.5) always has a *feasible* solution (i.e. there are variables $\boldsymbol{\lambda}$ such that all constraints are satisfied) for any symmetric matrix M with vanishing diagonal. The theory of linear programming then guarantees the existence of an optimal solution with at most $n(n-1)/2$ non-zero entries, see Proposition 2.3.

For any symmetric $n \times n$ matrix A with vanishing diagonal, we define an associated multi-qubit gate $\text{GZZ}(A)$, where GZZ stands for “global ZZ interactions”,

$$\text{GZZ}(A) := e^{i\frac{1}{2}\mathbf{Z}^T A \mathbf{Z}}. \quad (2.6)$$

Here, the decomposition of A is found using the LP (2.5) and involves at most $n(n-1)/2$ different encodings \mathbf{m} . Recall from Section 1.3, that these encodings can be emulated with suitable X layers and hence $\text{GZZ}(A)$ can be implemented using at most $n(n-1)/2 + 1$ such layers. We call the exact number of X layers the *encoding cost* of $\text{GZZ}(A)$, and $\mathbf{1}^T \boldsymbol{\lambda}$ the *total GZZ time*. For this derivation, we have intentionally been agnostic of the physical details of the ion trap but note that the values of $\lambda_{\mathbf{m}}$ and therefore t depend on the (physical) coupling matrix J .

Finally, given an optimal decomposition of A , it is also possible to minimize the total number of X gates needed for the implementation. Since every X gate introduces noise, such a minimization improves the fidelity of GZZ gates in practice. By Eq. (1.4), the number of X gates needed to change the encoding from \mathbf{m} to \mathbf{m}' is exactly the number of signs in \mathbf{m} that have to be flipped to obtain \mathbf{m}' . Since the resulting gate e^{-iH} does not depend on the order of encodings \mathbf{m} , one can minimize the total number of sign flips over all possible orderings. However, finding an optimal ordering generally corresponds to solve a traveling salesman problem on the support of $\boldsymbol{\lambda}$ and is thus NP-hard [31]. Nevertheless, there are good heuristic algorithms, e.g. Christofides’s algorithm introduced in Ref. [32].

Before demonstrating how to use the flexibility and the all-to-all connectivity of GZZ gates for compiling, we discuss some theoretical aspects as well as limitations and extensions of the above approach. We conclude by presenting numerical results for the synthesis of $\text{GZZ}(A)$ gates for randomly chosen matrices A .

2.1 Theoretical aspects

First, we show the existence of a solution for the LP (2.5) via frame theoretic arguments, then we investigate the sparsity of optimal solutions from a geometrical viewpoint. Let us define the

$n(n-1)/2$ -dimensional subspace of symmetric matrices with vanishing diagonal by

$$\text{Sym}_0(\mathbb{R}^n) := \{M \in \text{Sym}(\mathbb{R}^n) \mid M_{ii} = 0 \ \forall i \in [n]\}. \quad (2.7)$$

Moreover, we denote the set of outer products generated by all possible encodings by

$$\mathcal{V} := \{\mathbf{m}\mathbf{m}^T \mid \mathbf{m} \in \{-1, +1\}^n, m_n = +1\}. \quad (2.8)$$

Due to the symmetry $\mathbf{m}\mathbf{m}^T = (-\mathbf{m})(-\mathbf{m})^T$ we can uniformly fix the value of one of the entries of \mathbf{m} . We chose the convention $m_n = +1$.

Definition 2.1. *Let V be a (finite-dimensional) Hilbert space. A set of vectors $v_1, \dots, v_N \in V$ is called a frame if their linear span is V . A frame is said to be tight if there exists a $a > 0$ such that for all $v \in V$*

$$a\|v\|^2 = \sum_{i=1}^N |\langle v, v_i \rangle|^2. \quad (2.9)$$

Moreover, a frame is said to be balanced if $\sum_{i=1}^N v_i = 0$.

With this definition, we obtain the following:

Theorem 2.2. *The set \mathcal{V} is a balanced tight frame for $\text{Sym}_0(\mathbb{R}^n)$. In particular, the LP (2.5) has a feasible solution for any $M \in \text{Sym}_0(\mathbb{R}^n)$.*

The proof that \mathcal{V} is a balanced tight frame can be found in Appendix B, along with other properties of \mathcal{V} . Since \mathcal{V} is a frame for $\text{Sym}_0(\mathbb{R}^n)$, any matrix $M \in \text{Sym}_0(\mathbb{R}^n)$ can be decomposed as $M = \sum_{\mathbf{m}} \lambda_{\mathbf{m}} \mathbf{m}\mathbf{m}^T$. In other words, the LP (2.5) has a feasible solution for any $M \in \text{Sym}_0(\mathbb{R}^n)$. Then, a standard linear programming argument shows that there is always an optimal solution with sparsity at most $n(n-1)/2$. Such a solution can be numerically found by using variants of the simplex algorithm (see e.g. Ref. [33] for more details). We formulate this fact as the following proposition and defer the proof to Appendix C, where we also show some geometric properties of optimal solutions of a more general class of LP's.

Proposition 2.3 (Sparsity of optimal solutions). *There exists an optimal solution to the LP (2.5) with sparsity $\leq n(n-1)/2$ for every $M \in \text{Sym}_0(\mathbb{R}^n)$. The simplex algorithm is guaranteed to return such an optimal solution.*

2.2 Practical limitations

In the previous section we showed how to implement a GZZ gate through the Ising-evolution time under at most $n(n-1)/2$ different encodings. However, in an actual ion trap, practical limitations might occur for very short evolution times. For this work, we neglect the potential error introduced by a finite recoding time, i.e. the time needed to perform X gates. During recoding we have additional Hamiltonian terms corresponding to the X gates simultaneously with the “always-on” Ising Hamiltonian (1.1). These cause unwanted effects and the introduced errors become non-negligible when the Ising-evolution time approaches the recoding time, see also Appendix A.3. Below we observe in our numerical results for the LP that some $\lambda_{\mathbf{m}}$ are below that recoding time. In this section, we address these problems and propose extensions to our approach to mitigate them. First, we discuss the amount of error we would make in an appropriate norm by ignoring too small $\lambda_{\mathbf{m}}$, i.e. by defining a threshold ε and setting $\lambda_{\mathbf{m}} \equiv 0$ if $\lambda_{\mathbf{m}} < \varepsilon$. To avoid small evolution times in the first place, we then define a mixed integer program (MIP) which can solve the problem exactly with a lower (and upper) bound on the $\lambda_{\mathbf{m}}$.

2.2.1 Truncation error

Suppose the target Hamiltonian H is decomposed as in Eq. (2.2). Given a threshold $\varepsilon > 0$ for the $\lambda_{\mathbf{m}}$, we define $C := \{\mathbf{m} \mid \lambda_{\mathbf{m}} \leq \varepsilon\}$ and approximate H by the Hamiltonian

$$H' = - \sum_{\mathbf{m} \notin C} \lambda_{\mathbf{m}} \mathbf{Z}^T (J \circ \mathbf{m}\mathbf{m}^T) \mathbf{Z}. \quad (2.10)$$

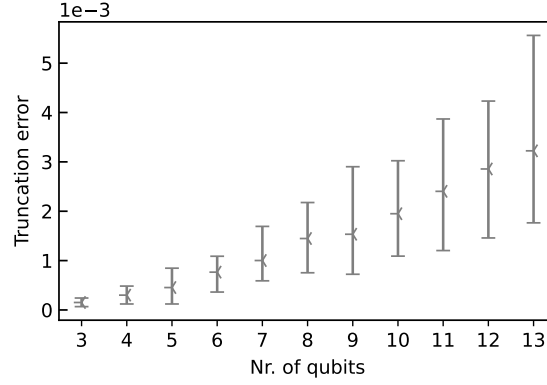


Figure 2: The error scaling, from Eq. (2.11), introduced by truncating all $\lambda_m < \varepsilon_l = 27\mu\text{s}$ from the solution of the LP is displayed. The error bars show the minimum/maximum deviation from the mean over 20 randomly sampled binary A matrices.

The diamond norm error made by replacing the time evolution operator of H with the one of H' , is upper bounded by half their spectral norm distance (see e.g. Ref. [34]):

$$\frac{1}{2} \left\| e^{-iH} - e^{-iH'} \right\|_{\infty} = \max_{x \in \mathbb{R}_2^n} \left| \sin\left(\frac{1}{2}(H_{x,x} - H'_{x,x})\right) \right|. \quad (2.11)$$

Here, we used that H and H' are diagonal in the computational basis with diagonal entries $H_{x,x} := \langle x | H | x \rangle$ and, moreover, that $|1 - e^{i\varphi}| = 2 |\sin(\varphi/2)|$. The difference of the diagonal entries is

$$H_{x,x} - H'_{x,x} = - \sum_{i \neq j} J_{ij} \sum_{\mathbf{m} \in C} \lambda_{\mathbf{m}} m_i m_j (-1)^{x_i + x_j}. \quad (2.12)$$

Since $|\sin(\theta)| \leq |\theta|$ for all $\theta \in \mathbb{R}$, we find that

$$\frac{1}{2} \left\| e^{-iH} - e^{-iH'} \right\|_{\infty} \leq \frac{1}{4} \sum_{i \neq j} |J_{ij}| \sum_{\mathbf{m} \in C} \lambda_{\mathbf{m}}. \quad (2.13)$$

Hence, the truncation error scales with the total truncated time and the interaction strength. Both depend on the number of participating qubits n , however not in a straightforward way. The scaling of the truncation error under realistic assumptions is showed in Figure 2.

2.2.2 The mixed integer program approach

To avoid small $\lambda_{\mathbf{m}}$ without introducing an additional error as above, we propose to add a lower bound on their values to the LP (2.5). To this end, we introduce additional binary variables \mathbf{b} , as in Ref. [35], which renders the optimization into the following mixed integer program (MIP):

$$\begin{aligned} & \text{minimize} && \alpha \mathbf{1}^T \boldsymbol{\lambda} + (1 - \alpha) \mathbf{1}^T \mathbf{b} \\ & \text{subject to} && M = \sum_{\mathbf{m}} \lambda_{\mathbf{m}} \mathbf{m} \mathbf{m}^T, \\ & && \varepsilon_l \mathbf{b} \leq \boldsymbol{\lambda} \leq \varepsilon_u \mathbf{b}, \\ & && \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^{2^{n-1}}, \\ & && \mathbf{m} \in \{-1, +1\}^n, \\ & && \mathbf{b} \in \mathbb{F}_2^{2^{n-1}}. \end{aligned} \quad (2.14)$$

Here $0 \leq \varepsilon_l < \varepsilon_u$ are bounds on the entries of $\boldsymbol{\lambda}$: The lower bound ε_l can be set to the minimal Ising-evolution time which can be realized in practice. The upper bound ε_u can be chosen freely, but small values of ε_u are favorable since the runtime of the MIP is generally shorter for smaller intervals $[\varepsilon_l, \varepsilon_u]$ as explained in Ref. [35]. Since $\max(\boldsymbol{\lambda})$ depends on $\max_{i < j} (M_{ij})$ one can select $\varepsilon_u \propto \max_{i < j} (M_{ij})$. The parameter $\alpha \in [0, 1]$ is used to assign weights to the optimization of $\mathbf{1}^T \mathbf{b}$ (sparsity) and $\mathbf{1}^T \boldsymbol{\lambda}$ (total GZZ time), respectively.

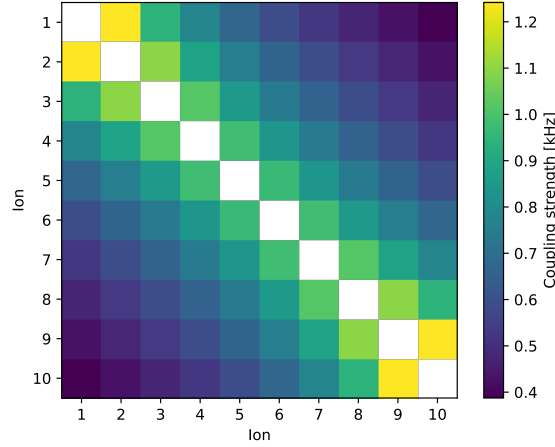


Figure 3: An example coupling matrix J for an ion trap with 10 ions and harmonic trap potential. J is determined by the physical parameters of the trap, see Section 2.3 for the concrete values, and can be computed by Eq. (A.12).

2.3 Numerical results

We investigate how well the proposed methods for the synthesis of $\text{GZZ}(A)$ gates performs in practice. To this end, we solve the LP (2.5) and the MIP (2.14) numerically for randomly chosen matrices A and compare the solutions to a naive approach. As explicated above, our goal is to minimize the total GZZ time $\mathbf{1}^T \lambda$ subject to $A/J = M = \sum_{\mathbf{m}} \lambda_{\mathbf{m}} \mathbf{m} \mathbf{m}^T$. The numerical results in this section show the performance of GZZ gates on all qubits, i.e. $n = N$.

To demonstrate the performance of our approach in a realistic setting we consider the ion trap architecture of Appendix A with a harmonic trap potential. Concretely, we take Ytterbium $^{171}\text{Yb}^+$ ions with Rabi frequency $\Omega = 2\pi 100\text{kHz}$, magnetic field gradient $B_1 = 100\text{T/m}$ and axial trap frequency $\omega_z = 2\pi 100\text{kHz}$. This determines the coupling matrix J via Eq. (A.12), see Ref. [36] for more details. An example coupling matrix J for 10 ions is shown in Figure 3. We made the Python code for its computation available on GitHub [37].

On a logical level, we consider a symmetric binary matrix $A \in \text{Sym}_0(\mathbb{F}_2^n)$ defined in Eq. (2.3), which indicates where the ZZ gates are located: For all $i < j$, $A_{ij} = 1$ if there is a ZZ gate between qubits i and j . We simulate a random ZZ gate layer by sampling the entries of the lower/upper triangular part of A uniformly from $\{0, 1\}$.

We compare the results of the LP and the MIP with a “naive approach”, which corresponds to a sequential execution of the ZZ gates. As before, we neglect the gate time for single-qubit gates. Moreover, “total gate time” refers to the total GZZ time, i.e. $\mathbf{1}^T \lambda$, for the LP and the MIP. For the naive approach, the “total gate time” is the time needed to execute the sequence of ZZ gates, i.e. $\sum_{i < j} A_{ij}/J_{ij}$, and the encoding cost is the number of ZZ gates $\sum_{i < j} A_{ij}$.

Figure 4 shows the numerical results for solving the LP (2.5) and the MIP (2.14). The encoding cost for the naive approach scales roughly with $n^2/4$ since on average half of the randomly chosen $n(n-1)/2$ entries of the lower triangular part of A do not vanish. In Figure 4 we find that the total GZZ time scales linearly with the number of participating qubits. In contrast, by Proposition 2.3 the number of encodings needed to implement a GZZ gate increases quadratically with the number of participating qubits. Thus, the time between the encodings becomes shorter and shorter, which results in arbitrarily small evolution times $\lambda_{\mathbf{m}}$ for many qubits, see also Section 2.2. Hence, the more qubits we consider, the more solutions $\lambda_{\mathbf{m}}$ of the LP are smaller than the lower bound ε_l , see Figure 2. This explains the deviation of the encoding cost of the MIP from that of the LP.

Figure 4 also shows that the MIP can be solved in practice and, similar to the LP, yields nearly time-optimal GZZ gates. Thus, even when taking practical limitations into account, GZZ gates can be implemented in linear time and an encoding cost of approximately $n(n-1)/2$.

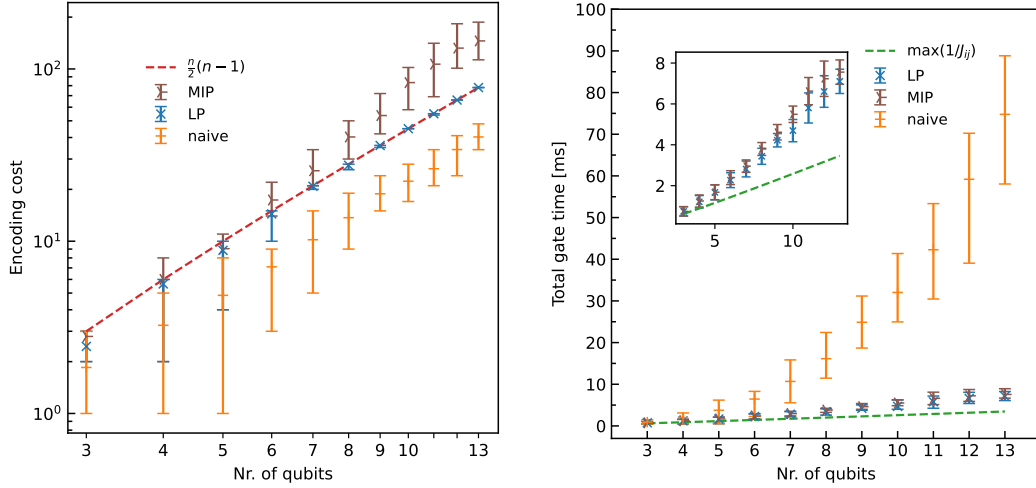


Figure 4: Comparing the performances of the LP and the MIP to the naive approach for the implementation of a random ZZ layer. The error bars show the minimum/maximum deviation from the mean over 20 randomly sampled binary A matrices. **Left:** The encoding cost to implement a random ZZ layer on n qubits. The naive approach only costs half as much as the LP, see Section 2.3. For many qubits, the LP approaches the upper bound (red dashed line) of Proposition 2.3, whereas the MIP exceeds this upper bound. **Right:** The total gate time of the naive approach scales quadratically with the number of qubits due to the quadratic scaling of the number of possible ZZ gates. The total gate time of both the LP and the MIP scales approximately linear. The inset shows a zoomed in version of the plot. One can see that the LP and the MIP roughly take twice as long as the slowest two-qubit ZZ gate (green dashed line).

Details of computer implementation. We use the Python package CVXPY [38, 39] with the GNU linear program kit simplex solver [40] to solve the LP, and the MOSEK solver [41] for the MIP. We change one parameter of MOSEK to improve the runtime at the expense of not finding the optimal solution. Concretely, we set the MOSEK parameter `MSK_DPAR_MIO_TOL_REL_GAP` to 0.6.

For the MIP we choose the lower bound $27\mu\text{s} = \varepsilon_l \leq \lambda_m$ for all m , which is motivated by the concrete ion trap setup: the duration of a robust X gate is about five times longer than the duration of a X gate (i.e. a π -pulse), which is roughly $\pi/\Omega = 5\mu\text{s}$ [42]. We further pick the upper bound $\varepsilon_u = 3/2 \max_{i < j} |M_{ij}|$. Note that if the interval $[\varepsilon_l, \varepsilon_u]$ is too narrow, we might not be able to find any feasible solution. If it is too wide, the runtime of the solver might increase [35]. We observed in our numerical studies (not shown) that both the total gate time and the encoding cost are essentially constant in α as long as α is not too close to the extremal values 0 and 1. We therefore, deliberately put equal weights on the two terms in the objective function and set $\alpha = 0.5$.

In practice, the simplex algorithm has a runtime which is polynomial in the problem size [43]. Since the LP (2.5) has 2^{n-1} variables, the runtime is exponential in n . For moderate n , this is however still manageable on modern hardware – a Laptop with Intel Core i7 Processor (8x 1.8 GHz) and 16 GB RAM needs on average only 20 seconds to solve the LP for $n = 13$. This is about the size of most ion trap quantum computers nowadays. The runtime of mixed integer programs is exponential in the worst case. Nevertheless, solving MIP (2.14) for $n = 13$ qubits using the MOSEK solver took on average about seven minutes on the same hardware.

An implementation of the LP and MIP used to generate Figure 4 is provided on GitHub [37].

3 Compilation with GZZ gates

The GZZ gates defined in Eq. (2.6) can be used in compilation schemes to improve multi-qubit gate counts over previous results [7, 8, 26]. Our main result is that we can implement any Clifford circuit on n qubits using only $n + 1$ GZZ gates, n two-qubit gates and single-qubit gates.

First, we introduce the notation used in later sections and derive some gate equivalences that will help us later. Then, we study the practically relevant case of compiling global Clifford unitaries. These ubiquitous unitaries play an important role as basic building blocks of quantum circuits, especially in fault-tolerant quantum computing, and are of major importance for cryptographic and tomographic protocols due to their statistical properties. For the compilation of Clifford unitaries, we make use of their Bruhat decomposition, also used in Ref. [44], and compile the entangling layers into GZZ gates. Interestingly, this compilation scheme for entangling Clifford layers can be generalized beyond pure Clifford circuits. As an example, we show how it can be used to compile an n -qubit QFT. Finally, we propose a compiling scheme for general diagonal unitaries which may be used as a step towards decomposing general unitaries.

3.1 Notation

We introduce some notation and link our GZZ gate from Eq. (2.6) to other known entangling gates. Subscripts on gates indicate on which qubits they act. For $\mathbf{x} \in \mathbb{F}_2^n$ and $\alpha \in [0, 2\pi)$ we thus have

$$\begin{aligned}
\text{Z rotation:} \quad & R_Z(\alpha)_j |\mathbf{x}\rangle = e^{i\alpha x_j} |\mathbf{x}\rangle, \\
\text{ZZ gate:} \quad & ZZ(\alpha)_{i,j} |\mathbf{x}\rangle = e^{i\alpha(x_i \oplus x_j)} |\mathbf{x}\rangle, \\
\text{Controlled Z rotation:} \quad & CR_Z(\alpha)_{i,j} |\mathbf{x}\rangle = e^{i\alpha x_i x_j} |\mathbf{x}\rangle, \\
\text{Controlled X gate:} \quad & CX_{i,j} |\mathbf{x}\rangle = |x_1, \dots, x_{j-1}, x_j \oplus x_i, x_{j+1}, \dots, x_n\rangle, \\
\text{Hadamard gate:} \quad & H_j |\mathbf{x}\rangle = \frac{1}{\sqrt{2}} |x_1, \dots, x_{j-1}\rangle (|0\rangle + (-1)^{x_j} |1\rangle) |x_{j+1}, \dots, x_n\rangle,
\end{aligned} \tag{3.1}$$

where \oplus denotes addition modulo 2. In particular, we denote the phase gate and the controlled-Z gate as

$$S_j := R_Z(\pi/2)_j \quad \text{and} \quad CZ_{i,j} := CR_Z(\pi)_{i,j}, \tag{3.2}$$

respectively. Note that in the definition of the ZZ gate α can be identified with entries of the matrix A in Eq. (2.3). In terms of integer arithmetic, we have $2xy = x + y - (x \oplus y)$ for $x, y \in \{0, 1\}$ which yields

$$CR_Z(\alpha)_{i,j} \equiv R_Z(\alpha/2)_i R_Z(\alpha/2)_j ZZ(-\alpha/2)_{i,j}. \tag{3.3}$$

Since the GZZ gate from Eq. (2.6) consists only of ZZ gates, we can express it as

$$GZZ(A) = e^{ia} \prod_{i < j} ZZ(-2A_{ij})_{i,j}, \tag{3.4}$$

where $A \in \text{Sym}_0(\mathbb{R}^n)$ is a symmetric matrix with vanishing diagonal and $a := \sum_{i < j} A_{ij}$. Similarly, a layer of arbitrary controlled R_Z rotations is characterized by $A \in \text{Sym}_0(\mathbb{R}^n)$ via

$$GCR_Z(A) := \prod_{i < j} CR_Z(A_{ij})_{i,j} = e^{-\frac{i}{4}a} GZZ(A/4) \prod_{i=1}^n R_Z(b_i/2)_i, \tag{3.5}$$

where we used Eq. (3.3) and abbreviated $b_i := \sum_j A_{ij}$. A general CX layer is given by

$$GCX(B) |\mathbf{x}\rangle := |B\mathbf{x}\rangle, \tag{3.6}$$

with a matrix $B \in \text{GL}_n(\mathbb{F}_2)$. We call $GCX(B)$ a *directed CX layer* if B is lower or upper triangular.

3.2 Clifford circuits

The *Clifford group* Cl_n is a finite subgroup of the unitary group $\text{U}(2^n)$ that is generated by the single-qubit Hadamard gate H_j and phase gate S_j , as well as the two-qubit $CX_{i,j}$ gate. Conversely, it is a natural task to decompose an arbitrary Clifford unitary $U \in \text{Cl}_n$ into these generators. This task is solved by a number of algorithms, see e.g. Refs. [45–47], with the same asymptotic gate count, but a differently structured output circuit.

Here, we make use of the so-called “Bruhat decomposition” [44, 46]. This decomposition has the advantage that the entangling gates are grouped either in CZ gate layers or directed CX layers, which both can be directly compiled into GZZ gates. More precisely, the algorithm in Ref. [44] writes any Clifford unitary in the form $-X-Z-CX-CZ-S-H-CX-CZ-S-$, where

- $-X-Z-$ is a layer of Pauli gates,
- $-CX-$ are layers of directed CX gates,
- $-CZ-$ are layers of controlled Z gates,
- $-S-$ are layers of phase gates S and
- $-H-$ is a layer consisting of Hadamard gates H (and permutation operations).

In the following, we concentrate on the decomposition of the CZ and CX layers as the remaining layers consist of local gates with straightforward implementation. As we show, the compilation of a directed CX layer is a lot more expensive than the compilation of a CZ layer.

As a corollary of CZ layer compilation, we show how to efficiently prepare multi-qubit stabilizer states with GZZ gates, i.e. states of the form $U|0\rangle$ where $U \in \text{Cl}_n$ is a Clifford unitary. Stabilizer states are important e.g. for the construction of mutually unbiased bases, or more generally, informationally complete positive operator valued measures (POVMs), and their preparation is thus of practical relevance for tomographic protocols, see e.g. Refs. [48, 49].

First we transform the directed CX layer by conjugating the CX gate targets with Hadamard gates. Then we use the structure of the resulting gate layer to reduce the encoding cost with the GZZ gate. Furthermore, we use the same method to reduce the encoding cost of the QFT. We underpin the advantages of our method with numerical simulations.

3.2.1 Implementing CZ layers

Since CZ gates commute, we can rewrite any CZ circuit, using Eqs. (3.2) and (3.5), as

$$\begin{aligned} \text{GCZ}(A) &:= \prod_{i < j} \text{CZ}_{i,j}^{A_{ij}} = \prod_{i < j} \text{CR}_Z(\pi A_{ij})_{i,j} \\ &= \text{GCR}_Z(\pi A) \\ &= e^{-\frac{i\pi}{4}a} \text{GZZ}\left(\frac{\pi}{4}A\right) \prod_{i=1}^n \text{R}_Z\left(\frac{\pi}{2}b_i\right)_i \\ &= e^{-\frac{i\pi}{4}a} \text{GZZ}\left(\frac{\pi}{4}A\right) \prod_{i=1}^n S_i^{b_i}, \end{aligned} \tag{3.7}$$

where $A \in \text{Sym}_0(\mathbb{F}^n)$ is again a symmetric matrix with zero diagonal and a and b_i are defined as in Section 3.1. Note that the phase gate S has order 4, so effectively only $b_i \bmod 4$ plays a role and the single-qubit gates are from the set $\{I, S, Z, S^\dagger\}$.

3.2.2 Stabilizer state preparation

Although any stabilizer state can be written as $U|0\rangle$ for some Clifford unitary $U \in \text{Cl}_n$, not the full Clifford group is needed to generate all stabilizer states. In fact, it is well known that any stabilizer state can be obtained by acting with local Clifford gates on *graph states* [50, 51]. Graph states are defined as

$$|A\rangle := \prod_{i < j} \text{CZ}_{i,j}^{A_{ij}} |+\rangle^n, \tag{3.8}$$

where $A \in \text{Sym}_0(\mathbb{F}_2^n)$ and $|+\rangle^n = H^{\otimes n}|0\rangle$. Hence, by the above, any stabilizer state can be prepared by an initial global Hadamard layer, a GZZ gate, and a final layer of single-qubit Clifford gates.

3.2.3 Decomposing directed CX layers

Let us now consider the two directed CX layers in the Bruhat decomposition [44]. With the notation introduced in Eq. (3.6), a *fan-out* gate with the following action on a state $\mathbf{x} \in \mathbb{F}_2^n$,

$$\text{GCX}(B_{FO})|\mathbf{x}\rangle = |x_1, x_2 \oplus x_1, \dots, x_n \oplus x_1\rangle, \quad (3.9)$$

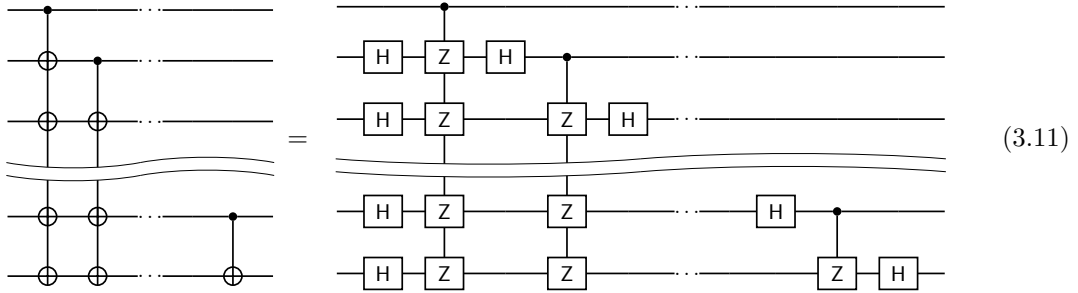
has the binary matrix

$$B_{FO} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & & & \\ \vdots & & \mathbb{1}_{n-1} & \\ 1 & & & \end{pmatrix}. \quad (3.10)$$

From the structure of the matrix B_{FO} it can be seen that any directed CX layer can be realized by at most $n - 1$ fan-out gates. Each fan-out gate is equivalent to a GCR_Z gate, conjugated with Hadamard gates on the target qubits. Since by Eq. (3.5) each GCR_Z can be realized with one GZZ gate, we need at most $n - 1$ GZZ gates to realize a directed CX layer, see Eq. (3.11) for an example. Concretely, the total encoding cost for a directed CX layer, implementing each of the $n - 1$ fan-out gates with one GZZ gate scales as $O(1/6n^3)$. Below Eq. (3.11) we show that $\lfloor \frac{n-1}{2} \rfloor$ GZZ gates are enough to implement a directed CX layer. Therefore, one requires $n - 1$ (if n is odd) or $n - 2$ (if n is even) GZZ gates, for the two directed CX layers appearing in the Bruhat decomposition of Ref. [44]. Since we can realize the CZ layer with exactly one GZZ gate, each Clifford circuit requires only $n + 1$ or n GZZ gates and only $n - 1$ or n two-qubit CZ gates for n odd or n even, respectively.

Fully directed CX layer. We call a directed CX layer *fully directed* if the corresponding gate $\text{GCX}(B)$ is characterized by a $n \times n$ matrix B with zeros in the upper triangular matrix and ones everywhere else. Fully directed CX layers are related to the textbook QFT, see Section 3.3 below.

Remember that we can represent any directed CX layer as a concatenation of $n - 1$ fan-out gates. Commuting Hadamard gates through each target of the fan-out gate transforms the controlled X to a controlled Z gate by $\text{HXH} = \text{Z}$. This converts a fan-out gate into a CZ-type fan-out gate, which we also call fan-out gate for short. We can thus transform a fully directed CX layer by applying $\mathbb{1} = H^2$ from the left and commuting one of the Hadamard gates to the right until it hits a control:



Note that such a transformation obviously works also for an arbitrary directed CX layer. The locations of the resulting H and CZ gates can be represented as tables

$$T_H = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ 1 & 0 & \dots & 0 & 1 \end{bmatrix} \quad \text{and} \quad T_{CZ} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}, \quad (3.12)$$

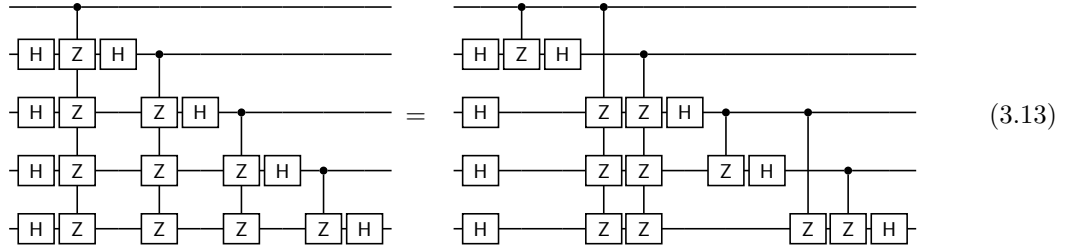
where T_{CZ} has the same form as the matrix B for the fully directed CX layer. The row index of these tables indicates the qubit on which the corresponding gate acts, while the column index

indicates the “time step”, i.e. the horizontal position in the circuit diagram. Thus, $T_{CZ}[i, j] = 1$ if qubit i is either a control or a target of a CZ at time step j , and $T_{CZ}[i, j] = 0$ if qubit i is idle at time step j . Similarly, $T_H[i, j] = 1$ if a Hadamard gate acts on qubit i at time step j , and $T_H[i, j] = 0$ if qubit i remains unchanged. For a fully directed CX layer, T_H and T_{CZ} always have this form. To locate the components of the circuit on the right-hand side of Eq. (3.11) one starts with reading the first column of T_H , then the first column of T_{CZ} and so on.

We now aim at reducing the encoding cost of the fully directed CX layer on n qubits. To this end, we reduce the supports m of the GZZ gates implementing the fan-out gates in the CX layer, since the encoding cost of a GZZ gate scales as $m(m-1)/2$, see Section 2. Consider an odd column i in T_{CZ} which corresponds to a CZ-type fan-out gate with control on qubit i . We split column i into two columns as follows: The first one representing a two-qubit CZ gate on qubits i and $i+1$, and the second column is the same as the original one except that it does not target qubit $i+1$. This splitting increases the number of columns in T_{CZ} by one. For example, for $n = 5$ and $i = 1$ we split the column $[1, 1, 1, 1, 1]^T$ into $[1, 1, 0, 0, 0]^T$ and $[1, 0, 1, 1, 1]^T$, where we keep in mind that the first nonzero entry in a column denotes the control qubit and hence has to appear in both parts. Furthermore, we update the Hadamard table T_H by inserting a zero column after column $i+1$ to account for the new column in T_{CZ} . The fan-out gate resulting from the split of the odd column i of T_{CZ} together with the even column $i+1$ can be implemented with one GZZ on i qubits.

Note that we can not move parts of columns of T_{CZ} to the left since there is always a Hadamard gate on the left of the control qubit of that column that blocks it. Therefore, we only split odd columns i and move them to the right.

This splitting of columns of T_{CZ} corresponds to moving the Hadamard gate on the $i+1$ qubit to the left or, equivalently, moving all CZ gates except the one acting on the $i+1$ qubit to the right, as exemplified in the following for $n = 5$ qubits:



For general n , the scheme works exactly the same as in this example.

The circuit before the splitting on the left-hand side of Eq. (3.13) can be implemented with $n-1$ GZZ gates each acting on $n, \dots, 2$ qubits respectively, where n is the number of qubits of the directed CX layer, and thus have an encoding cost of $\sum_{i=2}^n i(i-1)/2 = n^3/6 + O(n^2)$. On the right-hand side we need $\lceil \frac{n-1}{2} \rceil$ CZ gates and $\lfloor \frac{n-1}{2} \rfloor$ GZZ gates, resulting in an encoding cost of

$$\left\lceil \frac{n-1}{2} \right\rceil + \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \frac{n-2i}{2} (n-2i-1) = \frac{1}{12} n^3 + O(n^2). \quad (3.14)$$

The first term comes from the encoding cost of the CZ gates, which is one per CZ gate. The second term comes from combining a GZZ gate on n qubits with a GZZ gate on $n-1$ qubits resulting in a GZZ gate on n qubits. The encoding cost in Eq. (3.14) has the same cubic scaling with n as before combining GZZ gates, but we were able to improve the coefficient from $1/6$ to $1/12$. Recall from Section 2.3 that the encoding cost of the naive approach scales only quadratic, so we trade higher encoding cost for faster gates. Each CZ gate can be implemented as described in Eq. (3.3) for $\alpha = \pi$ by a single ZZ gate and two additional single-qubit $R_Z(\pi/2) \equiv \sqrt{Z} = S$ gates on the control and target qubit, respectively. Since the S gates do commute with ZZ and GZZ gates but do not commute with Hadamard gates, we can combine most of the S gates to an S^k gate, where $k \in \{0, 1, 2, 3\}$. There are two Hadamard gates on $n-1$ qubits and none on the first qubit, therefore we have $2(n-1) + 1 S^k$ gates. To summarize, in addition to $\lceil \frac{n-1}{2} \rceil$ CZ gates and $\lfloor \frac{n-1}{2} \rfloor$ GZZ gates we need $2n-1 S^k$ gates to implement a fully directed CX layer on n qubits.

Our method optimizes both the encoding cost and the total gate time of the directed CX layer. We expect to lose time-optimality for a fully directed CX layer if we split any GZZ gate into

smaller pieces and therefore reduce the encoding cost. In the extreme case, we would split all GZZ gates into two-qubit ZZ gates and thereby would end up with the naive approach explained in Section 2.3. Our method combines parts of the i -th column of T_{CZ} with the $i + 1$ -th column to a GZZ gate on $n - i + 1$ qubits. One could think that moving parts of the larger GZZ gate farther to the right might improve the encoding cost. But the support of the part left behind, and therefore the encoding cost increases the farther we push the other part to the right. In the extreme case of pushing all parts as far as possible to the right we end up with a “transposed” table where the lower triangular part is zero, and we did not achieve any reduction of the encoding cost.

Algorithm 1 Moving Hadamard gates.

Input: T_{CZ}

Initialize T_H as in Eq. (3.12)

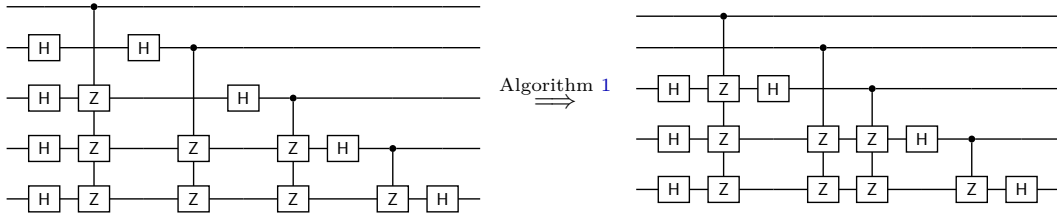
 $h_{\max} \leftarrow 0$ ▷ Position of the rightmost H that has already been moved left
for $i = 1, \dots, n - 1$ **do**
 $T_H[i, i] \leftarrow 0$ ▷ H on i -th qubit leaves its position
 $c \leftarrow \max\{j = 0, \dots, i - 1 | T_{CZ}[i, j] = 1\} + 1$ ▷ Find position directly after first CZ to the left
if $c = \text{NaN}$ **then** ▷ No CZ found (max was taken on an empty set)
 $T_H[i, 0] \leftarrow 0$ ▷ Cancel H in the first layer
else if $c = i$ **then** ▷ Unable to move left, attempt to move right
if $\{j = i, \dots, n - 1 | T_{CZ}[i, j] = 1\} = \emptyset$ **then** ▷ If no CZ is to the right...
 $T_H[i, n - 1] \leftarrow 1$ ▷ ...move H to the last layer, ...
else
 $T_H[i, i] \leftarrow 1$ ▷ ...otherwise remain in place.
else
 $h_{\max} \leftarrow \max\{h_{\max}, c\}$ ▷ Find the more restrictive condition (either CZ on current qubit or H on previous)
 $T_H[i, h_{\max}] \leftarrow 1$ ▷ Move H to the target layer
Output: T_H

Arbitrary directed CX layer. Until now, we considered only fully directed CX layers which in practice is a very special case. More common are arbitrary directed CX layers which corresponds to a $\text{GCX}(B)$ with $B \in \mathbb{F}_2^{n \times n}$ still being lower/upper triangular but more sparse. This sparsity, which translates to the table T_{CZ} , can be used to further reduce the encoding cost. One might be able to move the Hadamard gates to the left/right which changes the support of the GZZ gates. We explain three different scenarios which can be easily verified by the simple structure of T_{CZ} and T_H . Obviously, the two Hadamard gates on any qubit $i > 1$ cancel if they are separated only by identities. Similarly, if there is no control or target of a CZ gate to the right of a Hadamard gate H, one can move H to the last position of the circuit (without cancellation). Otherwise, it might still be possible to push the Hadamard gate on qubit i to the left until it hits a target of some CZ gate. However, one needs to be careful to not disrupt any previously generated GZZ gates with qubit i in its support. This can be accomplished by disallowing the Hadamard gate to move across other Hadamard layers. Algorithm 1 implements these moves of the Hadamard gates by updating the table T_H accordingly.

Algorithm 2 pools multiple columns of T_{CZ} together into a single GZZ gate similar as for fully directed CX layers. This takes into account the sparsity of T_{CZ} and the positions of the Hadamard gates, i.e. T_H generated by Algorithm 1. As for fully directed CX layers, we can split a column of T_{CZ} into a two-qubit CZ gate and a GZZ gate if necessary. The algorithm starts from the left and tries to move columns of T_{CZ} , or parts of it, to the right. The following cases occur: If a column of T_{CZ} has a Hadamard gate left to the first nonzero entry, i.e. the control of the fan-out gate, then it can only be moved to the right. If a column of T_{CZ} has no Hadamard gate left to the first nonzero entry, then this column can be combined with the previous column. If a column has one Hadamard gate to the right, then the column can be split into a two-qubit CZ gate whose target is the qubit on which the Hadamard gate acts, and a GZZ gate which can be moved to the right. Due to the structure of T_{CZ} and T_H there is never a Hadamard gate to the right of the first nonzero entry, i.e. the control of the fan-out gate, of a column.

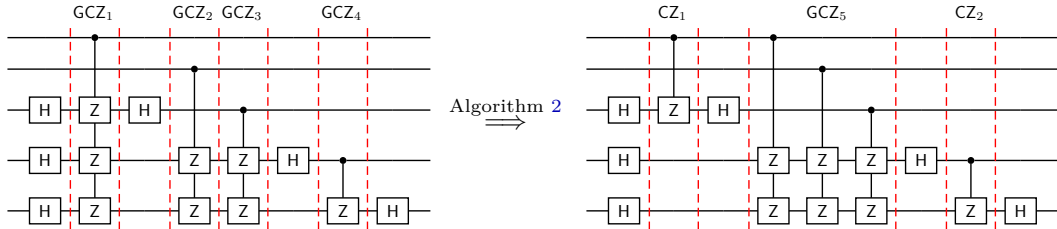
Algorithm 2 Moving CZ gates.**Input:** T_{CZ}, T_H **for** $i = 1, \dots, n-1$ **do** **if** $T_H[:, i] \wedge T_{CZ}[:, i]$ has at least one 1 **then** $\triangleright T_{CZ}[:, i]$ has Hadamard gates to the left **if** $T_H[:, i+1]$ has exactly one 1 **then** $\triangleright T_{CZ}[:, i]$ has one Hadamard gate to the right Split $T_{CZ}[:, i]$ into: $T_{CZ}^1 := T_H[:, i+1] \oplus e_i$ and $\triangleright e_i = [0, \dots, 0, 1, 0, \dots, 0]^T$ with 1 at the i th position. $T_{CZ}^2 := T_H[:, i+1] \oplus T_{CZ}[:, i]$ **if** T_{CZ}^2 has at least two 1's **then** \triangleright Check that T_{CZ}^2 is not trivial Move T_{CZ}^2 to the right. **else** $\triangleright T_{CZ}[:, i]$ has no Hadamard gates to the left Move $T_{CZ}[:, i]$ to the left.**Output:** T_{CZ} (modified)

Example. Consider the following circuit implementing an arbitrary directed CX layer on $n = 5$ qubits where the first two fan-out gates do not target qubits 2 and 3, respectively:



Algorithm 1 cancels two Hadamard gates on the second qubit which is not targeted by the first fan-out gate, i.e. $T_{CZ}[2, 1] = 0$. On the third qubit, Algorithm 1 moves the rightmost Hadamard gate one layer to the left since $T_{CZ}[3, 2] = 0$ but $T_{CZ}[3, 1] = 1$.

Algorithm 2 takes the output of Algorithm 1 and splits GCZ_1 into CZ_1 and GCZ'_1 since it has Hadamard gates to the left and right. Then it pools GCZ'_1 , GCZ_2 and GCZ_3 together into a single gate GCZ_5 :



Note that any GCZ gate is equivalent to a GZZ up to single-qubit R_Z rotations. The total encoding cost of the compiled circuit is $2 + n/2(n-1) = 12$ whereas the original circuit has the encoding cost $6 + 3 + 3 + 1 = 13$.

In this section we discussed how to implement the entangling operations of a Clifford unitary. We only need one GZZ gate to implement one fan-out gate, as one can see in Eq. (3.5), (3.10) and Hadamard gate commutation, instead of two multi-qubit gates used in the compilation schemes in Refs. [7, 8]. We further showed that we only require $n+1$ GZZ and n two-qubit CZ gates to implement a Clifford unitary on n qubits. Due to the flexibility of our GZZ gate, we further optimized the implementation of a directed CX layer to reduce the encoding cost.

The compilation of directed CX circuits via Algorithms 1 and 2 is available as a Python implementation on GitHub [37].

3.2.4 Numerical results for the directed CX layer

We demonstrate the performance of Algorithms 1 and 2 for compiling an arbitrary directed CX layer. Since the CX layer is the most costly gate layer, in the Bruhat decomposition we only present

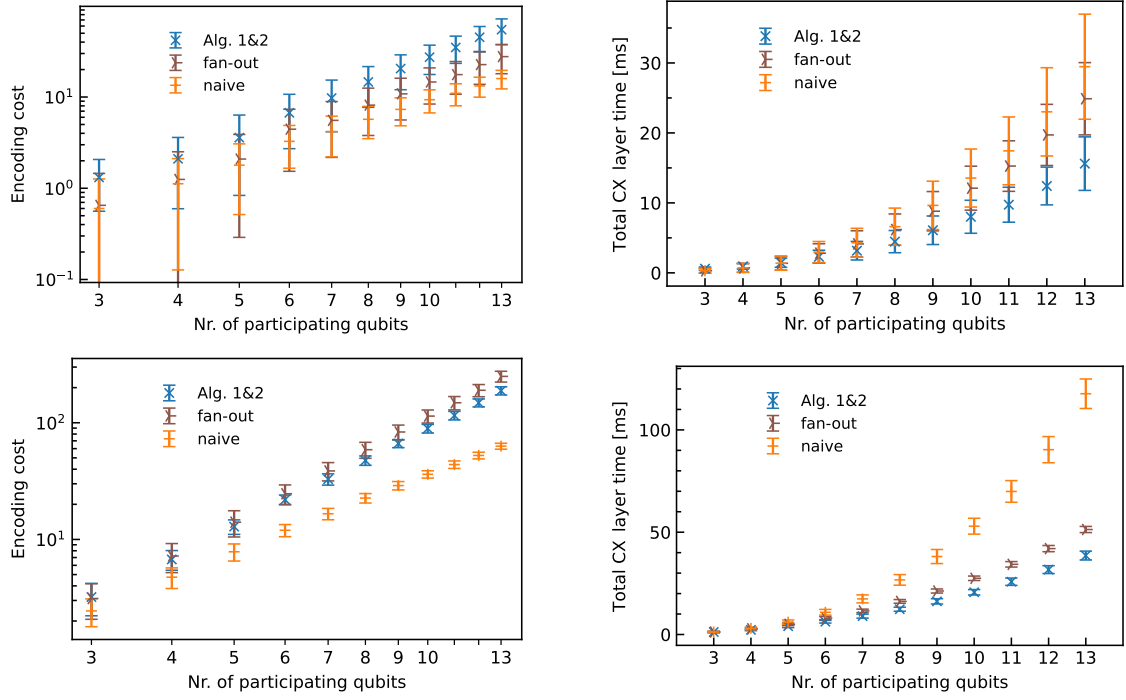


Figure 5: Comparing the performances of Algorithms 1 and 2, fan-out and the naive approach for the implementation of a random directed CX layer. The error bars show the variance over 100 samples. The numerical values are obtained using the hardware specific parameters from Section 2.3. **Top and bottom:** Sparse and dense T_{CZ} with the probability 0.2 and 0.8 for picking a “1”, respectively. **Left:** The encoding cost of a random directed CX layer on n qubits. In the top left the large variance in the encoding cost is due to the sparsity of T_{CZ} . In contrast, for dense T_{CZ} (bottom left) Algorithms 1 and 2 lead to a reduced encoding cost compared to the fan-out approach. **Right:** Neglecting local gates, for both sparse and dense T_{CZ} the Algorithms 1 and 2 yields the lowest total CX layer time.

the compilation of this layer.

Consider a directed CX layer with randomly chosen entries of the lower triangular part of T_{CZ} , and T_H as in Eq. (3.12). We distinguish between the naive implementation, the implementation of the fan-out gates directly as a GZZ gate (and local gates) and the application of Algorithms 1 and 2 to T_{CZ} and T_H . Like in Section 2.3, the naive implementation corresponds to the sequential execution of two-qubit ZZ gates. Therefore, the encoding cost and the total gate time for the naive approach is $\sum_{i < j} T_{CZ}[i, j]$ and $\sum_{i < j} T_{CZ}[i, j]/J_{ij}$, respectively.

Implementing the fan-out gate directly associates one GZZ gate with each column of T_{CZ} . We further take advantage of the sparsity of T_{CZ} . If we consider the fan-out gate represented by the i -th column of T_{CZ} , then for all j with $T_{CZ}[i, j] = 0$ we can exclude all the j -th qubits from the participation in that GZZ gate, reducing the encoding cost. Thus, for the fan-out gate implementation we have $n - 1$ symmetric matrices A with ones in the first row/column and zeros everywhere else. The total encoding cost for the fan-out approach is therefore the sum of the encoding costs for the $n - 1$ GZZ(A) gates. The same holds for the total CX layer time as the sum of the total GZZ times.

Algorithms 1 and 2 take a different advantage of the sparsity of T_{CZ} by commuting Hadamard gates and combining parts of multiple fan-out gates to one GZZ gate. As stated above, we need at most $\lfloor \frac{n-1}{2} \rfloor$ GZZ gates. If k fan-out gates are combined, the resulting GZZ gate is characterized by a symmetric matrix A where the first k rows (and also columns) can have non-zero values. The total encoding cost for the Algorithms 1 and 2 is the sum of the encoding costs of the $\lfloor \frac{n-1}{2} \rfloor$ GZZ gates plus the encoding cost of the $\lceil \frac{n-1}{2} \rceil$ ZZ gates. The same holds for the total CX layer time as the sum of the total GZZ times and the times for the $\lceil \frac{n-1}{2} \rceil$ ZZ gates. Note that we neglect local gates in our considerations.

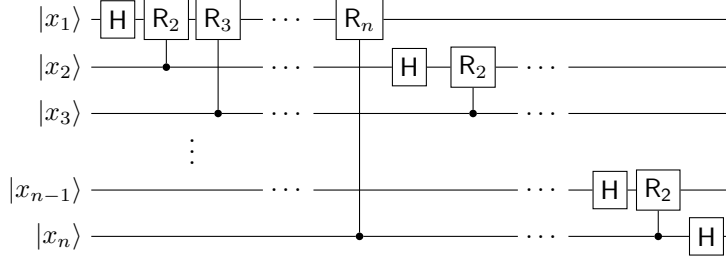


Figure 6: The quantum Fourier transform (with reversed order of the output qubits, i.e. without the swapping gates at the end).

Figure 5 shows that for a dense directed CX layer Algorithms 1 and 2 have a significant advantage in the total CX layer time over the naive and the fan-out implementation. Also, the encoding cost is reduced compared to the fan-out implementation. For sparse T_{CZ} the advantage is still visible in the total CX layer time, but the difference between the approaches is less substantial.

3.3 Quantum Fourier transform

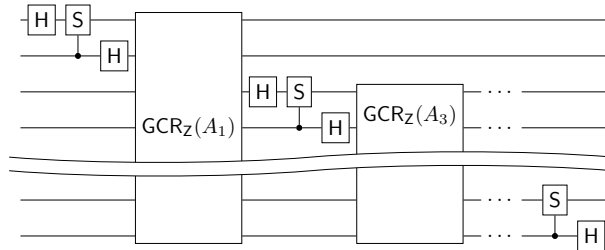
The quantum Fourier transform is an essential ingredient in many quantum algorithms. To define the corresponding unitary operator on an n -qubit register, we identify the elements of the computational basis $|x_1, \dots, x_n\rangle$ for $x_j \in \mathbb{F}_2$ with integers in binary representation, i.e. $x = \sum_{j=1}^n x_j 2^{n-j}$. The QFT is then given as

$$\text{QFT}|x\rangle = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} e^{2\pi i xy 2^{-n}} |y\rangle = \frac{1}{2^{n/2}} \bigotimes_{j=1}^n \left(|0\rangle + e^{2\pi i x 2^{-j}} |1\rangle \right). \quad (3.15)$$

The latter form immediately leads to the efficient quantum circuit in Figure 6 which uses $n(n-1)/2$ controlled R_Z -rotations with angles $2\pi/2^j$ for $j = 2, \dots, n$ and n Hadamard gates. For convenience, we introduce the shorthand notation $R_j := R_Z(2\pi/2^j)$, in particular $Z := R_1$, $S := R_2 = \sqrt{Z}$, and $T := R_3 = \sqrt[4]{Z}$. In the QFT circuit in Figure 6 we can collect for each $j < n$ the subsequent controlled R_{n-j}, \dots, R_1 to a (multi-qubit) $\text{GCR}_Z(A)$ gate specified by the $(n-j+1) \times (n-j+1)$ symmetric matrix

$$A = 2\pi \begin{pmatrix} 0 & 2^{-1} & \dots & 2^{-n+j} \\ 2^{-1} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 2^{-n+j} & 0 & \dots & 0 \end{pmatrix}. \quad (3.16)$$

Since our compilation scheme for the fully directed CX layer in Section 3.2.3 is agnostic of the rotation angles, up to local R_Z rotations, we can apply it directly to the QFT circuit. This yields the compiled QFT circuit:



Note that we are able to achieve this form without any Hadamard gate obstructing the required movement of CR_Z gates. The combination of diagonal multi-qubit gates across non-diagonal local gates has been carried out for small systems e.g. in Refs. [19, 28]. However, this approach requires numerics in unitary groups beyond $\text{SU}(2)$, which we avoid to ensure scalability (w.r.t. the Hilbert

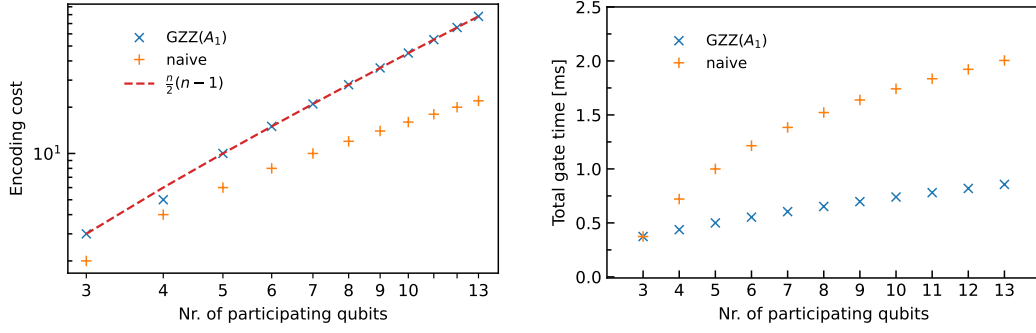


Figure 7: **Left:** The encoding cost, i.e. the different encodings, we need to implement $\text{GZZ}(A_1)$ on n qubits. For the naive approach the encoding cost only increases with $2(n-2)$, since that is the number of non-zero entries in A_1 . **Right:** Due to the exponentially small non-zero entries in A_1 even the naive approach seems to have an asymptotic linear scaling of the total gate time. As before, the numerical values are obtained using the hardware specific parameters from Section 2.3.

space dimension). We also do not address the compilation of local R_Z rotations here, see e.g. Ref. [52] which covers that topic.

After our compilation scheme we have $\lceil \frac{n-1}{2} \rceil$ CS gates (controlled- \sqrt{Z} gates) and $\lfloor \frac{n-1}{2} \rfloor$ GCR_Z gates characterized by the symmetric matrix

$$A_j := 2\pi \begin{pmatrix} 0 & 0 & 2^{-2} & \dots & 2^{-n+j} \\ 0 & 0 & 2^{-1} & \dots & 2^{-n+j+1} \\ 2^{-2} & 2^{-1} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2^{-n+j} & 2^{-n+j+1} & 0 & \dots & 0 \end{pmatrix}. \quad (3.17)$$

A $\text{GCR}_Z(A_j)$ gate can be implemented as described in Section 3.1 using a $\text{GZZ}(A_j)$ and additional single-qubit R_Z gates on the qubits they act on. For the QFT, we can push the resulting R_Z gates on the target qubits to the end of circuit, and likewise we can push the ones on the control qubits to the beginning.

In conclusion, we can apply the fully directed CX layer compiling scheme to implement a QFT circuit with the same amount of GZZ gates and the same encoding cost. The most difficult part in the QFT are the local R_Z rotations with exponentially small angles. Due to the small values in the matrices A_j one can also expect a practical issue in the implementation with too small λ_m , see Section 2.2.

Numerical results for the QFT

We provide the numerical results for the performance of the $\text{GZZ}(A_j)$ gate with A_j from Eq. (3.17) in Figure 7. It is a priori unclear how the exponentially small entries in A_j effect the time spent in each encoding λ_m . We set $j = 1$ since $\text{GZZ}(A_1)$ acts on n qubits and is therefore the most costly GZZ gate in the QFT. As in Section 2.3 we compare the result of the GZZ gate implementation via the LP (2.5) against the naive approach, i.e. the sequential implementation of two-qubit controlled R_j gates. As before, we neglect the finite recoding time, i.e. the time for executing the X gate layers, and only consider the time needed to execute a $\text{GZZ}(A_1)$ gate or the naive approach. Note that in Figure 7 we consider only a single $\text{GZZ}(A_1)$ gate, while the total QFT circuit consists of $\lfloor \frac{n-1}{2} \rfloor$ $\text{GZZ}(A_j)$ gates.

Due to the exponentially small entries in A_1 we might in practice run into the problem of too small λ_m for a QFT on a moderate amount of qubits.

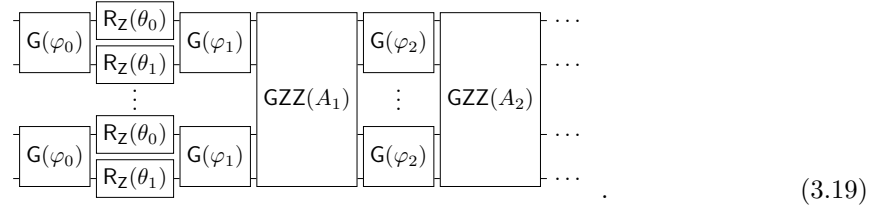
3.4 Circuit reduction for quantum chemistry applications

Simulating molecular dynamics is probably one of the main applications for quantum computations. Molecular dynamics are governed by the Coulomb Hamiltonian, consisting of the kinetic energy terms and of the Coulomb interactions between the electrons and the nuclei. It is common to neglect the kinetic terms of the nuclei which is called “Born-Oppenheimer approximation” [53].

The determination of approximate ground and excited eigenstates of the remaining electronic Hamiltonian is a hard task even on a quantum device [54]. Since simulating the time-dependent Schrödinger equation is more natural on a quantum computer there has been much effort to approximate eigenstates by solving the time-dependent Schrödinger equations [55–57]. The corresponding time evolution can be approximated by factorizing the time dependent interactions into $m-1$ layers [22]:

$$U_t \approx U_{Ext} \hat{G}(\varphi_m) \left[\prod_{k=1}^{m-1} \text{GZZ}(A_k) \hat{G}(\varphi_k) \right] \hat{R}_Z(\theta_{0,1}) \hat{G}(\varphi_0), \quad (3.18)$$

where the hat denotes layers of the gates and $G(\varphi)$ is a Givens rotation on two qubits. U_{Ext} represents the constant nucleus-nucleus Coulomb interaction and corresponds to a global phase which we henceforth omit. $A_k \in \text{Sym}_0(\mathbb{R}^n)$ denotes the total coupling matrix, characterizing the GZZ gate. The right-hand side can then be represented as the circuit



The implementation of the GZZ gates is straightforward, see Section 2, and we again do not address the compilation of local $R_Z(\theta)$ rotations in this work. Therefore, we focus on the decomposition of the Givens rotation

$$G(\varphi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) & 0 \\ 0 & -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \equiv \begin{array}{c} \text{S} \text{---} \oplus \text{---} R_Y(-\varphi) \text{---} \oplus \text{---} S^\dagger \\ \text{S} \text{---} H \text{---} \bullet \text{---} R_Y(-\varphi) \text{---} \bullet \text{---} H \text{---} S^\dagger \end{array} \quad (3.20)$$

where $R_Y(\varphi) = e^{-iY\varphi/2}$ is a rotation around the y -axis. Using the Euler decomposition, up to global phases we can express $R_Y(\varphi)$ as

$$R_Y(\varphi) \equiv \sqrt{X}^\dagger R_Z(\varphi) \sqrt{X} \equiv S^\dagger R_X(\varphi) S. \quad (3.21)$$

Inserting this into Eq. (3.20) we obtain

$$G(\varphi) \equiv \begin{array}{c} \text{S} \text{---} \sqrt{X} \text{---} \oplus \text{---} R_Z(-\varphi) \text{---} \oplus \text{---} \sqrt{X}^\dagger \text{---} S^\dagger \\ \text{S} \text{---} H \text{---} S \text{---} \bullet \text{---} R_X(-\varphi) \text{---} \bullet \text{---} S^\dagger \text{---} H \text{---} S^\dagger \end{array}. \quad (3.22)$$

Using the commutation rule

$$CX_{2,1} R_Z(\varphi)_1 \equiv ZZ_{1,2}(\varphi) CX_{2,1} \quad (3.23)$$

twice, first we commute the R_Z gate with the right CX , then conjugate the R_X gate and the left CX with Hadamard gates and commute them. This gives

$$G(\varphi) \equiv \begin{array}{c} \text{S} \text{---} \sqrt{X} \text{---} H \text{---} ZZ(-\varphi) \text{---} H \text{---} ZZ(-\varphi) \text{---} \sqrt{X}^\dagger \text{---} S^\dagger \\ \text{S} \text{---} H \text{---} S \text{---} H \text{---} \bullet \text{---} ZZ(-\varphi) \text{---} \bullet \text{---} S^\dagger \text{---} H \text{---} S^\dagger \end{array}. \quad (3.24)$$

Since any layer of ZZ gates can be implemented with only one GZZ gate, see Eq. (3.4), we can collect the parallel ZZ($-\varphi$) gates of the Givens rotation layer $\hat{G}(\varphi)$ into a single GZZ($-\varphi A_{NN}$) gate. Overall, this results in two GZZ gates per Givens rotation layer. Here, $A_{NN} \in \text{Sym}_0(\mathbb{F}_2^n)$ denotes the total coupling matrix that pairwise couples next-neighbor qubits, i.e. qubits 1 and 2, qubits 3 and 4, etc. Concretely, A_{NN} takes the form

$$A_{NN} = \begin{pmatrix} 0 & 1 & & & \\ 1 & 0 & & & \\ & & \ddots & & \\ & & & 0 & 1 \\ & & & 1 & 0 \end{pmatrix}. \quad (3.25)$$

Usually, the subdiagonal of the hardware-given coupling matrix J contains the largest values and thus corresponds to the fastest ZZ gates. This results in very fast GZZ($-\varphi A_{NN}$) gates.

A straightforward implementation of the Givens rotation layer with Eq. (3.20) results in n local R_Y rotations with arbitrary angle. On a quantum computer with a finite native gate set the rotations with arbitrary angle might be challenging to implement. Although there are methods to decompose rotations with arbitrary angle into a native gate set, see e.g. Refs. [58–60], they always come with a considerable encoding cost.

To sum up, to approximate molecular dynamics as described by the circuit (3.19) one needs $2(m+1)$ GZZ($-\varphi_k A_{NN}$) gates, $(m-1)$ GZZ(A_k) gates, n local $R_Z(\theta_{0,1})$ gates and some local Clifford gate.

3.5 Decomposing general diagonal unitaries

Since the GZZ gate is diagonal, it is natural to look for a compilation procedure to implement general diagonal unitaries using the GZZ gate together with local R_Z and Hadamard gates. We assume that we are able to implement R_Z gates with arbitrarily small angles which in practice might be difficult. Yet, there are algorithms to approximate any R_Z -angle using only Clifford+T gates [59] or Clifford+ \sqrt{T} [52]. This compilation scheme leads us beyond Clifford circuits and should be understood as a stepping stone towards more general compilation schemes.

First we introduce the representation of diagonal unitaries by phase polynomials [61, 62]. Then we show how certain terms of the phase polynomial can be implemented in parallel. We optimize the order of these parallelized layers such that the encoding cost and the total GZZ time is reduced.

Any diagonal n -qubit unitary acts as

$$U_f|\mathbf{x}\rangle := e^{2\pi i f(\mathbf{x})}|\mathbf{x}\rangle, \quad (3.26)$$

with $\mathbf{x} \in \mathbb{F}_2^n$ for some pseudo-Boolean function $f : \mathbb{F}_2^n \rightarrow \mathbb{R}$. Such a function can be uniquely expressed as a multilinear *phase polynomial* [63, Theorem 1.1],

$$f(\mathbf{x}) = \sum_{\mathbf{y} \in \mathbb{F}_2^n} \alpha_{\mathbf{y}} \chi_{\mathbf{y}}(\mathbf{x}) \quad (3.27)$$

with coefficients or phases $\alpha_{\mathbf{y}} \in \mathbb{R}$ and *parity function* $x \mapsto \chi_{\mathbf{y}}(x) = y_1 x_1 \oplus \dots \oplus y_n x_n$ for each *parity* $\mathbf{y} \in \mathbb{F}_2^n$. Additionally taking into account basis state transformations $|\mathbf{x}\rangle \mapsto |g(\mathbf{x})\rangle$ the above extends to the so-called *sum-over-paths representation*, which was used in Refs. [61, 62].

Terms in the expansion of f of the form αx_i and $\alpha(x_i \oplus x_j)$ can directly be implemented as $R_Z(\alpha)_i$ and $ZZ(\alpha)_{i,j}$ gate, respectively, and are therefore easy. We thus only consider terms with more than two variables x_i .

We define the unitary representing one term in $f(\mathbf{x})$ by

$$U_{i,\mathbf{y}}|\mathbf{x}\rangle := e^{2\pi i \alpha_{\mathbf{y}} \chi_{\mathbf{y}}(\mathbf{x})}|\mathbf{x}\rangle, \quad (3.28)$$

where i is a qubit index. The role of the subscript i gets apparent in the circuit representation of

the unitary $U_{i,\mathbf{y}}$

$$\begin{aligned}
 U_{i,\mathbf{y}} &\equiv \text{Quantum circuit diagram with } m \text{ horizontal lines labeled } x_{l_1}, \dots, x_{l_{m-1}}, x_i = x_{l_m}. \\
 &\quad \text{The circuit consists of } m \text{ layers of } CX \text{ gates. The } i\text{-th layer has a } R_Z(\alpha_{\mathbf{y}}) \text{ gate on the } i\text{-th qubit, followed by } m \text{ } CX \text{ gates.} \\
 &= \text{Quantum circuit diagram with } m \text{ horizontal lines labeled } x_{l_1}, \dots, x_{l_{m-1}}, x_i = x_{l_m}. \\
 &\quad \text{The circuit consists of } m \text{ layers of } CX \text{ gates. The } i\text{-th layer has } m \text{ Hadamard (H) gates on the } i\text{-th qubit, followed by } m \text{ } CX \text{ gates.} \\
 \end{aligned} \tag{3.29}$$

where $l = (k \in [n] : y_k \neq 0) \in [n]^m$ is a sequence containing the m non-zero components of \mathbf{y} and with the conjugating Hadamard gates, the R_Z gate and all the CX gate targets on the i -th qubit. Note that we omit $\alpha_{\mathbf{y}}$ in $U_{i,\mathbf{y}}$ since the following discussion is agnostic of the phase $\alpha_{\mathbf{y}}$. The domain of $U_{i,\mathbf{y}}$ is given by the support of the parity $\text{supp}(\mathbf{y}) := \{k \mid y_k \neq 0\}$. Since the unitaries of the two CX layers conjugating the R_Z gate in $U_{i,\mathbf{y}}$ are permutation matrices one has the freedom in choosing any $i \in \text{supp}(\mathbf{y})$. That is, for any $i, j \in \text{supp}(\mathbf{y})$, $U_{i,\mathbf{y}}$ and $U_{j,\mathbf{y}}$ implement the same unitary. Applying Eq. (3.5) to the right-hand side of Eq. (3.29), we find that each $U_{i,\mathbf{y}}$ can be implemented with two GZZ gates, one $R_Z(\alpha_{\mathbf{y}})$ gate, four Hadamard gates and some $R_Z(\pi/2)$ gates.

Clearly, two such diagonal unitaries $U_{i,\mathbf{y}}$ and $U_{i',\mathbf{y}'}$ can be implemented in parallel if their supports are disjoint, i.e. if $\text{supp}(\mathbf{y}) \cap \text{supp}(\mathbf{y}') = \emptyset$. Since our GZZ gates are time-optimal, this results in a time-optimal implementation of those unitaries, assuming as above a time-optimal implementation of the single-qubit R_Z gates. Denote by $\mathcal{S} := \{\text{supp}(\mathbf{y}) \mid \mathbf{y} \in \mathbb{F}_2^n, |\mathbf{y}| > 2\}$ the set of all supports of the parities without the easy terms. Algorithm 3 is a heuristic algorithm, which parallelizes $U_{i,\mathbf{y}}$ with disjoint supports such that the resulting support is as large as possible.

Algorithm 3 Parallelizing supports.

Input: \mathcal{S} ▷ Set of supports

$\mathcal{L} \leftarrow \emptyset$

while $\mathcal{S} \neq \emptyset$ **do**

 Choose $s_k \in \mathcal{S}$

$\mathcal{S} \leftarrow \mathcal{S} \setminus \{s_k\}$

▷ Remove s_k from \mathcal{S}

$s \leftarrow \{s_k\}$

while There exists $s_i \in \mathcal{S}$ s.t. $s \cap s_i = \emptyset$ and $\max_{s_i \in \mathcal{S}} |s \cup s_i|$ **do**

$s \leftarrow s \cup \{s_i\}$

▷ Append disjoint support with maximal union size

$\mathcal{S} \leftarrow \mathcal{S} \setminus \{s_i\}$

$\mathcal{L} \leftarrow \mathcal{L} \cup s$

▷ Append to set of parallelized layers

Output: \mathcal{L}

▷ Set of parallelized layers

Between any two consecutive unitaries $U_{i,\mathbf{y}}$ and $U_{i',\mathbf{y}'}$ with support overlap $o := \text{supp}(\mathbf{y}) \cap \text{supp}(\mathbf{y}') = s \cap s' \neq \emptyset$ two types of cancellation can happen. First, the Hadamard gates cancel, provided that they act on the same qubit. This is achieved by choosing the qubit i on which the Hadamard gates act, concretely by demanding $i = i' \in o$. Thereafter, one can combine the two CZ layers of $U_{i,\mathbf{y}}$ and $U_{i',\mathbf{y}'}$ and implement them with just one GZZ gate, see Eq. (3.5). Doing so, a second cancellation happens automatically: One can see from the right-hand side of Eq. (3.29) that between $U_{i,\mathbf{y}}$ and $U_{i',\mathbf{y}'}$, $2|o|$ CZ gates cancel and therefore $|o|$ less qubits participate in the GZZ gate. This leads to a quadratic reduction $O(|o|^2)$ of the encoding cost, which for a n qubit GZZ gate is $n(n-1)/2$.

Let $\mathcal{L}_r := \{u \in \mathcal{L} \mid |u| = r\}$, where \mathcal{L} is the set of all parallelized layers returned by Algorithm 3, be the subset of all parallelized layers that contain r disjoint supports of unitaries $U_{i,\mathbf{y}}$. The set \mathcal{L}_r is important for the placement of the Hadamard gates. For example, if two parallelized layers $u, u' \in \mathcal{L}_r$ are executed consecutively and there exist i and j for all $s_1, \dots, s_r \in u$ and $s'_1, \dots, s'_r \in u'$

such that $s_i \cap s'_j \neq \emptyset$, then all interleaving Hadamard gates cancel if they are placed on the qubits contained in $s_i \cap s'_j$.

We now extend this argument to all parallelized layers $u^{(k)} \in \mathcal{L}_r$. If there is a qubit contained in the repeated overlap of the r supports of all consecutive parallelized layers, all Hadamard gates between the parallelized layers cancel. Concretely, for the cancellation of all Hadamard gates there must exist indices $i_k \in \{1, \dots, r\}$ for all $s_1^{(k)}, \dots, s_r^{(k)} \in u^{(k)}$ such that $\bigcap_k^{|\mathcal{L}_r|} s_{i_k}^{(k)} \neq \emptyset$. We further note that if we apply $u \in \mathcal{L}_r$ and $v \in \mathcal{L}_{r'}$ successively for $r \neq r'$, then it is impossible that all interleaving Hadamard gates cancel.

The next lemma gives the encoding cost for the set of parallelized layers \mathcal{L}_r and deals with the placement of the Hadamard gates by introducing few ancilla qubits.

Lemma 3.1. *The set of parallelized layers \mathcal{L}_r can be implemented using $2r$ Hadamard gates, $|\mathcal{L}_r|/r$ non-Clifford R_Z rotations and $|\mathcal{L}_r| + 1$ GZZ gates by introducing at most r ancilla qubits.*

Proof. Each $u^{(k)} \in \mathcal{L}_r$ contains $2r$ Hadamard gates. The position of the $2r$ Hadamard gates can be chosen freely on the r supports $s_1^{(k)}, \dots, s_r^{(k)} \in u^{(k)}$. So we want to find qubits q_i in the overlap of all supports, i.e. $q_i \in \bigcap_k^{|\mathcal{L}_r|} s_{i_k}^{(k)}$ with for each $i = 1, \dots, r$. If this is not possible for qubit q_i , i.e. $\bigcap_k^{|\mathcal{L}_r|} s_{i_k}^{(k)} = \emptyset$, we add an ancilla qubit to all supports containing q_i and set q_i to that qubit. One sees that in the worst case, setting all q_1, \dots, q_r to the ancilla qubits and adding them to the supports s_1, \dots, s_r for all $u \in \mathcal{L}_r$ results in the cancellation of all interleaving Hadamard gates, leaving only $2r$ Hadamard gates. The GZZ gate count results from combining the two GZZ gates of subsequent u 's as discussed above. \square

Note that the r ancilla qubits can be reused by different \mathcal{L}_r since we do not encode any information on them. For an n -qubit diagonal unitary one needs $r \leq \lfloor \frac{n}{3} \rfloor$ ancilla qubits to ensure that all Hadamard gates cancel. The upper bound comes from the fact that we consider only supports with $|y| > 2$ so at most $\lfloor \frac{n}{3} \rfloor$ many supports can be in parallel.

Assume now, that we have r ancilla qubits or that all interleaving Hadamard gates cancel. Then one can freely permute the order of the parallelized layers. Such reordering does not change the GZZ gate count, but it possibly reduces the amount of qubits participating in the GZZ gates and therefore the encoding cost. We thus want to find an order such that the cancellation of CZ gates is maximized, i.e. we want to maximize the overlap of the supports of consecutive parallelized layers. We define the *shared support size* between two parallelized layers $u, u' \in \mathcal{L}_r$ with r elements as $S_{u,u'} := \sum_{i=0}^r |s_i \cap s'_i|$, where $s_i \in u$ and $s'_i \in u'$. Choose the ordering of elements in \mathcal{L}_r that maximizes the shared support size $S_{u,u'}$ over all pairs of parallelized layers $u, u' \in \mathcal{L}_r$. This relates to the traveling salesman problem in the formulation of Ref. [64] with each parallelized layer as a vertex and weights $S_{u,u'}$ on the edge between u and u' . Hence the reordering of the parallelized layers cannot be solved efficiently. However, there are heuristic algorithms, e.g. the Christofides algorithm [32] which only takes $O(|\mathcal{L}_r|^3)$ steps and guarantees that the solution is within a factor $3/2$ of the optimal solution.

In the worst case of the proposed compiling method for general diagonal unitaries, we consider the set of all supports \mathcal{S} without easy gates, i.e., without single- and two-qubit gates. This set has $|\mathcal{S}| = 2^n - (n + n(n-1)/2) = 2^n - n(n+1)/2$ elements, and contains for every element $s \in \mathcal{S}$ also the complement \bar{s} with $|s \cup \bar{s}| = n$. Therefore, we can always parallelize two gates, the one corresponding to s and that corresponding to \bar{s} . We thus only have \mathcal{L}_2 which has size $|\mathcal{L}_2| = |\mathcal{S}|/2 = 2^{n-1} - n(n+1)/4$. Using Lemma 3.1, this requires only two ancilla qubits and $|\mathcal{S}|/2$ GZZ gates.

In this section we proposed a method to decompose arbitrary diagonal unitaries into local gates and GZZ gates. We showed that this leads to an optimization problem which in general is NP-hard. Using heuristic algorithms might still achieve a significant reduction in the encoding cost.

Example. We illustrate the compilation scheme explained above on a diagonal unitary on five qubits. Let $\mathbf{x} \in \mathbb{F}_2^5$ and define a diagonal unitary as in Eq. (3.26) with

$$\begin{aligned} f(\mathbf{x}) = & \frac{1}{2} (x_1 \oplus x_2 + x_3 \oplus x_4 + x_4 \oplus x_5 \\ & + x_1 \oplus x_2 \oplus x_3 + x_2 \oplus x_4 \oplus x_5 \\ & + x_2 \oplus x_3 \oplus x_4 \oplus x_5). \end{aligned} \quad (3.30)$$

The factor $1/2$ is chosen for convenience and yields a $R_Z(\pi) = Z$ gate in the circuit representation of Eq. (3.29) for each term in f . Conjugating these Z gates with Hadamard gates as in Eq. (3.29) we get one $HZH = X$ gate for each term.

Note that in the above we considered only $s_i = \text{supp}(\mathbf{y}_i) > 2$, since all terms with $s_i = 2$ can be implemented with a single GZZ gate. To keep this example short we here also allow $s_i = 2$ since this gives more non-trivial possibilities to parallelize layers. We illustrate the diagonal unitary with the supports s_i in the circuit diagram below. Applying Algorithm 3 yields

$$(3.31)$$

We now group the parallelized layers u_i together in the sets $\mathcal{L}_2 = \{u_1, u_2, u_3\}$ and $\mathcal{L}_1 = \{u_4\}$. In this example we do not make use of ancilla qubits. One can see that the placement of the Hadamard gates should be on $s_1 \cap s_5 \cap s_4 = \{x_2\}$ for the supports s_1, s_5, s_4 and on $s_6 \cap s_3 \cap s_2 = \{x_4\}$ for the supports s_6, s_3, s_2 such that all the interior Hadamard gates cancel. For s_7 the Hadamard gate can be placed on x_2 or x_4 . Since $|s_4 \cap s_7| = |s_2 \cap s_7|$ both choices are equally good for the cancellation of CZ gates. We chose the Hadamard position for s_7 to be x_4 . The shared support size S_{u_i, u_j} between two parallelized layers for the set \mathcal{L}_2 can be calculated and expressed as the matrix

$$S = \begin{pmatrix} * & 4 & 3 \\ 4 & * & 2 \\ 3 & 2 & * \end{pmatrix} \quad (3.32)$$

With $S_{u_2, u_1} = 4$ and $S_{u_1, u_3} = 3$ we get the optimal order for \mathcal{L}_2 as u_2, u_1, u_3 , and $2(S_{u_2, u_1} + S_{u_1, u_3}) = 14$ CZ gates cancel. Using Eq. (3.29) and $HZH = X$, the resulting circuit is

$$(3.33)$$

One needs six GZZ gates to implement the diagonal unitary. The encoding cost for each GZZ_i in Eq. (3.33) with support s_i is $|s_i|(|s_i| - 1)/2$, such that the total implement cost is $10 + 3 + 6 + 1 + 3 + 6 = 29$. For the circuit on the left-hand side of Eq. (3.31) we require two GZZ gates for each support s_i . We have 7 supports s_i , i.e. 14 GZZ gates which leads to the total encoding cost is $2(2 + 2 + 2 + 2 + 3 + 3 + 6) = 40$. Thus, in this example our compilation scheme reduces the encoding cost by $\approx 25\%$.

4 Conclusion

In this work, we showed how to synthesize a time-optimal multi-qubit gate, the GZZ gate, on an abstract quantum computing platform. The only requirements on the platforms are that (I) single-qubit rotations can be executed in parallel, (II) it offers global Ising-type interactions with all-to-all connectivity, and (III) that there is a way to exclude certain qubits from the participation in the multi-qubit gate. We showed that and how to realize these requirements in a microwave controlled ion trap using magnetic gradient-induced coupling (MAGIC).

Arbitrary couplings between the qubits can be generated via X gate layers interleaved by the Ising-type evolution. The required X gate layers as well as the durations of the evolution times are determined by solving an LP. In numerical experiments we showed that the gate time of the resulting GZZ gates scales approximately linear with the number of participating qubits under reasonable assumptions on an implementing physical platform. Based on these time-optimal GZZ gates, we presented an improved compiling strategy for Clifford circuits, and applied this strategy to compile the QFT. Moreover, we applied the GZZ gates to the simulation of molecular dynamics, and presented a compiling strategy for general diagonal unitaries. This can be thought of as a step towards compilation strategies for arbitrary unitaries.

In the future, it will be interesting to investigate how the GZZ gates perform on a real-world ion trap, and how robust they can be made against errors similar to the error mitigation scheme for the DAQC setting [65]. Moreover, we hope that our time-optimal gate synthesis method will be applied to small ion trap registers which are embedded in a quantum processing unit (QPU) module.

Acknowledgements

We are grateful to the MIQRO collaboration, in particular to Robert Jördens for helpful discussions and valuable feedback on our manuscript and Christof Wunderlich for discussions on the platform and the QFT. Moreover, we would like to thank Christian Gogolin for making us aware of the connection to molecular dynamics and Lennart Bittel for fruitful discussions on convex optimization.

This work has been funded by the German Federal Ministry of Education and Research (BMBF) within the funding program “quantum technologies – from basic research to market” via the joint project MIQRO (grant numbers 13N15522 and 13N15521).

Appendices

A Ion trap quantum computing with microwave control

In this section we explain in more detail how qubits are encoded into cold ions stored in a trap, how MAGIC can be used to tailor interactions between these qubits, and how microwaves perform single-qubit gates. A full in-depth treatment can be found in Refs. [15, 36].

In short, pairs of hyperfine states of ions are interpreted as the computational basis states of qubits. These ions are cooled down to form a “Coulomb crystal”, a stable configuration confined by the trap potential. An inhomogeneous magnetic field superimposed with the trap makes the crystal equilibrium depend on the internal state (the qubits). Changes in the internal state hence lead to excitations (phonons), which can be interpreted as interactions between the qubits. The position-dependent Zeeman effect makes individual qubit transitions addressable. Microwaves drive Rabi oscillations on these transitions to perform single-qubit gates.

A.1 Ytterbium ions and qubits

We consider ions with nuclear spin $I = \frac{1}{2}$ and total electron angular momentum $J = \frac{1}{2}$, for example single-ionized Ytterbium-171 ($^{171}\text{Yb}^+$). These values imply that the “ground state” (the lowest main quantum number) is spanned by four “hyperfine” sublevels. Since nuclear spin and electron angular momentum couple to each other through an interaction term $\propto \mathbf{I} \cdot \mathbf{J}$, the full Hamiltonian is not diagonal in the naive product basis $|m_I, m_J\rangle$. One instead introduces the total angular momentum $\mathbf{F} := \mathbf{I} + \mathbf{J}$. The corresponding quantum numbers F and $m_F = -F, \dots, F$ can be used to label the hyperfine eigenstates, which group into a singlet $|F = 0, m_F = 0\rangle$ and a triplet $|F = 1, m_F = -1, 0, +1\rangle$. The singlet forms the overall ground state and is separated by an energy gap from the triplet states, which all have the same energy. The energy difference is called the “hyperfine structure constant” of the ion. This quantity is usually determined experimentally and typically lies in the microwave regime. For $^{171}\text{Yb}^+$, it corresponds to a frequency of $f_0 = 12.642812118466(2)\text{GHz}$ [66].

Placing the ion in an external magnetic field \mathbf{B} yields new dipole terms $\propto \mathbf{B} \cdot \mathbf{I}$ and $\propto \mathbf{B} \cdot \mathbf{J}$ in the Hamiltonian. Since these new terms do not preserve F , the $|F, m_F\rangle$ basis is no longer an eigenbasis. However, they still preserve m_F , so the new energy eigenstates are superpositions of states with equal m_F . The “outer” triplet states $|F = 1, m_F = \pm 1\rangle$ remain basis states due to their unique m_F , but $|F = 0, m_F = 0\rangle$ and $|F = 1, m_F = 0\rangle$ form two orthogonal linear combinations with amplitudes depending on the field strength $B := |\mathbf{B}|$. Also the eigenvalues of the Hamiltonian become B -dependent, and the previous triplet degeneracy is lifted. For the present case of four states, the problem can be solved analytically leading to the Breit-Rabi formula [29]. We are mainly interested in the weak field limit $B \rightarrow 0$, commonly known as the Zeeman effect. The energy shift of the states $|F = 1, m_F = \pm 1\rangle$ is exactly proportional to B (“linear Zeeman effect”), while the other two eigenvalues are field-independent to first order (“magnetic insensitive”). However, they do have a higher-order dependence starting from $\propto B^2$, which is known as the “quadratic Zeeman effect”, see also Figure 1b. Not only are these eigenvalues close to the zero-field values, but also the linear combinations have a dominant component of either $|F = 0, m_F = 0\rangle$ or $|F = 1, m_F = 0\rangle$. So although this is technically wrong, one casually keeps the $|F, m_F\rangle$ nomenclature for the basis states even in the presence of a small magnetic field.

We now wish to “encode” a qubit into this four-dimensional state space, i.e. to pick two of the energy eigenstates as computational basis states $|0\rangle$ and $|1\rangle$. A priori, there are six possible choices. The Zeeman splitting between the $F = 1$ states is typically on the order of MHz, so orders of magnitude smaller than the hyperfine splitting. Also, the transitions between $|F = 1, m_F = 0\rangle$ and the other two triplet states are degenerate to first order in B . This is why one usually chooses $|0\rangle := |F = 0, m_F = 0\rangle$ and only considers the three remaining possibilities which we refer to as

$$\begin{aligned} \sigma^+ \text{ qubit} & \quad \text{with } |1\rangle := |F = 1, m_F = +1\rangle, \\ \pi \text{ qubit} & \quad \text{with } |1\rangle := |F = 1, m_F = 0\rangle, \\ \sigma^- \text{ qubit} & \quad \text{with } |1\rangle := |F = 1, m_F = -1\rangle. \end{aligned}$$

One may also label the qubits by the m_F of their $|1\rangle$ state. The B -dependent transitions frequencies between $|0\rangle$ and $|1\rangle$ can then be written

$$\omega(m_F, B) = 2\pi f_0 + m_F \frac{\mu B}{\hbar} + O(B^2), \quad (\text{A.1})$$

where μ has units of a magnetic moment ($\mu \approx \mu_B$, the Bohr magneton, for $^{171}\text{Yb}^+$). This reflects both the quadratic ($m_F = 0$) and the linear ($m_F = \pm 1$) Zeeman effect.

A.2 Trap Hamiltonian and MAGIC

So far, we only considered a single ion. In an ion trap quantum computer, N such ions are confined in an (effective) potential generated by DC and RF electrodes. We assume that the potential geometry and N have been selected such that the ions form a “linear Coulomb crystal”, i.e. a one-dimensional chain stabilized by the external trap potential and the mutual Coulomb repulsion of the ions, see Figure 1a. If the crystal has been cooled sufficiently, we can disregard the radial dynamics and focus on the direction of the ion chain which we choose as the z axis. The system is described by a Hamiltonian with three contributions: the kinetic energy $T(\mathbf{p}) \propto \mathbf{p}^2$ of the ions with momenta $\mathbf{p} = (p_1, \dots, p_N)^T$, and the external and Coulomb potentials

$$V_{\text{tot}}(\mathbf{z}) := \sum_{i=1}^N V_{\text{ext}}(z_i) + K \sum_{i < j}^N \frac{1}{|z_i - z_j|}, \quad (\text{A.2})$$

with ions positions $\mathbf{z} := (z_1, \dots, z_N)^T$ and Coulomb constant K . Again, after sufficient cooling it is reasonable to assume that the system is close to an equilibrium configuration $\bar{\mathbf{z}}$. The potential can then be expanded in terms of the elongation $\mathbf{q} := \mathbf{z} - \bar{\mathbf{z}}$. The first order term vanishes due to equilibrium, such that

$$V_{\text{tot}}(\bar{\mathbf{z}} + \mathbf{q}) = V_{\text{tot}}(\bar{\mathbf{z}}) + \frac{1}{2} \mathbf{q}^T \mathbf{H}_V(\bar{\mathbf{z}}) \mathbf{q} + O(|\mathbf{q}|^3), \quad (\text{A.3})$$

where H_V is the Hessian matrix of V_{tot} . We drop the constant term and neglect the higher-order terms in the following (harmonic approximation). This results in a quadratic many-body Hamiltonian with a well-known spectrum of phonon excitations.

We add two more ingredients to the system: An inhomogeneous magnetic field with constant gradient B_1 in z direction,

$$\mathbf{B}(z) = B(z)\hat{e}_z, \quad B(z) = B_0 + B_1 z, \quad (\text{A.4})$$

and the internal dynamics of ionic qubits with chosen bases specified by $\mathbf{m}_F := (m_{F,1}, \dots, m_{F,N})^T$. Plugging Eq. (A.4) into Eq. (A.1) yields a qubit-specific transition frequency that depends on both z_i and $m_{F,i}$. In the Hamiltonian, the splitting can be expressed using the Pauli Z operator of the qubit. Expanded to first order in \mathbf{q} , the corresponding term reads

$$\begin{aligned} -\frac{\hbar}{2} \sum_{i=1}^N \omega(m_{F,i}, B(z_i)) Z_i &= -\frac{\hbar}{2} \sum_{i=1}^N \left(\omega_i^{(0)} + m_{F,i} \frac{\mu B_1}{\hbar} q_i \right) Z_i \\ &= -\frac{\hbar}{2} \sum_{i=1}^N \omega_i^{(0)} Z_i - \frac{\mu B_1}{2} \mathbf{q}^T (\mathbf{m}_F \circ \mathbf{Z}), \end{aligned} \quad (\text{A.5})$$

with $\mathbf{Z} := (Z_1, \dots, Z_N)^T$ the local Z operators and frequencies $\omega_i^{(0)}$ that depend on $m_{F,i}$, \bar{z}_i and various constants. The first term of the result is a typical Zeeman term involving only the qubits, while the second term couples external (\mathbf{q}) and internal (\mathbf{Z}) degrees of freedom. This last term is the central ingredient of magnetic gradient-induced coupling (MAGIC).

Collecting all contributions gives the Hamiltonian

$$H = T(\mathbf{p}) + \frac{1}{2} \mathbf{q}^T H_V(\bar{\mathbf{z}}) \mathbf{q} - \frac{\mu B_1}{2} \mathbf{q}^T (\mathbf{m}_F \circ \mathbf{Z}) - \frac{\hbar}{2} \sum_{i=1}^N \omega_i^{(0)} Z_i. \quad (\text{A.6})$$

It is still second-order in the external degrees of freedom, but no longer purely quadratic. We fix this by completing the square in \mathbf{q} ,

$$\begin{aligned} H &= T(\mathbf{p}) + \frac{1}{2} \left(\mathbf{q} - \frac{\mu B_1}{2} H_V^{-1}(\bar{\mathbf{z}}) (\mathbf{m}_F \circ \mathbf{Z}) \right)^T H_V(\bar{\mathbf{z}}) \left(\mathbf{q} - \frac{\mu B_1}{2} H_V^{-1}(\bar{\mathbf{z}}) (\mathbf{m}_F \circ \mathbf{Z}) \right) \\ &\quad - \frac{\hbar}{2} \sum_{i=1}^N \omega_i^{(0)} Z_i - \frac{1}{2} \left(\frac{\mu B_1}{2} \right)^2 (\mathbf{m}_F \circ \mathbf{Z})^T H_V^{-1}(\bar{\mathbf{z}}) (\mathbf{m}_F \circ \mathbf{Z}). \end{aligned} \quad (\text{A.7})$$

The existence of the inverse Hessian $H_V^{-1}(\bar{\mathbf{z}})$ is guaranteed by the assumption of a stable equilibrium, which implies that H_V is positive definite. Note also that H_V and its inverse are symmetric matrices. In Eq. (A.7), it is evident that the first line is quadratic in the external degrees of freedom “up to a state dependent translation”, while the second line only involves the qubits. Formally, this means that we can conjugate the Hamiltonian with the unitary

$$U := \exp \left(i \frac{\mu B_1}{2\hbar} \mathbf{p}^T H_V^{-1}(\bar{\mathbf{z}}) (\mathbf{m}_F \circ \mathbf{Z}) \right) \quad (\text{A.8})$$

to separate external and internal dynamics. Noting that U commutes with all terms of H except for the one involving \mathbf{q} , this results in

$$\tilde{H} := U H U^\dagger = H_{\text{phonons}}(\mathbf{p}, \mathbf{q}) + H_{\text{qubits}}(\mathbf{Z}), \quad (\text{A.9})$$

with $H_{\text{phonons}}(\mathbf{p}, \mathbf{q})$ being a standard many-body phonon Hamiltonian that is easily solvable in normal coordinates. The internal part is simply the second line of Eq. (A.7), which we rewrite as

$$H_{\text{qubits}}(\mathbf{Z}) = -\frac{\hbar}{2} \sum_{i=1}^N \omega_i^{(0)} Z_i - \frac{\hbar}{2} \mathbf{Z}^T J(\mathbf{m}_F) \mathbf{Z}. \quad (\text{A.10})$$

Here, we used the Hadamard product identity (for vectors \mathbf{a}, \mathbf{b} and a matrix C)

$$(\mathbf{a} \circ \mathbf{b})^T C (\mathbf{a} \circ \mathbf{b}) = \mathbf{a}^T (C \circ \mathbf{b} \mathbf{b}^T) \mathbf{a} \quad (\text{A.11})$$

and defined

$$\begin{aligned} J(\mathbf{m}_F) &:= J \circ \mathbf{m}_F \mathbf{m}_F^T, \\ J &:= \frac{1}{\hbar} \left(\frac{\mu B_1}{2} \right)^2 \mathbf{H}_V^{-1}(\bar{\mathbf{z}}). \end{aligned} \quad (\text{A.12})$$

The two terms of Eq. (A.10) commute, so we can treat them separately, or remove the first one entirely by another unitary transformation. In any case, the local operators Z_i do not entangle the qubits, so we focus our analysis on the second term. The diagonal entries of $J(\mathbf{m}_F)$ simply generate a constant term $-\frac{\hbar}{2} \text{Tr}[J(\mathbf{m}_F)] \mathbb{1}$, which amounts to a global and hence unobservable phase. Without loss of generality, we can thus set the diagonal terms of J to zero. Finally, for the discussion on the logical layer, we choose units in which $\hbar = 1$. After these simplifying steps, Eq. (A.10) becomes the Hamiltonian (1.1) on which our analysis is based.

We provide Python code for the computation of coupling matrices based on the trap potential and physical parameters on GitHub [37].

A.3 Microwaves and single-qubit gates

A comprehensive discussion of the physics of microwave-controlled single-qubit gates, in particular within the MAGIC scheme, goes even beyond the scope of this appendix. It involves common concepts from quantum optics like the rotating wave approximation and Lamb-Dicke parameter, extended to the situation with a permanent inhomogeneous magnetic field. This extension is crucial, because it is the magnetic gradient that enables reasonable gate times even for microwave radiation which are otherwise much slower than optical gates. We recommend Refs. [15, 36] for more details.

Important for our analysis is that resonant microwave excitation of a hyperfine transition induces a Rabi Hamiltonian on the corresponding qubit,

$$H_{\text{Rabi}}(\Omega, \phi) = \frac{\hbar\Omega}{2} (\cos(\phi)X + \sin(\phi)Y). \quad (\text{A.13})$$

Here, X and Y are Pauli operators, Ω is the Rabi frequency which is proportional to the microwave amplitude, and ϕ is a parameter controlled by the relative phase between the microwave and essentially the qubit's Larmor precession. Resonant excitation means that the microwave frequency matches the transition frequency $\omega(m_F, B)$ of the targeted qubit. While the microwave is switched on, one (or several, if the experimental setup allows for multitone signals) Hamiltonian(s) of the form (A.13) act simultaneously with (A.10), potentially with different Rabi frequencies Ω_i . As these Hamiltonians do not commute, the time evolution of the system usually eludes exact analytical treatment. However, in applications one aims to make the Rabi frequencies orders of magnitude larger than the entries of J , which govern the entangling dynamics. This allows for the approximation that the gates induced by the Hamiltonian (A.13) are instantaneous, or equivalently that the Hamiltonian (A.10) is negligible while the microwave is on. The validity of this approximation and the errors introduced by it are discussed in Section 2.2 and Ref. [26]. Time evolution under the Rabi Hamiltonian (A.13) for a time θ/Ω implements a family of Bloch rotation gates,

$$\mathbf{R}(\theta, \phi) := \exp\left(-\frac{i\theta}{\hbar\Omega} H_{\text{Rabi}}(\Omega, \phi)\right). \quad (\text{A.14})$$

On the Bloch sphere, they rotate by an angle θ around an axis in the xy -plane specified by azimuth ϕ . The X and H gates necessary for our analysis are generated by

$$\mathbf{X} = i\mathbf{R}(\pi, 0), \quad \mathbf{H} = i\mathbf{R}(\pi, 0)\mathbf{R}\left(\frac{\pi}{2}, \frac{\pi}{2}\right). \quad (\text{A.15})$$

A.4 Physical and virtual recoding

As mentioned before, the π qubit with $m_F = 0$ is called “magnetic insensitive” because it depends on B only in second order. This can be used to exclude (up to quadratic corrections) qubits from the interaction and perform “subset operations”. Corresponding rows and columns in the coupling matrix $J(\mathbf{m}_F)$ in Eq. (A.12) manifestly vanish. However, insensitivity also means that all π qubits have roughly the same transition frequency, see Eq. (A.1). Hence, they are not separately addressable, and we cannot perform independent single-qubit operations on them. The advantages and disadvantages of π qubits naturally pose the question, whether one can change the qubit basis during computation.

This can be answered in the affirmative, but one intermediately has to consider (at least) three-dimensional subspaces instead of qubits. Let X_{\pm} denote the X gate on the σ^{\pm} qubit of the same ion. These gates “swap” the two states on which they act, and one can easily confirm that the local sequences $X_+ X_- X_+$ and $X_- X_+ X_-$ both effectively swap the states $|F = 1, m_F = +1\rangle$ and $|F = 1, m_F = -1\rangle$. Thus, they “recode” between the σ^+ and σ^- qubit.

Coding into and out of the π qubit is slightly more complicated due to the mentioned degenerate transitions. Yet we can use the global swap-like operation $X_0^{\otimes N}$, which applies an X gate to all π qubits at once (assuming that shifts due to the quadratic Zeeman effect do not become prohibitive). Sequences of the form

$$X_0^{\otimes N} (\dots \otimes I \otimes \dots \otimes X_+ \otimes \dots \otimes X_- \otimes \dots) X_0^{\otimes N}, \quad (\text{A.16})$$

i.e. with an arbitrary combination of local X_{\pm} and idle gates in between two global $X_0^{\otimes N}$ gates, achieve a recoding between π and σ^{\pm} qubits (for an X_{\pm} gate) or just swap back and forth, effectively doing nothing (for an idle gate).

The described sequences of three rounds of microwave gates are able to physically manipulate the way quantum information is encoded in the ions. However, our main concern is not how information is stored, but how it can be processed. To that end one usually does not need to recode between the physical σ^+ and σ^- qubits, but can instead use single layers of X gates (on the physical qubit in use) to emulate the opposite encoding. This can significantly reduce the number of single-qubit gates in a circuit, and we make heavy use of this possibility. Using the notation of Eq. (1.3), a detailed derivation of this technique reads

$$\begin{aligned} X^s \exp \left(\frac{it}{2} Z^T J(\mathbf{m}_F) Z \right) X^s &= \exp \left(\frac{it}{2} X^s Z^T X^s J(\mathbf{m}_F) X^s Z X^s \right) \\ &= \exp \left(\frac{it}{2} ((-1)^s \circ Z)^T J(\mathbf{m}_F) ((-1)^s \circ Z) \right) \\ &= \exp \left(\frac{it}{2} Z^T \left(J(\mathbf{m}_F) \circ ((-1)^s ((-1)^s)^T) \right) Z \right) \\ &= \exp \left(\frac{it}{2} Z^T J((-1)^s \circ \mathbf{m}_F) Z \right), \end{aligned} \quad (\text{A.17})$$

where we used that Pauli strings X^s are self-inverse and applied the Hadamard product identity Eq. (A.11). Negative signs in $(-1)^s$ exchange, from the perspective of the J coupling, the σ qubits, but leave π qubits invariant. If this technique is used consecutively, subsequent X gate layers can be combined into a single one,

$$X^s X^{s'} = X^{s \oplus s'}, \quad (\text{A.18})$$

saving even more single-qubit gates, as detailed in the main text.

B Frame theory

In this section we connect our approach to frame theory, which is concerned with spanning sets for Hilbert spaces, i.e. with generalizations of the notion of a basis, see e.g. Ref. [67].

We consider the space of symmetric matrices with vanishing diagonal that is defined by

$$\text{Sym}_0(\mathbb{R}^n) := \{M \in \text{Sym}(\mathbb{R}^n) \mid M_{ii} = 0 \ \forall i \in [n]\}. \quad (\text{B.1})$$

Moreover, we denote the set of outer products of all possible encodings $\mathbf{m} = (-1)^{\mathbf{b}}$, where $\mathbf{b} \in \mathbb{F}_2^n$, by

$$\mathcal{V} := \{\mathbf{m}\mathbf{m}^T \mid \mathbf{m} \in \{-1, +1\}^n, m_n = +1\}. \quad (\text{B.2})$$

We further define what it means for a frame to be harmonic:

Definition B.1. [67, Definition 11.1] Let G be a finite abelian group, and let \hat{G} be the set of irreducible characters of G . A tight frame $\{\phi_i\}_{i \in J}$ for \mathbb{R}^k is called a *harmonic* if it is unitarily equivalent to

$$(\xi|_J)_{\xi \in \hat{G}} \subset \mathbb{R}^{|J|} \equiv \mathbb{R}^k, \quad (\text{B.3})$$

where $J \subset G$ and $|J| = k$.

The following theorem extends Theorem 2.2 and shows that \mathcal{V} is a frame with certain properties.

Theorem B.2. \mathcal{V} is a balanced equal-norm harmonic tight frame for $\text{Sym}_0(\mathbb{R}^n)$.

Proof. Let $k = n(n-1)/2 = \dim(\text{Sym}_0(\mathbb{R}^n))$ and $l = 2^{n-1} = |\mathcal{V}|$. We denote the synthesis operator of \mathcal{V} by $V : \mathbb{R}_{\geq 0}^l \rightarrow \text{Sym}_0(\mathbb{R}^n) : \boldsymbol{\lambda} \mapsto \sum_{\mathbf{m}} \lambda_{\mathbf{m}} \mathbf{m}\mathbf{m}^T$ which can be represented by a matrix $V \in \{-1, +1\}^{l \times k}$. We further denote the column $\mathbf{v}(\mathbf{m})$ of V , containing the lower triangular elements of $\mathbf{m}\mathbf{m}^T$. First we want to show that each row of V corresponds to a row of a Hadamard matrix. Since the elements of any Hadamard matrix $H_{\mathbf{y}, \mathbf{x}}$ (with normalization factor) are given by Walsh functions $H_{\mathbf{y}, \mathbf{x}} = 2^{-n/2} W_{\mathbf{y}}(\mathbf{x}) = 2^{-n/2} (-1)^{\mathbf{y} \cdot \mathbf{x}}$ for $\mathbf{y}, \mathbf{x} \in \mathbb{F}_2^n$, this amounts to showing that the rows of V correspond to Walsh functions $W_{\mathbf{y}}(\mathbf{x}) := (-1)^{\mathbf{y} \cdot \mathbf{x}}$, where $\mathbf{y} \cdot \mathbf{x} = \sum_i y_i x_i$.

Writing $\mathbf{m} = (-1)^{\mathbf{b}}$, the components of the matrix $\mathbf{m}\mathbf{m}^T$ are $(\mathbf{m}\mathbf{m}^T)_{gh} = (-1)^{b_g \oplus b_h}$. Then the entries of the columns $\mathbf{v}(\mathbf{m})$ are $\mathbf{v}(\mathbf{m})_{gh} = (-1)^{b_g \oplus b_h} = (-1)^{\mathbf{b} \cdot (\mathbf{e}^g \oplus \mathbf{e}^h)}$, where the tuple g, h is the index of the row of V and $\mathbf{e}^i \in \mathbb{F}_2^n$ denotes the i -th standard unit vector.

This shows that $\mathbf{v}(\mathbf{m})_{gh} = W_{\mathbf{b}}(\mathbf{e}^g \oplus \mathbf{e}^h)$. So the $\mathbf{e}^g \oplus \mathbf{e}^h$ encodes the row indices and the \mathbf{b} encodes the column indices of the Hadamard matrix. Since we consider all $\mathbf{b} \in \mathbb{F}_2^n$ we have all columns of the Hadamard matrix. This shows that each row of V corresponds to a row of a Hadamard matrix.

Since the rows/columns of a Hadamard matrix form an orthonormal basis one can use the row construction of tight frames from orthogonal projections [67, Theorem 2.3] to get an equal-norm tight frame. Furthermore, the Hadamard matrix is the character table of the cyclic group $C_2^n = C_2 \times \cdots \times C_2$ which is abelian. This can be seen by its recursive definition $H_n = H_1 \otimes H_{n-1}$, where H_1 is the character table of C_2 . Thus, Ref. [67, Theorem 11.1] implies that this equal-norm tight frame is also a harmonic.

It is balanced if $\sum_{\mathbf{b}} (-1)^{b_g \oplus b_h} = 0$ for all $g, h = 1, \dots, n$ with $g \neq h$. Since we only consider the lower/upper triangular matrix without the diagonal $g = h$, our frame is indeed balanced. \square

Corollary B.3. The normalized frame $\{\mathbf{m}\mathbf{m}^T/\sqrt{k}\}_{\mathbf{m} \in \{-1, +1\}^n}$ for $\text{Sym}_0(\mathbb{R}^n)$ forms a spherical 2-design.

Proof. The normalized frame elements $\mathbf{m}\mathbf{m}^T/\sqrt{k}$ form a set of unit vectors in \mathbb{R}^k and $\{\mathbf{m}\mathbf{m}^T/\sqrt{k}\}_{\mathbf{m} \in \{-1, +1\}^n}$ is a normalized balanced tight frame by Theorem B.2. Then, by Refs. [68, Proposition 1.2] and [67, Proposition 6.1], $\{\mathbf{m}\mathbf{m}^T/\sqrt{k}\}_{\mathbf{m} \in \{-1, +1\}^n}$ is also a spherical 2-design. \square

Theorem B.4. The entries of the Gram matrix $P = V^T V$ are given by

$$P_{\mathbf{m}, \mathbf{m}'} = \langle \mathbf{v}(\mathbf{m}), \mathbf{v}(\mathbf{m}') \rangle = \frac{n}{2}(n-1) - 2\Delta_{\mathbf{b}, \mathbf{b}'}(n - \Delta_{\mathbf{b}, \mathbf{b}'}), \quad (\text{B.4})$$

where $\mathbf{v}(\mathbf{m})$ are the columns of V , containing the lower triangular elements of $\mathbf{m}\mathbf{m}^T$, and $\Delta_{\mathbf{b}, \mathbf{b}'} := |\mathbf{b} \oplus \mathbf{b}'|$ is the Hamming distance between \mathbf{b} and \mathbf{b}' .

Proof. As in the proof above, write the (g, h) -th entry of the column $\mathbf{v}(\mathbf{m})$ as

$$\mathbf{v}(\mathbf{m})_{gh} = (-1)^{b_g \oplus b_h}. \quad (\text{B.5})$$

This yields

$$\begin{aligned}
\langle \mathbf{v}(\mathbf{m}), \mathbf{v}(\mathbf{m}') \rangle &= \sum_{g < h} (-1)^{b_g \oplus b_h} (-1)^{b'_g \oplus b'_h} \\
&= \sum_{g < h} (-1)^{b_g \oplus b'_g \oplus b_h \oplus b'_h} \\
&= \sum_{g < h} (-1)^{c_g \oplus c_h},
\end{aligned} \tag{B.6}$$

where we set $\mathbf{c} := \mathbf{b} \oplus \mathbf{b}'$. The summand after the last equation is the outer product $\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T$ for $\tilde{\mathbf{m}} \in \{-1, +1\}^n$ and $\tilde{\mathbf{m}} = (-1)^{\mathbf{c}}$. If the Hamming distance $\Delta_{\mathbf{b}, \mathbf{b}'}$ vanishes, then $\tilde{\mathbf{m}} = (+1, \dots, +1)$ and $\langle \mathbf{v}(\mathbf{m}), \mathbf{v}(\mathbf{m}') \rangle = n(n-1)/2$ which is the maximal entry of the Gram matrix located at the diagonal. If $\Delta_{\mathbf{b}, \mathbf{b}'} \neq 0$, $\tilde{\mathbf{m}}$ contains $\Delta_{\mathbf{b}, \mathbf{b}'}$ -many summands -1 , such that the lower triangular part of $\tilde{\mathbf{m}}\tilde{\mathbf{m}}^T$ contains $\Delta_{\mathbf{b}, \mathbf{b}'}(n - \Delta_{\mathbf{b}, \mathbf{b}'})$ -many summands -1 . Therefore,

$$\langle \mathbf{v}(\mathbf{m}), \mathbf{v}(\mathbf{m}') \rangle = \frac{n}{2}(n-1) - 2\Delta_{\mathbf{b}, \mathbf{b}'}(n - \Delta_{\mathbf{b}, \mathbf{b}'}). \tag{B.7}$$

□

C Convex optimization arguments for the linear program (LP)

We investigate the sparsity and geometric properties of the optimal solutions of the LP (2.5). First, we prove Proposition 2.3, which is a standard result in linear programming:

Proposition C.1 (Sparse optimal solutions). *There is an optimal solution to the LP (2.5) with sparsity $\leq n(n-1)/2$ for every $M \in \text{Sym}_0(\mathbb{R}^n)$. The simplex algorithm is guaranteed to return such an optimal solution.*

Proof. The LP (2.5) has $m = n(n-1)/2$ equality constraints. The feasible polytope is the convex polytope obtained by intersecting the $(n-m)$ -dimensional subspace defined by those with the positive cone $x \geq 0$. To define a vertex of the feasible polytope, a point x has to saturate at least $n-m$ many inequalities $x_i \geq 0$, hence, it has at most $n - (n-m) = m$ many non-zero entries. The optimal solutions are obtained by minimizing over the feasible polytope and thus correspond, in general, to a face of this polytope. Any vertex of this face defines an optimal solution which has to be at least $m = n(n-1)/2$ -sparse. The last statement follows since the simplex algorithm only returns vertices of the feasible polytope. □

Alternative algorithms like interior point methods should be avoided. Since those are not guaranteed to return vertices of the feasible polytope if there is an entire face of optimal solutions, their solution will generally be a dense vector.

We can say a bit more about the solutions of the LP (2.5) by geometrical observations. These observations hold more generally for LPs of the following form.

Definition C.2. *Suppose that vectors $v_1, \dots, v_N \in \mathbb{R}^d$ are the vertices of a full-dimensional polytope P and the origin is contained in the convex hull of P , i.e. $0 \in \text{conv}(P)$. Given a vector $u \in \mathbb{R}^d$, we define the following linear program:*

$$\begin{aligned}
&\text{minimize} && \langle \mathbf{1}, x \rangle = \sum_{i=1}^N x_i \\
&\text{subject to} && u = Vx, \\
&&& x \geq 0
\end{aligned} \tag{C.1}$$

where $V = \sum_{i=1}^N v_i e_i^\top \in \mathbb{R}^{d \times N}$.

Lemma C.3. *In the setting of Definition C.2, the following holds:*

- (i) *The point $u \in \mathbb{R}^d$ lies in a cone generated by a face F of P . In particular, there is a feasible solution x of the LP (C.1) with support only on the vertices of F .*

- (ii) Every feasible solution with support on the vertices of F has the same objective value, namely $\langle f, u \rangle$ where f is a normal vector of F , i.e. $F \subset \{y \in \mathbb{R}^d \mid \langle f, y \rangle = 1\}$. This value does not depend on the choice of normal vector.
- (iii) There is a feasible solution with sparsity $\leq \dim F + 1 \leq d$ and objective value $\langle f, u \rangle$.
- (iv) The optimal solutions of the LP (C.1) correspond exactly to the possible conical combinations of u in $\text{cone}(F^*) := \{\sum_i x_i v_i \mid v_i \in F^*, x_i \in \mathbb{R}_{\geq 0}\}$ where F^* is the lowest-dimensional face of P such that $u \in \text{cone}(F^*)$. In particular, the minimum is given by $\langle f, u \rangle$ where f is some normal vector of F^* .

Proof. Statement (i) follows from the observation that given a point $u \in \mathbb{R}^d \setminus 0$, there is a unique $\lambda > 0$ such that λu lies on a face F of P . Thus, $\text{cone}(F)$ contains u . More precisely, the cones spanned by the facets of P form a partition of \mathbb{R}^d where the intersections between any two cones is either $\{0\}$ or a cone spanned by a lower-dimensional face of P .

For statement (ii), let w.l.o.g. v_1, \dots, v_s be the vertices that lie in F . By assumption, $u \in \text{cone}(v_1, \dots, v_s)$ and hence $u = \sum_{i=1}^s x_i v_i$. We find

$$\langle f, u \rangle = \sum_{i=1}^s x_i \langle f, v_i \rangle = \sum_{i=1}^s x_i = \langle \mathbf{1}, x \rangle. \quad (\text{C.2})$$

Note that the objective value $\langle f, u \rangle$ is necessarily the same, for any choice of normal vector.

Statement (iii) follows by triangulating the face F with simplices. These simplices have $\dim F + 1$ vertices and u has to lie in the cone spanned by one of those. Thus, there is a feasible solution with sparsity at most $\dim F + 1 \leq d$ and objective value $\langle f, u \rangle$.

Finally, let x^* be an optimal solution. Suppose that there is a $i \in \text{supp}(x^*)$ such that the vertex v_i is not in F . Then, we can find a supporting hyperplane of F with normal vector f such that $\langle f, v_i \rangle < 1$. We thus have

$$\langle f, u \rangle = \sum_{j \neq i} x_j^* \langle f, v_j \rangle + x_i^* \langle f, v_i \rangle < \sum_j x_j^* = \langle \mathbf{1}, x^* \rangle. \quad (\text{C.3})$$

According to (ii), $\langle f, u \rangle$ is the objective value of a feasible solution with support on F . Hence, x^* could not have been optimal, and all optimal solutions have to have support on F . Clearly, all previous arguments still hold if we find a face $F' \subset F$ of P such that $u \in \text{cone}(F')$ and hence the statement follows. \square

D Acronyms

DAQC	digital-analog quantum computing	3
EASE	efficient, arbitrary, simultaneously entangling	2
LP	linear program	2
MAGIC	magnetic gradient-induced coupling	1
MIP	mixed integer program	2
MS	Mølmer-Sørensen	2
NISQ	noisy and intermediate-scale quantum	1
POVM	positive operator valued measure	12
QFT	quantum Fourier transform	2
QPU	quantum processing unit	25

References

- [1] J. Preskill, *Quantum computing in the NISQ era and beyond*, [Quantum](#) **2**, 79 (2018), [arXiv:1801.00862 \[quant-ph\]](#).

- [2] D. A. Lidar and T. A. Brun, *Quantum error correction* (Cambridge University Press, 2013).
- [3] X. Wang, A. Sørensen, and K. Mølmer, *Multibit gates for quantum computing*, *Phys. Rev. Lett.* **86**, 3907 (2001), [arXiv:quant-ph/0012055 \[quant-ph\]](#).
- [4] T. Monz, P. Schindler, J. T. Barreiro, M. Chwalla, D. Nigg, W. A. Coish, M. Harlander, W. Hänsel, M. Hennrich, and R. Blatt, *14-qubit entanglement: Creation and coherence*, *Phys. Rev. Lett.* **106**, 130506 (2011), [arXiv:1009.6126 \[quant-ph\]](#).
- [5] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, *Experimental comparison of two quantum computing architectures*, *PNAS* **114**, 3305 (2017), [arXiv:1702.01852 \[quant-ph\]](#).
- [6] E. A. Martinez, T. Monz, D. Nigg, P. Schindler, and R. Blatt, *Compiling quantum algorithms for architectures with multi-qubit gates*, *New J. Phys.* **18**, 063029 (2016), [arXiv:1601.06819 \[quant-ph\]](#).
- [7] D. Maslov and Y. Nam, *Use of global interactions in efficient quantum circuit constructions*, *New J. Phys.* **20**, 033018 (2018), [arXiv:1707.06356 \[quant-ph\]](#).
- [8] J. van de Wetering, *Constructing quantum circuits with global gates*, *New J. Phys.* **23**, 043015 (2021), [arXiv:2012.09061 \[quant-ph\]](#).
- [9] N. Grzesiak, A. Maksymov, P. Niroula, and Y. Nam, *Efficient quantum programming using EASE gates on a trapped-ion quantum computer*, *Quantum* **6**, 634 (2022), [arXiv:2107.07591 \[quant-ph\]](#).
- [10] S. Bravyi, D. Maslov, and Y. Nam, *Constant-cost implementations of Clifford operations and multiply controlled gates using global interactions*, *Phys. Rev. Lett.* **129**, 230501 (2022), [arXiv:2207.08691](#).
- [11] C. Figgatt, A. Ostrander, N. M. Linke, K. A. Landsman, D. Zhu, D. Maslov, and C. Monroe, *Parallel entangling operations on a universal ion-trap quantum computer*, *Nature* **572**, 368 (2019), [arXiv:1810.11948 \[quant-ph\]](#).
- [12] Y. Lu, S. Zhang, K. Zhang, W. Chen, Y. Shen, J. Zhang, J.-N. Zhang, and K. Kim, *Global entangling gates on arbitrary ion qubits*, *Nature* **572**, 363 (2019), [arXiv:1901.03508 \[quant-ph\]](#).
- [13] N. Grzesiak, R. Blümel, K. Wright, K. M. Beck, N. C. Pienti, M. Li, V. Chaplin, J. M. Amini, S. Debnath, J.-S. Chen, and Y. Nam, *Efficient arbitrary simultaneously entangling gates on a trapped-ion quantum computer*, *Nat. Commun.* **11**, 2963 (2020), [arXiv:1905.09294](#).
- [14] F. Mintert and C. Wunderlich, *Ion-trap quantum logic using long-wavelength radiation*, *Phys. Rev. Lett.* **87**, 257904 (2001), [arXiv:quant-ph/0104041 \[quant-ph\]](#).
- [15] C. Wunderlich, *Conditional spin resonance with trapped ions*, in *Laser Physics at the Limits*, edited by H. Figger, C. Zimmermann, and D. Meschede (Springer, 2002) pp. 261–273, [arXiv:quant-ph/0111158 \[quant-ph\]](#).
- [16] N. Timoney, I. Baumgart, M. Johanning, A. F. Varon, C. Wunderlich, M. B. Plenio, and A. Retzker, *Quantum Gates and Memory using Microwave Dressed States*, *Nature* **476**, 185 (2011), [arXiv:1105.1146 \[quant-ph\]](#).
- [17] C. Ospelkaus, U. Warring, Y. Colombe, K. R. Brown, J. M. Amini, D. Leibfried, and D. J. Wineland, *Microwave quantum logic gates for trapped ions*, *Nature* **476**, 181 (2011), [arXiv:1104.3573 \[quant-ph\]](#).
- [18] A. Khromova, C. Piltz, B. Scharfenberger, T. Gloger, M. Johanning, A. Varón, and C. Wunderlich, *Designer spin pseudomolecule implemented with trapped ions in a magnetic gradient*, *Phys. Rev. Lett.* **108**, 220502 (2012), [arXiv:1112.5302](#).
- [19] C. Piltz, T. Sriarunothai, S. S. Ivanov, S. Wölk, and C. Wunderlich, *Versatile microwave-driven trapped ion spin system for quantum information processing*, *Sci. Adv.* **2**, e1600093 (2016), [arXiv:1509.01478](#).
- [20] B. Lekitsch, S. Weidt, A. G. Fowler, K. Mølmer, S. J. Devitt, C. Wunderlich, and W. K. Hensinger, *Blueprint for a microwave trapped-ion quantum computer*, *Sci. Adv.* **3**, e1601540 (2017), [arXiv:1508.00420 \[quant-ph\]](#).
- [21] S. Wölk and C. Wunderlich, *Quantum dynamics of trapped ions in a dynamic field gradient using dressed states*, *New J. Phys.* **19**, 083021 (2017), [arXiv:1606.04821 \[quant-ph\]](#).
- [22] J. Cohn, M. Motta, and R. M. Parrish, *Quantum filter diagonalization with compressed double-factorized Hamiltonians*, *PRX Quantum* **2**, 040352 (2021), [arXiv:2104.08957 \[quant-ph\]](#).
- [23] A. Sørensen and K. Mølmer, *Spin-spin interaction and spin squeezing in an optical lattice*, *Phys. Rev. Lett.* **83**, 2274 (1999), [arXiv:quant-ph/9903044](#).

- [24] A. Sørensen and K. Mølmer, *Entanglement and quantum computation with ions in thermal motion*, *Phys. Rev. A* **62**, 022311 (2000), [arXiv:quant-ph/0002024](#).
- [25] T. Choi, S. Debnath, T. A. Manning, C. Figgatt, Z. X. Gong, L. M. Duan, and C. Monroe, *Optimal quantum control of multimode couplings between trapped ion qubits for scalable entanglement*, *Phys. Rev. Lett.* **112**, 190502 (2014), [arXiv:1401.1575 \[quant-ph\]](#).
- [26] A. Parra-Rodriguez, P. Lougovski, L. Lamata, E. Solano, and M. Sanz, *Digital-analog quantum computation*, *Phys. Rev. A* **101**, 022305 (2020), [arXiv:1812.03637 \[quant-ph\]](#).
- [27] D. Maslov and B. Zindorf, *Depth optimization of CZ, CNOT, and Clifford circuits*, *IEEE Transactions on Quantum Engineering* **3**, 1 (2022), [arXiv:2201.05215 \[quant-ph\]](#).
- [28] S. S. Ivanov, M. Johanning, and C. Wunderlich, *Simplified implementation of the quantum Fourier transform with Ising-type Hamiltonians: Example with ion traps*, [arXiv:1503.08806](#) (2015).
- [29] G. Breit and I. I. Rabi, *Measurement of nuclear spin*, *Phys. Rev.* **38**, 2082 (1931).
- [30] S. Bourdeauducq, whitequark, R. Jördens, D. Nadlinger, Y. Sionneau, and F. Kermarrec, *ARTIQ 10.5281/zenodo.6619071* (2021), Version 6.
- [31] I. Dumer, D. Micciancio, and M. Sudan, *Hardness of approximating the minimum distance of a linear code*, *IEEE Trans. Inf. Theory* **49**, 22 (2003).
- [32] N. Christofides, *Worst-case analysis of a new heuristic for the travelling salesman problem*, Tech. Rep. (Defense Technical Information Center, 1976).
- [33] H. Karloff, *Linear Programming* (Birkhäuser Boston, 1991) pp. 23–47.
- [34] M. Kliesch and I. Roth, *Theory of quantum system certification*, *PRX Quantum* **2**, 010201 (2021), tutorial, [arXiv:2010.05925 \[quant-ph\]](#).
- [35] N. B. Karahanoglu, H. Erdoğan, and Ş. İ. Birbil, *A mixed integer linear programming formulation for the sparse recovery problem in compressed sensing*, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013) pp. 5870–5874.
- [36] M. Johanning, A. F. Varón, and C. Wunderlich, *Quantum simulations with cold trapped ions*, *J. Phys. B* **42**, 154009 (2009), [arXiv:0905.0118 \[quant-ph\]](#).
- [37] P. Baßler and M. Zipper, *Source code for “Synthesis of and compilation with time-optimal multi-qubit gates”*, <https://github.com/matt-zipp/arXiv-2206.06387> (2022).
- [38] S. Diamond and S. Boyd, *CVXPY: A Python-embedded modeling language for convex optimization*, *J. Mach. Learn. Res.* **17**, 1 (2016).
- [39] A. Agrawal, R. Verschuere, S. Diamond, and S. Boyd, *A rewriting system for convex optimization problems*, *J. Control Decis.* **5**, 42 (2018), [arXiv:1709.04494](#).
- [40] Free Software Foundation, *GLPK (GNU Linear Programming Kit)* (2012), version: 0.4.6.
- [41] MOSEK ApS, *MOSEK Optimizer API for Python 9.3.14* (2022).
- [42] S. Kukita, H. Kiya, and Y. Kondo, *Short composite quantum gate robust against two common systematic errors*, *J. Phys. Soc. Japan* **91**, 104001 (2022), [arXiv:2112.12945 \[quant-ph\]](#).
- [43] D. A. Spielman and S.-H. Teng, *Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time*, *Journal of the ACM* **51**, 385 (2004), [arXiv:cs/0111050 \[cs\]](#).
- [44] S. Bravyi and D. Maslov, *Hadamard-free circuits expose the structure of the Clifford group*, *IEEE Trans. Inf. Theory* **67**, 4546 (2021), [arXiv:2003.09412 \[quant-ph\]](#).
- [45] S. Aaronson and D. Gottesman, *Improved simulation of stabilizer circuits*, *Phys. Rev. A* **70**, 052328 (2004), [arXiv:quant-ph/0406196 \[quant-ph\]](#).
- [46] D. Maslov and M. Roetteler, *Shorter stabilizer circuits via Bruhat decomposition and quantum circuit transformations*, *IEEE Trans. Inf. Theory* **64**, 4729 (2018), [arXiv:1705.09176 \[quant-ph\]](#).
- [47] R. Duncan, A. Kissinger, S. Perdrix, and J. van de Wetering, *Graph-theoretic simplification of quantum circuits with the ZX-calculus*, *Quantum* **4**, 279 (2020), [arXiv:1902.03178 \[quant-ph\]](#).
- [48] R. Kueng and D. Gross, *Qubit stabilizer states are complex projective 3-designs*, [arXiv:1510.02767 \[quant-ph\]](#).
- [49] H.-Y. Huang, R. Kueng, and J. Preskill, *Predicting many properties of a quantum system from very few measurements*, *Nat. Phys.* **16**, 1050 (2020), [arXiv:2002.08953 \[quant-ph\]](#).
- [50] D. Schlingemann, *Stabilizer codes can be realized as graph codes*, [arXiv:quant-ph/0111080 \[quant-ph\]](#) (2001).
- [51] D. Schlingemann and R. F. Werner, *Quantum error-correcting codes associated with graphs*, *Phys. Rev. A* **65**, 012308 (2001), [arXiv:quant-ph/0012111 \[quant-ph\]](#).

- [52] V. Kliuchnikov, K. Lauter, R. Minko, A. Paetznick, and C. Petit, *Shorter quantum circuits*, [arXiv:2203.10064 \[quant-ph\]](#) (2022).
- [53] M. Born and R. Oppenheimer, *Zur Quantentheorie der Molekeln*, *Ann. Phys.* **389**, 457 (1927).
- [54] J. Kempe, A. Kitaev, and O. Regev, *The complexity of the local Hamiltonian problem*, *SIAM J. Comput.* **35**, 1070 (2006), [arXiv:quant-ph/0406180 \[quant-ph\]](#).
- [55] R. M. Parrish and P. L. McMahon, *Quantum filter diagonalization: Quantum eigendecomposition without full quantum phase estimation*, [arXiv:1909.08925](#) (2019).
- [56] K. Klymko, C. Mejuto-Zaera, S. J. Cotton, F. Wudarski, M. Urbanek, D. Hait, M. Head-Gordon, K. B. Whaley, J. Moussa, N. Wiebe, W. A. de Jong, and N. M. Tubman, *Real time evolution for ultracompact Hamiltonian eigenstates on quantum hardware*, *PRX Quantum* **3**, 020323 (2022), [arXiv:2103.08563](#).
- [57] N. H. Stair, R. Huang, and F. A. Evangelista, *A multireference quantum Krylov algorithm for strongly correlated electrons*, *J. Chem. Theory Comput.* **16**, 2236 (2020), PMID: 32091895.
- [58] V. Kliuchnikov, D. Maslov, and M. Mosca, *Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and T gates*, *Quantum Inform. Comput.* **13**, 607 (2013), [arXiv:1206.5236 \[quant-ph\]](#).
- [59] N. J. Ross and P. Selinger, *Optimal ancilla-free Clifford+T approximation of z-rotations*, *Quantum Inform. Compu.* **16**, 901 (2016), [arXiv:1403.2975](#).
- [60] A. Bouland and T. Giurgica-Tiron, *Efficient universal quantum compilation: An inverse-free Solovay-Kitaev algorithm*, [arXiv:2112.02040 \[quant-ph\]](#).
- [61] M. Amy, P. Azimzadeh, and M. Mosca, *On the CNOT-complexity of CNOT-PHASE circuits*, *Quantum Sci. Technol.* **4**, 015002 (2018), [arXiv:1712.01859 \[quant-ph\]](#).
- [62] M. Amy, D. Maslov, and M. Mosca, *Polynomial-time T-depth optimization of Clifford+T circuits via matroid partitioning*, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **33**, 1476 (2014), [arXiv:1303.2042 \[quant-ph\]](#).
- [63] R. O’Donnell, *Analysis of Boolean Functions* (Cambridge University Press) [arXiv:2105.10386 \[cs.DM\]](#).
- [64] G. Dantzig, R. Fulkerson, and S. Johnson, *Solution of a large-scale traveling-salesman problem*, *Operations Research Society of America* **2**, 393 (1954).
- [65] P. García-Molina, A. Martin, and M. Sanz, *Noise in digital and digital-analog quantum computation*, [arXiv:2107.12969 \[quant-ph\]](#).
- [66] P. T. Fisk, M. J. Sellars, M. A. Lawn, and G. Coles, *Accurate measurement of the 12.6 GHz “clock” transition in trapped $^{171}\text{Yb}^+$ ions*, *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* **44**, 344 (1997).
- [67] S. F. D. Waldron, *An Introduction to Finite Tight Frames*, Applied and Numerical Harmonic Analysis (Springer New York, New York, NY, 2018).
- [68] R. B. Holmes and V. I. Paulsen, *Optimal frames for erasures*, *Linear Algebra Its Appl.* **377**, 31 (2004).

B Paper - Time-optimal multi-qubit gates: Complexity, efficient heuristic and gate-time bounds

Title: Time-optimal multi-qubit gates: Complexity, efficient heuristic and gate-time bounds

Authors: Pascal Baßler, Markus Heinrich, Martin Kliesch

Journal: Quantum

Publication status: Published

Contribution by PB: Main author (input approx 70%)

A summary of this publication is presented in section 4.2.

The research objective was devised by MK and me as a continuation of our previous work [32]. MK and I proofed the complexity results of the linear program approach. MH and I worked together trying to prove Conjecture 16 on the total gate time bounds, and we derived the proof for Theorem 15 together. I derived all results from section 5, 6 and 7.1 with the help from regular discussions with MH and MK. I developed the Python code and carried out all numerical simulations. I wrote an initial draft for the main text, which was then extended by my co-authors and finally rewritten and proofread in joint meetings by all authors.

Time-optimal multi-qubit gates: Complexity, efficient heuristic and gate-time bounds

P. Baßler¹, M. Heinrich¹, and M. Kliesch²

¹ Institute for Theoretical Physics, Heinrich Heine University Düsseldorf, Germany

² Institute for Quantum Inspired and Quantum Optimization, Hamburg University of Technology, Germany

Multi-qubit entangling interactions arise naturally in several quantum computing platforms and promise advantages over traditional two-qubit gates. In particular, a fixed multi-qubit Ising-type interaction together with single-qubit X-gates can be used to synthesize global ZZ-gates (GZZ gates). In this work, we first show that the synthesis of such quantum gates that are time-optimal is NP-hard. Second, we provide explicit constructions of special time-optimal multi-qubit gates. They have constant gate times and can be implemented with linearly many X-gate layers. Third, we develop a heuristic algorithm with polynomial runtime for synthesizing fast multi-qubit gates. Fourth, we derive lower and upper bounds on the optimal GZZ gate-time. Based on explicit constructions of GZZ gates and numerical studies, we conjecture that any GZZ gate can be executed in a time $O(n)$ for n qubits. Our heuristic synthesis algorithm leads to GZZ gate-times with a similar scaling, which is optimal in this sense. We expect that our efficient synthesis of fast multi-qubit gates allows for faster and, hence, also more error-robust execution of quantum algorithms.

1 Introduction

Any quantum computation requires to decompose its logical operations into the platform's native instruction set. The performance of the computation depends heavily on the available instructions and their implementation in the quantum hardware. In particular for early quantum devices a major challenge is posed by their short decoherence times, which limits the runtime of a quantum computation significantly. Therefore, it is not only necessary to optimize the number of native instructions but also their execution time.

Ising-type interactions give rise to an important and rich class of Hamiltonians ubiquitous in several quantum computing platforms [1–5]. Previously, we have utilized these Ising-type interactions in a new synthesis method [6]. In particular, we have considered the problem of synthesizing time-optimal multi-qubit gates on a quantum computing platform that supports the following basic operations:

- (I) single-qubit rotations can be executed in parallel, and
- (II) it offers a fixed Ising-type interaction.

The corresponding synthesis of global ZZ-gates (GZZ gates) is given by the minimization of the overall gate time, which can be written as a linear program (LP) [6]. This LP is exponentially large in the number of qubits. It has been unclear whether such multi-qubit gates can be synthesized in a computationally efficient way while keeping the gate time optimal.

In this work, we prove that this synthesis problem is NP-hard and provide a close-to-optimal efficient heuristic solution. To establish this hardness result, we draw the connection between the synthesis problem and graph theory. The *cut polytope* is defined as the convex hull of binary vectors representing the possible cuts of a given graph [7]. We provide a polynomial time reduction of the

P. Baßler: bassler@hhu.de

M. Kliesch: martin.kliesch@tuhh.de

membership problem of the cut polytope to the synthesis of time-optimal multi-qubit gates. Since this membership problem is NP-complete [8], the synthesis of time-optimal multi-qubit gates is NP-hard. This is akin to the NP-hardness of finding an optimal control pulse for multi-qubit gates using the Mølmer-Sørensen mechanism [4, 5].

We provide several ways to circumvent the hardness of time-optimal GZZ-gates. First, we provide explicit constructions of time-optimal multi-qubit gates realizing nearest-neighbor coupling under physically motivated assumptions. Such constructed nearest-neighbor multi-qubit gates exhibit a constant gate time and can be implemented with only linearly many single-qubit gate layers. We then use these ideas to define relaxations of the underlying linear program, leading to a hierarchy of polynomial-time algorithms for the synthesis of fast multi-qubit gates. By increasing the level in the hierarchy, this heuristic approach can be adapted to provide substantially better approximations to the optimal solution at the cost of higher polynomial runtime. For a small number of qubits, numerical experiments show that the so-obtained gate times are close to the optimal solution and come with significant runtime savings.

Among others, we prove bounds on the minimal multi-qubit gate time, and conjecture that it scales at most linear with the number of qubits. This claim is supported by a class of explicit constructions of time-optimal multi-qubit gates achieving the linear upper bound. Moreover, we provide numerical evidence that these explicit solutions in fact yield the longest gate time for a small number of qubits.

We expect our results to be useful for the implementation of time-optimal multi-qubit gates in noisy and intermediate scale quantum (NISQ) devices and beyond. The polynomial-time heuristic algorithm makes it possible to efficiently synthesize fast multi-qubit gates for a growing number of qubits. The here considered multi-qubit gates have been useful in a number of different applications, some of which we investigated in previous work [6]. Furthermore, they are the main building blocks for a class of Instantaneous Quantum Polynomial time (IQP) circuits which might be classically hard to simulate [9]. More recently, it was shown that quantum memory circuits and boolean functions can be implemented with a constant number of GZZ gates and additional ancilla qubits [10]. Moreover, there is an ancilla-free construction of multi-qubit Clifford circuits using at most 26 GZZ gates [11]. As noted in Ref. [11], there is also a shorter implementation for $n \leq 2^{12}$, requiring only $2(\log_2(n) + 1)$ GZZ gates. This implementation is based on a decomposition in Ref. [12] and the log-depth implementation of a CX circuit in formula (4) of Ref. [13].

This paper is structured as follows: We first give a brief introduction to the synthesis of time-optimal multi-qubit gates [6]. In Section 4 we proof that the time-optimal multi-qubit synthesis problem is NP-hard. However, in Section 5 we explicitly construct a certain class of time-optimal multi-qubit gates with constant gate time. The heuristic algorithm based on the ideas of the previous section is introduced in Section 6. Section 7 provides gate time bounds for time-optimal multi-qubit gates. Finally, Section 8 presents numerical evidence for the conjectured linear gate-time scaling and numerical benchmarks for the heuristic algorithm.

2 Synthesizing multi-qubit gates with Ising-type interactions

In this section, we give a short introduction to the synthesis of time-optimal multi-qubit gates. For more details we refer to the first two sections of Ref. [6].

On the abstract quantum computing platform with n qubits specified by the requirements (I) and (II) above, interactions between the qubits are generated by an Ising-type Hamiltonian

$$H_{ZZ} := - \sum_{i < j}^n J_{ij} \sigma_z^i \sigma_z^j, \quad (1)$$

where σ_z^i is the Pauli-Z operator acting on the i -th qubit. Note, that diagonal terms, where $i = j$, are excluded since they only change the Hamiltonian by an energy offset. By J we denote the symmetric matrix with entries J_{ij} in the upper triangular part and vanishing diagonal. We call J the *physical coupling matrix*.

Conjugating the Hamiltonian H_{ZZ} with X gates on the qubits indicated by the binary vector

$\mathbf{b} \in \mathbb{F}_2^n$ yields

$$\begin{aligned} \sigma_x^{\mathbf{b}} H_{ZZ} \sigma_x^{\mathbf{b}} &= - \sum_{i < j}^n J_{ij} \sigma_x^{b_i} \sigma_x^{b_j} \sigma_z^i \sigma_z^j \sigma_x^{b_i} \sigma_x^{b_j} \\ &= - \sum_{i < j}^n J_{ij} (-1)^{b_i} (-1)^{b_j} \sigma_z^i \sigma_z^j \\ &= - \sum_{i < j}^n J_{ij} m_i m_j \sigma_z^i \sigma_z^j =: H(\mathbf{m}), \end{aligned} \quad (2)$$

where we define the *encoding* $\mathbf{m} := (-1)^{\mathbf{b}}$ entry wise. The sign of the interaction between qubit i and j is given by $m_i m_j \in \{-1, +1\}$. We call $H(\mathbf{m})$ the *encoded Hamiltonian*.

Given time steps $\lambda_{\mathbf{m}} \geq 0$ during which the encoding \mathbf{m} is used, we consider unitaries of the form

$$\prod_{\mathbf{m}} e^{-i \lambda_{\mathbf{m}} H(\mathbf{m})} = e^{-i \sum_{\mathbf{m}} \lambda_{\mathbf{m}} H(\mathbf{m})} =: e^{-iH}, \quad (3)$$

where we used that the diagonal Hamiltonians $H(\mathbf{m})$ mutually commute. For all possible encodings $\mathbf{m} \in \{-1, +1\}^n$ we collect the time steps $\lambda_{\mathbf{m}}$ in a vector $\boldsymbol{\lambda} \in \mathbb{R}^{2^n}$ and interpret $t = \sum_{\mathbf{m}} \lambda_{\mathbf{m}}$ as the total gate time of the unitary e^{-iH} , implemented by the sequence of unitaries (3). Moreover, we use the symmetry

$$(-\mathbf{m})(-\mathbf{m})^T = \mathbf{m} \mathbf{m}^T \quad (4)$$

to reduce the degrees of freedom in $\boldsymbol{\lambda}$ from 2^n to 2^{n-1} by adding up $\lambda_{\mathbf{m}} + \lambda_{-\mathbf{m}}$ to a single time step.

The so generated unitary is the time evolution operator under the *total Hamiltonian*

$$H = - \sum_{i < j}^n A_{ij} \sigma_z^i \sigma_z^j, \quad (5)$$

where we have defined the *total coupling matrix*

$$A := J \circ \sum_{\mathbf{m}} \lambda_{\mathbf{m}} \mathbf{m} \mathbf{m}^T, \quad (6)$$

and used the linearity of the Hadamard (entry-wise) product \circ . By construction, A is a symmetric matrix with vanishing diagonal. Let us define the $\binom{n}{2}$ -dimensional subspace of symmetric matrices with vanishing diagonal by

$$\text{Sym}_0(\mathbb{R}^n) := \{M \in \text{Sym}(\mathbb{R}^n) \mid M_{ii} = 0 \ \forall i \in [n]\}. \quad (7)$$

For $A \in \text{Sym}_0(\mathbb{R}^n)$, we define an associated multi-qubit gate $\text{GZZ}(A)$, where **GZZ** stands for “global ZZ interactions”,

$$\text{GZZ}(A) := \exp \left(i \sum_{i < j}^n A_{ij} \sigma_z^i \sigma_z^j \right). \quad (8)$$

To determine which matrices can be decomposed as in Eq. (6), we denote the non-zero index set of a symmetric matrix as $\text{nz}(M) := \{(i, j) \mid M_{ij} \neq 0, i < j\}$. Then, the subspace of matrices $A \in \text{Sym}_0(\mathbb{R}^n)$ that can be decomposed as in Eq. (6) is exactly given by the condition $\text{nz}(A) \subseteq \text{nz}(J)$, which we assume to hold from now on. Thus, all-to-all connectivity enables to decompose any coupling matrix $A \in \text{Sym}_0(\mathbb{R}^n)$ but is not strictly required by our approach. We call the number of encodings \mathbf{m} needed for the decomposition the *encoding cost* of $\text{GZZ}(A)$, and $\sum_{\mathbf{m}} \lambda_{\mathbf{m}}$ the *total GZZ time*. Note, that both quantities depend on the chosen decomposition.

It is convenient to abstract the following analysis from the physical details given by J . For a matrix $A \in \text{Sym}_0(\mathbb{R}^n)$ with $\text{nz}(A) \subseteq \text{nz}(J)$, its possible decompositions are in one-to-one correspondence with the decompositions of the matrix $M := A \oslash J \in \text{Sym}_0(\mathbb{R}^n)$ where

$$(A \oslash J)_{ij} := \begin{cases} A_{ij}/J_{ij}, & i \neq j \text{ and } J_{ij} \neq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

We further define the linear operator

$$\begin{aligned} \mathcal{V} : \mathbb{R}_{\geq 0}^{2^{n-1}} &\rightarrow \text{Sym}_0(\mathbb{R}^n), \\ \boldsymbol{\lambda} &\mapsto \sum_{\mathbf{m}} \lambda_{\mathbf{m}} \mathbf{m} \mathbf{m}^T, \end{aligned} \quad (10)$$

represented in the standard basis by a matrix

$$V \in \{-1, +1\}^{\binom{n}{2} \times (2^{n-1})}. \quad (11)$$

Let $\mathbf{v} : \text{Sym}_0(\mathbb{R}^n) \rightarrow \mathbb{R}^{\binom{n}{2}}$ be the (row-wise) vectorization of the upper triangular part of the matrix input such that the columns of V are given by $\mathbf{v}(\mathbf{m} \mathbf{m}^T)$. Our objective is to minimize the total gate time and the amount of \mathbf{m} 's needed to express the matrix $M \in \text{Sym}_0(\mathbb{R}^n)$. To this end we formulate the following linear program (LP):

$$\begin{aligned} &\text{minimize} && \mathbf{1}^T \boldsymbol{\lambda} \\ &\text{subject to} && V \boldsymbol{\lambda} = \mathbf{v}(M), \\ &&& \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^{2^{n-1}}, \end{aligned} \quad (12)$$

where $\mathbf{1} = (1, 1, \dots, 1)$ is the all-ones vector such that $\mathbf{1}^T \boldsymbol{\lambda} = \sum_{\mathbf{m}} \lambda_{\mathbf{m}}$. A *feasible solution* is an assignment of the variables that fulfills all constraints of the optimization problem and an *optimal solution* is a feasible solution which also minimizes/maximizes the objective function. Throughout this paper, we indicate optimal solutions by an asterisk, e.g. $\boldsymbol{\lambda}^*$. Note, that the LP (12) has a feasible solution for any symmetric matrix M with vanishing diagonal [6, Theorem II.2]. The theory of linear programming then guarantees the existence of an optimal solution with at most $\binom{n}{2}$ non-zero entries [6, Proposition II.3].

A standard tool in convex optimization is *duality* [14] which will be used in Section 7. The dual LP to the LP (12) reads as follows:

$$\begin{aligned} &\text{maximize} && \langle M, \mathbf{y} \rangle \\ &\text{subject to} && V^T \mathbf{y} \leq \mathbf{1}, \\ &&& \mathbf{y} \in \mathbb{R}^{\binom{n}{2}}, \end{aligned} \quad (13)$$

with the inner product $\langle \cdot, \cdot \rangle : \text{Sym}_0(\mathbb{R}^n) \times \mathbb{R}^{\binom{n}{2}} \rightarrow \mathbb{R}$, $\langle M, \mathbf{y} \rangle \mapsto \mathbf{v}(M)^T \mathbf{y}$. Here, inequalities between vectors are to be understood entry-wise. A simple, but important fact is the following: If $\boldsymbol{\lambda}^*$ is an optimal solution to the primal LP (12), then any feasible solution \mathbf{y} to the dual LP (13) provides a *lower bound* as $\langle M, \mathbf{y} \rangle \leq \mathbf{1}^T \boldsymbol{\lambda}^*$. Moreover, the feasibility of the LP (12) implies that we have *strong duality*: if \mathbf{y}^* is a dual optimal solution, then we have equality between the optimal values, $\langle M, \mathbf{y}^* \rangle = \mathbf{1}^T \boldsymbol{\lambda}^*$.

3 Main results

We want to highlight our main contributions. First, we provide the hardness results for the synthesis of time-optimal GZZ gates.

Theorem (Theorem 6). *The decision version of the LP (12), is NP-complete.*

We say that the synthesis of time-optimal multi-qubit gates (solving LP (12)) is NP-hard in the sense of the function problem extension of the decision problem class NP [15]. We circumvent the hardness of the time-optimal multi-qubit gate synthesis by providing an explicit construction of GZZ gates realizing next neighbor coupling with constant gate time and linear encoding cost. The assumption of a constant subdiagonal of J is physically motivated and can be realized in an ion trap [16].

Theorem (Theorem 10, informal). *Let the subdiagonal of J and A be a constant with values c and φ respectively. Then $\text{GZZ}(A)$ on n qubits has the encoding cost of $d \leq 2n$ and constant total GZZ time $2\varphi/c$. This total gate time is optimal.*

In Section 6 we define Algorithm 3 and introduce LP (41), a polynomial time heuristic to synthesize GZZ gates with small, but not necessarily minimal, gate time. This heuristic does not rely on any further assumptions and is applicable for arbitrary $J, A \in \text{Sym}_0(\mathbb{R}^n)$. In Section 8 we show numerically that this heuristic leads to GZZ gate times close to the optimum while solving the synthesis problem much faster.

Furthermore, we proof lower and upper bounds on the GZZ gate time.

Theorem (Theorem 15). *The optimal total gate time of $\text{GZZ}(A)$ with $A \in \text{Sym}_0(\mathbb{R}^n)$ is lower and upper bounded by*

$$\|A \odot J\|_{\ell_\infty} \leq \mathbf{1}^T \boldsymbol{\lambda}^* \leq \|A \odot J\|_{\ell_1}. \quad (14)$$

Here, the upper bound scales quadratic in n for a constant matrix $M = A \odot J$. This upper bound is loose in the sense that it also holds for the GZZ gates constructed by the heuristic in Section 6. Therefore, we tighten the upper bound to a linear scaling in n .

Conjecture (Conjecture 16). *The optimal gate time of $\text{GZZ}(A)$ with $A \in \text{Sym}_0(\mathbb{R}^n)$ is tightly upper bounded by*

$$\mathbf{1}^T \boldsymbol{\lambda}^* \leq \|A \odot J\|_{\ell_\infty} \cdot \begin{cases} n, & \text{for odd } n, \\ n-1, & \text{for even } n. \end{cases} \quad (15)$$

We provide evidence for this conjecture using an explicit construction (Theorem 21) that realizes the conjectured upper bound for any n , as well as numerical evidence for $n \leq 8$ (Fig. 1). Unfortunately, we were not able to prove this result and state the challenges in Appendix A.

4 Synthesizing time-optimal GZZ gates is NP-hard

In this section, we investigate the complexity of solving the gate synthesis problem stated as LP (12). We observe that LP (12) is an optimization over the convex cone generated by

$$\mathcal{E}_n := \{\mathbf{m}\mathbf{m}^T \mid \mathbf{m} \in \{-1, +1\}^n, m_n = +1\}, \quad (16)$$

which is the set of outer products generated by all possible encodings \mathbf{m} . Due to the symmetry Eq. (4) we can uniformly fix the value of one entry of \mathbf{m} . We chose the convention $m_n = +1$. In the literature \mathcal{E}_n is also known as the *elliptope* of rank one matrices [17]. In the following, we consider the polytope

$$\text{conv}(\mathcal{E}_n) := \left\{ \sum_i \lambda_i \mathbf{r}_i \mid \sum_i \lambda_i = 1, \lambda_i \geq 0, \mathbf{r}_i \in \mathcal{E}_n \right\}, \quad (17)$$

and show the connection to graph theory, in particular to the cuts of graphs.

Definition 1 (cut polytope [7]). *Let $K_n = (V_n, E_n)$ be a complete graph with n vertices. Denote $\delta(X)$ the set of all edges with one endpoint in $X \subset V_n$ and the other endpoint in its complement \bar{X} , i.e., $\delta(X)$ defines the cut between X and \bar{X} . Let $\chi^{\delta(X)} \in \{0, 1\}^{|E_n|}$ denote the characteristic vector of a cut, with $\chi_e^{\delta(X)} = 1$ if $e \in \delta(X)$ and $\chi_e^{\delta(X)} = 0$ otherwise. We define the cut polytope as the convex hull of the characteristic vectors*

$$\text{CUT}_n := \text{conv}\left\{ \chi^{\delta(X)} \in \{0, 1\}^{|E_n|} \mid X \subseteq V_n \right\}. \quad (18)$$

Lemma 2. *For all n , CUT_n is isomorphic to $\text{conv}(\mathcal{E}_n)$.*

Proof. For each $X \subset V_n$ we set $m_i = +1$ if $i \in X$ and $m_i = -1$ if $i \in \bar{X}$. Note, that there are 2^{n-1} different pairs of X and \bar{X} . We then have $m_i m_j = -1$ if $i \in X$ and $j \in \bar{X}$ (or the other way around), and $m_i m_j = +1$ if $i, j \in X$ or $i, j \in \bar{X}$. So the characteristic vector can be written as $\chi_e^{\delta(X)} = (1 - m_i m_j)/2$ for each edge $e \in E_n$ connecting vertices i and j . This is clearly a bijective affine map between the vertices and thus CUT_n is isomorphic to $\text{conv}(\mathcal{E}_n)$. \square

The following decision problems are membership problems, where the task is to decide if a given element \mathbf{x} belongs to a set or not. In our case \mathbf{x} is a vector and the set is a polytope.

Problem 3 (CUT_n membership).

Instance *The adjacency matrix $M \in \text{Sym}_0(\mathbb{Q}_{\geq 0}^n)$ of a weighted undirected graph with non-negative weights.*

Question *Is $M \in \text{CUT}_n$?*

Problem 4 ($\text{conv}(\mathcal{E}_n)$ membership).

Instance *The matrix $M \in \text{Sym}_0(\mathbb{Q}^n)$.*

Question *Is $M \in \text{conv}(\mathcal{E}_n)$? That is, does there exist a decomposition $M = \sum_i \lambda_i \mathbf{r}_i$ with $\mathbf{r}_i \in \mathcal{E}_n$ such that $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$?*

It is well known that the membership problem of the cut polytope, Problem 3, and Problem 4 are NP-complete [8, 18, 19]. Next, we state the decision version of our gate synthesis optimization.

Problem 5 (time-optimal multi-qubit gate synthesis).

Instance *The matrix $M = A \otimes J \in \text{Sym}_0(\mathbb{Q}^n)$ and a constant $K \in \mathbb{Q}_{\geq 0}$.*

Question *Is there a decomposition $M = \sum_i \lambda_i \mathbf{r}_i$ with $\mathbf{r}_i \in \mathcal{E}_n$ such that $\lambda_i \geq 0$ and $\sum_i \lambda_i \leq K$?*

Theorem 6. *Problem 5, which is the decision version of the LP (12), is NP-complete.*

Proof. A solution of Problem 5 can be verified in polynomial time since there always exists a decomposition $\sum_i \lambda_i \mathbf{r}_i$ of M with minimal $\sum_i \lambda_i$ which has at most $\binom{n}{2} = n/2(n-1)$ non-zero terms [6, Proposition II.3]. Therefore, Problem 5 is NP.

To show that Problem 5 is NP-hard, we construct a polynomial-time mapping reduction from Problem 4 to Problem 5. Given the matrix $M \in \text{Sym}_0(\mathbb{Q}^n)$ and a constant $K \in \mathbb{Q}_{\geq 0}$ as an instance for Problem 5, let $\lambda_i \geq 0$ be the positive coefficients of the decomposition. If we find $\sum_i \lambda_i < K$, then we can always add additional λ 's such that equality holds. We choose the additional λ 's as the coefficients of the decomposition for the all-zero matrix, see e.g. Lemma 8 below with $\mathbf{k} = \mathbf{1}_n$ for an explicit construction. We define the matrix $M' := M/K$ and the positive coefficients $\lambda'_i := \lambda_i/K$. Then $M' = \sum_i \lambda'_i \mathbf{r}_i$ with $\mathbf{r}_i \in \mathcal{E}_n$ and $\sum_i \lambda'_i = 1$. \square

5 Time-optimal GZZ synthesis for special instances

Although, solving the LP (12) is NP-hard we present explicit optimal solutions for certain families of instances which is equivalent to constructing time-optimal GZZ gates. The constructions of this section yield a qubit-independent total GZZ time which satisfies the optimal lower bound of Lemma 7 below. Moreover, we show that some GZZ gates can be synthesized with an encoding cost independent of the number of qubits. However, most of these constructions assume constant values for the elements of the band-diagonal of the physical coupling matrix J . These assumptions are relaxed throughout this section providing explicit optimal solutions for physically relevant cases. These results build the foundation for the heuristic algorithm for fast GZZ gate synthesis in the next section.

By $\|M\|_{\ell_p}$ we denote the ℓ_p -norm of a symmetric matrix M , which is given as the ℓ_p -norm of a vector $\mathbf{v}(M)$ containing all lower/upper triangular matrix elements in some order. First, we proof a lower bound on the optimal total GZZ time which can be used to verify time-optimality.

Lemma 7. *For any $M \in \text{Sym}_0(\mathbb{R}^n)$ the optimal objective function value of the LP (12) is lower bounded by*

$$\|M\|_{\ell_\infty} \leq \mathbf{1}^T \boldsymbol{\lambda}^*. \quad (19)$$

Proof. The lower bound can be verified by the fact that the matrix representation V of the linear operator in Eq. (10) only has entries ± 1 and that $\boldsymbol{\lambda}^*$ is non-negative. Thus, it holds that $\|M\|_{\ell_\infty} = \|V\boldsymbol{\lambda}^*\|_{\ell_\infty} \leq \mathbf{1}^T \boldsymbol{\lambda}^*$. \square

Next, we provide calculation rules for the coupling matrix $A \in \text{Sym}_0(\mathbb{R}^n)$ of the GZZ(A) gate. These rules are inherited from matrix exponentials. Let $A_1, A_2, A_3 \in \text{Sym}_0(\mathbb{R}^n)$ then

- (i) $\text{GZZ}(A_1 + A_2) = \text{GZZ}(A_1) \text{GZZ}(A_2)$,
- (ii) $\text{GZZ}(A_1 \oplus A_2) = \text{GZZ}(A_1) \otimes \text{GZZ}(A_2)$ and
- (iii) $\text{GZZ}(A_1 \circ A_2) = \text{GZZ}(A_3)$, where the coupling matrices can be decomposed as

$$A_1 := \sum_{k=1}^{d_1} \lambda_{\mathbf{m}_k} \mathbf{m}_k \mathbf{m}_k^T, \quad A_2 := \sum_{k=1}^{d_2} \beta_{\mathbf{v}_k} \mathbf{v}_k \mathbf{v}_k^T, \quad A_3 := \sum_{k=1}^{d_3} \gamma_{\mathbf{w}_k} \mathbf{w}_k \mathbf{w}_k^T,$$

with $d_3 = d_1 d_2$, $\mathbf{w}_k = (\mathbf{m}_i \circ \mathbf{v}_j)_{k=d_2 i+j}$ and $\gamma_{\mathbf{w}_k} = (\lambda_{\mathbf{m}_i} \beta_{\mathbf{v}_j})_{k=d_2 i+j}$ for $j = 1, \dots, d_2$ and $i = 0, \dots, d_1 - 1$.

In Item (iii) we can also express $\gamma = \lambda \otimes \beta$ with the Kronecker product. Note, that these generated GZZ gates are not necessarily time-optimal.

We denote the $k \times k$ identity matrix by $\mathbf{1}_k$, the k dimensional all-ones vector by $\mathbf{1}_k$ and the $k \times k$ matrix of ones with vanishing diagonal by $E_k := \mathbf{1}_k \mathbf{1}_k^T - \mathbf{1}_k$. With the next two lemmas, we bound the encoding cost and total gate time for special cases of Item (i) and Item (ii), where the total gate time is constant. These results will provide a basis for all other constructions. The following result constructs total coupling matrices $A \in \text{Sym}_0(\mathbb{R}^n)$ on arbitrary subsets of qubits.

Lemma 8. *Let the coupling matrix J be constant, i.e. $J_{ij} = c \in \mathbb{R}_{\geq 0}$ for all $i, j \in [n]$ with $i \neq j$. For any $\varphi \in \mathbb{R}$ and $s \in \mathbb{Z}_{\geq 0}$ the GZZ(A) with the matrix*

$$A = \varphi \bigoplus_{i=1}^s E_{k_i}, \quad (20)$$

with $k_i \geq 1$ for $i = 1, \dots, s$ has the encoding cost of $d = 2^{\lceil \log_2(s) \rceil} \leq 2s$ and constant total GZZ time φ/c . This total gate time is optimal.

Proof. W.l.o.g. we set $c = \varphi = 1$. We have $n := \sum_{i=1}^s k_i$ qubits. Note, that $E_{k_i} = E_1 = 0$ only contributes to an entry in the diagonal of A , and hence this qubit does not participate in the GZZ(A) gate. We denote the $d \times d$ Hadamard matrix by $H^{d \times d}$ and the matrix consisting of its first s columns by $H^{d \times s}$ where $s \leq n$. The orthogonality of the columns of any Hadamard matrix yields $(H^{d \times s})^T H^{d \times s} = d \mathbf{1}_s$. Replacing each i -th column by k_i copies of it we obtain the $d \times n$ -matrix $H_{\mathbf{k}}^{d \times n}$ from $H^{d \times s}$. Then, we attain the diagonal block matrix structure

$$(H_{\mathbf{k}}^{d \times n})^T H_{\mathbf{k}}^{d \times n} - d \mathbf{1}_n = d \bigoplus_{i=1}^s E_{k_i}. \quad (21)$$

We set the diagonal elements on the left-hand side to zero, since they only contribute to an energy offset in the Hamiltonian. Take each row of $H_{\mathbf{k}}^{d \times n}$ as a possible vector $\mathbf{m} \in \{-1, 1\}^n$ to construct the total coupling matrix

$$A = \frac{1}{d} \sum_{\mathbf{m} \in \text{rows}(H_{\mathbf{k}}^{d \times n})} \mathbf{m} \mathbf{m}^T, \quad (22)$$

i.e., the time steps have been chosen as $\lambda_{\mathbf{m}} = \frac{1}{d} [\mathbf{m} \in \text{rows}(H_{\mathbf{k}}^{d \times n})]$, with the Iverson bracket $[\cdot]$. Clearly, we have $\sum_{\mathbf{m}} \lambda_{\mathbf{m}} = 1$. If we take the constants c and φ into account, we just multiply Eq. (22) by φ/c which gives a total GZZ time of φ/c . Furthermore, this total GZZ time is optimal since $\|M\|_{\ell_{\infty}} = \|A \odot J\|_{\ell_{\infty}} = \varphi/c$ satisfies the lower bound of Lemma 7. \square

The encoding cost of GZZ(A) considered in this lemma can be reduced if redundant encodings $\mathbf{m} = \mathbf{m}' \in \{-1, 1\}^n$ are present by adding the corresponding time steps $\lambda_{\mathbf{m}} + \lambda_{\mathbf{m}'}$. It can be further reduced by using the Hadamard conjecture [20, 21]. If the Hadamard conjecture holds, then there exists Hadamard matrices of any dimension divisible by 4. It is known that the Hadamard conjecture holds for dimensions $s \leq 668$ [22, 23], thus the encoding cost can be reduced to $d = 4 \lceil s/4 \rceil \leq s - 4$ in this regime. The encoding cost of the following Lemma 9, Theorems 10 and 11 and the efficient heuristic in Section 6 can be reduced in the same fashion.

The assumption in Lemma 8 of a constant coupling matrix, $J_{i \neq j} = c$, is physically unreasonable. Therefore, we relax this assumption for block sizes $k_i = 2$ which corresponds to pairwise next neighbor couplings. We use Lemma 8 to construct time-optimal GZZ gates for a certain family of coupling matrices.

Lemma 9. *Let n be even and the coupling matrix J be constant on the first subdiagonal (the other elements are arbitrary), i.e. $J_{ij} = c$ for $i \in [n-1]$ and $j = i \pm 1$. Then*

$$\text{GZZ}(A) = \bigotimes_{i=1}^{n/2} \text{GZZ}(\varphi E_2), \quad (23)$$

has the encoding cost of $d = 2^{\lceil \log_2(n) - 1 \rceil} < n$ and constant total GZZ time φ/c . This total gate time is optimal.

Proof. Since the matrix E_2 only has non-zero entries in the first subdiagonal, the coupling matrix J only needs to be constant there. The claim follows immediately from Lemma 8 by setting $k_1 = \dots = k_{n/2} = 2$. Clearly, the total GZZ time is optimal since it saturates the lower bound of Lemma 7. \square

Lemma 9 guarantees an encoding cost of $d < n$, which is a quadratic saving compared to the general LP solution with encoding cost $\binom{n}{2}$. We note, that

$$A = \varphi \bigoplus_{i=1}^{n/2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (24)$$

corresponds to parallel $\text{ZZ}(\varphi)$ gates, which find applications in simulating molecular dynamics [24]. The assumption of a constant subdiagonal of J can be realized in an ion trap platform by applying an anharmonic trapping potential [16].

By combining (i), Lemma 8 and Lemma 9 we obtain the GZZ gate for next neighbor coupling.

Theorem 10. *Let the subdiagonal of J be constant, i.e. $J_{ij} = c$ for $i \in [n-1]$ and $j = i \pm 1$. Then $\text{GZZ}(A)$ on n qubits with*

$$A = \varphi \begin{pmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & 1 & 0 & 1 & \\ & & 1 & 0 & 1 \\ & & & \ddots & \ddots \end{pmatrix}, \quad (25)$$

has the encoding cost of $d \leq 2n$ (for $n > 4$) and constant total GZZ time $2\varphi/c$. This total gate time is optimal.

Proof. We set $c = \varphi = 1$ w.l.o.g. For now, we assume that n is even. Then

$$\begin{aligned} A &= \bigoplus_{i=1}^{n/2} E_2 + E_1 \bigoplus \left(\bigoplus_{i=1}^{n/2-1} E_2 \right) \bigoplus E_1 \\ &= \begin{pmatrix} 0 & 1 & & & \\ 1 & 0 & & & \\ & & 0 & 1 & \\ & & 1 & 0 & \\ & & & \ddots & \ddots \end{pmatrix} + \begin{pmatrix} 0 & & & & \\ & 0 & 1 & & \\ & 1 & 0 & & \\ & & & 0 & 1 \\ & & & 1 & 0 \\ & & & & \ddots \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & 1 & 0 & 1 & \\ & & 1 & 0 & 1 \\ & & & \ddots & \ddots \end{pmatrix} \end{aligned} \quad (26)$$

Again, the diagonal entries do not contribute to the interactions. The first term can be implemented, using Lemma 9, with the encoding cost $d_1 = 2^{\lceil \log_2(n) - 1 \rceil}$ and the GZZ time of 1. The

second term can be implemented, using Lemma 8, as

$$\bigotimes_{i=1}^s \text{GZZ}(\varphi E_{k_i}), \quad (27)$$

where $s = n/2 + 1$, $k_1 = k_s = 1$ and $k_i = 2$ for $i = 2, \dots, n/2$, with the encoding cost $d_2 = 2^{\lceil \log_2(n+2) - 1 \rceil}$ and the GZZ time of 1. Adding the encoding costs $d = d_1 + d_2$ and GZZ times of both terms yields the desired result. If n is not even, then repeat the previous steps for $n + 1$ but in the end reduce the dimension of all the resulting \mathbf{m} by discarding the last entry.

This construction corresponds to a feasible solution of the primal LP (12) with the objective function value 2. To show optimality it suffices to construct a feasible solution for the dual LP (13) with the objective function value of 2. First, consider the case $n = 3$ with the total coupling matrix

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \Rightarrow \mathbf{v}(A)^T = (1, 0, 1), \quad (28)$$

and the matrix representation of the linear operator

$$V^T = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix}, \quad (29)$$

as in Eq. (10). A feasible dual solution satisfying $V^T \mathbf{y} \leq \mathbf{1}$ is $\mathbf{y} = (1, -1, 1)^T$. Thus, we can verify optimality for $n = 3$ since the objective function value is $\langle A, \mathbf{y} \rangle = \mathbf{v}(A)^T \mathbf{y} = 2$. Now, we consider arbitrary $n > 3$. Extending the dual solution for the case $n = 3$ with zeros $\mathbf{y} = (1, -1, 1, 0, \dots, 0)^T$ does not change the objective function value $\langle A, \mathbf{y} \rangle = \mathbf{v}(A)^T \mathbf{y} = 2$. Such an extended dual solution is still feasible since V^T restricted to the first three columns only has rows which are already contained in Eq. (29) due to the symmetry of Eq. (4). \square

Algorithm 1 is the pseudocode implementation of Theorem 10. It takes the number of qubits n , the constant value c of the subdiagonal of J and the factor φ of A as input and returns the sparse vector $\boldsymbol{\lambda}$, containing the time steps. The encodings \mathbf{m} are given by the indices of the non-zero elements $\lambda_{\mathbf{m}} \neq 0$.

Algorithm 1 Synthesize GZZ(A) as in Theorem 10.

Input: n, c, φ

Initialize $\boldsymbol{\lambda} = \mathbf{0} \in \mathbb{R}^{2^n}$

is_odd $\leftarrow false$

if n odd **then**

$n \leftarrow n + 1$

 is_odd $\leftarrow true$

Let $H_1^{d_1 \times d_1}$ be a Hadamard matrix

▷ With $d_1 = 2^{\lceil \log_2(n) - 1 \rceil}$ as in Lemma 9

$H_1^{d_1 \times \frac{n}{2}} \leftarrow \frac{n}{2}$ columns of $H_1^{d_1 \times d_1}$

▷ It does not matter which columns

$H_1^{d_1 \times n} \leftarrow$ duplicate columns $i = 1, \dots, \frac{n}{2}$ of $H_1^{d_1 \times \frac{n}{2}}$

Let $H_2^{d_2 \times d_2}$ be a Hadamard matrix

▷ With $d_2 = 2^{\lceil \log_2(n+2) - 1 \rceil}$

$H_2^{d_2 \times (\frac{n}{2} + 1)} \leftarrow \frac{n}{2} + 1$ columns of $H_2^{d_2 \times d_2}$

▷ It does not matter which columns

$H_2^{d_2 \times n} \leftarrow$ duplicate columns $i = 2, \dots, \frac{n}{2}$ of $H_2^{d_2 \times (\frac{n}{2} + 1)}$

for $j \in \{1, 2\}$ **do**

if is_odd **then**

 Delete one column of $H_j^{d_j \times n}$

▷ It does not matter which column

for $\mathbf{m} \in \text{rows}(H_j^{d_j \times n})$ **do**

$\lambda_{\mathbf{m}} \leftarrow \frac{\varphi}{d_j c}$

Output: $\boldsymbol{\lambda}$

▷ Can efficiently be saved in a sparse data format

The following theorem does not require any additional assumptions on J . It shows, how the LP (12) can be supplemented with the explicit solution to exclude certain qubits.

Theorem 11 (Excluding qubits). *Let N be the total number of qubits on the quantum hardware and $n = N - s$ be the participating qubits in the GZZ gate. Synthesize the $\text{GZZ}(A)$ gate with $A \in \text{Sym}_0(\mathbb{R}^n)$, using the LP (12). Then, the total encoding cost (on N qubits) is at most $\binom{n}{2} 2^{\lceil \log_2(s+1) \rceil}$ and the total GZZ time, $\mathbf{1}^T \boldsymbol{\lambda}^*$, (on N qubits) is the same as for the LP run on n qubits.*

Proof. Assume w.l.o.g. that all qubits to be excluded are at the end of the qubit array. Let $k_1 = n$ and $k_2, \dots, k_{s+1} = 1$. Using Lemma 8 we obtain an encoding cost of $d_1 = 2^{\lceil \log_2(s+1) \rceil}$ and total GZZ time 1 to generate the matrix

$$A_1 = \begin{pmatrix} E_n & 0 \\ 0 & 0 \end{pmatrix}. \quad (30)$$

Solving the LP (12) for a matrix $A \in \text{Sym}_0(\mathbb{R}^n)$ yields the encoding cost $d_2 = \binom{n}{2}$ and the total GZZ time $\mathbf{1}^T \boldsymbol{\lambda}^*$. We define the extension of $A \in \text{Sym}_0(\mathbb{R}^n)$ by $A_2 \in \text{Sym}_0(\mathbb{R}^N)$. The extension can be done by appending s arbitrary elements in $\{-1, +1\}$ to all vectors $\mathbf{m} \in \{-1, +1\}^n$ given by the LP (12). Clearly,

$$A_1 \circ A_2 = \begin{pmatrix} E_n & 0 \\ 0 & 0 \end{pmatrix} \circ \begin{pmatrix} A & * \\ * & * \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix}. \quad (31)$$

By (iii), the total encoding cost is $d_1 d_2$ and the total GZZ time is

$$\sum_{i=1}^{d_1} \sum_{j=1}^{d_2} 1/d_1 \lambda_j^* = \mathbf{1}^T \boldsymbol{\lambda}^*. \quad (32)$$

Consider now an arbitrary coupling matrix J . Then

$$J \circ A_1 \circ A_2 = J \circ \begin{pmatrix} E_n & 0 \\ 0 & 0 \end{pmatrix} \circ \begin{pmatrix} \tilde{A} & * \\ * & * \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix}, \quad (33)$$

where $\tilde{A} = A \odot J$ is decomposed by the LP (12). \square

Algorithm 2 takes the total number of qubits N , the total coupling matrix A (on n qubits) and the set $\mathcal{Z} := \{i \in [N] \mid \text{exclude qubit } i\}$ as input and returns the sparse vector $\boldsymbol{\gamma}$, containing the time steps. The encodings \mathbf{w} are given by the indices of the non-zero elements $\gamma_{\mathbf{w}} \neq 0$.

Algorithm 2 Excluding qubits.

Input: N, A, \mathcal{Z} (set of qubit indices to be excluded)

Initialize $\boldsymbol{\gamma} = \mathbf{0} \in \mathbb{R}^{2^N}$

$s \leftarrow |\mathcal{Z}|$

$\bar{\mathcal{Z}} \leftarrow \{i \in [N] \mid i \notin \mathcal{Z}\}$

Let $H_1^{d_1 \times d_1}$ be a Hadamard matrix

\triangleright With $d_1 = 2^{\lceil \log_2(s+1) \rceil}$

$H_1^{d_1 \times (s+1)} \leftarrow s+1$ columns of $H_1^{d_1 \times d_1}$

\triangleright It does not matter which columns

$H_1^{d_1 \times N} \leftarrow$ duplicate one column of $H_1^{d_1 \times (s+1)}$ $n-1$ times and place them at indices $i \in \bar{\mathcal{Z}}$

$\boldsymbol{\lambda}^* \leftarrow$ Solve $A = \sum_{\mathbf{m}} \lambda_{\mathbf{m}} \mathbf{m} \mathbf{m}^T$ \triangleright using LP (12)

for $\mathbf{v} \in \text{rows}(H_1^{d_1 \times N})$ **do**

for $\mathbf{m} \in \{\mathbf{m} \mid \lambda_{\mathbf{m}}^* \neq 0\}$ **do**

\triangleright There are at most $\binom{n}{2}$ such \mathbf{m}

$\tilde{\mathbf{m}} \leftarrow$ extend \mathbf{m} with arbitrary elements from $\{-1, +1\}$ at indices $i \in \mathcal{Z}$.

$\mathbf{w} \leftarrow \mathbf{v} \circ \tilde{\mathbf{m}}$

$\gamma_{\mathbf{w}} \leftarrow \frac{\lambda_{\mathbf{m}}^*}{d_1}$

Output: $\boldsymbol{\gamma}$

\triangleright Can efficiently be saved in a sparse data format

We showed explicit constructions of time-optimal GZZ gates for total coupling matrices $A \in \text{Sym}_0(\mathbb{R}^n)$ with diagonal block structure and next neighbor couplings. The resulting GZZ gates have a constant gate time and require only linear many encodings to be implemented.

6 Efficient heuristic for fast GZZ gates

In this section, we build on the results of Lemma 8 to derive a heuristic algorithm for synthesizing $\text{GZZ}(A)$ gates with low total gate time for any $A \in \text{Sym}_0(\mathbb{R}^n)$. This algorithm runs in polynomial time as opposed to the general LP (12), which we have shown in Theorem 6 to be NP-hard. The runtime saving is due to the restriction of the ellipsope \mathcal{E}_n in Eq. (16), with exponential many elements, to a set with polynomial many elements. This restriction yields a polynomial sized LP which can be solved in polynomial time. In practice, the simplex algorithm has a runtime that scales polynomial in the problem size [25].

Recall the modified Hadamard matrix $H_{\mathbf{k}}^{d \times n}$ defined in the proof of Lemma 8, where we used the rows of $H_{\mathbf{k}}^{d \times n}$ as encodings \mathbf{m} to generate block diagonal target coupling matrices under some assumptions. Here, s is the number of block matrices on the diagonal of the target matrix, $\mathbf{k} \in \mathbb{N}^s$ contains the dimensions for each block and $d = 2^{\lceil \log_2(s) \rceil}$ is the required number of encodings to construct such a block diagonal matrix. From now on, we only consider $\mathbf{k} = (j, 1 \dots 1) \in \mathbb{N}^s$ such that

$$(H_{\mathbf{k}}^{d \times n})^T H_{\mathbf{k}}^{d \times n} - d\mathbf{1}_n = d(E_j \oplus E_1 \oplus \dots \oplus E_1), \quad (34)$$

see Eq. (21). The requirement that $\sum_i k_i = n$ implies that such a vector \mathbf{k} has $s = n - j + 1$ entries. Permuting the columns of $H_{\mathbf{k}}^{d \times n}$ results in the same permutation of the rows and columns of the right-hand side of Eq. (34). We denote the set of all column-permuted $H_{\mathbf{k}}^{d \times n}$ by $\mathcal{C}^{(j)}$. A specific element of $\mathcal{C}^{(j)}$ is denoted by $H_{\mathbf{r}}^{d \times n}$, where $\mathbf{r} \in \mathbb{N}^j$ is an ordered multi-index $r_1 < \dots < r_j$ indicating which columns of $H_{\mathbf{r}}^{d \times n}$ are identical. For example, $\mathbf{r} = (2, 5, 6)$ indicates that the columns of $H_{\mathbf{r}}^{d \times n}$ with indices 2, 5 and 6 are identical, i.e. replacing column 5 and 6 with column 2. This notation will be useful later.

Definition 12. For any $j \in \{2, 3, \dots, n\}$, we define the restricted ellipsope

$$\mathcal{E}_n^{(j)} := \left\{ \mathbf{m} \mathbf{m}^T \mid \mathbf{m} \text{ is a row of } H_{\mathbf{r}}^{d \times n} \in \mathcal{C}^{(j)} \right\}. \quad (35)$$

Further we define

$$\mathcal{E}_n^{[j]} := \bigcup_{i=2}^j \mathcal{E}_n^{(i)}. \quad (36)$$

We choose the definition in Eq. (35) similar as in Eq. (16). Next, we show the size scaling of the restricted ellipsope. This directly translates to the size and runtime of the heuristic synthesis optimization.

Proposition 13. For any $j \in \{2, 3, \dots, n\}$, the number of different encodings \mathbf{m} generating the restricted ellipsope $\mathcal{E}_n^{[j]}$ scales as $\mathcal{O}(n^{j+1})$.

Proof. Note, that $|\mathcal{C}^{(j)}| = \binom{n}{j}$ since there are j duplicate columns in $H_{\mathbf{r}}^{d \times n}$. The binomial coefficient can be bounded by $\binom{n}{j} \leq n^j/j!$. Since there are $d = 2^{\lceil \log_2(n-j+1) \rceil} < 2(n-j+1)$ rows of $H_{\mathbf{r}}^{d \times n}$ we have a rough upper bound of the number different encodings generating the restricted ellipsope, $|\mathcal{E}_n^{(j)}| \leq d \binom{n}{j} < 2(n-j+1) \binom{n}{j} < 2n^{j+1}/j!$. The first inequality is due to possible redundant encodings in the definition of $\mathcal{E}_n^{(j)}$. Similarly, we can upper bound $|\mathcal{E}_n^{[j]}| \leq \sum_{i=2}^j |\mathcal{E}_n^{(i)}| < 2 \sum_{i=2}^j n^{i+1}/i!$ which is a polynomial of order $j+1$. \square

We denote the convex cone generated by a set V by

$$\text{cone}(V) := \left\{ \sum_i \lambda_i v_i \mid \lambda_i \geq 0, v_i \in V \right\}. \quad (37)$$

With that, we are ready to present the main result of this section.

Theorem 14. $\text{cone}(\mathcal{E}_n^{(2)}) = \text{Sym}_0(\mathbb{R}^n)$.

Proof. W.l.o.g. we can assume $d = n$ and denote $H_{\mathbf{r}}^{d \times d} \in \mathcal{C}^{(2)}$ by $H_{(r_1, r_2)}^d$ with the property

$$(H_{(r_1, r_2)}^d)^T H_{(r_1, r_2)}^d - d\mathbf{1}_d = d e_{(r_1, r_2)}, \quad (38)$$

where $e_{(r_1, r_2)}$ is an element of the standard basis for symmetric matrices with vanishing diagonal. By Eq. (38) it holds $\text{Sym}_0(\mathbb{R}_{\geq 0}^n) \subseteq \text{cone}(\mathcal{E}_n^{(2)})$, i.e., symmetric matrices with non-negative entries are in the convex cone.

It is left to show that $\text{Sym}_0(\mathbb{R}_{< 0}^n) \subseteq \text{cone}(\mathcal{E}_n^{(2)})$, i.e., that also symmetric matrices with negative entries are in the convex cone. To show this inclusion we define $H_{(r_1, -r_2)}^d$ similar as $H_{(r_1, r_2)}^d$ except the duplicate column at r_2 is multiplied by -1 such that

$$(H_{(r_1, -r_2)}^d)^T H_{(r_1, -r_2)}^d - d\mathbf{1}_d = -d e_{(r_1, r_2)}. \quad (39)$$

We have to show that for each row $\mathbf{m} \in \text{rows}(H_{(r_1, -r_2)}^d)$ there exist \tilde{r}_1 and \tilde{r}_2 such that $\mathbf{m} \in \text{rows}(H_{(\tilde{r}_1, \tilde{r}_2)}^d)$. This can be verified straightforwardly for $d = 4$ by checking all rows. W.l.o.g. we show that the hypothesis holds for any $H_{(r_1, -r_2)}^{2d}$ by assuming it holds for $H_{(r_1, -r_2)}^d$. The Sylvester-Hadamard matrix is constructed inductively according to

$$H^{2d} = \begin{pmatrix} H^d & H^d \\ H^d & -H^d \end{pmatrix}. \quad (40)$$

We consider three cases for $H_{(r_1, -r_2)}^{2d}$.

Case 1. For a $H_{(r_1, -r_2)}^{2d}$ with identical columns, up to minus sign, at indices $r_1, r_2 \in [d]$ or $r_1, r_2 \in [d+1, 2d]$ the hypothesis holds by our assumption by choosing $\tilde{r}_1, \tilde{r}_2 \in [d]$ or $\tilde{r}_1, \tilde{r}_2 \in [d+1, 2d]$ respectively.

Case 2. Considering the first d rows of $H_{(r_1, -r_2)}^{2d}$ with identical columns, up to minus sign, at indices $r_1 \in [d]$ and $r_2 \in [d+1, 2d]$. This case is equivalent to *Case 1.* with $r_1, r_2 \in [d+1, 2d]$ since only the column at r_2 is negated.

Case 3. Considering the last d rows of $H_{(r_1, -r_2)}^{2d}$ with identical columns, up to minus sign, at indices $r_1 \in [d]$ and $r_2 \in [d+1, 2d]$. These rows are included in the last d rows of $H_{(\tilde{r}_1, \tilde{r}_2)}^{2d}$ with $\tilde{r}_1, \tilde{r}_2 \in [d+1, 2d]$ and $\tilde{r}_2 = r_2$ since the column r_2 is negated which is equivalent to just duplicating a column of $-H^d$.

We have shown that for each row $\mathbf{m} \in \text{rows}(H_{(r_1, -r_2)}^d)$ there exist \tilde{r}_1 and \tilde{r}_2 such that $\mathbf{m} \in \text{rows}(H_{(\tilde{r}_1, \tilde{r}_2)}^d)$. Therefore, $\text{Sym}_0(\mathbb{R}_{< 0}^n) \cup \text{Sym}_0(\mathbb{R}_{\geq 0}^n) = \text{Sym}_0(\mathbb{R}^n) = \text{cone}(\mathcal{E}_n^{(2)})$. The last equality follows from the definition of cone and $\mathcal{E}_n^{(2)}$. \square

Theorem 14 shows that the constraint of LP (12) can always be fulfilled only considering $\mathbf{m}\mathbf{m}^T \in \mathcal{E}_n^{(2)}$. Similar to Eq. (10) we define the restricted linear operator $\mathcal{V}^{[j]} : \mathbb{R}_{\geq 0}^s \rightarrow \text{Sym}_0(\mathbb{R}^n) : \boldsymbol{\lambda} \mapsto \sum_{\mathbf{m}} \lambda_{\mathbf{m}} \mathbf{m}\mathbf{m}^T$ for all $\mathbf{m}\mathbf{m}^T \in \mathcal{E}_n^{[j]}$ with $h = |\mathcal{E}_n^{[j]}|$, represented by a matrix $V^{[j]} \in \{-1, +1\}^{\binom{n}{2} \times h}$. We define the *restricted LP*^j to be

$$\begin{aligned} & \text{minimize} && \mathbf{1}^T \boldsymbol{\lambda} \\ & \text{subject to} && V^{[j]} \boldsymbol{\lambda} = \mathbf{v}(M), \\ & && \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^h. \end{aligned} \quad (41)$$

Algorithm 3 summarizes the steps to construct $\mathcal{E}_n^{[j]}$ and therefore the matrix representation of the restricted linear operator $V^{[j]}$. In practice, Algorithm 3 has to be executed only once per number of qubits n since the constraints of LP (41) can be fulfilled for all $M \in \text{Sym}_0(\mathbb{R}^n)$. This is due to the fact that $\mathcal{E}_n^{(2)} \subseteq \mathcal{E}_n^{[j]}$ for any $2 \leq j \leq n$ and $\text{cone}(\mathcal{E}_n^{(2)}) = \text{Sym}_0(\mathbb{R}^n)$ (Theorem 14). The time and space complexity of Algorithm 3 scales polynomially in n as shown in Proposition 13.

Therefore, the restricted LP^j is also of polynomial size for a fixed j . Increasing j leads to better approximations due to the enlarged search space for the optimal solution. Note, that the runtime of the mixed integer program (MIP) defined in [6, Section 2.2.2] also benefits from using $\mathcal{E}_n^{[j]}$.

As mentioned in Section 5 the dimension of the Hadamard matrices in Eq. (34) can be reduced to $d = 4\lceil(n-1)/4\rceil \leq n-5$ if we assume that the Hadamard conjecture holds. Therefore, the runtime of the restricted LP^j is reduced as well if such Hadamard matrices are used. In Section 8 we numerically benchmark the heuristic algorithm with and without the reduced runtime of the restricted LP^j .

Algorithm 3 Constructing $\mathcal{E}_n^{[j]}$.

Input: n, j

Initialize $\mathcal{E}_n^{[j]} = \emptyset$

for $i = 2, \dots, j$ **do**

Initialize $\mathcal{E}_n^{(i)} = \emptyset$

Let $H^{d \times d}$ be a Hadamard matrix

▷ With $d = 2^{\lceil \log_2(n-i+1) \rceil}$

$H^{d \times (n-i+1)} \leftarrow n-i+1$ columns of $H^{d \times d}$

▷ It does not matter which columns

for all \mathbf{r} such that $1 \leq r_1 < \dots < r_i \leq n$ **do**

▷ There are $\binom{n}{i}$ such \mathbf{r}

$H_{\mathbf{r}}^{d \times n} \leftarrow$ duplicate one column of $H^{d \times (n-i+1)}$ $i-1$ times to indices r_1, \dots, r_i

$\mathcal{E}_n^{(i)} \leftarrow \mathcal{E}_n^{(i)} \cup \{\mathbf{m}\mathbf{m}^T \mid \mathbf{m} \text{ is a row of } H_{\mathbf{r}}^{d \times n}\}$

$\mathcal{E}_n^{[j]} \leftarrow \mathcal{E}_n^{[j]} \cup \mathcal{E}_n^{(i)}$

Output: $\mathcal{E}_n^{[j]}$

7 Bounds on the total GZZ time

Our main analytic result (Theorem 15) is that the optimal total GZZ time $\mathbf{1}^T \boldsymbol{\lambda}^*$ is lower and upper bounded by the norms $\|M\|_{\ell_\infty}$ and $\|M\|_{\ell_1}$, respectively. Note, that for a dense matrix M , its norm $\|M\|_{\ell_1}$ scales quadratic with the number of qubits n . We conjecture an improved upper bound on the total GZZ time for dense M that scales at most linear with the number of qubits n . We support this conjecture with explicit solutions for the LP (12) reaching this bound for any n and numerical results validating the conjecture for $n \leq 8$.

Theorem 15. *The optimal total gate time of $\text{GZZ}(A)$ with $A \in \text{Sym}_0(\mathbb{R}^n)$ is lower and upper bounded by*

$$\|M\|_{\ell_\infty} \leq \mathbf{1}^T \boldsymbol{\lambda}^* \leq \|M\|_{\ell_1}, \quad (42)$$

where $M := A \otimes J$. Equality holds for the lower bound for all matrices $M = C\mathbf{m}\mathbf{m}^T$ for any $\mathbf{m} \in \{-1, +1\}^n$ and $C \geq 0$.

Proof. The lower bound has been shown in Lemma 7. Equality in the lower bound holds for $M = C\mathbf{m}\mathbf{m}^T$ by setting $\lambda_{\mathbf{m}} = C = \|M\|_{\ell_\infty}$ and $\lambda_{\mathbf{m}'} = 0$ for all $\mathbf{m}' \neq \mathbf{m}$. We use the explicit construction of the standard basis elements for symmetric matrices from the proof of Theorem 14 to show the upper bound. To be precise, we have

$$\frac{|M_{ij}|}{d} (H_{\mathbf{r}}^{d \times n})^T H_{\mathbf{r}}^{d \times n} - d\mathbf{1}_n = M_{ij} e_{(i,j)}, \quad (43)$$

where $\mathbf{r} = (i, j)$ if $M_{ij} \geq 0$ or $\mathbf{r} = (\tilde{i}, \tilde{j})$ if $M_{ij} < 0$ as in Eqs. (38) and (39), respectively. We define $\boldsymbol{\lambda}^{(i,j)}$ with the entries $\lambda_{\mathbf{m}}^{(i,j)} := \frac{|M_{ij}|}{d} [\mathbf{m} \in \text{rows}(H_{\mathbf{r}}^d)]$. According to Lemma 8 we have $\mathbf{1}^T \boldsymbol{\lambda}^{(i,j)} = |M_{ij}|$. Adding $\boldsymbol{\lambda}^{(i,j)}$ for all $i < j$ yields the upper bound $\|M\|_{\ell_1}$. \square

These bounds get tighter the sparser M is. If M has only one non-zero value, then clearly $\|M\|_{\ell_\infty} = \mathbf{1}^T \boldsymbol{\lambda}^* = \|M\|_{\ell_1}$. Furthermore, these bounds also hold for the heuristic, which we presented in Section 6.

Next, we state our conjecture that the optimal gate time of $\text{GZZ}(A)$ scales at most linear with the number of qubits.

Conjecture 16. *The optimal gate time of $\text{GZZ}(A)$ with $A \in \text{Sym}_0(\mathbb{R}^n)$ is tightly upper bounded by*

$$\mathbf{1}^T \boldsymbol{\lambda}^* \leq \|M\|_{\ell_\infty} \cdot \begin{cases} n, & \text{for odd } n, \\ n-1, & \text{for even } n. \end{cases} \quad (44)$$

Hence, it provides a tighter bound for dense M than Theorem 15. To support the claim of Conjecture 16 we first construct explicit dual and primal feasible solutions for the case $M = -E_n$ for the LP's (12) and (13), respectively. Then optimality is given by showing equality of the objective function values of the primal and dual problem. We further show that the case $M = -E_n$ leads to the same objective function value as $M = -\mathbf{m}\mathbf{m}^T$ for any $\mathbf{m} \in \{-1, 1\}^n$. Finally, we provide numerical evidence that the conjecture holds for $n \leq 8$.

For practical purposes it is important to keep in mind that the platform given J matrix might also scale with the number of qubits resulting in a qubit dependent constant $\|A \odot J_n\|_{\ell_\infty} = \|M_n\|_{\ell_\infty}$.

7.1 Explicit solutions for $M = -\mathbf{m}\mathbf{m}^T$

The following lemma will be used in the proof of the explicit feasible solution of the dual problem for M being of the form $M = -\mathbf{m}\mathbf{m}^T$. We can identify $\mathbf{m} \in \{-1, 1\}^n$ with $\mathbf{b} \in \mathbb{F}_2^n$ via $\mathbf{m} = (-1)^{\mathbf{b}}$ as explained in Section 2.

Lemma 17. *It holds that*

$$P_{\mathbf{b}} := \sum_{i < j}^n (-1)^{b_i \oplus b_j} = \binom{n}{2} - 2|\mathbf{b}|(n - |\mathbf{b}|), \quad (45)$$

for any binary vector $\mathbf{b} \in \mathbb{F}_2^n$. We denote the Hamming weight by $|\mathbf{b}|$.

Proof. Let $\mathbf{m} = (-1)^{\mathbf{b}}$. If the Hamming weight $|\mathbf{b}|$ vanishes, then $\mathbf{m} = (+1, \dots, +1)$ and $P_{\mathbf{b}} = \binom{n}{2}$ which is the maximal value. If $|\mathbf{b}| \neq 0$, \mathbf{m} contains $|\mathbf{b}|$ entries -1 , such that the upper triangular part of $\mathbf{m}\mathbf{m}^T$ contains a rectangle of -1 's with length $|\mathbf{b}|$ and width $n - |\mathbf{b}|$ so the total amount of -1 's is $|\mathbf{b}|(n - |\mathbf{b}|)$. Therefore,

$$P_{\mathbf{b}} = \binom{n}{2} - 2|\mathbf{b}|(n - |\mathbf{b}|). \quad (46)$$

□

Lemma 18 (explicit dual feasible solution). *Let $M = -E_n$, then there is an explicit feasible solution \mathbf{y} to the dual LP (13) with*

$$\langle -E_n, \mathbf{y} \rangle = \begin{cases} n, & \text{for odd } n, \\ n-1, & \text{for even } n. \end{cases} \quad (47)$$

Proof. We assume that $y = y_1 = y_2 = \dots$. Therefore, it suffices to show that

$$y \sum_{i < j}^n (-1)^{b_i \oplus b_j} \leq 1. \quad (48)$$

From Lemma 17 we know that $\mathbf{y} = 1/\min(P_{\mathbf{b}})\mathbf{1}$ satisfies the constraint in Eq. (48). The minimum $\min(P_{\mathbf{b}}) = -\lfloor n/2 \rfloor$ is reached for $|\mathbf{b}| = \lceil n/2 \rceil$ or $|\mathbf{b}| = \lfloor n/2 \rfloor$ which can be verified from the expression of $P_{\mathbf{b}}$ in Lemma 17. Thus, we obtain $\mathbf{y} = -1/\lfloor n/2 \rfloor \mathbf{1}$. The objective function evaluates to

$$\langle -E_n, \mathbf{y} \rangle = -\mathbf{1}^T \mathbf{y} = \frac{\binom{n}{2}}{\lfloor n/2 \rfloor} = \begin{cases} n, & \text{for odd } n, \\ n-1, & \text{for even } n. \end{cases} \quad (49)$$

□

For the construction of a feasible solution to the primal problem we first require the following result.

Lemma 19. *Let $k < n$ be natural numbers. Then*

$$\sum_{\mathbf{b} \in \mathbb{F}_2^n : |\mathbf{b}|=k} |b_i \oplus b_j| = 2 \binom{n-2}{k-1} \quad (50)$$

for all $i, j \in [n]$ with $i \neq j$.

Proof. Consider the case $n = 4$ and $k = 2$. We get $\binom{4}{2} = 6$ binary vectors with $|\mathbf{b}| = 2$. It can be easily verified that $\sum_{|\mathbf{b}|=2} |b_i \oplus b_j| = 4 = 2 \binom{2}{1}$. Now, we assume that for a given n and $k < n$ the Eq. (50) holds. It suffices to verify Eq. (50) for $i \leq n$ and $j = n+1$. We fix k , define $n' := n+1$ and take a $\mathbf{b} \in \mathbb{F}_2^{n'}$. We have $\binom{n+1}{k}$ binary vectors with $|\mathbf{b}| = k$.

For the case $b_{n+1} = 0$ we have $\binom{n}{k}$ such vectors and

$$\sum_{\substack{|\mathbf{b}|=k \\ b_{n+1}=0}} |b_i \oplus b_{n+1}| = \sum_{|\mathbf{b}|=k} |b_i| = \binom{n-1}{k-1}, \quad (51)$$

for all $i \leq n$. There are $\binom{n-1}{k-1}$ different ways in placing $k-1$ 1's.

For the case $b_{n+1} = 1$ we have $\binom{n}{k-1}$ such vectors and

$$\sum_{\substack{|\mathbf{b}|=k \\ b_{n+1}=1}} |b_i \oplus b_{n+1}| = \sum_{|\mathbf{b}|=k} |b_i \oplus 1| = \binom{n-1}{k-1}, \quad (52)$$

for all $i \leq n$. There are $\binom{n-1}{k-1}$ different ways in placing $k-1$ 0's.

Combining the two cases $b_{n+1} = 0$ and $b_{n+1} = 1$ we obtain

$$\sum_{|\mathbf{b}|=k} |b_i \oplus 0| + |b_i \oplus 1| = 2 \binom{n-1}{k-1} = 2 \binom{n'-2}{k-1}, \quad (53)$$

for all $i \leq n$. □

We motivate the next lemma with the result of the explicit dual feasible solution for $M = -E_n$ from Lemma 18 and the complementary slackness condition. The complementary slackness condition for a linear program states that if the i -th inequality of the dual problem is a strict inequality for a feasible solution \mathbf{y} , then the i -th component of a feasible solution of the primal problem $\boldsymbol{\lambda}$ is zero:

$$(V^T \mathbf{y})_i < 1 \Rightarrow \lambda_i = 0. \quad (54)$$

We use this in the following lemma to construct a feasible solution for the primal LP (12).

Lemma 20 (explicit primal feasible solution). *Let $M = -E_n$, then there is an explicit feasible solution $\boldsymbol{\lambda}$ to the primal LP (12) with*

$$\mathbf{1}^T \boldsymbol{\lambda} = \begin{cases} n, & \text{for odd } n, \\ n-1, & \text{for even } n. \end{cases} \quad (55)$$

Proof. For this proof we define

$$k := \begin{cases} n, & \text{for odd } n, \\ n-1, & \text{for even } n. \end{cases} \quad (56)$$

We only consider binary vectors of the set $S := \{\mathbf{b} \in \mathbb{F}_2^n \mid |\mathbf{b}| = \lceil n/2 \rceil, \lfloor n/2 \rfloor \text{ and } b_n = 0\}$. This set is motivated by the complementary slackness condition and Lemma 18. It can be calculated that $|S| = \binom{k}{\lfloor k/2 \rfloor}$ using the recurrence relation

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \quad (57)$$

for the binomial coefficients. We show that

$$\sum_{\mathbf{b} \in S} (-1)^{b_i \oplus b_j} = -D_n, \quad (58)$$

for a constant $D_n > 0$, which we calculate later. If not explicitly stated, all equations in this proof containing i, j hold for all $i, j \in [n]$, $i \neq j$. We denote λ_S by all $\lambda_{\mathbf{m}} = \lambda_{\mathbf{b}}$ corresponding to the encoding $\mathbf{m} = (-1)^{\mathbf{b}}$ with $\mathbf{b} \in S$. If Eq. (58) holds, we can choose a $\lambda_S = 1/D_n \mathbf{1}$ resulting in

$$\sum_{\mathbf{b} \in S} \lambda_{\mathbf{b}} (-1)^{b_i \oplus b_j} = -1, \quad (59)$$

which implies feasibility for $M = -E_n$. It is left to show Eq. (58) and determine D_n .

First, we consider odd n . By definition of S we have that $b_n = 0$ for all \mathbf{b} . Therefore, we obtain $\binom{n-1}{\lfloor n/2 \rfloor} + \binom{n-1}{\lceil n/2 \rceil}$ binary vectors with $|\mathbf{b}| = \lceil n/2 \rceil$ or $|\mathbf{b}| = \lfloor n/2 \rfloor$ respectively. Counting the occurrences of “−1” in the sum of Eq. (58) is equivalent to counting the occurrences of “1” in the sum

$$\begin{aligned} \sum_{\mathbf{b} \in S} |b_i \oplus b_j| &= 2 \left(\binom{n-3}{\lfloor \frac{n}{2} \rfloor - 1} + \binom{n-3}{\lceil \frac{n}{2} \rceil - 1} \right) \\ &= 2 \binom{n-2}{\lfloor \frac{n}{2} \rfloor}, \end{aligned} \quad (60)$$

where we used Lemma 19, the recurrence relation for the binomial coefficients and $\lceil n/2 \rceil - 1 = \lfloor n/2 \rfloor$. Counting the occurrences of “+1” in the sum of Eq. (58) yields

$$\binom{n-1}{\lfloor \frac{n}{2} \rfloor} + \binom{n-1}{\lceil \frac{n}{2} \rceil} - 2 \binom{n-2}{\lfloor \frac{n}{2} \rfloor} = \binom{n}{\lfloor \frac{n}{2} \rfloor} - 2 \binom{n-2}{\lfloor \frac{n}{2} \rfloor}. \quad (61)$$

where we used $\binom{n}{\lfloor n/2 \rfloor + 1} = \binom{n}{\lfloor n/2 \rfloor}$ for odd n . We now evaluate Eq. (58) for odd n

$$\begin{aligned} -D_n &= \sum_{\mathbf{b} \in S} (-1)^{b_i \oplus b_j} = \binom{n}{\lfloor \frac{n}{2} \rfloor} - 4 \binom{n-2}{\lfloor \frac{n}{2} \rfloor} \\ &= \binom{n}{\lfloor \frac{n}{2} \rfloor} - \frac{n+1}{n} \binom{n}{\lfloor \frac{n}{2} \rfloor} \\ &= \frac{-1}{n} \binom{n}{\lfloor \frac{n}{2} \rfloor}. \end{aligned} \quad (62)$$

The case for even n follows the same steps as for odd n , resulting in

$$\begin{aligned} -D_n &= \sum_{|\mathbf{b}| = \frac{n}{2}} (-1)^{b_i \oplus b_j} = \binom{n-1}{\frac{n}{2}} - \frac{n}{n-1} \binom{n-1}{\frac{n}{2}} \\ &= \frac{-1}{n-1} \binom{n-1}{\frac{n}{2}}. \end{aligned} \quad (63)$$

Equations (62) and (63) show that $D_n = 1/k \binom{k}{\lfloor k/2 \rfloor}$ and that Eq. (58) holds. Since $|S| = \binom{k}{\lfloor k/2 \rfloor}$ the objective function value is

$$\mathbf{1}^T \lambda_S = \binom{k}{\lfloor k/2 \rfloor} \frac{1}{\frac{1}{k} \binom{k}{\lfloor k/2 \rfloor}} = k. \quad (64)$$

□

From the equality of the objective functions for the primal and dual problem from Lemma 20 and Lemma 18 respectively we know that the proposed dual/primal feasible solutions for $M = -E_n$ are in fact optimal solutions. Now, we show that the GZZ gate time with $M = -\mathbf{m}\mathbf{m}^T$ for any $\mathbf{m} \in \{-1, +1\}^n$ is the same as for the case $M = -E_n$.

Theorem 21. If $A \odot J =: M = -C(\mathbf{m}\mathbf{m}^T)$ for any $\mathbf{m} \in \{-1, +1\}^n$ and $C \geq 0$, then the optimal gate time of $\text{GZZ}(A)$ is

$$\mathbf{1}^T \boldsymbol{\lambda}^* = \|M\|_{\ell_\infty} \begin{cases} n, & \text{for odd } n, \\ n-1, & \text{for even } n. \end{cases} \quad (65)$$

Proof. The statement has been shown for the case $M = -E_n$ by constructing an explicit solution. It is left to show that the cases $M = -C(\mathbf{m}\mathbf{m}^T)$ for a constant $C \geq 0$ yield the same objective function value. Since the objective function and the constraints are linear we can w.l.o.g. assume $C = 1$. It is clear, that $\text{sign}(M_{ij}) = \text{sign}(-m_i m_j) = -(-1)^{c_i \oplus c_j}$ for all $i < j$, with $\mathbf{m} = (-1)^{\mathbf{c}}$. Let $\mathbf{y} = -y \text{sign}(\mathbf{v}(M))$ for a $y \in \mathbb{R}$, then each constraint of $V^T \mathbf{y} \leq \mathbf{1}$ of the dual LP (13) reads as

$$y \sum_{i < j} (-1)^{b_i \oplus b_j \oplus c_i \oplus c_j} = y \sum_{i < j} (-1)^{\tilde{b}_i \oplus \tilde{b}_j} \leq 1, \quad (66)$$

with $\tilde{\mathbf{b}} := \mathbf{b} \oplus \mathbf{c}$ element wise. Consider the ordered set of all $\mathbf{b} \in \mathbb{F}_2^n$ with $b_n = 0$, then $\tilde{\mathbf{b}}$ is just a permutation of that set. Due to the permutation symmetry of the qubits the optimal value of the LP (12) for any $M = -(\mathbf{m}\mathbf{m}^T)$ is the same as for the case $M = -E_n$. Setting $\mathbf{y} = -y \text{sign}(\mathbf{v}(M))$ with $y = 1/\lfloor n/2 \rfloor$ as in the proof of Lemma 18 yields an optimal solution to the dual LP (13) for the case $M = -(\mathbf{m}\mathbf{m}^T)$. \square

Note that, trivially, the lower bound $\mathbf{1}^T \boldsymbol{\lambda}^* = \|M\|_{\ell_\infty}$ is reached if $M = C(\mathbf{m}\mathbf{m}^T)$ for any $\mathbf{m} \in \{-1, +1\}^n$ and $C \geq 0$.

One possibility to prove Conjecture 16 is to show, that the matrix $M = -E_n$ maximizes the value of the LP (12) among all matrices $M \in \text{Sym}_0([-1, +1]^n)$. To this end, consider the LP

$$\begin{aligned} & \text{maximize} && \|\mathbf{y}\|_{\ell_1} \\ & \text{subject to} && V^T \mathbf{y} \leq \mathbf{1}, \\ & && \mathbf{y} \in \mathbb{R}^{\binom{n}{2}}, \end{aligned} \quad (67)$$

which is independent of $M \in \text{Sym}_0([-1, +1]^n)$. It holds that

$$\max_{V^T \mathbf{y} \leq \mathbf{1}} \left(\max_{\|M\|_{\ell_\infty} \leq 1} \langle M, \mathbf{y} \rangle \right) = \max_{V^T \mathbf{y} \leq \mathbf{1}} \|\mathbf{y}\|_{\ell_1} \quad (68)$$

according to $\|\mathbf{x}\|_{\ell_1} = \max_{\|\mathbf{p}\|_{\ell_\infty} \leq 1} \mathbf{p}^T \mathbf{x}$. Therefore, the optimal objective value of LP (67) is an upper bound on all optimal objective values of LP (12). Clearly, the constructed solution in Lemma 18 is feasible for LP (67). Unfortunately, proving that this constructed solution is optimal is quite challenging, as we discuss in Appendix A.

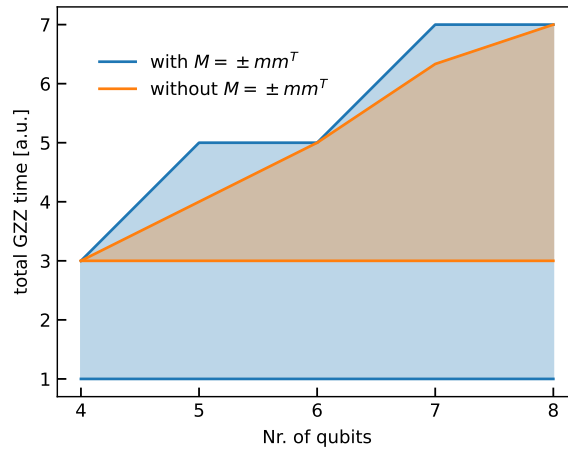


Figure 1: The optimal objective function value of the dual LP (13) over the number of qubits. **Blue:** The range of optimal values for all binary $M \in \text{Sym}_0(\{-1, +1\}^n)$. **Orange:** The range of optimal values for all binary $M \in \text{Sym}_0(\{-1, +1\}^n)$ without $M = \pm(\mathbf{m}\mathbf{m}^T)$ for any $\mathbf{m} \in \{-1, +1\}^n$.

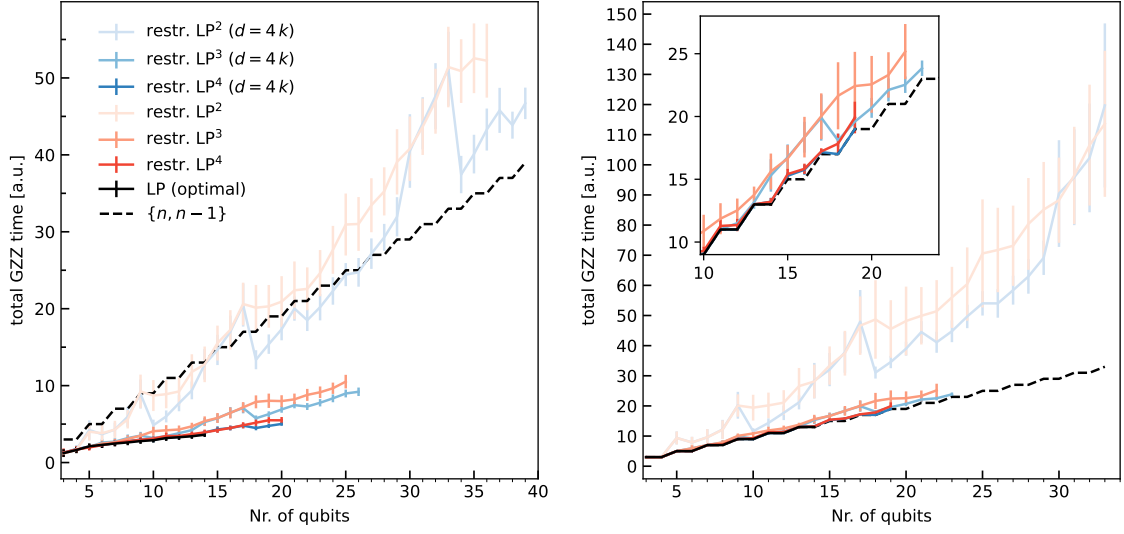


Figure 2: Comparing the performance of the original (optimal) LP (12) and the restricted LP^j (41) for $j = 2, 3, 4$. For each line we let the LP's run for a fixed time. The black dashed line is the upper bound for the original LP (12) from Conjecture 16. The reddish lines show the total GZZ times for the restricted LP^j (41) for $j = 2, 3, 4$. The blueish lines show the total GZZ times for the restricted LP^j (41) using Hadamard matrices of dimension $d = 4k$ to construct the restricted linear operator $V^{[j]}$. **Left:** Average case scaling of the total GZZ times. The data points and error bars show the mean and the standard deviation over 100 uniformly sampled matrices $A_{ij} = A_{ji} \in [-1, 1]$ for $i < j$. **Right:** Conjectured worst-case scaling of the total GZZ times. The data points and error bars show the mean and the standard deviation over 100 binary matrices $A = -\mathbf{m}\mathbf{m}^T$ with uniformly sampled encodings $\mathbf{m} \in \{-1, 1\}^n$.

8 Numerical results

8.1 Numeric validation of Conjecture 16 for small n

For the numeric validation of the conjecture for small n we solve the dual LP (13) for all binary $M \in \text{Sym}_0(\{-1, +1\}^n)$ of which there are $2^{\binom{n}{2}}$. For $n = 3$ there are only binary matrices of the form $M = \pm(\mathbf{m}\mathbf{m}^T)$ and by Theorem 21 the conjecture holds. Figure 1 shows that Conjecture 16 holds for $n \leq 8$. For odd $n \leq 8$ the cases $M = -(\mathbf{m}\mathbf{m}^T)$ are in fact the only cases reaching the upper bound. This can be seen in Fig. 1 by the blue area exceeding the orange area, which only consists of the optimal values for all binary matrices without $M = \pm(\mathbf{m}\mathbf{m}^T)$ for any $\mathbf{m} \in \{-1, +1\}^n$.

8.2 Numerical benchmark for the heuristic

We compare the performance of the restricted LP^j (41) to the original LP (12). To this end we provide numerical results on the average-case and the conjectured worst-case scaling of the total GZZ time. The left plot in Fig. 2 shows the average-case scaling of the total GZZ time. GZZ(A) gates with uniformly sampled matrix elements $A_{ij} = A_{ji} \in [-1, 1]$ for all $i < j$ and $i, j \in [n]$ are synthesized. For the worst-case scaling of the right plot in Fig. 2, GZZ(A) gates with binary matrices $A = -\mathbf{m}\mathbf{m}^T$ with uniformly sampled encodings $\mathbf{m} \in \{-1, 1\}^n$ are synthesized. We note that these are the matrices with the conjectured worst-case scaling for the LP and that the LP^j might have different worst-case matrices A . For convenience, we have set $M = A$, i.e. setting $J = E_n$ which omits the hardware specific time units given by the quantum platform. For realistic J 's and GZZ(A) gate times we refer to [6]. The Python package CVXPY [26, 27] with the GNU linear program kit simplex solver [28] is used to solve the LP (12) and the restricted LP^j (41).

Figure 2 shows the total GZZ time over the number of qubits n for the restricted LP^j (41) and LP (12). We assigned a fixed runtime (20 minutes) for each LP to synthesize GZZ(A) gates for all sample matrices A and as many n as possible. Clearly, the runtime of the heuristic algorithm is much shorter than the runtime of the optimal LP (12). Increasing the hierarchy of the heuristic

$j > 2$ reduces the total GZZ time significantly while still maintaining a short runtime. The total GZZ time obtained from the heuristic also seems to scale linear with the number of qubits although with a different scaling constant. As mentioned in Sections 5 and 6 the size of the restricted LP^j (41), and therefore the runtime, can be further reduced. This reduction is achieved by using Hadamard matrices of dimension $d = 4k$ with $k \in \mathbb{N}$ instead of Sylvester-Hadamard matrices of dimension $d = 2^k$ in the construction of the restricted linear operator $V^{[j]}$. This is based on the Hadamard conjecture, which is known to be true for $d \leq 668$ [22, 23]. Surprisingly, using these Hadamard matrices with $d = 4k$ not only yields shorter runtime of the heuristic algorithm but also a significant reduction of the total GZZ time compared to the original restricted LP^j (41).

Our numerical results show that the heuristic algorithm approximates well the optimal total GZZ time, while maintaining a short runtime. This holds true for both, the average-case and the conjectured worst-case scaling. Therefore, we hope that this heuristic will prove to be an important tool to implement fast GZZ gates in practice.

9 Conclusion

We investigated the time-optimal multi-qubit gate synthesis introduced in Ref. [6]. We show that synthesizing time-optimal multi-qubit gates in our setting is NP-hard. However, we also provide explicit solutions for certain cases with constant gate time and a polynomial-time heuristics to synthesize fast multi-qubit gates. Our numerical simulations suggest that these heuristics provide good approximations to the optimal GZZ gate time. Furthermore, tight bounds on the scaling of the optimal multi-qubit gate times were shown. More precisely, we showed that the optimal multi-qubit gate time scales at most as $\|A \oslash J\|_{\ell_1}$, the ℓ_1 -norm of the element-wise division of the total and physical coupling matrices A and J , respectively. We also conjectured that the optimal GZZ gate time scales at most linear with the number of qubits. Our results are practical to estimate the execution time of a given circuit, where the entangling gates are implemented as GZZ gates. The execution time is a crucial parameter, in particular, in the NISQ era since it is limiting the length of a gate sequence due to finite coherence time.

It is our hope to proof the conjectured linear scaling of the optimal GZZ gate time in the near future. Moreover, we would like to test and verify our proposed time-optimal multi-qubit gate synthesis methods in an experiment. Depending on the quantum platform we would like to develop adapted error mitigation schemes for the GZZ gates and investigate their robustness against errors.

Acknowledgements

We are grateful to Lennart Bittel and Arne Heimendahl for fruitful discussions on complexity theory and convex optimization, respectively. We also want to thank Frank Vallentin for valuable comments on our conjecture and proof ideas.

This work has been funded by the German Federal Ministry of Education and Research (BMBF) within the funding program “quantum technologies – from basic research to market” via the joint project MIQRO (grant number 13N15522) and the Fujitsu Services GmbH as part of an endowed professorship “Quantum Inspired and Quantum Optimization”.

Appendices

A Challenges in proving Conjecture 16

In this section, we want to discuss some obstacles encountered trying to proof Conjecture 16. Conjecture 16 holds, if we show that our constructed solutions from Section 7.1 are optimal solutions for LP (67). Meaning that there is no other feasible solution resulting in a larger objective function value compared to our constructed solution.

We tried an inductive proof which turned out to be intricate due to the additional degrees of freedom in each induction step. Furthermore, we utilized the connection to graph theory from Lemma 2 to transform the LP (67) to a LP over the cut polytope. There, the challenge is the affine mapping between the elliptope and the cut polytope in Eq. (16) and Definition 1, which alters the optimization problem crucially. In the following, we discuss another approach based on showing the sufficiency of the Karush–Kuhn–Tucker (KKT) conditions in more detail.

A.1 Concave program

A convex linear program in standard form minimizes a convex objective function over a convex set. But LP (67) *maximizes* a convex objective function over a convex set. Such optimizations are called *concave programs*. It is known that the maximum is attained at the extreme points of the polytope $V^T \mathbf{y} \leq \mathbf{1}$ and therefore might have many local optima [29]. There are several equivalent sufficient conditions for optimality [30]. Here, we investigate one in detail. First, we define the *conjugate function* for a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$

$$f^*(\mathbf{x}) := \sup \{ \mathbf{x}^T \mathbf{y} - f(\mathbf{y}) \mid \mathbf{y} \in \mathcal{D}(f) \}, \quad (69)$$

with $\mathcal{D}(f)$ the domain of f . Furthermore, we need the *support function* $h_A : \mathbb{R}^n \rightarrow \mathbb{R}$ for a closed convex set A

$$h_A(\mathbf{x}) := \sup \{ \mathbf{x}^T \mathbf{y} \mid \mathbf{y} \in A \}. \quad (70)$$

Then the sufficient optimality condition in our case is

$$\|\mathbf{y}^*\|_{\ell_1} = \sup \left\{ h_A(\mathbf{x}) - (\|\mathbf{x}\|_{\ell_1})^* \mid \mathbf{y} \in \mathbb{R}^{\binom{n}{2}} \right\}, \quad (71)$$

with $A = \{ \mathbf{x} \in \mathbb{R}^{\binom{n}{2}} \mid V^T \mathbf{x} \leq \mathbf{1} \}$. The conjugate function of $\|\mathbf{x}\|_{\ell_1}$ is

$$(\|\mathbf{x}\|_{\ell_1})^* = \begin{cases} 0, & \text{if } \|\mathbf{x}\|_{\ell_\infty} \leq 1, \\ \infty, & \text{otherwise.} \end{cases} \quad (72)$$

Then we have

$$\begin{aligned} \|\mathbf{y}^*\|_{\ell_1} &= \sup \{ h_A(\mathbf{y}) \mid \|\mathbf{y}\|_{\ell_\infty} \leq 1 \} \\ &= \sup \{ \mathbf{x}^T \mathbf{y} \mid \|\mathbf{y}\|_{\ell_\infty} \leq 1, V^T \mathbf{x} \leq \mathbf{1} \} \\ &= \sup \{ \|\mathbf{x}\|_{\ell_1} \mid V^T \mathbf{x} \leq \mathbf{1} \}, \end{aligned} \quad (73)$$

which is the same formulation as the original LP (67).

A.2 Dualization

If we can formulate the dual LP to the primal LP (67) and find a feasible solution, then the dual objective function value upper bounds the primal objective function value by weak duality [31]. The standard form of an optimization problem with only linear inequality constraints is

$$\begin{aligned} &\text{minimize} && f(\mathbf{y}) \\ &\text{subject to} && A\mathbf{y} \leq \mathbf{b}. \end{aligned} \quad (74)$$

Then, the Lagrange dual function is given by

$$g(\boldsymbol{\lambda}) = -\mathbf{b}^T \boldsymbol{\lambda} - f^*(-A^T \boldsymbol{\lambda}), \quad (75)$$

where f^* denotes the conjugate function [14]. For LP (67) we have $f(\mathbf{y}) = -\|\mathbf{y}\|_{\ell_1}$ (minus sign due to the minimization in the standard optimization form).

$$\begin{aligned} (-\|\mathbf{x}\|_{\ell_1})^* &= \sup \left\{ \mathbf{x}^T \mathbf{y} + \|\mathbf{x}\|_{\ell_1} \mid \mathbf{y} \in \mathbb{R}^{\binom{n}{2}} \right\} \\ &= \sup \left\{ \mathbf{x}^T \mathbf{y} + \mathbf{x}^T \mathbf{z} \mid \mathbf{y} \in \mathbb{R}^{\binom{n}{2}}, \|\mathbf{z}\|_{\ell_\infty} \leq 1 \right\} \\ &= \sup \left\{ \mathbf{x}^T \mathbf{y} \mid \mathbf{y} \in \mathbb{R}^{\binom{n}{2}} \right\}, \end{aligned} \quad (76)$$

which clearly is unbounded if $\mathbf{x} \neq \mathbf{0}$. Therefore, we cannot formulate the dual to LP (67).

A.3 Invexity

The KKT conditions are optimality conditions for non-linear optimization problems. Invexity is a generalization of convexity in the sense that the KKT conditions are necessary and sufficient for optimality [32]. By invexity we mean that the objective and constraint functions of the optimization problem are *Type 1 invex* functions.

Definition 22. Consider the standard form of an optimization problem

$$\begin{aligned} &\text{minimize} && f(\mathbf{y}) \\ &\text{subject to} && g(\mathbf{y}) \leq \mathbf{0}, \\ &&& \mathbf{y} \in \mathcal{S}, \end{aligned} \quad (77)$$

where $\mathcal{S} \subseteq \mathbb{R}^m$ is defined by $g(\mathbf{y}) \leq \mathbf{0}$. Then f and g are called Type 1 invex functions at point $\mathbf{y}^* \in \mathcal{S}$ w.r.t. a common function $\eta(\mathbf{y}, \mathbf{y}^*) \in \mathbb{R}^m$, if for all $\mathbf{y} \in \mathcal{S}$,

$$\begin{aligned} f(\mathbf{y}) - f(\mathbf{y}^*) &\geq \eta(\mathbf{y}, \mathbf{y}^*)^T \nabla f(\mathbf{y}^*), \\ -g(\mathbf{y}^*) &\geq \eta(\mathbf{y}, \mathbf{y}^*)^T \nabla g(\mathbf{y}^*) \end{aligned} \quad (78)$$

hold. It suffices to consider only the active constraints, i.e. the constraints, where equality holds $g(\mathbf{y}^*) = 0$. [32]

Let K be the scaling factor of the conjectured upper bound, i.e.

$$K := \begin{cases} n, & \text{for odd } n, \\ n-1, & \text{for even } n. \end{cases} \quad (79)$$

In the case of LP (67) we have $\mathcal{S} = \left\{ \mathbf{y} \in \mathbb{R}^{\binom{n}{2}} \mid V^T \mathbf{y} \leq \mathbf{1} \right\}$ and want to show that $\mathbf{y}^* = -\mathbf{1}/\lfloor n/2 \rfloor$ is a global optimum. Furthermore, we have

$$\begin{aligned} f(\mathbf{y}) &= -\|\mathbf{y}\|_{\ell_1}, & f(\mathbf{y}^*) &= -K, & \nabla f(\mathbf{y}^*) &= \mathbf{1}, \\ g(\mathbf{y}) &= V^T \mathbf{y} - \mathbf{1}, & g(\mathbf{y}^*) &= 0, & \nabla g(\mathbf{y}^*) &= V|_a, \end{aligned} \quad (80)$$

where $V|_a$ are the columns of V such that $(V|_a)^T \mathbf{y}^* = \mathbf{1}$, i.e. the active constraints. To show invexity we have to find a common $\eta(\mathbf{y}, \mathbf{y}^*) \in \mathbb{R}^m$ such that

$$\begin{aligned} \|\mathbf{y}\|_{\ell_1} &\leq K - \eta^T(\mathbf{y}, \mathbf{y}^*) \mathbf{1} \quad \text{and} \\ (V^T|_a) \eta(\mathbf{y}, \mathbf{y}^*) &\leq \mathbf{0}, \end{aligned} \quad (81)$$

for all $\mathbf{y} \in \mathcal{S}$. It is quite challenging to find a $\eta(\mathbf{y}, \mathbf{y}^*) \in \mathbb{R}^m$ satisfying both inequalities. In particular, ansätze motivated from the geometry in small dimensions eventually fail for larger n .

B Acronyms

IQP	Instantaneous Quantum Polynomial time	2
KKT	Karush–Kuhn–Tucker	20
LP	linear program	1
MIP	mixed integer program	13
NISQ	noisy and intermediate scale quantum	2

References

- [1] X. Wang, A. Sørensen, and K. Mølmer, *Multibit gates for quantum computing*, *Phys. Rev. Lett.* **86**, 3907 (2001), [arXiv:quant-ph/0012055](#).
- [2] T. Monz, P. Schindler, J. T. Barreiro, M. Chwalla, D. Nigg, W. A. Coish, M. Harlander, W. Hänsel, M. Hennrich, and R. Blatt, *14-qubit entanglement: Creation and coherence*, *Phys. Rev. Lett.* **106**, 130506 (2011), [arXiv:1009.6126](#).
- [3] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, *Superconducting qubits: Current state of play*, *Annual Review of Condensed Matter Physics* **11**, 369 (2020), [arXiv:1905.13641](#).
- [4] C. Figgatt, A. Ostrander, N. M. Linke, K. A. Landsman, D. Zhu, D. Maslov, and C. Monroe, *Parallel entangling operations on a universal ion-trap quantum computer*, *Nature* **572**, 368 (2019), [arXiv:1810.11948](#).
- [5] Y. Lu, S. Zhang, K. Zhang, W. Chen, Y. Shen, J. Zhang, J.-N. Zhang, and K. Kim, *Scalable global entangling gates on arbitrary ion qubits*, *Nature* **572**, 363 (2019), [arXiv:1901.03508](#).
- [6] P. Bakler, M. Zipper, C. Cedzich, M. Heinrich, P. H. Huber, M. Johanning, and M. Kliesch, *Synthesis of and compilation with time-optimal multi-qubit gates*, *Quantum* **7**, 984 (2023), [arXiv:2206.06387](#).
- [7] F. Barahona and A. R. Mahjoub, *On the cut polytope*, *Mathematical Programming* **36**, 157 (1986).
- [8] M. R. Garey and D. S. Johnson, *Computers and intractability*, Vol. 29 (W. H. Freeman and Company, New York, 2002).
- [9] M. J. Bremner, A. Montanaro, and D. J. Shepherd, *Average-case complexity versus approximate simulation of commuting quantum computations*, *Phys. Rev. Lett.* **117**, 080501 (2016), [arXiv:1504.07999](#).
- [10] J. Allcock, J. Bao, J. F. Doriguello, A. Luongo, and M. Santha, *Constant-depth circuits for Uniformly Controlled Gates and Boolean functions with application to quantum memory circuits*, [arXiv:2308.08539](#) (2023).
- [11] S. Bravyi, D. Maslov, and Y. Nam, *Constant-cost implementations of Clifford operations and multiply controlled gates using global interactions*, *Phys. Rev. Lett.* **129**, 230501 (2022), [arXiv:2207.08691](#).
- [12] S. Bravyi and D. Maslov, *Hadamard-free circuits expose the structure of the Clifford group*, *IEEE Trans. Inf. Theory* **67**, 4546 (2021), [arXiv:2003.09412](#).
- [13] D. Maslov and B. Zindorf, *Depth optimization of CZ, CNOT, and Clifford circuits*, *IEEE Transactions on Quantum Engineering* **3**, 1 (2022), [arxiv:2201.05215](#).
- [14] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, 2009).
- [15] E. Rich, *The problem classes FP and FNP*, in *Automata, Computability and Complexity: Theory and Applications* (Pearson Education, 2007) pp. 510–511.
- [16] M. Johanning, *Isospaced linear ion strings*, *Appl. Phys. B* **122**, 71 (2016).
- [17] M. Laurent and S. Poljak, *On a positive semidefinite relaxation of the cut polytope*, *Linear Algebra and its Applications* **223–224**, 439 (1995).
- [18] M. M. Deza and M. Laurent, *Geometry of Cuts and Metrics*, 1st ed., Algorithms and Combinatorics (Springer Berlin Heidelberg, 2009).
- [19] M. E.-Nagy, M. Laurent, and A. Varvitsiotis, *Complexity of the positive semidefinite matrix completion problem with a rank constraint*, *Springer International Publishing*, 105 (2013), [arXiv:1203.6602](#).

- [20] R. E. A. C. Paley, *On orthogonal matrices*, [Journal of Mathematics and Physics](#) **12**, 311 (1933).
- [21] A. Hedayat and W. D. Wallis, *Hadamard matrices and their applications*, [The Annals of Statistics](#) **6**, 1184 (1978).
- [22] H. Kharaghani and B. Tayfeh-Rezaie, *A Hadamard matrix of order 428*, [Journal of Combinatorial Designs](#) **13**, 435 (2005).
- [23] D. Ž. Đoković, O. Golubitsky, and I. S. Kotsireas, *Some new orders of Hadamard and Skew-Hadamard matrices*, [Journal of Combinatorial Designs](#) **22**, 270 (2014), [arXiv:1301.3671](#).
- [24] J. Cohn, M. Motta, and R. M. Parrish, *Quantum filter diagonalization with compressed double-factorized Hamiltonians*, [PRX Quantum](#) **2**, 040352 (2021), [arXiv:2104.08957](#).
- [25] D. A. Spielman and S.-H. Teng, *Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time*, [Journal of the ACM](#) **51**, 385 (2004), [arXiv:cs/0111050](#).
- [26] S. Diamond and S. Boyd, *CVXPY: A Python-embedded modeling language for convex optimization*, [J. Mach. Learn. Res.](#) **17**, 1 (2016), [arXiv:1603.00943](#).
- [27] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, *A rewriting system for convex optimization problems*, [J. Control Decis.](#) **5**, 42 (2018), [arXiv:1709.04494](#).
- [28] Free Software Foundation, *GLPK (GNU Linear Programming Kit)* (2012), version: 0.4.6.
- [29] A. T. Phillips and J. B. Rosen, *A parallel algorithm for constrained concave quadratic global minimization*, [Mathematical Programming](#) **42**, 421 (1988).
- [30] M. Dür, R. Horst, and M. Locatelli, *Necessary and sufficient global optimality conditions for convex maximization revisited*, [Journal of Mathematical Analysis and Applications](#) **217**, 637 (1998).
- [31] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms, 3rd edition* (John Wiley & sons, 2013).
- [32] M. A. Hanson, *Inverity and the Kuhn–Tucker Theorem*, [Journal of Mathematical Analysis and Applications](#) **236**, 594 (1999).

C Paper - General, efficient, and robust Hamiltonian engineering

Title: General, efficient, and robust Hamiltonian engineering

Authors: Pascal Baßler, Markus Heinrich, Martin Kliesch

Journal: Unpublished

Publication status: Accepted at PRX Quantum

Contribution by PB: Main author (input approx 90%)

A summary of this publication is presented in chapter 4.

The idea for the project came from the generalization of our first two works to local Hamiltonians [32, 33]. The main results and their proofs were then developed by me, and subsequently discussed with all co-authors. The proofs were carefully checked and corrected by MH and MK. Potential proof ideas for the Numerical Observation III.7 and III.8 were elaborated in detail with all authors. All Python code was developed by me and the numerical simulations were carried out by me. I wrote the initial draft of the manuscript, with significant contributions by MK. The manuscript was then proofread and improved by all my co-authors.

General, efficient, and robust Hamiltonian engineering

P. Bakler,^{1,2,*} M. Heinrich,^{1,3} and M. Kliesch^{2,†}

¹*Heinrich Heine University Düsseldorf, Faculty of Mathematics and Natural Sciences, Düsseldorf, Germany*

²*Hamburg University of Technology, Institute for Quantum Inspired and Quantum Optimization, Hamburg, Germany*

³*Institute for Theoretical Physics, University of Cologne, Cologne, Germany[‡]*

Implementing the time evolution under a desired target Hamiltonian is critical for various applications in quantum science. Due to the exponential increase of parameters in the system size and due to experimental imperfections this task can be challenging in quantum many-body settings.

We introduce an efficient and robust scheme to engineer arbitrary local many-body Hamiltonians. This is achieved by applying simple single-qubit gates simultaneously to an always-on system Hamiltonian, which we assume to be native to a given platform. These sequences are constructed by efficiently solving a linear program (LP) which minimizes the total evolution time. In this way, we can engineer target Hamiltonians that are only limited by the locality of the Pauli terms in the system Hamiltonian. Based on average Hamiltonian theory and by using robust composite pulses, we make our schemes robust against errors including finite pulse time errors and various calibration errors.

To demonstrate the performance of our scheme, we provide numerical simulations. In particular, we solve the Hamiltonian engineering problem for arbitrary two-body Hamiltonians on a 2D square lattice with 225 qubits in only 60 seconds. Moreover, we simulate the time evolution of Heisenberg Hamiltonians for smaller system sizes with a fidelity larger than 99.9%, which is orders of magnitude better than non-robust implementations.

I. INTRODUCTION

Simulating a target Hamiltonian on a quantum system is a central problem in quantum computing, with applications in general gate-based and digital-analog quantum computing, as well as quantum simulations [1–6] and quantum chemistry [7–9]. It is commonly believed that simulating the dynamics of a quantum system is one of the most promising tasks for showing a practical advantage of quantum computers over classical computers [7, 10, 11], perhaps already on noisy and intermediate scale quantum (NISQ) devices [12, 13]. Especially for the latter, it is essential to have an efficient and fast implementation of the target Hamiltonian due to short coherence times.

The idea to engineer a target Hamiltonian by interleaving the evolution under a fixed system Hamiltonian with single-qubit gates originated in the nuclear magnetic resonance (NMR) community. However, the first approaches require additional single-qubit gates to decouple interactions, rescale them, and couple them again, resulting in long pulse sequences [14–18]. Recently, there has been a renewed interest in designing pulse sequences to change the effective dynamics governed by a given Hamiltonian. In particular, there has been impressive progress in the design of robust global pulses, identical pulses on each qubit, to change the global properties of a given Hamiltonian [19]. This approach has already been generalized to qudit systems [20, 21] and implemented in experiments with ultracold atomic Rydberg gas and NV centers in diamond [22, 23]. In our work, we design ro-

bust sequences of local pulses to change any interaction coupling in a given Hamiltonian, with similar robustness properties as for global pulses [19]. However, due to the local pulses in our method we are able to implement the dynamics of a much larger family of target Hamiltonians.

So far, other approaches utilizing local pulses have certain limitations. They either require NP-hard classical pre-processing to find the pulse sequence [24], rely on specific structures in the system Hamiltonian [25], or require an infinite single-qubit gate set which might be a problem for the fast control electronics in an experiment [26–28]. These works are also limited to two-body interactions, and to the best of our knowledge, no general scheme has been developed for efficiently engineering individual many-body interactions. Since there has been an increasing effort in realizing the latter in experiments [4, 29–32], such a general scheme would allow to simulate quantum chemistry Hamiltonians with genuine many-body interactions. Moreover, some recent Hamiltonian learning schemes rely on “reshaping” unknown many-body Hamiltonians to diagonal Hamiltonians which can be efficiently done with our proposed method [33–35].

In previous works, we have proposed a Hamiltonian engineering method for Ising Hamiltonians based on linear programming and applied it to multi-qubit gate synthesis and compiling problems [36, 37]. A related linear program approach for commuting Ising and next-neighbour interactions with an efficient relaxation was introduced in Refs. [38, 39].

In this work, we generalize the linear programming method to a significantly larger family of local Hamiltonians and make it substantially more efficient. More concretely, our method allows to efficiently engineer arbitrary target Hamiltonians, the interaction graph of which agrees with the one of the system Hamiltonian. To this

* bassler@hhu.de

† martin.kliesch@tuhh.de

‡ markus.heinrich@uni-koeln.de

end, free evolutions under the system Hamiltonian are interleaved with Pauli or single-qubit Clifford gates. A crucial feature of our method is that it minimizes the total evolution time, leading to a fast implementation of the target Hamiltonian. For larger systems, optimally solving the minimizing linear program is, however, no longer efficiently possible. To this end, we introduce a relaxation that only scales with the number of interaction terms and not directly with the system size, and thus, provides an efficient method to engineer Hamiltonians. This relaxation still yields an exact decomposition of the target Hamiltonian, but the total evolution time may no longer be minimal. The latter can, however, be decreased by expanding the search space for the relaxed problem, providing a trade-off between the runtime of the classical pre-processing and the evolution time of the implementation. For r interactions in the interaction graph of the system Hamiltonian (r free parameters), we propose an efficient method for finding the single-qubit gate sequence which only scales linear with r , i.e. not directly dependent on the number of qubits. To implement the target evolution with a product formula, we therefore only require $2Cr$ single-qubit gate layers, where C is a constant only depending on the chosen product formula. Note that similar previous approaches require $O(r^2)$ single-qubit gate layers [14–18] or are classically hard to solve [24]. We also observe that the total evolution time to implement the dynamics under the target Hamiltonian scales only sublinear with the number of qubits.

We also investigate the effects of dominant experimental error sources and introduce a general framework to cancel them. To this end, we generalize a method by Votto *et al.* [18] based on average Hamiltonian theory (AHT) to mitigate these errors. As a result, we are able to make our methods robust against finite pulse time and single-qubit rotation angle errors. We leverage the generality of our method to combine it with robust composite pulses, making it robust against many experimental error sources.

In section III we provide a general framework for engineering Hamiltonians using a linear program (LP). In sections III A and III B the efficient Hamiltonian engineering method by Pauli and Clifford conjugation are presented. In section IV we provide methods to investigate and mitigate the effect of errors on both, the Pauli and Clifford conjugation methods. Finally, in section V we show the efficiency and robustness of our methods in numerical simulations of a realistic setting.

II. PRELIMINARIES

We denote by P^n the set of n -qubit Pauli operators, which can be indexed by binary vectors $\mathbf{a} = (\mathbf{a}_x, \mathbf{a}_z) \in \mathbb{F}_2^{2n}$ such that

$$P_{\mathbf{a}} = P_{(\mathbf{a}_x, \mathbf{a}_z)} = i^{\mathbf{a}_x \cdot \mathbf{a}_z} X(\mathbf{a}_x) Z(\mathbf{a}_z). \quad (1)$$

Here, $X(\mathbf{x}) := X^{x_1} \otimes \cdots \otimes X^{x_n}$ and $Z(\mathbf{z}) := Z^{z_1} \otimes \cdots \otimes Z^{z_n}$ where X, Z are the single-qubit Pauli matrices. Sometimes, we also call $P_{\mathbf{a}}$ a *Pauli string*. They satisfy the following commutator relation

$$P_{\mathbf{a}} P_{\mathbf{b}} = (-1)^{\langle \mathbf{a}, \mathbf{b} \rangle} P_{\mathbf{b}} P_{\mathbf{a}}, \quad (2)$$

for any $\mathbf{a}, \mathbf{b} \in \mathbb{F}_2^{2n}$ with the binary symplectic form on \mathbb{F}_2^{2n} defined by $\langle \mathbf{a}, \mathbf{b} \rangle := \mathbf{a}_x \cdot \mathbf{b}_z + \mathbf{a}_z \cdot \mathbf{b}_x$, where the sum is taken in \mathbb{F}_2 , i.e., modulo 2. We refer to the identity as $I \in \mathbb{C}^{2 \times 2}$ and Pauli matrices as $X, Y, Z \in \mathbb{C}^{2 \times 2}$. The identity operator on n qubits is denoted by $\mathbb{1}_n \in \mathbb{C}^{2^n \times 2^n}$, and often we write $\mathbb{1}$ instead of $\mathbb{1}_n$.

The set P^n forms a basis for the complex matrices, i.e. any $H \in \mathbb{C}^{2^n \times 2^n}$ can be written as

$$H = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n}} J_{\mathbf{a}} P_{\mathbf{a}}, \quad (3)$$

with $J_{\mathbf{a}} \in \mathbb{C}$. If H is a Hermitian operator, which we denote by $H \in \text{Herm}(\mathbb{C}^{2^n})$, then $J_{\mathbf{a}} \in \mathbb{R}$. We collect all $J_{\mathbf{a}}$ in a vector $\mathbf{J} \in \mathbb{R}^{4^n}$ with indices $\mathbf{a} \in \mathbb{F}_2^{2n}$. We define the *support* of a Pauli string $P_{\mathbf{a}}$ as $\text{supp}(P_{\mathbf{a}}) := \text{supp}(\mathbf{a}) := \{i \in [n] | a_{x_i} \neq 0 \text{ or } a_{z_i} \neq 0\}$, i.e. the qubit indices on which $P_{\mathbf{a}}$ acts non-trivially. For any constant k , we call the Hamiltonian

$$H = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n}} J_{\mathbf{a}} P_{\mathbf{a}}. \quad (4)$$

k -local if $J_{\mathbf{a}} = 0$ whenever $|\text{supp}(\mathbf{a})| > k$. If H is 2-local, its *interaction graph* is the graph with the sites/qubits as vertices and two vertices are connected by an edge iff H has a corresponding interaction term. For higher localities, edges are replaced by hyperedges given by the supports of the $P_{\mathbf{a}}$, resulting in an *interaction hypergraph*.

III. HAMILTONIAN ENGINEERING BY LINEAR PROGRAMMING

We wish to implement the time evolution of a *target Hamiltonian* H_T by having access to some simple gates (e.g., certain single-qubit gates) plus the time evolution under some *system Hamiltonian* H_S . The system Hamiltonian can be an arbitrary Hamiltonian that is native to the considered quantum platform. That is, we wish to change H_S to achieve an effective Hamiltonian H_T . To this end, we write H_S and H_T in their Pauli decompositions

$$H_S = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} J_{\mathbf{a}} P_{\mathbf{a}}, \quad H_T = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} A_{\mathbf{a}} P_{\mathbf{a}}, \quad (5)$$

with $J_{\mathbf{a}}, A_{\mathbf{a}} \in \mathbb{R}$. We exclude the term $P_{\mathbf{0}} = P_{(0, \dots, 0)} = I^{\otimes n}$, leading to a global and thus unobservable phase. Our goal is to simulate the time evolution under H_T by the one under H_S interleaved with layers of single-qubit gates. Recall that we have the identity $e^{-itU^\dagger H U} =$

$U^\dagger e^{-itH} U$ for any Hamiltonian H and unitary U . Hence, we seek a decomposition of the form

$$H_T = \sum_{i=1}^L \lambda_i \mathbf{S}_i^\dagger H_S \mathbf{S}_i, \quad (6)$$

with $\lambda_i > 0$ and $\mathbf{S}_i = S_i^{(1)} \otimes \cdots \otimes S_i^{(n)}$ representing a layer of single-qubit gates, which we later set to be either Pauli or Clifford gates. Indeed, if all terms in eq. (6) mutually commute, then the time evolution under H_T for time t can be implemented exactly by the time evolution under H_S interleaved with layers of single-qubit gates $\mathbf{S}_i \mathbf{S}_{i-1}^\dagger$ at times $t' = t \sum_{j=1}^{i-1} \lambda_j$ for $i \geq 1$ (setting $\mathbf{S}_0 = \mathbf{S}_{L+1} = \mathbb{1}$). In this way, the time evolution under Hamiltonians $\mathbf{S}_i^\dagger H_S \mathbf{S}_i$ is effectively implemented for consecutive times $t\lambda_i$ with $i = 1, 2, \dots$ and leads to the overall time evolution under H_T . If the terms in eq. (6) do not commute, then the time evolution under H_T can be approximated in a similar fashion as for the commuting terms, utilizing suitable Hamiltonian simulation methods (e.g. product formulas).

The suitability of this approach depends highly on the decomposition (6). We base our Hamiltonian engineering method on linear programming, which allows us to both minimize the total evolution time and reduce the number of terms, i.e. layers, in Eq. (6). To this end, we consider the effect of $H_S \mapsto \mathbf{S}_i^\dagger H_S \mathbf{S}_i$ on the Pauli coefficients J_a which can be captured by a column vector $\mathcal{W}_i \in \mathbb{R}^r$ with elements

$$\mathcal{W}_{a,i} = \frac{1}{2^n} \text{Tr} \left(P_a \left(\mathbf{S}_i^\dagger H_S \mathbf{S}_i \right) \right). \quad (7)$$

Here, r is the number of Pauli terms $\{P_a\}$ we can generate from the ones in H_S by any \mathbf{S}_i . Let s be the number of possible single-qubit layers on n qubits, and define the matrix $\mathcal{W} \in \mathbb{R}^{r \times s}$ by taking all vectors \mathcal{W}_i as columns. Then, a decomposition (6) can be found by solving the following LP minimizing the total *relative evolution time*:

$$\begin{aligned} & \text{minimize} \quad \mathbf{1}^T \boldsymbol{\lambda} \\ & \text{subject to} \quad \mathcal{W} \boldsymbol{\lambda} = \mathbf{A}, \quad \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^s, \end{aligned} \quad (\text{LP})$$

where $\mathbf{1} = (1, 1, \dots, 1)$ is the all-ones vector such that $\mathbf{1}^T \boldsymbol{\lambda} = \sum_i \lambda_i$ and $\mathbf{A} \in \mathbb{R}^r$ represents the Pauli coefficients of the target Hamiltonian H_T . Conjugation of H_S with arbitrary single-qubit gates yields a r which is limited by the interaction graph of H_S .

(LP) is central for our approach to Hamiltonian engineering and its analysis plays an important role in this work. For this purpose, we require some basic notions and properties from the theory of linear programming [40] which we will briefly introduce in the following. A *feasible solution* $\boldsymbol{\lambda}$ is a vector which satisfies all constraints of (LP). An *optimal solution* is a feasible solution that also minimizes the objective function and is indicated by an asterisk as $\boldsymbol{\lambda}^*$. If (LP) has a feasible solution, then there also exists a r -sparse optimal solution

$\boldsymbol{\lambda}^*$ [41], corresponding to a decomposition as in eq. (6) with only r terms. Such a r -sparse optimal solution can be found using the simplex algorithm which, in practice, has a runtime that scales polynomial in the problem size $r \times s$ [42, 43].

The first main contribution of our work is a hierarchy of relaxations of (LP) with drastically reduced size that still allows for an exact decomposition as in eq. (6). For this, we need general sufficient conditions on the matrix \mathcal{W} such that (LP) has a feasible solution for any $\mathbf{A} \in \mathbb{R}^r$.

Definition III.1. We say that a matrix $\mathcal{W} \in \mathbb{R}^{r \times s}$ is feasible if for each $\mathbf{A} \in \mathbb{R}^r$ there exists a $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^s$ such that $\mathcal{W} \boldsymbol{\lambda} = \mathbf{A}$.

This definition captures the constraints in (LP). There is a simple sufficient condition for the feasibility of the matrix \mathcal{W} :

Proposition III.2. Let $\mathcal{W} \in \mathbb{R}^{r \times s}$ such that $\ker(\mathcal{W}^T) = \{\mathbf{0}\}$. If there exists $\mathbf{x} \in \mathbb{R}^s$ such that $\mathcal{W} \mathbf{x} = \mathbf{0}$ and $\mathbf{x} > \mathbf{0}$, then for any $\mathbf{A} \in \mathbb{R}^r$ there exists $\mathbf{x}' \in \mathbb{R}^s$ such that $\mathcal{W} \mathbf{x}' = \mathbf{A}$ and $\mathbf{x}' \geq \mathbf{0}$.

This proposition follows directly from two well-known results in convex analysis.

Lemma III.3 (Farkas [44]). Let $\mathcal{W} \in \mathbb{R}^{r \times s}$ and $\mathbf{A} \in \mathbb{R}^r$. Then exactly one of the following assertions is true:

1. $\exists \mathbf{x} \in \mathbb{R}^s$ such that $\mathcal{W} \mathbf{x} = \mathbf{A}$ and $\mathbf{x} \geq \mathbf{0}$.
2. $\exists \mathbf{y} \in \mathbb{R}^r$ such that $\mathcal{W}^T \mathbf{y} \geq \mathbf{0}$ and $\mathbf{A}^T \mathbf{y} < \mathbf{0}$.

Lemma III.4 (Stiemke [45]). Let $\mathcal{W} \in \mathbb{R}^{r \times s}$. Then exactly one of the following assertions is true:

1. $\exists \mathbf{x} \in \mathbb{R}^s$ such that $\mathcal{W} \mathbf{x} = \mathbf{0}$ and $\mathbf{x} > \mathbf{0}$.
2. $\exists \mathbf{y} \in \mathbb{R}^r$ such that $\mathcal{W}^T \mathbf{y} \succeq \mathbf{0}$.

Proof of Prop. III.2. We show that the second assertion of Farkas lemma is not possible. By lemma III.4, there does not exist a \mathbf{y} such that $\mathcal{W}^T \mathbf{y} \succeq \mathbf{0}$. From $\ker(\mathcal{W}^T) = \{\mathbf{0}\}$ it follows that if $\mathcal{W}^T \mathbf{y} = \mathbf{0}$, then $\mathbf{y} = \mathbf{0}$ which contradicts $\mathbf{A}^T \mathbf{y} < \mathbf{0}$ in the second assertion of lemma III.3. Finally, the case $\mathcal{W}^T \mathbf{y} < \mathbf{0}$ directly contradicts the second assertion of lemma III.3. \square

To provide a geometric interpretation of proposition III.2 we first define the *convex hull of the column vectors* of \mathcal{W}

$$\text{conv}(\mathcal{W}) := \left\{ \mathbf{u} \in \mathbb{R}^r \mid \mathcal{W} \mathbf{x} = \mathbf{u}, \mathbf{x} \geq \mathbf{0}, \sum_i x_i = 1 \right\}, \quad (8)$$

and the *interior of a polytope* P

$$\begin{aligned} \text{int}(P) &:= \\ & \left\{ \mathbf{u} \in P \mid \exists \varepsilon > 0 \text{ s.t. } \|\mathbf{u} - \mathbf{x}\| < \varepsilon \quad \forall \mathbf{x} \in \mathbb{R}^r \Rightarrow \mathbf{x} \in P \right\}. \end{aligned} \quad (9)$$

(LP) is feasible if the convex hull of the column vectors of \mathcal{W} has an interior that contains the origin, $\mathbf{0} \in$

$\text{int}(\text{conv}(\mathcal{W}))$. The conditions of proposition III.2 can be efficiently verified: If the LP

$$\begin{aligned} & \text{minimize} && \mathbf{1}^T \mathbf{x} \\ & \text{subject to} && \mathcal{W}\mathbf{x} = \mathbf{0}, \\ & && \mathbf{x} \geq \mathbf{1} \end{aligned} \quad (10)$$

has a feasible solution and \mathcal{W} has full rank, then \mathcal{W} is feasible. Note that proposition III.2 applies to general matrices and will be useful for the efficient relaxations.

Numerical implementation All LPs and mixed integer linear programs (MILPs) in our work are modelled with the Python package CVXPY [46, 47] and solved with the MOSEK solver [48] on a workstation with 130 GB RAM and the AMD Ryzen Threadripper PRO 3975WX 32-Cores processor. Furthermore, we made the code to reproduce all figures in this work publicly available on GitHub [49].

A. Pauli conjugation

Conjugation of a system Hamiltonian H_S with a Pauli string P_b leads to

$$P_b^\dagger H_S P_b = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} (-1)^{\langle \mathbf{a}, \mathbf{b} \rangle} J_{\mathbf{a}} P_{\mathbf{a}}, \quad (11)$$

which follows from the commutation relations (2). We seek a decomposition of H_T of the form

$$\begin{aligned} H_T &= \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} A_{\mathbf{a}} P_{\mathbf{a}} \\ &= \sum_{\mathbf{b} \in \mathbb{F}_2^{2n}} \lambda_{\mathbf{b}} P_b^\dagger H_S P_b \\ &= \sum_{\mathbf{a} \neq \mathbf{0}, \mathbf{b}} \lambda_{\mathbf{b}} (-1)^{\langle \mathbf{a}, \mathbf{b} \rangle} J_{\mathbf{a}} P_{\mathbf{a}}, \quad (\lambda_{\mathbf{b}} \geq 0). \end{aligned} \quad (12)$$

The $4^n \times 4^n$ matrix with entries $W_{\mathbf{ab}} = (-1)^{\langle \mathbf{a}, \mathbf{b} \rangle}$ is called the *Walsh-Hadamard matrix*. Here, we only need a sub-matrix $W^{(r \times s)} \in \{-1, 1\}^{r \times s}$ defined by choosing r row-indices $\mathbf{a} \in \mathbb{F}_2^{2n}$ and s column-indices $\mathbf{b} \in \mathbb{F}_2^{2n}$. We call $W^{(r \times s)}$ a *partial Walsh-Hadamard matrix*. By eq. (12), it is clear that only the Pauli terms with $J_{\mathbf{a}} \neq 0$ contribute to the target Hamiltonian H_T . Therefore, we define $\text{nz}(\mathbf{J}) := \{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\} \mid J_{\mathbf{a}} \neq 0\}$, and require that the Pauli coefficients satisfy

$$\text{nz}(\mathbf{A}) \subseteq \text{nz}(\mathbf{J}). \quad (13)$$

This requirement can be eliminated with single-qubit Clifford conjugation, which we will discuss in detail in section III B. In addition to the restriction $\mathbf{a} \in \text{nz}(\mathbf{J})$ given by the system Hamiltonian, we may also restrict the set of Pauli strings $\mathbf{b} \in \mathcal{F} \subseteq \mathbb{F}_2^{2n}$, as long as there is still a solution to eq. (12). Comparing the Pauli coefficients in eq. (12), we can write this as

$$A_{\mathbf{a}} = J_{\mathbf{a}} \sum_{\mathbf{b} \in \mathcal{F}} W_{\mathbf{ab}}^{(r \times s)} \lambda_{\mathbf{b}} \quad \forall \mathbf{a} \in \text{nz}(\mathbf{J}), \quad (14)$$

with the number of non-zero Pauli coefficients $r := |\text{nz}(\mathbf{J})|$ and $s = |\mathcal{F}|$. Hence, the constraint in (LP) becomes $\mathbf{A} = \mathbf{J} \odot (W^{(r \times s)} \boldsymbol{\lambda})$, where \odot denotes element-wise multiplication and $\mathbf{A}, \mathbf{J} \in \mathbb{R}^r$ are restricted to $\mathbf{a} \in \text{nz}(\mathbf{J})$. For the following analysis, it is convenient to set $\mathbf{M} := \mathbf{A} \oslash \mathbf{J}$ using the element-wise division $(\mathbf{A} \oslash \mathbf{J})_{\mathbf{a}} := A_{\mathbf{a}} / J_{\mathbf{a}}$ for all $\mathbf{a} \in \text{nz}(\mathbf{J})$. This results in the following LP:

$$\begin{aligned} & \text{minimize} && \mathbf{1}^T \boldsymbol{\lambda} \\ & \text{subject to} && W^{(r \times s)} \boldsymbol{\lambda} = \mathbf{M}, \quad \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^s. \end{aligned} \quad (\text{PauliLP})$$

The runtime of a LP scales with its size, i.e. the number of constraints r and the number of variables s . We discuss in detail the existence of a solution $\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^s$ and bounds on the total relative evolution time $\mathbf{1}^T \boldsymbol{\lambda}$ in appendix A. Here, $r = |\text{nz}(\mathbf{J})| \leq 4^n - 1$ is fixed by the system Hamiltonian and a priori $s = 4^n$, thus solving (PauliLP) becomes computationally intensive already at moderate system size. To overcome this, we propose a simple and efficient relaxation of (PauliLP) in section III A 1: For any $r = |\text{nz}(\mathbf{J})|$, we sample $s \geq 2r$ Pauli strings at random, leading to a feasible LP with high probability. As we have $r = \text{poly}(n)$ for local Hamiltonians, this indeed yields an efficient relaxation in the system size n . Note that this relaxation still leads to an exact decomposition; however, the relative evolution time $\mathbf{1}^T \boldsymbol{\lambda}$ may not be minimal anymore.

1. Efficient relaxation

We provide an efficient method to construct a feasible matrix $W^{(r \times s)}$ with $r = |\text{nz}(\mathbf{J})|$ rows and $s \geq 2r$ columns. The construction is rather simple and consists of sampling s vectors $\mathbf{b} \in \mathbb{F}_2^{2n}$ uniformly at random and taking the corresponding partial Walsh-Hadamard matrix with entries $W_{\mathbf{ab}}^{(r \times s)} = (-1)^{\langle \mathbf{a}, \mathbf{b} \rangle}$ where $\mathbf{a} \in \text{nz}(\mathbf{J})$. Thus, engineering a Hamiltonian with r non-zero Pauli terms leads to a (PauliLP), whose number of constraints and variables both scale linearly with r .

To ensure feasibility of the sub-sampled matrix, we invoke proposition III.2. Thus, we have to check that $\text{conv}(W^{(r \times s)})$ has a non-empty interior and $\mathbf{0} \in \text{conv}(W^{(r \times s)})$. First, we state general results for i.i.d. copies $\mathbf{x}_1, \dots, \mathbf{x}_s$ of a random vector \mathbf{x} . Then, we relate the results to the partial Walsh-Hadamard matrix $W^{(r \times s)}$. Here, we assume that s is large enough and that $\mathbf{x}_1, \dots, \mathbf{x}_s$ are in general position such that $\text{conv}(\mathbf{x}_1, \dots, \mathbf{x}_s)$ always has non-empty interior. Thus, we focus on the condition $\mathbf{0} \in \text{conv}(\mathbf{x}_1, \dots, \mathbf{x}_s)$.

Definition III.5. Let $\mathbf{x}_1, \dots, \mathbf{x}_s$ be i.i.d. copies of an arbitrary random vector \mathbf{x} in \mathbb{R}^r and define

$$p_{s, \mathbf{x}} := \mathbb{P}(\mathbf{0} \in \text{conv}(\mathbf{x}_1, \dots, \mathbf{x}_s)). \quad (15)$$

Intuitively, one would expect that the probability should increase quickly with the number of samples s , and taking s of the order of r should make $p_{s, \mathbf{x}}$ reasonably large.

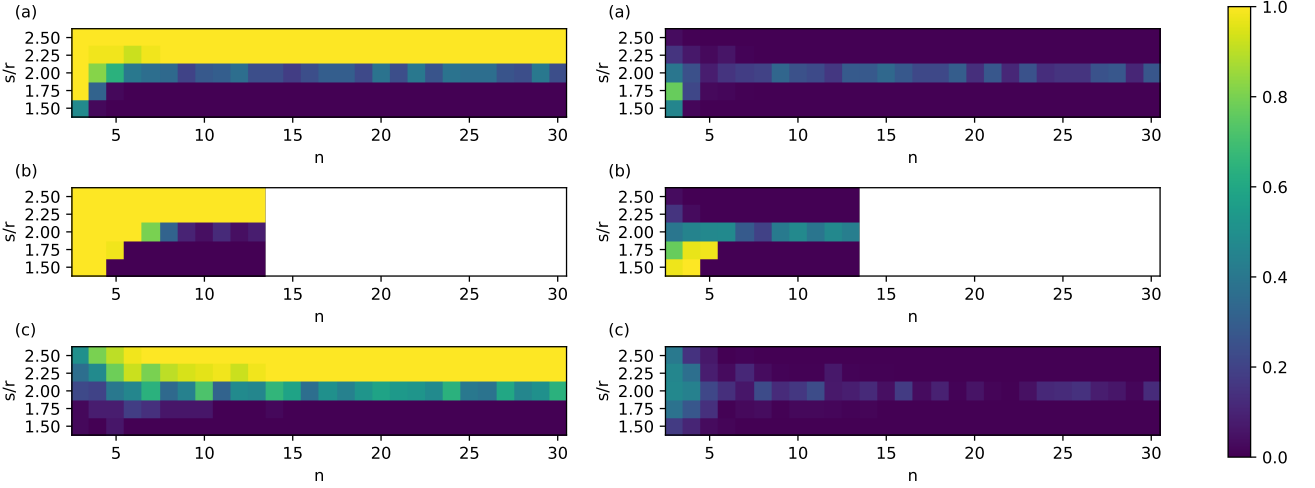


Figure 1. Select r rows of the Walsh-Hadamard matrix to obtain the partial Walsh-Hadamard matrix $W^{(r \times 4^n)}$, then randomly sample s columns to obtain $W^{(r \times s)}$. **(a)** and **(b)**: For a k -local system Hamiltonian with $k \leq 2$ and $k \leq 3$ respectively, resulting in $r = \sum_{i=1}^k 3^i \binom{n}{i}$ Pauli terms. **(b)**: We stopped the numerical experiments at $n = 13$ due to the large amount of all-to-all interactions with support $k \leq 3$. **(c)**: For random Hamiltonians with $r = n^2$ Pauli terms selected uniformly at random. **Left**: The plots show the success frequency for “ $W^{(r \times s)}$ is feasible” over 50 samples (yellow) for $n = 3, \dots, 30$ and $s/r = 1.5, \dots, 2.5$. There is a sharp transition at $s/r = 2$ in all cases. **Right**: The difference of our numerical observation to Wendel’s formula (17).

Lemma III.6 ([50], Proposition 4). *Let $\mathbf{x} \in \mathbb{R}^r$ be an arbitrary random vector with $\mathbb{E}[\mathbf{x}] = 0$ and $\mathbb{P}(\mathbf{x} \neq 0) > 0$. Then we have*

$$0 < p_{r+1, \mathbf{x}} < p_{r+2, \mathbf{x}} < \dots < p_{r+l, \mathbf{x}} \rightarrow 1 \quad \text{for } l \rightarrow \infty, \quad (16)$$

and $p_{s, \mathbf{x}} = 0$ if $s \leq r$.

Indeed, letting \mathbf{w} be the r -dimensional random vector drawn uniformly from the columns of the partial Walsh-Hadamard matrix $W^{(r \times 4^n)}$, we can readily verify $\mathbb{E}[\mathbf{w}] = 0$ and thus lemma III.6 applies. However, the question of how large s shall be taken still remains. In the convex geometry literature, we find an elegant solution, at least for spherically symmetric random vectors, in the form of **Wendel’s theorem** [51]: If the random vector \mathbf{x} has a spherically symmetric distribution around $\mathbf{0}$, then the probability (15) is given by

$$p_{s, \mathbf{x}} = 1 - \frac{1}{2^{s-1}} \sum_{k=0}^{r-1} \binom{s-1}{k}. \quad (17)$$

This distribution shows a sharp transition from ≈ 0 to ≈ 1 at $s = 2r$. Wendel’s theorem does however not apply to the random columns \mathbf{w} of $W^{(r \times 4^n)}$ because of the lack of spherical symmetry (see appendix B for details). Instead, the result (17) of Wendel’s theorem provides an upper bound on the probabilities $p_{s, \mathbf{w}}$ [50, 52], and adequate lower bounds prove to be tricky to derive.

Nevertheless, we numerically observe that the random columns \mathbf{w} of $W^{(r \times 4^n)}$ follow the behavior that we would expect from Wendel’s theorem: We observe a sharp transition of $p_{s, \mathbf{w}}$ at the same position as for spherical symmetric distributions, approximating the upper

bound (17). In this sense, $p_{s, \mathbf{w}}$ is optimal, maximizing the success probability for finding a feasible $W^{(r \times s)}$.

Numerical Observation III.7 (figure 1). *Let $W^{(r \times 4^n)}$ be a partial Walsh-Hadamard matrix with $r = |\text{nz}(\mathbf{J})|$, and let \mathbf{w} be a r -dimensional random vector drawn uniformly from $\text{col}(W^{(r \times 4^n)})$. Then, we numerically observe that Wendel’s statement (17) approximately holds, i.e. the random submatrix $W^{(r \times s)}$ is feasible with high probability provided we take $s \geq 2r$ samples of \mathbf{w} .*

Let $W^{(r \times s)}$ be the partial Walsh-Hadamard matrix obtained from $W^{(r \times 4^n)}$ by randomly sampling s columns $\mathbf{b} \stackrel{\text{i.i.d.}}{\sim} \text{unif}(\mathbb{F}_2^{2n})$ with $s \geq 2r$. As $W^{(r \times s)}$ is feasible with high probability, it can be used in (PauliLP) to engineer any target Hamiltonian with a Pauli decomposition compatible with the system Hamiltonian, i.e. with $\text{nz}(\mathbf{A}) \subseteq \text{nz}(\mathbf{J})$. One $W^{(r \times s)}$ can be reused for different Hamiltonians with a Pauli decomposition with r terms, since the construction of $W^{(r \times s)}$ is independent of the choice of $\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}$. As $r = \text{poly}(n)$ for local Hamiltonians, the relaxed (PauliLP) can be solved efficiently, and it provides an exact decomposition of the target Hamiltonian. The evolution time $\mathbf{1}^T \boldsymbol{\lambda}$ may however not be minimal anymore. The quality of the relaxation can be improved by increasing s , thus expanding the search space. This provides a flexible trade-off between the runtime of (PauliLP) and the optimality of $\boldsymbol{\lambda}$, see figure 2. The number of interactions r in figure 2 grows quadratic for (a) and (c) and even cubic for (b) with the number of qubits n . However, $\sum \lambda = \mathbf{1}^T \boldsymbol{\lambda}$ grows only linear with n .

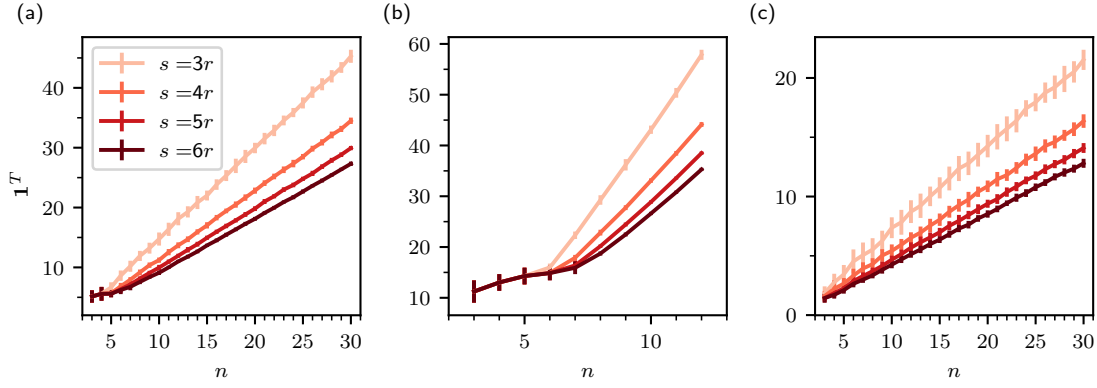


Figure 2. Solving (PauliLP) for 50 uniform random samples $\mathbf{M} = \mathbf{A} \oslash \mathbf{J}$ from $[-1, 1]^r$. For each sample a feasible $W^{(r \times s)}$ is generated as in numerical Observation III.7 with $s = 3r, \dots, 6r$. The average objective function value $\sum \lambda = \mathbf{1}^T \lambda$ over the number of qubits n is shown. The error bars represent the sample standard deviation. The r interactions of the plots (a), (b) and (c) are chosen the same way as in figure 1.

B. Clifford conjugation

In section III A, we introduced the Hamiltonian engineering method based on conjugation with Pauli gates. In this section, we extend this method to a certain set of single-qubit Clifford gates. This extension allows us to change the Pauli terms in the system Hamiltonian, such the only restriction in engineering a target Hamiltonian comes from the interaction (hyper)graph of the system Hamiltonian H_S . The same restriction also holds for conjugation with arbitrary single-qubit gates, making Clifford conjugation a powerful Hamiltonian engineering method. The simplest set of single-qubit Clifford gates with full expressivity are the square-root Pauli gates ($\frac{\pi}{2}$ rotations)

$$\begin{aligned} \sqrt{X} &= \frac{1}{2} \begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}, \quad \sqrt{Y} = \frac{1}{2} \begin{pmatrix} 1+i & -1-i \\ 1+i & 1+i \end{pmatrix}, \\ \sqrt{Z} &= \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \end{aligned} \quad (18)$$

$S^\dagger P S$	\sqrt{X}	\sqrt{Y}	\sqrt{Z}	\sqrt{X}^\dagger	\sqrt{Y}^\dagger	\sqrt{Z}^\dagger
X	X	Z	$-Y$	X	$-Z$	Y
Y	$-Z$	Y	X	Z	Y	$-X$
Z	Y	$-X$	Z	$-Y$	X	Z

Table I. Conjugation of Pauli gates $P \in \mathbf{P}$ with square-root Pauli gates $S \in \{\sqrt{X}, \sqrt{Y}, \sqrt{Z}, \sqrt{X}^\dagger, \sqrt{Y}^\dagger, \sqrt{Z}^\dagger\}$.

As displayed in table I, the conjugation of a Pauli gate with a square-root Pauli gate does not only flip the sign but also changes the interaction type (rotation axis). The change of the interaction type increases the set of Hamiltonians reachable by Clifford conjugation compared to the Pauli conjugation. In the following, we consider a gate set that behaves very similar to the

conjugation with Pauli gates:

$$\mathcal{C}_{XY} := \{Z\} \cup \left\{ QD \mid Q, D \in \{\sqrt{X}, \sqrt{Y}, \sqrt{X}^\dagger, \sqrt{Y}^\dagger\} \right\}. \quad (19)$$

Likewise, we can define gate sets \mathcal{C}_{ZY} and \mathcal{C}_{XZ} , for which conclusions analogous to the following ones may be drawn. Conjugation of Pauli terms with the Clifford gates $\sqrt{X}\sqrt{Y}$ or $\sqrt{Y}^\dagger\sqrt{X}^\dagger$ changes the interaction type and leaves the sign of the Pauli coefficient unchanged. The signs of the conjugated Pauli terms depend on the rotation direction, i.e. the placement of “ \dagger ”, of the square-root Pauli gates; see table II for examples. Therefore, we label an element in \mathcal{C}_{XY} by $c := (p, \mathbf{b}) \in \mathbb{F}_3 \times \mathbb{F}_2^2$, where $p \in \mathbb{F}_3$ represents the changes in the interaction type and $\mathbf{b} \in \mathbb{F}_2^2$ captures the sign flips similar to the Pauli conjugation. Denote $\mathcal{C}_{XY}^{\otimes n}$ a string of single-qubit gates on n qubits from the gate set \mathcal{C}_{XY} . We label each $S_c \in \mathcal{C}_{XY}^{\otimes n}$ with $\mathbf{c} = (c_1, \dots, c_n) = (\mathbf{p}, \mathbf{b}) \in \mathbb{F}_3^n \times \mathbb{F}_2^{2n}$, where $c_i \in \mathbb{F}_3 \times \mathbb{F}_2^2$ represents the single-qubit Clifford gate from \mathcal{C}_{XY} on the i -th qubit. In table II, we show the effect of conjugating a Pauli gate $P_a \in \mathbf{P}$ with $S_c \in \mathcal{C}_{XY}^{\otimes n}$. Conjugation of H_S with $S_c \in \mathcal{C}_{XY}^{\otimes n}$ leads to

$$\begin{aligned} S_c^\dagger H_S S_c &= \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{0\}} J_{\mathbf{a}} S_c^\dagger P_{\mathbf{a}} S_c \\ &= \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{0\}} (-1)^{\langle \mathbf{a}, \mathbf{b} \rangle} J_{\pi_{\mathbf{p}}(\mathbf{a})} P_{\mathbf{a}}, \end{aligned} \quad (20)$$

with $\mathbf{c} = (\mathbf{p}, \mathbf{b}) \in \mathbb{F}_3^n \times \mathbb{F}_2^{2n}$ and the permutation $\pi_{\mathbf{p}} : \mathbb{F}_2^{2n} \rightarrow \mathbb{F}_2^{2n}$ with $\pi_{\mathbf{p}}(\mathbf{a}) := (\pi_{p_1}(a_1), \dots, \pi_{p_n}(a_n))$ given by the local permutations $\pi_0, \pi_1, \pi_2 : \mathbb{F}_2^2 \rightarrow \mathbb{F}_2^2$ in the

$\mathbf{c} \backslash \mathbf{a}$		(a_x, a_z)	$P_{\mathbf{a}}$	$(0,0)$	$(1,0)$	$(1,1)$	$(0,1)$
p	\mathbf{b}	S_c	$P_{\mathbf{a}}$	I	X	Y	Z
0	$(0,0)$	I	I	I	X	Y	Z
	$(1,0)$	X	I	X	$-Y$	$-Z$	
	$(1,1)$	Y	I	$-X$	Y	$-Z$	
	$(0,1)$	Z	I	$-X$	$-Y$	Z	
1	$(0,0)$	$\sqrt{X}\sqrt{Y}$	I	Z	X	Y	
	$(1,0)$	$\sqrt{X}^\dagger\sqrt{Y}$	I	Z	$-X$	$-Y$	
	$(1,1)$	$\sqrt{X}^\dagger\sqrt{Y}^\dagger$	I	$-Z$	X	$-Y$	
	$(0,1)$	$\sqrt{X}\sqrt{Y}^\dagger$	I	$-Z$	$-X$	Y	
2	$(0,0)$	$\sqrt{Y}^\dagger\sqrt{X}^\dagger$	I	Y	Z	X	
	$(1,0)$	$\sqrt{Y}\sqrt{X}$	I	Y	$-Z$	$-X$	
	$(1,1)$	$\sqrt{Y}\sqrt{X}^\dagger$	I	$-Y$	Z	$-X$	
	$(0,1)$	$\sqrt{Y}^\dagger\sqrt{X}$	I	$-Y$	$-Z$	X	

Table II. The coloured cells show the conjugated Pauli gates $S_c^\dagger P_{\mathbf{a}} S_c$ for $P_{\mathbf{a}} \in \mathbf{P}$ conjugated by the single-qubit Clifford gate $S_c \in \mathcal{C}_{XY}$. We label an element in \mathcal{C}_{XY} by $c := (p, \mathbf{b}) \in \mathbb{F}_3 \times \mathbb{F}_2^2$, where $p \in \mathbb{F}_3$ captures the permutation of the conjugated Pauli terms, and \mathbf{b} captures the sign flips similar to the Pauli conjugation. This can be easily verified with table I.

two-line notation

$$\begin{aligned}
\pi_0 &:= \begin{pmatrix} (0,0) & (1,0) & (1,1) & (0,1) \\ (0,0) & (1,0) & (1,1) & (0,1) \end{pmatrix}, \\
\pi_1 &:= \begin{pmatrix} (0,0) & (1,0) & (1,1) & (0,1) \\ (0,0) & (1,1) & (0,1) & (1,0) \end{pmatrix}, \\
\pi_2 &:= \begin{pmatrix} (0,0) & (1,0) & (1,1) & (0,1) \\ (0,0) & (0,1) & (1,0) & (1,1) \end{pmatrix}.
\end{aligned} \tag{21}$$

We want to find a decomposition of H_T with $\lambda_c \geq 0$ such that

$$H_T = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} A_{\mathbf{a}} P_{\mathbf{a}} = \sum_{\mathbf{c} \in \mathbb{F}_3^n \times \mathbb{F}_2^{2n}} \lambda_c S_c^\dagger H_S S_c. \tag{22}$$

We define the matrix $W(\mathbf{J}) \in \mathbb{R}^{(4^n-1) \times 12^n}$ with entries

$$W(\mathbf{J})_{ac} := (-1)^{\langle \pi_{\mathbf{p}(\mathbf{a}), \mathbf{b}} \rangle} J_{\pi_{\mathbf{p}(\mathbf{a})}}, \tag{23}$$

for $\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}$, excluding the identity term $P_{\mathbf{a}} = I^{\otimes n}$ with $\mathbf{a} = \mathbf{0}$, and $\mathbf{c} \in \mathbb{F}_3^n \times \mathbb{F}_2^{2n}$. Similar to the Pauli conjugation we define the submatrix $W(\mathbf{J})^{(r \times s)} \in \mathbb{R}^{r \times s}$ with entries given in eq. (23) for r row-indices $\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}$ and s column-indices $\mathbf{c} \in \mathbb{F}_3^n \times \mathbb{F}_2^{2n}$. Up to the permutation of \mathbf{J} , the same Walsh-Hadamard matrix structure as in $W^{(r \times s)}$ from section III A is present in $W(\mathbf{J})^{(r \times s)}$. In terms of the matrix $W(\mathbf{J})$ it follows that

$$H_T = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} \sum_{\mathbf{c} \in \mathbb{F}_3^n \times \mathbb{F}_2^{2n}} \lambda_c W(\mathbf{J})_{ac} P_{\mathbf{a}}. \tag{24}$$

Due to the permutation in eq. (23) we are no longer restricted by the non-zero coefficients as in the Pauli conjugation. From eq. (24) it follows that for each

$A_{\mathbf{a}} \neq 0$ there has to be at least one $J_{\hat{\mathbf{a}}} \neq 0$ such that $\text{supp}(\hat{\mathbf{a}}) = \text{supp}(\mathbf{a})$. Therefore, we define

$$\text{suppnz}(\mathbf{J}) := \{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\} \mid \exists \hat{\mathbf{a}} \in \text{nz}(\mathbf{J}), \text{supp}(\hat{\mathbf{a}}) = \text{supp}(\mathbf{a})\}, \tag{25}$$

and require that the Pauli coefficients satisfy

$$\text{nz}(\mathbf{A}) \subseteq \text{suppnz}(\mathbf{J}). \tag{26}$$

In physical terms, this means that we are only restricted by the interaction graph of the system Hamiltonian H_S and have full flexibility in the kind of interactions and the interaction strength. In addition to the restriction $\mathbf{a} \in \text{suppnz}(\mathbf{J})$, given by the system Hamiltonian, we can also consider a restricted set of conjugating Clifford strings $\mathbf{c} \in \mathcal{F} \subseteq \mathbb{F}_3^n \times \mathbb{F}_2^{2n}$, as long as there still exists a solution. Then, the restricted eq. (24) can be reformulated as

$$A_{\mathbf{a}} = \sum_{\mathbf{c} \in \mathcal{F}} W(\mathbf{J})_{ac}^{(r \times s)} \lambda_c \quad \forall \mathbf{a} \in \text{suppnz}(\mathbf{J}), \tag{27}$$

with $r := |\text{suppnz}(\mathbf{J})|$ and $s = |\mathcal{F}|$. The constraint in (LP) then reads $\mathbf{A} = W(\mathbf{J})^{(r \times s)} \boldsymbol{\lambda}$, with $\mathbf{A}, \mathbf{J} \in \mathbb{R}^r$ restricted to $\mathbf{a} \in \text{suppnz}(\mathbf{J})$. With that, we define the following LP

$$\begin{aligned}
&\text{minimize} \quad \mathbf{1}^T \boldsymbol{\lambda} \\
&\text{subject to} \quad W(\mathbf{J})^{(r \times s)} \boldsymbol{\lambda} = \mathbf{A}, \quad \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^s.
\end{aligned} \tag{ClifLP}$$

There always exists a feasible solution of (ClifLP) with $W(\mathbf{J})^{(r \times 12^n)}$, which follows similarly to corollary A.1 from the Walsh-Hadamard structure in $W(\mathbf{J})^{(r \times 12^n)}$. In general, we have $s \leq 12^n$ and $r \leq 4^n - 1$. In the next section, we propose a simple and efficient relaxation of (ClifLP), where $s \geq 2r$ can be chosen for arbitrary $r = |\text{suppnz}(\mathbf{J})|$.

1. Efficient relaxation

The size of (ClifLP) scales as $O(12^n)$ if all possible Clifford layers are considered. Therefore, a relaxation is necessary to solve this LP in practice. Analogous to the Pauli conjugation method, section III A 1, sampling $s \geq 2r$ columns at random from $W(\mathbf{J})^{(r \times 12^n)}$ results in a feasible matrix $W(\mathbf{J})^{(r \times s)}$ with high probability. As in section III A 1, we test this statement numerically by checking the feasibility conditions of proposition III.2 for random submatrices.

Numerical Observation III.8 (figure 3). *Let $W(\mathbf{J})^{(r \times 12^n)}$ be a matrix with $r = |\text{suppnz}(\mathbf{J})|$ and entries as in eq. (23), and let \mathbf{w} be a r -dimensional random vector drawn uniformly from $\text{col}(W(\mathbf{J})^{(r \times 12^n)})$. Then, we numerically observe that Wendel's statement (17) approximately holds, i.e. the random submatrix $W(\mathbf{J})^{(r \times s)}$ is feasible with high probability provided we take $s \geq 2r$ samples of \mathbf{w} .*

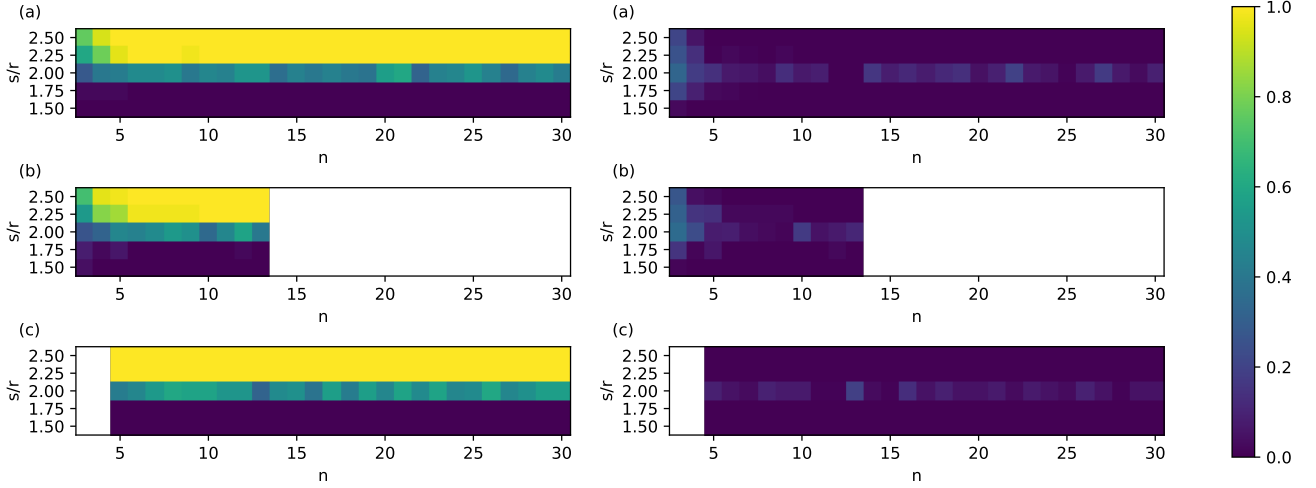


Figure 3. Select r rows of $W(\mathbf{J})$ to get the partial matrix $W(\mathbf{J})^{(r \times 12^n)}$, then randomly sample s columns to obtain $W(\mathbf{J})^{(r \times s)}$. **(a)** and **(b)**: For a k -local system Hamiltonian with $k \leq 2$ and $k \leq 3$ respectively, resulting in $r = \sum_{i=1}^k 3^i \binom{n}{i}$ Pauli terms. **(b)**: We stopped the numerical experiments at $n = 13$ due to the large amount of all-to-all interactions with support $k \leq 3$. **(c)**: For each $k = 1, \dots, 5$ choose randomly 10 different supports $\text{supp}(\mathbf{a})$ and consider all possible interactions on these supports. This results in $r \leq 10 \sum_{k=1}^5 3^k$ (“ \leq ”, since for $k = 1$ there are only n different supports). This represents a sparse random 5-local Hamiltonian. **(a)**, **(b)**, **(c)**: For a k -local interaction term $P_{\mathbf{a}}$ there exist 3^k distinct Pauli terms $P_{\mathbf{a}'}$ with the same support $\text{supp}(\mathbf{a}) = \text{supp}(\mathbf{a}')$, i.e. corresponding to the same (hyper)edge in the interaction graph. For each (hyper)edge the set of non-zero interactions is sampled randomly both in the interaction strength $J_{\mathbf{a}} \stackrel{\text{i.i.d.}}{\sim} \text{unif}([-1, +1])$ and the interaction type (index \mathbf{a} is chosen randomly). So, the amount, index, and values of the non-zeros in \mathbf{J} are random. **Left**: The plots show the success frequency over 50 samples for “ $W(\mathbf{J})^{(r \times s)}$ is feasible” (yellow) for each $n = 3, \dots, 30$ and $s/r = 1.5, \dots, 2.5$. There is a sharp transition at $s/r = 2$ in all cases. **Right**: The difference of our numerical observation from Wendel’s statement (17).

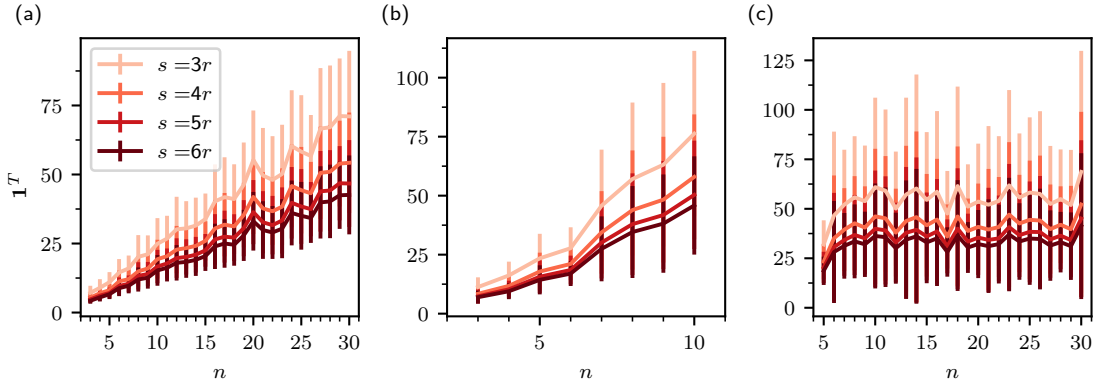


Figure 4. Solving (Clifflp) for 50 uniform random samples \mathbf{A} from $[-1, 1]^r$. For each sample a feasible $W(\mathbf{J})^{(r \times s)}$ is generated as in numerical Observation III.8 with $s = 3r, \dots, 6r$. The average objective function value $\sum \boldsymbol{\lambda} = \mathbf{1}^T \boldsymbol{\lambda}$ over the number of qubits n is shown. The error bars represent the sample standard deviation. The r interactions and \mathbf{J} of the plots (a), (b) and (c) are chosen the same way as in figure 3.

From the definition of feasibility, we know that (Clifflp) always has a solution for arbitrary \mathbf{A} , i.e. arbitrary \mathbf{J}, \mathbf{A} such that $\text{nz}(\mathbf{A}) \subseteq \text{supp}(\mathbf{J})$. Furthermore, the quality of the relaxation can be improved by increasing s , which leads to an expanded search space, see figure 4. This provides again a flexible trade-off between the runtime of (Clifflp) and the optimality of $\boldsymbol{\lambda}$. The scaling of $\mathbf{1}^T \boldsymbol{\lambda}$ in figure 4 (c) is independent of

n since $r \leq 10 \sum_{k=1}^5 3^k$ is independent of n . For the Clifford conjugation, the sample standard deviation is larger compared to the Pauli conjugation. This is due to the increased set of target Hamiltonians which are reachable by the Clifford conjugation. Here, we also randomly sample the number and the position of the non-zero values in \mathbf{J} .

C. Hamiltonian engineering with unknown Hamiltonians

In practice, sometimes not all coupling coefficients in \mathbf{J} might be known. For example, when an experiment aims to realize two-body interactions but also acquires unwanted three-body terms of unknown strength. Such system Hamiltonians with unknown or only partially known coupling strengths can still be used for engineering based on the Pauli conjugation method from section III A. Solving (PauliLP) for an $\mathbf{M} \in \mathbb{R}^r$ yields the target Hamiltonian

$$\sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{0\}} M_{\mathbf{a}} J_{\mathbf{a}} P_{\mathbf{a}} = H_T. \quad (28)$$

The potentially unknown coefficients $J_{\mathbf{a}}$ are modified by an element-wise multiplication $\mathbf{M} \odot \mathbf{J}$. Interesting choices for $M_{\mathbf{a}}$ are -1 and 0 , inverting the sign of or canceling the term $P_{\mathbf{a}}$, respectively, without the knowledge of $J_{\mathbf{a}}$. For each term in the system Hamiltonian, a different $M_{\mathbf{a}}$ can be chosen. Therefore, engineering known terms $M_{\mathbf{a}} = A_{\mathbf{a}}/J_{\mathbf{a}}$ in the Hamiltonian while canceling or inverting other (potentially unknown) terms $M_{\mathbf{a}} = 0$ or $M_{\mathbf{a}} = -1$ is possible. Using such a \mathbf{M} in (PauliLP) with dummy values for the unknown $J_{\mathbf{a}}$ inverts the signs or cancels the interactions.

In section V A we apply the approach to cancel unknown 3-body terms in a 2D lattice Hamiltonian.

D. Hamiltonian engineering with 2D lattice Hamiltonian

The scaling of our method is independent of the system size and only depends on the number of interactions. To highlight this fact, we consider a system Hamiltonian with a 2D square lattice interaction graph. Our Hamiltonian with constant couplings between neighboring qubits is motivated by experimental quantum platforms such as interacting superconducting qubits or cold atoms in optical lattices [53–55].

We consider a system and target Hamiltonian

$$H_S = J \sum_{i,j \in E} P_{\mathbf{a}_i} P_{\mathbf{a}_j}, \quad H_T = \sum_{i,j \in E} A_{\mathbf{a}} P_{\mathbf{a}_i} P_{\mathbf{a}_j}, \quad (29)$$

with $\text{supp}(\mathbf{a}) \in E$, representing all possible two-body interactions on a 2D square lattice interaction graph with the set of edges E , and constant interaction strength in the system Hamiltonian.

Figure 5 shows that a target Hamiltonian H_T with arbitrary couplings \mathbf{A} with this interaction graph can be engineered for 225 qubits in about 60 seconds of classical runtime using the Pauli conjugation method (PauliLP).

IV. ERROR ROBUSTNESS AND MITIGATION TECHNIQUES

To successfully apply the Pauli or Clifford conjugation in practice it is necessary to make the resulting pulse sequences robust against dominant errors. In this section, we provide mitigation techniques for our efficient conjugation methods, which come with little overhead. The simultaneous action of the single-qubit pulse and system Hamiltonian is called the finite pulse time error. It has been shown that this error is detrimental to approaches similar to ours [56]. Therefore, our focus lies on the error due to a finite single-qubit pulse duration. Furthermore, we provide a modification to combine our Clifford method with robust composite pulses, making it robust against many different errors occurring in experiments. We achieve robustness against finite pulse time effects and rotation angle errors using a similar approach as in Votto et al. [18]. Their study focuses on specifically designed π -pulse sequences (i.e. Pauli gates), so-called Walsh sequences, for engineering XY-Hamiltonians. In our work, we generalize their approach to general local system Hamiltonians and arbitrary single-qubit pulses.

AHT is a well-known approach in NMR that allows to investigate the dynamics under a time-dependent Hamiltonian by approximating it with the one of a time-dependent Hamiltonian [57, 58]. This approximation is done by a low order Magnus expansion [59]. We utilize AHT to investigate the error due to a finite single-qubit pulse time. We will heavily use that the error term in the average Hamiltonian has the same interaction graph as the system Hamiltonian.

In the following, we give an explicit form of the finite pulse time error in the average Hamiltonian when interleaving a system Hamiltonian with arbitrarily many layers of single-qubit pulses. For the Clifford conjugation method, we find that the finite pulse time error can be exactly cancelled by a slight modification of (CliffLP). This general investigation enables us to also mitigate finite pulse time error in combination with other pulse errors by replacing the single-qubit Clifford pulses in our Clifford conjugation method with robust composite pulses. The latter are designed to compensate for experimental errors by implementing a gate with a specific pulse sequence. In this way, dominant error sources can be cancelled or suppressed, which includes rotation angle error (Rabi frequency errors), off-resonance errors [60, 61], phase errors [62], pulse shape errors [63–65], or non-stationary, non-Markovian noise [66] or crosstalk [67].

For the Pauli conjugation method, the finite pulse time error on the interaction graph of the system Hamiltonian can only be partially mitigated by modifying (PauliLP). However, the freedom in choosing the rotation direction in the π pulse can be leveraged in addition to the modified (PauliLP) to completely cancel the finite pulse time error term in the average Hamiltonian. Moreover, this simultaneously cancels first order effects of rotation angle errors in the single-qubit pulses.

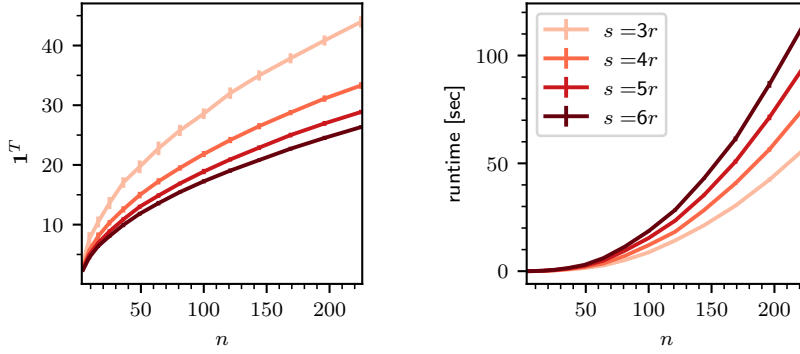


Figure 5. A two-body system Hamiltonian on a 2D square lattice with $n = 4, \dots, 225$ qubits as in eq. (29) and a constant $\mathbf{J} = \mathbf{1}$. The number of interactions is $r = 3^2|E|$ with the set of edges E of a 2D square lattice. We sample 50 target Hamiltonians with uniform random samples \mathbf{A} from $[-1, 1]^r$. For each sample, a feasible $W^{(r \times s)}$ is generated with $s = 3r, \dots, 6r$, using the efficient method from section III A 1. **Left:** The average and the sample standard deviation of the objective function value $\sum \lambda = \mathbf{1}^T \lambda$ over the number of qubits n is shown. **Right:** The average and the sample standard deviation of the runtime for solving (PauliLP) over the number of qubits n is shown.

We introduce basic concepts required for the following chapters in section IV A. Then, in section IV B, we start with the investigation of the finite pulse time error when conjugating a system Hamiltonian with arbitrary many general single-qubit pulses. In section IV C, we apply the general results to present our robust Clifford conjugation methods. Finally, in section IV D, we present our robust Pauli conjugation method.

A. Preliminaries

For our robust methods, we require two well-known approaches to approximate the time evolution under non-commuting Hamiltonians. First, we present general Trotter product formulae to approximate the time evolution under a linear combination of time-independent Hamiltonians. Recently, the performance of the Suzuki-Trotter approximation has been greatly improved [68]. Second, we present the Magnus expansion to approximate the time evolution under a time-dependent Hamiltonian by a time-independent Hamiltonian.

1. Product formula

The time evolution under a Hamiltonian $H = \sum_{i=1}^L H_i$ can generally be approximated by a *product formula*

$$e^{-itH} \approx e^{-i\alpha_q H_{i_q}} \dots e^{-i\alpha_1 H_{i_1}}, \quad (30)$$

with the number of evolution steps q and $i_1, \dots, i_q \in [L]$. We call a product formula *deterministic* if i_1, \dots, i_q can be found by a deterministic algorithm. For our methods we do not require, that $\alpha_1, \dots, \alpha_q \in \mathbb{R}$ are chosen deterministically. The best known deterministic product formulae are the Suzuki-Trotter formulae [69, 70]. The

first- and second order approximations are given by

$$e^{-itH} \approx \left(\prod_{i=1}^L e^{-i \frac{t}{n_{\text{Tro}}} H_i} \right)^{n_{\text{Tro}}} =: S_1(t/n_{\text{Tro}})^{n_{\text{Tro}}} \quad (31)$$

and

$$e^{-itH} \approx \left(\prod_{i=1}^L e^{-i \frac{t}{2n_{\text{Tro}}} H_i} \prod_{i=1}^L e^{-i \frac{t}{2n_{\text{Tro}}} H_i} \right)^{n_{\text{Tro}}} =: S_2(t/n_{\text{Tro}})^{n_{\text{Tro}}}, \quad (32)$$

respectively, with the *number of Trotter cycles* n_{Tro} . The number of Trotter cycles can be increased to improve the accuracy. Throughout our work we use the convention $\prod_{i=1}^L A_i := A_L \dots A_1$ and $\overleftarrow{\prod}_{i=1}^L A_i := A_1 \dots A_L$ to indicate the order of the products of non-commuting operators. The $2k$ -th order Suzuki-Trotter formula for $k > 1$ is defined recursively by

$$S_{2k}(t) := S_{2k-2}(u_k t)^2 S_{2k-2}((1-4u_k)t) S_{2k-2}(u_k t)^2, \quad (33)$$

with $u_k := (4 - 4^{(2k-1)^{-1}})^{-1}$. For a $2k$ -th order Suzuki-Trotter formula the approximation error in spectral norm is bounded by

$$\|S_{2k}(t/n_{\text{Tro}})^{n_{\text{Tro}}} - e^{-itH}\| \leq O \left(\left(t \sum_{i=1}^L \|H_i\| \right)^{2k+1} n_{\text{Tro}}^{-2k} \right), \quad (34)$$

with the spectral norm $\|H_i\|$ [71].

2. Average Hamiltonian theory and the Magnus expansion

Given a time-dependent Hamiltonian $H(t)$, it is sometimes useful to consider a time-independent effective or

average Hamiltonian H_{av} satisfying

$$\mathcal{U}(T) \approx e^{-iT H_{\text{av}}}, \quad (35)$$

where $\mathcal{U}(T)$ is the time evolution operator defined by the differential equation

$$\frac{d\mathcal{U}(t)}{dt} = -iH(t)\mathcal{U}(t), \quad \text{and} \quad \mathcal{U}(0) = \mathbb{1}. \quad (36)$$

This average Hamiltonian can be expressed by the Magnus expansion as follows

$$H_{\text{AV}} = H_{\text{av}}^{(1)} + H_{\text{av}}^{(2)} + \dots, \quad (37)$$

where the first and second order terms are explicitly given by

$$H_{\text{av}}^{(1)} := \frac{1}{T} \int_0^T H(\tau) d\tau \quad (38)$$

and

$$H_{\text{av}}^{(2)} := \frac{1}{2iT} \int_0^T \int_0^\tau [H(\tau), H(\tau')] d\tau' d\tau. \quad (39)$$

The Magnus expansion converges if $\int_0^T \|H(\tau)\|_2 d\tau < \pi$ [72]. As a rule of thumb, the Magnus expansion converges rapidly if

$$\max_{\tau \in [0, T]} \|H(\tau)\|_2 T \ll 1, \quad (40)$$

where the Schatten 2-norm is used [73]. Throughout this work we only consider the first order approximation and write $H_{\text{av}} = H_{\text{av}}^{(1)}$.

B. General robust conjugation method

We start with a general discussion of the finite pulse time error when conjugating a system Hamiltonian H_S with multiple layers of general single-qubit pulses. To this end, we define the Hamiltonian of a general single-qubit pulse layer and the operators relevant for the following discussion.

Definition IV.1. Let $\mathcal{G} \subset \text{Herm}(\mathbb{C}^2)$ be a set of single-qubit pulse generators. A single-qubit pulse layer is

given by a local generator h_i chosen from \mathcal{G} for each qubit $i \in [n]$, the (single-qubit) rotation directions $\mathbf{s} \in \mathbb{F}_2^n$ and the finite pulse time $t_p > 0$. The Hamiltonian generating the single-qubit pulse layer on n qubits is given by

$$H(t_p, \mathbf{s}, \mathbf{h}) := \frac{1}{t_p} \sum_{i=1}^n (-1)^{s_i} h_i, \quad (41)$$

with $\mathbf{h} := (h_1, \dots, h_n)$. The single-qubit pulse layer Hamiltonian is completely specified by the tuple $c := (t_p, \mathbf{s}, \mathbf{h})$, and we write $H_c := H(t_p, \mathbf{s}, \mathbf{h})$. This generates the evolution operator for one single-qubit pulse layer

$$\mathbf{S}_c(t) := e^{-itH_c}, \quad \text{and} \quad \mathbf{S}_c := \mathbf{S}_c(t_p). \quad (42)$$

More generally, we consider a sequence of n_L single-qubit pulse layers and introduce the layer index $\ell \in [n_L]$. The evolution for the ℓ -th single-qubit pulse layer is specified by the tuple $c^{(\ell)} := (t_p^{(\ell)}, \mathbf{s}^{(\ell)}, \mathbf{h}^{(\ell)})$, and we define the tuple $\mathbf{c} := (c^{(1)}, \dots, c^{(n_L)})$. Moreover, we define the single-qubit pulse block and the partial single-qubit pulse block as

$$\mathbf{S}_{\mathbf{c}} := \prod_{\ell=1}^{n_L} \mathbf{S}_{c^{(\ell)}}, \quad (43)$$

and

$$\mathbf{S}_{\mathbf{c} \geq D} := \begin{cases} \prod_{\ell=D}^{n_L} \mathbf{S}_{c^{(\ell)}}, & 1 \leq D \leq n_L \\ \mathbb{1}, & \text{otherwise}, \end{cases} \quad (44)$$

respectively. Similarly, we define the pulse time for the single-qubit pulse block $T_p := \sum_{\ell=1}^{n_L} t_p^{(\ell)}$ and for the partial single-qubit pulse block $T_p^{\leq D} := \sum_{\ell=1}^D t_p^{(\ell)}$.

To illustrate definition IV.1 we provide the Hamiltonian for the single-qubit Pauli pulses from section III A. The Hamiltonian for a single-qubit Pauli pulse layer is given by the π pulse time t_p , an arbitrary rotation direction $\mathbf{s} \in \mathbb{F}_2^n$ and the generators $h_i = \frac{\pi}{2} P_i$, with the Pauli operator P_i given by $P_{\mathbf{b}_i}$ from eq. (11).

Recall from section III that ideally, we would like to implement the conjugation $\mathbf{S}_{\mathbf{c}}^\dagger e^{-it\lambda_c H_S} \mathbf{S}_{\mathbf{c}} = e^{-it\lambda_c \mathbf{S}_{\mathbf{c}}^\dagger H_S \mathbf{S}_{\mathbf{c}}}$, with the system Hamiltonian H_S and the relative evolution time λ_c associated with a single-qubit pulse block $\mathbf{S}_{\mathbf{c}}$. However, due to the finite duration of the single-qubit pulse we get the time evolution

$$U(t\lambda_c) = \left(\prod_{\ell=1}^{n_L} e^{-it_p^{(\ell)} (H_S - H_{c^{(\ell)}})} \right) e^{-it\lambda_c H_S} \left(\prod_{\ell=1}^{n_L} e^{-it_p^{(\ell)} (H_S + H_{c^{(\ell)}})} \right), \quad (45)$$

with the finite duration of the ℓ -th single-qubit pulse layer $t_p^{(\ell)} > 0$, see figure 6. First, we provide the average Hamiltonian for one conjugation with a single-qubit

pulse block to investigate the effect of the finite pulse duration.

Lemma IV.2. The approximation of $U(t\lambda_c)$ in first or-

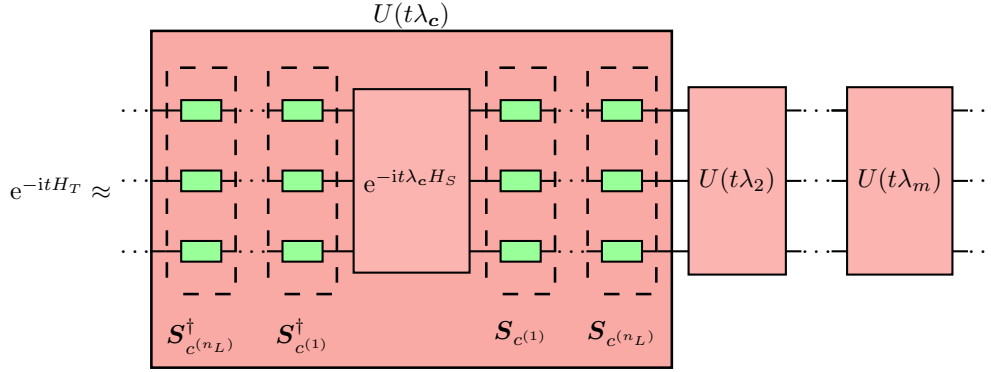


Figure 6. Exemplary quantum circuit for approximating the target evolution. We only implement simple single-qubit pulses in the presence of an always-on system Hamiltonian H_S .

der Magnus expansion is given by $e^{-iH_{av,c}(t)}$ with

$$H_{av,c}(t) = t\lambda_c \mathbf{S}_c^\dagger H_S \mathbf{S}_c + H_{err,c}, \quad (46)$$

where

$$H_{err,c} := 2 \sum_{\ell=1}^{n_L} \mathbf{S}_{c \geq (\ell+1)}^\dagger \int_0^{t_p^{(\ell)}} \mathbf{S}_{c(\ell)}^\dagger(t) H_S \mathbf{S}_{c(\ell)}(t) dt \mathbf{S}_{c \geq (\ell+1)}. \quad (47)$$

The average Hamiltonian $H_{av,c}(t)$ has the same interaction graph as H_S . Moreover, the error for truncating the Magnus expansion after the first order can be bounded in spectral norm by

$$\|U(t\lambda_c) - e^{-iH_{av,c}(t)}\| \leq O((2T_p + t\lambda_c)^2 \|H_S\|^2). \quad (48)$$

We define the matrices capturing the effect of the ideal evolution and the finite pulse time error to state a LP that cancels the finite pulse time error in the average Hamiltonian (in first order).

Lemma IV.3. Let $W(\mathbf{J})^{(r \times s)}, E(\mathbf{J})^{(r \times s)} \in \mathbb{R}^{r \times s}$ be the matrices representing the ideal conjugation and the finite pulse time error in the Pauli basis respectively. The entries are given by

$$W(\mathbf{J})_{ac}^{(r \times s)} := \frac{1}{2^n} \text{Tr} \left(P_a \left(\mathbf{S}_c^\dagger H_S \mathbf{S}_c \right) \right), \quad (49)$$

and

$$E(\mathbf{J})_{ac}^{(r \times s)} := \frac{1}{2^n} \text{Tr} (P_a H_{err,c}), \quad (50)$$

and can be calculated in polynomial time in the number of qubits for a local system Hamiltonian H_S .

The proofs of lemma IV.2 and lemma IV.3 can be found in appendix C. Let $W(\mathbf{J})^{(r \times s)}$ be feasible (as in definition III.1) and let $\sum_c \lambda_c \mathbf{S}_c^\dagger H_S \mathbf{S}_c$ be able to modify all interaction terms in the interaction graph of H_S . Then, the LP

$$\begin{aligned} & \text{minimize} \quad \mathbf{1}^T \boldsymbol{\lambda} \\ & \text{subject to} \quad W(\mathbf{J})^{(r \times s)} \boldsymbol{\lambda} + E(\mathbf{J})^{(r \times s)} \mathbf{1} = \mathbf{A}, \\ & \quad \quad \quad \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^s \end{aligned} \quad (\text{robustLP})$$

always has a solution for any $\mathbf{A} \in \mathbb{R}^r$. This follows directly from definition III.1. Now, we are ready to state the main result of this section.

Theorem IV.4. The target time evolution e^{-itH_T} can be efficiently approximated by a deterministic product formula implementing a product of $U(t\lambda_c)$, with λ_c from (robustLP). Moreover, this approximation is robust against the finite pulse time effect. The only approximation errors are given by the ones from the Magnus approximation (lemma IV.2) and the approximation error from the deterministic product formula.

Proof. The chosen product formula determines the number of implemented single-qubit conjugations $U(t\lambda_c)$ for each c . Let this number be n_c . The finite pulse time error term in (robustLP) has to be rescaled to account for n_c since each implementation of $U(t\lambda_c)$ causes the associated finite pulse time error. This can be done by multiplying the c -th column of $E(\mathbf{J})^{(r \times s)}$ by n_c .

For simplicity, we assume the first order Trotter approximation in eq. (31). To account for the number of implemented single-qubit conjugations we have to rescale $E(\mathbf{J})^{(r \times s)}$ by $n_c = n_{\text{Tro}}$ for all c . Let $tH_{av} := \sum_c^s H_{av,c}(t)$, with λ_c from (robustLP). Let the target Hamiltonian be

$$H_T = \sum_{a \in \mathbb{R}_2^{2^n} \setminus \{0\}} A_a P_a. \quad (51)$$

By the constraint of (robustLP), we have $H_{av} = H_T$. The time evolution governed by the target Hamiltonian H_T can be approximated with

$$\begin{aligned} e^{-itH_T} &= e^{-itH_{av}} \\ &\approx \left(\overrightarrow{\prod}_c e^{-iH_{av,c}(\frac{t}{n_{\text{Tro}}})} \right)^{n_{\text{Tro}}} \\ &\approx \left(\overrightarrow{\prod}_c U(t\lambda_c/n_{\text{Tro}}) \right)^{n_{\text{Tro}}}, \end{aligned} \quad (52)$$

where the first approximation is given by the Trotter scheme and the second approximation is given by the first order Magnus expansion from lemma IV.2. \square

Remark 1. The Trotter approximation error can be bounded as in eq. (34) and depends on the evolution time $t \sum_c \lambda_c$. The approximation error for truncating the Magnus expansion increases with the finite pulse duration, see eq. (48). Note, that the truncation error bound in eq. (48) is not tight and might be improved [74].

Remark 2. The (robustLP) is efficiently solvable with the relaxation from section IIIB1 if $s \geq 2r$ with r the number of interaction terms that can be modified. To approximate the evolution under the target Hamiltonian H_T with n_{Tro} Trotter cycles we have to implement at most $n_{\text{Tro}}s$ evolution blocks $U(t\lambda_c)$, even if $\lambda_c = 0$. Since the finite pulse time errors for all single-qubit pulse layers are taken into account in (robustLP). Each evolution block contains $2n_L$ single-qubit pulse layers (possibly being part of a robust composite pulse sequence). Then, the total number of single-qubit pulse layers is at most $2n_L n_{\text{Tro}}s$.

The number of single-qubit pulse layers can be reduced by formulating the (MILP), which we explain in section IVE. This (MILP) reduces the number of evolution blocks in one Trotter cycle from $s \geq 2r$ to $s \approx r$. However, MILP's are in general hard to solve. The size of (MILP) is drastically reduced due to our efficient relaxation, and solving it for small instances is still feasible.

To summarize, our efficient relaxation plays a central role in both cases for solving the (robustLP) and (MILP). The (MILP) is feasible only for small instances whereas the (robustLP) is efficiently solvable at the cost of more single-qubit pulse layers.

In the next sections, we leverage the general robust conjugation method to robustify the Clifford and Pauli conjugation methods.

C. Robust Clifford conjugation method

For the Clifford conjugation method we have two single-qubit pulse layers $n_L = 2$ of $\pi/2$ pulses, splitting the Pauli pulses into two pulses.

Definition IV.5 (Single-qubit Clifford pulse block). The single-qubit Clifford pulse block \mathcal{S}_c is specified by the tuple $\mathbf{c} = ((t_p^{(1)}, \mathbf{s}^{(1)}, \mathbf{h}^{(1)}), (t_p^{(2)}, \mathbf{s}^{(2)}, \mathbf{h}^{(2)}))$, with the finite pulse times of one $\pi/2$ pulse, and we have $t_p^{(1)} = t_p^{(2)}$. The rotation direction $\mathbf{s}^{(\ell)}$ is fixed, and we set $s_i^{(\ell)} = 1$ if on the i -th qubit and the ℓ layer we have $\sqrt{(\cdot)}^\dagger$ and $s_i^{(\ell)} = 0$ if we have $\sqrt{(\cdot)}$. The generators are $h_i = \frac{\pi}{4t_p^{(\ell)}} P_i^{(\ell)}$, with the Pauli operators from the square-root Pauli gates $P_i^{(\ell)}$.

The Clifford method can be made robust against the finite pulse time error by direct application of theorem IV.4. Our results for the efficient relaxation of the Clifford conjugation method in section IIIB1 ensures that (robustLP) is feasible and that all interaction terms in the interaction graph of H_S can be modified.

Corollary IV.6 (Robustness against finite pulse time errors). *Let there be two layers of $\pi/2$ pulses, representing the single-qubit pulses for the Clifford conjugation as in definition IV.5. Then, the finite pulse time effect can be suppressed using theorem IV.4.*

The Clifford conjugation method can also be made robust by taking advantage of the rich field of robust composite pulses. The $\pi/2$ pulses in each of the two layers in the Clifford conjugation method can be made robust by replacing each pulse with robust composite pulses of length $n_L/2$. Then, the finite pulse time error is different but can still be corrected using theorem IV.4.

Corollary IV.7 (Robustness against pulse errors). *Let there be a robust composite pulse sequence for $\pi/2$ pulses of length $n_L/2$. Replacing each $\pi/2$ pulse in the Clifford conjugation method by such robust composite pulses yields n_L single-qubit pulse layers. Then, the finite pulse time effect is different from corollary IV.6 but still can be suppressed by theorem IV.4.*

The ability to modify any interaction in the interaction graph of H_S and the combination with robust composite pulses makes our Clifford conjugation robust against a wide range of different errors in experiments. Note, that composite pulses with a short overall duration are more beneficial due to the faster convergence of the Magnus expansion eq. (40). In section VB we combine the SCROFULOUS pulses [75] and the SCROBUTUS pulses [61] with the robust Clifford conjugation to implement arbitrary Heisenberg Hamiltonians.

D. Robust Pauli conjugation method

The Pauli conjugation can only change the Pauli coefficients in the system Hamiltonian but not modify the type of interactions. Consequently, the general robust conjugation method is not directly applicable. Therefore, we generalize the robustness conditions of Votto et al. [18] to arbitrary local Hamiltonians.

Definition IV.8 (Single-qubit Pauli pulse layer). The single-qubit Pauli pulse layer \mathcal{S}_c is specified by the tuple $\mathbf{c} = (t_p, \mathbf{s}, \mathbf{h})$, with the finite pulse time t_p of one π pulse. The rotation direction is $\mathbf{s} \in \mathbb{F}_2^n$ and can be chosen freely for π pulses. The generators are $h_i = \frac{\pi}{2} P_i$, with the Pauli operator P_i given by $P_{\mathbf{b}_i}$ from eq. (11).

For the Pauli conjugation $U(t\lambda_c)$ simplifies to

$$U(t\lambda_c) = e^{-it_p(H_S - H_c)} e^{-it\lambda_c H_S} e^{-it_p(H_S + H_c)}. \quad (53)$$

Next, we provide the average Hamiltonian for the conjugation of the system Hamiltonian with a single-qubit Pauli pulse layer to investigate the effect of the finite pulse time.

Lemma IV.9. *Consider all labels $\mathbf{a} \in \mathbb{F}_2^{2n}$ for the non-zero interactions $J_{\mathbf{a}} \neq 0$ in the system Hamiltonian.*

Then, the approximation of $U(t\lambda_c)$ for the Pauli conjugation in first order Magnus expansion is given by $e^{-iH_{av,c}(t)}$ with

$$H_{av,c}(t) = t\lambda_c \mathbf{S}_c^\dagger H_S \mathbf{S}_c + H_{err,c}, \quad (54)$$

where

$$H_{err,c} = \sum_{\mathbf{a} \in \mathbb{F}_2^{2^n} \setminus \{0\}} \left(J_{\mathbf{a}} E_{\mathbf{a},c}^{(r \times s)} P_{\mathbf{a}} + R_{\mathbf{a},c} \right). \quad (55)$$

We call the first term in eq. (55) the interaction term, with

$$E_{\mathbf{a},c}^{(r \times s)} := \frac{4t_p}{\pi} \int_0^{\frac{\pi}{2}} \left(\prod_{i \in \text{supp}(\mathbf{a})} (\cos^2(\theta) + (-1)^{\langle \mathbf{a}_i, \mathbf{b}_i \rangle} \sin^2(\theta)) \right) d\theta, \quad (56)$$

which we collect as entries of the matrix $E^{(r \times s)} \in \mathbb{R}^{r \times s}$. We call the second term $R_{\mathbf{a},c}$ the rest term, and it is proportional to $(-1)^{\mathbf{e}_{\mathbf{a}} \cdot \mathbf{s}}$, with $\mathbf{s} \in \mathbb{F}_2^n$ representing the chosen rotation direction of the π pulses and $\mathbf{e}_{\mathbf{a}} \in \mathbb{F}_2^n$ encodes the sign flips due to the finite pulse time error such that $e_{\mathbf{a},i} = 0$ for all $i \notin \text{supp}(\mathbf{a})$.

All proofs in this section can be found in appendix D. We define the LP similar to (robustLP)

$$\begin{aligned} & \text{minimize} \quad \mathbf{1}^T \boldsymbol{\lambda} \\ & \text{subject to} \quad W^{(r \times s)} \boldsymbol{\lambda} + E^{(r \times s)} \mathbf{1} = \mathbf{M}, \\ & \quad \quad \quad \boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^s, \end{aligned} \quad (\text{robustPauliLP})$$

with $W^{(r \times s)} \in \mathbb{R}^{r \times s}$ and $\mathbf{M} \in \mathbb{R}^r$ from section III A and $E^{(r \times s)} \in \mathbb{R}^{r \times s}$ from lemma IV.9. Similar to theorem IV.4 we can cancel the effect of the interaction terms in eq. (55) by implementing $U(t\lambda_c)$, with λ_c from (robustPauliLP). However, there still remains the rest terms $R_{\mathbf{a},c}$ in eq. (55) which can be cancelled by selecting the π pulse direction appropriately.

Proposition IV.10 (Robustness against finite pulse time errors). *The target time evolution e^{-itH_T} can be approximated by a deterministic product formula implementing $U(t\lambda_c)$, with λ_c from (robustPauliLP), and choosing the rotation directions $\mathbf{s} \in \mathbb{F}_2^n$ of the π pulses such that $\sum_{\mathbf{s}} (-1)^{\mathbf{e}_{\mathbf{a}} \cdot \mathbf{s}} = 0$ for all $\mathbf{e}_{\mathbf{a}} \in \mathbb{F}_2^n$ with $e_{\mathbf{a},i} = 0$ for all $i \notin \text{supp}(\mathbf{a})$ for any \mathbf{a} with $J_{\mathbf{a}} \neq 0$, as in lemma IV.9. The only approximation errors are given by the ones from the Magnus approximation (lemma IV.2) and the approximation error from the deterministic product formula.*

Proof. Similar as in the proof of theorem IV.4 the columns of $E^{(r \times s)}$ in (robustPauliLP) has to be rescaled by the number of implementations $U(t\lambda_c)$ which we denote by n_c . For simplicity we assume the first order Trotter approximation in eq. (31).

We start with the rest term in the average Hamiltonian in eq. (55) which is proportional to $(-1)^{\mathbf{e}_{\mathbf{a}} \cdot \mathbf{s}}$.

Let's assume we require κ different rotation directions $\mathbf{s}_{(1)}, \dots, \mathbf{s}_{(\kappa)} \in \mathbb{F}_2^n$ such that $\sum_{j=1}^{\kappa} (-1)^{\mathbf{e}_{\mathbf{a}} \cdot \mathbf{s}_{(j)}} = 0$ for all $\mathbf{e}_{\mathbf{a}} \in \mathbb{F}_2^n$ with $e_{\mathbf{a},i} = 0$ for all $i \notin \text{supp}(\mathbf{a})$ for any \mathbf{a} with $J_{\mathbf{a}} \neq 0$. A non-optimal choice of $\mathbf{s}_{(j)}$ are all possible binary combinations without the all-zero vector, then $\kappa = 2^n - 1$. An efficient choice is provided in proposition IV.12 below. To indicate the used rotation direction, we specify the single-qubit Pauli pulse layer by the tuple $\mathbf{c}_{(j)} = (t_p, \mathbf{s}_{(j)}, \mathbf{h})$. Note, that the generators do not change. Let the partial average Hamiltonian be the average Hamiltonian without the rest error term

$$\begin{aligned} \tilde{H}_{av,c}(t) &:= \sum_{j=1}^{\kappa} H_{av,c_{(j)}} \left(\frac{t}{\kappa} \right) \\ &= t\lambda_c \mathbf{S}_c^\dagger H_S \mathbf{S}_c + \kappa \sum_{\mathbf{a} \in \mathbb{F}_2^{2^n} \setminus \{0\}} E_{\mathbf{a},c}^{(r \times s)} J_{\mathbf{a}} P_{\mathbf{a}}. \end{aligned} \quad (57)$$

The evolution under the partial average Hamiltonian can be approximated by

$$e^{-it\tilde{H}_{av,c}(t)} \approx \prod_{j=1}^{\kappa} e^{-iH_{av,c_{(j)}} \left(\frac{t}{\kappa} \right)} \approx \prod_{j=1}^{\kappa} U(t\lambda_{c_{(j)}}/\kappa), \quad (58)$$

where the first approximation is given by a first order Trotter scheme where we set $n_{\text{Tro}} = 1$ for simplicity, and the second approximation is given by the first order Magnus expansion from lemma IV.2. Let $tH_{av} := \sum_{\mathbf{c}} \tilde{H}_{av,c}(t)$, with λ_c from (robustPauliLP). Let the target Hamiltonian be

$$H_T = \sum_{\mathbf{a} \in \mathbb{F}_2^{2^n} \setminus \{0\}} A_{\mathbf{a}} P_{\mathbf{a}}. \quad (59)$$

By the constraint of (robustPauliLP), we have $H_{av} = H_T$. The time evolution governed by the target Hamiltonian H_T can be approximated with

$$\begin{aligned} e^{-itH_T} &= e^{-itH_{av}} \\ &\approx \left(\prod_{\mathbf{c}} e^{-i\tilde{H}_{av,c_{(j)}} \left(\frac{t}{n_{\text{Tro}}} \right)} \right)^{n_{\text{Tro}}} \\ &\approx \left(\prod_{\mathbf{c}} \prod_{j=1}^{\kappa} U \left(\frac{t\lambda_{c_{(j)}}}{\kappa n_{\text{Tro}}} \right) \right)^{n_{\text{Tro}}}, \end{aligned} \quad (60)$$

where the first approximation is given by the Trotter scheme. To account for the number of implemented Pauli conjugations $U(t\lambda_c)$ we have to rescale $E^{(r \times s)}$ by $n_c = \kappa n_{\text{Tro}}$ for all \mathbf{c} . \square

Note, that even if $\lambda_c = 0$ the evolution block $U(t\lambda_c)$ still has to be implemented with zero free evolution time. The number of evolution blocks can be reduced by formulating a MILP, which we explain in section IV E.

The robustness against finite pulse time errors simultaneously implies robustness against the first order effects of rotation angle errors.

Proposition IV.11. *Let the rotation directions $\mathbf{s} \in \mathbb{F}_2^n$ of the π pulses such that $\sum_{\mathbf{s}} (-1)^{\mathbf{e}_{\mathbf{a}} \cdot \mathbf{s}} = 0$ for all $\mathbf{e}_{\mathbf{a}} \in \mathbb{F}_2^n$ with $e_{\mathbf{a},i} = 0$ for all $i \notin \text{supp}(\mathbf{a})$ for any \mathbf{a} with $J_{\mathbf{a}} \neq 0$, as in lemma IV.9. Then, the rotation angle errors in the first order Taylor approximation cancels.*

Given a system Hamiltonian with only two-body interactions, i.e. only interactions $P_{\mathbf{a}}$ with $|\text{supp}(\mathbf{a})| = 2$, a good choice for the π pulse rotation directions $\mathbf{s}_{(j)} \in \mathbb{F}_2^n$ for $j = 1, \dots, \kappa$ can be found by utilizing the orthogonality property of Walsh-Hadamard matrices.

Proposition IV.12. *Let $\kappa = 2^{\lceil \log_2(n+1) \rceil} \leq 2n$ and let $W^{(\kappa \times \kappa)}$ be the $\kappa \times \kappa$ dimensional Walsh-Hadamard matrix. Choose n distinct columns from $W^{(\kappa \times \kappa)}$ without the first column and define the resulting partial Walsh-Hadamard matrix as $W^{(\kappa \times n)}$. Let $(-1)^{\mathbf{s}_{(j)}}$ be the j -th row of $W^{(\kappa \times n)}$. Then, for $\mathbf{a} \in \mathbb{F}_2^{2n}$ and $\mathbf{e}_{\mathbf{a}} \in \mathbb{F}_2^n$ with $|\text{supp}(\mathbf{a})| = 2$ and $e_{\mathbf{a},i} = 0$ for all $i \notin \text{supp}(\mathbf{a})$ we obtain $\sum_{j=1}^{\kappa} (-1)^{\mathbf{e}_{\mathbf{a}} \cdot \mathbf{s}_{(j)}} = 0$.*

E. Reducing the number of single-qubit pulses

Let W, E be either $W(\mathbf{J})^{(r \times s)}, E(\mathbf{J})^{(r \times s)}$ or $W^{(r \times s)}, E^{(r \times s)}$ as in (robustLP) or (robustPauliLP) respectively. In these LPs the sum of the finite pulse time effects for all considered evolution blocks $U(t\lambda_{\mathbf{c}})$ is given by the vector $E \cdot \mathbf{1} \in \mathbb{R}^r$. The solution λ in the (robustLP) and (robustPauliLP) is sparse [41]. However, when mitigating the finite pulse time errors as in theorem IV.4 and proposition IV.10 then all the single-qubit pulse conjugations have to be implemented (with zero free evolution time if $\lambda_{\mathbf{c}} = 0$). The LP's can be extended with additional binary variables $\mathbf{z} \in \{0, 1\}^s$ and an additional constraint to only consider the finite pulse time errors for the single-qubit pulses with non-zero free evolution time [76]. This yields the mixed integer linear program (MILP)

$$\begin{aligned} & \text{minimize} && \alpha \mathbf{1}^T \lambda + (1 - \alpha) \mathbf{1}^T \mathbf{z} \\ & \text{subject to} && W\lambda + E\mathbf{z} = \mathbf{M}, \\ & && c_l \mathbf{z} \leq \lambda \leq c_u \mathbf{z}, \\ & && \lambda \in \mathbb{R}_{\geq 0}^s, \quad \mathbf{z} \in \{0, 1\}^s. \end{aligned} \quad (\text{MILP})$$

The free parameter $\alpha \in [0, 1]$ assigns weights to the minimization of the free evolution times $\alpha = 1$ or the minimization of the number of single-qubit pulse layers $\alpha = 0$. $0 \leq c_l < c_u$ are lower and upper bounds on the entries of λ . The interval $[c_l, c_u]$ has to be large enough such that (MILP) has a solution. Although (MILP) is in general hard to solve, there are many powerful heuristics and software packages to solve such optimizations [48].

If not otherwise stated, we used the parameters $c_l = 10^{-6}$, $c_u = 10^3$ and $\alpha = 10^{-2}$. We used MOSEK to solve (MILP) with parameter MSK_DPAR_MIO_TOL_REL_GAP set to 1.0 [48].

V. NUMERICAL HAMILTONIAN SIMULATIONS

To benchmark our methods, we perform numerical simulations that model the most relevant error sources occurring in practice. We consider a device with 2D lattice interactions, motivated by superconducting qubit platforms, and one with all-to-all connectivity modelling an ion trap. However, the presented methods are also applicable to other platforms such as cold atoms.

We numerically simulate the time evolution given by the robust Pauli and Clifford conjugation methods by explicitly computing products of matrix exponentials. More precisely, the decomposition of the target Hamiltonian is implemented using the first- or second-order Trotter formula,

$$U_{\text{sim}} = \left(\prod_{\mathbf{c}}^{\rightarrow} U \left(\frac{t\lambda_{\mathbf{c}}}{n_{\text{Tro}}} \right) \right)^{n_{\text{Tro}}} \quad (61)$$

or

$$U_{\text{sim}} = \left(\prod_{\mathbf{c}}^{\leftarrow} U \left(\frac{t\lambda_{\mathbf{c}}}{2n_{\text{Tro}}} \right) \prod_{\mathbf{c}}^{\rightarrow} U \left(\frac{t\lambda_{\mathbf{c}}}{2n_{\text{Tro}}} \right) \right)^{n_{\text{Tro}}}, \quad (62)$$

where a single evolution block $U(t\lambda_{\mathbf{c}})$ consists of the time evolution under the system Hamiltonian conjugated by single-qubit pulses as in eq. (45), and thus explicitly models finite pulse time errors.

We model the rotation angle and off-resonance errors of one single-qubit pulse layer, similar to definition IV.1, as

$$H_{c,(\varepsilon,f)} = \frac{1}{t_p} \sum_{i=1}^n ((1 + \varepsilon_i)(-1)^{s_i} h_i + f_i Z_i), \quad (63)$$

where $\varepsilon_i \stackrel{\text{i.i.d.}}{\sim} \text{unif}([0, \varepsilon])$ is the *relative angle error* and $f_i \stackrel{\text{i.i.d.}}{\sim} \text{unif}([0, f])$ is the *off-resonance error* on the i -th qubit. The relative angle error is sampled once and represents faulty and inhomogeneous control of the single-qubit pulses.

To capture the quality of the implementation, U_{sim} is compared to the target evolution $U_T = e^{-itH_T}$. As a measure of quality, we use the *average gate infidelity*

$$1 - F_{\text{avg}}(U_{\text{sim}}, U_T) = 1 - \frac{\text{Tr}(U_T^\dagger U_{\text{sim}}) + 1}{d + 1}, \quad (64)$$

with the Hilbert space dimension d .

A. Simulation of a 2D lattice model

We consider a quantum platform with a native 2×3 lattice Hamiltonian with $n = 6$ qubits as an abstract model for interacting superconducting qubits,

$$H_S = J \sum_{ij} Z_i Z_j + \sum_{ijk} E_{ijk} X_i X_j X_k. \quad (65)$$

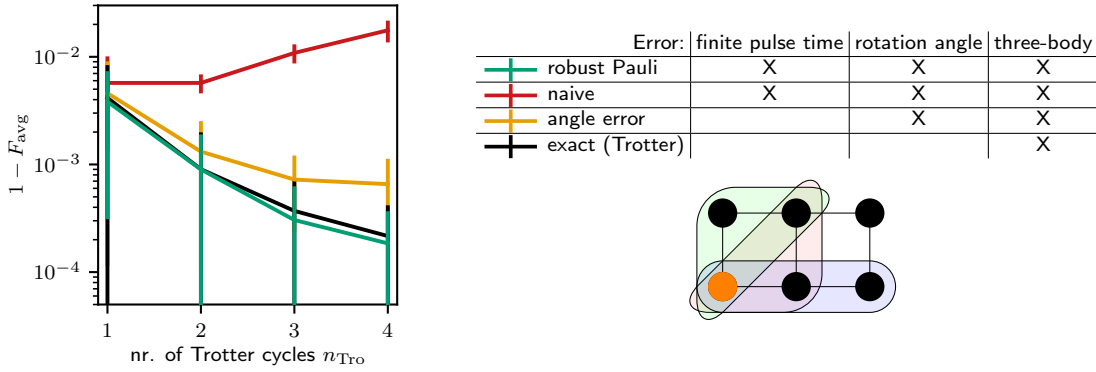


Figure 7. **Left:** The sample mean and standard deviation of the average gate infidelities eq. (64) for implementing the time evolution of e^{-itH_T} with H_T from eq. (66) and $t = 1$ s over the number of Trotter cycles n_{Tro} is shown. The sample mean and standard deviation are calculated over 50 random samples of the target Hamiltonians H_T . Note, that each Trotter cycle contains $\kappa = 8$ evolution blocks $U(t\lambda_c)$ with different rotation directions for the π pulses. **Top Right:** Table indicating which error types are present for the different simulations. **Bottom right:** Example 2×3 2D square lattice on $n = 6$ qubits with the two-body interactions, solid black lines, and the three-body interactions for the lower left qubit, colored areas.

We assume that there are unwanted three-body interactions of unknown strength, where we consider all possible nearest neighbor three-body interactions. An example of the considered two- and three-body interactions are depicted in figure 7 as solid black lines and colored areas respectively. The two-body coupling coefficients are constant, $J = 10^3$ Hz. The three-body coupling coefficients are uniformly sampled $E_{ijk} \stackrel{\text{i.i.d.}}{\sim} \text{unif}([-10^2, 10^2] \cdot \text{Hz})$ after the design of the pulse sequences and therefore considered as unknown. The finite π pulse time is $t_p = 10^{-7}$ s. We want to implement the Ising Hamiltonian

$$H_T = \sum_{ij} A_{ij} Z_i Z_j, \quad (66)$$

with coupling coefficients $A_{ij} \stackrel{\text{i.i.d.}}{\sim} \text{unif}([10^{-1}, 1] \cdot \text{Hz})$. The pulse errors are modeled as in eq. (63) with $\varepsilon = 10^{-1}$ and no off-resonance error.

In figure 7 we compare the naive Pauli conjugation from section III A to the robust Pauli conjugation from section IV D to generate the target Hamiltonian H_T . For the robust Pauli conjugation the relative evolution times λ are the solutions of the (MILP) for the (robustPauliLP). We use the π pulse rotation directions as in proposition IV.12, canceling all two-body terms in the average Hamiltonian. Therefore, for $n = 6$ qubits $\kappa = 2^{\lceil \log_2(n+1) \rceil} = 8$ different rotation directions are required. To cancel the unwanted and unknown three-body interaction terms we apply the results from section III C to both approaches. In the angle error and exact (Trotter) data we apply the same sequences as for the naive but with different errors, see table in figure 7. To approximate the time evolution under H_T we use the first-order Trotter formula from eq. (61) for all sequences. Increasing the number of Trotter cycles n_{Tro} improves the accuracy of the Trotter approximation in absence of other errors. However, an increased number

of Trotter cycles n_{Tro} also yields an increased number of evolution blocks $U(t\lambda_c)$. From figure 7 it is clear that the finite pulse time errors and angle errors in the naive sequences quickly accumulate even for small number of Trotter cycles. However, the robust Pauli sequences converge as quickly as the Trotter sequences without any errors.

B. Simulation of Heisenberg Hamiltonians with an ion trap model

We consider an ion trap with Ytterbium ions in an external magnetic field gradient [77]. The effective system Hamiltonian is

$$H_S = \sum_{i \neq j}^n J_{ij} Z_i Z_j, \quad (67)$$

where the coupling coefficients J_{ij} are calculated for a harmonic trapping potential affecting the equilibrium positions of the ions in the magnetic field gradient (cf. Ref. [36, App. A]). We consider all-to-all connectivity, thus the number of reachable interactions is $r = \frac{n}{2}(n-1)$ for the Pauli conjugation and $r = 3^2 \frac{n}{2}(n-1)$ for the Clifford conjugation. The coupling coefficients J_{ij} are proportional to $(B_1/\omega)^2$, with a magnetic field gradient of $B_1 = 40$ T/m and a trap frequency of $\omega = 2\pi 500$ kHz. The finite π pulse time is $t_p = 2 \mu\text{s}$, which is proportional to π/Ω with the Rabi frequency $\Omega = \omega/2$. Moreover, we consider the rotation angle errors of strength $\varepsilon = 10^{-1}$ and off-resonance errors of strength $f = 10^{-1}$, which are modeled as in eq. (63).

As a target Hamiltonian, we consider the general Heisenberg Hamiltonian

$$H_T = \sum_{i \neq j}^n (A_{ij}^x X_i X_j + A_{ij}^y Y_i Y_j + A_{ij}^z Z_i Z_j), \quad (68)$$

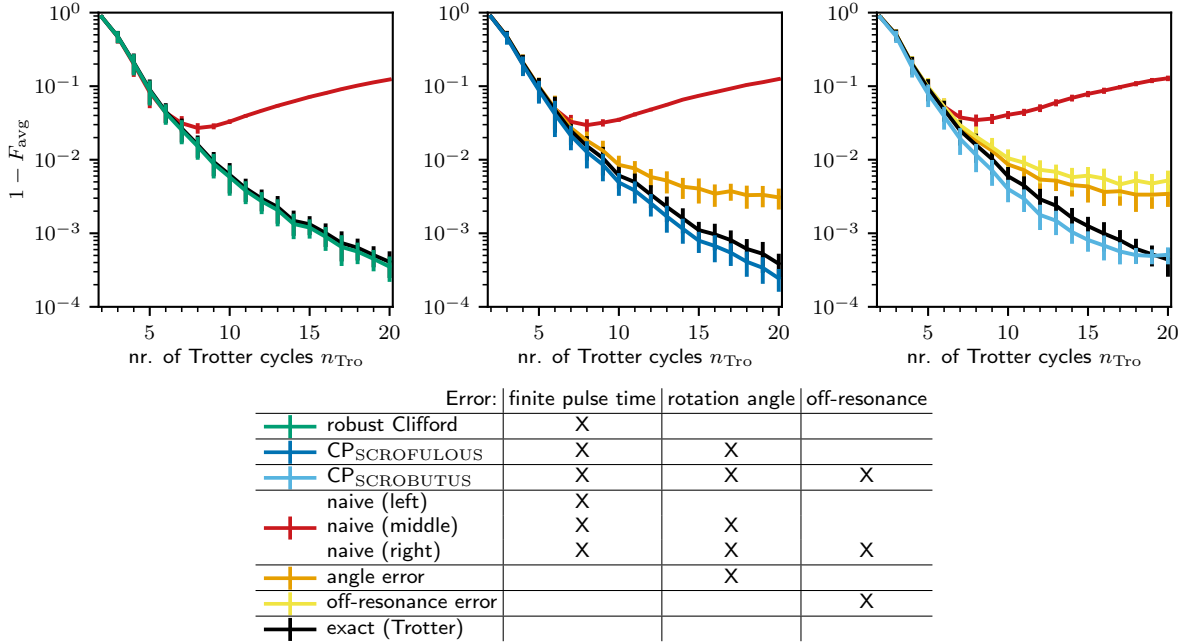


Figure 8. **Top:** The sample mean and standard deviation of the average gate infidelities eq. (64) for implementing the time evolution of e^{-itH_T} with H_T from eq. (68) and $t = 1$ sec over the number of Trotter cycles n_{Tro} is shown. The sample mean and standard deviation are calculated over 50 random samples of the Heisenberg Hamiltonians H_T on $n = 8$ qubits. **Top Left:** The Clifford conjugation robust against finite pulse time errors (dark green) is compared to the non-robust Clifford conjugation (red). **Top Middle:** The Clifford conjugation in combination with the SCROFULOUS pulse sequence [75] robust against finite pulse time errors and rotation angle errors (dark blue) is compared to the non-robust Clifford conjugation (red). **Top Right:** The Clifford conjugation in combination with the SCROBUTUS pulse sequence [61] robust against finite pulse time errors, rotation angle errors and off-resonance errors (light blue) is compared to the non-robust Clifford conjugation (red). **Bottom:** Table indicating which error types are present for the different simulations.

with coupling coefficients $A_{ij}^x, A_{ij}^y, A_{ij}^z \stackrel{\text{i.i.d.}}{\sim} \text{unif}([10^{-1}, 1] \cdot \text{Hz})$ and all-to-all connectivity.

In figure 8 we compare the naive Clifford conjugation from section III B to the robust Clifford, CPSCROFULOUS and CPSCROBUTUS Clifford conjugations from section IV C to generate the target Hamiltonian H_T . The robust Clifford method is robust against the finite pulse time error. The CPSCROFULOUS method is additionally robust against rotation angle errors. It is given as a combination of our robust Clifford conjugation with the SCROFULOUS composite pulses [75]. Finally, the CPSCROBUTUS method is robust against the finite pulse time error, rotation angle errors and off-resonance errors. It is given as a combination of our robust Clifford conjugation with the SCROBUTUS composite pulses [61]. For the robust Clifford, CPSCROFULOUS and CPSCROBUTUS Clifford conjugation the relative evolution times λ are the solutions of the (MILP) version of (robustLP). For the angle error, off-resonance error and exact (Trotter) data we apply the same sequences as for the naive data but with different errors, see table in figure 8. To approximate the time evolution under H_T we use the second-order Trotter formula from eq. (62) for all sequences. As for the Pauli conjugation method in section V A, we can observe that the finite pulse time errors and angle errors in the naive sequences quickly accumulate even for mod-

erate number of Trotter cycles. However, the robust Clifford, CPSCROFULOUS and CPSCROBUTUS sequences converge as quickly as the Trotter sequences without any errors, and seem to be even more accurate at moderate numbers of Trotter cycles.

Note, that the rotation angle error robustness in the CPSCROFULOUS Clifford conjugation is stronger than in the robust Pauli conjugation in section V A. Since each $\pi/2$ pulse (replaced by robust composite pulses) is robust against different angle errors whereas in the robust Pauli conjugation the angle errors are assumed to be constant over several implementations of $U(t\lambda_c)$.

VI. DISCUSSION AND OUTLOOK

We consider a quantum computing or quantum simulation platform that has one single entangling Hamiltonian as *system Hamiltonian* and provide efficient, general and robust methods to engineer arbitrary local Hamiltonians with the same interaction graph from it. Our methods only rely on the use of Pauli or single-qubit Clifford gates and explicitly allows for robust composite pulses. They can be directly used in experiments by applying the pulse sequences generated by the provided Python code [49].

The pulse sequences for the single-qubit gates are obtained by solving a suitable linear program (LP), the classical runtime of which depends polynomially only on the number of Pauli terms in the system Hamiltonian and its interaction graph and can thus be efficiently solved for local Hamiltonians. The Pauli conjugation method is even applicable if only partial knowledge of the system Hamiltonian is available, and can be used to cancel unwanted, but unknown interaction terms. Moreover, the quantum simulation runtime can be reduced at the cost of a higher classical runtime, which provides a flexible trade-off.

Another major advantage of our methods is the robustness against experimental imperfections. Simulation errors introduced by finite pulse times can be explicitly compensated in the computation of the pulse sequence. The Clifford conjugation method can be combined with robust composite pulses, making it robust against major experimental errors. We discuss in detail the effect of finite pulse time errors and rotation angle errors and show that these can be fully mitigated by modified pulse sequences in combination with robust composite pulses. Due to their generality and efficiency, we expect that our methods will find many applications in quantum computation and quantum simulation, such as the fast synthesis of multi-qubit gates, or analogue quantum simulation for problems arising in many-body physics. Furthermore, some recent Hamiltonian learning schemes rely on “reshaping” an unknown Hamiltonian to a diagonal Hamiltonian which can be done efficiently and in a robust way with our Pauli conjugation method [33–35].

In the future, we would like to extend the efficient

Hamiltonian engineering method to fermionic systems for digital and analogue quantum simulations. Another promising future research direction could be the investigation of finite pulse time effects for continuous robust pulses similar to section IV B. Moreover, the design of robust pulses tailored to the conjugation methods might be another interesting direction.

ACKNOWLEDGEMENTS

We are grateful to Ivan Boldin, Patrick Huber, Markus Nünnerich and Rodolfo Muñoz Rodríguez for a productive dialogue on the ion trap platform, and to Matthias Zipper and Christopher Cedzich for fruitful discussions on gate designs for ion trap platforms. Moreover, we thank Matteo Votto for a constructive exchange about his impressive work and the robust sequences therein. We also thank Gaurav Bhole for making us aware of his results. Furthermore, we want to thank Özgün Kum for invaluable comments on our manuscript.

This work has been funded by the German Federal Ministry of Education and Research (BMBF) within the funding program “Quantum Technologies – from Basic Research to Market” via the joint project MIQRO (grant number 13N15522); the ERDF of the European Union and by Fonds of the Hamburg Ministry of Science, Research, Equalities and Districts (BWFGB); and the Fujitsu Services GmbH as part of the endowed professorship “Quantum Inspired and Quantum Optimization.”

-
- [1] R. Somma, G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, *Simulating physical phenomena by quantum networks*, *Phys. Rev. A* **65**, 042323 (2002), [arXiv:quant-ph/0108146](#).
 - [2] R. Blatt and C. F. Roos, *Quantum simulations with trapped ions*, *Nat. Phys.* **8**, 277 (2012).
 - [3] D. Wecker, M. B. Hastings, N. Wiebe, B. K. Clark, C. Nayak, and M. Troyer, *Solving strongly correlated electron models on a quantum computer*, *Phys. Rev. A* **92**, 062318 (2015), [arXiv:1506.05135](#).
 - [4] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. S. Zibrov, M. Endres, M. Greiner, V. Vuletić, and M. D. Lukin, *Probing many-body dynamics on a 51-atom quantum simulator*, *Nature* **551**, 579 (2017), [arXiv:1707.04344](#).
 - [5] J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, P. Becker, H. Kaplan, A. V. Gorshkov, Z.-X. Gong, and C. Monroe, *Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator*, *Nature* **551**, 601 (2017), [arXiv:1708.01044](#).
 - [6] N. Defenu, A. Leroose, and S. Pappalardi, *Out-of-equilibrium dynamics of quantum many-body systems with long-range interactions*, *Physics Reports* **1074**, 1 (2024), [arXiv:2307.04802](#).
 - [7] I. Kassal, J. D. Whitfield, A. Perdomo-Ortiz, M.-H. Yung, and A. Aspuru-Guzik, *Simulating chemistry using quantum computers*, *Annual Review of Physical Chemistry* **62**, 185 (2011), [arXiv:1007.2648](#).
 - [8] D. Wecker, B. Bauer, B. K. Clark, M. B. Hastings, and M. Troyer, *Gate-count estimates for performing quantum chemistry on small quantum computers*, *Phys. Rev. A* **90**, 022305 (2014), [arXiv:1312.1695](#).
 - [9] J. Olson, Y. Cao, J. Romero, P. Johnson, P.-L. Dallaire-Demers, N. Sawaya, P. Narang, I. Kivlichan, M. Wasielewski, and A. Aspuru-Guzik, *Quantum information and computation for chemistry*, [arXiv:1706.05413](#) (2017).
 - [10] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, *Toward the first quantum simulation with quantum speedup*, *Proceedings of the National Academy of Sciences* **115**, 9456 (2018), [arXiv:1711.10980](#).
 - [11] A. J. Daley, I. Bloch, C. Kokail, S. Flannigan, N. Pearson, M. Troyer, and P. Zoller, *Practical quantum advantage in quantum simulation*, *Nature* **607**, 667 (2022).
 - [12] J. Preskill, *Quantum Computing in the NISQ era and beyond*, *Quantum* **2**, 79 (2018), [arXiv:1801.00862](#).
 - [13] R. Trivedi, A. Franco Rubio, and J. I. Cirac, *Quantum advantage and stability to errors in analogue quantum simulators*, *Nature Communications* **15**, 6507 (2024), [arXiv:2212.04924](#).
 - [14] D. W. Leung, I. L. Chuang, F. Yamaguchi, and Y. Yamamoto, *Efficient implementation of coupled logic gates*

- for quantum computation, *Phys. Rev. A* **61**, 042310 (2000), [arXiv:quant-ph/9904100](#).
- [15] D. Leung, *Simulation and reversal of n -qubit Hamiltonians using Hadamard matrices*, *Journal of Modern Optics* **49**, 1199 (2002), [arXiv:quant-ph/0107041](#).
- [16] J. L. Dodd, M. A. Nielsen, M. J. Bremner, and R. T. Thew, *Universal quantum computation and simulation using any entangling Hamiltonian and local unitaries*, *Phys. Rev. A* **65**, 040301 (2002), [arXiv:quant-ph/0106064](#).
- [17] M. A. Nielsen, M. J. Bremner, J. L. Dodd, A. M. Childs, and C. M. Dawson, *Universal simulation of Hamiltonian dynamics for quantum systems with finite-dimensional state spaces*, *Phys. Rev. A* **66**, 022317 (2002), [arXiv:quant-ph/0109064](#).
- [18] M. Votto, J. Zeiher, and B. Vermersch, *Universal quantum processors in spin systems via robust local pulse sequences*, *Quantum* **8**, 1513 (2024), [arXiv:2311.10600](#).
- [19] J. Choi, H. Zhou, H. S. Knowles, R. Landig, S. Choi, and M. D. Lukin, *Robust dynamic Hamiltonian engineering of many-body spin systems*, *Phys. Rev. X* **10**, 031002 (2020), [arXiv:1907.03771](#).
- [20] S. Choi, N. Y. Yao, and M. D. Lukin, *Dynamical engineering of interactions in qudit ensembles*, *Phys. Rev. Lett.* **119**, 183603 (2017), [arXiv:1703.09808](#).
- [21] H. Zhou, H. Gao, N. T. Leitao, O. Makarova, I. Cong, A. M. Douglas, L. S. Martin, and M. D. Lukin, *Robust Hamiltonian engineering for interacting qudit systems*, *Phys. Rev. X* **14**, 031017 (2024), [arXiv:2305.09757](#).
- [22] S. Geier, N. Thaicharoen, C. Hainaut, T. Franz, A. Salzinger, A. Tebben, D. Grimshandl, G. Zürn, and M. Weidemüller, *Floquet Hamiltonian engineering of an isolated many-body spin system*, *Science* **374**, 1149 (2021), [arXiv:2105.01597](#).
- [23] H. Zhou, J. Choi, S. Choi, R. Landig, A. M. Douglas, J. Isoya, F. Jelezko, S. Onoda, H. Sumiya, P. Cappellaro, H. S. Knowles, H. Park, and M. D. Lukin, *Quantum metrology with strongly interacting spin systems*, *Phys. Rev. X* **10**, 031003 (2020), [arXiv:1907.10066](#).
- [24] M. Garcia-de Andoin, A. Saiz, P. Pérez-Fernández, L. Lamata, I. Oregi, and M. Sanz, *Digital-analog quantum computation with arbitrary two-body Hamiltonians*, *Phys. Rev. Res.* **6**, 013280 (2024), [arXiv:2307.00966](#).
- [25] D. Hayes, S. T. Flammia, and M. J. Biercuk, *Programmable quantum simulation by dynamic Hamiltonian engineering*, *New J. Phys.* **16**, 10.1088/1367-2630/16/8/083027 (2014), [arXiv:1309.6736](#).
- [26] C. Figgatt, A. Ostrander, N. M. Linke, K. A. Landsman, D. Zhu, D. Maslov, and C. Monroe, *Parallel entangling operations on a universal ion-trap quantum computer*, *Nature* **572**, 368 (2019), [arXiv:1810.11948](#).
- [27] Y. Lu, S. Zhang, K. Zhang, W. Chen, Y. Shen, J. Zhang, J.-N. Zhang, and K. Kim, *Global entangling gates on arbitrary ion qubits*, *Nature* **572**, 363 (2019), [arXiv:1901.03508](#).
- [28] N. Grzesiak, R. Blümel, K. Wright, K. M. Beck, N. C. Pienti, M. Li, V. Chaplin, J. M. Amini, S. Debnath, J.-S. Chen, and Y. Nam, *Efficient arbitrary simultaneously entangling gates on a trapped-ion quantum computer*, *Nat. Commun.* **11**, 2963 (2020), [arXiv:1905.09294](#).
- [29] J. K. Pachos and M. B. Plenio, *Three-spin interactions in optical lattices and criticality in cluster Hamiltonians*, *Phys. Rev. Lett.* **93**, 056402 (2004), [arXiv:quant-ph/0401106](#).
- [30] H. P. Büchler, A. Micheli, and P. Zoller, *Three-body interactions with cold polar molecules*, *Nature Physics* **3**, 726 (2007), [arXiv:cond-mat/0703688](#).
- [31] X. Peng, J. Zhang, J. Du, and D. Suter, *Quantum simulation of a system with competing two- and three-body interactions*, *Phys. Rev. Lett.* **103**, 140501 (2009), [arXiv:0809.0589](#).
- [32] K. Zhang, H. Li, P. Zhang, J. Yuan, J. Chen, W. Ren, Z. Wang, C. Song, D.-W. Wang, H. Wang, S. Zhu, G. S. Agarwal, and M. O. Scully, *Synthesizing five-body interaction in a superconducting quantum circuit*, *Phys. Rev. Lett.* **128**, 190502 (2022), [arXiv:2109.00964](#).
- [33] H.-Y. Huang, Y. Tong, D. Fang, and Y. Su, *Learning many-body Hamiltonians with Heisenberg-limited scaling*, *Phys. Rev. Lett.* **130**, 200403 (2023), [arXiv:2210.03030](#).
- [34] M. Ma, S. T. Flammia, J. Preskill, and Y. Tong, *Learning k -body Hamiltonians via compressed sensing*, [arXiv:2410.18928](#) (2024).
- [35] H.-Y. Hu, M. Ma, W. Gong, Q. Ye, Y. Tong, S. T. Flammia, and S. F. Yelin, *Ansatz-free Hamiltonian learning with Heisenberg-limited scaling*, [arXiv:2502.11900](#) (2025).
- [36] P. Bakler, M. Zipper, C. Cedzich, M. Heinrich, P. H. Huber, M. Johanning, and M. Kliesch, *Synthesis of and compilation with time-optimal multi-qubit gates*, *Quantum* **7**, 984 (2023), [arXiv:2206.06387](#).
- [37] P. Bakler, M. Heinrich, and M. Kliesch, *Time-optimal multi-qubit gates: Complexity, efficient heuristic and gate-time bounds*, *Quantum* **8**, 1279 (2024), [arXiv:2307.11160](#).
- [38] G. Bhole, T. Tsunoda, P. J. Leek, and J. A. Jones, *Rescaling interactions for quantum control*, *Phys. Rev. Applied* **13**, 034002 (2020), [arXiv:1911.04806](#).
- [39] T. Tsunoda, G. Bhole, S. A. Jones, J. A. Jones, and P. J. Leek, *Efficient Hamiltonian programming in qubit arrays with nearest-neighbor couplings*, *Phys. Rev. A* **102**, 032405 (2020), [arXiv:2003.07815](#).
- [40] S. P. Boyd and L. Vandenberghe, *Convex optimization* (Cambridge University Press, Cambridge, UK ; New York, 2004).
- [41] I. Bárány, *A generalization of Carathéodory's theorem*, *Discrete Mathematics* **40**, 141 (1982).
- [42] D. A. Spielman and S.-H. Teng, *Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time*, *Journal of the ACM* **51**, 385 (2004), [arXiv:cs/0111050](#).
- [43] E. Bach and S. Huiberts, *Optimal smoothed analysis of the simplex method*, [arXiv:2504.04197](#) (2025).
- [44] J. Farkas, *Theorie der einfachen Ungleichungen*, *Journal für die Reine und Angewandte Mathematik* **124**, 1 (1902).
- [45] E. Stiemke, *Über positive Lösungen homogener linearer Gleichungen*, *Mathematische Annalen* **76**, 340 (1915).
- [46] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, *A rewriting system for convex optimization problems*, *J. Control Decis.* **5**, 42 (2018), [arXiv:1709.04494](#).
- [47] S. Diamond and S. Boyd, *CVXPY: A Python-embedded modeling language for convex optimization*, *J. Mach. Learn. Res.* **17**, 1 (2016), [arXiv:1603.00943](#).
- [48] MOSEK ApS, *MOSEK Optimizer API for Python 9.3.14* (2022).
- [49] P. Bakler, *Source code for "Efficient Hamiltonian engineering"*, <https://github.com/paba92/EffHamEng> (2024).
- [50] S. Hayakawa, T. Lyons, and H. Oberhauser, *Estimating the probability that a given vector is in the convex hull of*

- a random sample, *Probability Theory and Related Fields* **185**, 705 (2023), [arXiv:2101.04250](#).
- [51] J. G. Wendel, *A problem in geometric probability*, *MATHEMATICA SCANDINAVICA* **11**, 109–112 (1962).
 - [52] U. Wagner and E. Welzl, *A continuous analogue of the upper bound theorem*, *Discrete & Computational Geometry* **26**, 205 (2001).
 - [53] F. Arute et al., *Quantum supremacy using a programmable superconducting processor*, *Nature* **574**, 505 (2019), [arXiv:1910.11333](#).
 - [54] L. Schmid, D. F. Locher, M. Rispler, S. Blatt, J. Zeiher, M. Müller, and R. Wille, *Computational capabilities and compiler development for neutral atom quantum processors—connecting tool developers and hardware experts*, *Quantum Sci. Technol.* **9**, 10.1088/2058-9565/ad33ac (2024), [arXiv:2309.08656](#).
 - [55] L. Henriët, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Raymond, and C. Jurczak, *Quantum computing with neutral atoms*, *Quantum* **4**, 327 (2020), [arXiv:2006.12326](#).
 - [56] V. P. Canelles, M. G. Algaba, H. Heimonen, M. Papič, M. Ponce, J. Rönkkö, M. J. Thapa, I. de Vega, and A. Auer, *Benchmarking Digital-Analog Quantum Computation*, [arXiv:2307.07335](#) (2023).
 - [57] U. Haeberlen and J. S. Waugh, *Coherent averaging effects in magnetic resonance*, *Phys. Rev.* **175**, 453 (1968).
 - [58] U. Haeberlen, *High Resolution NMR in Solids* (Academic Press, 1976).
 - [59] W. Magnus, *On the exponential solution of differential equations for a linear operator*, *Communications on Pure and Applied Mathematics* **7**, 649 (1954).
 - [60] M. Bando, T. Ichikawa, Y. Kondo, and M. Nakahara, *Concatenated composite pulses compensating simultaneous systematic errors*, *Journal of the Physical Society of Japan* **82**, 014004 (2013), [arXiv:1209.4247](#).
 - [61] S. Kukita, H. Kiya, and Y. Kondo, *Short composite quantum gate robust against two common systematic errors*, *Journal of the Physical Society of Japan* **91**, 104001 (2022), [arXiv:2112.12945](#).
 - [62] B. T. Torosov and N. V. Vitanov, *Composite pulses with errant phases*, *Phys. Rev. A* **100**, 023410 (2019), [arXiv:1904.13168](#).
 - [63] G. T. Genov, D. Schraft, T. Halfmann, and N. V. Vitanov, *Correction of arbitrary field errors in population inversion of quantum systems by universal composite pulses*, *Phys. Rev. Lett.* **113**, 043001 (2014), [arXiv:1403.1201](#).
 - [64] G. T. Genov, M. Hain, N. V. Vitanov, and T. Halfmann, *Universal composite pulses for efficient population inversion with an arbitrary excitation profile*, *Phys. Rev. A* **101**, 013827 (2020), [arXiv:2002.08321](#).
 - [65] H.-N. Wu, C. Zhang, J. Song, Y. Xia, and Z.-C. Shi, *Composite pulses for optimal robust control in two-level systems*, *Phys. Rev. A* **107**, 023103 (2023).
 - [66] C. Kabytayev, T. J. Green, K. Khodjasteh, M. J. Biercuk, L. Viola, and K. R. Brown, *Robustness of composite pulses to time-dependent control noise*, *Phys. Rev. A* **90**, 012316 (2014), [arXiv:1402.5174](#).
 - [67] B. T. Torosov, S. S. Ivanov, and N. V. Vitanov, *Narrowband and passband composite pulses for variable rotations*, *Phys. Rev. A* **102**, 013105 (2020), [arXiv:2004.08456](#).
 - [68] M. E. S. Morales, P. C. S. Costa, G. Pantaleoni, D. K. Burgarth, Y. R. Sanders, and D. W. Berry, *Selection and improvement of product formulae for best performance of quantum simulation*, [arXiv:2210.15817](#) (2022).
 - [69] H. F. Trotter, *On the product of semi-groups of operators*, *Proceedings of the American Mathematical Society* **10**, 545 (1959).
 - [70] M. Suzuki, *General theory of fractal path integrals with applications to many-body theories and statistical physics*, *Journal of Mathematical Physics* **32**, 400 (1991).
 - [71] A. M. Childs, Y. Su, M. C. Tran, N. Wiebe, and S. Zhu, *Theory of Trotter error with commutator scaling*, *Phys. Rev. X* **11**, 011020 (2021), [arXiv:1912.08854](#).
 - [72] P. C. Moan and J. Niesen, *Convergence of the Magnus series*, *Foundations of Computational Mathematics* **8**, 291 (2008), [arXiv:0609198](#).
 - [73] A. Brinkmann, *Introduction to average Hamiltonian theory. I. Basics*, *Concepts in Magnetic Resonance Part A* **45A**, e21414 (2016).
 - [74] K. Sharma and M. C. Tran, *Hamiltonian simulation in the interaction picture using the Magnus expansion*, [arXiv:2404.02966](#) (2024).
 - [75] H. K. Cummins, G. Llewellyn, and J. A. Jones, *Tackling systematic errors in quantum logic gates with composite rotations*, *Phys. Rev. A* **67**, 042308 (2003), [arXiv:quant-ph/0208092](#).
 - [76] N. B. Karahanoglu, H. Erdoğan, and Ş. İ. Birbil, *A mixed integer linear programming formulation for the sparse recovery problem in compressed sensing*, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013) pp. 5870–5874.
 - [77] C. Piltz, T. Sriarunothai, S. S. Ivanov, S. Wölk, and C. Wunderlich, *Versatile microwave-driven trapped ion spin system for quantum information processing*, *Sci. Adv.* **2**, e1600093 (2016), [arXiv:1509.01478](#).
 - [78] A. Arnal, F. Casas, and C. Chiralt, *A general formula for the Magnus expansion in terms of iterated integrals of right-nested commutators*, *Journal of Physics Communications* **2**, 035024 (2018), [arXiv:1710.10851](#).

APPENDIX

Appendix A: Properties of solutions of (PauliLP)

In this section, we show the existence of solutions, provide lower and upper bounds on the relative evolution time, and discuss the tightness of these bounds. First, a direct consequence of proposition III.2 is that (PauliLP) is feasible or arbitrary Hamiltonians H_S and H_T satisfying $\text{nz}(\mathbf{A}) \subseteq \text{nz}(\mathbf{J})$.

Corollary A.1 (existence of solutions). *Let*

$$H_S = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} J_{\mathbf{a}} P_{\mathbf{a}} \quad \text{and} \quad H_T = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} A_{\mathbf{a}} P_{\mathbf{a}}, \quad (\text{A1})$$

with $\text{nz}(\mathbf{A}) \subseteq \text{nz}(\mathbf{J})$. Then the partial Walsh-Hadamard matrix $W^{(r \times 4^n)}$ with $r = |\text{nz}(\mathbf{J})|$ leads to a feasible (PauliLP).

Proof. We know that all rows of the Walsh-Hadamard matrix are linearly independent. Furthermore, each row of the Walsh-Hadamard matrix sums to zero, thus $\mathbf{x} = \mathbf{1} > \mathbf{0}$ is a solution to $W^{(r \times 4^n)} \mathbf{x} = \mathbf{0}$. By proposition III.2 we know that (PauliLP) always has a feasible solution even if we consider an arbitrary subset of rows. \square

To bound the optimal solutions $\mathbf{1}^T \boldsymbol{\lambda}^*$ of (PauliLP) we define the dual LP

$$\begin{aligned} & \text{maximize} \quad \mathbf{M}^T \mathbf{y} \\ & \text{subject to} \quad (W^{(r \times 4^n)})^T \mathbf{y} \leq \mathbf{1}, \mathbf{y} \in \mathbb{R}^r. \end{aligned} \quad (\text{A2})$$

As in the previously considered case of two-body Ising interactions [37], we have the following bounds on the value of (PauliLP):

Theorem A.2 (bounds on solutions). *The optimal objective function value of (PauliLP) with a partial Walsh-Hadamard matrix $W^{(r \times 4^n)}$ is bounded by*

$$\|\mathbf{M}\|_{\ell_\infty} \leq \mathbf{1}^T \boldsymbol{\lambda}^* \leq \|\mathbf{M}\|_{\ell_1}. \quad (\text{A3})$$

Proof. The lower bound can be verified by the fact that $W^{(r \times 4^n)}$ in (PauliLP) only has entries ± 1 and that $\boldsymbol{\lambda}^*$ is non-negative. Thus, it holds that $\|\mathbf{M}\|_{\ell_\infty} = \|W^{(r \times 4^n)} \boldsymbol{\lambda}^*\|_{\ell_\infty} \leq \mathbf{1}^T \boldsymbol{\lambda}^*$.

To show the upper bound, we first define the set of feasible solutions for the dual LP (A2)

$$\mathcal{F} := \left\{ \mathbf{y} \in \mathbb{R}^r \mid (W^{(r \times 4^n)})^T \mathbf{y} \leq \mathbf{1} \right\}. \quad (\text{A4})$$

Next, consider the partial Walsh-Hadamard matrix $W^{((4^n-1) \times 4^n)}$ without the first row, corresponding to the identity Pauli term $P_{\mathbf{a}} = I^{\otimes n}$ with $\mathbf{a} = \mathbf{0}$. We show that

$$\mathcal{S} := \left\{ \mathbf{y} \in \mathbb{R}^{4^n-1} \mid (W^{((4^n-1) \times 4^n)})^T \mathbf{y} \leq \mathbf{1} \right\} \quad (\text{A5})$$

is a simplex. A simplex is formed by affine independent vectors. Vectors of the form $(1, \mathbf{v}_i)^T$ are linearly dependent if and only if the vectors \mathbf{v}_i are affine dependent. From linear dependence it follows that there is a vector $\mathbf{t} \neq \mathbf{0}$ such that $\sum_i t_i (1, \mathbf{v}_i)^T = \mathbf{0}$. Thus $\sum_i t_i = 0$ and $\sum_i \mathbf{v}_i = \mathbf{0}$, and the vectors \mathbf{v}_i are affine dependent. For any n , the Walsh-Hadamard matrix $W = (\mathbf{1}, (W^{((4^n-1) \times 4^n)})^T)$ has linearly independent rows and columns. Therefore, $W^{((4^n-1) \times 4^n)}$ has affine independent columns, and \mathcal{S} forms a simplex. With $\mathcal{H} = \{\mathbf{y} \in \mathbb{R}^{4^n-1} \mid \|\mathbf{y}\|_{\ell_\infty} \leq 1\}$ we denote the $4^n - 1$ dimensional hypercube. Note that the rows of $W^{(r \times 4^n)}$ are rows of the $4^n \times 4^n$ dimensional Walsh-Hadamard matrix. We define the embedding $g : \mathbb{R}^r \rightarrow \mathbb{R}^{(4^n-1)}$ by appending zeros, such that $(W^{(r \times 4^n)})^T \mathbf{y} = (W^{((4^n-1) \times 4^n)})^T g(\mathbf{y})$. Therefore the objective value of the dual LP (A2) can be upper bounded as follows

$$\begin{aligned} \max_{\mathbf{y} \in \mathcal{F}} \langle \mathbf{M}, \mathbf{y} \rangle &= \max_{g(\mathbf{y}) \in \mathcal{S}} \langle g(\mathbf{M}), g(\mathbf{y}) \rangle \\ &\leq \max_{\mathbf{x} \in \mathcal{S}} \langle g(\mathbf{M}), \mathbf{x} \rangle \\ &\leq \max_{\mathbf{x} \in \mathcal{H}} \langle g(\mathbf{M}), \mathbf{x} \rangle = \|\mathbf{M}\|_{\ell_1}. \end{aligned} \quad (\text{A6})$$

The upper bound for $\mathbf{1}^T \boldsymbol{\lambda}^*$ follows by strong duality. \square

Next, we show a (not complete) set of instances of (PauliLP), yielding solutions λ^* which satisfy the upper bound of theorem A.2. Such M constitute the worst cases.

Proposition A.3. *If $M \in \{-w_i \mid w_i \text{ is the } i\text{-th column of } W^{((4^n-1) \times 4^n)}\}$, then $\mathbf{1}^T \lambda^* = \|M\|_{\ell_1}$.*

Proof. Let $y = -w_i$ be a negative column of $W^{((4^n-1) \times 4^n)}$. Then, by orthogonality of the rows/columns of the Walsh-Hadamard matrix, we obtain

$$\left((W^{((4^n-1) \times 4^n)})^T (-w_i) \right)_j = \begin{cases} -(4^n - 1), & i = j \\ 1, & \text{otherwise.} \end{cases} \quad (\text{A7})$$

Therefore, $y = -w_i$ is a feasible solution $(W^{((4^n-1) \times 4^n)})^T y \leq \mathbf{1}$ to the dual LP (A2). The dual objective function value is $M^T y = \|w_i\|_{\ell_1} = 4^n - 1$. Next, we show a feasible solution of (PauliLP). For a $M = -w_i$ define

$$\lambda_j = \begin{cases} 0, & i = j \\ 1, & \text{otherwise.} \end{cases} \quad (\text{A8})$$

Clearly, this satisfies $W^{((4^n-1) \times 4^n)} \lambda = -w_i$. Furthermore, the primal objective function value $\mathbf{1}^T \lambda = 4^n - 1$ is the same as for the dual objective function value, which shows optimality. It is easy to see that $\|M\|_{\ell_1} = 4^n - 1$. \square

Appendix B: Comments on the efficient relaxation for the Pauli conjugation

As mentioned in the main text, Wendel's theorem is applicable to spherical symmetric distributions. This would imply that sampling a certain column from $W^{(r \times 4^n)}$ has the same probability as sampling the same column with a flipped sign. Recall that for a partial Walsh-Hadamard matrix $W_{ab}^{(r \times 4^n)} = (-1)^{\langle a, b \rangle}$, with $\langle a, b \rangle = a_x \cdot b_z + a_z \cdot b_x$. It holds that

$$W_{ab}^{(r \times 4^n)} = -W_{ab}^{(r \times 4^n)} \quad \forall a, b \in \mathbb{F}_2^{2n}, \quad (\text{B1})$$

for $|a| \equiv 1 \pmod{2}$ and with the binary complement \bar{b} is given element-wise given by $\bar{x} := 1 - x$ for any $x \in \{0, 1\}$. If the decomposition of H_S has only terms $J_a P_a$ with odd $|a|$, then we can apply Wendel's theorem directly. In this case, we have a success probability of $1/2$ to find a feasible $W^{(r \times 2^r)}$ (with r interactions) if we sample $2r$ many $b \in \mathbb{F}_2^{2n}$ uniformly random. However, an example of such a Hamiltonian with two-body interactions is

$$H = \sum_{i=1}^n (J_i^X X_i + J_i^Y Y_i) + \sum_{i \neq j}^n (J_{ij}^{XZ} X_i Z_j + J_{ij}^{YZ} Y_i Z_j), \quad (\text{B2})$$

and with commuting interactions is

$$H = \sum_{i=1}^n J_i^X X_i + \sum_{ijk}^n J_{ijk}^{XXX} X_i X_j X_k, \quad (\text{B3})$$

with arbitrary coupling constants. Unfortunately, for a general Hamiltonian, we cannot use Wendel's theorem to construct a relaxation for (PauliLP).

Recently, lower bounds on $p_{s,x}$ have been proposed for arbitrary distributions [50]. However, their results rely on the halfspace depth (or Tukey depth), which is hard to compute. It is a measure of how extreme a point is with respect to a distribution of random points.

Definition B.1 (halfspace depth). *Let x be an arbitrary r -dimensional random vector. Then, the halfspace depth at the origin is defined as*

$$\alpha_x := \inf_{\|c\|=1} \mathbb{P}(c^T x \leq 0). \quad (\text{B4})$$

The halfspace depth α_x is the minimum (fraction) number of points in a halfspace with the origin on the boundary.

Theorem B.2 ([50, Theorem 14]). *Let x be an arbitrary r -dimensional random vector. Then, for each positive integer $s \geq 3r/\alpha_x$, we have*

$$p_{s,x} > 1 - \frac{1}{2^r}. \quad (\text{B5})$$

Let \mathbf{w} be a r -dimensional random vector drawn uniformly from $\text{col}(W^{(r \times 4^n)})$. From corollary A.1, we find the trivial lower bound $1/4^n \leq \alpha_{\mathbf{w}}$, since at least one point is in an arbitrary halfspace with the origin on its boundary. Finding a constant lower bound $1/\beta \leq \alpha_{\mathbf{w}}$ would imply that for $s \geq 3r\beta$ we find a feasible $W^{(r \times s)}$ with high probability. One needs to show that at least $4^{O(n)}$ points are contained in an arbitrary halfspace with the origin on its boundary.

Appendix C: Proofs for the general robust conjugation method

We provide the proofs for section IV B. Some proofs provide more technical details which aims for an easy implementation into a programming language. For the sake of easy readability we repeat the definitions of the main text.

Definition IV.1. Let $\mathcal{G} \subset \text{Herm}(\mathbb{C}^2)$ be a set of single-qubit pulse generators. A single-qubit pulse layer is given by a local generator h_i chosen from \mathcal{G} for each qubit $i \in [n]$, the (single-qubit) rotation directions $\mathbf{s} \in \mathbb{F}_2^n$ and the finite pulse time $t_p > 0$. The Hamiltonian generating the single-qubit pulse layer on n qubits is given by

$$H(t_p, \mathbf{s}, \mathbf{h}) := \frac{1}{t_p} \sum_{i=1}^n (-1)^{s_i} h_i, \quad (41)$$

with $\mathbf{h} := (h_1, \dots, h_n)$. The single-qubit pulse layer Hamiltonian is completely specified by the tuple $\mathbf{c} := (t_p, \mathbf{s}, \mathbf{h})$, and we write $H_{\mathbf{c}} := H(t_p, \mathbf{s}, \mathbf{h})$. This generates the evolution operator for one single-qubit pulse layer

$$\mathbf{S}_{\mathbf{c}}(t) := e^{-itH_{\mathbf{c}}}, \quad \text{and} \quad \mathbf{S}_{\mathbf{c}} := \mathbf{S}_{\mathbf{c}}(t_p). \quad (42)$$

More generally, we consider a sequence of n_L single-qubit pulse layers and introduce the layer index $\ell \in [n_L]$. The evolution for the ℓ -th single-qubit pulse layer is specified by the tuple $\mathbf{c}^{(\ell)} := (t_p^{(\ell)}, \mathbf{s}^{(\ell)}, \mathbf{h}^{(\ell)})$, and we define the tuple $\mathbf{c} := (\mathbf{c}^{(1)}, \dots, \mathbf{c}^{(n_L)})$. Moreover, we define the single-qubit pulse block and the partial single-qubit pulse block as

$$\mathbf{S}_{\mathbf{c}} := \prod_{\ell=1}^{n_L} \mathbf{S}_{\mathbf{c}^{(\ell)}}, \quad (43)$$

and

$$\mathbf{S}_{\mathbf{c} \geq D} := \begin{cases} \prod_{\ell=D}^{n_L} \mathbf{S}_{\mathbf{c}^{(\ell)}}, & 1 \leq D \leq n_L \\ \mathbf{1}, & \text{otherwise,} \end{cases} \quad (44)$$

respectively. Similarly, we define the pulse time for the single-qubit pulse block $T_p := \sum_{\ell=1}^{n_L} t_p^{(\ell)}$ and for the partial single-qubit pulse block $T_p^{\leq D} := \sum_{\ell=1}^D t_p^{(\ell)}$.

We consider the time evolution

$$U(t\lambda_{\mathbf{c}}) = \left(\prod_{\ell=1}^{n_L} e^{-it_p^{(\ell)}(H_S - H_{\mathbf{c}^{(\ell)}})} \right) e^{-it\lambda_{\mathbf{c}} H_S} \left(\prod_{\ell=1}^{n_L} e^{-it_p^{(\ell)}(H_S + H_{\mathbf{c}^{(\ell)}})} \right), \quad (C1)$$

with the finite duration of the ℓ -th pulse layer $0 < t_p^{(\ell)}$.

Lemma IV.2. The approximation of $U(t\lambda_{\mathbf{c}})$ in first order Magnus expansion is given by $e^{-iH_{\text{av},\mathbf{c}}(t)}$ with

$$H_{\text{av},\mathbf{c}}(t) = t\lambda_{\mathbf{c}} \mathbf{S}_{\mathbf{c}}^{\dagger} H_S \mathbf{S}_{\mathbf{c}} + H_{\text{err},\mathbf{c}}, \quad (46)$$

where

$$H_{\text{err},\mathbf{c}} := 2 \sum_{\ell=1}^{n_L} \mathbf{S}_{\mathbf{c} \geq (\ell+1)}^{\dagger} \int_0^{t_p^{(\ell)}} \mathbf{S}_{\mathbf{c}^{(\ell)}}^{\dagger}(t) H_S \mathbf{S}_{\mathbf{c}^{(\ell)}}(t) dt \mathbf{S}_{\mathbf{c} \geq (\ell+1)}. \quad (47)$$

The average Hamiltonian $H_{\text{av},\mathbf{c}}(t)$ has the same interaction graph as H_S . Moreover, the error for truncating the Magnus expansion after the first order can be bounded in spectral norm by

$$\|U(t\lambda_{\mathbf{c}}) - e^{-iH_{\text{av},\mathbf{c}}(t)}\| \leq O((2T_p + t\lambda_{\mathbf{c}})^2 \|H_S\|^2). \quad (48)$$

Proof. We consider the time evolution during the ℓ -th layer of single-qubit pulses in the interaction frame with respect to the single-qubit pulse Hamiltonian $H_{c(\ell)}$ [57, 73]

$$e^{-it_p^{(\ell)}(H_S \pm H_{c(\ell)})} = e^{\mp it_p^{(\ell)} H_{c(\ell)}} \mathcal{U}_{\pm}^{(\ell)}(t_p^{(\ell)}) = \mathbf{S}_{c(\ell)}^{\pm 1} \mathcal{U}_{\pm}^{(\ell)}(t_p^{(\ell)}), \quad (\text{C2})$$

with the interaction frame propagator $\mathcal{U}_{\pm}^{(\ell)}(t_p^{(\ell)})$. The interaction frame propagator has to fulfill

$$\frac{d\mathcal{U}_{\pm}^{(\ell)}(t)}{dt} = -i (\mathbf{S}_{c(\ell)}^{\mp 1}(t) H_S \mathbf{S}_{c(\ell)}^{\pm 1}(t)) \mathcal{U}_{\pm}^{(\ell)}(t). \quad (\text{C3})$$

Inserting eq. (C2) into eq. (C1) yields

$$U(t\lambda_c) = \left(\prod_{\ell=1}^{n_L} \mathbf{S}_{c(\ell)}^{-1} \mathcal{U}_{-}^{(\ell)}(t_p^{(\ell)}) \right) e^{-it\lambda_c H_S} \left(\prod_{\ell=1}^{n_L} \mathbf{S}_{c(\ell)} \mathcal{U}_{+}^{(\ell)}(t_p^{(\ell)}) \right). \quad (\text{C4})$$

We define $\tilde{\mathcal{U}}_{-}^{(\ell)}(t)$ as

$$\frac{d\tilde{\mathcal{U}}_{-}^{(\ell)}(t)}{dt} := \mathbf{S}_{c \geq \ell}^{-1} \frac{d\mathcal{U}_{-}^{(\ell)}(t)}{dt} \mathbf{S}_{c \geq \ell} = -i (\mathbf{S}_{c \geq \ell}^{-1} \mathbf{S}_{c(\ell)}(t) H_S \mathbf{S}_{c(\ell)}^{-1}(t) \mathbf{S}_{c \geq \ell}) \tilde{\mathcal{U}}_{-}^{(\ell)}(t), \quad (\text{C5})$$

and $\tilde{\mathcal{U}}_{+}^{(\ell)}(t)$ as

$$\frac{d\tilde{\mathcal{U}}_{+}^{(\ell)}(t)}{dt} := \mathbf{S}_{c \geq (\ell+1)}^{-1} \frac{d\mathcal{U}_{+}^{(\ell)}(t)}{dt} \mathbf{S}_{c \geq (\ell+1)} = -i (\mathbf{S}_{c \geq (\ell+1)}^{-1} \mathbf{S}_{c(\ell)}^{-1}(t) H_S \mathbf{S}_{c(\ell)}(t) \mathbf{S}_{c \geq (\ell+1)}) \tilde{\mathcal{U}}_{+}^{(\ell)}(t); \quad (\text{C6})$$

where $\tilde{\mathcal{U}}_{-}^{(\ell)}(t)$ and $\tilde{\mathcal{U}}_{+}^{(\ell)}(t)$ are defined such that we can commute the exact evolution of the single-qubit pulse layers to the free evolution under H_S . It directly follows that

$$U(t\lambda_c) = \left(\prod_{\ell=1}^{n_L} \tilde{\mathcal{U}}_{-}^{(\ell)}(t) \right) e^{-it\lambda_c \mathbf{S}_c^{-1} H_S \mathbf{S}_c} \left(\prod_{\ell=1}^{n_L} \tilde{\mathcal{U}}_{+}^{(\ell)}(t) \right). \quad (\text{C7})$$

The Hamiltonian governing the evolution of $U(t\lambda_c)$ is defined piecewise as

$$H_c(\tilde{t}) := \begin{cases} \mathbf{S}_{c \geq (\ell+1)}^{-1} \mathbf{S}_{c(\ell)}^{-1}(\tilde{t}) H_S \mathbf{S}_{c(\ell)}(\tilde{t}) \mathbf{S}_{c \geq (\ell+1)}, & T_p^{\leq (\ell-1)} \leq \tilde{t} < T_p^{\leq \ell} \\ \mathbf{S}_c^{-1} H_S \mathbf{S}_c, & T_p \leq \tilde{t} < T_p + t\lambda_c \\ \mathbf{S}_{c \geq \ell}^{-1} \mathbf{S}_{c(\ell)}(\tilde{t}) H_S \mathbf{S}_{c(\ell)}^{-1}(\tilde{t}) \mathbf{S}_{c \geq \ell}, & T_p + T_p^{\leq (\ell-1)} + t\lambda_c \leq \tilde{t} < T_p + T_p^{\leq \ell} + t\lambda_c. \end{cases} \quad (\text{C8})$$

The effective Hamiltonian is approximated by the first-order term in the Magnus expansion up to time $T = 2T_p + t\lambda_c$, yielding the time independent average Hamiltonian

$$\begin{aligned} H_{\text{av},c}(t) &= \int_0^T H_c(\tilde{t}) d\tilde{t} \\ &= t\lambda_c \mathbf{S}_c^{-1} H_S \mathbf{S}_c + \sum_{\ell=1}^{n_L} \mathbf{S}_{c \geq (\ell+1)}^{-1} \int_0^{t_p^{(\ell)}} \mathbf{S}_{c(\ell)}^{-1}(t) H_S \mathbf{S}_{c(\ell)}(t) dt \mathbf{S}_{c \geq (\ell+1)} + \mathbf{S}_{c \geq \ell}^{-1} \int_0^{t_p^{(\ell)}} \mathbf{S}_{c(\ell)}(t) H_S \mathbf{S}_{c(\ell)}^{-1}(t) dt \mathbf{S}_{c \geq \ell} \\ &= t\lambda_c \mathbf{S}_c^{-1} H_S \mathbf{S}_c + 2 \sum_{\ell=1}^{n_L} \mathbf{S}_{c \geq (\ell+1)}^{-1} \int_0^{t_p^{(\ell)}} \mathbf{S}_{c(\ell)}^{-1}(t) H_S \mathbf{S}_{c(\ell)}(t) dt \mathbf{S}_{c \geq (\ell+1)}. \end{aligned} \quad (\text{C9})$$

We denote the term corresponding to the finite pulse time error by

$$H_{\text{err},c} := 2 \sum_{\ell=1}^{n_L} \mathbf{S}_{c \geq (\ell+1)}^{-1} \int_0^{t_p^{(\ell)}} \mathbf{S}_{c(\ell)}^{-1}(t) H_S \mathbf{S}_{c(\ell)}(t) dt \mathbf{S}_{c \geq (\ell+1)}. \quad (\text{C10})$$

Then, the average Hamiltonian is

$$H_{\text{av},c}(t) = t\lambda_c \mathbf{S}_c^{-1} H_S \mathbf{S}_c + H_{\text{err},c}. \quad (\text{C11})$$

Note, that conjugation of the system Hamiltonian H_S with single-qubit operations as in $\mathbf{S}_c^{-1} H_S \mathbf{S}_c$ and $H_{\text{err},c}$ always preserves the locality and thus the interaction graph of H_S .

Finally, we prove the error bounds on the truncation of the Magnus expansion after the first order. The k -th order term of the Magnus expansion can be written as

$$H_{\text{av}}^{(k)} = \sum_{\sigma \in S_k} (-1)^{d_b} \frac{d_a! d_b!}{k!} \int_0^T d\tau_1 \int_0^{\tau_1} d\tau_2 \cdots \int_0^{\tau_{k-1}} d\tau_k H_c(\tau_{\sigma(1)}) H_c(\tau_{\sigma(2)}) \cdots H_c(\tau_{\sigma(k)}), \quad (\text{C12})$$

where $T = 2T_p + t\lambda_c$, S_k denotes the group of permutations σ of the set $[k]$ the number of ascents d_a and descents d_b are defined as

$$d_a := |\{i \in [k-1] \mid \sigma(i) < \sigma(i+1)\}| \quad \text{and} \quad d_b := |\{i \in [k-1] \mid \sigma(i) > \sigma(i+1)\}|, \quad (\text{C13})$$

and it holds that $d_a + d_b = k$ [78]. Finally, we argue that

$$\|H_{\text{av}}^{(k)}\| \leq O\left((2T_p + t\lambda_c)^k \max_{\tau \in [0, 2T_p + t\lambda_c]} \|H_c(\tau)\|^k\right) = O((2T_p + t\lambda_c)^k \|H_S\|^k), \quad (\text{C14})$$

where the equality follows from the definition of $H_c(\tau) = U^{-1} H_S U$ for some unitaries U and the invariance of the spectral norm under unitary transformation. The error bound in spectral norm follows from the Duhamel principle in Ref. [74, App. A]

$$\|U(t\lambda_c) - e^{-iH_{\text{av},c}(t)}\| \leq \sum_{k=2}^{\infty} \|H_{\text{av}}^{(k)}\| = O((2T_p + t\lambda_c)^2 \|H_S\|^2), \quad (\text{C15})$$

assuming that $\|H_{\text{av}}^{(2)}\| > \sum_{k=3}^{\infty} \|H_{\text{av}}^{(k)}\|$, i.e. the Magnus expansion converges. \square

Lemma IV.3. Let $W(\mathbf{J})^{(r \times s)}, E(\mathbf{J})^{(r \times s)} \in \mathbb{R}^{r \times s}$ be the matrices representing the ideal conjugation and the finite pulse time error in the Pauli basis respectively. The entries are given by

$$W(\mathbf{J})_{\mathbf{ac}}^{(r \times s)} := \frac{1}{2^n} \text{Tr} \left(P_{\mathbf{a}} \left(\mathbf{S}_c^\dagger H_S \mathbf{S}_c \right) \right), \quad (\text{49})$$

and

$$E(\mathbf{J})_{\mathbf{ac}}^{(r \times s)} := \frac{1}{2^n} \text{Tr} (P_{\mathbf{a}} H_{\text{err},c}), \quad (\text{50})$$

and can be calculated in polynomial time in the number of qubits for a local system Hamiltonian H_S .

Proof. Recall, that the system Hamiltonian has the form

$$H_S = \sum_{\mathbf{a} \in \mathbb{F}_2^{2^n} \setminus \{0\}} J_{\mathbf{a}} P_{\mathbf{a}}. \quad (\text{C16})$$

Let $H_{c^{(\ell)}} = \frac{\theta^{(\ell)}}{t_p^{(\ell)}} \sum_{i=1}^n (-1)^{s_i^{(\ell)}} h_i^{(\ell)}$ be an arbitrary layer of single-qubit rotations, with the rotation angle $\theta^{(\ell)}$, the pulse duration $t_p^{(\ell)}$ the generators $h_i^{(\ell)} := p_{x,i}^{(\ell)} X + p_{y,i}^{(\ell)} Y + p_{z,i}^{(\ell)} Z$ and the rotation direction $s_i^{(\ell)} \in \mathbb{F}_2$. For the sake of a clear notation we omit the layer index (ℓ) for now. The single-qubit pulse on the i -th qubit can be written as

$$S_{c_i}(t) := e^{-it \frac{\theta}{t_p} h_i} = \cos(t \frac{\theta}{t_p}) I - i(-1)^{s_i} \sin(t \frac{\theta}{t_p}) h_i. \quad (\text{C17})$$

Then, $\mathbf{S}_c(t) = e^{-itH_c} = \bigotimes_{i=1}^n S_{c_i}(t)$. The conjugation with a single-qubit pulse layer changes the interaction term as

$$\mathbf{S}_c^{-1}(t) J_{\mathbf{a}} P_{\mathbf{a}} \mathbf{S}_c(t) = J_{\mathbf{a}} \bigotimes_{i=1}^n S_{c_i}^{-1}(t) P_{\mathbf{a}_i} S_{c_i}(t). \quad (\text{C18})$$

The effect of such a conjugation on the i -th qubit is given by

$$S_{c_i}^{-1}(t) P_{\mathbf{a}_i} S_{c_i}(t) = \cos^2(\tilde{\theta}) P_{\mathbf{a}_i} + \sin^2(\tilde{\theta}) h_i P_{\mathbf{a}_i} h_i + i[h_i, P_{\mathbf{a}_i}] (-1)^{s_i} \sin(\tilde{\theta}) \cos(\tilde{\theta}), \quad (\text{C19})$$

with $\tilde{\theta} := t \frac{\theta}{t_p}$. Equation (C19) can be further decomposed into Pauli terms $S_{c_i}^{-1}(t)P_{\mathbf{a}_i}S_{c_i}(t) = g_{x,i}(\tilde{\theta})X + g_{y,i}(\tilde{\theta})Y + g_{z,i}(\tilde{\theta})Z$, with

$$\begin{aligned} g_{x,i}(\tilde{\theta}) &:= \begin{cases} \cos^2(\tilde{\theta}) + (p_{x,i}^2 - p_{y,i}^2 - p_{z,i}^2) \sin^2(\tilde{\theta}), & \text{if } P_{\mathbf{a}_i} = X \\ 2p_{x,i}p_{y,i} \sin^2(\tilde{\theta}) + 2p_{z,i}(-1)^{s_i} \sin(\tilde{\theta}) \cos(\tilde{\theta}), & \text{if } P_{\mathbf{a}_i} = Y \\ 2p_{x,i}p_{z,i} \sin^2(\tilde{\theta}) - 2p_{y,i}(-1)^{s_i} \sin(\tilde{\theta}) \cos(\tilde{\theta}), & \text{if } P_{\mathbf{a}_i} = Z, \end{cases} \\ g_{y,i}(\tilde{\theta}) &:= \begin{cases} 2p_{x,i}p_{y,i} \sin^2(\tilde{\theta}) + 2p_{z,i}(-1)^{s_i} \sin(\tilde{\theta}) \cos(\tilde{\theta}), & \text{if } P_{\mathbf{a}_i} = X \\ \cos^2(\tilde{\theta}) + (p_{y,i}^2 - p_{x,i}^2 - p_{z,i}^2) \sin^2(\tilde{\theta}), & \text{if } P_{\mathbf{a}_i} = Y \\ 2p_{y,i}p_{z,i} \sin^2(\tilde{\theta}) - 2p_{x,i}(-1)^{s_i} \sin(\tilde{\theta}) \cos(\tilde{\theta}), & \text{if } P_{\mathbf{a}_i} = Z, \end{cases} \\ g_{z,i}(\tilde{\theta}) &:= \begin{cases} 2p_{x,i}p_{z,i} \sin^2(\tilde{\theta}) - 2p_{y,i}(-1)^{s_i} \sin(\tilde{\theta}) \cos(\tilde{\theta}), & \text{if } P_{\mathbf{a}_i} = X \\ 2p_{y,i}p_{z,i} \sin^2(\tilde{\theta}) + 2p_{x,i}(-1)^{s_i} \sin(\tilde{\theta}) \cos(\tilde{\theta}), & \text{if } P_{\mathbf{a}_i} = Y \\ \cos^2(\tilde{\theta}) + (p_{z,i}^2 - p_{x,i}^2 - p_{y,i}^2) \sin^2(\tilde{\theta}), & \text{if } P_{\mathbf{a}_i} = Z, \end{cases} \end{aligned} \quad (\text{C20})$$

and if $P_{\mathbf{a}_i} = I$, then $S_{c_i}^{-1}(t)P_{\mathbf{a}_i}S_{c_i}(t) = I$. Assume, that the interaction $J_{\mathbf{a}}P_{\mathbf{a}}$ is k -local, i.e. the interaction acts on the qubits $i \in \text{supp}(\mathbf{a})$ and $|\text{supp}(\mathbf{a})| = k$. Applying $A \otimes (B + C) = A \otimes B + A \otimes C$ the conjugation in eq. (C18) yields

$$\bigotimes_{i=1}^n S_{c_i}^{-1}(t)P_{\mathbf{a}_i}S_{c_i}(t) = \sum_{\substack{\tilde{\mathbf{a}} \in \mathbb{F}_2^n \\ \text{supp}(\tilde{\mathbf{a}}) = \text{supp}(\mathbf{a})}} \left(\prod_{i \in \tilde{\mathbf{a}}} g_{\tilde{\mathbf{a}}_i, i}(\tilde{\theta}) \right) P_{\tilde{\mathbf{a}}}, \quad (\text{C21})$$

where we identify $g_{(1,0),i} = g_{x,i}$, $g_{(1,1),i} = g_{y,i}$ and $g_{(0,1),i} = g_{z,i}$. This sum has at most 3^k terms, which is constant for a system Hamiltonian H_S with a fixed interaction graph.

With that we are ready to compute $E(\mathbf{J})_{\mathbf{ac}}^{(r \times s)}$ by calculating the Pauli coefficients of

$$H_{\text{err}, \mathbf{c}} = 2 \sum_{\ell=1}^S \mathbf{S}_{\mathbf{c}^{\geq(\ell+1)}}^{-1} \int_0^{t_p^{(\ell)}} \mathbf{S}_{\mathbf{c}^{(\ell)}}^{-1}(t) H_S \mathbf{S}_{\mathbf{c}^{(\ell)}}(t) dt \mathbf{S}_{\mathbf{c}^{\geq(\ell+1)}}. \quad (\text{C22})$$

We start with the integral

$$\begin{aligned} \int_0^{t_p^{(\ell)}} \mathbf{S}_{\mathbf{c}^{(\ell)}}^{-1}(t) H_S \mathbf{S}_{\mathbf{c}^{(\ell)}}(t) dt &= \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} J_{\mathbf{a}} \int_0^{t_p^{(\ell)}} \bigotimes_{i=1}^n S_{c_i^{(\ell)}}^{-1}(t) P_{\mathbf{a}_i} S_{c_i^{(\ell)}}(t) dt \\ &= \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} \sum_{\substack{\tilde{\mathbf{a}} \in \mathbb{F}_2^n \\ \text{supp}(\tilde{\mathbf{a}}) = \text{supp}(\mathbf{a})}} J_{\mathbf{a}} \frac{t_p^{(\ell)}}{\theta^{(\ell)}} \int_0^{\theta^{(\ell)}} \left(\prod_{i \in \tilde{\mathbf{a}}} g_{\tilde{\mathbf{a}}_i, i}(\tilde{\theta}) \right) d\tilde{\theta} P_{\tilde{\mathbf{a}}} \\ &=: \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} E_{\mathbf{a}}^{(\ell)} P_{\mathbf{a}}, \end{aligned} \quad (\text{C23})$$

where $E_{\mathbf{a}}^{(\ell)}$ can be efficiently calculated by integrating trigonometric polynomials and summing all contributions from the terms with the same support $\text{supp}(\tilde{\mathbf{a}}) = \text{supp}(\mathbf{a})$. Next, the conjugation $\mathbf{S}_{\mathbf{c}^{\geq(\ell+1)}}^{-1}(\cdot) \mathbf{S}_{\mathbf{c}^{\geq(\ell+1)}}$ in eq. (C22) corresponds to applying eq. (C21) for each $\tilde{\ell} = 1, \dots, \ell + 1$ to $E_{\mathbf{a}}^{(\ell)} P_{\mathbf{a}}$ with $t = t_p^{(\tilde{\ell})}$ or $\tilde{\theta} = \theta^{(\tilde{\ell})}$. Then, we obtain

$$H_{\text{err}, \mathbf{c}} = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} E(\mathbf{J})_{\mathbf{ac}}^{(r \times s)} P_{\mathbf{a}}. \quad (\text{C24})$$

The entries of $W(\mathbf{J})^{(r \times s)}$ can be calculated similarly by applying eq. (C21) for each $\ell = 1, \dots, n_L$ to $J_{\mathbf{a}}P_{\mathbf{a}}$ with $t = t_p^{(\ell)}$ or $\tilde{\theta} = \theta^{(\ell)}$. Together, we obtain

$$\mathbf{S}_{\mathbf{c}}^{-1} H_S \mathbf{S}_{\mathbf{c}} = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} W(\mathbf{J})_{\mathbf{ac}}^{(r \times s)} P_{\mathbf{a}}. \quad (\text{C25})$$

□

Appendix D: Proofs for the robust Pauli conjugation method

Lemma IV.9. Consider all labels $\mathbf{a} \in \mathbb{F}_2^{2n}$ for the non-zero interactions $J_{\mathbf{a}} \neq 0$ in the system Hamiltonian. Then, the approximation of $U(t\lambda_{\mathbf{c}})$ for the Pauli conjugation in first order Magnus expansion is given by $e^{-iH_{\text{av},\mathbf{c}}(t)}$ with

$$H_{\text{av},\mathbf{c}}(t) = t\lambda_{\mathbf{c}}\mathbf{S}_{\mathbf{c}}^{\dagger}H_S\mathbf{S}_{\mathbf{c}} + H_{\text{err},\mathbf{c}}, \quad (54)$$

where

$$H_{\text{err},\mathbf{c}} = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} \left(J_{\mathbf{a}} E_{\mathbf{a},\mathbf{c}}^{(r \times s)} P_{\mathbf{a}} + R_{\mathbf{a},\mathbf{c}} \right). \quad (55)$$

We call the first term in eq. (55) the interaction term, with

$$E_{\mathbf{a},\mathbf{c}}^{(r \times s)} := \frac{4t_p}{\pi} \int_0^{\frac{\pi}{2}} \left(\prod_{i \in \text{supp}(\mathbf{a})} (\cos^2(\theta) + (-1)^{\langle \mathbf{a}_i, \mathbf{b}_i \rangle} \sin^2(\theta)) \right) d\theta, \quad (56)$$

which we collect as entries of the matrix $E^{(r \times s)} \in \mathbb{R}^{r \times s}$. We call the second term $R_{\mathbf{a},\mathbf{c}}$ the rest term, and it is proportional to $(-1)^{\mathbf{e}_{\mathbf{a}} \cdot \mathbf{s}}$, with $\mathbf{s} \in \mathbb{F}_2^n$ representing the chosen rotation direction of the π pulses and $\mathbf{e}_{\mathbf{a}} \in \mathbb{F}_2^n$ encodes the sign flips due to the finite pulse time error such that $e_{\mathbf{a},i} = 0$ for all $i \notin \text{supp}(\mathbf{a})$.

Proof. From lemma IV.2 with $n_L = 1$ and $H_{\mathbf{c}} = H_{c(1)}$ we get the Hamiltonian corresponding to the finite pulse time effect

$$H_{\text{err},\mathbf{c}} := 2 \int_0^{t_p} e^{itH_{\mathbf{c}}} H_S e^{-itH_{\mathbf{c}}} dt. \quad (D1)$$

Recall, that the system Hamiltonian has the form

$$H_S = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} J_{\mathbf{a}} P_{\mathbf{a}}. \quad (D2)$$

Before we compute $H_{\text{err},\mathbf{c}}$ we investigate the conjugation $e^{itH_{\mathbf{c}}} H_S e^{-itH_{\mathbf{c}}}$ for a single-qubit. A π pulse on the i -th qubit can be written as

$$S_{c_i}(t) := e^{-it\frac{\pi}{2t_p}(-1)^{s_i}P_{\mathbf{b}_i}} = \cos\left(\frac{\pi}{2}\frac{t}{t_p}\right) I - i(-1)^{s_i}P_{\mathbf{b}_i} \sin\left(\frac{\pi}{2}\frac{t}{t_p}\right) = \cos(\theta)I - i(-1)^{s_i}P_{\mathbf{b}_i} \sin(\theta), \quad (D3)$$

with the Pauli generator $P_{\mathbf{b}}$ from eq. (11) and $\theta(t) := \frac{\pi}{2}\frac{t}{t_p}$. Then the effect of the conjugation on the i -th qubit is given by

$$\begin{aligned} S_{c_i}^{-1}(\theta)P_{\mathbf{a}_i}S_{c_i}(\theta) &= \underbrace{\left(\cos^2(\theta) + (-1)^{\langle \mathbf{a}_i, \mathbf{b}_i \rangle} \sin^2(\theta)\right)}_{=: \alpha_{\mathbf{a}_i, \mathbf{b}_i}(\theta)} P_{\mathbf{a}_i} + (-1)^{s_i} \underbrace{\cos(\theta) \sin(\theta)}_{\beta(\theta)} i[P_{\mathbf{b}_i}, P_{\mathbf{a}_i}] \\ &= \alpha_{\mathbf{a}_i, \mathbf{b}_i}(\theta)P_{\mathbf{a}_i} + (-1)^{s_i} \beta(\theta) i[P_{\mathbf{b}_i}, P_{\mathbf{a}_i}], \end{aligned} \quad (D4)$$

with the binary symplectic form $\langle \cdot, \cdot \rangle$ from eq. (2) and the commutator $[\cdot, \cdot]$. Let

$$F(e) := \begin{cases} P_{\mathbf{a}} \alpha_{\mathbf{a}_i, \mathbf{b}_i}(\theta), & e = 0 \\ i[P_{\mathbf{b}_i}, P_{\mathbf{a}_i}] \beta(\theta), & e = 1. \end{cases} \quad (D5)$$

Inserting H_S in eq. (D1) and applying $A \otimes (B + C) = A \otimes B + A \otimes C$ yields

$$\begin{aligned}
H_{\text{err},c} &= \frac{4t_p}{\pi} \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} J_{\mathbf{a}} \int_0^{\frac{\pi}{2}} \left(\bigotimes_{i \in \text{supp}(\mathbf{a})} (P_{\mathbf{a}_i} \alpha_{\mathbf{a}_i, \mathbf{b}_i}(\theta) + i[P_{\mathbf{b}_i}, P_{\mathbf{a}_i}]\beta(\theta)(-1)^{s_i}) \right) d\theta \\
&= \frac{4t_p}{\pi} \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} \left(J_{\mathbf{a}} \int_0^{\frac{\pi}{2}} \prod_{i \in \text{supp}(\mathbf{a})} \alpha_{\mathbf{a}_i, \mathbf{b}_i}(\theta) d\theta P_{\mathbf{a}} + \sum_{\substack{\mathbf{e} \in \mathbb{F}_2^n \setminus \{\mathbf{0}\} \\ e_i = 0 \ \forall i \notin \text{supp}(\mathbf{a})}} \left(\prod_{i \in \text{supp}(\mathbf{a})} (-1)^{e_i + s_i} \right) \left(\bigotimes_{i \in \text{supp}(\mathbf{a})} F(e_i) \right) \right) \\
&= \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} J_{\mathbf{a}} E_{\mathbf{a},c}^{(r \times s)} P_{\mathbf{a}} + \sum_{\substack{\mathbf{e} \in \mathbb{F}_2^n \setminus \{\mathbf{0}\} \\ e_i = 0 \ \forall i \notin \text{supp}(\mathbf{a})}} (-1)^{\mathbf{e} \cdot \mathbf{s}} \bigotimes_{i \in \text{supp}(\mathbf{a})} F(e_i).
\end{aligned} \tag{D6}$$

Moreover, we define $E_{\mathbf{a},c}^{(r \times s)} := \frac{4t_p}{\pi} \int_0^{\frac{\pi}{2}} \left(\prod_{i \in \text{supp}(\mathbf{a})} \alpha_{\mathbf{a}_i, \mathbf{b}_i}(\theta) \right) d\theta$. To conclude, we have the average Hamiltonian

$$H_{\text{av},c}(t) = t\lambda_{\mathbf{c}} \mathbf{S}_{\mathbf{c}}^{\dagger} H_S \mathbf{S}_{\mathbf{c}} + H_{\text{err},c}, \tag{D7}$$

from lemma IV.2, and we calculated the finite pulse time error term

$$H_{\text{err},c} = \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} \left(J_{\mathbf{a}} E_{\mathbf{a},c}^{(r \times s)} P_{\mathbf{a}} + R_{\mathbf{a},c} \right), \tag{D8}$$

with

$$E_{\mathbf{a},c}^{(r \times s)} := \frac{4t_p}{\pi} \int_0^{\frac{\pi}{2}} \left(\prod_{i \in \text{supp}(\mathbf{a})} (\cos^2(\theta) + (-1)^{\langle \mathbf{a}_i, \mathbf{b}_i \rangle} \sin^2(\theta)) \right) d\theta \tag{D9}$$

and

$$R_{\mathbf{a},c} := \sum_{\substack{\mathbf{e} \in \mathbb{F}_2^n \setminus \{\mathbf{0}\} \\ e_i = 0 \ \forall i \notin \text{supp}(\mathbf{a})}} (-1)^{\mathbf{e} \cdot \mathbf{s}} \bigotimes_{i \in \text{supp}(\mathbf{a})} F(e_i), \tag{D10}$$

which proofs the lemma. \square

Proposition IV.11. *Let the rotation directions $\mathbf{s} \in \mathbb{F}_2^n$ of the π pulses such that $\sum_{\mathbf{s}} (-1)^{\mathbf{e}_{\mathbf{a}} \cdot \mathbf{s}} = 0$ for all $\mathbf{e}_{\mathbf{a}} \in \mathbb{F}_2^n$ with $e_{\mathbf{a},i} = 0$ for all $i \notin \text{supp}(\mathbf{a})$ for any \mathbf{a} with $J_{\mathbf{a}} \neq 0$, as in lemma IV.9. Then, the rotation angle errors in the first order Taylor approximation cancels.*

Proof. We model the rotation angle error of a perfect π pulse $e^{-i(-1)^{s_i} \frac{\pi}{2} P_{\mathbf{b}_i}}$ with the rotation direction $(-1)^{s_i}$ and $s_i \in \mathbb{F}_2$ by

$$\tilde{S}_{c_i} = e^{-i(-1)^{s_i} \frac{\pi + \varepsilon_i}{2} P_{\mathbf{b}_i}} = -\frac{\varepsilon_i}{2} \text{Id} - i(-1)^{s_i} P_{\mathbf{b}_i} + O(\varepsilon_i^2), \tag{D11}$$

where we used the first-order Taylor expansion of sine and cosine. Conjugating a Pauli operator with an imperfect Pauli pulse results in

$$\tilde{S}_{c_i}^{\dagger} P_{\mathbf{a}_i} \tilde{S}_{c_i} = P_{\mathbf{b}_i} P_{\mathbf{a}_i} P_{\mathbf{b}_i} + i(-1)^{s_i} \frac{\varepsilon_i}{2} [P_{\mathbf{a}_i}, P_{\mathbf{b}_i}] + O(\varepsilon_i^2). \tag{D12}$$

We use the identity $U^{\dagger} e^{-itH} U = e^{-itU^{\dagger} H U}$ to compute the effective Hamiltonian of a system Hamiltonian H_S conjugated with a layer of imperfect Pauli pulses $\tilde{\mathbf{S}}_{\mathbf{c}} = \bigotimes_{i=1}^n \tilde{S}_{c_i}$,

$$\begin{aligned}
\tilde{\mathbf{S}}_{\mathbf{c}}^{\dagger} H_S \tilde{\mathbf{S}}_{\mathbf{c}} &= \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} J_{\mathbf{a}} \bigotimes_{i=1}^n \left(\tilde{S}_{c_i}^{\dagger} P_{\mathbf{a}_i} \tilde{S}_{c_i} \right) \\
&= \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} J_{\mathbf{a}} \bigotimes_{i=1}^n \left(P_{\mathbf{b}_i} P_{\mathbf{a}_i} P_{\mathbf{b}_i} + i(-1)^{s_i} \frac{\varepsilon_i}{2} [P_{\mathbf{a}_i}, P_{\mathbf{b}_i}] + O(\varepsilon_i^2) \right) \\
&= \sum_{\mathbf{a} \in \mathbb{F}_2^{2n} \setminus \{\mathbf{0}\}} \left(J_{\mathbf{a}} P_{\mathbf{b}} P_{\mathbf{a}} P_{\mathbf{b}} + \sum_{\substack{\mathbf{e} \in \mathbb{F}_2^n \\ e_i = 0 \ \forall i \notin \text{supp}(\mathbf{a})}} \left(\prod_{i \in \text{supp}(\mathbf{a})} (-1)^{e_i + s_i} \varepsilon_i^{e_i} \right) \left(\bigotimes_{i \in \text{supp}(\mathbf{a})} F(e_i) + O(\varepsilon_i^2) \right) \right),
\end{aligned} \tag{D13}$$

where we used $A \otimes (B + C) = A \otimes B + A \otimes C$ and

$$F(e) := \begin{cases} P_{b_i} P_{a_i} P_{b_i}, & e = 0 \\ i_{\frac{1}{2}}[P_{a_i}, P_{b_i}], & e = 1. \end{cases} \quad (\text{D14})$$

The second term in the sum is the first-order angle error contribution, and has the same sign structure $(-1)^{e_i + s_i}$ as the rest term $R_{a,c}$ in eq. (D6). Therefore, choosing the signs $\mathbf{s} \in \mathbb{F}_2^n$, such that the rest term $R_{a,c}$ cancels, simultaneously cancels the first order angle error contribution in eq. (D13). \square

The proof of the following result is similar as the proof in [37, Lemma 8].

Proposition IV.12. *Let $\kappa = 2^{\lceil \log_2(n+1) \rceil} \leq 2n$ and let $W^{(\kappa \times \kappa)}$ be the $\kappa \times \kappa$ dimensional Walsh-Hadamard matrix. Choose n distinct columns from $W^{(\kappa \times \kappa)}$ without the first column and define the resulting partial Walsh-Hadamard matrix as $W^{(\kappa \times n)}$. Let $(-1)^{\mathbf{s}^{(j)}}$ be the j -th row of $W^{(\kappa \times n)}$. Then, for $\mathbf{a} \in \mathbb{F}_2^{2n}$ and $\mathbf{e}_a \in \mathbb{F}_2^n$ with $|\text{supp}(\mathbf{a})| = 2$ and $e_{a,i} = 0$ for all $i \notin \text{supp}(\mathbf{a})$ we obtain $\sum_{j=1}^{\kappa} (-1)^{\mathbf{e}_a \cdot \mathbf{s}^{(j)}} = 0$.*

Proof. From the choice of $(-1)^{\mathbf{s}^{(j)}}$ it directly follows that $\sum_{j=1}^{\kappa} (-1)^{\mathbf{e} \cdot \mathbf{s}^{(j)}} = 0$ for all $\mathbf{e} \in \mathbb{F}_2^n$ with $|\mathbf{e}| = 1$, since the sum over all rows of a Walsh-Hadamard matrix especially $W^{(\kappa \times n)}$ is zero. We now have to show that $\sum_{j=1}^{\kappa} (-1)^{\mathbf{e} \cdot \mathbf{s}^{(j)}} = 0$ for all $\mathbf{e} \in \mathbb{F}_2^n$ with $|\mathbf{e}| = 2$. For any $\mathbf{e} \in \mathbb{F}_2^n$ with $|\mathbf{e}| = 2$ we can write $(-1)^{\mathbf{e} \cdot \mathbf{s}} = (-1)^{s_i} (-1)^{s_k} = \left(((-1)^{\mathbf{s}}) ((-1)^{\mathbf{s}})^T \right)_{ik}$ with $i, k \in [n]$ and $i \neq k$. Then, we obtain

$$\sum_{j=1}^{\kappa} (-1)^{\mathbf{e} \cdot \mathbf{s}^{(j)}} = \sum_{j=1}^{\kappa} \left(((-1)^{\mathbf{s}^{(j)}}) ((-1)^{\mathbf{s}^{(j)}})^T \right)_{ik}, \quad (\text{D15})$$

with $i, k \in [n]$ and $i \neq k$. The orthogonality property of the Walsh-Hadamard matrix yields

$$\sum_{j=1}^{\kappa} ((-1)^{\mathbf{s}^{(j)}}) ((-1)^{\mathbf{s}^{(j)}})^T = (W^{(\kappa \times n)})^T W^{(\kappa \times n)} = \kappa I, \quad (\text{D16})$$

where the non-diagonal entries $i \neq k$ correspond to the sum eq. (D15) and are zero. \square