ABSTRACT

Non-Classical Methods for the Simulation and Design of Quantum Materials

Colin M. Burdine, Ph.D.

Mentor: Enrique P. Blair, Ph.D.

Materials science has transformed human technology, ranging from robust structural materials like steel alloys to the sophisticated semiconductor devices that drive modern electronics. Many of these materials can be studied and understood computationally through classical methods such as Density Functional Theory, which apply traditional first-principles simulations of material physics. However, the simulation of quantum materials (whose properties are fundamentally shaped by quantum mechanics) poses a significant challenge due to their intricate electronic structure and unique emergent phenomena, such as magnetism, topological states, and superconductivity. Classical methods often struggle to accurately model these systems, especially when quantum effects such as strong electron correlations or entanglement are involved. This has manifested as a significant research gap between the computational modeling of these materials and the data obtained through experimental measurements.

This dissertation investigates non-classical approaches, specifically machine learning (ML) and quantum computing, to address these limitations. ML enables efficient simulation and prediction by uncovering hidden patterns in quantum systems,

while quantum computing leverages the principles of quantum mechanics to simulate materials at a fundamentally precise level. Specifically, we introduce novel machine learning models for high-throughput screening of superconducting materials and the prediction of electronic band structure in solid state materials. We also present ongoing work applying this model to large and complex solid state systems and devices. We also introduce a new quantum computational method for efficiently simulating open quantum systems and leverage this method to simulate selected systems on modern quantum hardware.

By developing novel ML architectures and quantum algorithms, this work aims to push the boundaries of quantum material simulation, bridging the gap between theory and experiment and accelerating the discovery of novel and potentially transformative materials.

Hold for signature page.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am profoundly grateful to my advisor, Dr. Erik Blair, for his gentle guidance and rigorous standards. Your mentorship has challenged me to think more deeply, work more diligently, listen more carefully, and speak with clarity and conviction. I would also like to thank the members of my dissertation committee and the Electrical and Computer Engineering department for their thoughtful feedback, encouragement, and for holding me to the high standards that have strengthened this work. Each discussion, critique, and suggestion has been an opportunity to grow not only as a scholar but also as a person.

To my colleagues, collaborators, and mentors across the engineering, physics, computer science, mathematics, and chemistry departments: thank you for your intellectual generosity and the countless conversations that helped me clarify ideas, uncover mistakes, and broaden my perspective.

To my friends: your support has kept me grounded through the long stretches of work and the successes and failures of graduate school. Time and again, you remind me that there is more to be enjoyed in life than the fruits of the mind's labor.

I am deeply appreciative of my family for their unwavering support. Mom and Dad, I am grateful for your constant encouragement and timely advice; I would not have made it this far without you and the sacrifices you have made. I am also thankful to my older brother and younger brother for their patience, support, and willingness to endure explanations of topics I sometimes struggled to articulate.

Finally, to all my teachers, past and present: thank you for showing me the value of questioning, for nurturing my curiosity, and for demonstrating that the path to understanding is rarely straight. You have reminded me that curiosity is most valuable when shared, and that growth is measured in resilience as much as in results. If this dissertation reflects anything, I hope it reflects the lessons you taught me– not just about science, but about the habits, persistence, and perspectives required to live a life of virtue and purpose.

To all my teachers, who showed the way

ATTRIBUTIONS

Below I outline my own contributions and those of my collaborators in the published chapters of this dissertation:

Chapter 4:

I would like to acknowledge help from my advisor and co-author Dr. Enrique P. Blair, who assisted with setting up the Quantum ESPRESSO calculations carried out in this paper, and revising the manuscript. I would also like to acknowledge Dr. Julia Y. Chan and her students (Dr. Kice Brown, Dr. Alexis Dominguez, Melissa Anderson, and Mario Plata) of the Department of Chemistry & Biochemistry and Materials Science at Baylor University for subsequent collaborative efforts carried out to synthesize superconductors from the list of candidates identified in this study.

Chapter 7:

I would like to acknowledge help from co-author Dr. Enrique P. Blair for assisting with the revision and formatting of the manuscript. I would like to thank Dr. Gerald Cleaver of the Department of Physics at Baylor University for his insightful discussion and proofreading of the early manuscript.

Chapter 8:

I would like to acknowledge help from co-author Dr. Enrique P. Blair for assisting with conceptualization, and facilitating contact with other co-authors. I developed the underlying theory, quantum code, classical-post-processing code, and planned all

of the quantum experiments in this study. Nora Bauer, and Dr. George Siopsis at the University of Tennessee Knoxville carried out the Quantinuum hardware experiments using the resources at the Oak Ridge Leadership Computing Facility. Experiments on the IonQ were carried out by myself through the Microsoft Azure Quantum program. All authors contributed to the writing and review of the manuscript.

Nora Bauer and Dr. George Siopsis acknowledge support by the National Science Foundation under award DGE-2152168 and the Department of Energy under awards DE-SC0024325 and DE-SC0024328. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. This research was also supported by compute credits from IonQ via the Microsoft Azure Quantum Credits program.

CHAPTER ONE

Introduction

1.1 Materials Science

Materials have played a preeminent role in the development of human technology, leading to the cultivation of societies, the development of new industries and economies, and the expansion of human prosperity. Since the first tools were developed in the Stone Age, innovations in our understanding and harnessing of matter have enabled new ways of living, revolutionizing the ways in which we work, communicate, and interact with the world. The discovery of new materials is constantly reshaping the world around us, spanning discoveries like steel alloys and silicate glass that enabled the construction of industrial-age skyscrapers, to the semiconductor technology that underpins modern electronics and microprocessors. Today, the science of materials has evolved into a mature field, encompassing a diverse array of scientific disciplines stemming from the myriad applications that materials have found in modern life.

The interdisciplinary scope of materials science has expanded into a sophisticated and multifaceted field, incorporating insights from physics, chemistry, biology and engineering to design new materials, or tailor existing ones for new applications. As a result, materials science is no longer merely about discovering and developing new materials but about understanding and engineering materials on an atomic and molecular level. This shift has allowed for unprecedented precision in designing materials with specific properties, enabling advances in fields such as renewable energy,

biomedicine, and aerospace engineering [1–3]. However, understanding and modeling materials at the atomic level is not without its challenges; indeed, as the scale at which one wishes to characterize materials reaches the atomic level, one must contend with the laws of quantum mechanics, which are known to give rise to a rich set of behaviors and properties that can be difficult to predict and intuitively understand. This inherent difficulty of studying materials at the atomic level as quantum-mechanical systems arises at both the experimental and theoretical levels.

At the experimental level, probing materials at the atomic scale often requires specialized techniques, such as scanning tunneling microscopy [4], atomic force microscopy [5], and ultrafast or coherent X-ray spectroscopy [6, 7], each of which has inherent limitations in resolution, sensitivity, and even the kinds of observable properties that can be measured simultaneously. Furthermore, experimental conditions must be tightly controlled, as materials exhibiting quantum-mechanical phenomena are often highly sensitive to temperature, pressure, and electromagnetic fields. As a result, isolating quantum effects or probing transient quantum states remains challenging, leading to data that can be difficult to interpret, reproduce, and reconcile with theoretical models.

At the theoretical level, many significant challenges also arise from the quantum-mechanical treatment of materials. Most notably, modeling quantum materials requires finding precise solutions to complex differential equations, such as the Schrödinger equation, which become computationally intractable for systems with large numbers of particles. Approximations and models, such as Kohn-Sham density functional theory [8], are commonly used to simplify calculations, but they have known limitations

and may not capture all relevant quantum interactions, especially for strongly correlated materials or systems with intricate magnetic or electronic structures. Moreover, emergent phenomena, such as entanglement, superconductivity, and topological states, introduce additional layers of complexity, requiring new mathematical frameworks and algorithms. Thus, despite advances in both experimental techniques and computational power, fully understanding materials under a rigorous quantum-mechanical treatment remains a formidable challenge.

1.1.1 Quantum Materials

Although the macroscopic properties of many materials can be explained using models based on Newtonian physics and classical electromagnetic theories, some materials exhibit properties that are characteristically quantum-mechanical in nature. In the scientific literature, these materials are often loosely categorized as *quantum materials*. Quantum materials are distinguished by properties that emerge from interactions at the atomic and subatomic levels, with many even exhibiting macroscopic phenomena that diverge significantly from those predicted by classical physics [9]. Among these are the resistance-free electron flow observed in superconductors [10] and the unique conductive surfaces of topological insulators [11], which retain insulating properties within their bulk. The more commonplace phenomenon of magnetism is also a quantum effect, arising from the spontaneous ordering of a material's electronic spin degrees of freedom. These characteristics are potentially transformative in the design of new kinds of devices and composite materials, especially at the nanoscopic and mesoscopic scales.

1.2 Motivation

Traditional models in materials science, often grounded in classical electromagnetic theory and mean-field approximations, have been applied successfully for decades to explain the behavior of many different materials. However, when applied to quantum materials, the predictive power of these models tend to fall short of explaining experimental measurements. These classical models tend to rely on simplifying assumptions, such as treating indistinguishable particles as effectively independent and non-interacting. These assumptions are quite useful, allowing for intractably complex interactions between particles to be reduced into effective mean-field models that are much less computationally demanding and allow for larger systems to be simulated without exorbitant computational costs. However, quantum materials exhibit phenomena like superconductivity, quantum magnetism, and topological states, which emerge from collective behaviors, entanglement, and electron correlations that cannot be captured using such simplifications. As a result, understanding and predicting the properties of quantum materials require approaches that account for the deeply interdependent and probabilistic nature of quantum mechanics, motivating the use of more sophisticated computational methods.

1.2.1 Classical Methods

To motivate the discussion of quantum materials, we will first discuss established computational methods which can be applied to simulate materials using effective classical models that forego a full quantum treatment of materials in favor of simplified mean-field models. Throughout this dissertation, we refer to these methods as *classical methods*. Classical methods, such as Hartree-Fock (HF) [12] and density

functional theory (DFT) [8], are widely used mean-field approaches that approximate complex quantum systems by reducing many-body quantum interactions between electrons to an averaged, effective field. In these methods, the complex interactions between electrons are simplified, with HF treating electron-electron interactions in a way that ignores electron correlation, while DFT approximates the system's energy by approximating the effects of electron correlation as closed-form functionals of electron density. These techniques offer powerful tools for modeling materials and have become foundational in materials science due to their computational efficiency and capacity to capture a range of electronic properties. Furthermore, in the treatment of periodic solid state crystals, various tight-binding models can also be applied, to further simplify the crystal momentum degrees of freedom that arise in periodic systems. However, because all of these approaches rely on mean-field approximations, they often struggle to accurately describe materials where electron correlation and quantum entanglement play a significant role. As a result, while HF and DFT remain useful for many applications, fully quantum treatments are often necessary for understanding the nuanced behaviors that arise in strongly correlated systems and quantum materials.

To provide a more rigorous quantum treatment of systems, more computationally expensive models can also be applied. Among these are popular approaches such as the density matrix renormalization group (DMRG) [13], the complete active space self-consistent field (CASSCF) [14], and the full configuration interaction (Full CI) methods [15]. However, these methods generally require resources that grow exponentially both with the system size and more generally with the degree of theoretical

rigor required. As a result, they are often too expensive for periodic solid-state systems, and can only be feasibly applied to isolated molecules or toy models of spin systems. In Figure 1.1, we illustrate the trade-off between less rigorous low-cost computational methods and very rigorous high-cost methods. The trade-off for classical methods (the red line) exhibits a roughly exponential scaling of computational cost with respect to both the size of the system being considered, as well as the degree of theoretical rigor that is to be applied.

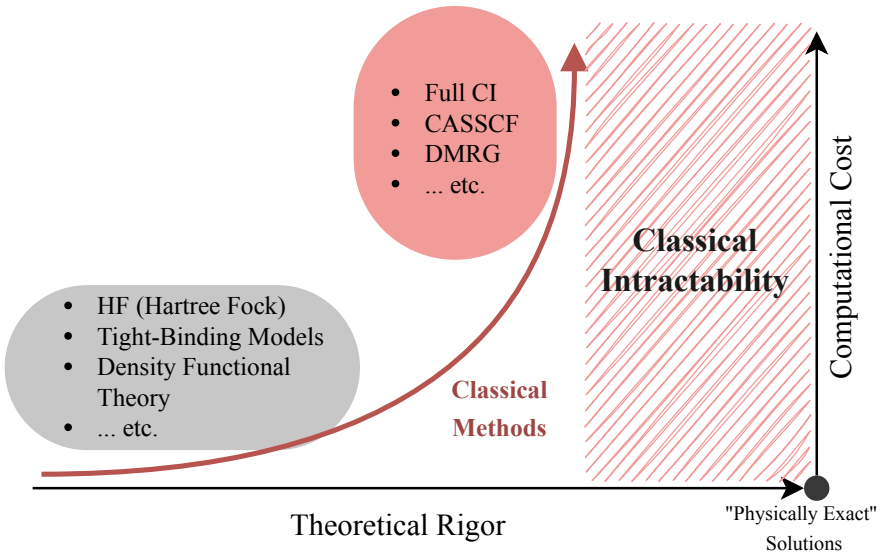

Figure 1.1: An illustration of the trade-off between theoretical rigor and computational cost when simulating materials via classical methods. Quantum materials often require a high degree of theoretical rigor to simulate, which cannot be easily attained through classical methods without significant computational expense. As a result, these materials tend to occupy a region of "classical intractability".

The unforgiving nature of the exponential scaling of classical methods implies that there exist certain systems (and certain scales at which these systems are simulated) for which the computational budget required by classical methods exceeds that of even state-of-the-art exascale supercomputers. In Figure 1.1, this is designated by the region of "classical intractability". A large number quantum materials lie within

this region because existing classical methods fail to fully reproduce their properties that are measured experimentally. In Chapter 2, we give some additional technical details regarding these materials and explore some examples of quantum materials where classical methods fail.

1.2.2 Non-Classical Methods

In this dissertation we will explore alternatives to Classical methods, which we designate as *Non-Classical Methods*. Non-Classical methods employ emerging hardware technologies in an effort to simulate the physics of materials with greater precision and scale. We will primarily focus on two of these technologies: *machine learning* and *quantum computing*.

1.2.2.1 Machine learning. Today, machine learning (ML) is a mature technology that has proven to be an indispensable tool in the study of quantum materials, enabling researchers to overcome known limitations of classical methods on certain difficult physical problems. Due to the wide availability of data (both experimental and simulated), ML models can be trained to identify complex patterns in quantum-mechanical systems, and predict material properties with remarkable speed and accuracy. Especially for complex quantum materials like superconductors, magnets, and topological materials, ML has been applied successfully to understand physical phenomena in these systems that would otherwise require immense computational resources. In the area of materials simulation, ML models are most commonly used to replace expensive computational subroutines in classical simulation methods, such as DFT. For these tasks, a large and flexible model, such as a deep neural network, is trained on a set of known data points, and the model is used to provide accurate

output for that given subroutine at a much lower computational cost. In the case of DFT, for example, neural networks have been applied successfully to learn the exchange-correlation functional of electronic density, which is not known in closed form and is computationally expensive to even approximate numerically.

Machine learning already has an established track record in many areas of materials science, and has proven itself as a versatile and essential tool for accelerating material simulation and discovery. This is especially true when classical methods are augmented with ML methods, as illustrated in Figure 1.2.

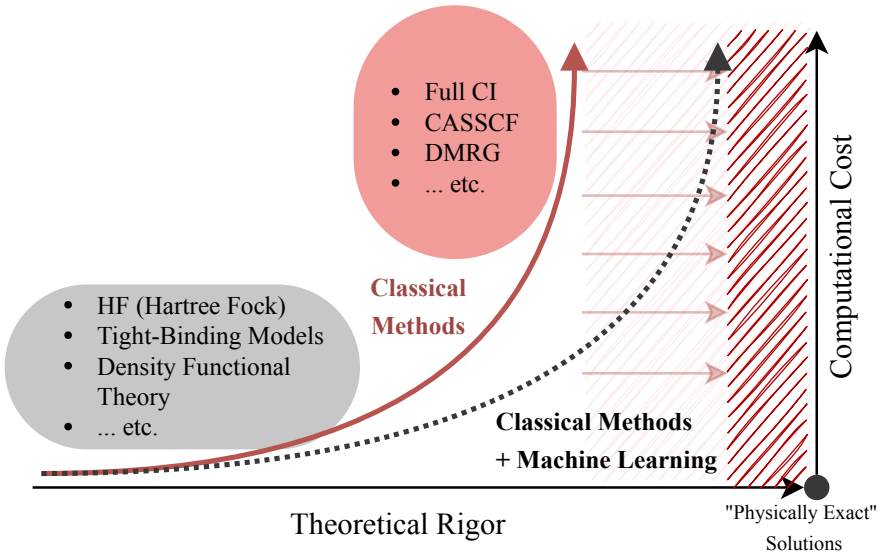

Figure 1.2: An illustration of how the non-classical method of machine learning can be applied to improve upon existing classical methods, allowing for increased physical rigor. By augmenting low-cost classical methods (grey bubble) with data trained on small-scale high-cost classical methods with quantum-level accuracy (red bubble), more complex systems can be simulated at a much lower computational cost.

By using computationally expensive classical methods to generate small-scale system training data, ML models have been successfully trained and applied to simulate systems of larger size and finer spatial resolution at low computational cost. Some

models have also been applied for the problem of direct property prediction, bypassing the computational cost of simulation altogether. As suggested in Figure 1.2, this allows for the exploration of materials that would otherwise require expensive simulations that inhabit the region of classical intractability.

Much like classical methods, ML methods also face some key challenges and limitations with quantum materials. First, quantum material studies require large, high-quality datasets, yet experimental and computational data on these systems are often sparse and noisy, hindering model reliability. Additionally, quantum materials tend to exhibit complex phenomena such as electron correlation and entanglement, which are challenging for ML models to capture, except through learning domain-specific heuristics that (at best) can only serve as effective mean-field theories. Finally, the most significant limitation of ML models is the problem of interpretability. Although ML models can make predictions with a substantial degree of accuracy (if given enough high-quality training data), these models frequently operate as "black boxes," yielding predictions without insights into the underlying physics of the materials under consideration.

1.2.2.2 Quantum computing. Quantum computing is an emerging but potentially revolutionary technology that leverages the fundamental principles of quantum mechanics to process information in ways that surpass the limits of traditional computing. Unlike traditional computers that use binary bits to represent information as 1's and 0's, quantum computers use qubits (quantum bits), which can exist in superpositions of 0 and 1 simultaneously. Moreover, individual qubits can become entangled to produce highly complex and correlated states. This enables quantum

computers to perform certain kinds of tasks in a manner that is highly parallelized and at scales that are not attainable on classical hardware. For example, 8 bits on a classical computer can be used to represent any integer from 0 to 255, whereas 8 qubits on a quantum computer can be used to encode a superposition (i.e., a probability distribution) over all possible numbers from 0 to 255. This exponential increase in the complexity of data stored on registers of qubits can be leveraged to achieve an exponential speedup over known classical computers for certain kinds of problems.

Although quantum computers can be applied to solve many kinds of problems, their capabilities in the near-term are especially promising for the problem of simulating quantum materials. Since the intricate electron interactions, correlations, and entanglements in these materials arise as a consequence of quantum mechanics, quantum computers can provide a much-needed computational advantage in simulating them. This idea is shown in Figure 1.3. With the development of quantum computational algorithms such as phase estimation, Hamiltonian simulation, and variational quantum eigensolver (VQE) methods, it has been shown theoretically that quantum methods for material simulation should yield an exponential speedup relative to classical methods; however, this quantum advantage remains to be demonstrated in many different practical settings.

Since quantum computing technology remains in an experimental phase, challenges still exist that must be solved to potentially realize any exponential advantage. These challenges include the limited number of qubits available on current hardware, as well as issues with qubit coherence and the error rates of single-qubit and multi-qubit operations, which restrict the accuracy and scalability of quantum simulations. Quantum computers are also highly sensitive to environmental noise, which

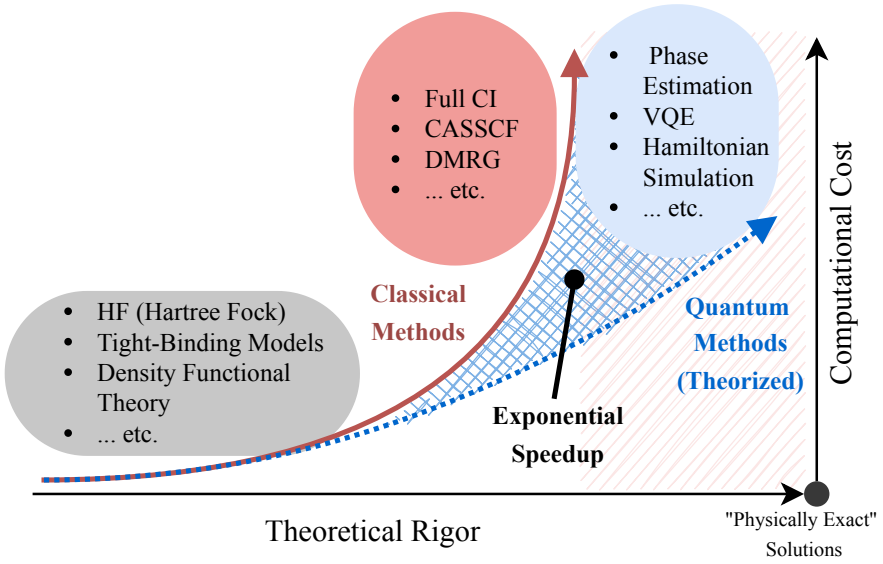

Figure 1.3: An illustration of how non-classical quantum computing methods could theoretically be applied to yield an exponential improvement over classical methods. Materials that require exponentially-scaling computational cost to simulate on standard computers are believed to only require only a linearly-scaling cost on quantum computers.

can disrupt calculations and degrade qubit coherence. These error syndromes are a significant problem when modeling complex materials, which require very precise calculations. Although there are ways of suppressing and eventually eliminating these errors entirely through quantum error correction protocols, these protocols require a scale and threshold of accuracy that has not yet been achieved on state-of-the-art hardware. In addition, many quantum algorithms for material simulation are in the early stages of development and tend to require hardware capabilities beyond what is currently available. However, finding hardware-friendly and fault-tolerant simulation algorithms remains an active area of research. Altogether, these technical and theoretical challenges limit the practical application of quantum computing in simulating quantum materials, though ongoing advancements in hardware and algorithm design continue to bring this potential closer to realization.

1.3 Problem Statement

As we have discussed, classical computational methods like density functional theory have been instrumental in advancing our understanding of materials and bridging the gap between *ab initio* theory and experiment. However, we have also noted the limitations of classical methods when applied to quantum materials, where quantum mechanical effects cannot be ignored. As a result, there is a pressing need for new approaches capable of capturing these quantum effects without sacrificing computational efficiency. Emerging non-classical methods, particularly machine learning and quantum computing, offer promising avenues to overcome these barriers, enabling more precise simulations and potentially accelerating the discovery and design of quantum materials. This dissertation seeks to explore and develop non-classical methods based in machine learning and quantum computing as potential solutions that may help bridge the gap between theory and experiment left by classical approaches in quantum material research.

1.3.1 Objectives and Scope

The landscape of quantum materials is vast, including many different types of materials. These range from high-temperature superconductors to topological materials, to exotic magnetic materials. As a result, an exhaustive survey of all quantum material phenomena would be too wide of a scope for any one book, and would likely constitute a few lifetimes' worth of work. Thus, we do not aim to address each of the pathologies of quantum materials individually in this dissertation; rather, we will focus on tools and selected methods that can be developed at a higher conceptual level and then demonstrate how they can be applied to a few specific (but well-curated)

examples. Additional examples will also be given that highlight the key contributions made to the scientific body of literature in the research leading up to this dissertation. Since there are two distinct branches of this work (ML methods and quantum computing methods), we give will give a separate treatment of both areas; though we will aim to highlight the strengths and weaknesses both in the individual context of their respective state-of-the-art hardware, and on the hardware that may be available in the future. Since quantum computing is a nascent technology, we will also refrain from attempting to make direct comparisons between the practical computational cost of classical versus quantum hardware, except in the case of idealized fault-tolerant quantum computation and idealized classical computation, where the two technologies can be made at least somewhat comparable.

1.3.2 Organization of this Dissertation

The remainder of this dissertation is organized as follows. In Chapter 2, we give an overview of quantum mechanics and the electronic structure problem under second quantization. We also provide a technical review of classical methods (such as Hartree-Fock and DFT), and how these methods can be applied in the context of property prediction and material discovery. We then conclude with a brief introduction to non-classical methods and provide a comprehensive literature review of these topics. In Chapter 3, we give a technical introduction to machine learning, with a heavy emphasis on models and deep neural network architectures that are suitable for material property prediction from crystal structure data. In Chapter 4, we present an atomistic graph neural network model for high-throughput superconductor discovery. This model classifies materials as superconducting and non-superconducting and

predicts the critical temperature for those classified as superconductors. In Chapter 5, we present a new physics-informed deep learning model architecture and leverage this model to predict the electronic structure of materials. This model directly incorporates the structure and symmetries of solutions to the Schrödinger equation for solid state materials, making it much more interpretable than existing models. We also show that this model can accurately and efficiently predict the band structure of solid state materials with complex electronic configurations.

In Chapter 6, we give a technical introduction to quantum computational methods for the simulation of material and chemical systems. We review the general theory of algorithms such as Hamiltonian simulation, variational quantum eigensolvers [], and phase estimation [], in addition to algorithms that can be run on near-term noisy quantum hardware [16]. We conclude this chapter with detailed discussion of the problem of open quantum system simulation and its relevance to simulating quantum materials. In Chapter 7, we propose a new method to simulate open quantum systems on near-term hardware. This method leverages a time-perturbative Kraus series representation of open quantum systems, which can significantly reduce computational cost. We show that this method can be applied efficiently on state-of-the-art hardware, allowing certain systems to be simulated more accurately than with existing methods. In Chapter 8, we apply this method to simulate selected physical systems on real trapped-ion quantum hardware. Finally in Chapter 9, we summarize the key ideas, theories, and results presented in this dissertation and give some comparative analysis of non-classical methods and a prospective outlook on future research directions that could be pursued.

In Figure 1.4, we illustrate the landscape of concepts discussed in the subsequent chapters of this dissertation with a conceptual mind map.

Figure 1.4: A conceptual mind map outlining the non-classical methods for quantum material simulation that are discussed in the subsequent chapters of this dissertation. In Chapter 2, we discuss the general ideas and theory behind non-classical methods (orange node). In Chapters 3-5, we discuss machine learning-based methods for materials discovery and simulation (red nodes). In Chapters 6-8, we discuss quantum computing-based methods, and how they can be applied to accelerate material simulation (blue nodes). Finally, in Chapter 9, we conclude the dissertation with a discussion of potential future directions and long-term impacts of non-classical methods in materials research (teal nodes).

CHAPTER TWO

Background and Motivation

2.1 Quantum Materials Research

2.1.1 Overview

Before we present the main body of work of this dissertation, we must first provide some necessary context on quantum materials. In this chapter, we will specifically focus on the foundations of quantum materials, discussing how their properties are governed by quantum mechanics. We will discuss the electronic structure that arises from solutions to the Schrödinger equation and why finding these solutions is problem that requires exponentially growing computational resources. We will then discuss how this manifests as outstanding challenges in the field of quantum materials science, especially in the areas of materials simulation and discovery. Finally, we will introduce the two branches of non-classical methods discussed in this dissertation that address these issues, and we will conclude by identifying the key research gaps that they address.

2.1.2 Review of Quantum Mechanics

The field of quantum mechanics arose in the early to mid 20th century to address growing gaps in the understanding of chemical systems at the atomic scale. Before the development of quantum theory, there existed two critical gaps in the understanding of the nature of objects like subatomic particles. The first problem was that of addressing the statistical versus deterministic interpretation of the nature

of these objects in a way that was consistent with experimental observations, while the second problem was in unifying the "particle vs. wave" interpretation of physical objects. To describe these phenomena, quantum mechanics introduces a complex-valued wave function $\psi(x)$, whose square amplitude $|\psi(x)|^2$ yields the likelihood of observing a particle at a position x in space. The wave function uniquely describes a quantum mechanical object up to a global complex phase (e.g. multiplication by $e^{i\theta}$ for $0 < \theta < 2\pi$), and it is subject to the L^2 normalization condition

$$\int |\psi(x)|^2 \, dx = 1 \tag{2.1}$$

which evokes the probabilistic interpretation of quantum objects. The behavior of quantum systems is governed by the Schrödinger equation, which models the non-relativistic behavior of wave-like systems as a first-order linear operator differential equation:

$$\frac{\partial}{\partial t}\psi = \frac{-i}{\hbar}\hat{H}\psi \tag{2.2}$$

Above, $i = \sqrt{-1}$ is the imaginary constant, $\hbar = h/(2\pi)$ is the reduced Planck constant, and $\hat{H}$ is the Hamiltonian operator. The Hamiltonian is a linear operator representing the sum of kinetic and potential energies of a wave function. For a single particle wave function with mass m under a static external potential $V(x)$, the Hamiltonian is

$$\hat{H} = -\frac{\hbar^2}{2m}\nabla^2 + V(x). \tag{2.3}$$

Integrating the dynamics generated by (2.2), the time-evolution of the wave function $\psi(x)$ takes the form

$$\psi(x,t) = \exp\left(\frac{-it}{\hbar}\hat{H}\right)\psi(x,0) \tag{2.4}$$

where $\psi(0)$ is the initial state at $t = 0$. Computing this operator exponential is difficult; however the exponential can be simplified in terms of the time-independent eigenfunctions $\psi_n(x)$ of the Hamiltonian operator $\hat{H}$:

$$\psi(x,t) = \exp\left(\frac{-it}{\hbar}\hat{H}\right)\psi(x,0) = \sum_n e^{itE_n/\hbar}\left(\int \psi_n^*(x)\psi(x,0)\ dx\right)\psi_n(x) \tag{2.5}$$

Each eigenfunction ψ_n has an associated eigenenergy E_n and arises as a solution to the eigenvalue problem known as the time-independent Schrödinger equation:

$$\hat{H}\psi_n(x) = E_n\psi_n(x) \tag{2.6}$$

Because the operator $\hat{H}$ is Hermitian (i.e., the complex operator analogue of a symmetric operator or matrix), $\hat{H}$ has real eigenvalues E_n, which correspond to a set of mutually orthogonal eigenfunctions ψ_n. The set of all pairs (ψ_n, E_n) that satisfy (2.6) provide important information about the physical system of interest.

The eigenvalues E_n are called "eigenenergies," because they represent the discrete energy levels that a single wave function can found to occupy when observed. The eigenfunctions $\psi_n(x)$ represent the associated modes of the system with energy E_n. They are uniquely determined up to a complex phase and the selection of an arbitrary orthogonal basis for the subspace spanned by other eigenfunctions ψ_n sharing the same eigenenergy E_n. (In this special case, we say the subspace of such eigenfunctions is degenerate).

The mutual orthogonality of the eigenfunctions ψ_n exists with respect to the L^2 inner product, namely:

$$\langle \psi_m \mid \psi_n \rangle = \int \psi_m^*(x)\psi_n(x)\ dx = \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases} \tag{2.7}$$

Above, the bracket notation $\langle \psi_m \mid \psi_n \rangle$ signifies the L^2 inner product. In the context of quantum mechanics, it is common to employ *Dirac notation*, which allows for symbolic manipulation of the left-hand and right-hand sides of the L^2 inner product, known as the "bra" and "ket" side respectively:

$$\begin{aligned} \langle \psi | &\equiv \left[\int\ dx\ \psi^*(x) \right] && \text{(“bra”)} \\ | \psi \rangle &\equiv \psi(x) && \text{(“ket”)} \end{aligned} \tag{2.8}$$

Due to the the bilinearity of the the L^2 inner product, the left-hand ("bra") and right-hand ("ket") wave functions can be treated as if they are linear operators. For example, taking the product of a "bra" and a "ket" results in the L^2 inner product bracket given in (2.7). Conversely taking the product of a "ket" and a "bra" operator results in an L^2 outer product operator. This notation facilitates the expansion of a wave function in a basis of complete orthonormal wave functions (e.g., the eigenstates $|\psi_n\rangle$), which is written symbolically as a linear combination of kets

$$|\psi\rangle = \sum_n c_n\, |\psi_n\rangle \tag{2.9}$$

where the expansion coefficients are $c_n = \langle \psi | \psi_n \rangle$. Owing to the completeness of the basis, they satisfy $\sum_n |c_n|^2 = 1$. The complete space of wave functions spanned by an orthonormal basis of wave functions $\{|\psi_n\rangle\}$ is known as a *Hilbert space*.

2.1.3 Quantum Mechanics of Many-Particle Systems

The quantum mechanical dynamics of single-particle systems under a static Hamiltonian $\hat{H}$ (as given in (2.3)) is relatively straightforward to solve using numerical techniques. However, the dynamics of multi-particle systems becomes much more complex when many interacting particles are considered. Indeed, it has been proven that solving for the eigenstates of general multi-particle systems and simulating their dynamics is a classically intractable problem. Recent studies in the field of quantum computing, for example, have shown that solving for the ground states of certain classes of Hamiltonians is at least as computationally hard as solving NP-complete problems, meaning the worst-case computational time required to solve for a many-particle ground state Ψ_0 scales exponentially with the number of particles in the system. In this section, we will give a brief review of the quantum mechanics of many-particle systems, and provide some necessary context for understanding why the problem of finding ground state solutions is computationally difficult.

In many-particle quantum mechanics, we extend the foundational concepts of single-particle quantum mechanics to model ensembles of interacting, but non-distinguishable particles. Due to this non-distinguishability, the representation of a many-particle wave function need only be concerned with how many particles occupy each state, and of which kind. For simplicity, we will primarily focus on systems which contain a single kind of particle and where interactions mediated by other particles are incorporated in an effective manner.

When transitioning from single particles to ensembles of many indistinguishable particles, we must expand the single-particle wave function Hilbert space into a many-particle *Fock space*, in which each single-particle wave function $|\psi_i\rangle$ can be

inhabited by any number of particles $n_i \geq 0$, called the *occupation number*. To denote such a many-particle wave function $|\Psi\rangle$ with a specific occupation configuration, we use the occupation number formalism

$$|\Psi\rangle = |n_1, n_2, n_3....\rangle \tag{2.10}$$

where any unoccupied states ($n_i = 0$) are omitted from the ket. Multi-particle ensembles are therefore represented as linear combinations over all possible state occupation numbers. The state with no particles occupied (denoted either $|0\rangle$ or $|\rangle$), is known as the *vacuum state*, and it is canonically defined to have energy $E = 0$ in a specified basis. To construct a many particle state, so-called creation and annihilation operators $\hat{a_i}^\dagger$ and $\hat{a_i}$ are introduced, which respectively increase and decrease the occupation numbers n_i:

$$\hat{a}_i^\dagger |0\rangle = |1_i\rangle\,, \qquad \hat{a}\,|1_i\rangle = |0\rangle \tag{2.11}$$

The annihilation operators, when acting on the vacuum state $|0\rangle$ produce an empty expression:

$$\hat{a}_i\,|0\rangle = 0 \tag{2.12}$$

The difficulty with modeling indistinguishable particles originates from the exchange statistics of the respective particles, which is a direct consequence of the particle's spin degrees of freedom. According to the standard model of particle physics, elementary particles can be partitioned into two categories: *fermions*, which have half-integer spin, and *bosons*, which have integer spin. Mathematically, the exchange statistics of fermions and bosons are often defined using the *canonical commutation*

relations (CCRs) of their creation and annihilation operators $\hat{a}$ and $\hat{a}^\dagger$. These relations make use of the operator commutator and anti-commutator, defined as follows:

$$\begin{aligned} [A, B] &= AB - BA \quad \text{(Commutator)} \\ \{A, B\} &= AB + BA \quad \text{(Anti-commutator)} \end{aligned} \tag{2.13}$$

We summarize the fermionic and bosonic exchange statistics and CCRs of the creation and annihilation operators ($\hat{a}_i$ and $\hat{a}_i$) in Table 2.1.

Table 2.1: A table of the two types of elementary particles, according to the Standard Model. In the CCRs column, δ_{ij} denotes the *Kronecker delta* ($\delta_{ij} = 1$ if $i = j$ and 0 otherwise).

Particle Type	Exchange ($m \neq n$)	CCRs	Spin Statistics	Examples
Fermions	$\hat{a}_i\hat{a}_j^\dagger \lvert\Psi\rangle = -(\hat{a}_j^\dagger\hat{a}_i \lvert\Psi\rangle)$	$\{\hat{a}_i, \hat{a}_j^\dagger\} = \delta_{ij}$ $\{\hat{a}_i, \hat{a}_j\} = 0$ $\{\hat{a}_i^\dagger, \hat{a}_j^\dagger\} = 0$	Spin 1/2 (Fermi-Dirac)	Electrons, Quarks, Neutrinos
Bosons	$\hat{a}_i\hat{a}_j^\dagger \lvert\Psi\rangle = \hat{a}_j^\dagger\hat{a}_i \lvert\Psi\rangle$	$[\hat{a}_i, \hat{a}_j^\dagger] = \delta_{ij}$ $[\hat{a}_i, \hat{a}_j] = 0$ $[\hat{a}_j^\dagger, \hat{a}_i^\dagger] = 0$	Spin 0 or 1 (Bose-Einstein)	Photons, Gluons, Higgs boson

In Table 2.1, we note that the CCRs for fermions and bosons are identical, except for the usage of the commutator and anticommutator, respectively. This key difference manifests in the exchange statistics of the particle, which illustrates how differences in the order of creation and annihilation of particles in separate states affect the overall sign of the wave function. In the case of fermions, the many-particle wave function Ψ accumulates a phase of -1, when creation/annihilation order is exchanged, while in the case of bosons, the wave function is left unchanged. For two simultaneous creation operations acting in the same mode, it can be deduced directly from the fermionic CCRs and properties of $\hat{a}, \hat{a}^\dagger$ that

$$\{\hat{a}_i, \hat{a}_i\} = \{\hat{a}_i^\dagger, \hat{a}_i^\dagger\} = 0 \qquad \Rightarrow \qquad \hat{a}_i^2 \lvert n_i\rangle = (\hat{a}_i^\dagger)^2 \lvert n_i\rangle = 0 \tag{2.14}$$

This result is known the *Pauli exclusion priniciple* and shows that no two electrons can occupy the same state, and therefore the occupation numbers are restricted to $n_i \in \{0, 1\}$. For bosons, no such restriction is observed on occupation numbers:

$$[\hat{a}_i, \hat{a}_i] = [\hat{a}_i^\dagger, \hat{a}_i^\dagger] = 0 \qquad \Rightarrow \qquad \hat{a}_i^2 \left|n_i\right\rangle = \begin{cases} \left|n_i - 2\right\rangle, & n_i \geq 2 \\ 0, & n_i < 2 \end{cases} \tag{2.15}$$

$$(\hat{a}_i^\dagger)^2 \left|n_i\right\rangle = \left|n_i + 2\right\rangle$$

Consequently, any number of bosons $n_i \geq 0$ may occupy a given mode.

The creation and annihilation operator algebras for fermion and bosons provide a systematic way of building a complete orthonormal Fock-space basis of occupation configurations $|n_1, n_2, n_3, ...\rangle$ indexed by the occupation numbers n_i of each mode. ψ_n. This basis is necessary for representing wave function ensembles consisting of combinations of multiple occupation configurations, as well as systems where the number of particles is not a fixed quantity. A generic many-particle wave function in Fock space is written as a superposition of multiple configurations

$$\left|\Psi\right\rangle = \sum_{\mathbf{n}=\{n_1,n_2,n_3,...\}} c_{\mathbf{n}} \left|\mathbf{n}\right\rangle, \tag{2.16}$$

where the complex-valued coefficients $c_{\mathbf{n}}$ satisfy the normalization condition $\sum_{\mathbf{n}} |c_{\mathbf{n}}|^2 = 1$.

It is apparent that the dimensionality of Fock space representations grow combinatorially with system size. For example, a system of N spin-1/2 fermions distributed over M orbitals spans a space of dimension at least

$$\dim(|\Psi\rangle) = \binom{M}{N}, \tag{2.17}$$

Moreover, if the number of fermions is in the ensemble is allowed to vary so that the wave function occupies the full Fock space, the dimensionality increases to

$$\dim(|\Psi\rangle) = \sum_{N=0}^{M} \binom{M}{N} = 2^M \tag{2.18}$$

2.1.4 Chemical Many-body Hamiltonians

In many-body quantum mechanics, the Hamiltonian H is expanded as an operator acting in the many-particle Fock space. This Hamiltonian must not only account for the interactions described in the single-particle Hamiltonian (2.3), but also incorporate the interactions between particles occupying different quantum states and configurations. These interactions are most naturally expressed in terms of creation and annihilation operators acting on the single-particle basis $|\psi_i\rangle$. The most general form of a Hamiltonian containing one-particle and two-particle interaction terms is

$$\hat{H} = \sum_{i,j} t_{ij} \hat{a}_i^\dagger \hat{a}_j + \frac{1}{2} \sum_{i,j,k,l} V_{ijkl} \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_l \hat{a}_k \tag{2.19}$$

where the coefficients t_{ij} are the single-particle Hamiltonian matrix elements

$$t_{ij} = \int \psi_i^*(x) \left(-\frac{\hbar^2}{2m} \nabla^2 + V(x) \right) \psi_j(x) \, dx \tag{2.20}$$

while the coefficients V_{ijkl} encode two-body interactions. For chemical systems consisting of interacting electrons (fermions), the two-body interaction is precisely the Coulomb repulsion:

$$V_{ijkl} = \iint \psi_i^*(x) \psi_j^*(x') \frac{e^2}{|x - x'|} \psi_k(x) \psi_l(x') \, dx \, dx' \tag{2.21}$$

This representation makes explicit that the complexity of a many-body problem is determined both by the number of accessible single-particle modes and by the structure of interparticle correlations.

In the context of quantum materials, the Hamiltonian above is rarely treated in its full microscopic form. Instead, reduced models are often introduced that retain the relevant degrees of freedom for a given physical regime. For example, tight-binding Hamiltonians approximate electronic interactions through an effective single-particle Hamiltonian with hopping terms t_{ij} between localized spin-degenerate orbitals:

$$\hat{H}_{TB} = \sum_{i,j} = t_{ij}\hat{a}_i^\dagger \hat{a}_j \tag{2.22}$$

The Hubbard model extends this by only incorporating the two-body interactions between the spin-up and spin-down electrons occupying the same spatial orbital:

$$\hat{H}_{Hub} = \sum_{ij} t_{ij}\hat{a}_i^\dagger \hat{a}_j + U \sum_i \hat{n}_{i\uparrow}\hat{n}_{i\downarrow} \tag{2.23}$$

Above, $\hat{n}_i = \hat{a}_i^\dagger \hat{a}_i$ is the orbital density operator.

The tight binding Hamiltonian (2.22) for a system with M orbitals has dimension $\dim(\hat{H}_{TB}) = M$, because it occupies a sparse non-interacting subspace of the full wave function Fock space. As a result, the many-body formulation of (2.6) (e.g., $\hat{H}\,|\Psi\rangle = E\,|\Psi\rangle$) can be solved efficiently to obtain the many-particle ground state Ψ_0 with energy E_0. This consists of diagonalizing the effective Hamiltonian matrix (with entries $\mathbf{H}_{ij} = t_{ij}$), which requires roughly $O(M^3)$ operations on a classical computer [17].

However, in the Hubbard Hamiltonian (2.23), the incorporation of the U terms results in a ground state wave function $|\Psi_0\rangle$ with correlated spin states, spanning the Fock space with dimension 2^M. These correlations that span outside the single-particle subspace are an example of the broader phenomenon of *quantum entanglement*. In these strongly entangled states, the number of independent coefficients

$c_{\mathbf{n}}$ required to specify an arbitrary wave function increases exponentially with the number of particles or sites.

This exponential scaling has dire computational consequences. Exact diagonalization of a Hamiltonian represented in this basis requires manipulating matrices whose dimension is 2^M, leading to time and memory costs that scale exponentially in system size. Even storing the full wave function becomes infeasible beyond modest values of N. The challenge is therefore not simply practical but structural: generic interacting states exhibit quantum entanglement, which prevents factorization into products of single-particle wave functions and therefore resists classical compression.

The computational hardness of many-particle quantum problems can be formalized within the framework of computational complexity theory. Determining ground-state energies of general local Hamiltonians belongs to complexity classes believed to be intractable for classical computation, and the worst-case scaling of exact algorithms is exponential in the number of degrees of freedom. Physically, this reflects the fact that interactions couple all configurations in Fock space, producing correlated states whose description requires resources comparable to the size of the space itself.

These observations motivate the central methodological developments of modern quantum materials research. Classical approaches rely on controlled approximations (e.g., mean-field theory, density functional theory, tensor-networks, and quantum Monte Carlo methods) that restrict the accessible portion of Fock space configurations. Quantum computing, by contrast, offers a representation in which quantum states are encoded natively (i.e., one fermion or boson per qubit), while

machine learning provides variational parameterizations capable of capturing high-dimensional correlations. The remainder of this dissertation seeks to capitalize on this intersection by introducing quantum and learning-based methods to navigate the exponentially large state spaces defined by many-body Hamiltonians and to enable predictive modeling of quantum materials.

2.2 Key Challenges in Quantum Materials Simulation

2.2.1 Material Simulation

The simulation of quantum materials is fundamentally a problem of solving interacting many-body Hamiltonians at scales where wave function correlation and structural complexity are essential. In principle, the microscopic behavior of a solid-state material is determined by the electronic Schrödinger equation under a periodic potential generated by the atomic lattice and modified by electron–electron interactions. In practice, however, direct solution of this problem is intractable for all but select quadratic systems due to the exponential growth of the many-body Hilbert space and the nonperturbative nature of electronic correlations [18].

A central challenge arises from the breakdown of single-particle approximations in regimes where two-body electronic interactions dominate kinetic energy and mean-field potential energy scales. In such systems (e.g., magnets [19], Mott insulators [20], heavy fermion compounds [21], and topological materials [22]) the low-energy physics cannot be described by independent non-interacting particles. Instead, collective effects such as localization, emergent magnetism, and topological order arise from correlated many-body states. Accurately capturing these phenomena requires

methods capable of representing entangled wave functions and capturing non-local interactions.

In addition to strong correlation, realistic materials also involve multiscale coupling between electronic, lattice, spin, and orbital degrees of freedom. Electron–phonon interactions, disorder, and external fields introduce additional dynamical modeling complexity, while nonequilibrium processes such as transport and optical excitation require time-dependent treatments [23]. These coupled effects significantly increase the dimensionality of the state space and complicate the construction of predictive models.

Existing computational frameworks address this complexity through controlled approximations. Density functional theory (DFT) provides tractable ground-state predictions but relies on approximate exchange-correlation functionals and often struggles in strongly correlated regimes [24]. Many-body perturbation methods such as the GW approximation [25] and dynamical mean-field theory (DMFT) [26] improve the treatment of correlations but at substantial computational cost and with limited scalability. Tensor-network approaches such as density matrix renormalization group (DMRG) methods [13] efficiently represent low-entanglement states in one dimension but encounter difficulties in higher-dimensional or volume-law regimes, while quantum Monte Carlo methods suffer from the fermionic sign problem in general settings [27, 28].

Beyond algorithmic limitations, a practical bottleneck emerges from the cost of generating high-fidelity simulation data. Accurate calculations are computationally intensive and therefore sparse across chemical and structural space, constraining

both benchmarking and model development. As a result, material simulation today operates under a persistent tradeoff between physical accuracy, computational tractability, and coverage of relevant design spaces. Addressing this tension is central to enabling predictive modeling of quantum materials.

2.2.2 Material Discovery

While simulation seeks to predict properties from a given Hamiltonian or structure, material discovery inverts this process: the objective is to identify systems whose microscopic structure produces targeted macroscopic behavior. This inverse problem is inherently high-dimensional and ill-posed, as multiple structural realizations can correspond to similar physical properties, and small perturbations in composition or symmetry can drastically alter emergent material properties.

The search space for quantum materials spans chemical composition, crystal structure, dimensionality, defects, strain, and interface effects. Each of these factors modifies the effective Hamiltonian governing the system, often in nonlinear and poorly understood ways. Consequently, discovery is not merely a combinatorial trial-and-error problem but a problem of navigating a structured, high-dimensional landscape defined jointly by atomic configurations and many-body electronic states [29, 30].

Experimental validation further constrains discovery workflows. Synthesis and characterization cycles are slow and resource-intensive, and measurements often provide incomplete information about microscopic structure. This effectively bottlenecks the availability of labeled data and introduces uncertainty and many uncontrolled variables into structure-property relationships. Computational screening via

high-throughput *ab initio* methods partially alleviates this issue, but it inherits the limitations of the simulation methods on which it depends.

A further challenge lies in *generalizability*. Models and heuristics developed for known classes of materials frequently fail when extrapolated to new chemical regimes or phases, particularly in strongly correlated or topological systems. The absence of generalizable representations and well-curated datasets that link atomic structure and many-body observables, while avoiding dataset sampling bias remains a major barrier to systematic discovery [31].

These limitations are the motivating impetus for the development of predictive frameworks capable of exploring materials spaces more efficiently. Such approaches aim to learn structure–property relationships, construct surrogate models of underlying physics (e.g., constructing Hamiltonians), and guide exploration toward regions likely to host desired chemical properties. Achieving this requires careful integration of physically-informed modeling (e.g. via physics-informed machine learning [32]) with interpretable and verifiable inference methods to develop models that generalize well across both materials space and Hilbert space. The emergence of machine learning and quantum computing as tools with distinct individual strengths reflects this need, offering new, distinctly *non-classical*, methods for material discovery.

CHAPTER THREE

Machine Learning Methods

3.1 Introduction

The discovery and engineering of quantum materials, systems whose macroscopic properties are governed by quantum mechanical phenomena, remain central challenges in condensed matter physics and materials science. Traditional materials modeling approaches, ranging from empirical interatomic potentials to first-principles electronic structure calculations, have enabled profound advances; however, they are limited by either accuracy or computational scalability. In parallel, the growth of curated materials databases and high-performance computing infrastructure has created unprecedented opportunities for data-driven modeling. These advances have culminated in the birth of the field of *materials informatics*, which seeks to apply techniques in machine learning, data science, computational chemistry, and solid state physics and bridge the gap between the foundational theory of materials, and material experiments carried out in the laboratory.

In particular, machine learning (ML) has emerged as a powerful tool for accelerating materials modeling. In electronic structure modeling, ML is typically employed for the task of constructing surrogate models for quantum mechanical observables [33–35] or identifying hidden structure in high-dimensional materials spaces [36–38]. In atomistic modeling, ML techniques are particularly powerful because they provide flexible, nonparametric approximations to complex many-body functions such as the potential energy surface, the electronic Hamiltonian, or derived

quantities including band structures and topology [39,40]. When properly constrained by physical symmetries and quantum mechanical structure, these models can achieve both predictive accuracy and transferability.

In this chapter, we give a brief introduction to machine learning and review the role of machine learning in materials research, with an emphasis on atomistic modeling of quantum materials. We focus on four themes: (i) high-throughput material discovery, (ii) integration with *ab initio* electronic structure methods, (iii) invariant and equivariant learning architectures, and (iv) quantum-informed modeling strategies that embed physical structure directly into ML frameworks. We then introduce the key contributions made by this dissertation research in the context of machine learning-based non-classical methods.

3.1.1 Fundamentals of Machine Learning

Machine learning (ML) is a framework for constructing parametric or non-parametric models that approximate unknown functions from data. Formally, one seeks to approximate a target mapping

$$f^\star : \mathcal{X} \to \mathcal{Y}, \tag{3.1}$$

given a finite dataset

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}, \tag{3.2}$$

where $x_i \in \mathcal{X}$ are inputs and $y_i \in \mathcal{Y}$ are corresponding outputs. In materials science, $\mathcal{X}$ may represent atomic configurations, compositions, or crystal structures, while $\mathcal{Y}$ may denote scalar properties (e.g., formation energies), vector quantities (e.g., forces),

or more structured objects (e.g., Hamiltonian matrices). At its core, supervised learning constructs a parameterized model $f_\theta(x)$ with parameters θ, and determines θ by minimizing a loss functional,

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(x_i), y_i)\,, \tag{3.3}$$

where ℓ is typically a norm-based discrepancy measure such as the mean squared error:

$$\ell_{MSE}(\hat{y}_i, y_i) = \|\hat{y}_i - y_i\|^2 \tag{3.4}$$

Optimization of a desired loss functional $\mathcal{L}$ is performed using gradient-based methods, most commonly stochastic gradient descent (SGD) and its variants [41]. When f_θ is differentiable with respect to θ, and the loss function ℓ is also differentiable, analytical (or numerically approximated gradients $\nabla_\theta \mathcal{L}$ can be efficiently computed via differentiation. The gradient $\nabla_\theta \mathcal{L}$ indicates the "direction" in which an infinitesimal change in the parameters θ will yield the greatest increase in $\mathcal{L}$:

$$\begin{aligned} \nabla_\theta \mathcal{L}(\theta) &= \frac{1}{N}\nabla_w \left\{\ell(f_\theta(x_i), y_i)\right\} \\ &= \frac{1}{N}\sum_{n=1}^{N} \frac{\partial \ell}{\partial \hat{y}}(f_\theta(x_n), y_n) \cdot \nabla_\theta f_\theta(x_i) \end{aligned} \tag{3.5}$$

The intuition behind gradient descent methods is that, provided the functional loss landscape $\mathcal{L}(\theta)$ is sufficiently smooth within the vicinity of the current configuration θ, iterative adjustments can be made to θ, in the direction of the greatest decrease of $\mathcal{L}$. Small iterative adjustments to θ can be made, such that the loss functional decreases to a local minimum. The magnitude of these adjustments (the *learning rate* η, chosen $0 < \eta \ll 1$) is a scalar value selected to be large enough to

ensure convergence, but not so large as to overstep local minima:

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\eta}{\|\nabla_\theta \mathcal{L}(\theta^{(t)})\|} \nabla_\theta \mathcal{L}(\theta^{(t)}) \tag{3.6}$$

This iterative scheme (or other, more optimized gradient-based methods such as *Adaptive Moment Estimation*) are applied until $\mathcal{L}$ converges to a local minimum.

3.1.1.1 Statistical perspectives. From a probabilistic standpoint, supervised learning can be interpreted as estimating the conditional expectation

$$f^\star(x) = \mathbb{E}\{Y \mid X = x\}, \tag{3.7}$$

under an unknown joint distribution $p(x, y)$. Because only finite samples are available, learning involves balancing two competing objectives: minimizing training error and ensuring generalization to unseen data. This trade-off is formalized through the concept of the *bias–variance decomposition*: highly flexible models may achieve low training error but overfit noise (resulting in high variance), whereas overly constrained models may underfit systematic structure (resulting in high bias). *Regularization techniques* can be applied to mitigate overfitting by constraining model complexity through on of a number of ways. Examples include imposing ℓ_2 (sum-of-squares) penalties on model parameters via weight decay [42], using early stopping during optimization [43], applying dropout [44], and using informative Bayesian prior distributions in probabilistic models [45].

3.1.1.2 Model classes. A variety of basic model classes are used in scientific machine learning. Here, we give a brief structured review of basic models used in machine learning, emphasizing their applications in the materials informatics literature:

- *Linear Models*: Linear regression assumes

$$f_\theta(x) = \mathbf{w}^T \phi(x) + w_0, \tag{3.8}$$

where $\phi(x) : \mathcal{X} \to \mathbb{R}^m$ is a vector feature map and $\mathbf{w} \in \mathbb{R}^m$ is a vector of linear weights with an optional bias factor $w_0 \in \mathbb{R}$. Despite their simplicity, linear models can be quite powerful when paired with expressive mappings ϕ that capture enough features that are linearly correlated with respect to the quantity to be estimated. In the materials informatics literature, there are extensive discussions regarding which feature maps ϕ (often called *material fingerprints*) are most effective at capturing material properties and can be computed directly from molecular or crystal structure. Although fingerprinting methods such as the string-based SMILES representations [46] are popular for predicting the properties of organic molecules, these methods often fail when applied to solid-state systems, as systems with similar bond structure and stoichiometry may exhibit multiple different crystal structures. Thus, when modeling solid-state materials, all aspects of the crystal geometry must be directly taken into account.

- *Kernel Methods*: Kernel methods such as support-vector machines (SVMs) and nearest-neighbor interpolation models [47], or Gaussian process regression (GPR) [48] models represent the predictor as a weighted sum over training examples:

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x, x_i), \tag{3.9}$$

where $k(\cdot, \cdot)$ is a positive-definite kernel (an inner product that encodes the similarity between x and x'). Kernel methods allow for learning basic nonlinear trends in data while providing structured interpolation between known data points. Gaussian process kernels are often used in material property prediction models, since they provide principled uncertainty quantification of predictions, which is particularly valuable in active learning and materials discovery workflows [49].

- *Neural Networks*: Neural networks (NNs) approximate functions via compositions of affine transformations and nonlinear activation functions:

$$f_\theta(x) = W_L \sigma\big(W_{L-1}\sigma(\cdots \sigma(W_1 x))\big), \tag{3.10}$$

where σ denotes a nonlinear activation (e.g., ReLU, GELU, SiLU, etc.) [50]. Deep networks (large L) can approximate highly complex functions and scale effectively to large datasets. In atomistic modeling, neural networks are often structured as graph-based architectures to reflect relational structure in atomic systems.

3.1.2 Scientific Machine Learning

In scientific applications, performance depends as much on representation as on model class. The feature map $\phi(x)$ encodes prior assumptions about invariances, locality, smoothness, or conservation laws. These assumptions constitute the model's *inductive bias*—the structural constraints that guide learning in the finite-data regime. For physically structured data such as atomistic systems, inductive biases commonly enforce translation, rotation, and permutation invariance. For more

complex mathematical objects (such as in electronic structure modeling), additional mathematical constraints such as Hermiticity of operators, gauge consistency, and symmetry under space group operations may be required. Embedding such symmetry structure directly into the neural network architecture often reduces computational complexity and improves extrapolation.

From the perspective of computational physics, machine learning is often viewed as a universal function approximation framework. The universal approximation theorem guarantees that sufficiently wide or deep neural networks can approximate continuous functions on compact domains [51]. However, in scientific domains, the objective is not merely approximation but physically consistent, computationally superior, and data-efficient approximation. Consequently, modern scientific machine learning emphasizes concepts such as symmetry preservation, the incorporation of physical constraints, governing mathematical equations, to ensure robust generalization beyond the training manifold [32].

These principles are particularly critical in quantum materials modeling, where the target functions (e.g., Hamiltonians, Green's functions, or band dispersions) possess rich mathematical structure. The remainder of this chapter explores how machine learning architectures are adapted to respect this structure in atomistic and electronic-scale modeling.

3.2 Machine Learning in Materials Research

Machine learning in materials science spans multiple scales, from coarse-grained structure-property correlations to atomistic and electronic-structure-level modeling. Although machine learning has been applied to the task of high-level material property

prediction, many of these modes are limited in their predictive capabilities, because they do not directly incorporate atomic and electronic structure [52, 53]. Therefore, recent research in materials research has focused on how atomic structure itself can be used as model inputs, without relying on embedding or fingerprinting schemes to extract descriptors of atomic or structural features.

At the atomistic level, the central task is the approximation of functions of the form

$$f : \{\mathbf{R}_i, Z_i\}_{i=1}^{N} \to \mathcal{O}, \tag{3.11}$$

where $\mathbf{R}_i$ and Z_i denote atomic positions and species, and $\mathcal{O}$ may represent total energies, forces, Hamiltonian matrix elements, band dispersions, density of states, or other observables. Because the underlying physical laws are invariant under translation, rotation, and permutation of identical atoms, any successful ML model must respect these symmetries either explicitly or implicitly.

In quantum materials, additional structure arises from lattice periodicity, and strongly quantum effects, such as spin–orbit coupling, gauge freedom, and the presence of topological invariants. Consequently, models must often operate not only on atomic coordinates but also on reciprocal-space representations, orbital bases, or graph-structured Hamiltonians, all of which must be appropriately structured for the learning task. The following sections review the principal methodological developments in this area.

3.2.1 High-throughput Material Discovery

High-throughput materials discovery refers to the systematic computational screening of large chemical and structural spaces to identify candidate materials with

desirable properties. Initiatives such as the Materials Project [54], AFLOW [55], and the Open Quantum Materials Database (OQMD) [56] have generated extensive repositories of DFT calculations, including relaxed geometries, formation energies, band structures, and elastic tensors.

Machine learning accelerates this process in several ways. First, surrogate models trained on DFT data can rapidly predict formation energies or band gaps for previously uncomputed structures, enabling pre-screening prior to expensive first-principles evaluation. Kernel methods (e.g., Gaussian process regression), ensemble techniques (e.g., random forests), and neural networks have all been employed for this purpose [57]. Descriptor-based approaches, such as pairwise Coulomb interaction matrices, smooth overlap of atomic positions (SOAP) [49], and many-body tensor representations such as atomic cluster expansions (ACE) [58], map atomic configurations to fixed-dimensional feature spaces, after which standard regression models are applied.

Second, active learning frameworks have been applied to iteratively couple ML models with electronic structure calculations. A model is trained on an initial dataset, uncertainty is quantified (via Bayesian methods or ensemble variance), and new data points are selected to maximally reduce model uncertainty. This closed-loop approach dramatically reduces the number of required first-principles evaluations.

For quantum materials, high-throughput discovery often targets topological phases, correlated electron systems, and materials with nontrivial band topology. Here, ML models may be trained to predict derived quantities such as $\mathbb{Z}_2$ invariants or symmetry indicators from structural and compositional features. Because such

properties depend sensitively on symmetry and spin–orbit interactions, careful encoding of crystallographic information is essential. Models such as DeepH [59, 60] have been recently developed to successfully integrate the symmetries of operators and other observables with spin-orbit coupling.

3.2.2 Ab Initio Modeling

Ab initio methods, most notably density functional theory (DFT), form the computational backbone of modern materials science. However, DFT scales approximately as $\mathcal{O}(N^3)$ with system size and becomes prohibitively expensive for large supercells, molecular dynamics trajectories, or strongly correlated systems requiring beyond-DFT treatments.

Recent work in computational chemistry and materials science has applied machine learning methods to augment *ab initio* modeling in several distinct ways, the most popular of which we briefly summarize below:

- *Interatomic Potentials:*

  Neural network potentials (NNPs) [39,61], Gaussian approximation potentials (GAP) [62], and related approaches learn the mapping

  $$\{\mathbf{R}_i, Z_i\}_{i=1}^{N} \mapsto E_{\text{tot}}, \tag{3.12}$$

  with forces obtained via automatic differentiation of the total energy with respect to atomic coordinates:

  $$\mathbf{F}_i = \nabla_{\mathbf{R}_i} \left[E_{tot}(\{\mathbf{R_i}, Z_i\}_{i=1}^{N}) \right] \tag{3.13}$$

When trained on DFT data, these models enable molecular dynamics simulations with near-*ab initio* accuracy at orders-of-magnitude lower computational cost. For quantum materials, this facilitates exploration of structural phase transitions, lattice instabilities, and finite-temperature effects.

- *Electronic Structure Surrogates:*

  Beyond total energies, ML models increasingly target electronic observables directly. Examples include direct prediction of band structures along high-symmetry paths, density of states, or even Hamiltonian matrix elements in localized orbital bases. Such models can be formulated either as regression problems over discretized k-space grids or as operator-learning tasks in which the Hamiltonian is represented as a graph-structured object.

- *Embedding and Hybrid Methods:*

  Machine learning is also used to accelerate components of electronic structure calculations, such as exchange–correlation functionals, density-to-potential mappings, or Green's function self-energies [63]. In correlated systems, ML can approximate expensive many-body solvers within dynamical mean-field theory (DMFT) or similar frameworks [64].

In each case, the principal challenge is ensuring that the learned representation preserves the mathematical structure of quantum mechanics: Hermiticity of operators, gauge covariance, and physical constraints such as particle conservation.

3.2.3 *Invariant and Equivariant Models*

A defining feature of atomistic systems is symmetry. The energy of an isolated system is invariant under global translations, rotations, and permutations of identical atoms:

$$E(\{\mathbf{R}_i\}) = E(\{\mathcal{R}\mathbf{R}_i + \mathbf{t}\}), \tag{3.14}$$

for any rotation $\mathcal{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$. Forces transform equivariantly under rotations:

$$\mathbf{F}_i \mapsto \mathcal{R}\mathbf{F}_i. \tag{3.15}$$

Early ML models enforced invariance by constructing handcrafted descriptors invariant under these operations. Modern approaches instead embed symmetry directly into neural architectures. Graph neural networks (GNNs) treat atoms as nodes and interatomic interactions as edges, naturally incorporating permutation invariance [65]. Message-passing neural networks (MPNNs) propagate information locally and can approximate many-body interactions [66].

More recently, *equivariant neural networks*, which are built on representations of the rotation/reflection group $O(3)$ or the Euclidean group $E(3)$, have enabled models in which intermediate features transform according to irreducible representations of the symmetries of 3D space. Such architectures (e.g., tensor field networks [67], $E(3)/O(3)$-equivariant graph networks [68]) preserve rotational covariance and *chirality* (i.e. the "handedness" of graphs embedded in 3D space) at every layer. This is particularly important when predicting vectorial or tensorial quantities such as forces, dipole moments, or stress tensors. In the context of learning the properties of

solid state materials or molecules, chirality plays a crucial role in distinguishing between certain molecules and atomic coordination environments [69]. This is especially critical for problems such as distinguishing between right-handed and left-handed molecules such as amino acids, which exhibit radically different interactions with the chiral amino acids present in the human body [70]. The importance of incorporating chirality ($p = \pm 1$) as a way of distinguishing molecular structures is illustrated in Figure 3.1.

In quantum materials, equivariance plays an even deeper role. Electronic wavefunctions transform under symmetry operations according to irreducible representations of the crystal's space group. Modeling Hamiltonian matrix elements or band structures therefore requires symmetry-aware architectures capable of respecting both real-space and reciprocal-space symmetries [71]. Failure to enforce these constraints can lead to unphysical symmetry breaking and poor generalization.

3.2.4 Quantum-Informed Models

While generic neural networks can approximate arbitrary continuous functions, unconstrained flexibility often leads to models that violate physical structure. Quantum-informed models instead embed prior knowledge from quantum mechanics directly into the architecture, feature representation, or loss functional. Several strategies are prominent:

- *Physics-Constrained Outputs*:

  Models predicting Hamiltonians or density matrices can enforce Hermiticity, positive semidefiniteness, or particle-number constraints by parameterizing

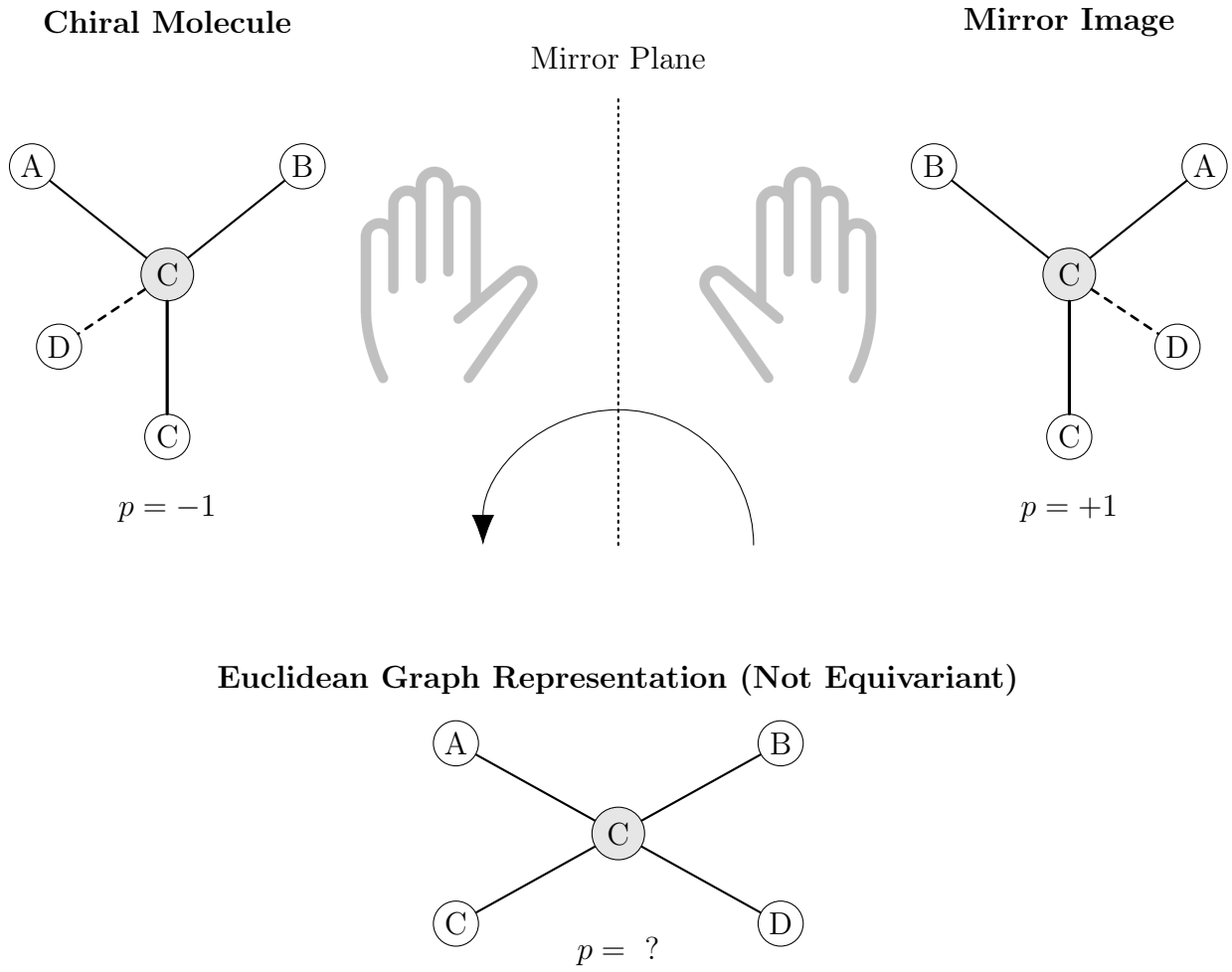

Figure 3.1: Traditional GNNs operating only on adjacency matrices or Euclidean distances and bond angles and cannot distinguish between certain chiral molecules without explicit stereochemical encoding. Conversely, equivariant models can distinguish between the chirality of molecules or atomic coordination environments.

outputs in structured forms (e.g., Hermiticity via $H = A + A^\dagger$). Similarly, spectral quantities can be constrained to satisfy sum rules.

- *Operator Learning*:

  Rather than predicting scalar observables, operator-learning approaches approximate maps between function spaces, such as

  $$\mathcal{F} : V_{\text{atomic}} \to \mathcal{H}_{\text{electronic}}, \tag{3.16}$$

  where $\mathcal{H}_{\text{electronic}}$ denotes a Hamiltonian operator in a chosen basis. This perspective aligns naturally with tight-binding or localized orbital representations and is particularly well suited for quantum materials in which low-energy physics is dominated by specific orbital manifolds.

- *Reciprocal-Space Learning*: Because features such as band topology and associated observables (such as Berry curvature) are defined in k-space, models that operate directly on reciprocal-space meshes or learn continuous band representations over the Brillouin zone provide a natural framework for topological materials modeling [72]. Incorporating periodic boundary conditions and gauge consistency across k-space is essential for stable training.

In the context of quantum materials, quantum-informed ML offers a principled path toward models that are both expressive and physically grounded. By integrating symmetry, operator structure, and domain knowledge, such approaches bridge the gap between data-driven learning and first-principles theory.

3.3 Overview of Research Contributions

Motivated by recent applications of machine learning in materials science and informatics, in the following two chapters we present two research projects that apply the theory and methodology introduced in this chapter. Below, we summarize the key insights and contributions made to the scientific literature through peer-reviewed publications:

3.3.1 High-Throughput Screening of Superconductors

Superconducting devices are crucial to emerging quantum technologies, contributing to innovations in the areas of computing, metrology, and communication systems. However, many of these devices are based on conventional BCS (Bardeen-Cooper-Schrieffer) superconductors, which have operating critical temperatures T_c that require expensive helium-based cryogenic cooling. This has motivated the exploration of unconventional non-BCS superconductors as an alternative platform for these technologies, since some of these superconductors exhibit desirable properties such as high T_c and topologically robust states. Although unconventional superconductors have been known to exist for several decades, a comprehensive theory of superconductivity in these materials has not yet been developed, making the discovery of new superconductors a challenging endeavor. In this paper, we present ongoing work in the autonomous discovery of conventional, unconventional, and exotic superconductors with deep learning methods. Using experimental data from over 36,000 superconducting materials, we demonstrate that graph neural network models trained on raw crystalline structure can be used to discover superconductors with desirable properties such as high T_c.

3.3.2 Quantum-Informed Neural Networks for Band Structure Prediction

Accurate electronic structure prediction underpins modern materials science and quantum chemistry but remains computationally intensive due to the iterative self-consistent field methods required by density functional theory (DFT). We introduce Quantum-Informed Neural Networks (QuINNs), a class of symmetry-preserving, physics-structured machine learning models designed to serve as differentiable surrogate quantum simulators. Unlike conventional graph neural networks that directly regress scalar observables, QuINNs learn to construct Gaussian orbital bases and pseudopotential parameterizations from atomic geometry, from which a physically consistent, Hermitian Hamiltonian is assembled and diagonalized. All quantum observables—including band structures, densities of states, and electronic orbitals—are derived from this learned Hamiltonian, preserving interpretability and enabling reconstruction of full electronic states.

QuINNs embed quantum mechanical constraints at the architectural level. Orbitals transform as irreducible representations of the Euclidean group $E(3)$, pseudopotentials are parameterized in rotationally equivariant Gaussian bases, and the Hamiltonian is constructed to be sparse, periodic, and block-diagonal in crystal momentum. By incorporating $O(3)$-equivariant graph neural networks and closed-form Gaussian integral evaluations, QuINNs retain the algebraic structure of ab initio simulation pipelines while eliminating iterative SCF loops at inference. The resulting models enable single forward-pass computation of electronic structure quantities and support end-to-end differentiability with respect to atomic geometry, making them

well-suited for inverse materials design, band structure engineering, and even quantum device optimization. This framework represents a conceptual shift from black-box property regression to structured Hamiltonian learning, offering a scalable and physically interpretable alternative to traditional electronic structure methods.

3.3.3 Summary and Outlook

Machine learning has transitioned from a peripheral tool to a central methodological pillar in materials science and materials informatics. In atomistic modeling of quantum materials, its impact is most pronounced when combined with physical structure: symmetry-aware architectures, operator-based representations, and hybridization with *ab initio* methods.

Future directions of research in this area include scalable operator-learning frameworks for many-body Hamiltonians, uncertainty quantification for deep molecular dynamics and equivariant architectures, as well as future integration with quantum computing platforms, leveraging the future scalability of hybrid quantum classical methods, which we discuss in the second half of this dissertation. As datasets grow and algorithms mature, the convergence of machine learning and quantum materials modeling is poised to redefine computational materials design and becomes a standard computational research tool with the future development of materials-oriented machine learning frameworks.

CHAPTER FOUR

High-Throughput Screening of Superconducting Materials

This chapter published as: C. Burdine and E. P. Blair, "Discovery of Novel Superconducting Materials with Deep Learning," *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Bellevue, WA, USA, 2023, pp. 1335-1341, doi: 10.1109/QCE57702.2023.00151.

Superconducting devices are crucial to emerging quantum technologies, contributing to innovations in the areas of computing, metrology, and communication systems. However, many of these devices are based on conventional BCS (Bardeen-Cooper-Schrieffer) superconductors, which have operating critical temperatures (T_c) that require expensive helium-based cryogenic cooling. This has motivated the exploration of unconventional non-BCS superconductors as an alternative platform for these technologies, since some of these superconductors exhibit desirable properties such as high T_c and topologically robust states. Although unconventional superconductors have been known to exist for several decades, a comprehensive theory of superconductivity in these materials has not yet been developed, making the discovery of new superconductors a challenging endeavor. In this section, we consider the problem of the autonomous discovery of conventional, unconventional, and exotic superconductors with deep learning methods. Using experimental data from over 36,000 superconducting materials, we demonstrate that graph neural network models trained on raw crystalline structure can be used to discover superconductors with desirable properties such as high T_c.

4.1 Introduction

Since their discovery over a century ago, superconductors have remained an area of intense research focus in materials science. Superconducting materials are known for exhibiting zero electrical resistance and ideal diagmagnetic behavior when cooled beneath their critical temperature T_c. Due to these useful properties, superconductors have seen many different industrial applications, ranging from magnetic resonance imaging (MRI) [73] to emerging nuclear fusion technology [74]. Superconductor research has also seen a recent boom in the area of quantum information, as superconductor-based technologies have shown promise in realizing scalable quantum information processors, sensors, and communication systems [75–77]. Many of these devices use conventional Bardeen-Cooper-Schrieffer (BCS) superconductors [78]. In BCS superconductors, the pairing of electrons in the superconducting state is known to be phonon-mediated; however, for unconventional (i.e. non-BCS) superconductors, the electron pairing mechanisms are not yet fully understood [79]. Despite this lack of of a supporting theory, unconventional superconductors are currently being explored as an alternative to conventional ones because of their desirable properties, such as high T_c and robust topological superconducting states [80]. For this reason, the discovery of new superconductors, especially unconventional ones, remains a challenging but potentially rewarding task. Since a vast amount of experimental data has been collected on known superconducting materials [81–83], data-driven deep learning approaches may be particularly suited to this task, allowing for high-throughput screening of material databases to identify new potential superconductors.

4.1.1 Related Work and Main Contribution

There have been a few previous attempts to apply various machine learning models to the discovery of superconductors, but a majority of these models used chemical-formula based representations of materials [82, 84, 85]. This is problematic, as the atomic structure of superconducting materials is known to play a crucial role in identifying them. Substantial work has also been done in identifying new BCS superconductors, with *ab-initio* methods like Eliashberg theory and density functional theory (DFT) being applied with great success in identifying conventional superconductors [86]. These methods incorporate both the atomic and electronic structure of a material, but are known to be very computationally demanding, making them unsuitable for screening large material databases. A promising alternative to these *ab initio* techniques involves applying graph neural networks, a type of deep learning model that directly predicts superconductor properties from a graph embedding of a material's crystal lattice. This embedding represents atoms as graph nodes and bonds between neighboring atoms as graph edges. Graph neural network architectures, such as the atomistic line graph neural network (ALIGNN) [34], have been successfully used to predict the T_c and Eliashberg spectrum of BCS superconductors from *ab initio* data [87]. However, these models have yet to be applied to the more general problem of identifying unconventional superconductors using experimental data. In this study, we address this issue and demonstrate that graph neural networks, such as the ALIGNN model, can predict superconducting T_c and identify both unconventional and conventional superconductors from experimental data. We also apply this model to the Materials Project database and identify over 600 potential superconducting materials.

4.2 Methods

4.2.1 Data Sources

The experimental data used in this study combined two large datasets of known superconductors hosted on the Materials Data Repository [81]. The first dataset (referred to here as *Supercon v1*) contained the chemical formulas and estimated T_C of over 14,000 known atmospheric pressure superconductors under many different doping schemes. This dataset was originally used in a T_c prediction study by Stanev et al. [82]. The second dataset (*Supercon v2*) was generated in a study by Foppiano et al. [83], which used a natural language machine learning model to scrape chemical formulas, T_c, and other important parameters (such as applied pressure, substrate, etc.) for over 20,000 superconductors from published academic papers. Although this dataset had better representation of more recently studied heavy fermion, magnetic, and hydride superconductors, many of the parsed formulas and T_c values were incorrect for various reasons, such as formula incompleteness or a magnetic material's Curie temperature (sometimes abbreviated T_c) being erroneously reported. These entries had to be corrected by hand or discarded entirely if an obvious correction could not be identified.

Altogether, our combined *Supercon v1* and *v2* dataset consisted of roughly 36,000 materials, which we classified into the 12 non-exclusive families shown in Fig. 4.1a. The most well-represented families in the dataset are the oxide and cuprate families, reflecting the significant research efforts that have been made toward engineering high-T_c ceramic superconductors. The alloy-based superconductors are also well-represented in our dataset, with members ranging from conventional alloys, such as Nb-Ti, to unconventional heavy fermionic paramagnets like UTe_2 [88]. In Fig. 4.1b,

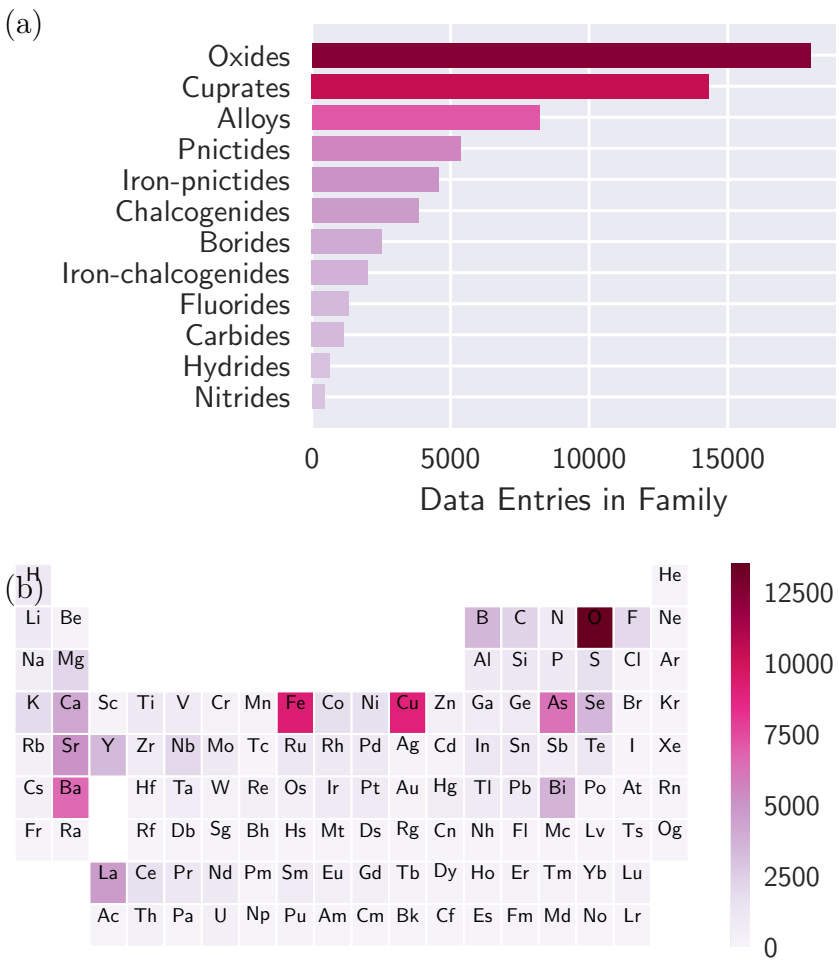

Figure 4.1: Demographics of the combined Supercon v1 and v2 datasets. (a) The number of superconductor data entries in our dataset for each non-exclusive family. (b) Number of data entries containing each element of the periodic table. A majority of entries in the dataset were oxide, cuprate and iron-based superconductors, which reflects the significant research interest in these materials.

we visualize the number of superconductors in the dataset containing each element in the periodic table.

4.2.2 Obtaining Crystal Structure

Superconductivity is known to be an atomic structure-dependant property, which means that structure must be incorporated into the prediction process in order make reliable predictions of whether or not a material exhibits the phenomenon.

Since the combined *Supercon v1* and *v2* datasets only listed the chemical formulas of each measured material, the structure of each superconductor had to be inferred based on chemical formula and any additional parameters (such as pressure) in each data entry. Structures for each superconductor were obtained by querying the Materials Project Database [54], and selecting the most similar thermodynamically stable material structure. We considered a material structure to be thermodynamically stable if its energy above hull did not exceed 0.05 eV/atom. For superconductors with applied pressure greater than 1 GPa, this stability requirement was not enforced and the closest material was instead selected manually. The similarity between materials was measured using the material composition distance metric:

$$d(M_1, M_2) = \sum_i \left| \frac{n_{E_i}(M_1)}{\sum_j n_{E_j}(M_1)} - \frac{n_{E_i}(M_2)}{\sum_k n_{E_k}(M_2)} \right|, \tag{4.1}$$

where $n_{E_i}(M)$ denotes the number of atoms of element E_i that appear in the chemical formula of M. For each superconductor in the dataset, we attempted to find an exact match in the Materials Project database with composition distance $d = 0$. If many matches were found, the match with the lowest formation energy was selected. Exceptions to this rule were made for some well-known materials, such as the carbon allotrope C_{60}, where the structure is nonambiguous.

4.2.3 Defect Materials

Many of the materials from the *Supercon v1* and *v2* datasets had formulas indicating the presence of defects in the form of substitutions, interstitials, or vacancies in the crystal lattice. If an exact thermodynamically stable match was found for a defect material's pristine structure, the positions of the lattice defects were inferred

through combinatorial search and optimization methods. In the literature, there are several different methods and models to predict the placement of defect sites in materials [89, 90]. For metals (which comprised a majority of our dataset), one of the simplest models is the embedded atom method [91], where defects are placed such that the total cohesive energy E_{coh} of the metal is minimized. Under a semi-classical treatment of defect-lattice interactions, E_{coh} can be approximated as:

$$E_{coh} = \sum_i G_i \left(\sum_{j \neq i} \rho_j^a(r_{ij}) \right) + \frac{1}{2} \sum_{i,j \neq i} U_{ij}(r_{ij}), \tag{4.2}$$

where $G_i(\rho)$ is the embedding energy function of atom i, $\rho_j^a(r_{ij})$ is the spherically averaged electron density, and U_{ij} is the electrostatic two-atom interaction at radius r_{ij}. If the lattice atoms are fixed and we assume an approximately linear embedding energy function (i.e. $G(\rho) = g\rho$), then the terms of E_{coh} can be re-written as a sum of lattice energy, and defect interaction potentials. Specifically, we write:

$$E_{coh} = E_{\text{lattice}} + \sum_{k,l} V_{k,l}(r_{kl}) + \sum_{k,k' \neq k} V_{k,k'}(r_{kk'}) \tag{4.3}$$

where k is the defect site index, l is the lattice site index, and $V_{k,l}(r)$ and $V_{k,k'}$ are the defect-lattice and defect-defect interaction potentials:

$$V_{k,l}(r) = g_l \rho_l^a(r) + g_k \rho_k^a(r) + U_{kl}(r) \tag{4.4}$$

$$V_{k,k'}(r) = 2 g_k \rho_k^a(r) + U_{kl}(r) \tag{4.5}$$

Finding the placement of defects that minimizes (4.3) requires a knowledge of the embedding energy coefficients g_i and radial density functions ρ^a for each defect and lattice atomic species. To obtain a rough initial placement of defects, we approximated $V_{k,l}$ and $V_{k,k'}$ with Gaussian pseudopotentials of the form

$$V_{i,j}(r) \approx \bar{g} \sqrt{\frac{2}{\pi}} \frac{(n_i + n_j)}{(R_i + R_j)} \exp\left(\frac{-2r^2}{(R_i + R_j)^2} \right) \tag{4.6}$$

where R_i, n_i are the atomic radii and number of electrons at site i, and $\bar{g} > 0$ is the mean effective embedding energy coefficient for the entire material. We used the Gaussian pseudopotential due its boundedness, monotonicity, and differentiability, which facilitates the optimization of E_{coh}. Since $\bar{g}$ is independent of i and j, we remark that $\bar{g}$ does not need to be known to find the positions of defects that minimizes E_{coh}.

To determine initial placement of substitution and vacancy defects, the pristine unit cell was expanded into a supercell of size at least $3\times3\times3$ that was large enough to accommodate at least one occurrence of every defect species indicated in the chemical formula. The positions of the supercell atoms were fixed, and (4.3) was optimized by a genetic algorithm, where crossover and mutation operations were applied to a large pool of sets of selected sites to produce each generation of configurations. 25% of the the sets with the lowest E_{coh} value were selected from each generation until (4.3) showed no average improvement between subsequent generations. The same algorithm was used to optimize the placement of interstitial defects, but the positions of the interstitial sites were relaxed to a local minimum via gradient descent after mutation and crossover operations were applied.

4.2.4 Structure Optimization

The introduction of defects in a material can distort the material lattice around the defect sites, sometimes resulting in a structure that is unstable altogether. To account for these changes and to verify the stability of generated defect structures, we performed more rigorous structural optimization calculations using density functional theory (DFT). Variable-cell relaxation calculations were performed in Quantum ESPRESSO [92] using the Perdew-Berke-Ernzerhof (PBE) exchange correlation

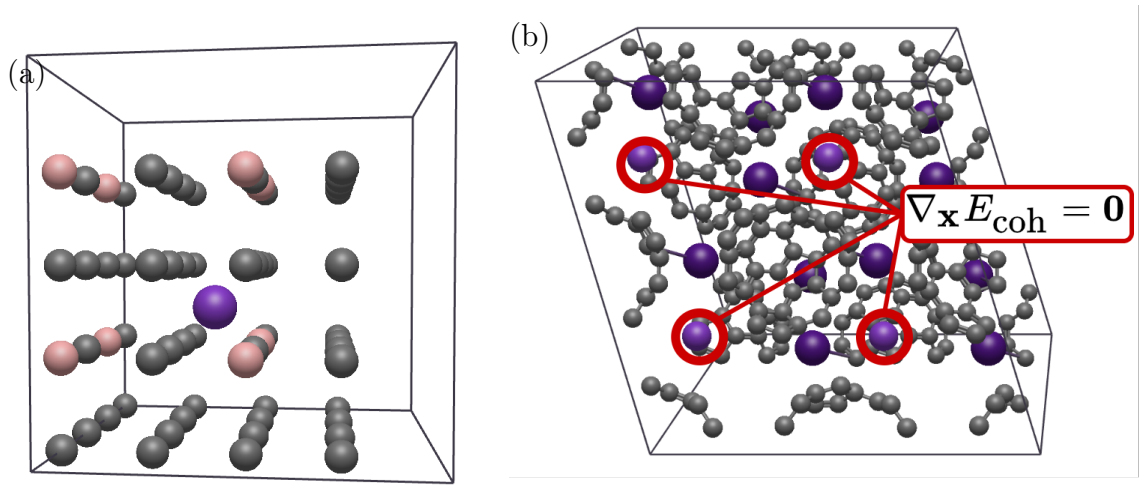

Figure 4.2: Visual examples of the results of genetic algorithm optimization on the selection of defect sites. (a) Result of substituting 8 B atoms (pink) and adding an interstitial K atom (purple) into a fictitious $4 \times 4 \times 4$ simple cubic carbon supercell. One can verify that this configuration is indeeed the global minimum of E_{coh}, which also maximizes the space between defect sites. (b) Result of adding interstitial K atoms to transform the superconducting fulleride Cs_3C_{60} into KCs_3C_{60}, where the positions of the interstitial K atoms were estimated via gradient descent with respect to E_{coh}. Even with complex supercell geometries, reasonable predictions of defect placement can be made.

functional [93] and pseudopotentials from the Standard Solid State Pseudopotential Efficiency library (SSSP-Efficiency) [94] to optimize both the atom positions and the lattice constant of the generated supercell. If the material had an associated pressure in the dataset, the relaxation calculation was carried out at the indicated pressure. The total energy and force conditions for convergence were set to 10^{-3} Ryd and 10^{-3} and Ryd/Bohr respectively. If these convergence criterion were not met, the structure was considered unstable and not included in the final structure dataset.

4.2.5 ALIGNN Model

To make predictions from atomic structure, we used the Atomistic Line Graph Neural Network (ALIGNN) model [34]. ALIGNN is a graph neural network model

that has attained great success in many different material property prediction problems. Currently, it is among the top performing deep learning models on the structure-based prediction problems in the Material Project's Matbench model benchmarking suite [95], and has even been applied successfully in the prediction of T_c for the limited case of conventional BCS superconductors [87]. A key benefit of this model is that in addition to representing a crystal lattice as a graph of atoms connected by bonds, it also employs a second graph of bonds, where every bond pair is connected by a bond angle. The use of this second graph (i.e. the atomistic line graph) is crucial, as it directly incorporates the bond geometry of the material, whereas using only a graph of atoms does not. For additional details on the ALIGNN model architecture, we refer the reader to the original paper by Choudhary et al. [85].

4.3 Results

4.3.1 Prediction of Critical Temperature

Now, we turn our attention to the problem of predicting superconducting critical temperature. In the case of unconventional superconductors, this problem is known to be exceptionally difficult since there are no general *ab initio* methods that can predict T_c in these materials. Fortunately, the abundance of experimental T_c data for superconductors makes deep learning-based approaches an attractive alternative to theoretical models. However, training deep learning models on experimental data comes with its own unique set of challenges, especially when the data was aggregated from multiple sources. In our dataset, measurements of superconducting T_c were obtained primarily from databases and academic papers. As a result, some of the T_c

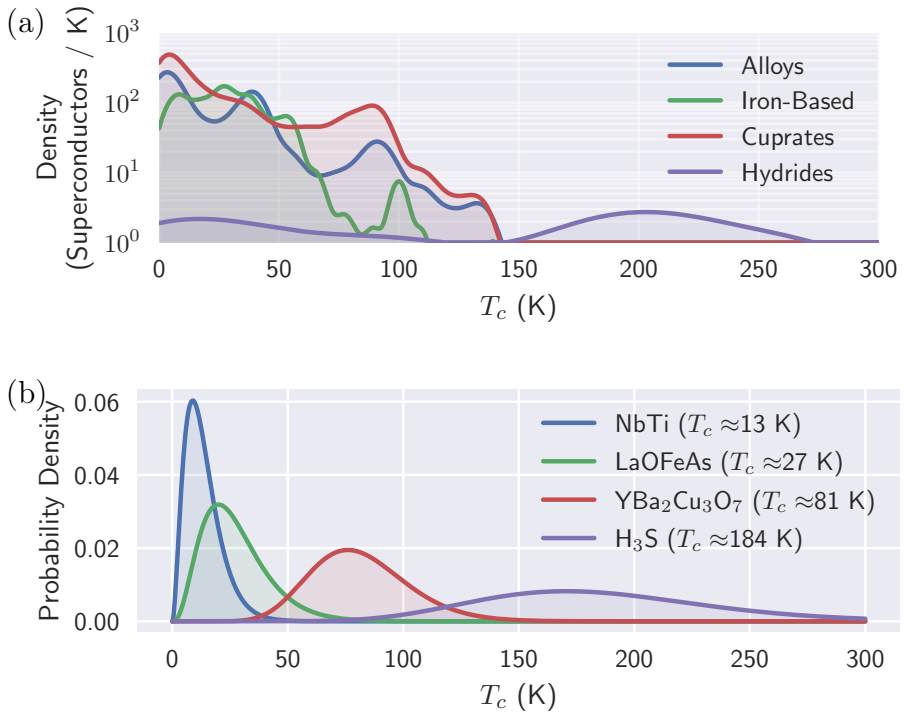

Figure 4.3: (a) Approximate distribution of the alloy, iron-based, cuprate, and hydride families of superconductors in the dataset with respect to T_c. (b) Examples of gamma distributions fitted to the empirical distributions of T_c for NbTi, LaOFeAs, $YBa_2Cu_3O_7$ (YBCO-123) and H_3S, which are members of the alloy, iron-based, cuprate, and hydride families respectively. Note that gamma distributions with mean $\mu = \kappa\theta \gg 0$ bear a close resemblance to a normal distribution, justifying the W_2 approximation used in (4.9).

values reported for the same material vary significantly due to differences in experimental conditions such as how the samples were prepared, and how the T_c of the sample was measured. This effect is perhaps most dramatic in the hydrides, where varying the pressure applied to the material in the range of 100 to 200 GPa can cause the T_c to vary by as much as 200 K [96]. Significant variations are also present in other superconductors such as FeSe. This material has a bulk T_c of $9K$, but when it is grown as a monolayer film on a $SrTiO_3$ substrate, a T_c value high as 100 K can be attained [97].

Our solution to this problem is to avoid predicting T_c directly and to instead predict an empirical probability distribution that estimates the range of T_c values attainable with a given material. If the empirical distribution of a material's T_c is positive and unimodal, a reasonable fit to the empirical distribution is given by the gamma distribution:

$$f(x) = (\Gamma(k)\theta^k)^{-1} x^{k-1} e^{-x/\theta}. \tag{4.7}$$

Above, κ and θ are the scale and shape parameters that characterize this distribution. A gamma distribution can be fit to empirical data using the central sample moment estimators $\hat{\kappa} = \mu^2/\sigma^2$ and $\hat{\theta} = \sigma^2/\mu$, where μ and σ are the sample mean and standard deviation. In Fig. 4.3a, we show the distribution of superconductors in the dataset with respect to T_c, while in Fig. 4.3b we plot examples of the gamma distributions fit to the reported T_c values.

To train the ALIGNN model to predict the distributions of T_c, we used the squared $p = 2$ Wasserstein loss function, which is a measure of "distance" between

probability distributions. It is given by:

$$W_2(f_1(x), f_2(x))^2 = \inf_{\gamma(x,y)} \mathbb{E}_{\gamma(x,y)} \left[|x - y|^2 \right], \tag{4.8}$$

where $\gamma(x, y)$ is a joint probability distribution with marginals equal to $f_1(x)$ and $f_2(y)$ (i.e. $\int_{-\infty}^{\infty} \gamma(x, y)\, dy = f_1(x)$ and $\int_{-\infty}^{\infty} \gamma(x, y)\, dx = f_2(y)$). Equation (4.8) does not readily admit a closed-form solution for gamma distributions, but if we assume that the distributions f_1 and f_2 are approximately normal with means μ_1, μ_2 and standard deviations σ_1, σ_2, we can use the approximation

$$W_2(f_1(x), f_2(x))^2 \approx |\mu_1 - \mu_2|^2 + |\sigma_1 - \sigma_2|^2, \tag{4.9}$$

which is the reduced form of the exact solution of (4.8) for univariate normal distributions [98].

Prior to training, the empirical T_c mean μ and standard deviation σ were computed for each material, and the materials were grouped by pristine material structure (i.e. the structure of the material without defects). These groups were then partitioned according to an 80%-10%10% training-validation test split. The motivation for this grouping was to maximize the independence of the training, test, and validation data. Although grouping decreases the diversity of examples in the training set and can result in a more pessimistic estimate of model accuracy, it is necessary to obtain a reliable estimate of how well the model can generalize to new materials that lie outside the support of the training data.

We trained the ALIGNN model to predict the mean μ and standard deviation σ of the empirical distribution of T_c for each material using the loss function (4.9). The number of training epochs and model hyperparameters were selected such that the validation set loss was minimized. The results of evaluating the trained model on

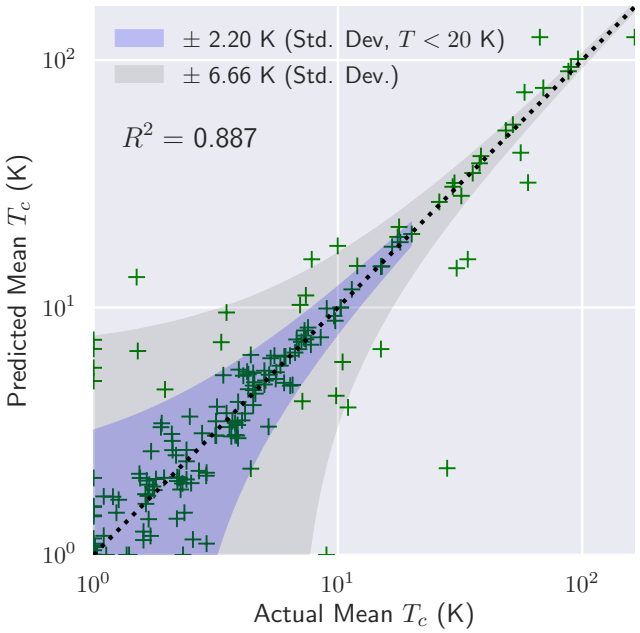

Figure 4.4: Log-scale plot of predicted versus actual mean T_c (i.e. μ) for superconductors in the test set. For low-temperature superconductors ($T < 20$ K), the model predicted the mean of the test set T_c distributions with a standard deviation of 2.20 K. For all T_c values, the prediction standard deviation was 6.66 K, and the R^2 value was 0.887. An R^2 value of 0.887 indicates reasonably good accuracy, especially considering that none of the structures in the test set shared the same pristine structure with a training set structure.

the reserved test set are shown in Fig. 4.4, where we found the standard deviation of the predicted μ to be ± 6.66 K with an R^2 value of 0.887. Most of this prediction error can be attributed to the high variation in μ measured for high-T_c superconductors, especially the hydrides. We also observed that predictions of mean T_c made for superconductors in the low-T_c regime (i.e. with $T_c < 20$ K) had a standard deviation of ± 2.20 K. This is roughly comparable to the degree of accuracy attained in other studies that applied graph-based models to BCS superconductors [34, 86].

4.3.2 Identification of Novel Superconductors

Having shown that the empirical distributions of T_c can be computed with reasonable accuracy, we now turn our attention to the problem of identifying superconductors. This problem is most easily interpreted as a binary classification task, where a model is trained to maximize the accuracy of classified examples. To perform this classification, we first modified the ALIGNN model architecture to serve as a binary classifier. Next, to generate the negative class training data, we sampled more than 10,000 distinct materials from the Materials Project database with elements that were drawn randomly according to the distribution visualized in Fig. 4.1b. Randomly sampled materials that had a composition distance less than 0.05 from any superconductor in the dataset were thrown out. The remaining list of randomly sampled materials were labeled as non-superconducting in the training data. These structures were combined with the superconducting structures in the dataset and split into 80%-10%-10% training, validation, and test sets. As with the T_c prediction model, structures derived from the same pristine material were grouped in order to increase the independence of the training, validations and test sets.

The classifier model was trained on the generated data using the cross-entropy loss function. After each epoch, the area under the Receiver Operating Characteristic (ROC) curve was evaluated for the validation set, and the model weights that attained the greatest area under the ROC curve (AUROC) on the validation set were saved. Overall, it was found that the model converged to optimal classification accuracy in roughly 30 epochs, while the training loss required roughly 80 epoch to converge.

(a)

Contained in Dataset		Predicted Superconductor: Yes	Predicted Superconductor: No
	Yes	Total: 102 $TaSe_3$, AlCu, W_2C, Nb_3, RhGe, Ce, $TlNiSe_2$, $SrIr_2$	Total: 92 $LaPd_2Al_3$, Ru_2Zr, In_3Sn, La_3Sn, $PdZr_2$
	No	Total: 95 $YbLuB_{24}$, Ti_3Ir*, $CePb_3$*, $SrAu_5$, $SrGe_3$*	Total: 917 Sc_3Nb, Yb_5Sn_3, $Ba(SmSe_2)_2$, $YAsO_4$, Th_2Zn

* Materials that are actual superconductors, but were not contained in the dataset.

(b)

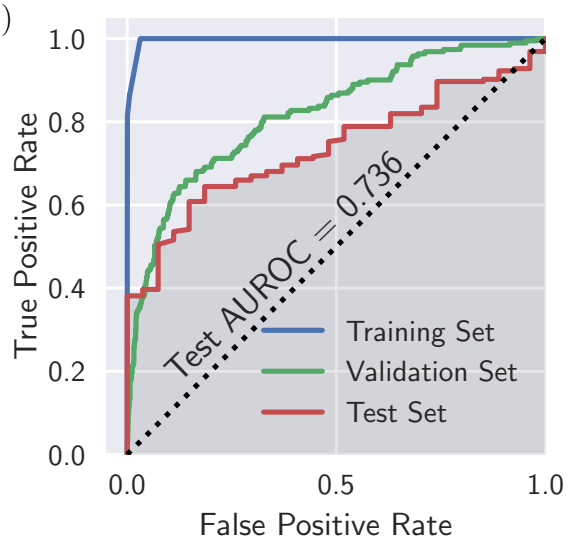

Figure 4.5: Results of the ALIGNN binary classifier model. (a) Test set confusion matrix with examples of each type of correct and incorrect classification. Some of the false positive predictions happened to be actual superconductors, but since they were not contained in the dataset, they were considered non-superconducting. (b) ROC curves for the training, validation and test sets, with a reported test set AUROC of 0.736. Typically, an AUROC of 0.7 or higher indicates good performance for hard classification tasks.

In Fig. 4.5a, we plot the confusion matrix for predictions made on the test set, giving examples of the correct and incorrect classifications made. At the default decision threshold, the classifier had an estimated true positive rate of roughly 0.52 and an estimated false-positive rate of 0.09. Since the randomly sampled materials that were not similar to any material in the dataset were considered non-superconducting in the training data, there were a few of these materials that were actual superconductors but were not contained in our dataset and thus were erroneously labeled as non-superconducting. Some of these materials resulted in false positive classifications (indicated with red text in Fig. 4.5a) even though they were in fact accurate classifications.

The predictions reported in the confusion matrix and their corresponding true positive and false positive rates were obtained using the model's default classification threshold, though the tradeoff between these rates can in fact be tuned along the ROC curve shown in Fig. 4.5d. For the test set, we obtained an AUROC of 0.736, suggesting that the model has a reasonable discriminative capability. Given that the materials were grouped by pristine material prior to being partitioned into training, validation and test sets, the AUROC being greater than 0.7 suggests that the classifier is capable of making predictions that generalize outside the training set. This is also supported by the presence of actual superconductors in the false positive quadrant of the confusion matrix, as these materials were not contained in the dataset but were nonetheless classified correctly.

4.3.3 High-Throughput Screening

To demonstrate the feasibility of using our binary classifier to identify new superconductors, we screened a collection of 40,000 metals, oxides, metallic compounds, and hydrides from the Materials Project database. To ensure that discovered candidates were sufficiently different from known superconductors, only materials with a composition distance of at least 0.1 from every superconducting material in the dataset were considered as candidates. Our classifier identified roughly 600 candidate superconductors consisting of mostly non-oxide materials, as most of the predicted superconduting oxide materials were too similar to those in the dataset. Some of the candidates have also not yet been experimentally observed. A sample of these candidates is given in Table 4.1.

There are many different types of potential superconductors represented in Table 4.1, including borides ($HfZrB_4$, B_3Os_2, V_3B_{100}), hydrides (CH_4, Ce_4H_11), and heavy fermion-like metals ($CeCuSb_2$, $HoUTe_3$). For each candidate, we also list if the material is predicted stable, if the material has been observed experimentally, the material's predicted mean T_c (using our trained T_c distribution regression model), and the closest superconductor in the dataset with respect to (4.1). With the exception of the identified hydride compounds, almost all of the identified candidates have an estimated mean T_c less than 20 K. In Fig. 4.6, we visualize the T_c distribution for six of the candidates in Table 4.1. For these six materials, the predicted distributions of T_c values range from 1 to 60 K, with the hydride compounds H_4C (crystallized methane) and NpH_3 having high mean T_c values of 15 and 51 K respectively. To be realized experimentally, these hydride compounds will likely need to be subjected to

Table 4.1: Sample of Identified Candidate Superconductors.

Materials Project ID	Formula	Stable	Experimentally Observed	Predicted Mean T_c	Closest Superconductor in Dataset
mp-672238	$CeCuSb_2$	Yes	Yes	1.88 K	Cu_2Sb
mp-1025564	$LuAl_2Pd_5$	Yes	No	8.12 K	Pd
mp-10898	$ScAlNi_2$	Yes	Yes	1.25 K	Ni_3Al
mp-573601	Th_7Ru_3	Yes	Yes	0.95 K	Th
mp-28280	$K_5V_3O_{10}$	Yes	Yes	4.93 K	KO_3
mp-1224184	$HfZrB_4$	Yes	No	2.57 K	HfB_2
mp-1228895	$AlGaSb_2$	No	No	4.15 K	AlSb
mp-1222266	Lu_3S_4	Yes	No	3.63 K	LuS
mp-1079796	Ti_3Pd	Yes	No	4.24 K	Ti
mp-1218331	Sr_3CaSi_8	No	No	1.88 K	$Sr(Si)_2$
mp-1021328	H_4C	Yes	No	50.97 K	H_2
mp-11494	$LuPb_3$	No	Yes	3.80 K	Pb
mp-1226890	Ce_4H_{11}	Yes	No	331.83 K	CeH_9
mp-22266	GdB_6	Yes	Yes	2.22 K	B_3
mp-1184695	Ho_3Er	No	No	6.22 K	Ho
mp-9605	$Hf_3Si_3Ni_2$	Yes	Yes	9.54 K	Hf_2Ni
mp-653995	$As_4Pb_8Cl_6O_{11}$	No	Yes	13.01 K	ClO_4
mp-2805	$LuAl_3$	Yes	Yes	1.29 K	Al
mp-1103958	$GdIr_3$	Yes	Yes	8.10 K	Ir
mp-1227704	$Ca_3Nb_4(O_6F)_2$	No	No	15.34 K	Ca_2NbO_4
mp-643270	K_3ZnH_5	No	Yes	20.38 K	KH
mp-1216517	V_3B_{100}	No	No	11.95 K	B
mp-1224291	$HfTaB_2$	No	No	3.07 K	HfB_2
mp-1190843	NpH_3	No	Yes	15.42 K	H
mp-292	V_3As	Yes	Yes	2.34 K	V
mp-28385	$Sc_5(Re_2Si_3)_4$	Yes	Yes	4.74 K	Re_2Si
mp-862665	$AcGe_3$	Yes	No	3.04 K	Ge_2
mp-1220204	$NdCuNi_4$	Yes	No	4.03 K	$NdNi_5$
mp-1220417	Nb_6PdRh	No	No	6.03 K	Nb_3Pd
mp-1585	V_3Co	Yes	Yes	3.97 K	V
mp-1216575	Tl_4Bi	No	No	6.92 K	Tl
mp-1091395	ZrGeOs	Yes	Yes	4.63 K	Os_2Zr
mp-1184672	$HfTi_3$	No	No	5.14 K	Ti
mp-1120790	Na_2VF_5	Yes	No	23.43 K	V
mp-1224195	$HoUTe_6$	No	No	2.38 K	$HoTe_3$
mp-20603	$CeTl_3$	Yes	Yes	0.96 K	Tl
mp-1196837	$Ca(Al_{10}Cr)_2$	No	Yes	2.81 K	Al
mp-12763	TbB_6	No	Yes	2.75 K	B
mp-20109	Th_2NiB_{10}	Yes	Yes	5.57 K	B_6Th
mp-1079348	B_3Os_2	Yes	Yes	2.22 K	OsB_2

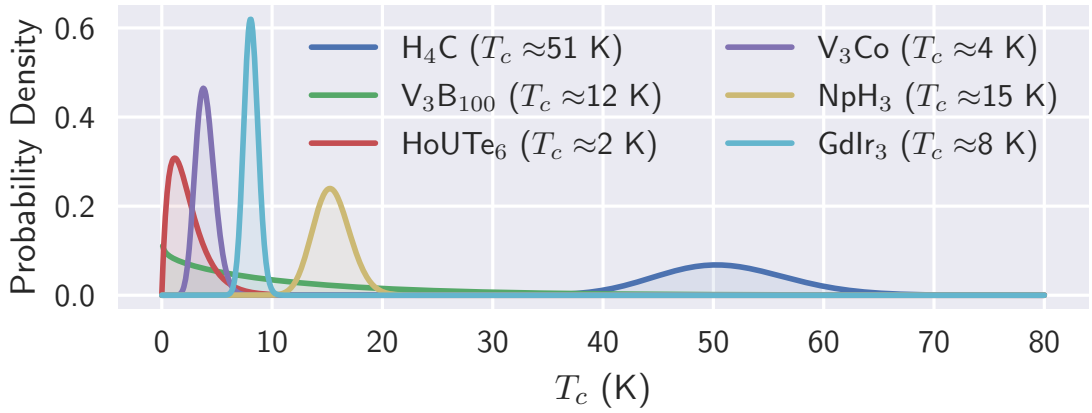

Figure 4.6: Predicted T_c distribution of selected superconductor candidates in Table 4.1. To generate these predictions, we used the ALIGNN T_c regression model we trained on known superconductors.

high pressures (in the range of 10 - 300 GPa) and possibly be doped with additional elements (such as Be in the case of CH_4 [99]).

4.4 Conclusion and Future Work

To summarize, we have demonstrated use of a material structure-based deep learning model (ALIGNN) to identify and predict the T_c of new superconductors using experimental data from the *Supercon v1* and *v2* datasets. Since the experimental data only contained the chemical formula of each measured superconductor, the crystalline structures in the dataset had to be inferred by first retrieving the pristine structure from the Materials Project database, predicting the rough locations of defects with a genetic algorithm (if indicated in the chemical formula), and finally relaxing the structure with more rigorous DFT calculations. Once we constructed the dataset, we trained a T_c regression model and a binary classifier to predict T_c and identify new superconducting materials. After using these models to screen the Materials Project Database, we identified a diverse set of more than 600 candidate superconductors.

Although this model serves as a strong proof-of-concept, there is still a need for significant improvements to be made in both the the quality of the dataset and the data features taken into account by the model. Currently, the model does not directly incorporate factors such as strain, applied pressure, or features such as the substrate on which a superconductor was grown, whether the sample was bulk or monolayer, monocrystalline or polycrystalline, etc. The model also does not directly incorporate information regarding the band structure or electronic interactions of a material, which are currently understood to play an important role in unconventional superconductivity, especially in iron-based, and rare earth magnetic superconductors [100,101]. Developing new deep learning models that can account for these electronic features may be a promising direction of further investigation.

CHAPTER FIVE

Quantum-Informed Machine Learning of Electronic Band Structure

5.1 Introduction

Electronic structure prediction lies at the heart of materials science and quantum chemistry, providing crucial insight into how atoms and electrons interact to determine the properties of matter. Traditional approaches, such as density functional theory (DFT), offer principled solutions to the quantum mechanical equations that govern these systems but remain computationally expensive, especially for large or complex materials. The emergence of machine learning in the physical sciences has inspired a new generation of surrogate models for accelerating quantum simulations. However, many such models suffer from limitations in physical interpretability, symmetry enforcement, or scalability. In response to these challenges, we introduce a novel class of models, namely *Quantum-Informed Neural Networks* (QuINNs), that blend rigorous quantum mechanical constraints with state-of-the-art machine learning architectures. QuINNs represent a unique, physically grounded innovation that addresses longstanding computational bottlenecks in electronic structure modeling.

QuINNs explicitly preserve the quantum mechanical structure of solutions to the electronic time-independent Schrodinger equation. This is achieved by using machine learning not to approximate energies or properties directly, but to predict the local basis functions and pseudopotentials from which an electronic Hamiltonian is assembled. Each component (atomic orbitals, potential terms, and Hamiltonian entries) is derived from Gaussian basis functions with learned parameters constrained

by symmetry. This approach stands in contrast to traditional DFT methods, which must iteratively solve a self-consistent field problem by evaluating functionals over the electron density, often requiring hundreds of iterations and matrix diagonalizations per system. It also differs from neural network-based surrogate models, which often regress energy levels or forces from raw atomic positions without a physically interpretable intermediate representation, such as the electronic orbital structure in 3D space [102, 103]. In QuINNs, all physical quantities (including the band energies and electron orbitals in 3D space) are derived through a physically meaningful Hamiltonian matrix. This results in models that are not only predictive, but physically structured and interpretable.

The utility of QuINNs lies in their ability to serve as a high-fidelity, low-cost alternative to ab initio electronic structure methods, with full support for differentiable workflows. Once trained, a QuINN model can compute band structures, density of states, Fermi levels, and various topological quantities directly from a crystal structure in a single forward pass, without requiring self-consistent convergence or numerical quadrature over the density. Furthermore, because every stage of the computation from orbital generation to Hamiltonian construction to diagonalization is differentiable, QuINNs can compute gradients of quantum observables with respect to atomic geometry. This makes them ideally suited for inverse design problems such as band gap tuning, defect placement, and lattice optimization. Existing DFT workflows, even when accelerated with GPU-based solvers or tight-binding approximations, still require expensive iterative loops and often lack the smooth differentiability needed for efficient optimization. By contrast, QuINNs support end-to-end learning and backpropagation, making them suitable for embedding within larger

deep learning architectures, such as reinforcement learning agents, or gradient-based materials design pipelines [104].

What makes QuINNs particularly novel is their architectural formulation, which incorporates both the symmetries of quantum mechanics and the structure of ab initio simulation pipelines in a learnable format. In conventional neural network architectures, one typically maps a vector of atomic positions to a target property (e.g., energy, force, or dipole moment). These architectures, while sometimes accurate, often ignore the algebraic and geometric constraints imposed by quantum mechanics, such as gauge symmetry, reflection symmetry, or the requirement that Hamiltonians be Hermitian and block-diagonal in momentum space. In contrast, QuINNs respect these constraints at every level: the orbitals transform as irreducible representations of the Euclidean group $E(3)$, the potential terms are expressed in rotationally equivariant Gaussian bases [105]; and the Hamiltonian is constructed to be sparse, Hermitian, and periodic with respect to the reciprocal lattice. This architectural design enforces structure-preserving inductive biases that dramatically reduce the number of parameters needed to learn quantum behavior, while retaining interpretability and extensibility to new systems. This fundamentally different modeling philosophy goes beyond empirical fitting or black-box learning, placing QuINNs in a new category of physically structured, symmetry-informed computational models.

Moreover, QuINNs exhibit a degree of generality and extensibility that further distinguishes them from prior approaches. Traditional tight-binding models, for example, are limited to small numbers of orbitals per atom and typically require hand-crafted parameterizations of hopping amplitudes and onsite energies. While some recent graph-based machine learning models do incorporate equivariance or

symmetry-aware attention mechanisms, they are still trained end-to-end to predict observables and do not generate an interpretable Hamiltonian from which electronic quantum states can be reconstructed. QuINNs, in contrast, serve as surrogate quantum simulators, providing not just predictions but complete quantum mechanical representations of materials systems. These representations can be exported, inspected, modified, and re-used for downstream simulation tasks. The introduction of QuINNs thus represents not merely a speedup, but a conceptual shift in how electronic structure can be modeled: from solving a series of self-consistent numerical equations to learning a structured, physically grounded map from atomic geometry to quantum electronic structure to macroscopic properties.

5.2 Background

5.2.1 Physics-Informed Neural Networks (PINNs)

In this chapter, we introduce the theoretical foundation of QuINN models, which arise as an instance of a broader class of model, namely the class of *Physics-Informed Neural Networks* (PINNs) [32, 106]. The core idea behind PINNs is to solve partial differential equations (PDEs) by training a neural network as a universal approximator to some unknown or computationally complex function, with the goal of learning either the solution to a PDE or some explicitly solvable form of the PDE itself. The “physics-informed" aspect of PINNs arises from both (a) how the neural network approximator is constrained so as to avoid physically unrealistic solutions, and (b) how known physical PDEs and boundary conditions are incorporated into the loss function to train the neural network. Approach (a) is used to incorporate

the basic symmetries and any "hard" limitations to the neural network approximator, while approach (b) is used to incorporate "soft" physical constraints to the model.

Consider, for example, a time-dependent non-linear PDE that admits a solution $u(x,t)$. The PDE can be written in the form:

$$\hat{\mathcal{N}}[u(x,t),\lambda] = 0 \text{ for all } x \in \Omega, t \in [0,T] \tag{5.1}$$

where Ω and $[0,T]$ represents the space and time domains of u, and $\hat{\mathcal{N}}$ is some (typically non-linear) operator parameterized by a set of features λ. These features λ should ideally include all quantities that are needed to obtain a unique solution to the PDE, such as initial conditions, boundary conditions, and any additional scalar or vector parameters related to the operator $\hat{\mathcal{N}}$. For example, if $\hat{\mathcal{N}}$ represents the viscous Burgers' equation, we would use

$$\hat{\mathcal{N}}[u(x,t),\lambda] = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - v\frac{\partial^2 u}{\partial x^2}, \tag{5.2}$$

where

$$\lambda = (u(x,0), v) \tag{5.3}$$

contains both the initial condition $u(x,0)$ and the viscosity parameter v [107]. In order to train a PINN to model the solution to a PDE, a sufficiently large data set of the form $\{(u_n, x_n, t_n, \lambda_n)\}_{n=1}^{N}$ must be obtained through simulation or experiment. We then introduce a neural network model $F(x,t,\lambda)$ that computes the solution to the PDE. The neural network inside the model F can be trained to either directly compute the solution $u(x,t)$, or the solution's time derivative. In the later case, one uses a neural network combined with a numerical integrator to compute F.

Training a PINN requires incorporating both the error of the model $F(x,t)$ with respect to the true solution $u(x,t)$ (the "data" loss) in addition to the error of the

residual of the PDE operator $\hat{\mathcal{N}}$ (the "PDE" loss). These loss functions are combined with weights w_{data} and w_{PDE} to serve as the total loss function used to train the PINN:

$$L_{\text{total}} = w_{\text{data}} L_{\text{data}} + w_{\text{PDE}} L_{\text{PDE}}. \tag{5.4}$$

Using a mean square error with respect to a data set $\{(u_n x_n, t_n, \lambda_n)\}_{n=1}^{N}$, the data loss assumes the form

$$L_{\text{data}} = \frac{1}{N} \sum_{n=1}^{N} (F(x_n, t_n, \lambda_n) - u_n)^2, \tag{5.5}$$

while the PDE loss assumes the form

$$L_{\text{PDE}} = \frac{1}{N} \sum_{n=1}^{N} (\hat{\mathcal{N}}[F(x_n, t_n, \lambda_n), \lambda_n])^2. \tag{5.6}$$

By choosing w_{data} and w_{PDE} carefully, the PINN can be trained in a manner that strikes an optimal balance between accuracy ($w_{\text{data}} \gg w_{\text{PDE}}$) and physical correctness ($w_{\text{PDE}} \gg w_{\text{data}}$).

5.2.2 Density Functional Theory

5.2.2.1 Quantum mechanics and the Schrödinger equation. QuINN models are a type of PINN model that are designed to solve for the ground state electronic structure of solids in a manner that is consistent with the Schrödinger Equation. The Schrödinger equation is a foundational equation that governs the behavior of quantum mechanical systems, such as the electrons in a solid material. We describe quantum-mechanical objects mathematically as a wave function, $|\psi(x)\rangle$, whose modulus squared $|\psi(x)|^2$ describes the probability density of the object in space. To denote many-body ensembles of particles (such as electrons), we use the notation

$|\Psi(x)\rangle$. The Schrödinger equation describes the non-relativistic time evolution of these wave functions as:

$$\frac{\partial}{\partial t}|\Psi(x)\rangle = \frac{-i}{\hbar}\hat{H}|\Psi(x)\rangle \tag{5.7}$$

where $\hat{H}$ is the many-body Hamiltonian operator, incorporating the sum of potential and kinetic energy contributions to the total energy of the system. Since the Schrodinger equation is first-order and linear, its solution is given by

$$|\Psi(\mathbf{x},t)\rangle = \exp\left(\frac{-i}{\hbar}\hat{H}\right)|\Psi(x,0)\rangle\,. \tag{5.8}$$

While this solution is mathematically correct, solving for the exponential of a many-body Hamiltonian operator $\hat{H}$ is generally not possible in closed form, unless a complete and orthonormal basis of eigenfunctions $|\Psi_n(\mathbf{x})\rangle$ with eigenvalues E_n of the operator $\hat{H}$ can be found. To do this, one must solve the *time-independent Schrodinger equation*, which is the eigenvalue problem

$$\hat{H}|\Psi_n(\mathbf{x})\rangle = E_n|\Psi_n(\mathbf{x})\rangle\,, \tag{5.9}$$

where E_n is the nth largest eigenvalue of $\hat{H}$. If $\hat{H}$ models a system of interacting electrons, we say that the lowest eigenvalue E_n of $\hat{H}$ is the *ground state energy*, while the associated eigenfunction $|\Psi_0(\mathbf{x})\rangle$ is the *ground state electronic structure*. In most cases, calculating the properties of atoms, molecules, and materials from first principles requires first knowing the electronic ground state. Thus, computing this state accurately and efficiently is crucial in the context of simulating materials and other atomic systems. To obtain the ground state of an electronic system, we must find the eigenstate of $\hat{H}$ with the lowest energy E_0. This amounts to solving the

variational optimization problem

$$\begin{aligned} &\text{Minimize: } E = \langle \Psi(x) | \, \hat{H} \, | \Psi(x) \rangle \\ &\text{Subject to: } \langle \Psi(\mathbf{x}) | \Psi(\mathbf{x}) \rangle = 1. \end{aligned} \tag{5.10}$$

The the many-body wave function $|\Psi\rangle$ that solves (5.10) is by definition the ground state electron structure, denoted as $|\Psi_0\rangle$, with ground state energy E_0.

Next, we consider the case of solid state materials and molecules. For these systems we use the Hamiltonian

$$\begin{aligned} \hat{H} = &\sum_{\mathbf{k},\sigma} \sum_{\mathbf{k}',\sigma'} h_{(\mathbf{k},\sigma),(\mathbf{k}',\sigma')} \hat{c}^{\dagger}_{\mathbf{k},\sigma} \hat{c}_{\mathbf{k}',\sigma'} \\ &+ \frac{1}{2} \sum_{\mathbf{k},\sigma} \sum_{\mathbf{k}',\sigma'} \sum_{\mathbf{q}} V_{\sigma,\sigma'}(\mathbf{q}) \hat{c}^{\dagger}_{\mathbf{k}+\mathbf{q},\sigma} \hat{c}^{\dagger}_{\mathbf{k}'-\mathbf{q},\sigma'} \hat{c}_{\mathbf{k},\sigma} \hat{c}_{\mathbf{k}',\sigma'} \\ &+ V_{\text{ext}}(\mathbf{x}) \end{aligned} \tag{5.11}$$

where the operators $\hat{c}^{\dagger}_{\mathbf{k},\sigma}/\hat{c}_{\mathbf{k},\sigma}$ create/annihilate an electron in a spin orbital σ with crystal momentum vector $\mathbf{k}$. For non-periodic systems like molecules, this degree of freedom disappears, and one only needs to consider $\mathbf{k} = \mathbf{q} = \mathbf{0}$. The coefficients $h_{(\mathbf{k},\sigma),(\mathbf{k}',\sigma')}$ comprise the entries of the single-particle Hamiltonian for two orbitals $\phi_{\mathbf{k},\sigma}(\mathbf{x})$ and $\phi_{\mathbf{k}',\sigma'}(\mathbf{x})$, computed as

$$h_{(\mathbf{k},\sigma),(\mathbf{k}',\sigma')} = \langle \phi_{\mathbf{k},\sigma}(\mathbf{x}) | \left(-\frac{1}{2}\nabla^2 - \mu + V_{\text{ext}}(\mathbf{x}) \right) | \phi_{\mathbf{k}',\sigma'}(\mathbf{x}) \rangle \tag{5.12}$$

where μ is the chemical potential and $V_{\text{ext}}(\mathbf{x})$ is the external potential induced by the positively charged atomic nuclei (assumed to be stationary under the Born-Oppenheimer approximation), and any external electrostatic interactions (such as an applied electric field).

In the full Hamiltonian, $\hat{H}$, the two-particle interaction term $V_{\sigma,\sigma'}(\mathbf{q})$ incorporates the Coulomb scattering of electrons resulting in a transfer of momentum $\mathbf{q} = \mathbf{k} - \mathbf{k}'$. Finally V_{ext} incorporates the external potential induced by the positively

charged atomic nuclei (assumed to be stationary under the Born-Oppenheimer approximation), and any external electrostatic interactions (such as an applied electric field). We note that in the equation above and in the remainder of this document, we will assume all physical quantities are expanded in atomic units, so that $\hbar = 1$ and $m_e = 1$, which increases the clarity of our notation.

5.2.2.2 Density functional theory and the ground state. In the previous section, we introduced the many-body electron Hamiltonian $\hat{H}$, which we expanded in terms of the creation/annihilation operators $\hat{c}^{\dagger}_{\mathbf{k},\sigma}/\hat{c}_{\mathbf{k},\sigma}$. For discrete systems containing N orbitals (including both the σ and $\mathbf{k}$ degrees of freedom), we can represent these operators as sparse matrices of size $2^N \times 2^N$. This means that the Hamiltonian $\hat{H}$ can be represented as a $2^N \times 2^N$ matrix, spanning the full Fock space of N distinct single-particle states, where each state is either occupied or unoccupied by an electron (resulting in the 2^N size). Diagonalizing this matrix is clearly intractable, even for molecular systems of modest size. As a result, we must resort to some type of approximation to make the problem tractable.

A powerful approach to avoid this intractability is given by density functional theory (DFT) methods [108,109], which first consider the non-interacting Hamiltonian

$$\hat{H}_{\mathrm{NI}} = \sum_{\mathbf{k},\sigma} \sum_{\mathbf{k}',\sigma'} h_{(\mathbf{k},\sigma),(\mathbf{k}',\sigma')} \hat{c}^{\dagger}_{\mathbf{k},\sigma} \hat{c}_{\mathbf{k}',\sigma'}, \tag{5.13}$$

where the Coulomb interaction terms are removed. Then, a mean-field treatment of the Coulomb potential is introduced, called the Hartree functional $\hat{\mathcal{J}}[n(\mathbf{x})]$. This potential accounts for the electrostatic Coulomb potential with respect to the electron density $n(\mathbf{x})$ in a mean field fashion:

$$\hat{\mathcal{J}}[n(\mathbf{x})] = \int_{\Omega} \frac{n(\mathbf{x})n(\mathbf{x}')}{\|\mathbf{x}-\mathbf{x}'\|}\, d\mathbf{x}' \tag{5.14}$$

The ground state energy of the many-body wave function can then can then be exactly written in the form

$$E_0 = \langle \Psi_0(\mathbf{x}) | \hat{H}_{\mathrm{NI}} | \Psi_0(\mathbf{x}) \rangle + \hat{\mathcal{J}}[n_0(\mathbf{x})] + \hat{\mathcal{V}}_{\mathrm{xc}}[n_0(\mathbf{x})] \tag{5.15}$$

where $|\Psi_0(\mathbf{x})\rangle$ is the many-body ground state, $\hat{\mathcal{J}}[n_0(\mathbf{x})]$ is the Hartree functional and $\hat{\mathcal{V}}_{\mathrm{xc}}[n_0(\mathbf{x})]$ is called the *exchange-correlation functional*, both of which are unique functionals of the ground state electron density $n_0(x)$ [108]. The argument for substituting this in place of the Coulomb interaction term can be summarized as follows:

(1) First, it can be shown that for systems with a well-defined number of electrons, the ground state electron density $n(\mathbf{x})$ uniquely determines the external potential V_{ext} and vice versa. This is a powerful result, known as the first Hohenberg-Kohn theorem [110]. It has two major consequences:

- Since the ground state density uniquely determines the external potential, which in turn determines the ground state many-body wavefunction $|\Psi_0(\mathbf{x})\rangle$, then for any quantum observable operator $\hat{O}$, the expectation value $\langle \hat{O} \rangle = \langle \Psi_0(\mathbf{x}) | \hat{O} | \Psi_0(\mathbf{x}) \rangle$ must also be a unique functional $\hat{\mathcal{O}}[n(\mathbf{x}))]$ of the ground state density.

- If we let $\hat{O} = \hat{H}$ this means that the ground state energy E_0 is a unique functional $\hat{\mathcal{H}}[n(\mathbf{x})]$ of the electron density. The same also applies if we let $\hat{O}$ consist of only the Coulomb interaction terms with coefficents $V(\mathbf{q})$ in the original electronic Hamiltonian. In this case, the resulting functional

is precisely the sum of the functionals $\hat{\mathcal{J}}[n(\mathbf{x})] + \hat{\mathcal{V}}_{\text{xc}}[n(\mathbf{x})]$ (i.e., the sum of the Hartree functional and the DFT exchange-correlation functional). In practice, this exchange correlation functional is not known to exist in closed form, but can be approximated numerically if certain simplifying physical assumptions are made about the electronic system of interest.

(2) Second, it can be shown that when the unique functional $\hat{\mathcal{H}}[n(\mathbf{x})]$ is minimized with respect to the electron density $n(\mathbf{x})$, then the minimizing density $n_0(\mathbf{x})$ is guaranteed to be the ground state electron density. This is known as the second Hohenberg-Kohn theorem [110].

Since the non-interacting Hamiltonian $\hat{H}_{NI}$ contains only single-particle terms, it can be represented in the single particle subspace as the dense $N \times N$ matrix $h_{(\mathbf{k},\sigma),(\mathbf{k}',\sigma')}$ in contrast to the sparse $2^N \times 2^N$ matrix representation of the full electronic Hamiltonian $\hat{H}$. In this single-particle framework, one can represent the electron density using the $N \times N$ electron density matrix ρ. The electron density matrix ρ corresponding to a many-body wave-function $\Psi(\mathbf{x})$ has entries

$$\rho_{(\mathbf{k},\sigma),(\mathbf{k}',\sigma')} = \langle \Psi(\mathbf{x}) | \, \hat{c}^{\dagger}_{\mathbf{k},\sigma} \hat{c}_{\mathbf{k}',\sigma'} \, | \Psi(\mathbf{x}) \rangle \, . \tag{5.16}$$

By definition, this density matrix is not normalized to unit trace ($\text{Tr}(\rho) = 1$), but instead is normalized to the total electron number:

$$\text{Tr}(\rho) = \langle \Psi(\mathbf{x}) | \, \hat{N} \, | \Psi(\mathbf{x}) \rangle = \int n(\mathbf{x}) \; dx. \tag{5.17}$$

Since the density matrix is expanded in terms of the basis of single-electron orbitals, then we can write the electron density as

$$n_{\rho}(\mathbf{x}) = \sum_{\mathbf{k},\sigma} \rho_{(\mathbf{k},\sigma),(\mathbf{k},\sigma)} |\phi_{\mathbf{k},\sigma}(\mathbf{x})|^2. \tag{5.18}$$

Now we can re-write the ground state optimization problem in terms of the density matrix ρ, the single-particle Hamiltonian matrix h, and the exchange correlation functional $\hat{\mathcal{V}}_{xc}$:

$$\begin{aligned} &\text{Minimize: } E = \text{Tr}(\rho\ h) + \hat{\mathcal{J}}[n_\rho(\mathbf{x})] + \hat{\mathcal{V}}_{xc}[n_\rho(\mathbf{x})] \\ &\text{Subject to: } \|\rho\| \leq 1 \end{aligned} \tag{5.19}$$

The constraint of $\|\rho\| \leq 1$ ensures that every electron state is occupied by at most one electron, but does allow for fractional electron occupations. Since any states with fractional occupations below the chemical potential μ can be fully filled with a single electron to decrease the total system energy, the density matrix ρ_0 that minimizes the ground state energy will not have any fractional occupations. In fact, allowing for fractional occupations while solving this minimization problem is a technique called *smearing*, which in some cases has been shown speed up convergence to a solution. Unfortunately, solving this optimization problem is quite difficult, owing to the fact that $\hat{\mathcal{V}}_{\text{xc}}(\mathbf{x})$ depends on the electron density $n_\rho(\mathbf{x})$, which in turn depends on ρ. Therefore, one cannot simply diagonalize h to obtain the ground state. Instead, one must resort to a number of numerical techniques that iteratively converge to the ground state density matrix.

5.2.2.3 Using Gaussian bases. There are many different numerical methods available to solve DFT problems, with varying level of stability and robustness of solution. Here, we will only focus one the most popular gradient-based methods, which are used in codebases like VASP [111] and Quantum ESPRESSO [92]. Typically, the type of methods that are available for use are also dependent on the basis of orbitals $\phi_{\mathbf{k},\sigma}(\mathbf{x})$ within which the density matrix is expanded. In periodic systems,

for example, bases of plane-waves are preferred, whereas for molecules, bases that are localized on atomic sites are preferred. In this section, we will focus on the use of Gaussian function bases, which are basis sets localized at atomic sites that can be modified (under the tight-binding approximation) to effectively model the physics of periodic systems as well. This is the basis set that we use in QuINN models.

In periodic crystal systems, it is necessary to take into account Bloch's theorem, which states that if the Schrodinger equation is solved within the unit cell Ω of a crystal structure to obtain a set of orbital wave functions $u_\sigma(\mathbf{x})$ satisfying the unit cell's periodic boundary conditions, then the solutions to the Schrodinger equation for the entire crystal (sometimes called "crystal orbitals") can be written in the form

$$\phi_{\mathbf{k},\sigma}(\mathbf{x}) = e^{i\mathbf{k}\cdot\mathbf{x}} u_\sigma(\mathbf{x}) \tag{5.20}$$

where $\mathbf{k}$ is a wave number confined to the reciprocal unit cell Ω^*, and $u_\sigma(\mathbf{x})$ is normalized such that

$$\int_\Omega |u_\sigma(\mathbf{x})|^2 \, d\mathbf{x} = 1. \tag{5.21}$$

Note that for non-periodic systems like molecules, it suffices to consider only $\mathbf{k} = \mathbf{0}$ such that the crystal momentum degrees of freedom disappear.

A Gaussian basis of electron orbitals uses the index $\sigma = (n, \ell, m)$ and takes the form

$$u_{(n,\ell,m)}(\mathbf{x}) = \sum_{\mathbf{r}} \left(\sum_{b=1}^{N_b} c_{nb} e^{-\alpha_{nb}(\mathbf{x}-\mathbf{r})^2} \right) R_{\ell m}(\mathbf{x} - \mathbf{r}) \tag{5.22}$$

where $\{c_{n,b}, \alpha_{n,b}\}$ are real Gaussian primitive coefficients with $\alpha_{n,b} \geq 0$. In the summation above, $\mathbf{r}$ assumes the positions of the images of each atomic site in a lattice (if the system is periodic), and $R_\ell^m(\mathbf{x})$ are the solid harmonic functions. Expanded in spherical coordinates, these functions can be written in terms of the spherical

harmonic functions $Y_{\ell m}(r,\theta,\varphi)$ or the associated Legendre polynomials $P_{\ell,m}(x)$:

$$R_{\ell m}(r,\theta,\varphi) = \sqrt{\frac{4\pi}{2\ell+1}} r^{\ell} Y_{\ell m}(\theta,\varphi) \tag{5.23}$$

$$= \sqrt{\frac{(\ell-m)!}{(\ell+m)!}} r^{\ell} P_{\ell,m}(\cos(\theta)) e^{im\varphi} \tag{5.24}$$

We observe that the solid harmonics are almost entirely real, except for the azimuthal phase $e^{im\varphi}$. To make them entirely real (which makes the numerical representation simpler), we define

$$\bar{R}_{\ell,\pm|m|}(\mathbf{x}) = ((-1)^m R_{\ell,|m|}(\mathbf{x}) \pm R_{\ell,-|m|}(\mathbf{x}))/\sqrt{\pm 2}, \tag{5.25}$$

which instead contain sines and cosines of the azimuth. In Gaussian DFT, usually, the real solid harmonics $\bar{R}_{\ell,m}$ are favored over the standard solid harmonics $R_{\ell,m}$, since they yield real-valued atomic orbitals $\bar{u}_{n,\ell,m}$ which can be expanded as

$$\bar{u}_{(n,\ell,\pm|m|)}(\mathbf{x}) = \sum_{\mathbf{r}} \left(\sum_{b=1}^{N_b} c_{nb} e^{-\alpha_{nb}(\mathbf{x}-\mathbf{r})^2} \right) \bar{R}_{\ell,m}. \tag{5.26}$$

The radial component of the orbital functions $\bar{u}_{(n,\ell,m)}(\mathbf{x})$ uses a power of r times a linear combination of primitive Gaussian functions ($e^{-\alpha r^2}$) to approximate the behavior of the wave function at a large distance from the position of the nucleus $\mathbf{r}$. Remarkably, using a sum of Gaussian functions for the radial component enables the closed form evaluation of overlap integrals between neighboring atoms [105]. Specifically, the integral

$$\begin{aligned} S_{(\mathbf{k},\sigma),(\mathbf{k}',\sigma')} &= \langle \phi_{\mathbf{k},\sigma}(\mathbf{x}-\mathbf{r}) | \phi_{\mathbf{k}',\sigma'}(\mathbf{x}-\mathbf{r}') \rangle \\ &= \int_{\Omega} \phi_{\mathbf{k},\sigma}(\mathbf{x}-\mathbf{r})^* \phi_{\mathbf{k}',\sigma'}(\mathbf{x}-\mathbf{r}')\ d\mathbf{x} \\ &= \delta(\mathbf{k}-\mathbf{k}') \sum_{\mathbf{r},\mathbf{r}'} e^{i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}')} \int_{\Omega} \bar{u}_{\sigma}(\mathbf{x}-\mathbf{r})^* \bar{u}_{\sigma'}(\mathbf{x}-\mathbf{r}')\ d\mathbf{x} \end{aligned} \tag{5.27}$$

admits an analytical solution that is differentiable with respect to the parameters α_{nb} and c_{nb}. We discuss methods for evaluating these integrals and their derivatives in Appendix B.

Although modeling the wavefunction at a large distance from the nucleus as a linear combination of Gaussians is sufficient for approximating these overlap integrals efficiently, it comes at the cost of the loss of on-site orthogonality between orbitals on the same site with different principal quantum numbers n. This, however, is typically not an issue if only the valence shell electrons are included in the calculation and the Coulomb interactions between the core and valence electrons are incorporated to form an on-site effective nuclear potential. This results in a screened ion potential $V_{\text{pseudo}}(r)$, often called a *pseudopotential* in DFT calculations. Because Gaussian orbitals only accurately model valence wave functions far away from the nuclei, these orbitals are also sometimes called *pseudo-orbitals*. As shown in Figure 5.1, the pseudo-orbitals and pseudopotentials accurately approximate the true values beyond a sufficiently large atomic radius, r_{atom}, which facilitates the evaluation of overlaps between neighboring atoms at a low computational cost.

The pseudopotential $V_{\text{pseudo}}(\mathbf{x})$ is approximately radially symmetric, except for some small degree of anisotropy induced by the Coulomb interaction with the outermost shell of the core electrons. This suggests that V_{pseudo} can also be approximated as a linear combination of primitive Gaussians of the form

$$V_{\text{pseudo}}(\mathbf{x}) = \sum_{\mathbf{r}} \left(\sum_{k=1}^{n_k} v_k e^{-\gamma_k (\mathbf{x}-\mathbf{r})^2} \right) \tag{5.28}$$

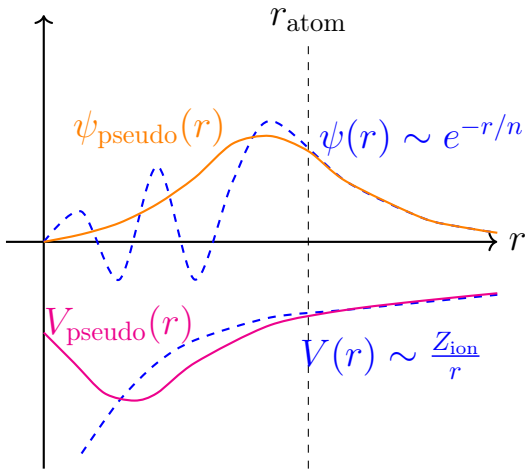

Figure 5.1: An illustration of a Gaussian approximation to a radial wave function $\psi(r)$ and a nuclear potential $V(r)$. Both the pseudo-orbital (orange) and pseudopotential (magenta) closely approximate the real wave function beyond the atomic radius r_{atom}.

for a set of real coefficients $W = \{v_k, \gamma_k\}_{k=1}^{N_k}$ where $\gamma_k > 0$. In practice, using $N_k \geq 5$ Gaussians is sufficient for obtaining accurate approximations to the effective pseudopotential. This expansion of V_{pseudo} in terms of primitive Gaussians also facilitates the efficient computation of the single-particle matrix form h of the non-interacting Hamiltonian $\hat{H}_{\text{NI}}$ for the valence electron shells of all atoms. Assuming that the long-range external potential interactions and kinetic energy interactions at a site $\mathbf{r}_{\text{atom}}$ are only significant between an on-site orbital and a neighboring orbital, the entries of h can be computed in the basis of crystal orbitals $\phi_{\mathbf{k},(n,l,m)}$ in closed form. These coefficients assume the form

$$\begin{aligned} h_{(\mathbf{k},\sigma),(\mathbf{k}',\sigma')} = \sum_{\mathbf{r},\mathbf{r}'} &\langle \phi_{\mathbf{k},\sigma}(\mathbf{x}-\mathbf{r})| \, V_{\text{ext}}(\mathbf{x}) \, |\phi_{\mathbf{k}',\sigma'}(\mathbf{x}-\mathbf{r}')\rangle \\ &+ \langle \phi_{\mathbf{k},\sigma}(\mathbf{x}-\mathbf{r})| \left(-\frac{1}{2}\nabla^2\right) |\phi_{\mathbf{k}',\sigma'}(\mathbf{x}-\mathbf{r}')\rangle - \mu \end{aligned} \tag{5.29}$$

The external potential term expands as

$$\begin{aligned} \langle \phi_{\mathbf{k},\sigma}(\mathbf{x}-\mathbf{r})| \, V_{\text{ext}}(\mathbf{x}) \, |\phi_{\mathbf{k}',\sigma'}(\mathbf{x}-\mathbf{r}')\rangle = \delta(\mathbf{k}-\mathbf{k}') e^{i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}')} \frac{1}{2} \Big[\\ \int_\Omega \bar{u}_\sigma(\mathbf{x}-\mathbf{r})^* \left(V_{\text{pseudo}}^{(\mathbf{r}_{\text{atom}})}(\mathbf{x}-\mathbf{r}) + V_{\text{pseudo}}^{(\mathbf{r}'_{\text{atom}})}(\mathbf{x}-\mathbf{r}') \right) \bar{u}_{\sigma'}(\mathbf{x}-\mathbf{r}') \, d\mathbf{x} \Big], \end{aligned} \tag{5.30}$$

where $V_{\text{pseudo}}^{(\mathbf{r}_{\text{atom}})}$ is the pseudopotential associated with the atomic species centered at $\mathbf{r}$. The kinetic energy term requires more detailed treatment depending on whether the orbitals are treated as effectively spin-paired or with spin-orbit coupling. For the simpler spin-paired case, we expand the kinetic term in the real basis orbitals $\bar{u}_\sigma$, to obtain:

$$\langle \phi_{\mathbf{k},\sigma}(\mathbf{x}-\mathbf{r})| \left(-\frac{1}{2}\nabla^2\right) |\phi_{\mathbf{k}',\sigma'}(\mathbf{x}-\mathbf{r}')\rangle = \left[\frac{1}{2}\mathbf{k}^2 + \mathbf{k}\cdot\mathbf{p}_{\sigma,\sigma'}(\mathbf{r},\mathbf{r}') + \frac{1}{2}\mathbf{p}^2_{\sigma,\sigma'}(\mathbf{r},\mathbf{r}')\right] \tag{5.31}$$

where $\mathbf{p}$ is the matrix of orbital momentum vectors (sometimes called the optical matrix)

$$\mathbf{p}_{\sigma,\sigma'}(\mathbf{r},\mathbf{r}) = -i\,\langle \bar{u}_\sigma(\mathbf{x}-\mathbf{r})|\,\nabla\,|\bar{u}_{\sigma'}(\mathbf{x}-\mathbf{r}')\rangle\,, \tag{5.32}$$

which can be evaluated in closed form for Gaussian orbitals $\bar{u}_\sigma, \bar{u}_{\sigma'}$ using the methods in Appendix B. In some cases, a low energy approximation can be applied that treats the $\mathbf{k}^2$ and $\mathbf{k}\cdot\mathbf{p}$ terms separately from the main calculation, which greatly simplifies the calculation of the kinetic terms. This is an approach known as $\mathbf{k}\cdot\mathbf{p}$ perturbation theory [112].

So far, we have shown that each component of the non-interacting Hamiltonian matrix h_{ij} can be computed in closed form when a basis of Gaussian orbitals and pseudopotentials is used. From this matrix, we can then compute a rough initial estimate of the non-interacting electron occupancies of each eigenstate, which we represent through the electron density matrix ρ_{NI}. This is achieved by diagonalizing $h = U\Lambda_E U^\dagger$, where Λ_E is the diagonal matrix of energy eigenvalues and populating the eigenstates with negative energy, which lie below the chemical potential μ. In

the case that the overlap between adjacent site orbitals is significant such that the orthogonality approximation of $S_{(\mathbf{k},\sigma),(\mathbf{k}',\sigma')} \approx I$ is no longer valid, one can instead solve the generalized eigenvalue problem

$$hU = \Lambda_E S U \tag{5.33}$$

where S is the positive semi-definite orbital overlap matrix. We then compute

$$\rho_{\mathrm{NI}} = U\ f(\Lambda_E)\ U^\dagger, \quad \text{where} \quad f(E) = \begin{cases} 1, & E \leq 0 \\ 0, & E > 0 \end{cases} \tag{5.34}$$

so that all energetically favorable states are occupied. If orbital smearing is desired, other functions $f(E)$ (such as the Fermi-Dirac distribution) can be applied. We note that in the spin-paired case, it is assumed that the spectrum of h is degenerate with respect to the spin degrees of freedom. We note that due to the orthogonality of Gaussian orbitals with different momentum vectors $\mathbf{k}$ (recall $S_{(\mathbf{k},\sigma),(\mathbf{k}',\sigma')} \sim \delta(\mathbf{k}-\mathbf{k}')$), h must be block-diagonal, with each block corresponding to a momentum vector $\mathbf{k}$. This means that the diagonalization procedure can be carried out independently for the components of h corresponding to each momentum value $\mathbf{k}$. If a large discrete mesh of $\mathbf{k}$ points is used to sample the first Brillouin zone (i.e. the set of low-energy electrons), this diagonalization procedure scales as $O(N)$ with respect to the $\mathbf{k}$-mesh size instead of as $O(N^3)$ for diagonalizing the entire matrix h all at once.

The electron density ρ_{NI} obtained from the non-interacting Hamiltonian does not take into account the Hartree functional or the exchange-correlation functional. One can, however, generate an improved estimate of the ground state density ρ based on a local potential representation of the Hartree functional $\hat{\mathcal{J}}[\rho]$ and a local approximation of the exchange-correlation functional $\hat{\mathcal{V}_{\mathrm{xc}}}[\rho]$. First, we observe that the

Hartree functional for a fixed density ρ can be expanded as the potential function

$$V_{\text{Hartree}}(\mathbf{x}', \rho) = \int_{\Omega} \frac{n_\rho(\mathbf{x}')}{\|\mathbf{x} - \mathbf{x}'\|} \, d\mathbf{x}', \tag{5.35}$$

which can be incorporated into the Hamiltonian matrix h directly as part of the external potential. As previously discussed, the exchange-correlation functional $\hat{\mathcal{V}}_{\text{xc}}[\rho]$ does not have a known closed form, but can be approximated through a number of analytical and empirical formulas. One of the simplest approximations of the exchange-correlation functional is given by the *local density approximation* (LDA) functional [8, 110], given by

$$\hat{\mathcal{V}}_{\text{xc}}^{\text{LDA}}[\rho] = \int V_{\text{xc}}(\mathbf{x}, \rho) n_\rho(\mathbf{x}) \, d\mathbf{x} \tag{5.36}$$

where $V_{\text{xc}}(\mathbf{x}, \rho)$ is typically chosen as the exchange-correlation energy of a homogeneous electron gas with density ρ. This has been shown to have the closed form

$$V_{\text{xc}}(\mathbf{x}, \rho) = -\frac{3}{4} \left(\frac{3 n_\rho(\mathbf{x})}{\pi} \right)^{1/3}. \tag{5.37}$$

Both of the Hartree and exchange-correlation potential terms V_{Hartree} and V_{xc} evaluated from the non-interacting density ρ_{NI} can be incorporated into the external potential V_{ext} to create an effective mean-field Hamiltonian matrix h^{eff}. Its entries are computed as follows:

$$\begin{aligned} h^{\text{eff}}_{(\mathbf{k},\sigma),(\mathbf{k}',\sigma')} = \sum_{\mathbf{r},\mathbf{r}'} & \langle \phi_{\mathbf{k},\sigma}(\mathbf{x} - \mathbf{r}) | \left(V_{\text{ext}}(\mathbf{x}) + V_{\text{Hartree}}(\mathbf{x}, \rho_{\text{NI}}) + V_{\text{xc}}(\mathbf{x}, \rho_{\text{NI}}) \right) | \phi_{\mathbf{k}',\sigma'}(\mathbf{x} - \mathbf{r}') \rangle \\ & + \langle \phi_{\mathbf{k},\sigma}(\mathbf{x} - \mathbf{r}) | \left(-\frac{1}{2} \nabla^2 \right) | \phi_{\mathbf{k}',\sigma'}(\mathbf{x} - \mathbf{r}') \rangle - \mu \end{aligned} \tag{5.38}$$

This Hamiltonian can then be diagonalized to obtain a better mean-field approximation of the density, ρ_{eff}, which approximates the effects of the background Coulomb interactions and exchange-correlation effects.

5.2.2.4 The Gaussian self-consistent field algorithm. In the previous section, we showed how the non-interacting Hamiltonian h was constructed and diagonalized ρ_{NI}. We then showed that a more accurate mean-field Hamiltonian could be obtained using the background Coulomb potential and exchange-correlation effects of the non-interacting density ρ_{NI} to construct an effective Hamiltonian h^{eff}, which could be diagonalized to obtain a more accurate ground state density ρ^{eff}. Naturally, this procedure can be extended to further refine the mean-field approximation of ρ and obtain a set of ground state electron occupations that are self-consistent with respect to h^{eff}. Starting with $\rho^{(0)} = \rho_{\text{NI}}$ as the initial ground state density, one can successively recompute the V_{Hartree} and V_{xc} potentials using $\rho^{(n-1)}$ and then construct the effective Hamiltonian $h^{(n)}$, which is diagonalized to obtain $\rho^{(n)}$. This process is known as the *self-consistent field* approach, and eventually converges to a density $\rho^{(N)}$ when a criterion $\left\|\rho^{(N-1)} - \rho^{(N)}\right\| \leq \varepsilon_\rho$ is met for a desired error tolerance ε:

$$h \to \rho_{\text{NI}} = \rho^{(0)} \to h^{(1)} \to \rho^{(1)} \to h^{(2)} \to \rho^{(2)} \to ... \to h^{(N)} \to \rho^{(N)} = \rho_{\text{self-consistent}} \tag{5.39}$$

Equivalently, one can also stop the self-consistent field loop when the total energy $E^{(n)} = \text{Tr}\big(\rho^{(n)} h^{(n)}\big)$ converges with $|E^{(n)} - E^{(n-1)}| \leq \varepsilon_E$ for some energy tolerance ε_E. To stabilize convergence, a technique called *density mixing* is usually employed, where newly calculated electron densities $\rho\{h^{(n+1)}\}$ are mixed with the previous electron

density according to the formula

$$\rho^{(n+1)} = \alpha\rho\{h^{(n+1}\} + (1-\alpha)\rho^{(n)} \tag{5.40}$$

for some mixing parameter $\alpha \in [0, 1]$.

In the case of variational Gaussian DFT where a minimal local basis of orthogonal pseudo-orbitals is allowed to vary with free parameters $c_{n,b}, \alpha_{n,b}$, one must carefully optimize these parameters while maintaining approximate self-consistency until convergence is reached. Optimization of these parameters can be achieved through analytical gradients computable through the methods discussed in Appendix B.1. To ensure that the system density matrix ρ remains approximately self-consistent while the basis parameters $C = \{\alpha_{n,b}, c_{n,b}\}$ are optimized, the gradient step sizes with respect to these parameters can be fixed to a small value η, ensuring that no parameters are varied too quickly:

$$C^{(n+1)} = C^{(n)} - \eta\frac{\nabla_C E^{(n)}}{\|\nabla_C E^{(n)}\|} \tag{5.41}$$

In Figure 5.2, we summarize the steps involved in the SCF calculation loop, which minimizes the ground state energy and optimizes self-consistency simultaneously.

In summary, we have outlined the foundational formalism of quantum many-body systems and the practical approximations used to simulate them, particularly through Density Functional Theory. By introducing the Schrodinger equation, the variational principle, and the construction of single- and two-body Hamiltonians, we've framed the motivation behind replacing the full many-body wavefunction with

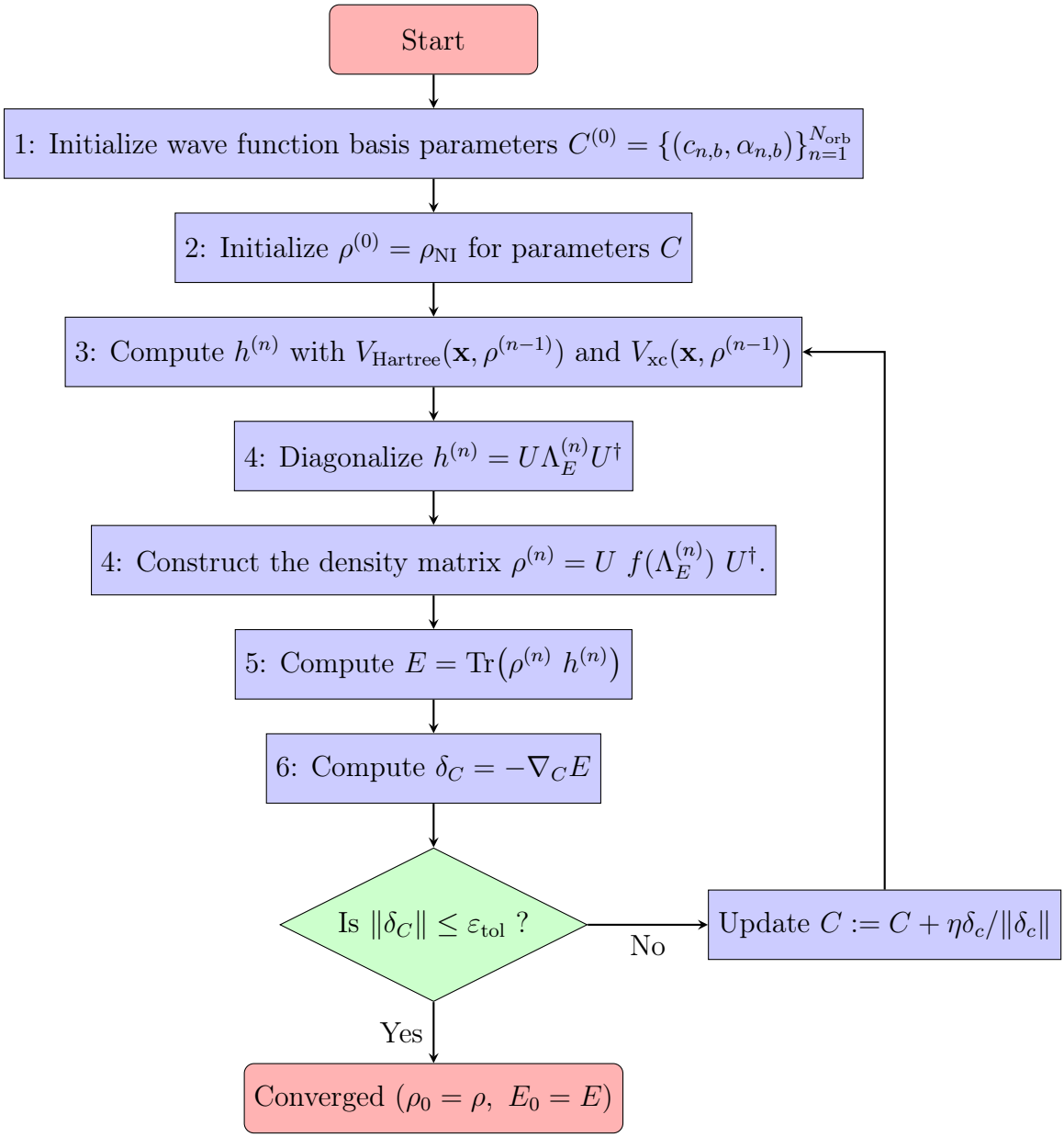

Figure 5.2: Flowchart summary of a Gaussian DFT self-consistent field calculation, where self-consistency and gradient-based optimization of the basis are performed simultaneously in the same loop. Depending on the stability or scale of the problem, the parameter η can be increased to stabilize convergence at the cost of a longer time to convergence.

a self-consistent electron density. We have also further established how exchange-correlation functionals and Hartree potentials are incorporated into a mean-field approximation, which is iteratively refined through the self-consistent field (SCF) algorithm. These methods, though computationally demanding, form the backbone

of ab initio electronic structure theory. In contrast to these methods, QuINNs approximate this self-consistent process through learnable, differentiable surrogates that retain quantum mechanical structure while enabling orders-of-magnitude speedups. The next section formalizes how QuINNs embed these constraints through symmetry-preserving neural architectures.

5.2.3 E(3)-Equivariant Neural Networks

In addition to incorporating the principles of PINNs, the actual model architecture of QuINNs are a specialized class of neural network model called a *graph neural network* (GNN) [65]. GNNs have become a powerful tool for modeling relational structures in data, particularly in the context of molecular and materials science. Standard GNNs operate on data structures called *graphs*, where nodes represent entities (e.g., atoms) and edges encode relationships between entities (e.g., bonds). While GNNs are effective at capturing localized properties of atomic systems [34, 113], they often cannot discriminate between the full set of three-dimensional symmetries present in a system, such as rotations and reflections. This limitation makes it difficult for traditional GNNs to distinguish between atomic systems that exhibit complex chiral asymmetries, which differ only by reflection but have distinct physical and chemical properties [66]. Unlike conventional GNNs, QuINNs leverage symmetries and domain-specific constraints derived from quantum mechanics to enhance their predictive capabilities for material and molecular systems. These networks integrate equivariant architectures, such as $E(3)$-equivariant neural networks, to ensure that learned representations respect the fundamental symmetries of the underlying physical system.

E(3)-equivariant neural networks ($E(3)$NNs) are an emerging neural network architecture for handling data that is embedded as a graph (i.e., a network of nodes connected by edges) in three-dimensional space [68]. Atomic systems like molecules and crystals can be naturally interpreted as graphs, where the nodes are atomic nuclei, and the edges consist of bonds between them. In addition to the existence of bonds, the three-dimensional geometry of the atomic configuration plays a significant role in the electronic properties present in a material or molecule. Indeed, many of the electronic properties of solids can be quite sensitive to atomic geometry, with some properties even emerging only in the presence of lattice strain or the introduction of defects into the material. Nonetheless, discovering how the lattice geometry can be directly incorporated into machine learning models has proven to be somewhat difficult, especially for the electronic properties of molecules and solid-state materials. This is due to the fact that obtaining a "fingerprint" of an atomic system to serve as input to a traditional GNN is not generally possible without discarding crucial information regarding the exact atomic geometry and any symmetries present in the system.

E(3)-equivariant neural networks, an emerging type of symmetry-aware graph neural network, address this issue by explicitly encoding Euclidean symmetries (i.e., the symmetries of 3D space) into their architecture, ensuring that learned representations and predicted features transform predictably under these transformations. The term "equivariant" refers to the mathematical definition of equivariant functions from group theory:

> A function $f : X \to Y$ is said to be *equivariant* under a group G if applying a group transformation $g \in G$ to the input transforms the

output in a predictable way:

$$f(T_g\mathbf{x}) = T'_g f(x) \quad \text{for all } g \in G,\ x \in X \tag{5.42}$$

where T_g and T'_g are representations of G on X and Y respectively.

In the context of $E(3)$NNs, the function f is the neural network, and X and Y consist of graphs where each node is embedded as a point in 3D space and can have additional associated scalar, vector, and tensor features with prescribed symmetries. Here, the group G is the Euclidean group in three dimensions ($E(3)$), which consists of all translations rotations, and reflections of a graph embedded in 3D space. Since graph neural networks can easily be made equivariant with respect to the translation of the graph and its features (e.g. by only incorporating the displacements between nodes), we will only consider the remaining symmetries of rotation and reflection. Designing neural networks that are equivariant under these two kinds of symmetries is somewhat non-trivial.

5.2.3.1 Spherical harmonics and tensor decomposition. To construct $E(3)$-equivariant neural network layers, the features of a neural network are decomposed into irreducible representations (often called *irreps*) of the rotation group $SO(3)$ using spherical harmonics. The group $SO(3)$ (called the *special orthogonal group in three dimensions*) consists of all rotations that map irreducible representations to themselves in a manner that is unitarily reversible and has determinant 1 (i.e. the "right-hand rule" for axis order is preserved). The irreps of $SO(3)$ provide the fundamental building blocks for group representations of rotations in 3D space, allowing complex transformations to be understood in terms of simpler, non-decomposable components.

For the special orthogonal group $SO(3)$, irreps are indexed by an integer ℓ, which corresponds to the angular momentum quantum number in physics, used for spherical harmonic representations of atomic orbitals. These irreps have dimensions $2\ell + 1$, representing the number of independent components that transform together under rotations. These components are indexed by the integer value m, which corresponds to the magnetic quantum number satisfying $-\ell \leq m \leq \ell$. The group $SO(3)$, which includes only rotations, admits two types of irreps:

- *Even Irreps ($\ell = 0, 2, 4, ...$):* These are symmetric under reflections and describe quantities such as scalar fields and symmetric tensors.

- *Odd Irreps ($\ell = 1, 3, 5, ...$):* These change sign under reflections and are associated with vector fields and antisymmetric tensors.

These are often referred to as the *true tensor* irreps of the group $SO(3)$. They describe how standard vector quantities (more generally, true tensor quantities) transform under rotations with mutually exclusive even or odd parity. Numerically, we represent each irrep as a feature vector f, which encodes important physical quantities such as scalars ($\ell = 0$), vectors ($\ell = 1$) and symmetric rank-2 tensors ($\ell = 2$). More concretely, each feature vector f can be represented as a function expanded in terms of spherical harmonics $Y_{\ell m}$:

$$f(\mathbf{r}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} f_{\ell m} Y_{\ell m}(\mathbf{r}). \tag{5.43}$$

where the components $f_{\ell m}$ transform under $SO(3)$ according to the Wigner-D matrices $D^{(\ell)}(R)$ [114]:

$$f'_{\ell m} = \sum_{m'=-\ell}^{\ell} D^{(\ell)}_{mm'}(R) f_{\ell m'}. \tag{5.44}$$

Each Wigner-D matrix corresponds to a 3×3 rotation matrix $R \in SO(3)$, and describes how the spherical harmonics for each angular quantum number ℓ transform together under these operations. This structure ensures that the learned features transform correctly under rotations, providing a principled approach for representing the features of 3D data and for constructing equivariant neural network layers.

In some cases, one may desire to develop a neural network model whose predictions are either strictly equivariant or strictly invariant under inversion, depending on how the physical quantities corresponding to the inputs and outputs transform under symmetry. For example, if a model is being developed to classify the chirality of molecules (i.e. either $+1$ or -1 based on the moleculer geometry), one would need the output to be a scalar value y; however, true scalar quantities correspond to $\ell = 0$ and are naturally even irreps of $SO(3)$ and thus are invariant under inversion, transforming as $y \mapsto y$. In contrast, the chirality of a molecule transforms as $y \mapsto -y$ under inversion symmetry. In this case, we would say that y behaves as a *pseudoscalar*, having even symmetry despite the fact that its dimension ($d = 1$) corresponds to an even ($\ell = 0$) irrep of $SO(3)$. It is clear that for systems where the chirality and inversion symmetries are to be respected by a neural network, a stronger form of equivariance that respects this inversion symmetry must be imposed. In particular, we must consider the irreps of the more general *orthogonal group* $O(3)$, which consist of rotation matrices with determinants ± 1 (not just 1, as is the case with $SO(3)$).

The irreps of $O(3)$ resemble those of $SO(3)$ and admit expansions with respect to the spherical harmonics $Y_{\ell m}$ with an additional parity term p, depending on whether or not an additional -1 sign is induced by inversion. Like the $SO(3)$ irreps, we expand

$$f(\mathbf{r}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \sum_{p=\pm 1} f_{\ell m p} Y_{\ell m}(\mathbf{r}), \tag{5.45}$$

however, the more general $O(3)$ irreps transform under a 3×3 rotation and reflection matrix $R \in O(3)$ as

$$f'_{\ell m} = \sum_{m'=-\ell}^{\ell} \tau_p^{(\ell)}(\det(R)) D^{(\ell)}_{mm'}(R/\det(R)) f_{\ell m'}. \tag{5.46}$$

where

$$\tau_p^{(\ell)}(s) = \begin{cases} |s| & s \geq 0 \\ p(-1)^\ell |s|, & s < 0 \end{cases} \tag{5.47}$$

In Table 5.1 we summarize the $O(3)$ irreps and give physical examples of each kind of irrep up to $\ell = 2$.

Table 5.1: Table of general irreducible representations f_{lmp} of the orthogonal group $O(3)$, which includes both arbitrary 3D rotations and reflections. Depending on the physical quantities represented by the inputs, outputs, and latent features of the model, different irreps must be chosen to ensure rotational equivariance under either $O(3)$ or $SO(3)$.

Degree	Dimension	Even Parity Irreps ($p = +1$)	Odd Parity Irreps ($p = -1$)
$\ell = 0$	$d = 1$	scalar (*e.g., charge density*)	pseudoscalar (*e.g. chirality/handedness*)
$\ell = 1$	$d = 3$	pseudovector (*e.g., angular momentum*)	vector (*e.g., displacement*)
$\ell = 2$	$d = 5$	symmetric rank-2 tensor (*e.g., intertia tensor*)	odd quadrupole moment tensor (*e.g., magnetic quadrupole moment*)
$\vdots$	$\vdots$	$\vdots$	$\vdots$
ℓ	$d = 2\ell + 1$	($f_{lmp} \mapsto f_{lmp}$ under inversion)	($f_{lmp} \mapsto -f_{lmp}$ under inversion)

We note that irreps can also represented in the basis of real spherical harmonic functions $\bar{Y}_{\ell m}(\mathbf{x})$, given by

$$\bar{Y}_{\ell \pm |m|}(\mathbf{x}) = ((-1)^m Y_{\ell,|m|}(\mathbf{x}_)) \pm Y_{\ell,-|m|}(\mathbf{x})/\sqrt{p} \tag{5.48}$$

such that the corresponding Clebsch-Gordon coefficients $\bar{C}^{\ell m}_{\ell_1 m_1 \ell_2 m_2}$ are entirely real-valued. In computational settings, these representations are usually preferred over the standard spherical harmonic irrep due to the fact that the representations are entirely real.

5.2.3.2 Equivariant convolutions. Now that we have considered how to represent Euclidean features as irreps, we must consider how convolution operations are performed. In E(3)-equivariant neural networks, the convolutional operator must be carefully constructed to respect the symmetries of three-dimensional space. Unlike conventional CNNs, where convolutional filters operate over regularly spaced grids (e.g., images), convolution in three-dimensional physical systems requires accounting for continuous rotational and reflectional symmetry. In an abstract sense, $E(3)$-equivariant convolutions extend traditional convolutions by integrating over the continuous symmetry group. Given input features $f : \mathbb{R}^3 \to \mathbb{R}^c$, an equivariant convolution is defined as:

$$(f * g)(\mathbf{r}) = \int_{\mathbb{R}^3} f(\mathbf{r}')g(\mathbf{r} - \mathbf{r}')d\mathbf{r}', \tag{5.49}$$

where $f(\mathbf{r})$ represents the input feature irrep (expanded in terms of spherical harmonic functions) and $g(\mathbf{r} - \mathbf{r}')$ is the convolution kernel, parameterized to ensure equivariance. Unlike standard convolutional kernels in traditional deep learning, which consist of freely trainable parameters, the kernel $g(\mathbf{r}-\mathbf{r}')$ in an $E(3)$-equivariant network must be carefully structured to transform in accordance with irreducible representations of

$SO(3)$ and $O(3)$. Specifically, the kernel is expanded in a basis of spherical harmonics:

$$g(\mathbf{r}-\mathbf{r}') = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} g_{\ell m}(\|\mathbf{r}-\mathbf{r}'\|) Y_{\ell m}((\mathbf{r}-\mathbf{r}')/\|\mathbf{r}-\mathbf{r}'\|) \tag{5.50}$$

Above, the functions $g_{\ell m}(\|\mathbf{r}-\mathbf{r}'\|)$ encode the radial dependence of the kernel and contain the trainable weights of the network. These radial functions are typically modeled using learnable multilayer perceptrons (MLPs) or radial basis functions that are optimized during training [115].

When computing the convolution, the expansion coefficients of the input f and the kernel g interact through the Clebsch-Gordan decomposition:

$$(f * g)_{lm} = \sum_{l_1, m_1, l_2, m_2} C^{lm}_{l_1 m_1 l_2 m_2} f_{l_1 m_1} g_{l_2 m_2}. \tag{5.51}$$

Here, $C^{lm}_{l_1 m_1 l_2 m_2}$ are the Clebsch-Gordan coefficients (also known as Wigner $3j$ symbols), which enforce the coupling between different irreps in a way that respects equivariance [116]. The trainable parameters of the network are contained within the radial parts of the kernel $g_{\ell m}(\|\mathbf{r}-\mathbf{r}'\|)$, meaning that the network learns how different orders of angular components should be weighted when computing the convolution. When the irreps of f and g contain even and odd $O(3)$ irreps with parity p_1 and p_2 respectively, the same Clebsch-Gordon coefficients are used with the implied parity connection rule. Under this rule, the Clebsch-Gordon coefficients are assumed to be zero except when the rule

$$p = p_1 p_2 \tag{5.52}$$

is satisfied. In Figure 5.3, we show a diagram of the irrep connection paths for a convolution of $SO(3)$ irreps ($f^{(\text{in})}$, $g^{(\text{in})}$) to obtain an $O(3)$ irrep $f^{(\text{out})}$. In this diagram, irreps whose outgoing lines are joined together at a point signify a nonzero Clebsch-Gordon coefficient where the parity connection rule above is satisfied.

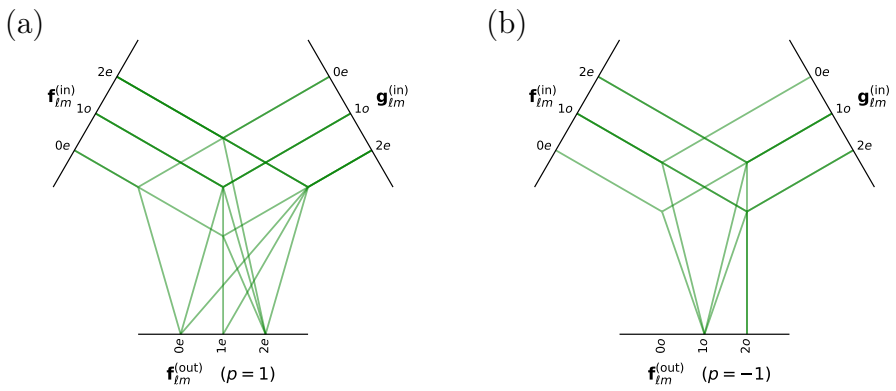

Figure 5.3: Visualization of the connection paths for the convolution of $SO(3)$ irreps $f^{(\mathrm{in})}$ with the irreps of an $SO(3)$ convolution kernel $g^{(\mathrm{in})}$ to obtain $O(3)$ irreps $f^{(\mathrm{out})}$. Both the even (a) and odd (b) components the irreps in $f^{(\mathrm{out})}$ are shown. In this example, the inputs and output irreps are truncated at $\ell_{\max} = 2$ and are denoted using the notation "ℓp", where ℓ is the angular number and p is either "e" (even, $p = 1$) or "o" (odd, $p = -1$).

By leveraging convolutions of this form, $E(3)$-NNs can achieve strong generalization in tasks where inversion, rotational, and translational symmetries are essential, such as molecular property prediction and force field modeling. Here, we aim to leverage $E(3)$-equivariance as a way of incorporating the symmetries of solutions to the Schrodinger equation. This will enable QuINNs to make more accurate and physically-informed predictions of electronic structure that are consistent with the predictions made by more expensive DFT methods.

In summary, we have introduced the architectural foundations of QuINNs as a class of $E(3)$-equivariant graph neural networks tailored for quantum systems. By leveraging irreducible representations of the orthogonal group $O(3)$, QuINNs encode atomic and orbital features as tensors that transform appropriately under rotations, reflections, and translations. Their convolutional layers respect these symmetries via spherical harmonic decompositions and Clebsch–Gordan coupling rules, ensuring that

relevant physical symmetries are preserved in both intermediate and output representations. This design allows QuINNs to model complex atomic interactions using physically structured embeddings that generalize across orientations and crystal symmetries. Importantly, the output of these layers is used to parameterize pseudopotentials and orbital basis functions, all of which are key ingredients in the downstream construction of Hamiltonians and band structures. In the next section, we show how these learned representations can be used to perform fast, interpretable, and differentiable electronic structure prediction.

5.3 Quantum-Informed Neural Networks

5.3.1 Features and Capabilities

Quantum-informed neural networks (QuINNs) represent a new approach to electronic structure prediction that integrates principles from physics-informed neural networks (PINNs) and equivariant neural networks (E(3)NNs). These models are designed to solve the Schrödinger equation for the ground-state electronic structure of solid-state materials and molecules by expanding electronic states in localized Gaussian bases. Unlike conventional ab initio methods, which rely on iterative self-consistent procedures, QuINNs directly predict electronic structure solutions without the need for gradient-based optimization or extensive training data.

A key feature of QuINNs is their ability to generate interpretable predictions within the framework of electronic band theory, providing a direct link between machine learning-based models and traditional quantum mechanical descriptions of materials. This interpretability, combined with their end-to-end differentiability, enables

efficient computation of gradients of electronic properties with respect to atomic geometry and other tunable parameters. As a result, QuINNs facilitate computational optimization of material properties, such as band gaps or charge transport characteristics, with respect to structural modifications, external fields, or defect engineering. By leveraging a physics-informed machine learning approach, QuINNs offer a scalable alternative to traditional electronic structure methods while preserving physical interpretability. This approach has the potential to significantly accelerate materials discovery and device optimization by reducing the need for exhaustive parameter searches in computational workflows. In the particular case of materials engineering, we have identified two key workflows for which QuINNs can be leverged to improve *ab initio* material design workflows:

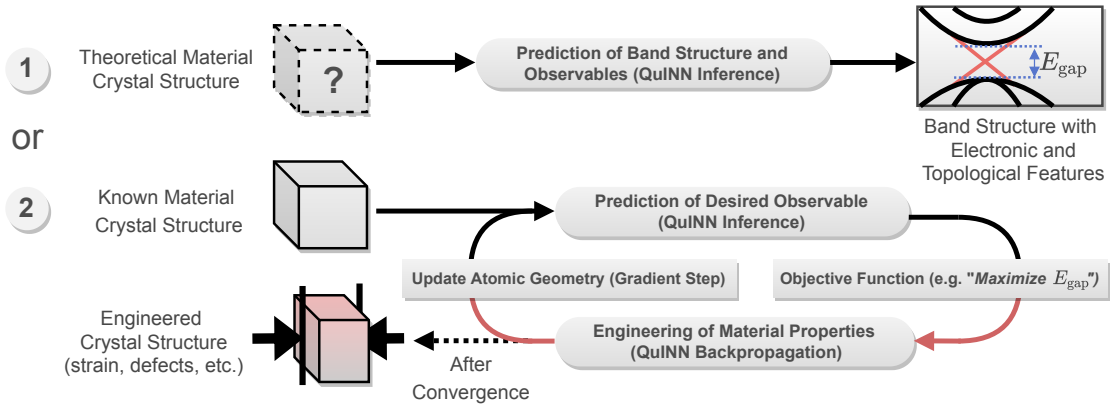

Figure 5.4: An illustration of two ways in which QuINNs can be applied to accelerate computational material science workflows, either (1) through rapid prediction of electronic structure, or (2) engineering desired electronic features or material properties with respect to controllable material parameters, such as the introduction of lattice strain or defects.

Figure 5.4 presents two complementary workflows within the QuINN framework for materials design, involving forward inference and inverse design, respectively.

In the first workflow, given a known or theoretically proposed crystal structure, a pre-trained QuINN model can be directly applied to predict its electronic band structure and other relevant observables. This inference capability enables rapid screening of materials by estimating their properties without requiring computationally intensive ab initio calculations. The second workflow enables inverse material design, where a desired electronic property is specified as an objective function, and a pre-trained QuINN model employs gradient-based backpropagation techniques to iteratively adjust atomic geometries—such as through strain or defect engineering—until convergence is reached. This approach facilitates the rational design of materials by optimizing structural parameters to achieve target functionalities, bridging the gap between computational prediction and experimental realization.

In addition to engineering bulk materials, a key benefit of QuINNs is their computational efficiency, enabling the simulation and optimization of electronic properties of large-scale devices, such as carrier transport, optical properties, etc.

5.3.2 High-Level Summary

As previously discussed, QuINNs integrate physics-based constraints with machine learning to efficiently solve the Schrödinger equation for self-consistent electronic structure prediction. In Figure 5.5, we present a high-level overview of the functionality of QuINN models:

In Figure 5.5, we show how QuINNs utilize E(3)-equivariant graph convolutional layers to encode crystal structure symmetries, ensuring rotational and translational invariance in the intermediate learned feature vectors f_{lmp}, which are expanded

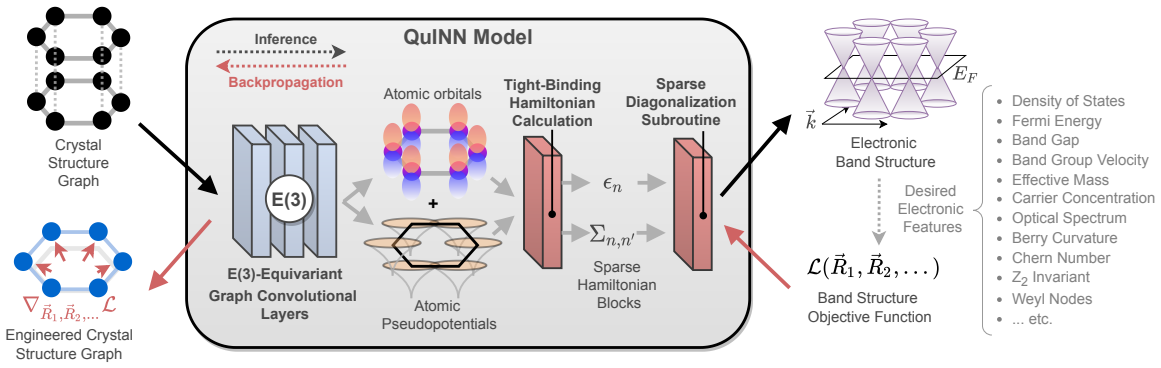

Figure 5.5: An overview of the QuINN model architecture. QuINNs allow for both the direct prediction of material properties and the optimization of atomic geometry with respect to arbitrary functions of electronic structure.

in terms of $O(3)$ irreducible representation (irreps) using the basis of spherical harmonics. These layers transform information from each atom's atomic coordination neighborhood into a feature space that captures the fundamental symmetries of the underlying quantum system. Additionally, atomic pseudopotentials are incorporated into the model, enabling an accurate representation of local electronic interactions while maintaining computational efficiency.

At the core of QuINNs is the construction of a mean-field tight-binding Hamiltonian h, which is assembled from localized Gaussian atomic orbitals and their interactions via the kinetic and potential terms. These Gaussian orbitals are parameterized by the set of basis parameters $C = \{\alpha_{nb}, c_{nb}\}$ for a fixed basis size N_b. While the basis width parameters α_{nb} are fixed to be identical and positive across each shell of parameters, the coefficients c_{nb} corresponding to the ground state electronic configuration are predicted by equivariant neural network layers while preserving all translation, rotation, and reflection symmetries.

Furthermore, all of the contributions to the potential associated with each atomic site are combined into an effective on-site pseudopotential function V_{pseudo}.

This pseudopotential incorporates both the screened ion potential for the valence shells in addition to the predicted Hartree (V_{Hartree}) and exchange-correlation (V_{xc}) contributions due to valence shell occupations in the ground state. By default, these potentials are assumed to be isotropic and parameterized by a set of parameters $W = \{v_k, \gamma_k\}$, with a fixed basis size of N_k. However, this representation can be extended to allow for anisotropic parameterization of the potential in terms of the spherical harmonics, just like the orbitals. This allows the model greater flexibility to account for induced anisotropies in the atomic potential such as static dipole and quadrupole moments.

Unlike traditional tight-binding methods that rely on empirical fitting and non-interacting models of the electronic state, QuINNs learn an optimal Hamiltonian representation through a combination of data-driven optimization and physical constraints. The Hamiltonian is structured in a block-sparse format, facilitating efficient numerical diagonalization through specialized subroutines. This step ensures that only relevant electronic interactions contribute to the final eigenvalue problem, significantly reducing computational overhead while preserving accuracy and physical-consistency in band structure calculations.

Once the Hamiltonian is diagonalized for each $\mathbf{k}$ vector to obtain the electronic band structure, any desired electronic property can be computed in closed form, such as density of states, Fermi energy, and various topological invariants (such as Berry curvature, Chern numbers, etc. [117]). These outputs provide interpretable insights into the material's electronic behavior and are compatible with standard band theory frameworks. The differentiable nature of QuINNs allows for backpropagation through the electronic structure pipeline, making it possible to compute gradients of properties

such as the electronic band gap with respect to atomic positions and other structural parameters. This capability is particularly useful for guiding materials design and optimization.

In contrast with established Gaussian DFT methods, QuINNs leverage equivariant layers to make direct predictions of the local pseudopotentials and orbital basis for each atomic site. These pseudopotentials accurately approximate the net effect of Hartree and exchange-correlation terms derived from the predicted electron density. As a result, the self-consistent loop is no longer needed, allowing for the direct prediction of the electronic ground state with only a single diagonalization operation. This affords a significant computational speedup, as shown in Figure 5.6. However, this may come at the cost of the loss of some physical accuracy, since the Hartree and exchange-correlation potentials are not rigidly required to be self-consistent with the predicted electron density. Instead, the self-consistency is incorporated directly into the loss function used to train the model. This trade-off between model flexibility and physical consistency can be fine-tuned by adjusting the weight in the corresponding loss function term, which is a standard practice for PINN models.

5.3.3 Equivariant Parameterization of Atomic Orbitals and Pseudopotentials

In this section, we will discuss in more technical detail how the $E(3)$ equivariant layers in QuINNs work and how they naturally admit a parameterization of atomic orbitals and pseudopotentials. We recall that QuINNs represent periodic atomic structures as graphs embedded in 3D space. This graph consists of collection of atomic positions $\mathbf{r}_i$ and atomic "edges" $(\mathbf{r}_i, \mathbf{r}_j)$ that connect interacting centers within some cutoff radius $r_{\max}$. Associated with each atomic center is the atomic species,

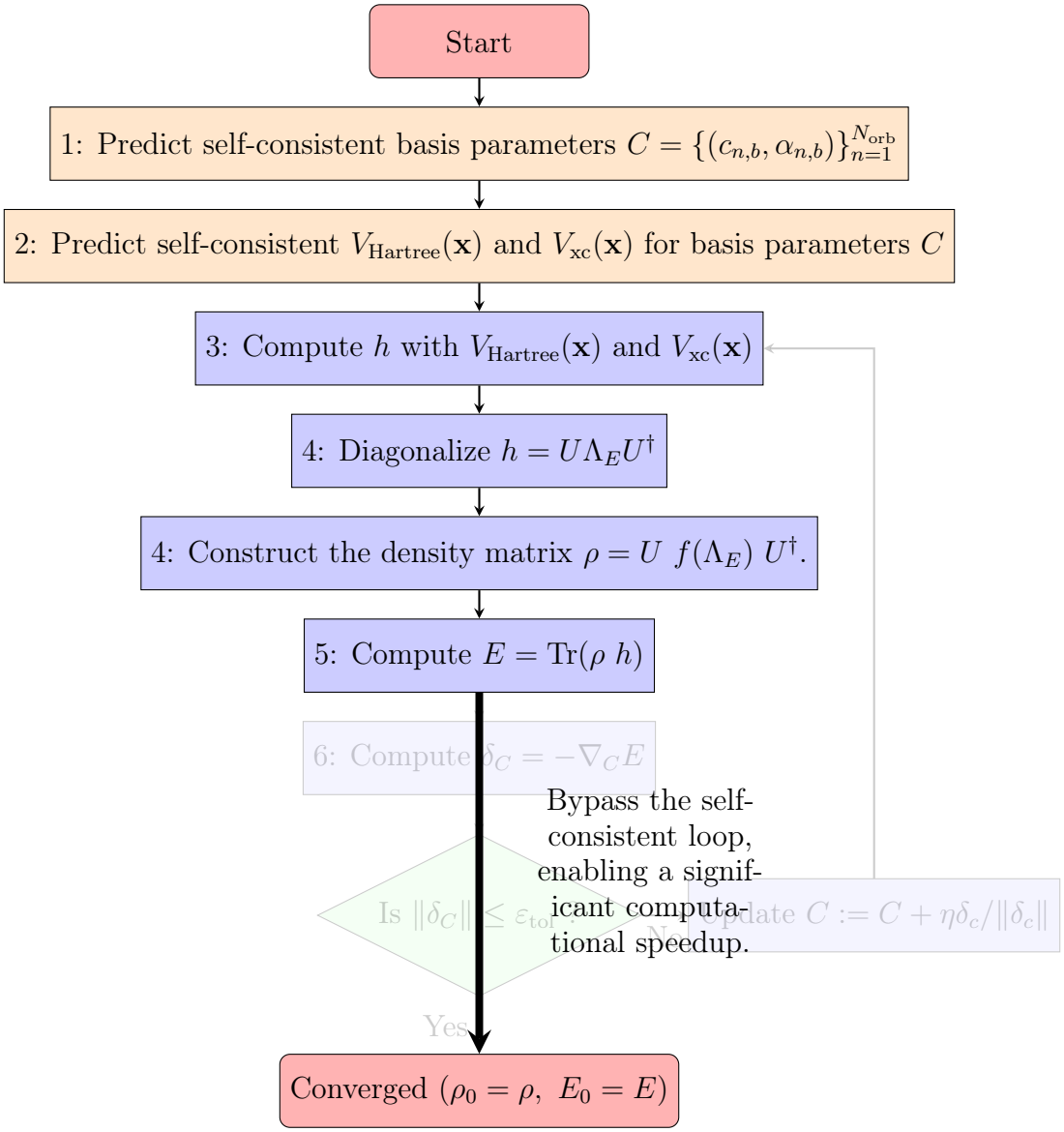

Figure 5.6: High-level flowchart summary of how QuINNs make direct predictions of ground state electronic structure, effectively bypassing the computational cost of the self-consistent loops required by Gaussian DFT methods. This requires the introduction of two equivariant neural network models (represented above by the orange boxes). These consist of (1) a model to predict the optimal self-consistent basis parameters C, and (2) a model to predict the combined effect of the ground state $V_{\text{Hartree}}(\mathbf{x})$ and $V_{\text{xc}}(\mathbf{x})$ with respect to the basis parameters C. For computational efficiency, these two models can be fused together into a single model that outputs the orbital parameters C and pseudopotential parameters W simultaneously, however we show them here as separate models for conceptual clarity, illustrating the dependence of W on C.

indicated by a one-hot encoding of its atomic number, Z_i, in addition to a vector of (optional) scalar features $\mathbf{u}_i$ associated with each site. These features are combined into a single $\ell = 0, m = 0$ scalar irrep feature vector $\mathbf{f}_{00}^{(0)}$, which serves as the input to the model. A sequence of L E(3) Convolutions are then performed, producing the intermediate irrep feature vectors $\mathbf{f}_{\ell mp,i}^{(n)}$ for each atomic site i in the atomic graph, as shown in Figure 5.7(a). These convolutions are applied to each neighboring site j in the neighborhood of site i, denoted $N(i)$, where the distance vectors $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ between neighboring sites are incorporated, and the resulting convolution irreps are summed together. This process can be summarized in three steps: First, a multilayer perceptron (MLP) is applied to a radial feature embedding of the distance between neighboring atoms, $\|\mathbf{r}_{ij}\|$, to obtain an edge embedding vector $\mathbf{s}_{ij}$ The radial feature embedding can be computed, for example, by applying an embedding function $S_n(r)$, which can be a soft one-hot encoding of the distance, or a family of orthogonal functions, such as the spherical Bessel functions $j_n(r)$ (in either case, $S_n(r)$ must be chosen such that $\lim_{r\to\infty} S_n(r) = 0$. Once the radial embedding is computed, the MLP layer is applied with weight matrix $w_{n,n'}$ and a nonlinear activation function $\sigma(x)$ to obtain the edge embedding vector $\mathbf{s}_{ij}$:

$$\mathbf{s}_{ij} = \sigma\left(\mathrm{MLP}(r)_n = \sum_{n'} w_{n,n'} S_{n'}(r)\right) \tag{5.53}$$

Second, the irreps of the convolution kernel g is computed by evaluating the spherical harmonic functions $Y_{\ell m}(\mathbf{x})$ for the unit distance vector $\mathbf{r}_{ij}/\|\mathbf{r}_{ij}\|$. This produces an irrep feature vector $\mathbf{y}_{\ell m,ij}$, which can be combined with the radial feature

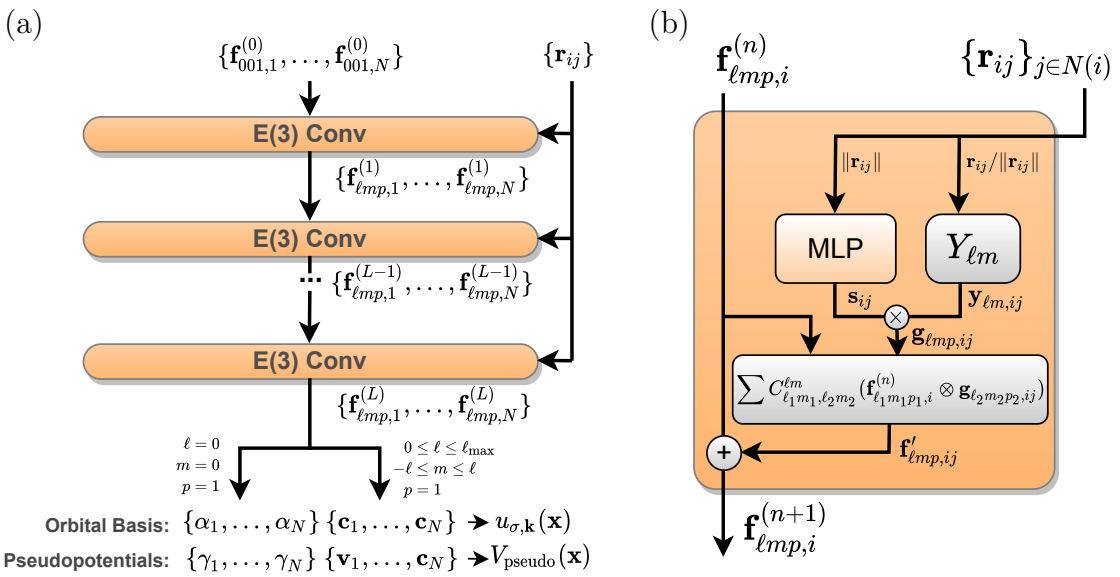

Figure 5.7: Summary of the $E(3)$-equivariant layers used to parameterize the basis of orbitals and pseudopotentials for each atomic site. (a): Overview of stacked $E(3)$ layers and intermediate irreps $\mathbf{f}^{(n)}_{\ell mp}$ passed between each layer. (b): Circuit diagram of an $E(3)$ equivariant layer containing a trainable multilayer perceptron (MLP).

vector $\mathbf{s}_{ij}$ to obtain the kernel irrep vectors $\mathbf{g}_{\ell mp,ij}$, where the parities are taken as $p = (-1)^{\ell}$:

$$\mathbf{g}_{\ell mp,ij} = (\mathbf{s}_{ij})(\mathbf{y}_{\ell m,ij}) = \mathrm{MLP}(\|\mathbf{r}_{ij}\|) Y_{\ell m}\left(\frac{\mathbf{r}_j - \mathbf{r}_i}{\|\mathbf{r}_j - \mathbf{r}_i\|}\right) \tag{5.54}$$

Third, the convolved irrep feature vectors $\mathbf{f}'_{\ell mp,ij}$ are computed via the Clebsch-Gondon coefficients $C^{\ell m}_{\ell_1,m_1,\ell_2,m_2}$ and the parity connection rule $p = p_1 p_2$. The resulting irreps are then added together with $\mathbf{f}^{(n)}_{\ell mp,i}$ to obtain $\mathbf{f}^{(n+1)}_{\ell mp,i}$:

$$\mathbf{f}^{(n+1)}_{\ell mp,i} = \sum_{j \in N(i)} \mathbf{f}'_{\ell mp,ij} = \sum_{j \in N(i)} (\mathbf{f}^{(n)}_i * \mathbf{g}_{ij}) \tag{5.55}$$

These three steps comprise the E(3)-convolution layer in a basic QuINN model architecture. For deeper models, a sequence of L E(3)-convolutional layers can be

chained together to allow the model greater flexibility. The irreps of these intermediate layers can include components up to a desired angular momentum $\ell_{\max}$, where higher values of ℓ correspond to more precise convolutions. In order to parameterize the on-site atomic orbitals, the output of the final $E(3)$ Conv layer must contain the following irreps for each atomic site:

- A vector of positive $(\ell, m, p) = (0, 0, 1)$ irreps $\boldsymbol{\alpha} = \{\alpha_{n\ell b}\}$, corresponding to the width of the bth basis Gaussian for the nth block of the atomic orbitals with angular quantum number ℓ. These coefficients are invariant under rotation and inversion.

- A vector of $(\ell, m, p) = (\ell, m, 1)$ irreps $\mathbf{c} = \{c_{n\ell mb}\}$ corresponding to the coefficient of the bth basis Gaussian for the nth block of the atomic orbitals with quantum numbers ℓ, m. These coefficients are equivariant with respect to rotation, and invariant under inversion.

Likewise, to parameterize the pseudopotentials, the final layer must also contain the following irreps for each atomic site:

- A vector of positive $(\ell, m, p) = (0, 0, 1)$ irreps $\boldsymbol{\gamma} = \{\gamma_b\}$, corresponding to the width of the bth basis Gaussian in the pseudopotential. These coefficients are invariant under rotation and inversion.

- A vector of $(\ell, m, p) = (\ell, m, (-1)^\ell)$ irreps $\mathbf{c} = \{c_{\ell m}\}$ corresponding to the coefficient of the bth basis Gaussian with quantum numbers ℓ, m. These coefficients are equivariant with respect to rotation and invariant under inversion. For most applications, including irreps up to $\ell = 2$ is sufficient to account for any induced dipole and quadrupole moments in the pseudopotential.

In Figure 5.7, we summarize both the design of the E(3)-convolutional layers and visualize the three-step process for computing convolutions.

5.3.4 Computing Electronic Structure

Next, we will discuss how the electronic structure can be computed from the parameterized atomic orbitals and pseudopotentials. In order to compute the electronic structure, one first needs to construct the effective mean-field Hamiltonian h with respect to the electron degrees of freedom $\mathbf{k}$. In QuINNs, a slightly simplified version of the mean-field Hamiltonian h is introduced, which employs the tight-binding approximation. This approximation assumes electrons are closely bound to the atomic lattices such that the Hartree and exchange-correlation contributions to the Hamiltonian can be decoupled into independent sub-Hamiltonians for each momentum vector $\mathbf{k}$. This provides sufficient accuracy to approximate valence shell states and low-energy conduction band states in the *first Brillouin zone*, which is the set of minimum-length images of all $\mathbf{k}$ vectors within the reciprocal unit cell. Indeed, it follows as a consequence of Bloch's theorem that the electronic structure of this zone is sufficient to represent the electronic structure of the entire crystal, and to compute arbitrary observables of the system, on account of the periodic lattice gauge symmetries of the system. Under the tight-binding approximation, the Hamiltonian matrix can then be written as $h(\mathbf{k})$, and its entries can be computed as:

$$\begin{aligned} h_{i,j}(\mathbf{k}) = e^{i\mathbf{k}\cdot\mathbf{r_{ij}}}\Big[\langle u_i(\mathbf{x})| \left(V_{\text{pseudo},i}(\mathbf{x}) + V_{\text{pseudo},j}(\mathbf{x})\right) |u_j(\mathbf{x})\rangle \Big] \\ + \frac{1}{2}\left[k^2 - 2i\mathbf{k}\cdot\langle u_i(\mathbf{x})| \nabla |u_j(\mathbf{x})\rangle - \langle u_i(\mathbf{x})| \nabla^2 |u_j(\mathbf{x})\rangle\right] \end{aligned} \tag{5.56}$$

where we assume $h_{i,j}(\mathbf{k}) = 0$ when $j \notin N(i)$ (orbital j is not in the neighborhood of orbital i). Above, the first term above in brackets represents the potential term,

while the additional pair of terms represent the kinetic contribution to the Hamiltonian. Both the kinetic and potential components of $h(\mathbf{k})$ can be computed in closed form. In fact, by grouping together blocks of these integrals centered on each site n, one can more efficiently compute blocks of overlap integrals between neighboring sites (denoted $\mathbf{\Sigma}_{n,n'}$) than by computing each nonzero entry of $h(\mathbf{k})$ individually. In Appendix **??**, we discuss in greater detail how these block integrals can be evaluated in a closed, differentiable form.

In addition to the off-diagonal elements of $h(\mathbf{k})$, the diagonal components are also critical to accurately modeling the electronic structure, since they represent the on-site energies of each orbital, both kinetic and potential. As shown in Figure 5.1, the Gaussian pseudopotentials and orbitals are intended to accurately approximate the true atomic functions beyond the atomic radius r_{atom}. As a result, computing the diagonal entries $h_{i,i}$ or $h_{i,j}$ when $\mathbf{r}_{ij} = 0$ may produce physically inaccurate estimates of the on-site orbital energies and can sometimes result in numerical instability with respect to the diagonalization of on-site atomic eigenstates. Given that all on-site orbitals are assumed to be directly orthogonal, these block components of $h(\mathbf{k})$ can be replaced by k-independent diagonal blocks, denoted by ϵ_n. The ϵ_n blocks can either be fixed constant values obtained from experimental atomic spectrum energies, or they can be directly predicted by another multi-layer equivariant neural network model. In general, using fixed on-site energies should be preferred to learned energies, since they fix the Hamiltonian and its eigenstate energies with respect to a well-defined vacuum energy. In Figure 5.8, we visualize the construction of the sparse block Hamiltonian $h(\mathbf{k})$ for a 2D graphene lattice, where valence shell s and p orbitals are included.

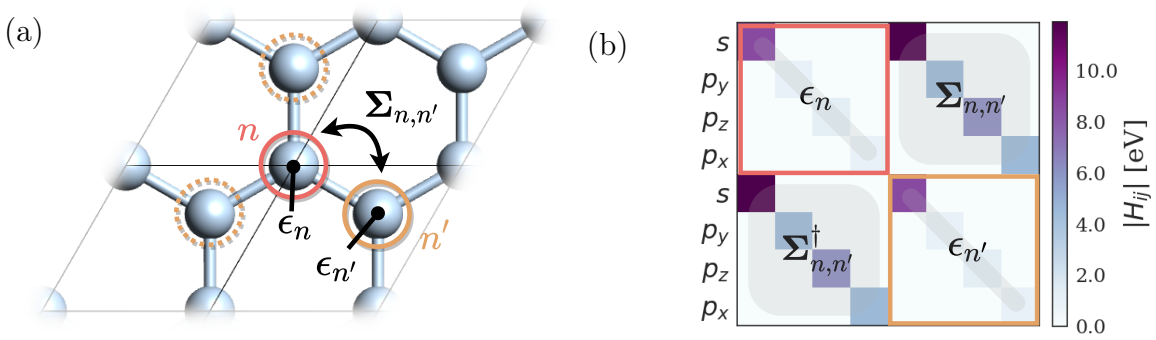

Figure 5.8: Illustration of the construction of the Hamiltonian (b) for a 2D graphene lattice (a). When computing the Hamiltonian matrix h for periodic materials, the periodic images of each site must be taken into account. The sums of the contributions of these images (including the Bloch orbital phases $e^{i\mathbf{k}\cdot(\mathbf{r}-\mathbf{r}')}$) comprise each interaction block $\mathbf{\Sigma}_{n,n'}$, while the on-site orbital energies $\epsilon_n, \epsilon_{n'}$ comprise the Hamiltonian diagonal. The blocks $\mathbf{\Sigma}_{n,n'}$ must be computed independently for each $\mathbf{k}$ vector.

When the Hamiltonian matrices are diagonalized to obtain the allowed electron energy levels for each momentum vector $\mathbf{k}$, these energy levels can be interpolated with respect to $\mathbf{k}$ to obtain the electronic band structure of a material. Specifically, we diagonalize

$$h(\mathbf{k}) = U(\mathbf{k})\ \Lambda_E(\mathbf{k})U(\mathbf{k})^\dagger \tag{5.57}$$

for each vector $\mathbf{k}$. Although the Hamiltonian $h(\mathbf{k})$ is non-periodic with respect to $\mathbf{k}$, the allowed energies $\Lambda_E(\mathbf{k})$ have the periodicity of the reciprocal unit cell. If the system of interest is a finite molecule, then only the $\mathbf{k} = 0$ point is considered, producing a single Hamiltonian that is free of these degrees of freedom. This $\mathbf{k} = 0$ point is often denoted by Γ, and marks the center of the first Brillouin zone. For larger cells containing many atoms, fewer $\mathbf{k}$-points are needed to accurately interpolate the band structure. In fact, in the limit of an infinitely large supercell comprised of identical copies of a reduced primitive cell, computing the electronic structure for only the $\mathbf{\Gamma}$ point is sufficient to obtain accurate estimates of the electronic structure. For

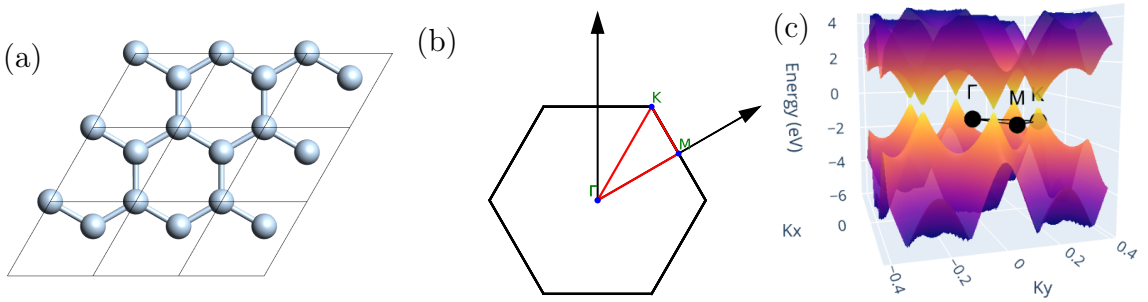

Figure 5.9: Visualization of the extended unit cell crystal structure (a) and first Brillouin zone and reciprocal lattice vectors (b) of graphene. Since graphene has 2D periodicity, the first Brillouin zone has two degrees of freedom, with the conventional Γ point centered at $\mathbf{k} = 0$. The remaining points $\mathbf{k} \in \mathbb{R}^2$ can be folded into the first Brillouin zone by computing the minimum-norm image of $\mathbf{k}$ with respect to the reciprocal cell lattice vectors (shown in (b)). In (c), the band structure predicted by a QuINN model trained on graphene electronic structure is shown as a function of $\mathbf{k}$, illustrating the dirac cones (hourglass-shaped features) that are characteristic topological features of graphene's electronic structure and effectively massless conduction band electrons.

small unit cells, typically the $\mathbf{k}$-points must be sampled uniformly over the reciprocal unit cell Ω^*, and the Hamiltonian must be computed for each point. One common sampling strategy is the *Monkhorst-Pack* mesh, which samples $k-points$ over uniform grid with respect to the reciprocal lattice vectors. For example, an $N_1 \times N_2 \times N_3$ Monkhorst-Pack mesh with a total of $M = N_1 N_2 N_3$ points is given by:

$$\{\mathbf{k}_m\}_{\text{Monkhorst-Pack}} = \left\{ \sum_{i=1,2,3} \frac{2n_i - N_i - 1}{2N_i} \mathbf{b}_i \;\middle|\; n_i = 1, 2, ..., N \right\} \tag{5.58}$$

This mesh ensures uniform sampling of points over Ω^* and can be used as a quadrature for integration and interpolation over Ω^*. When combined with atomic structure, this $\mathbf{k}$-point mesh can be used to compute the electronic band structure as a function of $\mathbf{k}$. In Figure 5.9, we give an example of the band structure computed as a function of $\mathbf{k}$ for the 2D material graphene. In Figure 5.10, we summarize the entire process of how QuINNs compute electronic structure using this $\mathbf{k}$-point mesh.

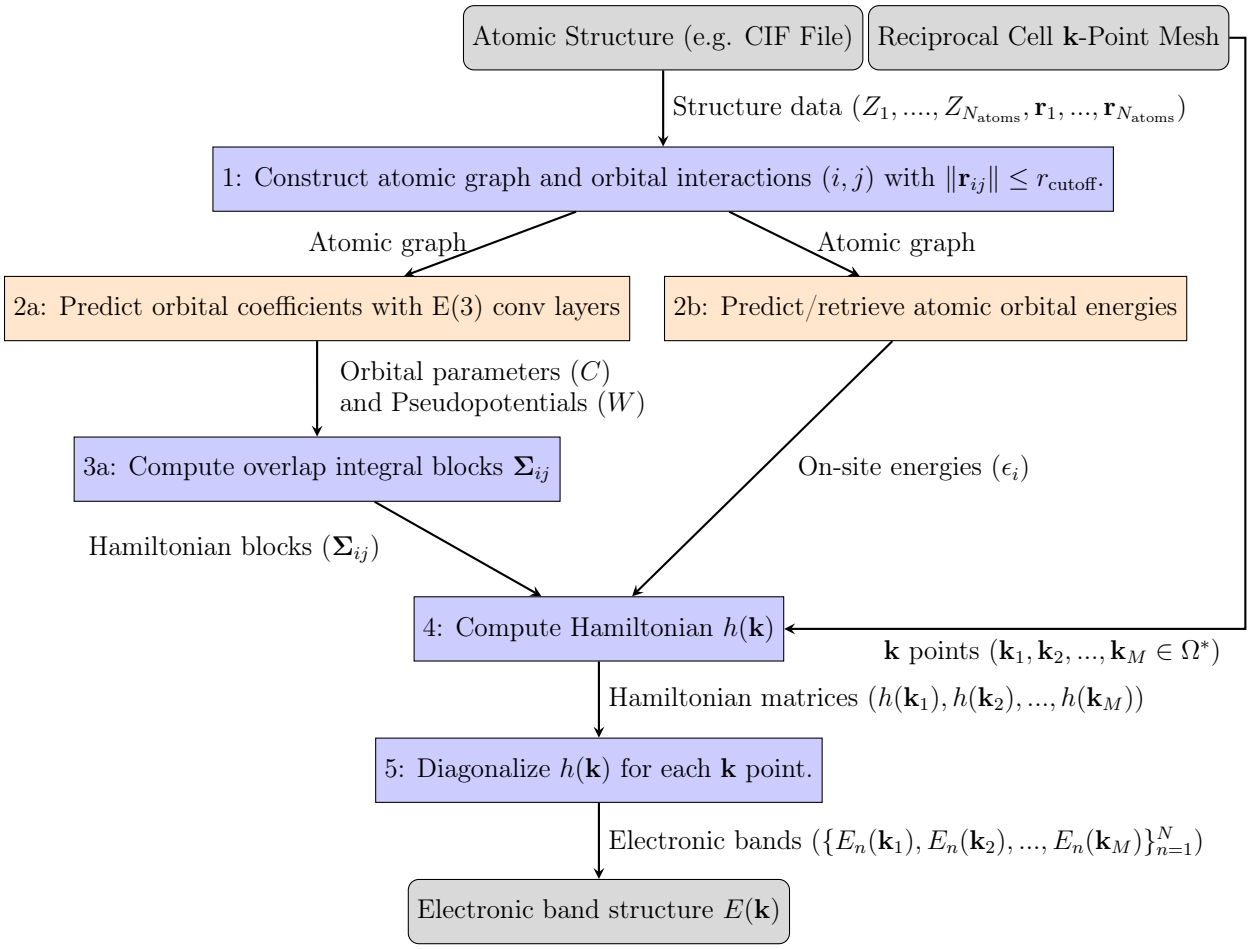

Figure 5.10: Data flowchart summarizing how QuINNs predict the electronic structure of solid state materials. From the electronic structure, almost any electronic property can be computed from first principles.

Once the electronic structure energies $E(\mathbf{k})$ are computed for a discrete $\mathbf{k}$-point mesh, the ground state electronic structure ρ_0 can be computed in density matrix form as a function of $\mathbf{k}$. Specifically, for a fixed chemical potential $\mu = 0$ and a diagonalized Hamiltonian $h(\mathbf{k}) = U(\mathbf{k})\ \Lambda_E(\mathbf{k})\ U(\mathbf{k})^\dagger$, we define

$$\rho_0(\mathbf{k}) = U(\mathbf{k})\ f(\Lambda_E(\mathbf{k}))\ U(\mathbf{k})^\dagger, \tag{5.59}$$

where $f(E) = 1$ for $E < 0$ and 0 otherwise. In the case of spin-paired orbitals, one should instead use $f(E) = 2$ for occupied states.

When modeling periodic systems, sometimes the chemical potential μ is unknown, but the number of electrons in the unit cell is a known fixed quantity (this is a standard assumption for most mean-field treatments of electrons in periodic solids). In these situations, the chemical potential corresponds with the *Fermi level* E_f of the material, which is the energy at which the probability of electron occupation is precisely 1/2. More generally, E_f is defined as the average energy required to add a single electron to the system. If the system has N_{elec} electrons per unit cell, the Fermi energy E_F satisfies:

$$N_{\text{elec}} = V_\Omega \int_{-\infty}^{E_F} D(\epsilon)\ d\epsilon \tag{5.60}$$

where V_Ω is the unit cell volume and $D(\epsilon)$ is the *density of states* (DOS). The DOS describes the number of electron states with energy ϵ per crystal unit volume, and it is given by

$$D(\epsilon) = \int_{\Omega^*} \sum_n \delta(E_n(\mathbf{k}) - \epsilon)\ d\mathbf{k} \tag{5.61}$$

If the reciprocal unit cell Ω^* is sampled with a fine $\mathbf{k}$-point mesh, the DOS can be approximated as:

$$D(\epsilon) = V_\Omega \sum_{m=1}^{M} d(E(\mathbf{k}_m) - \epsilon) \tag{5.62}$$

where d is a smearing delta function (a comon choice is $d(\epsilon) = \frac{d}{d\epsilon} f_\beta(\epsilon)$, where f_β is the Fermi-Dirac distribution with inverse temperature β). Using this approximate DOS, one can solve for the Fermi energy E_f, and determine crucial properties, such as whether a given material is a metal, insulator, or semiconductor. For insulating and semiconducting materials, the band gap can be also be directly estimated.

Finally, we can also compute the ground state expectation value of any electronic mean-field quantum observable $\hat{O}$ expanded as a matrix $\mathbf{O}(\mathbf{k})$ in the basis of Bloch orbitals $\phi_{\sigma\mathbf{k}}(\mathbf{x})$. To do this, we compute:

$$\langle \hat{O} \rangle = \int_{\Omega^*} \mathrm{Tr}(\rho_0(\mathbf{k})\ \mathbf{O}(\mathbf{k}))\ d\mathbf{k}. \tag{5.63}$$

In addition to the expectation value of observables and the density of states, other properties derivable from the topological features of solid state electonic structure can also be computed, such as Fermi-surfaces (if they exist) and the projections of bands and density of states onto the orbitals of each site. The projections of the contribution of each orbital and site onto the electronic structure can be extracted from the unitary $U(\mathbf{k})$ that diagonalizes the Hamiltonian $h(\mathbf{k})$. Furthermore, the Fermi surface, an important level curve of the electronic structure in the first Brillouin zone that dictates the conductivity and topological properties of metals, can be computed efficiently by evaluating the computing the band structure with high resolution around points in the first Brillouin zone that cross the Fermi level E_f. In

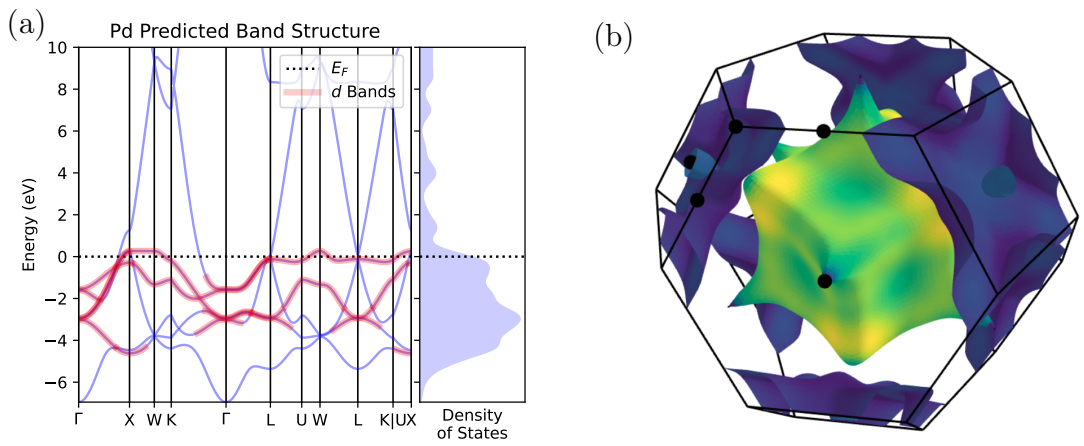

Figure 5.11: Visualizations of d-band and s-band contribution of electronic structure of FCC palladium. (a): The band structure of bulk Pd, as predicted by a pre-trained QuINN prototype model. The bands are shaded red in proportion to the d-orbital contribution of each eigenstate with energy $E(\mathbf{k})$. (b): Calculation of the Fermi surface of Pd in the First Brillouin zone, illustrating the large surface-area contribution of the d-orbitals (blue) and the s-orbitals (green). A visualization of the canonical high-symmetry path used in (a) can be found in Figure 5.12(c).

Figure 5.11, we give examples for computing both of these quantities for bulk FCC palladium, a metal that is commonly used as a reaction catalyst [118].

5.4 Training QuINNs

As a type of physics-informed neural network, QuINNs must be trained in a way that affords the model a great deal of flexibility to fit the data is it trained on, but also penalize the model for deviating from the expected solution obtained from more physically rigorous solutions to the Schrodinger equation. In this section, we will briefly discuss how QuINNs can be trained, and how the training data and loss function contribute to the accuracy, equivariance, and physical consistency of the model.

5.4.1 Generation of Band Structure Training Data

First, we must discuss how training data is generated for QuINN models. For solid state materials, plane-wave density functional theory methods remain the gold standard for accurate mean-field modeling of electronic structure in periodic systems. For materials with small unit cells, DFT calculations can be carried out efficiently and accurately using established software such as Quantum ESPRESSO [119]. These software systems have support for many different DFT exchange-correlation functionals and support the incorporation of many different effects, ranging from relativistic spin-orbit coupling to Van der Waals interactions. Most DFT codes support the calculation of electronic band structure $E(\mathbf{k})$ along arbitrary paths in the first Brillouin zone (or reciprocal unit cell, if desired). Depending on the symmetries present in the system, the first Brillouin zone can be divided into similarly shaped sub-zones, where the band structure in each sub-zone is irreducibly isomorphic to that of another sub-zone up to a rotation and possible reflection of the electronic bands. For this reason, the band structure is often visualized as a function along a two-dimensional path around the points that comprise the exterior of this region. This is referred to as the *canonical high-symmetry path*, and the associated points at the zone corners are called *critical points*. In Figure 5.12, we visualize the first Brillouin zones and canonical high-symmetry paths corresponding to different crystal systems. The convex polytope that encloses the canonical high-symmetry path is called the *irreducible Brillouin zone*, and it represents the set of $\mathbf{k}$-points on which the band structure must be evaluated. For all other $\mathbf{k} \in \mathbb{R}^3$, the band structure can be directly inferred from the bands inside the irreducible Brillouin Zone. As a result, QuINN models only need to be trained on $\mathbf{k}$-points sampled within this zone. Furthermore, since bands often

attain their extrema along the canonical high-symmetry path forming the exterior of this region, fitting QuINN models to the band structure corresponding to $\mathbf{k}$-points along this path is often sufficient for obtaining accurate bands in the interior Brillouin zone and, by extension, all $\mathbf{k} \in \mathbb{R}^3$. Since there a finite number of 3D, 2D, and 1D space and point groups, the number of types of irreducible Brillouin zones is finite, and thus many DFT softwares and crystallographic software packages support the generation of these canonical paths, such as those seen in Figure 5.12.

In addition to computing the electronic band structure, many DFT codebases support computing the projection of the density (expanded in the basis of plane waves) onto localized Gaussian orbitals. Specifically, for a set of atomic sites centered at a positions $\mathbf{r}_i$, the projection of the density of a plane-wave electronic eigenstate $\psi_{n,\mathbf{k}}(\mathbf{x})$ onto a localized basis of spherical harmonic orbitals is computed as the vector-valued function $\boldsymbol{\pi}_n^{(\mathrm{DFT})}(\mathbf{k})$, which has coefficients

$$[\boldsymbol{\pi}_n^{(\mathrm{DFT})}(\mathbf{k})]_{\ell mi} = \frac{1}{A} \int_\Omega |\psi_{n\mathbf{k}}^{(\mathrm{DFT})}(\mathbf{x})|^2 Y_{\ell m}(\mathbf{x} - \mathbf{r}_i)\ d\mathbf{x}, \tag{5.64}$$

Above, A is a normalization factor chosen such that $\sum_{\ell,m,i}\ [\boldsymbol{\pi}_n(\mathbf{k})]_{\ell mi} = 1$. These projections of atomic orbitals onto the electronic eigenstates are unique for all non-degenerate bands (i.e. bands with unique energies $E_n(\mathbf{k})$). To obtain sufficient data for training a QuINN model, the electronic bands $E_n(\mathbf{k})$ and associated projection vectors $\boldsymbol{\pi}_n(\mathbf{k})$ evaluated along the canonical high-symmetry paths for each material that is to be included in the training dataset.

The equivariance of QuINNs significantly reduces the number of distinct materials needed to train QuINNs, and eliminates the need to augment material data by

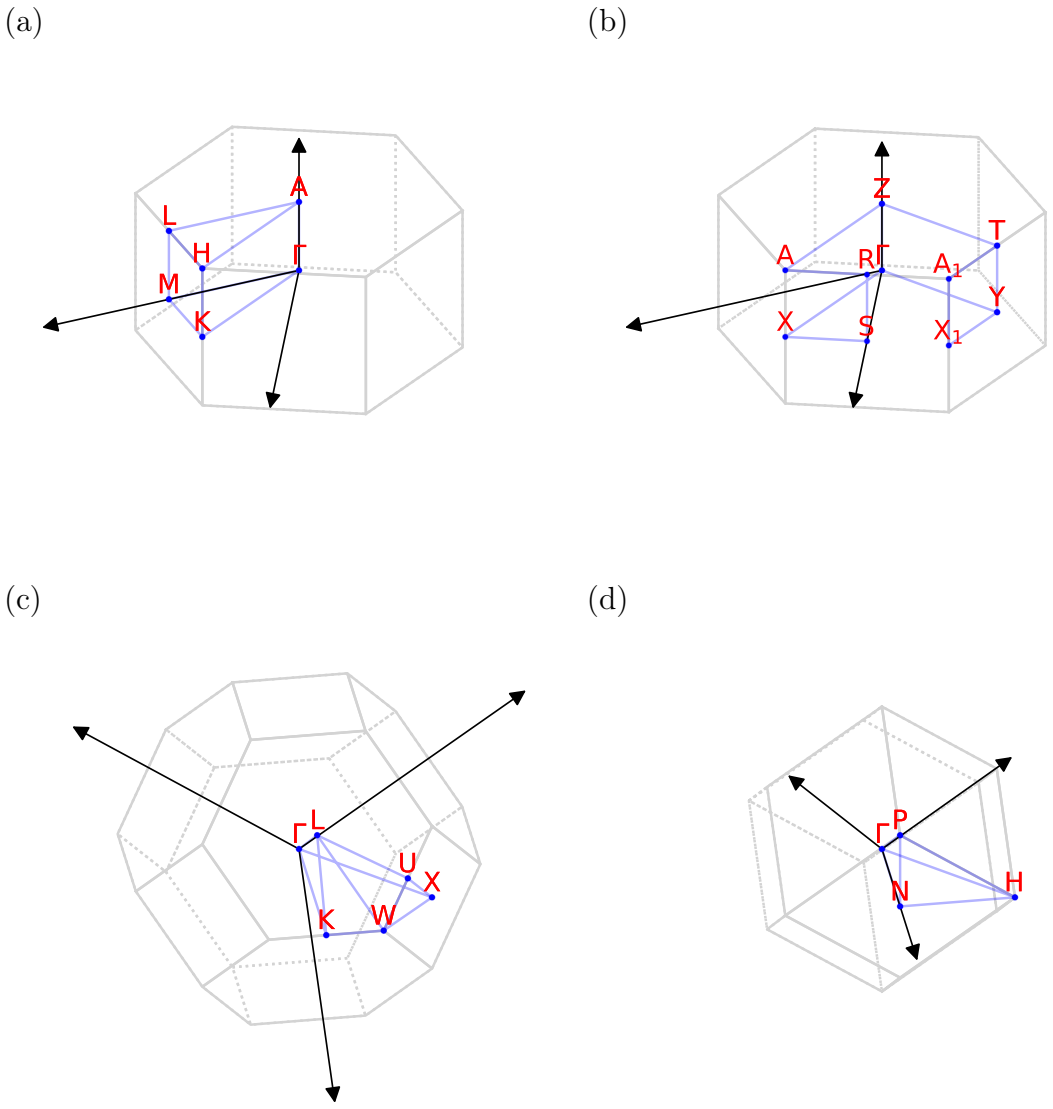

Figure 5.12: Example visualizations of the first Brillouin zone, which is the set of minimal-image $\mathbf{k}$ vectors in the reciprocal unit cell Ω^* of 3D periodic materials. Zones are shown for materials with (a) hexagonal, (b) base-centered orthorhombic, (c) face-centered cubic, and (d) body-centered cubic structures. For each Brillouin zone, the canonical high-symmetry path and critical points are shown. Due to the equivariance of the model and the lattice gauge-invariance of the Hamiltonian, a model trained on the electronic structure enclosed by these high-symmetry paths within the first Brillouin zone is sufficient for obtaining accurate electronic structure at all crystal momentum vectors $\mathbf{k} \in \mathbb{R}^3$.

including rotations, reflections, translations [115,120]. Furthermore, due to the natural lattice gauge invariance of the tight-binding Hamiltonian, one also does not need to include any symmetry-reducible supercells in the training data. However, because QuINNs treat each element independently, one must ensure that all element bond pairs for which accurate predictions are desired must be represented in the training data. In addition, structures of different dimensionality (such as 2D thin films versus bulk 3D structures) must also be included in the training data, to ensure that the dimensionally-dependent orbital anisotropy is correctly learned. Once sufficient training data is obtained via DFT calculations or other first-principles methods, each structure in the training set will consist of the following:

- An atomic structure graph with atomic species $\{Z_i\}_{i=1}^{N_{\text{atoms}}}$ and positions $\{\mathbf{r}_i\}_{n=1}^{N_{\text{atoms}}}$. If the structure is periodic, the lattice vector matrix $\mathbf{A}$ must be included, in addition to the interatomic distance vectors $\mathbf{r}_{ij}$ for each periodic image.

- A canonical high-symmetry path consisting of irreducible Brillouin zone $\mathbf{k}$ points $\{\mathbf{k}_m\}_{m=1}^{M}$.

- The electronic band structure for the first N eigenstates along the high-symmetry path $\{E_n(\mathbf{k})\}_{n=1}^{N}$.

- The electronic band structure projection vector $\{\boldsymbol{\pi}_n(\mathbf{k})\}_{n=1}^{N}$ along the high-symmetry path, where the entries $[\boldsymbol{\pi}_n(\mathbf{k})]_{\ell mi}$ correspond to the projection of the electronic eigenstate onto the spherical harmonic orbital $Y_{\ell m}$ at site i.

5.4.2 *Gradient Descent Optimization*

QuINN models are trained to fit a dataset by minimizing the average value of a loss function across all structures through gradient descent. The key operation in gradient decent operation is the computation of the gradient of the loss function with respect to the $E(3)$-convolutional layer weights. These gradients can be computed efficiently in closed form, provided that the calculation of the electronic structure can be written as the composition of functions whose derivatives are known in closed form. If implemented using the primitive functions supported in an automatic differentiation software framework (such as the popular PyTorch [121] or TensorFlow [122] machine learning libraries), gradients can be computed automatically and efficiently through backpropagation. The basic loss function employed by QuINNs is of the form

$$L_{\text{QuINN}} = w_{\text{bands}} L_{\text{bands}} + w_{\text{proj}} L_{\text{proj}} \tag{5.65}$$

which closely resembles the loss function of a PINN. This consists of two weighted components: the electronic band loss L_{bands} and the projection loss L_{proj}. The band loss corresponds to the data loss function in PINNs, which measures the error of each predicted energy level in the electronic structure along a canonical irreducible Brillouin zone path $\{\mathbf{k}_m\}_{m=1}^{M}$. Using the mean square error criterion, the band loss takes the form

$$L_{\text{bands}} = \frac{1}{NM} \sum_{n=1}^{N} \sum_{m=1}^{M} (E_n^{(\text{pred})}(\mathbf{k}_m) - E_n^{(\text{true})}(\mathbf{k}_m))^2. \tag{5.66}$$

In addition to predicting accurate band structure, the contribution of each site and atomic orbital must be incorporated into the loss. This is achieved by ensuring that the occupation probabilities of each eigenstate are orthogonal and consistent with respect to the projections obtained with DFT. This ensures that QuINN models

"learn" which bands are associated with each site and orbital for each $\mathbf{k}$ point. These projections can be read out as vectors $\boldsymbol{\pi}_n^{(\mathrm{pred})}(\mathbf{k})$ from the rows of the adjoint unitary matrix $U^\dagger(\mathbf{k})$ corresponding to the band energy $E_n(\mathbf{k})$. The entries of these vectors are given by

$$[\boldsymbol{\pi}_n^{(\mathrm{pred})}(\mathbf{k})]_{\ell m i} = |[U^\dagger(\mathbf{k})_n]_{\ell m i}|^2 \tag{5.67}$$

where ℓ, m denote the projection onto the corresponding orbital at site i. The projection loss function L_{proj} is then computed by evaluating the norm squared of the difference between the projection vectors $\boldsymbol{\pi}_n$ computed by DFT and the QuINN model. Since the projections are only uniquely defined for non-degenerate energies $E_n(\mathbf{k})$, one must be careful when defining the projection loss to ensure it is not sensitive to the non-uniqueness of degenerate eigenstates. One way of handling degeneracy is to continuously "soften" the projection loss when the predicted energies are close to degenerate values. This is achieved by using a L_{proj} loss function of the form

$$L_{\mathrm{proj}} = \frac{1}{NM} \sum_{n=1}^{N} \sum_{m=1}^{M} \chi(\Delta E_n^{(\mathrm{pred})}(\mathbf{k}_m)) \left\| \boldsymbol{\pi}_n^{(\mathrm{pred})}(\mathbf{k}) - \boldsymbol{\pi}_n^{(\mathrm{true})}(\mathbf{k}) \right\|^2 \tag{5.68}$$

where ΔE_n is the band gap factor, computed as

$$\Delta E_n = \min(|E_{n+1} - E_n|, |E_{n-1}, E_n|), \tag{5.69}$$

and where $\chi(\Delta E)$ is the softening function

$$\chi(\Delta E) = 1 - \exp\left(-\frac{(\Delta E)^2}{2\varepsilon}\right) \tag{5.70}$$

for a small energy gap ε.

Next, we consider the problem of how the gradients of the loss function L_{QuINN} are computed with respect to the model weights. The $E(3)$-equivariant layers in QuINNs are constructed from simple matrix and tensor operations with non-linear

activation functions, so backpropagation can be applied using methods from standard neural networks. However, backpropagating gradients through the computation and subsequent diagonalization of the $\mathbf{k}$-dependent Hamiltonian matrices $h(\mathbf{k})$ warrants some more detailed discussion, since computing derivatives for these operations is non-trivial, and requires some special treatment to avoid gradient instability issues.

First, as a consequence of the Hellmann-Feynmann theorem, partial derivatives of the eigenvalues of a parameterized Hamiltonian $h_\lambda(\mathbf{k})$ with parameter λ are given by

$$\frac{\partial}{\partial\lambda}E_n(\mathbf{k}) = \langle\psi_n(\mathbf{k})| \frac{\partial h_\lambda}{\partial\lambda}(\mathbf{k}) |\psi_n(\mathbf{k})\rangle \tag{5.71}$$

and therefore the gradient of L_{bands} (which only includes band energy error) with respect to model weights $\mathcal{W}$ can be computed in terms of the band matrix $\Lambda_E = \text{diag}(E_0, E_1, ..., E_N)$ obtained for each $\mathbf{k}$ point:

$$\frac{\partial L_{\text{bands}}}{\partial\mathcal{W}} = \frac{\partial L_{\text{bands}}}{\partial\Lambda_E(\mathbf{k})}\left(U(\mathbf{k})\frac{\partial\Lambda_E}{\partial W}(\mathbf{k})U(\mathbf{k})^\dagger\right). \tag{5.72}$$

Likewise, extending Hellmann-Feynmann with first-order eigenstate perturbation theory, the derivatives of the eigenstates $|\psi_n\rangle$ of a Hamiltonian h_λ with parameters λ take the form

$$\frac{\partial}{\partial\lambda}|\psi_n(\mathbf{k})\rangle = \left(\sum_{n'\neq n}\frac{1}{E_n(\mathbf{k}) - E_{n'}(\mathbf{k})}\langle\psi_n(\mathbf{k})|\frac{\partial h_\lambda(\mathbf{k})}{\partial\lambda}|\psi_{n'}(\mathbf{k})\rangle\right)|\psi_n(\mathbf{k})\rangle. \tag{5.73}$$

When the eigenstates are degenerate, these gradients are not well-defined. However, due to the presence of the softening function $\chi(\Delta E)$, the projection error of degenerate eigenstates is suppressed, resulting in stable projection gradients. For non-degenerate eigenstates, the gradients are only well-defined modulo a complex phase of $|\psi_n(\mathbf{k})\rangle$. For this reason, it is preferable to only compute gradients with respect to the modulus

squared of the eigenstate vectors, which is guaranteed to be independent of the representation of the eigenbasis. This is why the projection loss is computed in terms of the modulus-squared projection vectors $\boldsymbol{\pi}_n$. In terms of the entries of the projection vectors $\boldsymbol{\pi}_n$, gradients with respect to a Hamiltonian parameter λ can be computed as

$$\left[\frac{\partial}{\partial\lambda}\boldsymbol{\pi}_n(\mathbf{k})\right]_{\ell mi} = 2\,\mathrm{Re}\left\{[\psi_n^*(\mathbf{k})]_{\ell mi}\left[\frac{\partial}{\partial\lambda}\,|\psi_n(\mathbf{k})\rangle\right]_{\ell mi}\right\}. \tag{5.74}$$

By combining all of the formulas above, the gradients of L_{proj} with respect to model weights $\mathcal{W}$ can be computed in closed form.

5.4.3 Training QuINNs

QuINN models are trained by minimizing the supervised loss function L_{QuINN} over a dataset consisting of atomic structures and their corresponding electronic structure, consisting of both energy bands $E(\mathbf{k})$ and orbital projections $\boldsymbol{\pi}_n(\mathbf{k})$ along the canonical high-symmetry Brillouin zone path. Training begins with symbolic representations of materials encoded as atomic graph objects, which are then compiled into backend-specific tensor representations of these graphs for differentiable computation. Each graph encodes atomic positions, species, bonding topology, and lattice information, which are passed along with a sequence of crystal momentum ($\mathbf{k}$) points to the model.

Optimization of L_{QuINN} is performed using standard gradient-based methods (e.g., *adaptive moment estimation* [123]), and models are trained over epochs using data loaders that stream batches of graph–bandstructure pairs. Due to the non-homogeneity of the atomic structure graphs, training must be done with small batch sizes, where each graph's Hamiltonian $h(\mathbf{k})$ is diagonalized separately and the losses are pooled manually. Due to the practical memory constraints during training, a

smaller batch size is generally preferred. The training loop tracks both training and validation losses and includes safeguards against numerical instability during diagonalization. Throughout the process, PyQuINN preserves compatibility with the backend autograd system that is used, enabling end-to-end differentiability and integration into larger or more application-specific learning pipelines.

Together, the design of the QuINN architecture and its training paradigm enable the surrogate modeling of quantum materials in a way that is both expressive and physically grounded. Unlike standard crystal graph GNNs, which may approximate properties directly from atomic graph topologies, QuINNs learn structured latent representations—such as orbital overlaps and pseudopotentials—that can be assembled into full Hamiltonians. Training proceeds by minimizing a supervised loss over predicted vs. true band structures, optionally including projection data, allowing the model to learn accurate representations of material-specific electronic structure. Because all operations in the QuINN pipeline are differentiable, the model supports not only forward prediction but also inverse design through gradient-based optimization. These capabilities make QuINNs a powerful tool for tasks ranging from high-throughput screening to automated discovery of materials with targeted electronic properties.

5.5 The PyQuINN package

To facilitate the development and exploration of QuINN model architectures we have developed a software package called PyQuINN, which provides many convenient tools for building, training, and deploying QuINN models to solve electronic structure and material property prediction problems, both in industry and research

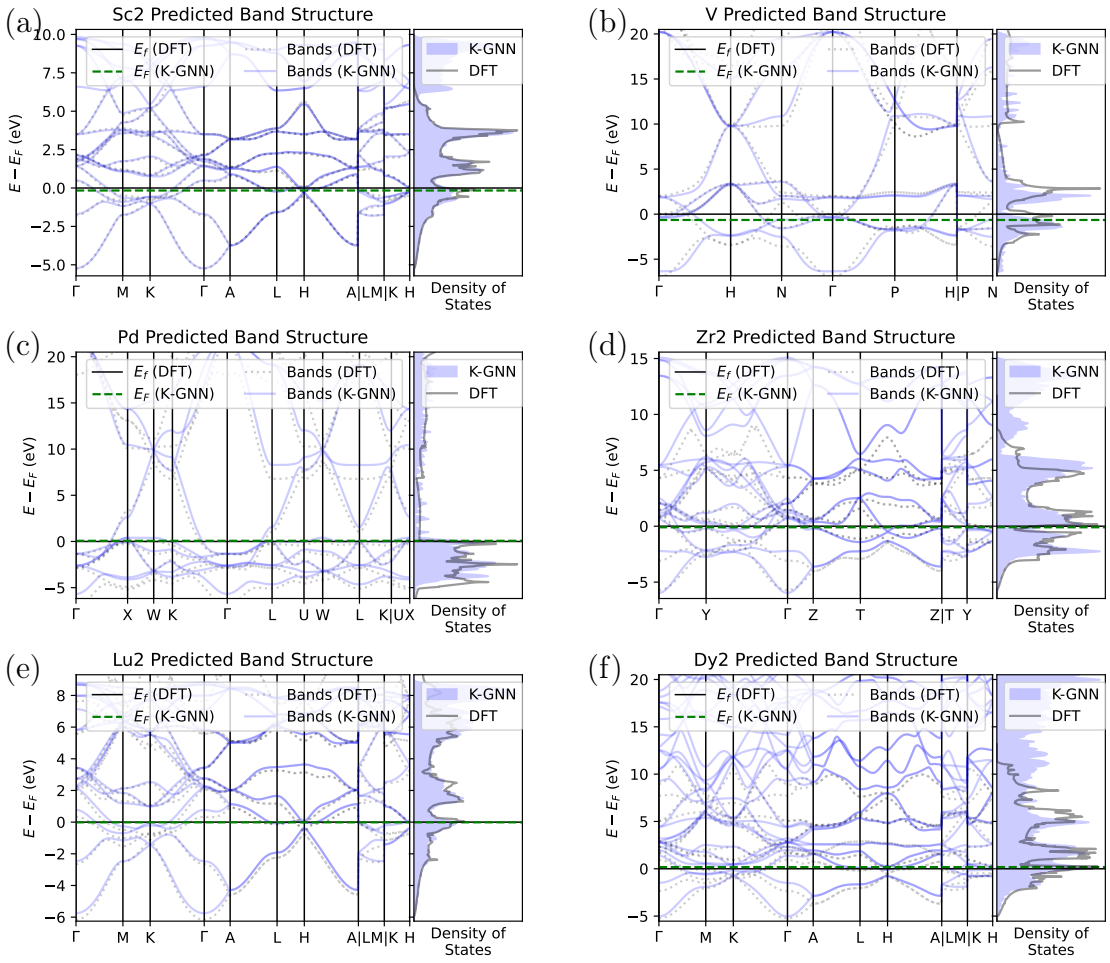

Figure 5.13: Examples of validation set predictions made by a QuINN prototype model (K-GNN) on single-element electronic structure datasets. Results are compared with *ab initio* DFT calculations of electronic band structure along canonical high-symmetry paths and show strong agreement, highlighting the generalizability of QuINN models to new atomic structures. For each figure, a QuINN model was trained on a dataset consisting of all known thermodynamically stable structures of the corresponding element. These datasets were obtained from the Materials Project database and partitioned using a 90%/10% train/validation set split. Visualizations of each high-symmetry path in the first Brillouin zone are shown above in Figure 5.12.

settings. The framework is written in the Python programming language is is designed to be backend-agnostic, allowing for the implementation of certain complex operations (such as training, inference, etc.) using various backend implementations of the model architecture. Although it is not listed publicly on the Python package index, the package can be installed from source code using the `pip` Python package manager. The primary backend for PyQuinn is the popular PyTorch machine learning framework [121], which supports automatic differentiation for various composable functions and allows users to define their own QuINN model layers. PyTorch's primitives also allow for efficient compilation of models to be trained on both CPU and GPU hardware. In this section, we give a brief summary of the PyQuINN framework and how it can be used to develop QuINNs to solve electronic structure problems.

5.5.1 Overview

PyQuINN is a Python framework for symbolic and differentiable modeling of quantum materials, designed to facilitate the rapid prototyping and deployment of machine-learned models for electronic structure prediction. At its core, this package provides the `AtomGraph` data structure, which represents atomic systems as symbolic graphs that represent an atomic configuration in 3D space with periodicity. These graphs can be constructed from standard crystallographic formats such as CIF files [124] and enriched with on-site and global feature data. PyQuINN supports the symbolic manipulation of these graphs, allowing researchers to programmatically build atomic structures in an easy-to-manipulate fashion prior to compilation into backend-specific representations (instances of the `BackendAtomGraph` object) that can serve as inputs for various machine learning frameworks, such as Pytorch. The

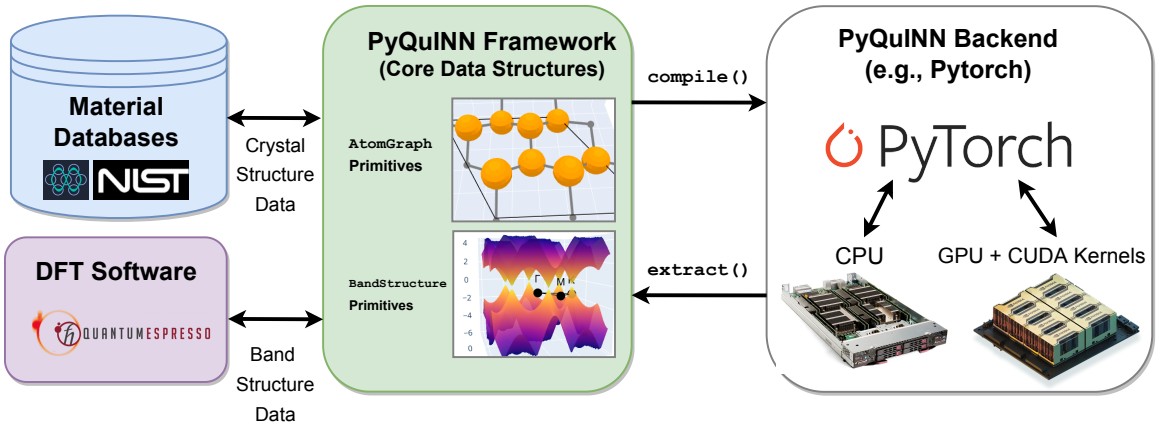

Figure 5.14: Illustration of the high-level features of the PyQuINN framework, showing the integration of the core PyQuINN framework with material databases such as the Materials Project, and DFT software such as Quantum ESPRESSO. The core framework provides access to data structures like `AtomGraph` and `BandStructure` which can be manipulated and compiled to serve as data structures in various backend frameworks, like PyTorch, which supports automatic differentiation and training of models on both CPU and GPU hardware.

package also provides utilities for visualizing atomic structures and computed band structures, and it includes helpful interfaces for importing/exporting atomic structures and electronic structure data from materials databases such as the Materials Project and ab initio tools such as Quantum ESPRESSO. These high-level features of the PyQuINN framework are illustrated in Figure 5.14.

5.5.2 Data Structures

To make the generation, curation of QuINN models easier, the core PyQuINN package supports primitive data structures that can be constructed directly using the Python interface, or saved/loaded in a variety of established file formats. Below, we briefly summarize each of these data structures and how they can be used to more efficiently develop QuINN models:

- The `AtomGraph` primitive in the PyQuINN package serves as a high-level, symbolic representation of atomistic systems, encoding both structural and electronic features in a graph-based format suitable for quantum-informed machine learning workflows. At its core, an `AtomGraph` encapsulates atomic species, fractional or Cartesian coordinates, periodic boundary conditions, unit cell geometry, and optional node- and graph-level features that describe properties such as atomic orbitals or target observables. Symbolic structures (where symbols are used in place of elements in a particular group) are also supported. `AtomGraph` structures can be constructed from crystallographic files (e.g., CIF [124]), programmatically manipulated (e.g., edge pruning, feature augmentation), and compiled into backend-specific numerical representations for efficient computation. When compiled with the `compile()` function, an `AtomGraph` is transformed into a `BackendAtomGraph` (such as the `TorchAtomGraph` when using the PyTorch backend), which maps the symbolic representation into differentiable tensor formats. This backend graph includes neighbor lists, unit cell image tracking, and support for tensor operations on arbitrary devices. The compiled graph salso upports in-place inference with QuINN models defined in the respective backend and can be serialized, transferred across devices, or exported back to a symbolic AtomGraph via the `extract()` method. Through this abstraction, AtomGraph enables seamless transitions between symbolic modeling, differentiable simulation, and visualization or export of structures, orbitals, and electronic observables, facilitating a simplified workflow for materials discovery.

- The `BandStructure` primitive in the PyQuINN package provides a symbolic representation of the electronic band structure of a material, capturing the dispersion of electronic energy levels along momentum space paths. It encodes arrays of k-points, corresponding energy eigenvalues (bands), and optional projection data that quantify the orbital or atomic contributions to specific bands. Designed to be interoperable with other PyQuINN primitives like `AtomGraph`, the BandStructure object supports import from external data sources and export to common file formats, as well as resampling utilities to align disparate datasets in k-space or band index. When compiled, a `BandStructure` is transformed into a backend-specific object (such as `TorchBandStructure`), which maps the electronic structure data into PyTorch tensors suitable for differentiable computation and hardware acceleration. These compiled objects preserve all semantic fields—including k-point paths, energy eigenvalues, projection weights, and chemical potential, supporting standard tensor operations for integration into QuINN model training pipelines. Users can detach, copy, move the data across hardware devices, or extract the data back into a symbolic `BandStructure` form for visualization or post-processing. This dual symbolic/numeric interface makes `BandStructure` a central component in workflows involving model-based band structure prediction, electronic structure learning, and topological material analysis.

- The `BandStructureModel` module in PyQuINN implements a learnable, differentiable model for predicting electronic band structures from atomistic inputs. It wraps a variety of different QuINN models of varying complexity

and provides methods to compute Hamiltonians, band energies, and orbital-resolved band projections given a compiled `AtomGraph` and a set of crystal momentum ($\mathbf{k}$) points. The model supports both forward prediction (bands, band projections) and end-to-end training via a custom loss function that incorporates optional projection targets. It includes a fit routine for supervised learning from datasets of atomic structures and target band structures, along with utilities for computing and projecting Bloch states, calculating chemical potentials, and evaluating losses. Designed for integration into backend machine learning frameworks (such as PyTorch), the module enables physically informed training and inference of band topology and dispersion in complex materials.

In Figure 5.15 we present a flowchart that summarizes how these primitive data structures can be used to predict and plot the electronic band structure using the functionality of a desired backend machine learning framework.

5.5.3 Data Visualization

PyQuINN provides lightweight yet expressive visualization capabilities for inspecting atomic structures, orbital configurations, and electronic band structures. These tools enable users to render atomic geometries with annotated species and bonds, visualize localized and extended electronic orbitals in real or reciprocal space, and plot computed band structures along high-symmetry paths with optional orbital or atomic projections. Integrated with standard Python plotting libraries like Matplotlib [125] and Plotly [126], PyQuINN's visualization modules support both interactive analysis during model development and rendering of high-quality figures

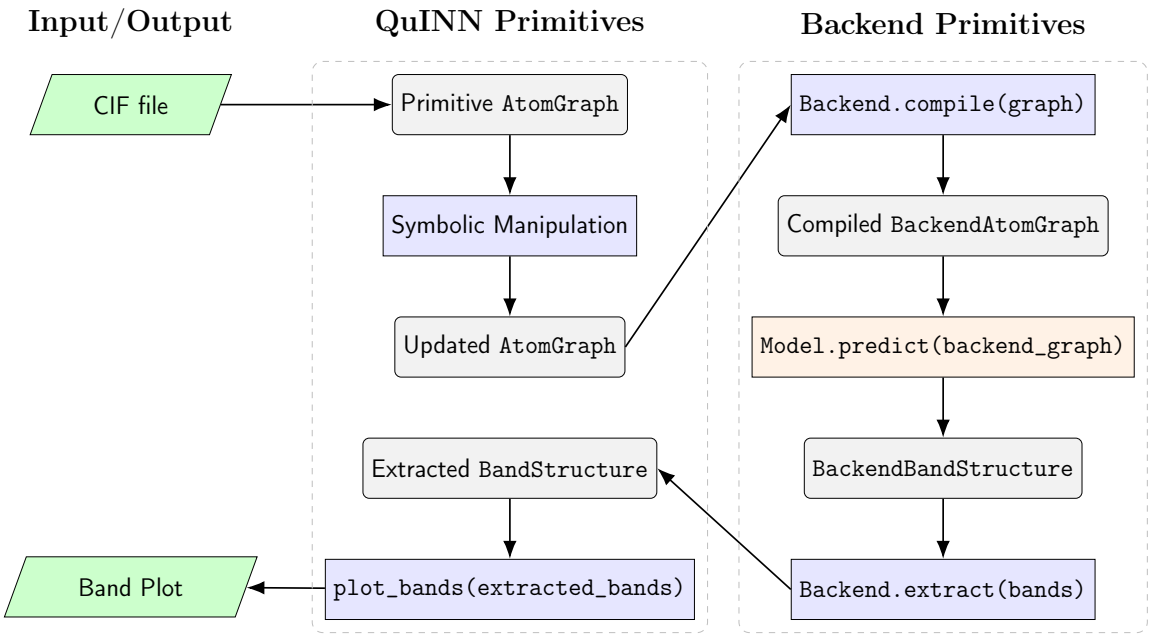

Figure 5.15: Flowchart of a basic band structure prediction workflow in the PyQuINN framework, illustrating the separation between data I/O, QuINN primitives, and backend-specific primitives (e.g., primitives using Pytorch data types). This allows for a more modular and extensible framework, which is unified via the interface provided by primitives data structures, such as `AtomGraph` and `BandStructure`.

for publication or presentation. This helps users interpret model outputs, verify data integrity, and explore structure–property relationships in quantum materials. In Figure 5.16, we show some examples of the interactive plots that can be generated using the Plotly Python library.

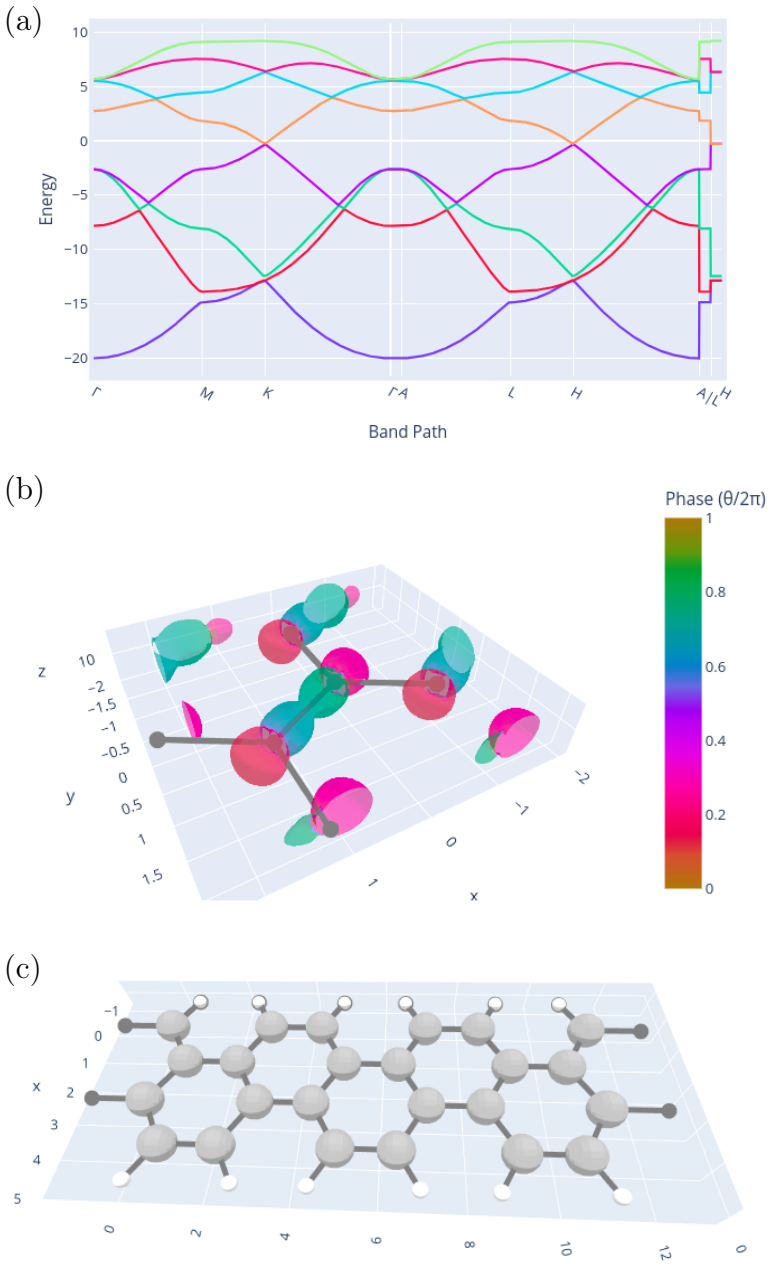

Figure 5.16: Examples of interactive visualizations generated with PyQuINN and the Plotly Python library. (a): Interactive electronic band stucture for bulk graphene, predicted by a QuINN model. (b): Interactive 3D visualization of an sp^2-hybridized Bloch eigenstate in bulk graphene, showing the formation of a π-bond. (c): Interactive 3D visualization of a hydrogen-saturated graphene nanoribbon.

CHAPTER SIX

Quantum Computational Methods

6.1 Introduction

In this section, we will give a basic review of quantum computing and quantum information theory, as applicable to the field of quantum chemical and material simulations. We start with a pedagogical introduction that provides a basic summary of the background of quantum computing and the foundational concepts needed to understand it. We then briefly discuss the differences between classical and quantum computation, and contextualize these differences by summarizing key results in computational complexity theory, and giving examples of relevant complexity classes that categorize the kinds of physical problems efficiently solvable on quantum computers, but not on classical ones. (For readers already familiar with quantum computing, this introduction section or even this entire chapter may be skipped.)

After the introduction, we review the key algorithms employed in the simulation of quantum materials, specifically focusing on algorithms for Hamiltonian simulation and eigenvalue spectral estimation. We then highlight open research areas (particularly in the simulation of open quantum systems) and provide a summary of the contributions to the field made by the work presented in this dissertation.

6.1.1 Background of Quantum Computing

6.1.1.1 Qubits and quantum gates. The quantum computer is an emerging computational paradigm that exploits coherent superposition, entanglement, and

contextuality to process information in a Hilbert space whose dimension scales exponentially with the number of qubits [127]. Unlike classical computation, which manipulates bit strings in $\{0,1\}^n$, quantum computation operates on wave function state vectors in $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ with time-evolution generated by controllable Hamiltonians. Typically, each subspace $\mathbb{C}^2$ corresponds to a *qubit* (a quantum bit), which encodes a quantum superposition of orthogonal 0 and 1 classical bit states. Qubit wave functions are subject to the same normalization conditions as standard quantum wavefunctions, and are represented with the Dirac notation

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{6.1}$$

where $\alpha, \beta \in \mathbb{C}$ with $|\alpha|^2 + |\beta|^2 = 1$. Often, it is easier to represent individual quantum states ($|\psi\rangle$ and qubit basis states ($|0\rangle, |1\rangle$) as vectors in the computational basis

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{6.2}$$

so that

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \tag{6.3}$$

Quantum operators (represented as matrices) act linearly on these vectors. For example, the *Pauli operators* are represented as

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{6.4}$$

In a quantum computer, combining n qubits together yields the exponentially-growing tensor-product space

$$\mathcal{H}_n = (\mathbb{C}^2)^{\otimes n}, \tag{6.5}$$

whose dimension is 2^n. Basis states are written as

$$|x\rangle = |x_1\rangle \otimes \cdots \otimes |x_n\rangle = |x_1 x_2 \cdots x_n\rangle\,, \qquad x_i \in \{0,1\}. \tag{6.6}$$

An arbitrary state therefore takes the form

$$|\Psi\rangle = \sum_{x\in\{0,1\}^n} c_x\,|x\rangle\,, \qquad \sum_x |c_x|^2 = 1. \tag{6.7}$$

When a quantum computation is carried out on a quantum computer an initial state $|\psi(t)\rangle$ at a time $t = 0$ is manipulated through the time-evolution operator $\hat{U}(t)$:

$$|\psi(t)\rangle = \hat{U(t)}\,|\psi(0)\rangle\,. \tag{6.8}$$

Specifically, $\hat{U}(t)$ is generated by carefully controlling the qubit state with a time-varying Hamiltonian $\hat{H}(t)$. The time-evolution according to the Schrodinger equation is:

$$\hat{U}(t) = \exp\left(-\frac{i}{\hbar}\int_0^t \hat{H}(t)\ dt\right) \tag{6.9}$$

An important property of $\hat{U}(t)$ is that is *unitary*, meaning it implements a reversible and orthonormal operation ($I = \hat{U}^\dagger\hat{U} = \hat{U}\hat{U}^\dagger$). The set of controllable unitary operations performable on a set of qubits are called *quantum gates*, and they are analogous to the logical gates (e.g., AND, OR, NOT, etc.) performable on a classical computer. For example. since the Pauli operators (6.4) are unitary, they are realizable as single-qubit quantum gates, called *Pauli gates*. The Pauli X gate, corresponds to the classical NOT gate, but the Pauli Y and Z operations correspond to quantum phase flips not producible on classical bits. Another example of a single-qubit gate is the *Hadamard operator* H, which prepares an equal superposition of the $|0\rangle$ and $|1\rangle$ states, or the paramererized *phase shift operator* $P(\theta)$, which induces a complex

phase $\theta \in [0, 2\pi)$ on the $|1\rangle$ state:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} \tag{6.10}$$

The quantum gates (6.4) and (6.10), when combined together in sequences, are sufficient for realizing all unitary single-qubit operations, as their matrix representations generate the 2-dimensional *special unitary group* $SU(2)$.

In addition to single-qubit gates, multi-qubit gates are essential for creating correlated states on a quantum computer that exhibit *quantum entanglement* between qubits. The canonical two-qubit gate used in quantum information is the *conditional-NOT* gate C_X, which conditionally performs a Pauli X gate (the quantum NOT) to a target qubit when a control qubit is in the $|1\rangle$ state. A C_X targeting the rightmost qubit (i.e., controlled by the leftmost qubit) in a two-qubit Hilbert space has the following matrix representation and action on the two-qubit basis:

$$C_X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \Rightarrow \quad \begin{cases} C_X |00\rangle = |00\rangle \\ C_X |01\rangle = |01\rangle \\ C_X |10\rangle = |11\rangle \\ C_X |11\rangle = |10\rangle \end{cases} \tag{6.11}$$

Remarkably, by combining the single-qubit gates (6.4) and (6.10) along with the two-qubit C_x gate (6.11) any arbitrary unitary in $SU(2^n)$ on n qubits can be implemented exactly in a finite number of at most $O(2^n)$ gates. Much like how the classical NAND gate is sufficient to universally compute any function $f : 0, 1^n \to \{0, 1\}^n$, the *universal quantum gate set* $\{X, Y, Z, H, P(\theta), C_x\}$ is sufficient to realize any unitary of

the form (6.9) corresponding to the time evolution of a quantum-mechanical system. Indeed, this potential for the universal simulation of quantum mechanical systems was one of the first proposed advantages of quantum computation [128].

Furthermore, it has even been shown that if the gates in any universal gate set can only be natively executed on a quantum computer up to a precision of at most $\delta\theta = \pm\pi/8$, one can still accurately approximate a quantum gate sequence of consisting of any sequence of m constant-qubit gates from another universal gate set to a desired precision ε using a sequence of $\mathcal{O}(m \log^c(m/\varepsilon))$ natively executable gates. This result is known as the *Solovay-Kitaev Theorem* [129, 130]: put simply, it means that one can not only "transpile" sequences of any universal gate set into another set of universal gates executable on another quantum computer, but do so efficiently.

For this reason, quantum computation is capable of universally simulating exponentially large quantum states using a finite number of qubits. It is an especially promising technology for simulating exponentially large quantum mechanical systems (such as molecules and materials), for which classical simulations would run out of memory. While most modern quantum computers, support the universal gate-based model of quantum computation, is also worth mentioning that some computational devices do not support the traditional gate-based model. For example, quantum annealing processors support thousands to tens of thousands of qubits, but are only capable of finding approximate solutions to structured optimization problems. Similarly, quantum photonic processors have recently emerged as another potential quantum quantum platform, but are not as effectively universal due to the weakness of photon interactions and non-linearities present in entanglement gates. These issues can be circumvented through various polynomial-overhead entanglement preparation

schemes, such as *Knill-Laflamme-Milburn* (KLM) [131], though significant engineering challenges persist that make such schemes difficult to realize in practice [132].

While we acknowledge that these alternative platforms are emerging as viable computational platforms, we will focus almost exclusively on the standard gate-based model of quantum computation in the remainder of this dissertation.

6.1.1.2 Measurement. While quantum computers are capable of preparing exponentially large quantum states that are intractable on classical computers, only a limited amount of information can be learned about a computed quantum wave function $|\psi(t)\rangle$. Indeed, a state prepared on a number of qubits can be *measured* on a quantum computer, resulting in the wave function spontaneously and randomly "collapsing" to one of each possible classical bit string state $|x_i\rangle$ with probability $P(x_i) = |c_{x_i}|^2$. Even though the computed distribution $P(x_i)$ may be exponentially complex, one can only produce one bit-string sample x_i from this wavefunction at the end of a computation.

Consequently, quantum computation is an inherently random process; however, for certain structured computational problems, one can use quantum gates to prepare a distribution $P(x_i)$ where the "correct" answer is measured with high probability, or where through multiple subsequent measurements, one can obtain useful information about the final wave function $|\psi(t)\rangle$ without knowing it in its entirety.

Thus, the process of quantum computation can be broken down into four stages stages: (1) State initialization, (2) quantum computation, (3) measurement, (4) post-processing of measured results. In many cases, the measurements processed in stage (4) are accumulated from multiple measurements obtained by iterations of

stages (1)-(3). In the literature, each iteration of stages (1)-(3) is referred to as a *shot*. We briefly discuss each of these stages below:

(1) *State initialization*: Canonically, all qubits on the device are initialized to the $|0\rangle$ state in the system, such that $|\psi(0)\rangle = |000\cdots0\rangle$. On modern *noisy intermediate-scale quantum* (NISQ) computers, state preparation requires careful calibration, and can be an imperfect process, requiring techniques like *error mitigation* to counteract the effects of state preparation noise [133].

(2) *Quantum Computation*: In this stage, a sequence of quantum gates native to the quantum computer is applied, analogous to the logical gates in classical computation, preparing a final quantum distribution $|\psi(t)\rangle$. On modern NISQ quantum hardware, each gate induces a small amount of noise into the wavefunction, limiting the number of operations that can be performed before the detrimental effects of noise dominate. This has motivated significant research efforts toward developing practical algorithms suitable for NISQ devices [16].

(3) *Measurement*: The wave function is measured, resulting in wavefunction collapse, producing a qubit bitstring x_i with probability $P(x_i) = |c_{x_i}|^2$. In some cases it common to estimate the expectation value of some Hermitian observable O. This can be achieved by performing multiple measurements in different Pauli bases. This is achieved by applying H and $P(\theta)$ transformation prior to measuring each qubit, a process that we will discuss in later sections.

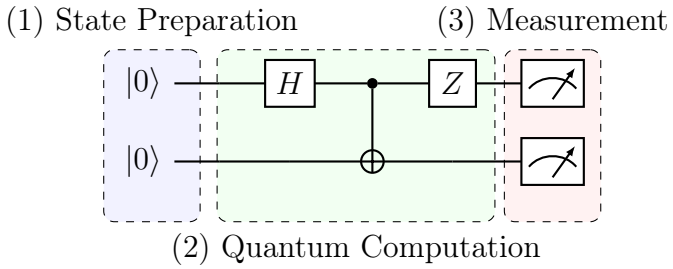

Figure 6.1: Basic quantum circuit diagram structure (read left-to-right) illustrating state initialization, quantum computation, and measurement. The blocks H and Z indicate Hadamard and Pauli Z gates. The two-qubit gate "$\bullet\!-\!\oplus$" denotes a C_x gate with control qubit "$\bullet$" and target qubit "$\oplus$".

(4) *Post-Processing*: The results of each measurement are combined through post-processing on a classical computer to produce an estimate of the desired outcome of the computation (e.g., the ground state energy E_0 of a physical system). In most situations, one can obtain as precise an estimate of an observable as desired by carefully adjusting the number of repeated shots per measurement taken.

The steps (1)-(3) in a quantum computation can be succinctly summarized through a *quantum circuit diagram*, such as the one shown in Figure 6.1.

6.1.2 Fundamentals of Quantum Computation

Quantum computation derives its power from three distinctly non-classical features of quantum mechanics: (1) superposition, (2) entanglement, and (3) contextuality. Each reflects a structural property of Hilbert space and operator algebras, and together they underpin quantum computational advantage. In this section, we briefly review these three foundational concepts and provide an overview of each basic principle:

(1) *Superposition*: The principle of superposition follows from the linearity of quantum mechanics. If $|\psi\rangle$ and $|\phi\rangle$ are valid states in a Hilbert space $\mathcal{H}$, then any normalized linear combination

$$|\chi\rangle = \alpha\,|\psi\rangle + \beta\,|\phi\rangle\,, \qquad |\alpha|^2 + |\beta|^2 = 1, \tag{6.12}$$

is also a valid state. For a single qubit,

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{6.13}$$

with $|0\rangle = (1\ 0)^T$ and $|1\rangle = (0\ 1)^T$ in the computational basis.

For an n-qubit system, the tensor-product structure yields

$$|\Psi\rangle = \sum_{x\in\{0,1\}^n} c_x\,|x\rangle\,, \tag{6.14}$$

where $|x\rangle$ labels computational basis states and $\sum_x |c_x|^2 = 1$. The dimension of this space grows as 2^n, allowing coherent amplitude encoding across exponentially many configurations.

Crucially, computational advantage does not arise from parallel evaluation alone, but from interference. Under unitary evolution, amplitudes combine constructively or destructively:

$$|\Psi'\rangle = U\,|\Psi\rangle\,. \tag{6.15}$$

Algorithms such as phase estimation and amplitude amplification (discussed in the subsequent sections) exploit structured interference to concentrate

probability weight onto desired outcomes while suppressing others. Superposition therefore enables a form of global amplitude processing that has no classical analogue.

(2) *Entanglement*: Quantum entanglement characterizes correlations that cannot be described by product states. A bipartite state $|\psi\rangle_{AB}$ is separable if it can be written as

$$|\psi\rangle_{AB} = |\phi\rangle_A \otimes |\varphi\rangle_B . \tag{6.16}$$

Otherwise, it is entangled. For example, the state prepared prior to measurement in Figure 6.1 is:

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}} \left(|00\rangle - |11\rangle\right) . \tag{6.17}$$

This state is entangled, because no single-qubit factorization exists. In many cases, quantum entanglement reflects the non-factorizability of the global wavefunction with respect to subsystem partitions. Operationally, it produces correlations that violate *Bell inequalities* and cannot be reproduced by local hidden-variable models [134,135]. In many-body systems, entanglement structure governs computational complexity: states obeying area laws admit efficient tensor-network representations, whereas volume-law entangled states generally do not.

From a computational perspective, entanglement enables nonlocal correlation processing across qubits. Universal gate sets combine single-qubit rotations with entangling gates (e.g., C_x) precisely because entanglement is necessary to generate the full unitary group on $(\mathbb{C}^2)^{\otimes n}$. Without entangling operations,

quantum circuits reduce to independent single-qubit evolutions and are efficiently classically simulable.

(3) *Contextuality*: Contextuality formalizes a deeper nonclassical feature: the outcome of a measurement cannot, in general, be assigned independently of the measurement context. More precisely, there exists no assignment of predetermined values to all observables that preserves functional relationships between commuting operators. This is captured by the *Kochen-Specker theorem* [136].

In computational terms, contextuality arises from the noncommutativity of operator algebras. If A and B do not commute, $[A, B] \neq 0$, then measurement outcomes depend on the order and grouping of measurements. This structure prevents a global classical description in terms of joint probability distributions over all observables.

Recent resource-theoretic formulations identify contextuality as a necessary ingredient for both polynomial and exponential quantum computational speedup in certain models. For example, the set of quantum *Clifford circuits* that act on stabilizer subspaces of quantum states admit efficient classical simulation. In terms of the quantum gates (6.4),(6.10), and (6.11) introduced above, Clifford circuits employ only the limited gate set

$$\mathcal{G}_{\text{Clifford}} = \{H, X, Y, Z, C_x, P(\pi/2)\} \tag{6.18}$$

and satisfy the the criterion that for any tensor product of Pauli gates $P \in \{X, Y, Z\}^{\otimes n}$ (sometimes called a *Pauli string*), $GPG^\dagger \in \{X, Y, Z\}^{\otimes n}$ up to a

complex phase for all $G \in \mathcal{G}_{\text{Clifford}}$. The result that any such circuit consisting of gates in $\mathcal{G}_{\text{Clifford}}$ can be efficiently simulated on a classical computer is known as the *Gottesman–Knill Theorem* [137].

the addition of non-Clifford small-phase-angle operations $P(\theta)$ for $\theta \in (0, \pi/4]$ introduces contextuality and renders the model universal. Thus contextuality distinguishes classically simulable subtheories from those capable of universal quantum computation. These small-angle gates are essential for operations such as the quantum Fourier transform (discussed in the next section), which is an essential subroutine in all known quantum algorithms that provide an exponential advantage over their classical counterparts.

Superposition enables coherent amplitude encoding, entanglement distributes correlations across subsystems, and contextuality precludes a consistent classical hidden-variable description of measurement statistics. Together, these principles expand the accessible state space and dynamical transformations beyond classical probabilistic computation.

A universal gate set consisting of arbitrary single-qubit rotations together with an entangling gate such as C_x gates generates a dense subset of $SU(2^n)$. (As mentioned earlier, the Solovay-Kitaev theorem guarantees efficient approximate transpilation to any other such universal gate set.) Within this framework, quantum algorithms engineer interference patterns across entangled states in a way that cannot be efficiently reproduced by classical stochastic processes.

These structural features of quantum computation (linearity, tensor-product composition, and noncommutative operator algebras) form the mathematical foundation of quantum computational advantage, which is especially relevant in the case of

simulating quantum materials an a level of near-exact physical precision. In the next section we review basic concepts in *Quantum Complexity Theory*, which seeks to categorize computational problems and classify them based on whether or not quantum computers can simulate them with reduced resource requirements (time, space, etc.) relative to classical computers.

6.1.3 Quantum Complexity Theory

From a complexity-theoretic standpoint, the class BQP (Bounded-Error Quantum Polynomial Time) contains problems believed to be intractable for classical computers, including discrete logarithms and integer factorization via Shor's algorithm [138]. Of more direct relevance to condensed matter physics and materials science, quantum computers are also naturally suited to simulate quantum many-body systems because their state space matches the tensor-product structure of interacting quantum degrees of freedom [139]. Here we give a brief survey of these classes, and aim to provide basic intuition regarding the kinds of computational problems they include.

6.1.3.1 Fundamental Complexity Classes: P, NP, BPP, BQP, and QMA.

Computational complexity theory studies the intrinsic difficulty of computational problems by classifying them according to the resources required to solve them. The most central resource is *time*– specifically, how the number of elementary computational steps scales with the size of the input. Problems whose solution time grows at most polynomially with input size are regarded as *efficiently solvable*. This notion gives rise to several foundational *time-complexity* classes, which are introduced below:

(1) *Deterministic Polynomial Time* (P):

The class P consists of all decision problems solvable in polynomial time by a deterministic Turing machine. Informally, these are problems for which a classical computer can produce a definite yes/no answer in time scaling as n^k for some fixed constant k, where n is the input size.

Examples include graph connectivity and shortest-path problems. The class P is widely regarded as capturing the notion of "efficient classical computation."

(2) *Nondeterministic Polynomial Time* (NP):

The class NP contains decision problems for which a proposed solution can be verified in polynomial time by a deterministic classical computer. Equivalently, NP consists of problems solvable in polynomial time by a nondeterministic computer (one that can "fork" into any number of independent but parallel branches of computation simultaneously).

The key conceptual distinction here is between "finding" a solution and "verifying" one. For example, given a candidate solution to a Boolean satisfiability instance, verifying that it satisfies all clauses is efficient, even though no efficient algorithm is known for finding such a solution in general. The famous P versus NP problem asks whether every efficiently verifiable problem is also efficiently solvable—that is, whether $P = NP$. It is widely conjectured that $P \neq NP$, though this remains unproven [140].

(3) *Bounded-Error Probabilistic Polynomial Time* (BPP):

The class BPP extends P by allowing classical algorithms to use randomness. A problem is in BPP if it can be solved in polynomial time by a probabilistic

Turing machine that produces the correct answer with probability at least 2/3 for every input. The specific constant 2/3 is arbitrary; any constant strictly greater than 1/2 yields an equivalent class, since independent repetitions reduce the error probability exponentially. It is widely believed that randomness does not increase the power of efficient classical computation in an essential way, and that $P = BPP$, though this equality has not been proven [141].

(4) *Bounded-Error Quantum Polynomial Time*(BQP):

The class BQP is the quantum analogue of BPP. It consists of decision problems solvable in polynomial time by a quantum computer with bounded error probability. Formally, a problem lies in BQP if there exists a uniform family of polynomial-size quantum circuits that outputs the correct answer with probability at least 2/3.

Quantum computation differs fundamentally from classical probabilistic computation due to superposition, entanglement, and interference. These features enable algorithms such as Shor's factoring algorithm [142] and Grover's search algorithm [143], which demonstrate provable speedups over the best known classical methods for certain problems. The relationship between BQP and NP remains unknown. In particular, it is not known whether quantum computers can efficiently solve NP-complete problems, through it is generally believed that quantum computers cannot efficiently solve all problems in NP, and that also there exist certain problems in BQP that are harder than NP [144].

(5) *Quantum Merlin-Arthur* (QMA):

The class QMA is the quantum generalization of NP. In NP, a classical prover ("Merlin") supplies a classical witness that a deterministic verifier checks efficiently. In QMA, Merlin provides a *quantum* state as a witness, and a polynomial-time quantum verifier ("Arthur") checks it with bounded error probability. Thus, QMA captures problems for which a quantum proof can be efficiently verified by a quantum computer. Many natural problems in quantum many-body physics (such as determining ground state energies of general Hamiltonians from scratch) are known to be complete for QMA [145, 146].

The containments

$$P \subseteq BPP \subseteq BQP \subseteq QMA \quad \text{and} \quad NP \subseteq QMA \tag{6.19}$$

are known, though several of these inclusions are believed to be strict. As discussed above, the precise relationships among BQP, NP, and QMA still remain central open questions in complexity theory. The literature consensus regarding the relationships between these classes is visualized in Figure 6.2.

Together, these classes delineate the landscape of efficient classical computation, efficient randomized computation, efficient quantum computation, and efficiently verifiable classical and quantum proofs. Understanding their relationships is fundamental to assessing the true power of quantum information processing. In Table 6.1, we summarize each of these complexity classes, giving examples of canonical computational problems contained in each.

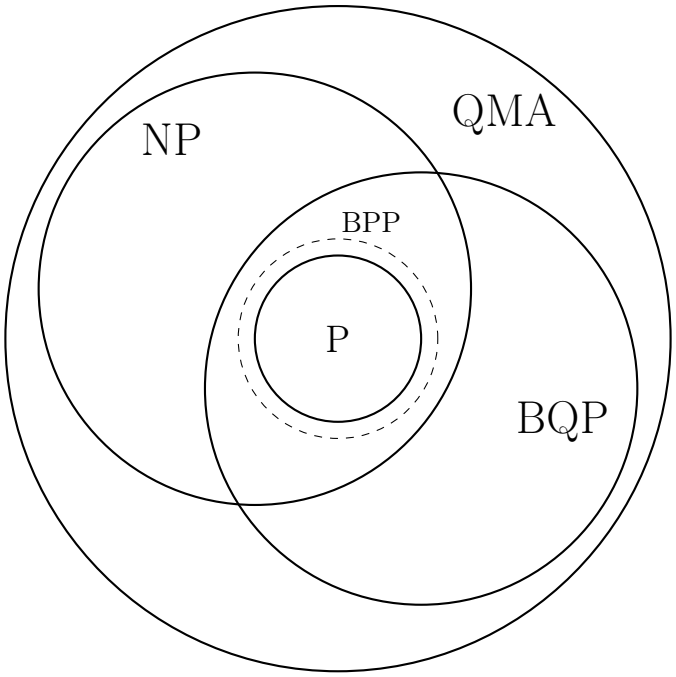

Figure 6.2: Common intuition regarding the relationships between the complexity classes P, BPP, BQP, NP, and QMA. It is widely conjectured (though unproven) that $P = BPP$. Both NP and BQP are known to be contained in QMA, while the precise relationship between BQP and NP remains unknown: it is not known whether either class contains the other or whether they are distinct. The question of whether $P = NP$ remains one of the most significant open problems in theoretical computer science, and it is widely believed that $P \neq NP$. Quantum computational advantage concerns the relationship between BQP and classical efficient computation (e.g., P or BPP), and does not depend on the resolution of the P versus NP question.

6.2 Quantum Computing for Quantum Material Simulation

In this section, we steer our discussion from general quantum computation to specific case of simulating physical quantum systems. We recall that for interacting electronic systems described by Hamiltonians of the form

$$H = \sum_{ij} h_{pq} \hat{a}_i^\dagger \hat{a}_j + \frac{1}{2} \sum_{ijkl} V_{ijkl} \hat{a}_i^\dagger \hat{a}_j^\dagger \hat{a}_k \hat{a}_l, \tag{6.20}$$

classical exact simulation scales exponentially in system size due to Hilbert space growth and entanglement structure. Tensor network approaches provide efficient

Table 6.1: Summary of selected classical and quantum complexity classes.

Class	Random	Full Name	Definition	Examples of Problems
P	✗	Polynomial Time	Deterministic classical algorithm running in time $O(n^k)$ for some constant k.	Primality testing, classical linear algebra operations.
BPP	✓	Bounded-Error Probabilistic Polynomial Time	Polynomial-time probabilistic classical algorithm accepting YES instances with probability $> 1/2$.	Randomized counting problems (e.g., #P-style comparisons of the number of solutions to problems in P), sampling problems.
NP	✗	Nondeterministic Polynomial Time	Solutions verifiable in polynomial time by a deterministic classical computer given a classical witness.	Unstructured optimization problems, brute force search.
BQP	✓	Bounded-Error Quantum Polynomial Time	Polynomial-time quantum algorithm with error probability $\leq 1/3$ (amplifiable).	Integer factoring, discrete logarithms, period-finding algorithms.
QMA	✓	Quantum Merlin–Arthur	Quantum analogue of NP: polynomial-time quantum verifier with access to a polynomial-size quantum witness.	Quantum optimization problems (e.g., finding the ground state of non-local Hamiltonians).

representations only when entanglement obeys area laws. Quantum hardware, by contrast, encodes fermionic Fock space into qubit registers via transformations that directly map creation annihilation operators ($\hat{a}^\dagger$ and $\hat{a}$) to multi-qubit gates and observables. When Hamiltonians such as (6.20) satisfy certain sparsity and locality constraints, they admit polynomial-depth circuits for both time evolution and eigenvalue estimation. In summary, quantum algorithms for materials simulation promise

polynomial to sub-exponential scaling for many important problems, such as ground-state energy estimation, spectral function estimation, and dynamical response, all of which are quantities central to the study of correlated electron systems and quantum materials.

6.2.1 Complexity of Electronic Structure Methods

Electronic structure theory seeks to determine properties of many-electron quantum systems, most notably ground-state energies and correlation functions. At its core lies the many-body Schrödinger equation, whose Hilbert space dimension grows exponentially with system size. This exponential growth underlies the computational difficulty of the problem.

From a complexity-theoretic standpoint, one typically considers the following decision problem: given a Hamiltonian H and two real numbers $a < b$ (separated by an inverse-polynomial promise gap), determine whether the ground-state energy E_0 satisfies $E_0 \leq a$ or $E_0 \geq b$. For general quantum Hamiltonians with local interactions, this problem is complete for the class QMA, the quantum analogue of NP [147]. Thus, unless $QMA = BQP$, no efficient quantum algorithm exists for solving arbitrary instances of the electronic structure problem.

This result is often contrasted with classical spin systems. For example, determining the ground state of certain classical Ising spin glasses is NP-complete, meaning that while a proposed configuration can be verified efficiently, no polynomial-time algorithm is known to find one in general [148]. The quantum generalization elevates the difficulty from NP to QMA, reflecting the added complexity of quantum superposition and entanglement.

Practical electronic structure methods navigate this worst-case hardness by imposing structure or approximation. Mean-field methods such as Hartree-Fock [12] and Kohn-Sham density functional theory [108] reduce the many-body problem to self-consistent single-particle equations that can be solved in polynomial time for fixed basis sets, though convergence guarantees and accuracy are not universal. Tensor-network approaches such as density matrix renormalization group (DMRG) methods exploit area-law entanglement structure and are efficient for one-dimensional gapped systems, but their cost scales exponentially in higher-dimensional or highly entangled regimes [13]. Finally, methods such as full configuration-interaction (Full CI) and complete active space self-consistent field (CASSCF) methods exactly solve the electronic structure for the full Fock space, or a carefully selected subspace thereof, which requires exponentially growing computational resources [15].

In addition to deterministic classical algorithms, randomized classical algorithms, including quantum Monte Carlo [27], can approximate expectation values in polynomial time per sample and thus resemble algorithms in BPP for sign-problem-free systems. However, the fermionic sign problem generically induces exponential variance, preventing worst-case polynomial guarantees for systems with complex electronic structure [28].

Quantum algorithms such as Hamiltonian simulation and phase estimation (discussed in the subsequent sections) lie in BQP and provide exponential speedups for certain spectral estimation tasks, such as factoring integers and solving discrete logarithms. Nevertheless, because the general local Hamiltonian problem is QMA-complete, these algorithms do not always circumvent worst-case complexity barriers without additional physical structure or promises. However, it has been shown that if

one employs an efficient classical method (e.g., Hartree-Fock) to produce an approximate ground state $|\Phi_0\rangle$ of a sparse local Hamiltonian with the promise of satisfying $\langle\Phi_0|\Psi_0\rangle < 1/2$, then quantum phase estimation can be used to solve the ground state decision problem in polynomial time [149].

To conclude, the conditions under which *quantum advantage* (a strong exponential speedup over classical computational methods) can be achieved in electronic structure problems are highly dependent on the structure of the physical Hamiltonian of interest. Under standard physical assumptions such as locality of strong interactions and that quantum computations are initialized with a sufficiently close "guess" of the ground state electronic structure, it is widely accepted that practical quantum advantage can be realized for most physical systems [139]. In Table 6.2, we summarize the known computational complexity calculations of various popular electronic structure methods, both classical an quantum. The methods contained in the class BQP capture these physical systems for which advantage is achievable.

The hierarchies of the computational complexity classes shown in Figure 6.2 emphasize a central message: the computational intractability of electronic structure is not merely a limitation of classical algorithms or electronic structure solvers, but it instead emerges in the formal complexity of many-body quantum systems themselves. While solving for the electronic structure of arbitrary complex Hamiltonians is still exponentially difficult, even on quantum computers, the combination of classical methods and quantum methods can likely produce an exponential quantum speedup for most systems, such as molecules and solid state materials.

Table 6.2: Computational Complexity classes of decision-problem formulations of popular electronic structure methods for quantum chemistry and materials science.

Class	Time to Solve (ground state energy)	Time to Verify (ground state energy)	Electronic Ground State Methods
P	$\mathcal{O}(n^k)$ (classical)	$\mathcal{O}(n^k)$ (classical)	Hartree-Fock, Mean-field Theories, DFT
BPP	$\mathcal{O}(n^k)$ (classical, $p <$ 1/3 of failure)	$\mathcal{O}(n^k)$ (classical)	Sign-Problem-Free Quantum Monte-Carlo methods.
NP	$2^{\mathcal{O}(n)}$ (classical)	$\mathcal{O}(n^k)$ (classical)	Full CI, CASSCF, DMRG
BQP	$\mathcal{O}(n^k)$ (quantum, $p <$ 1/3 of failure)	$\mathcal{O}(n^k)$ (classical)	Quantum Eigensolvers (with phase estimation and classical initialization)
QMA	$2^{\mathcal{O}(n)}$ (quantum, $p <$ 1/3 of failure)	$\mathcal{O}(n^k)$ (quantum, $p <$ 1/3 of failure)	VQE, Quantum Eigensolvers (no initialization)

6.2.2 Quantum Algorithms for Electronic Structure

Quantum simulation aims to compute properties of quantum many-body systems using a controllable quantum device whose Hilbert space scales exponentially with the number of qubits. In electronic structure and condensed matter physics, the central object of interest is the many-body Hamiltonian (6.20).

For quantum chemistry in a basis of N spin orbitals, the number of terms scales as $\mathcal{O}(N^4)$ in the worst case, though symmetry and integral screening reduce this in practice. Quantum algorithms for such Hamiltonians fall into three primary categories:

(1) Variational quantum algorithms (optimized for modern noisy quantum hardware)

(2) Hamiltonian simulation (implementing e^{-iHt} to evolve a prepared wave function $|\psi\rangle$),

(3) Eigenvalue and spectral estimation (extracting E_k from H; efficient, but difficult to implement on modern noisy hardware).

These categories represent a trade-off between being realizable on modern quantum hardware and being computationally efficient. For example, variational quantum eigensolver (VQE) methods have been identified as the flagship application of near-term noisy quantum hardware [150]; however, to attain a true quantum computational advantage on future fault-tolerant hardware, methods like quantum phase estimation will likely become standard.

6.2.2.1 Mapping Hamiltonians to Qubit Operators. Before reviewing how the three methods enumerated above contribute to quantum computational advantage for chemistry problems, we must review how fermionic Hamiltonians can be mapped to both observables and operators that can be applied and measured on quantum computers. We recall that the general form of a second-quantized electronic Hamiltonian can be written in the form (6.20), consisting of sums of products of fermionic creation and annihilation operators $\hat{a}^\dagger, \hat{a}$. These operators can be represented in many ways, provided they satisfy the anti-commutative canonical commutation relations

$$\{\hat{a}_i, \hat{a}_j^\dagger\} = \{\hat{a}_i^\dagger, \hat{a}_j\} = \delta_{ij}, \qquad \{\hat{a}_i, \hat{a}_j\} = \{\hat{a}_i^\dagger, \hat{a}_j^\dagger\} = 0 \tag{6.21}$$

and that one can define a unique vacuum state $|0\rangle$ satisfying $\hat{a}\,|0\rangle = 0$.

The most intuitive ways of representing a set of N $\hat{a}_i$ annihilation operators is through a direct 1-1 mapping of the full Fock space occupation number basis, that is:

$$|n_1, n_2, ...n_N\rangle \quad \Leftrightarrow \quad |n_1 n_2 ... n_N\rangle\,, \tag{6.22}$$

so that the zero state $|00...0\rangle$ is the vacuum state, and when any qubit reads $|1\rangle$, it corresponds to an electron occupying the respective mode. Then, to create and annihilate an electron from the vacuum state, one can apply the σ^+ and σ^- operators

$$\sigma^+ = \begin{pmatrix} 0 & 1 \\ & \\ 0 & 0 \end{pmatrix}, \qquad \sigma^- = (\sigma^+)^\dagger = \begin{pmatrix} 0 & 0 \\ & \\ 1 & 0 \end{pmatrix} \tag{6.23}$$

to the coresponding qubit. However, these operators alone are fully commutative instead of anti-commutative. To ensure that the CCRs (6.21) are completely satisfied, we must employ a phase encoding of the of the operators to force them to anti-commute. The simplest such encoding is the *Jordan-Wigner mapping* [151], defined as follows:

$$\begin{aligned} \hat{a}_i &\quad \leftrightarrow \quad \sigma_i^- \otimes \left(\bigotimes_{n=1}^{i-1} Z_n \right), \\ \hat{a}_i^\dagger &\quad \leftrightarrow \quad \sigma_i^+ \otimes \left(\bigotimes_{n=1}^{i-1} Z_n \right) \end{aligned} \tag{6.24}$$

These operators, while non-unitary, can be written as a linear combination of two unitary operations, each consisting of tensor products of the form

$$P_j = \sigma_1 \otimes \sigma_2 \otimes .. \otimes \sigma_n \tag{6.25}$$

where each $\sigma_n \in \{I, X, Y, Z\}$ is either one of the Pauli operators in (6.4) or the identity operator I. These composite operators P_i are called *Pauli strings*. Under the Jordan Wigner mapping, one obtains:

$$\hat{a}_j = \frac{1}{2} P_j^{(X)} + \frac{i}{2} P_j^{(Y)} = \frac{1}{2} \left[X_i \otimes \left(\bigotimes_{n=1}^{i-1} Z_n \right) \right] + \frac{i}{2} \left[Y_i \otimes \left(\bigotimes_{n=1}^{i-1} Z_n \right) \right] \tag{6.26}$$

In addition to Jordan-Wigner, there exist other encodings, such as parity mappings, or the slightly more efficient Bravyi-Kitaev mapping [152,153], which integrates aspects of both the parity and Jordan-Wigner mappings. These methods all admit representations of $\hat{a}_i$ in terms of a weighted sum of Pauli strings. Since the product of

any number of Pauli strings is also a Pauli string, it follows that one can represent a Hamiltonian consisting of products $\hat{a}^\dagger, \hat{a}$ operators as products of linear combinations of Pauli operators, which expand into a larger linear combination of Pauli strings. Thus, any Hamiltonian can be represented in the form

$$\hat{H} = \sum_j c_j P_j. \tag{6.27}$$

The Pauli strings P_j can each be estimated as expectation values of a wave function $|\psi(t)\rangle$ computed from some quantum circuit by transforming the wave function into a basis that diagonalizes P_j (requiring at most a single H and $P(\theta)$ gate) and then taking a linear combinations of measured $|0\rangle$ and $|1\rangle$ probabilities from multiple computational shots. Alternatively, the P_j can be implemented as quantum gates via sequences of single-qubit Pauli operations.

6.2.2.2 Variational Quantum Algorithms. Variational algorithms are popular, due to the fact that they require relatively shallow circuits, and are therefore ideal for execution on near-term NISQ hardware. For tasks such as computing the ground state electronic structure of molecules and certain quantum spin lattice structures, the *variational quantum eigensolver* method is by far the most popular, and is believed to be one of the few quantum algorithms that offer some near-term practical quantum utility. In this section, we give a brief review of VQE and its implementation.

When compiling quantum circuits for VQE, one typically assumes that fermionic operators have already been mapped to linear combinations of Pauli operators in the form (6.27). This allows for one to efficiently estimate the expectation value of a Hamiltonian wave function $|\psi\rangle$ represented on th The key quantum utility of VQE

is the quantum computation of a carefully parameterized wave function $|\psi(\theta)\rangle$ (having parameters θ) that closely approximates the true ground state of a quantum Hamiltonian using a full Fock-space representation of ψ that would otherwise require exponential growth in memory to represent on a classical computer.

The goal of VQE is to approximate the ground state energy $E_0 = \min_{\psi} \langle\psi| \hat{H} |\psi\rangle$. VQE accomplishes this through a *variational ansatz*, which is a parameterized quantum circuit that prepares a the trial state $|\psi(\theta)\rangle = U(\theta)|0\rangle$. In many cases, the ansatz is structured to preserve the symmetries and average particle count so that the expected number of electrons are included in the Hamiltonian; however, *hardware-efficient ansätze* have also been developed to exploit the gates that are efficiently executable on NISQ hardware. These ansätze are a popular choice for certain modern chemistry problems [154].

The VQE algorithm works by repeatedly evaluating the total energy $E_{\text{total}} = \langle\psi(\theta)| \hat{H} |\psi(\theta)\rangle$ and iteratively optimizing the parameters θ using a classical optimizer until E_{total} converges. This closed optimization loop is summarized in Figure 6.3. Often, to ensure the algorithm converges quickly, the quantum wave function is initialized to match the approximate wave function computable in polynomial time via classical mean-field methods like Hartree-Fock or DFT. This initializes the wave function as a product of single-qubit states

$$|\Phi\rangle = |\Phi_1\rangle \otimes |\Phi_2\rangle \otimes ... \otimes |\Phi_N\rangle \tag{6.28}$$

which can be prepared efficiently from the vacuum state $|00...0\rangle$ using only single-qubit gates.

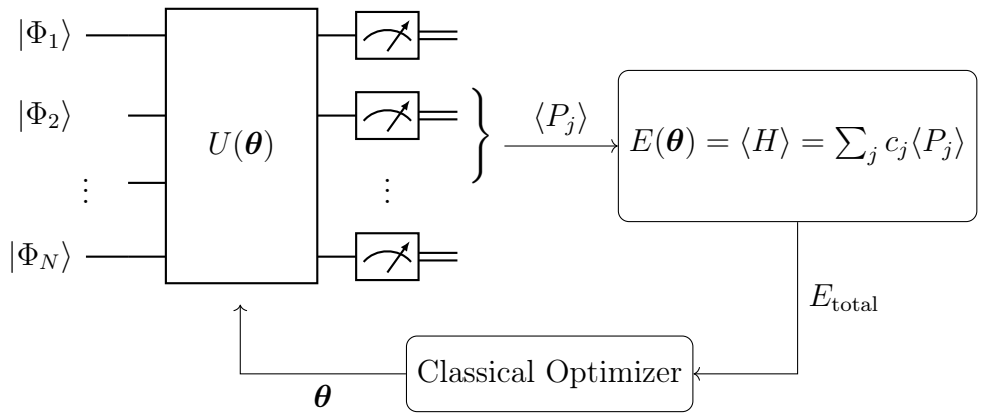

Figure 6.3: Optimization loop of the variational quantum eigensolver (VQE) method initialized with the mean-field approximate state $|\Phi\rangle$.

Although we will not give a detailed theoretical discussion of how ansätze are designed, it is worth remarking that the success of the VQE method hinges largely on the expressiveness and efficiency of the chosen ansatz. Chemically motivated ansätze like UCCSD (unitary coupled cluster with single/double excitations) [155] embed known structure of electronic wavefunctions into the quantum circuit, often yielding significantly improved convergence relative to purely hardware-driven parameterizations.

This problem of slow convergence is also seen more broadly in the landscape of variational quantum algorithms, and is known as the *barren plateau* problem. The barren plateau problem occurs when the energy (or variational "cost function") landscape of a variational quantum circuit becomes extremely flat. In such cases, changing a parameter produces almost no measurable change in the energy, so the optimizer has little information about which direction will lower the energy further, even if quantum gradient-based optimizers are used. As the number of qubits or the depth of the circuit increases, this flatness can become more severe: the gradients of the

energy with respect to the parameters become exponentially small. In practice this means that a very large number of measurements would be required just to determine whether a small change in a parameter increases or decreases the energy.

One way to understand this behavior is to note that very deep or highly flexible quantum circuits can generate states that resemble essentially random states in Hilbert space [156]. For such random states, expectation values of physical observables tend to cluster around a typical value, making the total energy largely insensitive to small parameter changes. The resulting landscape therefore resembles a vast, nearly level plateau with very little slope to guide the optimization. For this reason, much current research focuses on designing structured ansätze and initialization strategies (often inspired by quantum chemistry methods such as coupled cluster theory [157]) that restrict the search to physically meaningful regions of the Hilbert space and thereby help avoid these flat landscapes [158].

6.2.2.3 Hamiltonian Simulation. Like variational quantum optimization methods, Hamiltonian simulation is also seen as one of the problems where practical computational speed-ups can be realized on near-term quantum computers. In this section, we give a review of the basic techniques (both for near-term and future quantum hardware) which are used to simulate the time-evolution of quantum chemical Hamiltonians. We recall that the time-dependent Schrödinger equation (with $\hbar = 1$)

$$i\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle, \tag{6.29}$$

has formal solution

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle. \tag{6.30}$$

Thus, implementing the unitary operator

$$U(t) = e^{-iHt} \tag{6.31}$$

is equivalent to simulating real-time quantum dynamics. The quantum advantage of Hamiltonian simulation stems the fact that time evolution in Hilbert space dimension 2^N can be performed with resources polynomial in N, provided N has suitable structure (namely, locality or sparsity) [139]. This is because most physically relevant Hamiltonians exhibit *k-locality*, meaning the Hamiltonian is finite and can be reduced to a linear combination of Pauli strings

$$\hat{H} = \sum_{j=1}^{N} c_j P_j, \tag{6.32}$$

where each string P_j acts non-trivially (i.e., with X, Y, Z Pauli operators) on at most k qubits. For mean-field lattice models, N typically scales linearly with system size; however, for arbitrary Hamiltonians in full second quantization, N scales polynomially in orbital count.

Ultimately, the complexity of Hamiltonian simulation does not just depend on the number of Pauli strings N in the Hamiltonian; rather, it is dependent on the number of non-commuting groups of Pauli strings. To illustrate this concept, we will first consider the case where the Hamiltonian decomposes into two non-commuting pieces,

$$H = A + B, \tag{6.33}$$

contrary to to the behavior of scalars (which always commute under products), if two operators do not commute then

$$e^{-i(A+B)t} \neq e^{-iAt} e^{-iBt}. \tag{6.34}$$

This is problematic; however, for a sufficiently small timestep δt it can be shown that the approximation

$$e^{-i(A+B)\delta t} = e^{-iA\delta t}e^{-iB\delta t} + \mathcal{O}(\delta t^2). \tag{6.35}$$

gives a good approximation in the limit of small δt. Dividing total time t into r steps ($t = r\delta t$) we find that

$$e^{-iHt} \approx \left(e^{-iAt/r}e^{-iBt/r}\right)^r, \tag{6.36}$$

with total error ϵ scaling as

$$\epsilon \approx \mathcal{O}\left(\frac{t^2}{r}\|[A,B]\|\right), \tag{6.37}$$

where $[A,B] = AB - BA$ denotes the operator commutator. Thus, for general Hamiltonians decomposed into L non-commuting groups of Pauli operators $H_\ell = \sum_j c_j P_j$, it can be shown that

$$e^{-iHt} \approx \left(\prod_{\ell=1}^{L} e^{-iH_\ell t/r}\right)^r. \tag{6.38}$$

In the literature, this is is known as a *Suzuki-Trotter product formula*, and the process of decomposing a Hamiltonian into an approximate time-evolution scheme is known as *Trotterization*. Higher-order Suzuki-Trotter formulas have also been derived, whcih systematically cancel leading commutator errors at the cost of requiring more complex quantum circuits. Although we will not prove it here, we note that for any commuting group of k-local Pauli operators P_j (satisfying $[P_i, P_j] = 0$) the exponential of $H_\ell = \sum_j c_j P_j$ can be computed efficiently using a polynomial number of gates [159].

In addition approximating time evolution via small time slices, other approaches exist which encode the Hamiltonian into a larger unitary operator (block

encoding) and then apply polynomial transformations using the *quantum signal processing* (QSP) [160] or *quantum singular value transform* (QSVT) [161, 162] algorithms. The resulting query complexity scales as

$$\mathcal{O}\left(\alpha t + \log(1/\epsilon)\right), \tag{6.39}$$

where ϵ is the desired error tolerance and α bounds the operator norm of H. Importantly, the dependence on precision is only logarithmic, and the scaling in t is linear and provably optimal; however realizing these algorithms is exceptionally difficult on modern noisy hardware, though some randomized approximations to this process have been proposed that may be feasible on NISQ hardware [163].

6.2.2.4 Quantum Phase Estimation. The *quantum phase estimation* (QPE) algorithm is one of the central subroutines in quantum computing and provides the foundation for several quantum algorithms, including BQP-complete problems, such as Shor's discrete logarithm and integer factoring algorithms [164]. In the context of quantum simulation, QPE can also be used to determine the eigenvalues of a Hamiltonian by exploiting the relationship between time evolution and wave function phase accumulation [165]. In this section, we will provide a very basic overview of the QPE algorithm, mentioning some of its applications in quantum chemistry.

The most essential component in the QPE algorithm that yields an exponential advantage over comparable classical algorithms is the inverse *quantum Fourier transform* (QFT). The QFT is the quantum analogue of the discrete Fourier transform, acting on the amplitudes of a quantum state to reveal phase information in the computational basis. For an M-qubit state $|x\rangle = |x_0 x_1 ... x_{M-1}\rangle$, the QFT maps it to

the superposition

$$|x\rangle \mapsto \frac{1}{\sqrt{2^M}} \sum_{k=0}^{2^M-1} e^{i2\pi xk/2^M} |k\rangle \tag{6.40}$$

effectively encoding the integer x into the relative phases of the computational basis states. In matrix form, this transformation is written

$$QFT = \frac{1}{\sqrt{2^M}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix}, \qquad \omega = e^{2\pi i/}. \tag{6.41}$$

An M-qubit QFT transformation can be implemented efficiently on a quantum computer using a sequence of Hadamard gates and controlled phase rotations, requiring only $O(M^2)$ gates. Similarly, its inverse ($QFT^\dagger$) can be applied with an identical number of operations by applying the inverse of the QFT gates in the reverse order.

Now we return to our discussion of the QPE algorithm. Suppose we are given a unitary operator U and an eigenstate $|\psi\rangle$ satisfying

$$U|\psi\rangle = e^{2\pi i\phi}|\psi\rangle, \tag{6.42}$$

where $\phi \in [0, 1)$ is an unknown phase. The goal of quantum phase estimation is to determine an approximation of ϕ to n bits of precision. The algorithm accomplishes this by encoding the phase into the amplitudes of a register of control qubits through controlled applications of powers of U.

The procedure begins with two registers: an n-qubit control register initialized in the state $|0\rangle^{\otimes n}$ and a second register prepared in the eigenstate $|\psi\rangle$. Hadamard

gates are applied to the control qubits to create a uniform superposition,

$$\frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} |k\rangle. \tag{6.43}$$

Each control qubit then applies a controlled power of the unitary operator to the target register. Because $|\psi\rangle$ is an eigenstate of U, these operations imprint phase factors $e^{2\pi i \phi 2^k}$ onto the amplitudes of the control register. The resulting state encodes the phase ϕ in the relative phases of the computational basis states.

Finally, an inverse quantum Fourier transform is applied to the control register, converting the phase information into a binary representation that can be read out by measurement. With high probability, the measurement outcome provides an n-bit binary estimate of ϕ. Figure 6.4 illustrates the standard quantum circuit for phase estimation, for an initial state $|\psi\rangle$ where ψ is initialized close to an eigenstate ψ_n ($\langle\psi|\psi_n\rangle \approx 1$).

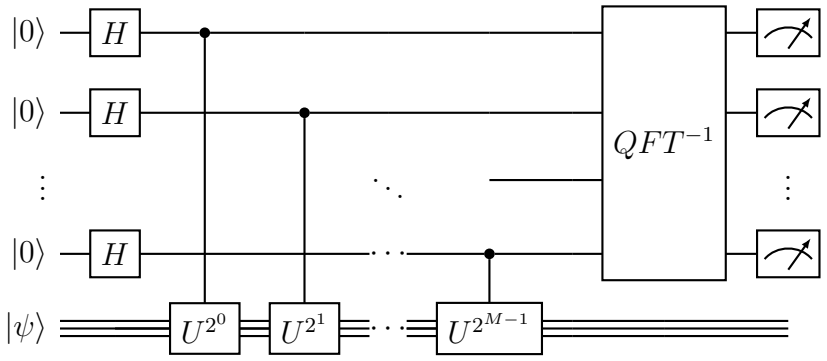

Figure 6.4: Quantum circuit for the M-qubit quantum phase estimation. algorithm. Controlled powers of the unitary operator encode the eigenphase into the control register, after which an inverse quantum Fourier transform is applied before measurement.

In quantum chemistry applications, the unitary operator is typically chosen as the time-evolution operator

$$U = e^{-iHt}, \tag{6.44}$$

where H is the electronic Hamiltonian. This subroutine is usually implemented though a controlled version of some Hamiltonian simulation algorithm, such as a Trotterized product formula.

If the input state $|\psi\rangle$ has significant overlap with an eigenstate $|\psi_n\rangle$ of H, the measured phase yields the corresponding energy eigenvalue. In this way, QPE provides a direct route to high-precision energy estimation, although it generally requires deeper circuits and longer coherence times than variational approaches. As a result, phase estimation is impractical on modern NISQ hardware, but is believed to be feasible on future quantum hardware that is fully fault-tolerant.

6.3 Simulation of Open Quantum Systems

In this section, we pivot to the harder problem of simulating *open quantum systems* on quantum computers. These systems arise when a quantum system of interest interacts with an external environment. In the idealized setting of closed-system quantum mechanics, the state of a system is described by a wavefunction $|\psi\rangle$ whose time evolution is governed by the Schrödinger equation. This description assumes that the system is perfectly isolated from its surroundings, so that its evolution is unitary and information is preserved, and thus its time-evolution is reversible. In realistic physical settings, however, such isolation is rarely achievable. Interactions

with external degrees of freedom are (such as phonons, electromagnetic fields, or measurement apparatus) can exchange energy and information with the system, leading to irreversible phenomena such as decoherence and energy dissipation.

To describe these effects, it is convenient to adopt the density matrix formalism. The state of the system is represented by a density operator ρ, which generalizes the wavefunction description to allow for statistical mixtures of quantum states. If the system S interacts with an environment E, the combined system evolves unitarily under a joint Hamiltonian H_{SE}. The effective state of the system alone is obtained by tracing out the environmental degrees of freedom,

$$\rho_S = \mathrm{Tr}_E(\rho_{SE}), \tag{6.45}$$

where ρ_{SE} is the density matrix of the combined system and environment. Because information about the environment is discarded in this partial trace, the resulting dynamics of ρ_S are generally non-unitary and can exhibit irreversible behavior such as decoherence.

In most models of the time-evolution of open quantum systems, it is common to assume that time-evolution is *Markovian* (e.g., information lost to the environment is never recovered, rendering the environment "memoryless"). In this limit quantum systems evolve according to the Lindblad master equation

$$\frac{d\rho}{dt} = -i[H, \rho] + \mathcal{D}(\rho), \tag{6.46}$$

where the term $-i[H, \rho]$ unitarily evolves ρ under the Schrödinger equation and $\mathcal{D}$ is the *Lindblad dissipator* [166], expanded as

$$\mathcal{D}(\rho) = \sum_i \gamma_i \left(L_i \rho L_i^\dagger - \frac{1}{2} \left\{ L_i^\dagger L_i, \rho \right\} \right) \tag{6.47}$$

where the L_i are *Lindblad operators* that model environment-mediated transitions in the system state with coupling coefficients $\gamma_i > 0$.

In many applications of quantum simulation, particularly in quantum chemistry, one is not primarily interested in the full many-body wavefunction of a large molecular system but rather in the properties of a smaller, chemically active region embedded within a larger environment. Quantum embedding methods exploit this observation by partitioning the total Hilbert space into multiple strongly-correlated subsystems S_j containing the degrees of freedom of interest and their associated environments E representing the surrounding electronic structure. The total Hamiltonian may then be written schematically as

$$\hat{H} = \sum_i \hat{H}_{S_j} + \hat{H}_E + \hat{H}_{S_j E}, \tag{6.48}$$

where $H_{S_j E}$ describes the coupling between a subsystem and the surrounding environment. Even if the combined system evolves unitarily, the effective dynamics of the subsystem alone are generally non-unitary due to entanglement and information exchange with the environment.

This perspective is particularly important for hybrid quantum-classical algorithms such as the variational quantum eigensolver (VQE), where the number of qubits available on near-term devices may limit the size of the system that can be treated explicitly. Quantum embedding approaches therefore attempt to simulate a small correlated subsystem on a quantum processor while representing the surrounding environment through an effective bath or embedding potential. In this framework, physically meaningful observables associated with the subsystem are obtained from

the reduced density matrix

$$\rho_S = \mathrm{Tr}_E(\rho_{SE}), \tag{6.49}$$

which is constructed by tracing out the environmental degrees of freedom. Local expectation values then follow from

$$\langle O_S \rangle = \mathrm{Tr}(\rho_S O_S), \tag{6.50}$$

allowing properties of the embedded region to be recovered without explicitly representing the full many-body environment on the quantum device.

As we will discuss in the following two chapters, quantum simulation approaches usually require the dilation of the density operator into a larger Hilbert space, and then applying either Trotterized evolution or block-encoded simulation via the QSVT. Since the digital time-evolution of the density matrix of an n-qubit system acts in a space of dimension 4^n, efficient encoding requires exploiting sparsity and structure in dissipative operators, and constitutes a significant challenge in quantum algorithm design.

6.4 Overview of Research Contributions

This dissertation develops scalable digital simulation algorithms tailored to quantum materials with environmental coupling and constrained hardware resources. In the followig two chapters, we present published work on the digital quantum simulation of open quantum systems, with an emphasis on designing algorithms suitable for NISQ hardware. Below, we briefly review the key contributions of this work:

6.4.1 *Trotterless Simulation Algorithms for Open Quantum Systems*

Traditional simulation relies on Trotterization of the Lindblad dynamics in (6.46) and subsequent simulation by approximating the exponential of the time, evolution operator. In this paper, we examining the decomposition of such exponentials into the form

$$e^{t(\mathcal{H}+\mathcal{L})}. \tag{6.51}$$

where $\mathcal{H}$ and $\mathcal{L}$ model effective Hamiltonian evolution and Linblad dissipative evolution. This work explores ways in which certain structured Hamiltonians satisfying the commutator relations involving $\mathcal{H}$ and $\mathcal{L}$ can actually admit representations that avoid Trotterization altogenter, allowing for the distribution of system dynamics over many short circuits, allowing for efficient "Trotterless" simulation suitable for NISQ devices.

6.4.2 *Applying Open Quantum System Simulation Methods to Trapped-Ion NISQ Hardware*

Trapped-ion platforms have demonstrated state-of-the art accuracy in processing quantum information with all-to-all connectivity and high-fidelity gates. This work applies the theory developed in previous published work to simulate selected open quantum systems on real quantum hardware. It employs measurement-based tomographic readout of quantum density matrices and the characterization and inversion of device noise channels to mitigate the effects of noise in the simulation results.

CHAPTER SEVEN

Trotterless Methods for the Simulation of Open Quantum Systems

This chapter published as: Burdine, Colin, and Enrique P. Blair. "Trotterless Simulation of Open Quantum Systems for NISQ Quantum Devices." *Advanced Quantum Technologies*, Oct. 2024, p. 2400240., doi: 10.1002/qute.202400240.

The simulation of quantum systems is one of the flagship applications of near-term NISQ (noisy intermediate-scale quantum) computing devices. Efficiently simulating the rich, non-unitary dynamics of open quantum systems remains challenging on NISQ hardware. Current simulation methods for open quantum systems employ time-stepped Trotter product formulas ("Trotterization") which can scale poorly with respect to the simulation time and system dimension. Here, we propose a new simulation method based on the derivation of a time-perturbative KrausThe simulation of quantum systems is one of the flagship applications of near-term NISQ (noisy intermediate-scale quantum) computing devices. Efficiently simulating the rich, non-unitary dynamics of open quantum systems remains challenging on NISQ hardware. Current simulation methods for open quantum systems employ time-stepped Trotter product formulas ("Trotterization") which can scale poorly with respect to the simulation time and system dimension. Here, we propose a new simulation method based on the derivation of a time-perturbative Kraus operator series representation of the system. We identify a class of open quantum systems for which this method produces circuits of time-independent depth, which may serve as a desirable alternative to Trotterization, especially on NISQ devices.

7.1 Introduction

The quantum computer is an emerging technology with the potential to greatly accelerate progress in the fields of chemistry, physics, and materials science [16, 139]. One of the most extensively researched near-term applications of quantum computing is the simulation of physical systems, since state-of-the art NISQ (noisy intermediate-scale quantum) hardware has enabled accurate simulations of various kinds of systems [167]. Recently, significant effort has been devoted toward the representation and simulation of closed many-body quantum systems, such as spin-lattice systems [168], fermionic systems [169, 170], and bosonic systems [171, 172]. However, relatively few studies have investigated the time evolution of open quantum systems, which exhibit complex non-unitary dynamics due to interactions with an environment. Since quantum computation is inherently unitary, efficiently simulating the non-unitary dynamics of open systems on quantum computers remains a challenging yet important problem, especially on state-of-the art NISQ hardware [173].

7.1.1 Related Work

Many different approaches have been taken to simulate open quantum systems, such as explicitly modeling the environment [174, 175], numerically evolving the non-unitary dynamics of the system [176–182], harnessing device noise to model the environment [183], or representing the system's dynamics in closed form, either as an arbitrary quantum channel [184] or a Kraus representation [185–187]. Variational approximations of the steady states of open systems have also been successfully demonstrated [188]. In a majority of these studies [177–183, 185–187], systems are considered under the Born-Markov approximation, in which environmental coupling

is weak and correlations are assumed to decay rapidly such that the system exhibits Markovian dynamics (i.e. the environment is effectively “memoryless"). The dynamics of Markovian systems are governed by one of a number of master equations, such as the stochastic Schrödinger equation [189] or the Lindblad equation [166]. The most common and readily generalizable approach to simulating Markovian systems is the use of Suzuki-Trotter product formulas (“Trotterization") [190, 191]. This approach uses discrete-time numerical methods to compute the time evolution of sparse non-commuting components of the Hamiltonian and any non-Hermitian operators that model environmental interactions [177, 192]. However, Suzuki-Trotter-based methods require quantum circuits that increase linearly with the time evolution period and quadratically with the system dimension [159, 177, 178]. This is problematic, especially if these methods are to be realized on NISQ devices, where factors such as device topology and noise must be taken into account [167].

Motivated by the constraints of modern quantum hardware, significant strides have been made toward finding representations of open quantum systems that either allow for quantum resource-efficient Trotterization or avoid Trotterization altogether. In a few cases, Trotterless methods for the time-evolution of open systems have been found [185–187], though these methods only apply to only a limited number of toy example systems, where the dynamics admit a known closed-form Kraus representation. For arbitrary open quantum systems, computing a minimal Kraus representation through established methods is known to be classically hard [193, 194], since it requires constructing and then diagonalizing the system's Choi matrix. For a system of dimension d, diagonalizing the Choi matrix requires $O(d^6)$ operations to derive

the d^2 Kraus operators corresponding to just a single evolution period t. Nonetheless, due to the inherent non-uniqueness of Kraus representations, it has remained an unexplored question whether there exist specific classes of systems for which Kraus representations can be derived (or even approximated) efficiently. It also remains to be seen if such representations may facilitate new NISQ-friendly simulation methods.

7.1.2 Main Results

In this paper, we make preliminary steps toward identifying and characterizing systems which admit “Trotterless" representations. More specifically, we identify a general class of Born-Markov open quantum systems described by the Lindblad master equation that admit a type of Kraus operator representation that we refer to as a “time-perturbative Kraus operator series". In this series representation, each Kraus operator can be written as a closed-form function of time. We show that the time evolution of systems that admit this representation can be computed to a desired order of accuracy for any evolution period $t \geq 0$ using a series of parameterized quantum circuits (one per Kraus operator) with a gate complexity independent of t. The closed-form t-independent property of these circuits makes them ideal candidates for execution on NISQ computers, since they do not employ Trotterization or other quantum digital simulation methods. This concept is illustrated in Figure 7.1.

To characterize the dynamics of Lindblad-type open quantum systems, we introduce the effective Hamiltonian superoperator $\mathcal{H}$ and Lindblad superoperator $\mathcal{L}$. We show that $\mathcal{H}$ and $\mathcal{L}$ can be used to identify this class of systems, provided that the following three general conditions are met:

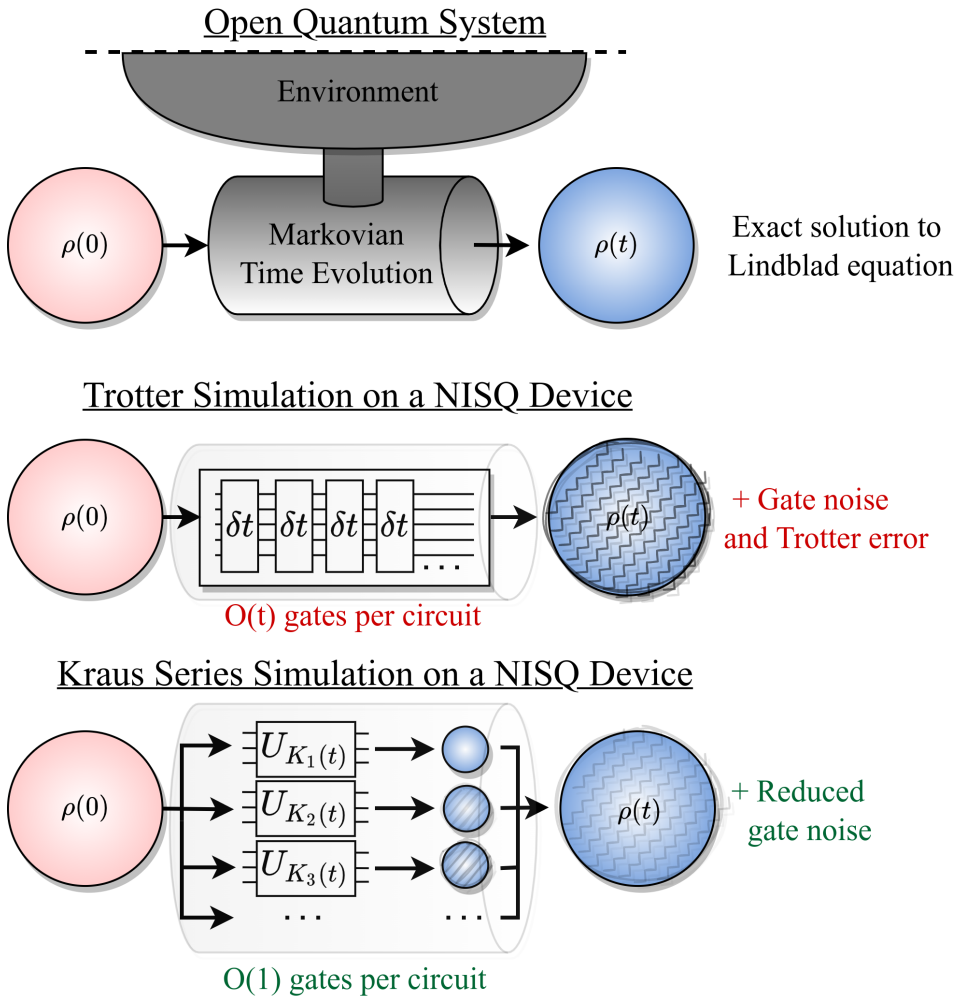

Figure 7.1: Overview of approaches to simulating Markovian open quantum systems. When Trotter-based methods are applied, deep circuits are required with $O(t)$ gate complexity. For complex systems, each Trotter step (δt) may require many gates to numerically approximate. On NISQ devices, this results in noise and Trotter approximation error. For certain kinds of systems, our time-perturbative Kraus series approach distributes the complexity of the simulation over many short circuits, thereby reducing the effect of noise.

(1) The system's effective Hamiltonian dynamics $\exp(t\mathcal{H})$ can be simulated efficiently via a quantum circuit with gate complexity independent of t.

(2) The system's Lindblad operators L_n are sparse and can be implemented through a quantum circuit of bounded gate complexity.

(3) The system's $\mathcal{H}$ and $\mathcal{L}$ superoperators satisfy the commutation relation $[\mathcal{H}, \mathcal{L}] = \alpha\mathcal{L} + c$ for real scalars $\alpha, c \geq 0$.

As a more concrete case, we show that systems with a Hamiltonian H and a set of sparse Lindblad operators $\{L_n\}_{n=1}^{N_L}$ that satisfy the commutation relations

(i) $[H, L_n^\dagger L_n] = 0$ (for all L_n)

(ii) $[L_n^\dagger L_n, L_{n'}^\dagger L_{n'}] = 0$ (for all $L_n, L_{n'}$)

(iii) $[H, L_n] = \nu L_n$
(for some $\nu \in \mathbb{C}$ with $\mathrm{Im}(\nu) \geq 0$, for all L_n)

(iv) $\sum_{n'} \gamma_{n'} [L_{n'}^\dagger L_{n'}, L_n] = \lambda L_n$
(for some $\lambda \in \mathbb{C}$ with $\mathrm{Re}(\lambda) \leq 0$, for all L_n)

all satisfy conditions 1-3 above and can be simulated with a series of quantum circuits, each of which has time-independent gate complexity at most $O(d^2)$, where d is the dimension of H. Although this series generally consists of an infinite number of circuits, we show that accurate approximations of the final state can still be made if the series is truncated to a finite number of circuits. This allows for a tunable trade-off between desired accuracy and number of circuits. We also show that for some systems, exact solutions can be obtained with a finite number of circuits.

Finally, we show that two widely studied open quantum systems, the multi-qubit continuous-time Pauli channel and the damped quantum harmonic oscillator, both satisfy conditions 1-3 and can be simulated efficiently with a series of "Trotterless" quantum circuits with t-independent depth. Both of these systems play a crucial role in modeling the Markovian decoherence of spin-1/2 fermionic systems and bosonic systems respectively.

7.2 Background

7.2.1 Lindblad Equation

In this paper we consider non-relativistic open quantum systems under the Born-Markov approximation. The time evolution of these systems forms a dynamical semigroup, and the master equation that generates their non-unitary dynamics can be written in the Lindbladian (also known as the diagonalized GKSL) form [166]:

$$\frac{d\rho}{dt} = -\frac{i}{\hbar}[H, \rho(t)] + \sum_n \gamma_n \left(L_n \rho(t) L_n^\dagger - \frac{1}{2}\{L_n^\dagger L_n, \rho(t)\} \right). \tag{7.1}$$

Above, H is the system Hamiltonian and the L_n are Lindblad operators with corresponding damping parameters $\gamma_n \geq 0$. These operators are often imposed phenomenologically on the system, since a large number of Lindblad operators (sometimes even an infinite number [195]) are usually needed to model the Markovian dynamics of the total unitary evolution of a system and environment. For example, in systems with quadratic Hamiltonians expressed in terms of bosonic creation and annihilation operators $\hat{b}_n^\dagger, \hat{b}_n$, i.e:

$$H = \sum_n c_n \hat{b}_n^\dagger \hat{b}_n, \tag{7.2}$$

the operators $\hat{b}_n$ are often used as the Lindblad operators L_n with appropriate choice of damping γ_n to model spontaneous emission and decay within the system. However, in the simulation of other systems, more complex Lindblad operators (e.g. depolarizing and dephasing operators) have been applied to model various forms of environment-induced decoherence. These kinds of Lindblad operators are seen in the continuous-time Pauli channel, which is used to model unbiased qubit decoherence in quantum information processors.

7.2.2 Kraus Operators

It can be shown that the time evolution of a system $\rho(t)$ satisfying (8.1) generates a completely positive trace-preserving (CPTP) map $\mathcal{E}_t$, which forms a quantum channel for each time evolution period $t \geq 0$ [166]. This channel can be represented non-uniquely as a sum of time-dependent Kraus operators:

$$\rho(t) = \mathcal{E}_t(\rho(0)) = \sum_i K_i(t)\rho(0)K_i^\dagger(t). \tag{7.3}$$

The Kraus operators $K_i(t)$ must satisfy the constraint

$$\sum_i K_i(t)K_i(t)^\dagger = 1 \tag{7.4}$$

for $t \geq 0$ Also, the time-dependence of the Kraus operators must be such that the time evolution map $\mathcal{E}_t$ satisfies the dynamical semigroup property

$$\mathcal{E}_{t_2}(\mathcal{E}_{t1}(\rho)) = \mathcal{E}_{(t_1+t_2)}(\rho), \tag{7.5}$$

where $\mathcal{E}_0 \equiv \mathcal{I}$ (the identity map).

7.2.3 *Superoperator Formalism*

Before proceeding, we must introduce the superoperator notation that we will use throughout the remainder of this paper. A superoperator is an operator that acts linearly on the space of operators. In this case, we will focus on superoperators that act linearly on the "vectorized" density matrix, $\vec{\rho}$, given by

$$\vec{\rho} = \sum_{i,j} \langle e_i | \rho | e_j \rangle (|e_i\rangle \otimes |e_j\rangle), \tag{7.6}$$

where the kets $|e_i\rangle$ form a complete orthonormal basis. It can be shown that superoperators acting by a left and right operator of an $n \times n$ density matrix ρ can be represented as an $n^2 \times n^2$ matrix equal to the Kronecker product of the left-acting operator and the transpose of the right-acting operator, which acts linearly on $\vec{\rho}$. For example, a superoperator $\mathcal{D}$ that acts on an operator ρ through left and right multiplication by the respective matrices A and $B^\dagger$ has the superoperator representation $(A \otimes \overline{B})$, where $\overline{B}$ denotes the complex conjugate of B. This gives rise to the following correspondence between operator and superoperator representations:

$$\mathcal{D}(\rho) = A\rho B^\dagger \;\Leftrightarrow\; \mathcal{D}\vec{\rho} = (A \otimes \overline{B})\vec{\rho}. \tag{7.7}$$

This correspondence is quite useful, since it lifts the dynamics of a system's density matrix into a larger space where the dynamics can be written as a matrix-vector product.

7.3 *Theory*

In this section we will introduce our proposed framework for simulating open quantum systems that satisfy conditions 1-3. First, we will give an overview of our general framework for characterizing the dynamics of open systems in terms of the algebra of the effective Hamiltonian superoperator $\mathcal{H}$ and the Lindblad superoperator

$\mathcal{L}$. Next, we will show how a Kraus series can be derived from $\mathcal{H}$ and $\mathcal{L}$ when the commutation relation in condition 3 is satisfied. We will also give examples of how useful identities in the Lindblad operator algebra can be applied to further reduce the complexity of the Kraus series. Finally, we will show how each Kraus operator in the series can be realized as a quantum circuit. A summary of this procedure is shown in Figure 7.2. At the end of this section, we will also give more detailed analysis of the convergence of the Kraus series, and describe how the series can be truncated to a finite number of terms while still providing accurate results.

7.3.1 General Framework

We will begin by using the correspondence in (7.7) to convert the right hand side of (8.1) into superoperator form. To simplify this process, we first introduce the effective Hamiltonian operator V_H and scaled Lindblad operators V_n:

$$V_H = H - \frac{i\hbar}{2}\sum_n \gamma_n L_n^\dagger L_n \tag{7.8}$$

$$V_n = \sqrt{\gamma_n} L_n \tag{7.9}$$

Next, we introduce the effective Hamiltonian superoperator $\mathcal{H}$ and Lindblad superoperator $\mathcal{L}$:

$$\mathcal{H} = \left(\frac{-i}{\hbar} V_H\right)^{\overline{\oplus} 2} \tag{7.10}$$

$$\mathcal{L} = \sum_n \left(\sqrt{\gamma_n} L_n\right)^{\overline{\otimes} 2} = \sum_n V_n^{\overline{\otimes} 2} \tag{7.11}$$

In (7.10), (7.11), and throughout the remainder of this paper, we use the notational shorthand

$$A^{\overline{\oplus} 2} \equiv A \otimes I + I \otimes \overline{A} \tag{7.12}$$

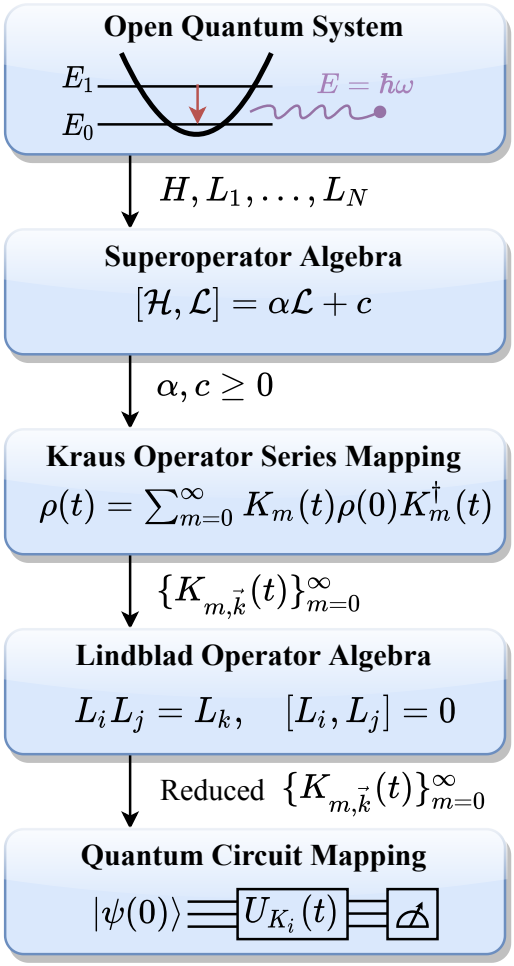

Figure 7.2: A summary of the steps required to realize open quantum systems satisfying conditions 1-3 as a series of quantum circuits. By exploiting the superoperator algebra, a Kraus series representation of a system can be found. This series can then be simplified using product and commutator identities of the Lindblad operator algebra (if any exist). Finally, each Kraus operator is mapped to a quantum circuit, which has depth independent of the time evolution period t.

to denote the conjugated Kronecker sum and

$$A^{\overline{\otimes}2} \equiv A \otimes \overline{A} \tag{7.13}$$

to denote the conjugated Kronecker product.

It is worth remarking that the effective Hamiltonian superoperator $\mathcal{H}$ does not generate a unitary evolution of the density matrix, though it can be written in a form that evolves ρ in a manner analogous to that of unitary evolution under the effective Hamiltonian V_H:

$$e^{t\mathcal{H}}\vec{\rho} \quad \Leftrightarrow \quad (e^{-itV_H/\hbar})\rho(e^{-itV_H/\hbar})^\dagger \tag{7.14}$$

The dynamics of (7.14) are non-unitary because the damping terms $\frac{-i\hbar}{2}\sum_n \gamma_n L_n^\dagger L_n$ in (7.8) are skew-Hermitian with non-positive purely imaginary eigenvalues. As a result, the trace and purity of ρ are not preserved with time. However, in the no-damping limit where $\gamma_n \rightarrow 0$, these terms vanish and $\mathcal{H}$ generates unitary dynamics.

Evolving the system under $\mathcal{H}$ alone is insufficient to account for the direct action of the environment on the system. To completely account for this action, we must also incorporate the Lindblad superoperator $\mathcal{L}$, given in (7.11). If we combine the action of $\mathcal{H}$ and $\mathcal{L}$ on a vectorized density matrix $\vec{\rho}$, we obtain a linear superoperator equivalent to the right hand side of the Lindblad equation (8.1), which is completely positive and trace-preserving:

$$\frac{d\vec{\rho}}{dt}(t) = (\mathcal{H} + \mathcal{L})\vec{\rho}(t). \tag{7.15}$$

We can verify that the equivalence of (7.15) to (8.1) holds by expanding $\mathcal{H}$ and $\mathcal{L}$ to obtain (7.16), and checking the operator-to-superoperator correspondence in (7.7):

$$\begin{aligned}\mathcal{H}+\mathcal{L} = &\left(\frac{-i}{\hbar}(H\otimes I - I\otimes\overline{H}) - \frac{1}{2}\sum_n \gamma_n(L_n^\dagger L_n\otimes I + I\otimes\overline{L_n}^\dagger\overline{L_n})\right)\\ &+\left(\sum_n \gamma_n(L_n\otimes\overline{L_n})\right).\end{aligned} \tag{7.16}$$

Since $(\mathcal{H}+\mathcal{L})$ acts linearly on $\vec{\rho}$, the time-evolution of the system is given by

$$\vec{\rho}(t) = \mathcal{E}_t\vec{\rho}(0) = e^{t(\mathcal{H}+\mathcal{L})}\vec{\rho}(0), \tag{7.17}$$

where $\mathcal{E}_t$ is the time evolution superoperator that satisfies (7.5). From equation (7.3), it follows that $e^{t(\mathcal{H}+\mathcal{L})}$ must be equal to a sum of time-dependent Kraus operators $K_i(t)$, which means that such a representation satisfies the equation

$$\mathcal{E}_t = \sum_i K_i(t)^{\overline{\otimes}2} = e^{t(\mathcal{H}+\mathcal{L})}. \tag{7.18}$$

Finding a set of $K_i(t)$ in a closed matrix form that satisfies this for all $t \geq 0$ is a nontrivial task, as it amounts to decomposing an operator exponential into a sum of terms that factor as a conjugated Kronecker product (i.e. $K_i(t)^{\overline{\otimes}2}$).

7.3.1.1 Existing approaches for obtaining a Kraus decomposition. A standard approach to obtain a Kraus decomposition from the Lindblad equation is outlined in [193] and [194]. This approach makes use of the Hermitian supermatrix formalism first developed by Choi [196], which requires computing the superoperator exponential $\mathcal{E}_t = e^{t(\mathcal{H}+\mathcal{L})}$ and constructing the Choi matrix $\mathcal{T}(t)$ associated with evolution over a time period t. In block-matrix form, the Choi matrix is constructed

as

$$\mathcal{T}(t) = \begin{pmatrix} \mathcal{E}_t(E_{11}) & \mathcal{E}_t(E_{12}) & \dots \\ \mathcal{E}_t(E_{21}) & \mathcal{E}_t(E_{22}) & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}, \tag{7.19}$$

where $\mathcal{E}_t$ is evaluated on all density matrix basis elements $E_{ij} = |e_i\rangle \langle e_j|$ for a set of kets $|e_i\rangle$ that form a complete orthonormal basis. If a system has dimension d, the Choi matrix has dimension $d^2 \times d^2$. For CPTP maps generated by the Lindblad equation, the Choi matrix is Hermitian positive definite and can be diagonalized in the form

$$\mathcal{T}(t) = \mathcal{U}\, \Xi\, \mathcal{U}^\dagger = \sum_{m=1}^{d^2} \xi_m \mathbf{u}_m \mathbf{u}_m^\dagger. \tag{7.20}$$

From the eigenvalues $\xi_m \geq 0$ and the eigenvectors $\mathbf{u}_m$ of the Choi matrix, a set of d^2 Kraus operators $K_m(t)$ are obtained by "unvectorizing" each $\mathbf{u}_m$ into a $d \times d$ matrix operator and multiplying by $\sqrt{\xi_m}$ such that

$$(K_m(t))_{ij} = \sqrt{\xi_m}(\mathbf{u}_m)_{i+dj}. \tag{7.21}$$

Although the Choi matrix is useful for characterizing CPTP maps, using it to derive Kraus representations of a system presents some practical challenges. First, diagonalizing an arbitrary $d^2 \times d^2$ Choi matrix requires $O(d^6)$ operations on a classical computer, which is very undesirable, especially for N-qubit systems where $d = 2^N$, or systems with $d = \infty$. Second, the Kraus operators that are recovered from the Choi matrix can not always be written as a closed function of time, which means that the Kraus operators $K_m(t)$ must be recalculated separately for each evolution period t. Finally, because obtaining this Kraus decomposition requires computing $\mathcal{E}_t = e^{t(\mathcal{H}+\mathcal{L})}$, one might as well bypass the Kraus representation altogether and instead numerically

evolve the initial state under $\mathcal{E}_t$. On a classical computer, this is achieved through numerically solving the Lindblad equation. On quantum computers, this is most commonly achieved by evolving the "vectorized" density matrix $\vec{\rho}$ under $\mathcal{E}_t$ as if it were a quantum state vector $|\vec{\rho}\rangle$ of size d^2:

$$|\vec{\rho}(t)\rangle = \mathcal{E}_t \, |\vec{\rho}(0)\rangle \, . \tag{7.22}$$

7.3.1.2 Existing approaches for simulating $\mathcal{E}_t$ on a quantum computer. Several methods have been proposed for simulating $\mathcal{E}_t$ on a quantum computer (e.g. see [177, 179, 180, 187]). A significant majority of these methods approximate the exponential $e^{t(\mathcal{H}+\mathcal{L})}$ by splitting the expanded form of $\mathcal{H} + \mathcal{L}$ in (7.16) into two or more non-commuting terms, whose exponentials individually are easy to realize as quantum circuits. Then, by applying the Trotter product formula (or higher-order Suzuki-Trotter formulas [191]) one can obtain an approximation of $\mathcal{E}_t$ in terms of the exponential of each part of $\mathcal{H} + \mathcal{L}$ individually [190]. This method is sometimes referred to as "Trotterization". For example, with a small time step δt, one can apply the second order Suzuki-Trotter approximation

$$\mathcal{E}_{\delta t} = e^{\delta t(\mathcal{H}+\mathcal{L})} = e^{(\delta t/2)\mathcal{H}} e^{(\delta t)\mathcal{L}} e^{(\delta t/2)\mathcal{H}} + O((\delta t)^3). \tag{7.23}$$

An approximation formula the full time-evolution superoperator $\mathcal{E}_t$ is then obtained by dividing the evolution period t into n discrete time steps ($\delta t = t/n$) and repeatedly applying (7.23), resulting in

$$\mathcal{E}_t = \left[e^{(t/2n)\mathcal{H}} e^{(t/n)\mathcal{L}} e^{(t/2n)\mathcal{H}}\right]^n + O(t^3/n^2). \tag{7.24}$$

If both $e^{\delta t\mathcal{H}}$ and $e^{\delta t\mathcal{L}}$ can be realized efficiently as unitary quantum circuits, then the final state of the density matrix encoded in a dimension d^2 state vector (as shown in

(7.22)) can be obtained by using (7.24) or a higher-order Trotter formula. However, this is problematic, as it requires repeatedly applying the $e^{\delta t\mathcal{H}}$ and $e^{\delta t\mathcal{L}}$ quantum circuits. In addition, the encoding of the density matrix as a statevector $|\vec{\rho}\rangle$ requires twice as many qubits to represent the system, requiring twice as many gates to evolve both the "left side" and "right side" of $\rho(0)$. For these reasons, we consider Trotter-based methods to be less desirable for NISQ computing applications. This motivates our search for classes of systems that admit non-minimal Kraus representations which do not require diagonalizing the Choi matrix, but nonetheless satisfy (7.18) and the standard Kraus operator properties (7.3), (7.4), and (7.5).

7.3.1.3 The environmental interaction picture. In this paper, we take a different approach to solving for a Kraus representation of the system that does not require the diagonalization of the Choi matrix. Instead, we solve for an infinite series of Kraus operators which are derived by applying the Zassenhaus product formula to the right-hand side of (7.18). The Zassenhaus product formula expands $e^{t(\mathcal{H}+\mathcal{L})}$ as the product of exponentials of commutators of $\mathcal{H}$ and $\mathcal{L}$ as follows:

$$\begin{aligned} e^{t(\mathcal{H}+\mathcal{L})} &= e^{t\mathcal{H}}e^{t\mathcal{L}}e^{-\frac{t^2}{2}[\mathcal{H},\mathcal{L}]} \\ &\quad \times e^{\frac{t^3}{6}([\mathcal{H},[\mathcal{H},\mathcal{L}]]+2[\mathcal{L},[\mathcal{H},\mathcal{L}]])}e^{-\frac{t^4}{24}(\ldots)}\ldots \\ &= e^{t\mathcal{H}}\prod_{m=1}^{\infty}\exp\left(t^m\mathcal{C}_m\right). \end{aligned} \tag{7.25}$$

For $m > 1$, $\mathcal{C}_m$ is a degree m homogeneous Lie polynomial, whereas for $m = 1$ we define $\mathcal{C}_1 = \mathcal{L}$. The exponents $\mathcal{C}_m$ can be obtained by recursively applying the Baker-Campbell-Hausdorff formula, or through one of the methods outlined in [197] or [198]. In Table 7.1, we list the commutator exponents $\mathcal{C}_m$ up to $m = 5$.

Table 7.1: Table of the Zassenhaus formula commutator exponents $\mathcal{C}_m$ in (7.25) up to $m = 5$.

m	$\mathcal{C}_m$
1	$\mathcal{L}$
2	$-\frac{1}{2}[\mathcal{H}, \mathcal{L}]$
3	$\frac{1}{6}([\mathcal{H}, [\mathcal{H}, \mathcal{L}]] + 2[\mathcal{L}, [\mathcal{H}, \mathcal{L}]])$
4	$-\frac{1}{24}([\mathcal{H}, [\mathcal{H}, [\mathcal{H}, \mathcal{L}]]] + 3[\mathcal{L}, [\mathcal{H}, [\mathcal{H}, \mathcal{L}]]] + 3[\mathcal{L}, [\mathcal{L}, [\mathcal{H}, \mathcal{L}]]])$
5	$\frac{1}{120}([\mathcal{H}, [\mathcal{H}, [\mathcal{H}, [\mathcal{H}, \mathcal{L}]]]] + 4[\mathcal{L}, [\mathcal{H}, [\mathcal{H}, [\mathcal{H}, \mathcal{L}]]]]) +$ $6[\mathcal{L}, [\mathcal{L}, [\mathcal{H}, [\mathcal{H}, \mathcal{L}]]]] + 4[\mathcal{L}, [\mathcal{L}, [\mathcal{L}, [\mathcal{H}, \mathcal{L}]]]]) +$ $6[[\mathcal{H}, \mathcal{L}], [\mathcal{H}, [\mathcal{H}, \mathcal{L}]]] + 12[[\mathcal{H}, \mathcal{L}], [\mathcal{L}, [\mathcal{H}, \mathcal{L}]]])$

Equation (7.25) re-frames the right hand side of (7.18) as the evolution of a product of Zassenhaus commutator exponentials ($\prod_{m=1}^{\infty} \exp(t^m \mathcal{C}_m)$) under the effective Hamiltonian superoperator $\mathcal{H}$. This motivates the introduction of the environment action superoperator

$$\mathcal{A}_{\text{env}}(t) = \prod_{m=1}^{\infty} e^{t^m \mathcal{C}_m}, \tag{7.26}$$

which incorporates the cumulative effect of the Lindblad superoperator $\mathcal{L}$ acting on the system over a time period $t > 0$. We can interpret $\mathcal{A}_{\text{env}}(t)$ as a superoperator that evolves the system in the "environmental interaction picture", which is related to the Schrödinger picture by the non-unitary (but nonetheless invertible) transformation $e^{t\mathcal{H}}$. It immediately follows that the time-evolution superoperator $\mathcal{E}_t$ can be written in the form

$$\mathcal{E}_t = e^{t\mathcal{H}} \mathcal{A}_{\text{env}}(t). \tag{7.27}$$

If both $e^{t\mathcal{H}}$ and $\mathcal{A}_{\text{env}}(t)$ can be realized efficiently on a quantum computer, then the time evolution of the entire system can be computed efficiently by first evolving $\rho(0)$ in the environmental interaction picture using $\mathcal{A}_{\text{env}}(t)$ and then transforming

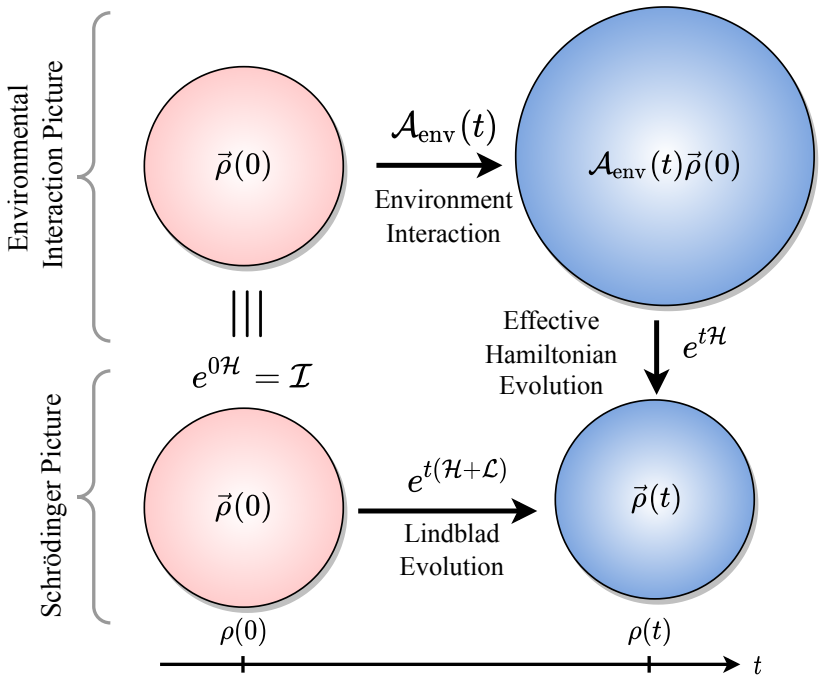

Figure 7.3: Illustration of the relationship between the evolution of the system in the Schrödinger picture and the environmental interaction picture. In the environmental interaction picture, the system is evolved with the superoperator $\mathcal{A}_{\text{env}}(t)$, which is not trace-preserving. This evolved state can be transformed back to the Schrodinger picture through evolution under the effective Hamiltonian, which takes the form (7.14). The state $\mathcal{A}_{\text{env}}(t)\vec{\rho}(0)$ is shown larger than the others to emphasize that the trace of the density matrix is not preserved in the environmental interaction picture. The transformation $e^{t\mathcal{H}}$ renormalizes the trace and returns the system to the Schrödinger picture.

back into the Schrödinger picture as suggested in Figure 7.3. However, computing $\mathcal{A}_{\text{env}}(t)$ for arbitrary systems may be quite difficult, since it requires computing the product of all Zassenhaus exponentials of the form $\exp(t^m\mathcal{C}_m)$. In the next section we will show that when a system satisfies certain superoperator commutation relations, these commutator exponentials reduce to a tractable form, which allows for the derivation of a closed form series expansion of $\mathcal{A}_{\text{env}}(t)$.

7.3.2 Derivation of Kraus Operators

In Section 8.1, we claimed to have found a class of systems which admit a Kraus operator series representation, and we characterized them as satisfying three general conditions (conditions 1-3). For now we will defer the discussion of conditions 1 and 2 to a later section and focus on condition 3, which requires that a system's $\mathcal{H}$ and $\mathcal{L}$ superoperators satisfy the commutation relation

$$[\mathcal{H}, \mathcal{L}] = \alpha\mathcal{L} + c \tag{7.28}$$

for $\alpha, c \in \mathbb{R}_{\geq 0}$. We will proceed by first deriving closed form Kraus operators for systems satisfying condition 3, and then show how they can be simulated with quantum circuits. To facilitate the derivation, we will consider the following two cases separately:

$$\text{Case (I):} \quad [\mathcal{H}, \mathcal{L}] = c \qquad (\text{for } c \geq 0) \tag{7.29}$$

$$\text{Case (II):} \quad [\mathcal{H}, \mathcal{L}] = \alpha\mathcal{L} + c \quad (\text{for } \alpha > 0, c \geq 0) \tag{7.30}$$

7.3.2.1 Case (I) Kraus representation. For case (I) systems, we apply the commutation relation (7.29) to the Zassenhaus formula (7.25) and observe that all commutator exponents $\mathcal{C}_m$ vanish except for $\mathcal{C}_1 = \mathcal{L}$ and $\mathcal{C}_2 = -\frac{c}{2}$.

$$e^{t(\mathcal{H}+\mathcal{L})} = e^{t\mathcal{H}} e^{t\mathcal{L}} e^{-\frac{ct^2}{2}} \tag{7.31}$$

Substituting the non-vanishing $\mathcal{C}_m$ terms into (7.26), we see that the environment action superoperator is

$$\mathcal{A}_{\text{env}}(t) = e^{-ct^2/2} e^{t\mathcal{L}}. \tag{7.32}$$

Expanding the $e^{t\mathcal{L}}$ in $\mathcal{A}_{\text{env}}(t)$ as a Taylor series, we obtain

$$\mathcal{A}_{\text{env}}(t) = \sum_{m=0}^{\infty} \frac{t^m}{m!} \mathcal{A}_{\text{env}}^{(m)}(t), \tag{7.33}$$

where each term $\mathcal{A}_{\text{env}}^{(m)}$ can be further expanded as a sum over all product sequences of m Lindblad operators:

$$\mathcal{A}_{\text{env}}^{(m)}(t) = e^{-\frac{ct^2}{2}} \mathcal{L}^m \tag{7.34}$$

$$= e^{-\frac{ct^2}{2}} \left(\sum_{n=1}^{N} V_n^{\overline{\otimes} 2} \right)^m \tag{7.35}$$

$$= e^{-\frac{ct^2}{2}} \sum_{\vec{k} \in \{1,2,...,N_L\}^m} \left(\prod_{j=1}^{m} \left[\sqrt{\gamma_{\vec{k}_j}} L_{\vec{k}_j} \right] \right)^{\overline{\otimes} 2} \tag{7.36}$$

Each of the $(N_L)^m$ summation terms in (7.36) is indexed by a vector $\vec{k} \in \{1, 2, ..., N_L\}^m$. This vector indicates which m (out of the total N_L) Lindblad operators are applied, and in which order.

By combining equations (7.27), (7.33), and (7.36), we can write the time evolution of the system as a time-perturbative Kraus series that satisfies (7.18). This series takes the superoperator form

$$\vec{\rho}(t) = \sum_{m=0}^{\infty} \sum_{\vec{k} \in \{1,2,...,N_L\}} K_{m,\vec{k}}(t)^{\overline{\otimes} 2} \vec{\rho}(0), \tag{7.37}$$

which is equivalent to

$$\rho(t) = \sum_{m=0}^{\infty} \sum_{\vec{k} \in \{1,2,...,N_L\}} K_{m,\vec{k}}(t) \rho(0) K_{m,\vec{k}}^{\dagger}(t). \tag{7.38}$$

The Kraus operators in the series are indexed by both a time-perturbative order m and a vector $\vec{k}$ corresponding to a sequence of Lindblad operators in the expansion of $\mathcal{A}_{\text{env}}^{(m)}$. The Kraus operators for the case (I) system are

$$K_{m,\vec{k}}^{(\text{I})}(t) = e^{\frac{-it}{\hbar} V_H} \sqrt{\frac{t^m e^{-\frac{ct^2}{2}}}{m!}} \prod_{j=1}^{m} \left(\sqrt{\gamma_{\vec{k}_j}} L_{\vec{k}_j} \right). \tag{7.39}$$

In the case of a single Lindblad operator L_1, the index vector $\vec{k}$ can be removed and the Kraus operators reduce to the much simpler form

$$K_m^{(\mathrm{I})}(t) = e^{\frac{-it}{\hbar}V_H}\sqrt{\frac{(\gamma_1 t)^m e^{-\frac{ct^2}{2}}}{m!}}L_1^m \tag{7.40}$$

for $m = 0, 1, 2, ...$ etc.

7.3.2.2 Case (II) Kraus representation. For case (II) systems, we apply the commutation relation (7.30) and observe that $\mathcal{C}_m \propto t^m\mathcal{L}$ for all commutator exponent terms $m > 2$ in the Zassenhaus expansion. In particular, it can be shown that (7.25) collapses to

$$e^{t(\mathcal{H}+\mathcal{L})} = e^{t\mathcal{H}}e^{(t+\alpha g(t,\alpha))\mathcal{L}}e^{cg(t,\alpha)}, \tag{7.41}$$

so we obtain

$$\mathcal{A}_{\mathrm{env}}(t) = e^{(t+\alpha g(t,\alpha))\mathcal{L}}e^{cg(t,\alpha)}, \tag{7.42}$$

where $g(t,\alpha)$ (derived in [197]) has the closed form

$$g(t,\alpha) = -\frac{e^{-\alpha t} + \alpha t - 1}{\alpha^2}. \tag{7.43}$$

Following the same procedure as in case (I), we expand $\mathcal{A}_{\mathrm{env}}(t)$ as the series

$$\mathcal{A}_{\mathrm{env}}(t) = \sum_{m=0}^{\infty}\frac{(t+\alpha g(t,\alpha))^m}{m!}\mathcal{A}_{\mathrm{env}}^{(m)}(t), \tag{7.44}$$

where

$$\mathcal{A}_{\mathrm{env}}^{(m)}(t) = e^{cg(t,\alpha)}\sum_{\vec{k}\in\{1,2,...,N_L\}^m}\left(\prod_{j=1}^{m}\left[\sqrt{\gamma_{\vec{k}_j}}L_{\vec{k}_j}\right]\right)^{\overline{\otimes}2}. \tag{7.45}$$

After simplifying each term, we obtain a Kraus series of the form (7.38) where the Kraus operators $K_{m,\vec{k}}^{(\mathrm{II})}(t)$ are given by

$$K_{m,\vec{k}}^{(\mathrm{II})}(t) = e^{\frac{-it}{\hbar}V_H}\sqrt{\frac{(1-e^{-\alpha t})^m e^{cg(t,\alpha)}}{\alpha^m\ m!}}\prod_{j=1}^{m}\left(\sqrt{\gamma_{\vec{k}_j}}L_{\vec{k}_j}\right). \tag{7.46}$$

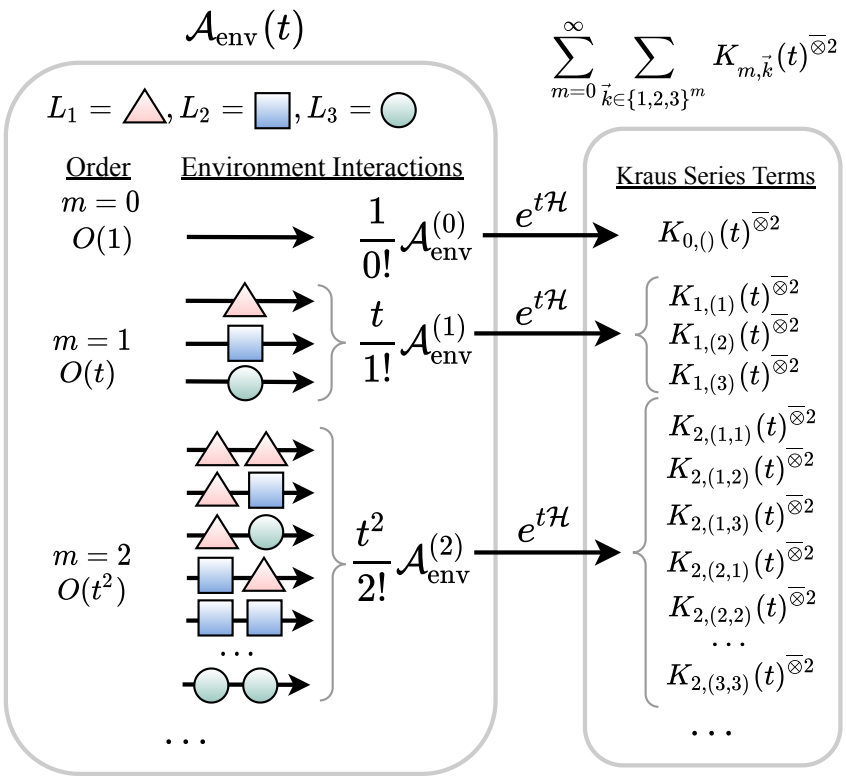

Figure 7.4: Illustration of the mapping between the environment action superoperator $\mathcal{A}_{\text{env}}(t)$ and the Kraus operator series of the form (7.38) for systems satisfying the superoperator commutation relation (7.28). In the environmental interaction picture, each sequence of m Lindblad operators (which models a specific mode of environmental interaction) corresponds to an order m term in a Kraus series that evolves the system in the Schrödinger picture.

In the case of a single Lindblad operator L_1, the Kraus operators reduce to

$$K_m^{(\text{II})}(t) = e^{\frac{-it}{\hbar}V_H}\sqrt{\frac{\gamma_1^m(1-e^{-\alpha t})^m e^{cg(t,\alpha)}}{\alpha^m\ m!}}L_1^m \tag{7.47}$$

for $m = 0, 1, 2, ...$ etc.

7.3.3 Interpreting and Simplifying the Kraus Series

We have derived a time-perturbative Kraus series representation of the system where each Kraus operator $K_{m,\vec{k}}(t)$ can be interpreted as the evolution of an ordered

sequence of Lindblad operators (i.e. $\prod_i L_{\vec{k}_i} = L_{\vec{k}_1} L_{\vec{k}_2} ... L_{\vec{k}_m}$) under the system's effective Hamiltonian. For the $m = 0$ Kraus operator, this sequence is empty and thus corresponds to evolution under the effective Hamiltonian only:

$$K_0(t) = e^{-itV_H/\hbar}. \tag{7.48}$$

The remaining Kraus operators contain nonempty sequences of $m > 0$ Lindblad operators. Each sequence can be interpreted as a mode of environmental interaction in which a number of physical processes occur in a specific order, such as particle exchange with the environment, environment-mediated transitions in state, or the creation and subsequent annihilation of virtual particles within the system. Through this lens, the terms of the Kraus series can be viewed in a manner analogous to the particle-path interpretation that arises from the time-perturbative treatment of a scattering process [199]. However, accounting for all order m modes of interaction in a system with many Lindblad operators requires a number of Kraus operators that grows exponentially with m unless one can simplify the Kraus series using commutator and product identities associated with the Lindblad operators.

The most straightforward simplification case to consider is when the Lindblad operators commute (i.e. $[L_i, L_j] = 0$, or more generally $[L_i^{\overline{\otimes}2}, L_j^{\overline{\otimes}2}] = 0$). This allows for the set of index vectors $\vec{k}$ to be grouped into equivalent sets according the number of times each Lindblad operator appears in the sequence $\prod_i L_{\vec{k}_i}$. Letting $\vec{\ell}$ denote a representative from each of these sets of $\vec{k}$ vectors for every order m, the Kraus series can be reduced to

$$\vec{\rho}(t) = \sum_{m=0}^{\infty} \sum_{\vec{\ell}} c_{m,\vec{\ell}} K_{m,\vec{l}}^{\overline{\otimes}2} \vec{\rho}, \tag{7.49}$$

where $c_{m,\ell}$ is a multinomial coefficient

$$c_{m,\ell} = \binom{m}{n_1(\vec{\ell}), n_2(\vec{\ell}), ..., n_{N_L}(\vec{\ell})} \tag{7.50}$$

which counts the number of ways $n_1(\vec{\ell}), n_2(\vec{\ell}), ..., n_{N_L}(\vec{\ell})$ instances of the respective operators $L_1, L_2, ..., L_{N_L}$ can be ordered to create an equivalent product sequence of length $m = n_1(\vec{\ell}) + n_2(\vec{\ell}) + ... + n_{N_L}(\vec{\ell})$. This reduces the number of Kraus series terms of each order m by a factor roughly proportional to $(N_L)^m/m!$ for $m > N_L$.

The time-perturbative Kraus series can also be reduced by incorporating product identities of the form $L_i L_j = L_k$. This most commonly occurs when a system's Lindblad operators form a multiplicative group up to multiplication by a constant. When the Lindblad operators are either nilpotent or are endowed with a finite group structure, it is possible to derive very useful power formulas of the form $L_i^n = \theta I$ (or more generally $(L_i^{\overline{\otimes} 2})^n = |\theta|^2 I^{\overline{\otimes} 2}$), where θ is some constant and I is the identity operator. These can be used to reduce the Kraus series to a finite number of terms up to a maximum order n. For example, with case (I) systems containing a single Lindblad operator L_1 satisfying $L_1^n = \theta I$, we can reduce the the infinite series of Kraus operators of the form (7.40) down to the finite sum of Kraus operators

$$K_m^{(I)}(t) = e^{\frac{-it}{\hbar} V_H} \sqrt{F_{n,m}^{\theta}(\gamma_1 t) e^{-\frac{ct^2}{2}}} L_1^m \tag{7.51}$$

for $m = 0, 1, 2, ..., n-1$. A similar form can be found for case (II) systems. Above, $F_{n,m}^{\theta}(x)$ denotes the generalized hyperbolic function

$$F_{n,m}^{\theta}(x) = \sum_{k=0}^{\infty} \frac{\theta^k}{(nk+m)!} x^{nk+m}, \tag{7.52}$$

which can be computed in closed form (following the method in [200]) using the formula

$$F^{\theta}_{n,m}(x) = \frac{1}{n}\theta^{-\frac{m}{n}} \sum_{k=0}^{n-1} \omega_n^{-mk} \exp\Big(\omega_n^k \theta^{\frac{1}{n}} x\Big), \tag{7.53}$$

where $\omega_n = e^{i2\pi/n}$ is an n-th primitive root of unity.

7.3.4 Representing Kraus Operators as Quantum Circuits

So far, we have shown that open quantum systems satisfying the commutation relation (7.29) or (7.30) admit a Kraus series representation, yet it remains to be discussed how the representations can be realized as quantum circuits. Since the Kraus operators $K_i(t)$ for a system are generally non-unitary, they must first be expanded to unitary operators to be implemented on a quantum computer. These expanded unitary operators can be viewed as acting on a representation of the system plus an ancillary environmental subsystem. This expansion is usually achieved though either a Stinespring dilation or a Sz.-Nagy dilation of the Kraus operators.

7.3.4.1 Stinespring dilations. The most common of unitary dilation methods is the Stinespring dilation [201], which uses the total unitary evolution of a set of system qubits which become coupled with a set of environmental qubits. It represents the total system as the tensor product of the system and environmental degrees of freedom, and requires finding a unitary $U_{\text{tot}}(t)$ that satisfies

$$\text{Tr}_{\text{env}}(U_{\text{tot}}(t)(\rho_{\text{sys}} \otimes \rho_{\text{env}})U_{\text{tot}}(t)^{\dagger}) = \sum_i K_i(t)\rho_{\text{sys}}K_i(t)^{\dagger}, \tag{7.54}$$

where $\rho_{\text{env}} = |\phi\rangle \langle\phi|_{\text{env}}$ is a pure state of rank one and ρ_{sys} is an arbitrary initial state.

If the initial state ρ_{sys} is pure (i.e. $\text{Tr}\big(\rho^2_{\text{sys}}\big) = 1$), we can decompose $\rho_{\text{sys}} = |\psi\rangle \langle\psi|_{\text{sys}}$ and simulate the evolution of the entire system by evolving $|\psi\rangle_{\text{sys}}$ only. This

is done by applying $U_{\text{tot}}(t)$ to the system state $|\psi\rangle_{\text{sys}}$ and environment state $|\phi\rangle_{\text{env}}$ as shown in Figure 7.5, after which quantities such as state occupation probabilities or the expectation values of observables can be estimated by measuring the system qubits only. This effectively traces out the environment subsystem in a manner that is consistent with (7.54).

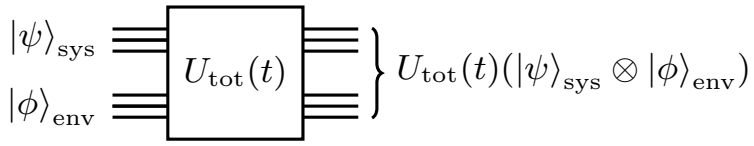

Figure 7.5: Quantum circuit diagram of a Stinespring dilation circuit simulating a system initialized in the pure state $|\psi\rangle_{\text{sys}}$.

In the case that ρ_{sys} is not pure, we can still simulate its evolution by diagonalizing it in the form

$$\rho_{\text{sys}} = \sum_j p_j \, |\psi_j\rangle \, \langle\psi_j|_{\text{sys}} \tag{7.55}$$

and then preparing a purification of ρ_{sys} via a set of ancillary prep qubits indexed by an orthonormal basis $|e_j\rangle$:

$$\rho_{\text{sys}} \sim \sum_j \sqrt{p_j}(|\psi_j\rangle_{\text{sys}} \otimes |e_j\rangle_{\text{prep}}). \tag{7.56}$$

This prepares a superposition of $|\psi_j\rangle_{\text{sys}}$ states that can used in place of $|\psi\rangle_{sys}$ in circuits such as Figure 7.5.

7.3.4.2 Sz.-Nagy dilations. Unlike a Stinespring dilation, a Sz.-Nagy dilation represents the total system as the direct sum of the system and the environmental degrees of freedom for each individual Kraus operator [202]. The unitary Sz.-Nagy

dilation of a Kraus operator $K_i(t)$ is given by

$$U_{K_i}(t) = \begin{pmatrix} K_i(t) & D_{K_i^\dagger}(t) \\ D_{K_i}(t) & -K_i(t)^\dagger \end{pmatrix} \tag{7.57}$$

where $D_{K_i}(t)$ is the defect operator of $K_i(t)$, defined as

$$D_{K_i}(t) = \sqrt{I - K_i(t)^\dagger K_i(t)}. \tag{7.58}$$

Representing U_{K_i} on a quantum computer requires only a single ancillary qubit. This qubit distinguishes the system degrees of freedom from the environmental degrees of freedom by its measured value ($|0\rangle$ for the system, and $|1\rangle$ for the environment). To compute expectation values of system observables $\hat{O}_{\text{sys}}$ with respect to the system's final state $\rho(t)$, one must construct a quantum circuit representation of $U_{K_i}(t)$ for each Kraus operator $K_i(t)$ as shown in Figure 7.6 and then compute the corresponding expectation values for each circuit while filtering out any $|1\rangle$ measurements in the ancillary qubits. This is done by defining the filtered observable

$$\hat{O}' = (\hat{O}_{\text{sys}} \otimes |0\rangle \langle 0|_{\text{anc}}) \tag{7.59}$$

and measuring the expectation value $\langle \hat{O}' \rangle_{K_i(t)}$ on both the system and Sz.-Nagy ancillary qubits for each Kraus circuit K_i. The measured expectation values for each Kraus operator circuit are then added together (on a classical computer) to recover the system expectation value

$$\langle \hat{O}_{\text{sys}} \rangle_{\rho(t)} = \sum_i \langle \hat{O}' \rangle_{K_i(t)}, \tag{7.60}$$

where $\langle O' \rangle_{K_i(t)}$ is defined as

$$\langle O' \rangle_{K_i(t)} = \text{Tr}\left(\hat{O}' U_{K_i(t)} (\rho(0) \otimes |0\rangle \langle 0|_{\text{anc}}) U_{K_i(t)}^\dagger \right). \tag{7.61}$$

If any ancilla qubits are used aside from those needed to realize the Sz.-Nagy dilations contained in a circuit (for instance, qubits used to prepare the purification of the mixed state in (7.56)) these qubits are ignored and not measured in the evaluation of $\langle \hat{O}' \rangle_{K_i(t)}$.

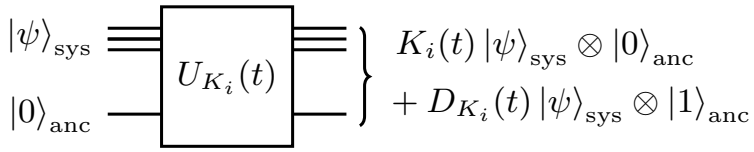

Figure 7.6: Quantum circuit diagram of a Sz.-Nagy dilation quantum circuit for the Kraus operator $K_i(t)$.

A key advantage of using Sz.-Nagy dilations is the lower representation complexity. Indeed, it is known that if the representation of a given Kraus operator is minimal, both the minimality and uniqueness (up to a unitary transform) of the representation $U_{K_i}(t)$ is guaranteed, per the Sz.-Nagy dilation theorem [202]. As a result, Sz.-Nagy dilations typically require fewer two-qubit entangling gates to implement in comparison to Stinespring dilations, as the dimension of the dilated unitaries in the form of (8.14) tend to be orders of magnitude less than those of a $U_{tot}(t)$ unitary that explicitly models environmental interactions. The disadvantage of using Sz.-Nagy dilations, however, is that they split the system dynamics over many quantum circuits, requiring one circuit per Kraus operator, which may be undesirable for systems with many Kraus operators. On NISQ devices, we argue that Sz.-Nagy dilations are preferable to Stinespring dilations, due to their lower gate complexity, which mitigates the effect of simulation noise, thereby allowing for more complex systems to be simulated.

7.3.4.3 Representation of case (I) and case (II) Kraus operators.. Now we will focus on the problem of representing the Kraus operators for the special cases (I) and (II), for which we derived the closed forms of (7.39) and (7.46) respectively. A key assumption we have made for these systems (as stated in condition 2 in the Introduction) is that the effective Hamiltonian dynamics (i.e. $\exp(t\mathcal{H})$) can be efficiently represented as a quantum circuit of t-independent gate complexity. In light of (7.14), this means that it is an equivalent condition for $e^{\frac{-it}{\hbar}V_H}$ to be efficiently realizable with t-independent gate complexity. One such realization is given by the Sz.-Nagy dilation

$$U_{\mathcal{H}}(t) = \begin{pmatrix} e^{\frac{-it}{\hbar}V_H} & D_{\exp\left(\frac{-it}{\hbar}V_H\right)^\dagger} \\ D_{\exp\left(\frac{-it}{\hbar}V_H\right)} & -(e^{\frac{-it}{\hbar}V_H})^\dagger. \end{pmatrix} \tag{7.62}$$

Now that we have addressed how $U_{\mathcal{H}}(t)$ can be represented, we can realize a full Kraus operator $K_{m,\vec{k}}(t)$ of the form (7.39) or (7.46) as a quantum circuit by first applying a sequence of Sz.-Nagy dilations $U_{A_{\vec{k}_i}}$ (each of which apply a re-scaled form $A_{\vec{k}_i}$ of the respective Lindblad operators $L_{\vec{k}_i}$) followed by $U_{\mathcal{H}}$. The general layout of this circuit is shown in Figure 7.7.

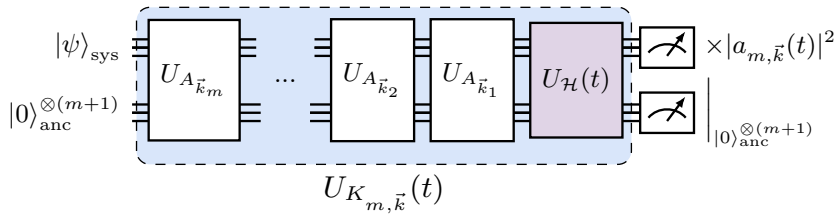

Figure 7.7: Quantum circuit diagram for the general multi-step implementation of the case (I) and case (II) Kraus operators $K_{m,\vec{k}}(t)$. First, a sequence of scaled Lindblad operators A_i are applied according to the index ordering $i = \vec{k}_m, \vec{k}_{m-1}, ..., \vec{k}_1$. Finally, $U_{\mathcal{H}}$ is applied and system and ancilla qubits are measured. When the ancillas all read $|0\rangle$ the measured probabilities (or expectation values) are re-scaled by $|a_{m,\vec{k}}(t)|^2$. If any of the ancillas are not $|0\rangle$, then the probability of that state is discarded.

The re-scaling of the Lindblad operators is necessary, since it is not generally guaranteed to be the case that $\|L_i\| \leq 1$. Thus, we define the re-scaled Lindblad operators as

$$A_i = \frac{L_i}{\|L_i\|} \tag{7.63}$$

and use them to construct the Sz.-Nagy dilations U_{A_i}, as shown in Figure 7.7. This can be done with standard unitary decomposition methods. While scaling is necessary to make the L_i operators realizable as quantum circuits, it requires that any measured observables under the final wave function must be subsequently unscaled by a squared factor ($|a_{m,\vec{k}}(t)|^2$) to obtain the correct value. The factor $a_{m,\vec{k}}(t)$ is time-dependent, because we also use it to incorporate all of the remaining time-dependent scalar values contained in the closed-form Kraus operators. Specifically, for the general case (I) Kraus operators of the form (7.39), we use

$$a^{(\mathrm{I})}_{m,\vec{k}} = \sqrt{\frac{t^m e^{-\frac{ct^2}{2}}}{m!}} \prod_{j=1}^{m} \sqrt{\gamma_{\vec{k}_j}} \left\| L_{\vec{k}_j} \right\|, \tag{7.64}$$

and likewise for the case (II) Kraus operators of the form (7.46), we use

$$a^{(\mathrm{II})}_{m,\vec{k}} = \sqrt{\frac{(1-e^{-\alpha t})^m e^{-cg(t,\alpha)}}{\alpha^m m!}} \prod_{j=1}^{m} \sqrt{\gamma_{\vec{k}_j}} \left\| L_{\vec{k}_j} \right\|. \tag{7.65}$$

When the expectation values of a filtered observable $\hat{O}'$ are measured for each of the Kraus circuits $U_{K_{m,\vec{k}}}(t)$, the results must be recombined with weights $|a_{m,\vec{k}}(t)|^2$ on a classical computer to recover the time-dependent expectation value of the original system observable:

$$\langle \hat{O}_{\mathrm{sys}} \rangle_{\rho(t)} = \sum_m \sum_{\vec{k}} |\alpha_{m,\vec{k}}(t)|^2 \langle \hat{O}' \rangle_{U_{K_{m,\vec{k}}}(t)}. \tag{7.66}$$

7.3.4.4 Representations of systems satisfying commutation relations (i)-(iv). So far, we have shown in a general way how the class of systems satisfying conditions

1-3 admit a time-perturbative Kraus series representation where each Kraus operator can be simulated with circuits of t-independent complexity. We will now focus on a subclass of these systems, which one can recognize immediately without analyzing the superoperator dynamics of $\mathcal{H}$ and $\mathcal{L}$. These are the systems which satisfy the commutation relations (i)-(iv) proposed in the introduction, which we will now prove satisfy conditions 1-3.

First, we will establish that a system satisfying the commutation relations (iii) and (iv) satisfies the commutation relation in condition 3. We expand $[\mathcal{H}, \mathcal{L}]$ in terms of V_H and the V_n operators defined in (7.8) and (7.9), and we obtain

$$[\mathcal{H}, \mathcal{L}] = \sum_n \frac{-i}{\hbar} \left([V_H, V_n] \otimes \overline{V_n} - V_n \otimes \overline{[V_H, V_n]} \right). \tag{7.67}$$

After expanding $[V_H, V_n]$ and applying (iii) and (iv), we obtain

$$[V_H, V_n] = \sum_{n'} \sqrt{\gamma_n} \left([H, L_n] + \frac{-i\hbar\gamma_{n'}}{2} [L_{n'}^\dagger L_{n'}, L_n] \right) \tag{7.68}$$

$$= \sqrt{\gamma_n} \left(\nu L_n + \frac{-i\hbar\lambda}{2} L_n \right) \tag{7.69}$$

$$= (\nu - i\hbar\lambda/2) V_n. \tag{7.70}$$

Substituting (7.70) into (7.67) and simplifying yields the commutation relation

$$[\mathcal{H}, \mathcal{L}] = (2\,\mathrm{Im}[\nu]/\hbar - \mathrm{Re}[\lambda])\mathcal{L} \tag{7.71}$$

which shows that the system satisfies condition 3 with $\alpha = 2\,\mathrm{Im}[\nu]/\hbar - \mathrm{Re}[\lambda] \geq 0$ and $c = 0$.

With regards to condition 2, we argue that as long as a system's Lindblad operators L_n can be truncated to a bounded matrix form, then the L_n can be re-scaled per (7.63), and can be subsequently realized as a dilated unitary operation.

This satisfies condition 2. The only condition that remains to be shown is condition 1, namely that $e^{t\mathcal{H}}$ can be simulated using a circuit of t-independent gate complexity. Using (7.14), we have already shown that it suffices to show $\exp(-itV_H/\hbar)$ can be realized via the Sz.-Nagy dilation (7.62) with t-independent gate complexity, which we will proceed to prove. From (i) and (ii) it immediately follows that V_H commutes with its Hermitian adjoint ($[V_H, V_H^\dagger] = 0$) and hence is a normal operator. This means that $\exp\left(\frac{-it}{\hbar}V_H\right)$ can be diagonalized in the form

$$e^{\frac{-it}{\hbar}V_H} = UW(t)\Lambda(t)U^\dagger, \tag{7.72}$$

where $W(t)$ and $\Lambda(t)$ are diagonal matrices with entries of the form $e^{-it\omega_i}$ and $e^{-t\lambda_i}$ respectively for real scalars ω_i, λ_i. From the positive semi-definiteness of all $L_n^\dagger L_n$ terms in V_H, we also know that $\lambda_i \geq 0$ for all i (which reflects the fact that $\exp\left(\frac{-it}{\hbar}V_H\right)$ is a contraction). Since $W(t)$ and $\Lambda(t)$ are diagonal, they can easily be implemented by parameterized quantum circuits. $W(t)$ is unitary, which means no ancilla qubits are required to realize it, whereas $\Lambda(t)$ must be implemented through a Sz.-Nagy dilation $U_{\Lambda(t)}$. Finally, the diagonalizing unitary U can be implemented through a number of unitary decomposition methods such as the Quantum Shannon decomposition [203], which can decompose an N-qubit unitary into a quantum circuit with at most $O(4^N)$ gates. We show the full quantum circuit diagram for realizing $U_{\mathcal{H}}(t)$ in Figure 7.8.

Since the parameterized unitaries $W(t)$ and $U_\Lambda(t)$ have 2^N degrees of freedom for N qubits, their circuit implementations must contain at least 2^N parameterized gates, ideally in such a manner that computing the gate parameters for a specific time t is computationally efficient. In Figure 7.9 we show a quantum circuit that implements $W(t)$ using a vector of 2^N real parameters $\vec{\theta} = (\theta_0\ \theta_1\ ...\ \theta_{2^N-1})^T$ corresponding to

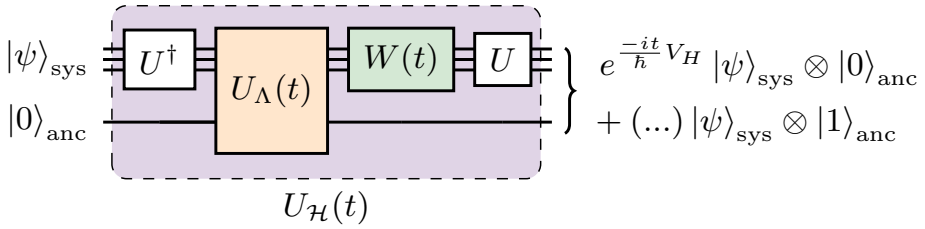

Figure 7.8: Quantum circuit diagram for computing $U_{\mathcal{H}}(t)$ when the commutation relations (i) and (ii) are satisfied and $\exp\left(\frac{-it}{\hbar}V_H\right)$ admits the decomposition (7.72). Since $W(t)$ and $\Lambda(t)$ are diagonal, they can easily be represented by parameterized quantum circuits such as those shown in Figure 7.9 and Figure 7.10 respectively.

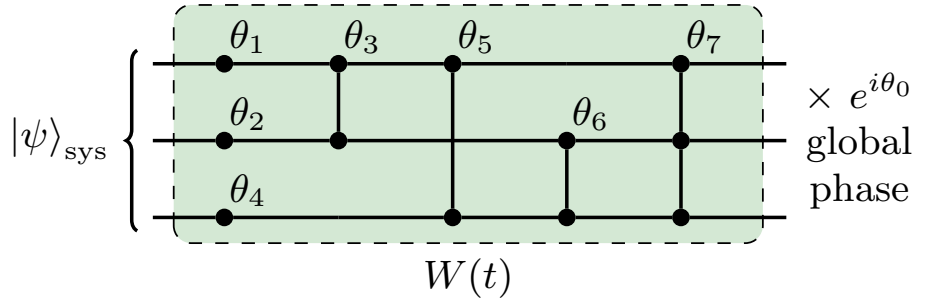

Figure 7.9: Quantum circuit diagram for an arbitrary $N = 3$ qubit diagonal unitary operator $W(t)$ implemented through a sequence of controlled phase gates. Since the global phase factor $e^{i\theta_0}$ has no effect on measurement outcomes, the parameter θ_0 is often ignored.

controlled phase gates, where the control qubits of each θ_n phase gate are the "1" bits of the binary representation of n. The parameters $\vec{\theta}$ that realize a $W(t)$ with diagonal elements of the form $e^{-it\omega_n}$ can be computed according to the matrix equation

$$\vec{\theta}(t) = tQ_N\vec{\omega} \tag{7.73}$$

where $\vec{\omega} = (\omega_0\ \omega_1\ ...\ \omega_{2^N-1})^T$ and Q_N is the N-qubit parameter transform matrix, given by:

$$Q_N = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}^{\otimes N}. \tag{7.74}$$

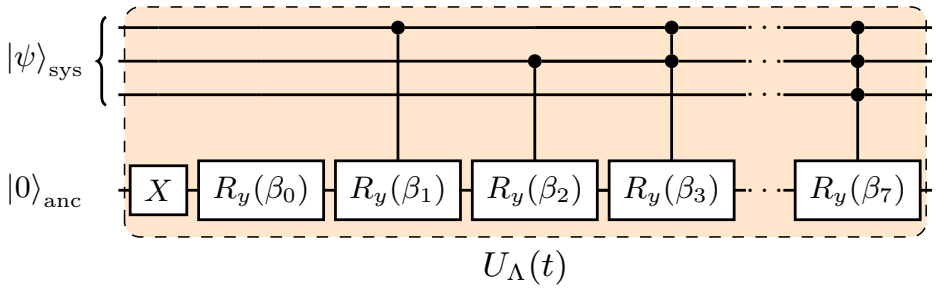

Figure 7.10: Quantum circuit diagram for the Sz.-Nagy dilation of an arbitrary 3-qubit diagonal contraction operator $\Lambda(t)$. It is implemented through a sequence of controlled R_y gates on the ancilla qubit. Here, $R_y(\beta) \equiv \exp\left(\frac{-i\beta}{2}Y\right)$, where Y is the Pauli Y operator.

In Figure 7.10, we show a similar quantum circuit that implements $U_\Lambda(t)$ by performing parameterized controlled R_y gates on a single ancilla qubit, with parameter vector $\vec{\beta} = (\beta_0 \ \beta_1 \ ... \ \beta_{2^N-1})^T$. In a manner similar to (7.73), we can obtain the parameter vector $\vec{\beta}(t)$ corresponding to the diagonal elements $\vec{v}(t)$ of $\Lambda(t)$ by computing

$$\vec{\beta}(t) = -2Q_N \arcsin(\vec{v}(t)). \tag{7.75}$$

The time-dependent parameter mappings of both $\vec{\theta}(t)$ in (7.73) and $\vec{\beta}(t)$ in (7.75) require a $2^N \times 2^N$ classical matrix multiplication by Q_N to determine the parameters for $W(t)$ and $\Lambda(t)$. While it may appear that the complexity of this parameter mapping might dominate the complexity of simulating a large system on a quantum computer, we emphasize that Q_N can be factored as a tensor power of a 2×2 matrix, and thus the multiplication of Q_N times a vector can in fact be carried out efficiently in $O(2^N \log(2^N))$ using a divide-and-conquer algorithm similar to the fast Walsh-Hadamard transform [204]. We give additional details regarding this parameter mapping process in Appendix D.1.

7.3.5 *Analysis of Accuracy and Time Complexity*

In this section, we combine the results of our previous analysis and derive bounds for time complexity, as well as provide some rough analysis of the error robustness of the proposed realization of systems satisfying conditions 1-3.

7.3.5.1 Accuracy of the Kraus series approximation. First, we analyze the convergence of the general time-perturbative Kraus series representations for case (I) and case (II) in the general case of a system with N_L Lindblad operators. We recall that the Kraus series (in superoperator form) is given by (7.37). For both case (I) and case (II) systems, the terms corresponding to each m in this series can be written in the factored form

$$\sum_{\vec{k}\in\{1,2,\ldots,N_L\}^m} K_{m,\vec{k}}(t)^{\overline{\otimes}2} = e^{t\mathcal{H}} h(t) \frac{(f(t)\mathcal{L})^m}{m!} \tag{7.76}$$

where $h(t)$ and $f(t)$ are both real non-negative functions of time. For case (I) and (II) systems, they take the form

$$h(t) = \begin{cases} e^{-ct^2/2}, & \text{Case (I)} \\ e^{cg(t,\alpha)}, & \text{Case (II)} \end{cases} \tag{7.77}$$

$$f(t) = \begin{cases} t, & \text{Case (I)} \\ (1-e^{-\alpha t})/\alpha, & \text{Case (II)} \end{cases}. \tag{7.78}$$

Since we have assumed $\mathcal{L}$ is bounded and admits a sparse matrix representation that can be realized with a quantum circuit (a consequence of condition 2 being satisfied), then the Kraus series converges pointwise for every $t \geq 0$. Moreover if $f(t)$ and $h(t)$ are bounded, then the series also converges uniformly for $t \geq 0$. Each term in the right-hand side of (7.76) corresponds to a term in a Taylor series approximation to

the exponential of $\mathcal{L}$. In Appendix D.2, we show that if the Kraus series is evaluated up to a finite order $m = M$ yielding an approximation $\vec{\rho}_M(t)$, the error with respect to the true time evolution $\vec{\rho}(t)$ can be bounded by

$$\|\vec{\rho}(t) - \vec{\rho}_M(t)\| \leq \left(\left\|e^{t\mathcal{H}}\right\| h(t)\eta(t)\right) \frac{(f(t)\,\|\mathcal{L}\|_{\text{HS}})^{M+1}}{(M+1)!}, \tag{7.79}$$

where $\eta(t)$ is given by

$$\eta(t) = \left(1 - \frac{f(t)\,\|\mathcal{L}\|_{\text{HS}}}{(M+1)}\right)^{-1}, \tag{7.80}$$

and $\|A\|_{\text{HS}} = \sqrt{\text{Tr}(A^\dagger A)}$ denotes the Hilbert-Schmidt norm (also known as the Frobenius norm for matrices). It can be shown that $\|\mathcal{L}\|_{\text{HS}}$ is bounded by

$$\|\mathcal{L}\|_{\text{HS}} = \left\|\sum_n \gamma_n L_n^{\overline{\otimes}2}\right\|_{\text{HS}} \leq \sum_n \gamma_n \|L_n\|_{\text{HS}}^2 \tag{7.81}$$

and that $\left\|e^{t\mathcal{H}}\right\|$ is bounded by

$$\left\|e^{t\mathcal{H}}\right\| \leq Ce^{-t\lambda_L} \leq C \tag{7.82}$$

where C is some constant and λ_L is the smallest eigenvalue of $(\sum_n \gamma_n L_n^\dagger L_n)$. For systems satisfying conditions (i)-(iv), $C = 1$ due to the normality of V_H.

Using (7.79), we can derive a rough lower bound on M needed to guarantee that $\|\vec{\rho}(t) - \vec{\rho}_M(t)\| \leq \varepsilon$ for a desired error tolerance ε. In Appendix D.2 we show that if M satisfies the inequalities

$$M \geq e(f(t)\,\|\mathcal{L}\|_{\text{HS}})^{1+\delta}, \tag{7.83}$$

(where $e \approx 2.72$) and

$$M \geq \frac{\log\left(\left\|e^{t\mathcal{H}}\right\| h(t)/\varepsilon\right)}{\delta \log(f(t)\,\|\mathcal{L}\|_{\text{HS}})} \tag{7.84}$$

for any $\delta > 0$, then $\|\vec{\rho}(t) - \vec{\rho}_M(t)\| \leq \varepsilon$. In the limit of $\delta \ll 1$ we observe that the number of terms M in the approximation scales almost linearly with $f(t)\,\|\mathcal{L}\|_{\text{HS}}$. For a fixed t and δ, we observe logarithmic scaling with respect to $1/\varepsilon$, which is also

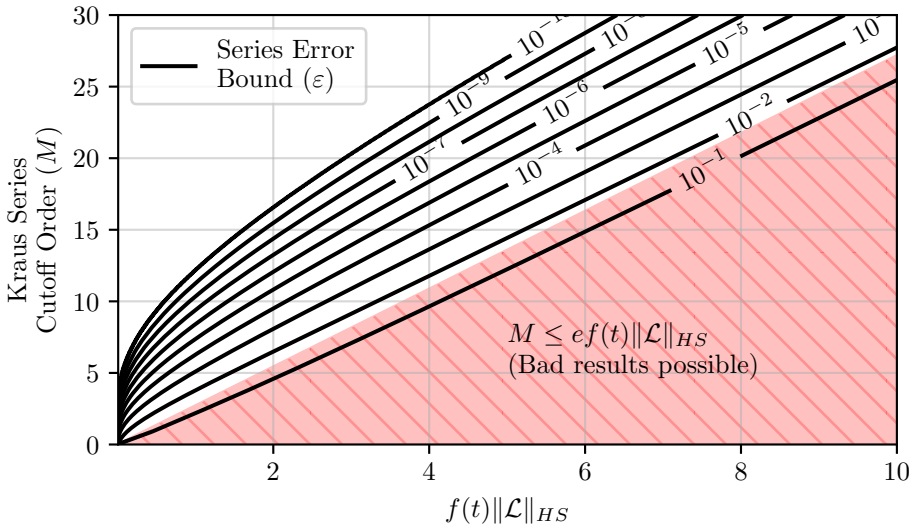

Figure 7.11: Contour plot of the series error bound ε with respect to M and $(f(t)\,\|\mathcal{L}\|_{\text{HS}})$ for the "worst case" values of $h(t) = \left\|e^{t\mathcal{H}}\right\| = 1$. For small values of $f(t)\,\|\mathcal{L}\|_{\text{HS}}$, M exhibits almost-linear scaling of the form $M \propto (f(t)\,\|\mathcal{L}\|_{\text{HS}})$, provided that M at least satisfies (7.83) and (7.84) for a small δ. If a cutoff order of $M < ef(t)\,\|\mathcal{L}\|_{\text{HS}}$ is chosen (thereby violating (7.83)), the truncated series is not guaranteed to give accurate results.

desirable. In Figure 7.11, we plot the contours of the error bound ε in (7.79) for small values of $f(t)\,\|\mathcal{L}\|_{\text{HS}}$ and verify that the approximately linear scaling of M with respect to $f(t)\,\|\mathcal{L}\|_{\text{HS}}$ holds in this regime.

For case (I) systems, the Kraus operators can be written in the form (7.76) with $f(t) = t$ and $h(t) = e^{-ct^2/2}$. Since $f(t)$ is linear and unbounded, our analysis suggests that selecting $M \propto t$ is sufficient in the worst case to yield good asymptotic accuracy with a fixed ε; however if $c > 0$ or $\lambda_L > 0$ then one only needs to scale M linearly with respect to t until the point at which $Ce^{-\lambda_L t}h(t) \ll \varepsilon$, which gives the asymptotic bound $M \sim O(1)$. For case (II) systems, we have $f(t) = (1 - e^{-\alpha t})/\alpha$ and $h(t) = e^{cg(t,\alpha)}$, which are both bounded by 1. This means that the case (II) Kraus

series converges uniformly and exponentially fast with respect to t and it suffices to select a constant cutoff $M \sim O(1)$.

To summarize, our analysis shows that for systems with $\alpha > 0$ or $c > 0$, the Kraus series converges quickly due to the rapid damping in the system, suggesting that the series cutoff M is asymptotically independent of t. However, in the "worst case" where $\alpha = c = \lambda_L = 0$, M must be selected in a manner that scales almost linearly with t.

7.3.5.2 Time complexity (general case). In the previous sections, we showed that each term in the Kraus series representation can be simulated by a circuit of the kind shown in Figure 7.7, which applies a sequence of m re-scaled Lindblad operators (realized as the unitaries $U_{A_{\vec{k}_i}}$) followed by the effective Hamiltonian time evolution (realized as $U_{\mathcal{H}}$). Using standard unitary decomposition methods, we argued that each operator L_n could be realized using $O(d^2)$ gates on an ideal quantum computer. Since NISQ devices are still far from ideal, we must justify that the $O(d^2)$ bound for unitary decomposition still holds under the constraints of NISQ quantum devices, such as a device having limited (but universal) native gate sets, or a device only allowing two-qubit interactions in accordance with the device's limited qubit topology. With regards to the first question, there is a substantial body of literature addressing it [129, 130, 170, 203, 205–208]. Perhaps the most celebrated of these results is the generalized Solovay-Kitaev algorithm [129,170], which from any circuit of n constant-qudit gates can construct a circuit with $O(n \log^c(n/\varepsilon))$ device-native gates up to approximation error ε. Recent improvements on this result have found a highly-efficient $c \leq 1.441$ [130]. With regards to the second question of topology-aware

decomposition of unitaries, there also exist many useful results [203, 207, 208]. One analysis has shown that even in the most restrictive linear qubit topology, the two-qubit gate cost for realizing arbitrary unitaries increases by at most a constant factor of 9 [203], thereby retaining the $O(d^2)$ asymptotic cost.

Now we will analyze the general time complexity of our method. If the effective Hamiltonian evolution $U_{\mathcal{H}}(t)$ can be realized with gate complexity $O(T_{\mathcal{H}}(d))$ in accordance with condition 1, we find that the total gate complexity of evaluating the entire Kraus series for N_L Lindblad operators up to order M is $\sum_{m=0}^{M}(N_L)^m(O(T_H(d)) + mO(d^2))$ in the most general case. This is asymptotically equivalent to

$$O((N_L)^M(T_{\mathcal{H}}(d) + Md^2)). \tag{7.85}$$

The prefactor $(N_L)^M$ shows a rather unforgiving exponential growth with respect to the number of Lindblad operators, though we suggest that this prefactor can be managed for systems with few Lindblad operators or systems with a Kraus series that can be simplified using the methods in Section 7.3.3. It is also worth commenting that the number of gates per circuit only grows as $O(T_{\mathcal{H}}(d) + Md^2)$, where we have shown the prefactor M to be asymptotically t-independent except in the rare case where the quantities λ_L, α, and c are all zero.

7.3.5.3 Time complexity (concrete case). Next, we consider the concrete case where the commutation relations (i)-(iv) are satisfied, and we derive asymptotic bounds for the time complexity $T_{\mathcal{H}}(t)$ of simulating $e^{t\mathcal{H}}$. For these systems, we found that V_H is normal and admits the parameterized circuit representation shown in Figure 7.8, where we proposed the circuit realizations of the diagonal unitaries $W(t)$ and $U_\Lambda(t)$ in Figure 7.9 and Figure 7.10 respectively. These circuits used a binary

encoding scheme and both make use of the fast N-qubit parameter transformation Q_N in (7.74), requiring (7.73) and (7.75) to be evaluated on a classical computer. These circuits use single-qubit unitaries with m control bits, each of which can be decomposed into $O(m)$ one-qubit and two-qubit gates, following the methods in [209]. This results in an overall gate complexity of $\sum_{n=1}^{2^N} O(\log(n)) = O(2^N \log(2^N))$ for N-qubit diagonal operators. For systems in a Hilbert space of dimension d, this means that the circuits for $W(t)$ and $U_\Lambda(t)$ have $O(d\log(d))$ gate complexity. Since the transform Q_N can also be computed in $O(2^N \log(2^N))$, the complexity of classically computing the time-dependent parameter mappings is also $O(d\log(d))$. Even though there exist more efficient representations of diagonal quantum operators requiring at most $O(d)$ gates [210], these representations use Gray code encodings of the target bits instead of a standard binary encoding, which have parameter transform matrices that cannot be factored as the Kronecker product of smaller matrices (e.g. the matrix $\eta^{\oplus}$ as introduced in [210]). In these methods, the classical computation of circuit parameters in $O(d^2)$ time dominates the $O(d)$ quantum gate complexity, which we consider to be less desirable than having a balanced $O(d\log(d))$ quantum and classical workload.

Since the sub-circuits for $W(t)$ and $U_\Lambda(t)$ in Figure 7.8 can both be realized with $O(d\log(d))$ quantum and classical complexity, the overall gate complexity of the system is limited by the quantum circuit implementing the unitary U that diagonalizes V_H. In general, this can be accomplished with $O(d^2)$ gates through a unitary decomposition of U, unless the system and its Lindblad operators are already transformed into a diagonalized basis (which is common for many kinds of simulations). If this is the case, the entire $U_{\mathcal{H}}(t)$ can be computed rather efficiently with time

complexity

$$T_{\mathcal{H}}(t) \sim O(d \log(d)) \tag{7.86}$$

for both the quantum and classical workloads. Otherwise, we have $T_{\mathcal{H}}(t) \sim O(d^2)$ in the general case.

7.4 Applications

7.4.1 The Multi-Qubit Continuous-Time Pauli Channel

The multi-qubit continuous-time Pauli channel is an important model that describes the dynamics of unbiased decoherence and noise in degenerate quantum systems, such as quantum information processors [211, 212] and coupled spin-1/2 particles [213]. The Pauli channel assumes a trivial Hamiltonian $H = 0$ and uses Lindblad operators consisting of N-qubit Pauli strings Π_n of the form

$$L_n = \Pi_n = (\sigma_{n1} \otimes \sigma_{n2} \otimes ... \otimes \sigma_{nN}), \tag{7.87}$$

where each σ_{ni} is one of the four single-qubit Pauli operators $\{I, X, Y, Z\}$. One important feature of Pauli strings is the fact that the conjugation of any operator by any two Pauli strings is commutative. Specifically, this means that

$$\Pi_n \Pi_{n'} \rho \Pi_{n'}^\dagger \Pi_n^\dagger = \Pi_{n'} \Pi_n \rho \Pi_n^\dagger \Pi_{n'}^\dagger \tag{7.88}$$

holds for any two Pauli strings $\Pi_n, \Pi_{n'}$ and any operator ρ. In addition, every Pauli string satisfies the identity $(\Pi_n^{\overline{\otimes} 2})^2 = I$. These two properties are important because they can be exploited to reduce the Kraus series representation of a multi-qubit continuous-time Pauli channel to a finite number of terms, as shown in Section 7.3.3. A proof of (7.88) is given in Appendix D.3.

Since $H = 0$ and $\Pi_n^\dagger \Pi_n = I$ for all Pauli strings, the system satisfies (i)-(iv) with $\lambda = \sum_n \gamma_n$ and $\nu = 0$, so $[\mathcal{H}, \mathcal{L}] = 0$, which results in a case (I) Kraus series representation. Although we could immediately write down the Kraus series terms in the form (7.39), we observe that (7.88) directly implies that $[V_n^{\overline{\otimes}2}, V_{n'}^{\overline{\otimes}2}] = 0$, which allows us to decompose the $e^{t\mathcal{L}}$ in (7.32) as the exponential of each Lindblad operator individually:

$$e^{t\mathcal{L}} = \exp\left(t \sum_{n=1}^{N_L} V_n^{\overline{\otimes}2}\right) = \prod_{n=1}^{N_L} e^{tV_n^{\overline{\otimes}2}}. \tag{7.89}$$

Since $(\Pi_n^{\overline{\otimes}2})^2 = I^{\overline{\otimes}2}$, we apply the hyperbolic identity $e^{t\Pi_n^{\overline{\otimes}2}} = I^{\overline{\otimes}2}\cosh(t) + \Pi_n^{\overline{\otimes}2}\sinh(t)$ and expand the right-hand side of (7.89) into

$$e^{t\mathcal{L}} = \prod_{n=1}^{N_L} [(\sqrt{\cosh(\gamma_n t)} I)^{\overline{\otimes}2} + (\sqrt{\sinh(\gamma_n t)} \Pi_n)^{\overline{\otimes}2}], \tag{7.90}$$

which we then substitute back into (7.32) and simplify to obtain a finite Kraus series of the form

$$e^{t(\mathcal{H}+\mathcal{L})} = \sum_{E \subseteq \{1,2,\ldots,N_L\}} K_E(t)^{\overline{\otimes}2}, \tag{7.91}$$

where each Kraus operator K_E is indexed by an "error set" E corresponding to one of the of the 2_L^N subsets of the Lindblad Pauli strings. The Kraus operators K_E take the form

$$K_E(t) = \sqrt{p_E(t)} \prod_{n \in E} \Pi_n, \tag{7.92}$$

where $p_E(t)$ can be interpreted as the probability for the subset of "Pauli errors" $\{\Pi_n\}_{n \in E}$ having occurred when the system's N qubits are measured at a time t. This probability $p_E(t)$ is given by

$$p_E(t) = \prod_{n=1}^{N_L} \frac{1 + (-1)^{\mathbb{I}_E(n)} e^{-\gamma_n t}}{2}, \tag{7.93}$$

where $\mathbb{I}_E(n) = 1$ when $n \in E$ and 0 otherwise.

The Kraus operators in (7.92) consist of time-dependent scalars $a_E(t) = \sqrt{p_E(t)}$ times products of Pauli strings. These products of Pauli strings are unitary and do not require a dilation to be realized efficiently on a quantum computer. In fact, by applying the product identities of Pauli operators, the product of Pauli strings $\prod_{n\in E} \Pi_n$ can be simplified to a single Pauli string $\Pi_E = \sigma_{E1} \otimes \sigma_{E2} \otimes ... \otimes \sigma_{EN}$ times a global phase of $e^{i\theta}$ which can be ignored. Each Kraus circuit U_{K_E} is thus time-independent and can be realized as shown in Figure 7.12, where the gate Π_E applies a Pauli gate σ_{Ei} to each qubit.

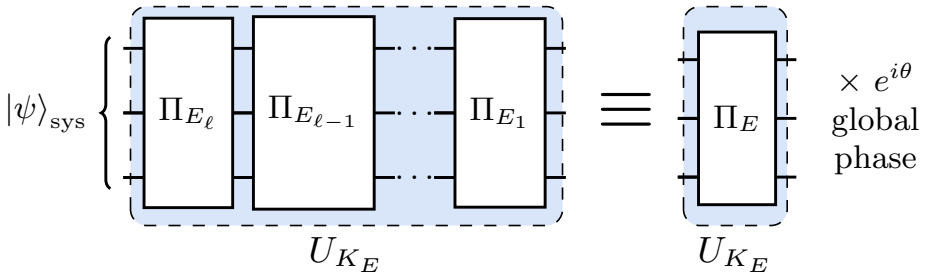

Figure 7.12: Kraus circuits for the multi-qubit continuous-time Pauli channel. Each sequence of Pauli strings indexed by the "Pauli error" set E can be reduced to a single Pauli string Π_E up to a global phase.

The Kraus circuits in Figure 7.12 require $O(\log(d))$ quantum gates, where $d = 2^N$ is the dimension of an N-qubit Pauli channel. The circuits are also parameterless, with all of the time-dependence of the circuit incorporated through the re-scaling constants $a_E(t) = \sqrt{p_E(t)}$. Remarkably, this means that it is possible to extrapolate the entire trajectory of observables to many different t values (both forward and backward in time) using only the results of each Kraus circuit applied to the initial state $\rho(0)$. This feature can be exploited in the NISQ error mitigation technique of

zero-noise extrapolation, where the result of a noisy quantum computation (modeled by a sparse Pauli channel [211]) can be evolved backward in time to suppress the effect of Markovian noise [214].

7.4.2 The Damped Quantum Harmonic Oscillator

The damped quantum harmonic oscillator is a foundational system that appears in many application areas in quantum mechanics, such as quantum optics [215] and quantum field theory [199]. This system is also being actively researched as a platform for fault-tolerant quantum information processing with bosonic codes [216, 217]. A quantum harmonic oscillator has a Hamiltonian of the form

$$H = \hbar\omega \left(\frac{1}{2} + \hat{N} \right), \tag{7.94}$$

where $\hat{N} = \hat{a}^\dagger \hat{a}$ is the state number operator and $\hat{a}^\dagger, \hat{a}$ are the raising and lowering operators of the system. If damping is applied to this system, the single Lindblad operator $L_1 = \hat{a}$ is used with damping γ_1. One can easily verify that this system satisfies the commutation relations (i)-(iv) with scalars $\nu = \hbar\omega$ and $\lambda = -\gamma_1$. It then follows that conditions 1-3 are satisfied, with $[\mathcal{H}, \mathcal{L}] = \gamma_1 \mathcal{L}$, resulting in a time-perturbative Kraus series representation of the form (7.47) with $\alpha = \gamma_1$ and $c = 0$. This simplifies to

$$K_m(t) = e^{-t[\frac{\gamma_1}{2}\hat{N} + i\omega(\frac{1}{2} + \hat{N})]} \sqrt{\frac{\gamma_1^m (1 - e^{-\gamma_1 t})^m}{m!}} \hat{a}^m, \tag{7.95}$$

which is consistent with prior derivations of a Kraus representation for this system [195].

Since $\hat{a}$ theoretically acts on an infinite number of modes of H, it is common to truncate $\hat{a}$ to act only on modes up to the highest initially occupied eigenstate $|m_{\max}\rangle$

in ρ_{sys}. This truncation makes $L_1 = \hat{a}$ representable as a $(m_{\max}+1) \times (m_{\max}+1)$ nilpotent matrix of index $m_{\max}+1$ (i.e. $L_1^{m_{\max}+1} = 0$):

$$L_1 = \begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & \dots & 0 \\ 0 & 0 & \sqrt{2} & 0 & \dots & 0 \\ 0 & 0 & 0 & \sqrt{3} & \dots & 0 \\ 0 & 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \sqrt{m_{\max}} \\ 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}. \tag{7.96}$$

This means that the Kraus series can be cut down to a manageable set of only $M = m_{\max}+1$ Kraus operators that need to be realized as quantum circuits. Here, we will assume $m_{\max} = 2^N - 1$, where N is the desired number of system qubits.

The Kraus operators $K_m(t)$ could be realized as circuits of the form shown in Figure 7.7, but this may be problematic because the repeated application of the unitary dilation U_{A_1}, where $A_1 = L_1/\|L_1\|$ requires many quantum gates, which may induce noise. Fortunately, this gate cost can be reduced significantly by fusing the operations of the effective Hamiltonian evolution (7.72) and the evaluation of powers of the truncated L_1 operator in (7.96). Specifically, we write

$$K_m(t) = W(t) D_m(t) \text{SUB}_m \tag{7.97}$$

where SUB_m is the quantum circuit that implements subtraction by m modulo 2^N on N qubits, and $D_m(t)$ is the diagonal operator

$$D_m(t) = \sum_{n=0}^{2^N-(m+1)} e^{-n\gamma_1 t/2} \sqrt{\frac{\gamma_1^m (1-e^{-\gamma_1 t})^m (n+m)!}{n! m!}} \left|e_n\right\rangle \left\langle e_n\right|, \tag{7.98}$$

with nonzero entries corresponding to the n-th super-diagonal of the matrix (7.96) raised to the power n. The operator $W(t) = e^{-it\omega(\frac{1}{2}+\hat{N})}$ is the time evolution operator for the oscillator Hamiltonian truncated to 2^N modes. To normalize the Kraus operators, we define $\Lambda_m(t) = D_m(t)/a_m(t)$ where $a_m(t) = \|D_m(t)\|$ is the normalization coefficient. Then, the normalized Kraus operators can be realized as the circuit shown in Figure 7.13.

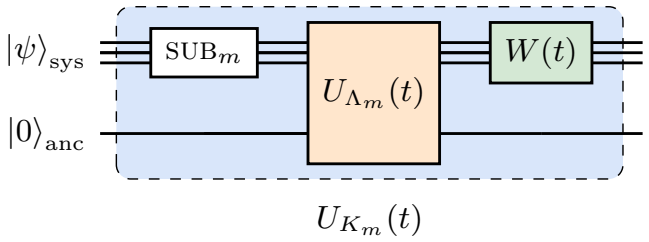

Figure 7.13: Kraus circuits for the damped quantum harmonic oscillator. These circuits fuse together the computation of L_1^m with the effective Hamiltonian evolution circuit $U_{\mathcal{H}}(t)$ in Figure 7.8.

As shown previously, the gate $U_{\Lambda_m}(t)$ in Figure 7.13 requires $O(d\log(d))$ quantum gates and $O(d\log(d))$ classical operations to compute the time-dependent gate parameters. However, $W(t)$ requires only $\log_2(d)$ parameterized gates, since it can be decomposed into only single-qubit phase gates of the form

$$W(t) = e^{i\theta_0(t)} P(\theta_{2^0}(t)) \otimes P(\theta_{2^1}(t)) \otimes ... \otimes P(\theta_{2^N}(t)) \tag{7.99}$$

where $\theta_{2^n}(t) = 2^n\omega t$ and $\theta_0(t)$ is a global phase. Finally, the SUB_m gate can be implemented using at most $\log(m)+1$ SUB_1 gates applied to subsets of the N system qubits, each of which requires $O(N)$ gates with $N-1$ shared ancilla qubits [218]. Thus, the SUB_m gate can be decomposed into $O(\log(m)\log(d))$ elementary gates.

In total, the Kraus circuit in Figure 7.13 requires $O(d\log(d))$ quantum gates and classical preprocessing operations, which is still more efficient than $O(d^2)$ for

more general systems satisfying conditions (i)-(iv). In Table 7.2 we summarize the time complexity (quantum and classical) and number of Kraus circuits required for both this system, the continuous-time Pauli channel, and general systems satisfying conditions (i)-(iv).

Table 7.2: Table of Kraus circuit resource requirements (quantum gate complexity, classical pre-processing complexity, and number of qubits) with respect to system dimension d. Results are shown for the multi-qubit continuous-time Pauli channel, the damped quantum harmonic oscillator, and for general systems satisfying the concrete conditions (i)-(iv) introduced in Section 8.1. For each system, bounds on the number of Kraus circuits are also shown.

	MQCT Pauli Channel	Damped QHO	Systems Satisfying (i)-(iv)
Resources (per Kraus circuit):			
Quantum Gate Complexity	$O(\log(d))$	$O(d\log(d))$	$O(d^2)$
Classical Pre-processing Complexity	$O(1)$	$O(d\log(d))$	$O(d\log(d))$
Number of Qubits	$\log_2(d)$	$2\log_2(d)$	$\log_2(d)+M+1$
Number of Kraus circuits	2^{N_L}	$m_{\max}+1 \leq d$	$\sum_{m=0}^{M}(N_L)^m < \infty$

7.4.3 More General Systems

We have shown that both damped harmonic oscillators and Pauli channels satisfy the concrete conditions (i)-(iv), and thus our method can be directly applied to these systems, resulting in the case (I) and case (II) Kraus series representations respectively. However, we briefly remark that there are more general types of systems for which the two examples studied here are special cases. For example, we can

generalize the Pauli channel system by considering an arbitrary quantum system with Hamiltonian H that exhibits some non-trivial unitary symmetries $U_1, U_2, ..U_{N_L}$ each satisfying $U_i H U_i^\dagger = H$. To model decoherence that drives this system toward a symmetric steady state, one can employ the Lindblad operators $L_i = U_i$ with appropriate choice of coupling γ_i. It immediately follows that $[H, U_i] = 0$ and (using the fact that $U_i^\dagger U_i = I$) the system satisfies conditions (i)-(iv) with $\nu = \lambda = 0$.

We also remark that the damped quantum harmonic oscillator can be readily generalized to model decay in quadratic bosonic systems. These systems have a Hamiltonian H that can be written in the form

$$H = \sum_{i,j} M_{ij} \hat{a}_i^\dagger \hat{a}_j + \frac{1}{2}(\Delta_{jk} \hat{a}_j^\dagger \hat{a}_k^\dagger + \Delta_{jk}^* \hat{a}_k \hat{a}_j) \tag{7.100}$$

where M is a Hermitian matrix, Δ is a symmetric matrix, and the $\hat{a}_i$ are bosonic annihilation operators. Under certain conditions, (7.100) can be diagonalized via a symplectic Bogoliubov transformation [18, 219] to obtain an equivalent Hamiltonian of the form (7.2), where the transformed bosonic annihilation operators $\hat{b}_n$ satisfy the same canonical commutation relations as the $\hat{a}_i$ operators. One can model particle loss in a selected subset of modes by applying the corresponding subset of commuting Lindblad operators $L_i = \hat{b}_i$. For these diagonalizable systems, it can be shown that conditions (i)-(iv) are satisfied and that our method can be applied to simulate them efficiently on NISQ hardware. This follows as a straightforward generalization of the damped harmonic oscillator (which models a single bosonic mode) to multiple bosonic modes.

7.5 Conclusion

In the beginning of this paper we introduced the superoperators $\mathcal{H}$ and $\mathcal{L}$ which can be used to describe the dynamics of open quantum systems under the Born-Markov approximation. For certain systems satisfying $[\mathcal{H}, \mathcal{L}] = \alpha\mathcal{L} + c$ for $\alpha, c \geq 0$, we showed that a time-perturbative Kraus series representation of the system could be obtained, where each Kraus operator in the series can be realized as a quantum circuit with bounded time-independent gate complexity. After analyzing the convergence of this Kraus series, we showed that the series can be truncated to a constant number of terms, which can be efficiently evaluated on NISQ quantum devices. To demonstrate how our framework can be applied, we analyzed two canonical open quantum systems, the multi-qubit Pauli channel and the damped quantum harmonic oscillator, and showed that they satisfy the concrete conditions (i)-(iv), and thus can be simulated efficiently using a series of Kraus circuits with t-independent depth.

We anticipate that this framework may be useful for not only simulating open quantum systems (such as the two example systems discussed here), but also as a means of quickly preparing states that lie along the trajectories of these systems. The fact that our method distributes the complexity of state preparation across several circuits of bounded depth may yield a critical advantage when run on NISQ devices, since it minimizes state preparation error and incurs a low per-circuit gate complexity overhead compared to Trotter-based methods.

7.5.0.1 Future work. Although the framework we have developed in this paper is promising for simulating the class of systems that satisfy conditions 1-3, it

does not admit a straightforward generalization to arbitrary Markovian open quantum systems governed by the Lindblad equation. Our method exploits the fact that the commutation relations between $\mathcal{H}$ and $\mathcal{L}$ are such that the Zassenhaus expansion (7.25) of the system dynamics collapses to a tractable form, allowing for the environment action superoperator $\mathcal{A}_{\text{env}}(t)$ to admit a series expansion. The discovery of different or more general sets of conditions that facilitate closed-form time evolution formulas of open quantum systems may be a promising direction of future research. In addition, the discovery of new applications or NISQ hardware-specific implementations of systems satisfying conditions 1-3 may also prove a fruitful area for further investigation. This framework is presently being applied in ongoing computations with NISQ quantum hardware.

CHAPTER EIGHT

Simulation of Open Quantum Systems on NISQ Quantum Devices

This chapter published as: C. Burdine, N. Bauer, G. Siopsis, E. P. Blair, "Efficient Simulation of Open Quantum Systems on NISQ Trapped-Ion Hardware." *Advanced Quantum Technologies*, Feb. 2025, 2400606., doi: 10.1002/qute.202400606

Simulating open quantum systems, which interact with external environments, presents significant challenges on noisy intermediate-scale quantum (NISQ) devices due to limited qubit resources and noise. In this paper, we propose an efficient framework for simulating open quantum systems on NISQ hardware by leveraging a time-perturbative Kraus operator representation of the system's dynamics. Our approach avoids the computationally expensive Trotterization method and exploits the Lindblad master equation to represent time evolution in a compact form, particularly for systems satisfying specific commutation relations. We demonstrate the efficiency of our method by simulating quantum channels, such as the continuous-time Pauli channel and damped harmonic oscillators, on NISQ trapped-ion hardware, including IonQ Harmony and Quantinuum H1-1. Additionally, we introduce hardware-agnostic error mitigation techniques, including Pauli channel fitting and quantum depolarizing channel inversion, to enhance the fidelity of quantum simulations. Our results show strong agreement between the simulations on real quantum hardware and exact solutions, highlighting the potential of Kraus-based methods for scalable and accurate simulation of open quantum systems on NISQ devices. This framework opens pathways for simulating more complex systems under realistic conditions in the near term.

8.1 Introduction

The efficient simulation of open quantum systems is crucial for the study of quantum algorithms, materials, communications, and many far-reaching aspects of quantum technology. This area is still developing, especially given the challenges associated with noisy intermediate-scale quantum (NISQ) devices. Simulating open quantum systems—which involve interactions with an external environment—presents additional complexity compared to closed quantum systems, making this an important and active area of research.

Open quantum systems with Markovian (memoryless) noise are typically described by the Lindblad master equation, which models the system's evolution in the presence of interaction with the environment. Work on simulating Lindblad-type dynamics using quantum simulators and NISQ hardware has involved studies of quantum noise channels, such as depolarizing, amplitude damping, and phase damping channels [220]. These studies are foundational because noise and decoherence naturally occur in quantum hardware, making it essential to simulate and understand such effects.

A popular approach to simulating open quantum systems on quantum devices uses Trotterization (as in closed systems) to discretize the time evolution of the system. While successful to some extent, this method becomes impractical on NISQ devices due to the large number of gates required [159, 221]. Other methods that have been explored are based on variational quantum algorithms which can be designed to be hardware-efficient, though these methods can still struggle with noise and require classical optimization [188, 222, 223]. An alternative approach is based on quantum trajectory (Monte Carlo) methods [224], which simulate the evolution of

a quantum system under stochastic noise or decoherence. These methods have been demonstrated on small-scale quantum devices, though their practical application is still limited by device noise [225]. Other methods include using the NISQ device noise for probabilistic error cancellation [183], matrix product state (MPS) numerical methods [181], and quantum imaginary time evolution (QITE) [177].

Certain Trotterless schemes have been devised that should be in principle better suited to NISQ hardware [226], but their application is limited to toy models, lacking generality. Simulation schemes for analog quantum simulators, such as trapped ions and Rydberg atoms, have been investigated under various environmental conditions [227]. These platforms allow for the exploration of dissipative processes and quantum systems interacting with external fields, though their scalability and programmability are limited compared to digital quantum devices.

Various challenges exist. Simulating open quantum systems often requires complex quantum circuits, which are sensitive to noise on NISQ devices, limiting the depth and number of qubits that can be effectively used. Developing error mitigation techniques is a central focus for making such simulations practical on NISQ devices. Finding efficient representations for dissipation processes (e.g., using Kraus operators or other reduced models) has been a significant research focus. While various studies have simulated specific types of open quantum systems on NISQ hardware, the methods often rely on approximations, and large-scale, high-fidelity simulations remain a challenge.

In this work, we introduce more efficient methods for simulating open quantum systems, including novel circuit designs and error mitigation strategies. These

approaches aim to push the boundaries of what is possible on NISQ devices and overcome some of the limitations of earlier methods. We focus on Kraus representation-based methods to simulate the dynamics of these systems. These methods exploit a representation of the system's time evolution in terms of Kraus operators, which model the interaction between the system and its environment. When specific commutation relations between the system's Hamiltonian and Lindblad operators are satisfied, this representation can be computed efficiently without Trotterization. The Kraus operators offer a more efficient and NISQ-friendly way of simulating these systems. We apply techniques to map the Kraus operators and time evolution operators to quantum circuits for implementation on actual NISQ hardware (IonQ Harmony and Quantinuum H1-1 devices).

Our results confirm that the Kraus representation-based method, combined with error mitigation, can effectively simulate open quantum systems on NISQ devices, with strong agreement between experimental results and theoretical predictions. They represent a significant step forward in the simulation of open quantum systems on NISQ devices and a critical step in bridging theoretical methods and real-world quantum computing applications.

This paper is organized as follows: In Section 8.2, we give a review of the motivating background theory, including the Lindblad equation, Trotterization, and Kraus representations. We also discuss how systems satisfying certain commutation relations admit Kraus representations that can be simulated efficiently, and we discuss some important special cases of these systems. In Section 8.3, we present the results of applying our framework to simulate selected open quantum systems on real quantum hardware. We simulate a two-qubit system undergoing decoherence,

modeled as a Pauli channel. The results show good agreement between the Kraus series-based simulation and the exact Lindblad equation solution. We also simulate a 2D quantum harmonic oscillator under damping, which is modeled using angular momentum operators as Lindblad operators. The results are compared with an exact solution, and strong agreement is observed, even in the presence of noise. For larger systems, the effect of noise becomes more significant. We present hardware-agnostic error mitigation techniques, such as Pauli channel fitting and quantum depolarizing channel inversion, that can be applied to improve simulation fidelity. Finally, in Section 8.4, we summarize our results and give directions for future research. Appendix E.1 contains additional details for mapping Kraus operators to quantum circuits for the relevant systems. Appendix E.2 contains additional details for applying Pauli noise models to density matrix simulations.

8.2 Background

8.2.1 Lindblad Equation

In this paper, we will consider open quantum systems under the Born-Markov approximation– that is, systems weakly coupled to a large environment in which correlations between the system and environment are short-lived. Under these assumptions, we are free to impose that the the dynamics of a system are Markovian (i.e. the environment is effectively "memoryless") and can be modeled by a completely-positive trace preserving (CPTP) map. In non-relativistic systems governed by the Schrodinger equation, the CPTP time evolution of a system can always be written in

the Lindblad form

$$\begin{aligned}\dot{\rho}(t) = &-\frac{i}{\hbar}[H, \rho(t)] \\ &+ \sum_{n=1}^{N} \gamma_n \left(L_n \rho(t) L_n^\dagger - \frac{1}{2}\{L_n^\dagger L_n, \rho(t)\} \right),\end{aligned} \tag{8.1}$$

where $\rho(t)$ is the system density matrix, H is the system Hamiltonian, and the L_n are Lindblad operators with positive damping coefficients γ_n. The Lindblad operators are often imposed phenomenologically on a system to model environment-induced decoherence [166].

8.2.2 Trotter-Based Methods

Solving Eq. (8.1) for arbitrary quantum systems as a closed-form function of time is known to be a difficult problem. As a result, one must commonly resort to numerical integration methods on quantum or classical computers. On quantum computers, the most popular of these methods are those that employ Trotter product formulas [190]. This is usually achieved by re-writing the right-hand side of Eq. (8.1) in the equivalent superoperator form, namely

$$\dot{\vec{\rho}}(t) = \mathcal{D}\vec{\rho}(t), \tag{8.2}$$

where $\mathcal{D}$ is a superoperator matrix that acts linearly on the "vectorized" density matrix $\vec{\rho} = \sum_{i,j} \rho_{ij}(|i\rangle \otimes |j\rangle)$. Next, the superoperator $\mathcal{D}$ is decomposed into a sum of non-commuting operators, e.g.

$$\mathcal{D} = \mathcal{D}_1 + \mathcal{D}_2 \tag{8.3}$$

such that the evolution of each component $\mathcal{D}_i$ for a small time step δt can be realized as an efficient sequence of quantum gates $U_{\mathcal{D}_i} \simeq \exp(\delta t \mathcal{D}_i)$. The key idea of a Trotter product formula is that one can repeatedly apply an alternating sequence

of the gates $U_{\mathcal{D}_i}$ over many discrete δt time steps such that the approximation error vanishes in the limit as $\delta t \to 0$. For example, applying a second order Trotter product formula [191] to Eq. (8.3) over an n-step time evolution period t ($\delta t = t/n$), we obtain the approximation:

$$\vec{\rho}(t) = \left[e^{(\delta t/2)\mathcal{D}_1} e^{\delta t \mathcal{D}_2} e^{(\delta t/2)\mathcal{D}_1}\right]^n \vec{\rho}(0) + O(t(\delta t)^2) \tag{8.4}$$

This approach to simulating open quantum system is appealing in that it only requires the repeated application of the quantum gate unitaries $U_{\mathcal{D}_i}$. However, it can be quite costly for long evolution periods t or systems where the unitaries $U_{\mathcal{D}_i}$ contain many gates [159]. This is especially known to be the case for dense bosonic systems, where computing the exponential of bosonic creation and annihilation operator terms in $\mathcal{D}$ is known to be quite expensive [228]. For this reason, Trotter-based techniques have remained difficult to apply on NISQ devices, except in the cases where the system dynamics can be straightforwardly mapped to device-native operations that can be performed with minimal noise.

8.2.3 Kraus Representation-Based Methods

For some systems, Trotterization can produce quantum circuits that require a large number of gates and qubits, which is undesirable for NISQ applications. However, for some special classes of systems, a closed-form solution to the Lindblad equation can be found that significantly reduces the complexity of Trotterization or even avoids Trotterization altogether. This closed form solution can always be written in the form of a time-dependent Kraus operator representation of the solution to Eq.

(8.1), given by:

$$\rho(t) = \sum_i K_i(t)\rho(0)K_i(t)^\dagger \tag{8.5}$$

where the $K_i(t)$ are time-dependent Kraus operators that satisfy the unital condition $\sum_i K_i(t)K_i(t)^\dagger = I$.

As a direct consequence of Choi's theorem [196], one can always solve for a minimal set of d^2 Kraus operators K_i corresponding to the time evolution of a $d \times d$ density matrix for a period t; however, finding this minimal set of Kraus operators for arbitrary systems is classically hard, requiring the diagonalization of the system's $d^2 \times d^2$ Choi matrix for each time step t [193]. Fortunately, it can be shown that when a system's Lindblad operators and Hamiltonian satisfy certain commutation relations, a series of non-minimal Kraus operators corresponding to the time-perturbative treatment of the system's time evolution can be solved for exactly, without requiring the use of diagonalization. Specifically, if a Hamiltonian and a set of Lindblad operators satisfy the commutation relations

(i) $[H, L_n^\dagger L_n] = 0$ (for all L_n)

(ii) $[L_n^\dagger L_n, L_{n'}^\dagger L_{n'}] = 0$ (for all $L_n, L_{n'}$)

(iii) $[H, L_n] = \nu L_n$
(for some $\nu \in \mathbb{C}$ with $\mathrm{Im}(\nu) \geq 0$, for all L_n)

(iv) $\sum_{n'} \gamma_{n'}[L_{n'}^\dagger L_{n'}, L_n] = \lambda L_n$
(for some $\lambda \in \mathbb{C}$ with $\mathrm{Re}(\lambda) \leq 0$, for all L_n)

then Eq. (8.1) can be solved in closed form by a time-perturbative Kraus series of the general form

$$\rho(t) = \sum_{m=0}^{\infty} \sum_{\vec{k} \in \{1,2,\ldots,N\}^m} K_{m,\vec{k}}(t)\rho(0)K_{m,\vec{k}}(t)^\dagger \tag{8.6}$$

where m is the order of each term and $\vec{k}$ is an index vector corresponding to each product sequence of m Lindblad operators. In the general case, each Kraus operator term in the time-perturbative series takes the form

$$K_{m,\vec{k}}(t) = T(t)\sqrt{\frac{f(t)^m}{m!}}\prod_{i=1}^{m}\left(\sqrt{\gamma}_{\vec{k}_i}L_{\vec{k}_i}\right). \tag{8.7}$$

Above, $f(t)$ is a positive function of time and $T(t)$ is the effective Hamiltonian evolution operator given by

$$T(t) = \exp\left(-\frac{it}{\hbar}H_{\text{eff}}\right) \tag{8.8}$$

where the effective Hamiltonian H_{eff} is

$$H_{\text{eff}} = H - \frac{i\hbar}{2}\sum_{n=1}^{N}\gamma_n L_n^\dagger L_n. \tag{8.9}$$

Because conditions (i) and (ii) are satisfied, the operator H_{eff} is diagonalizable in the same basis as the original Hamiltonian H with eigenvalues of the form

$$E_{\text{eff}} = E - \frac{i\hbar}{2}\sum_{n=1}^{N}\gamma_n\xi_n \tag{8.10}$$

where E is a real eigenvalue of H and each ξ_n is a non-negative real eigenvalue of $L_n^\dagger L_n$. Since the imaginary component of each eigenvalue E_{eff} is non-positive, the effective Hamiltonian evolution operator $T(t)$ is a contraction, meaning $\left\|T(t)\rho T(t)^\dagger\right\| \leq \|\rho\|$ for any density matrix ρ and time $t \geq 0$. The time-dependence of the Kraus operator terms in Eq. (8.7) is described by the positive function $f(t)$, which takes different forms, depending on the values of ν and λ as defined in conditions (iii) and (iv). In the most general case, $f(x)$ takes the form

$$f(t) = \begin{cases} t & \alpha = 0 \\ (1 - e^{-\alpha t})/\alpha & \alpha > 0 \end{cases} \tag{8.11}$$

where $\alpha = 2\,\mathrm{Im}(\nu)/\hbar - \mathrm{Re}(\lambda)$ is a positive constant. When $\alpha = 0$, we observe that the worst case time-dependence of the Kraus operators is such that $K_{m,\vec{k}}(t) \sim \sqrt{t^m/m!}$. This case corresponds to systems with unitary symmetries (e.g. Pauli channels), where the unitary symmetry operators are used as Lindblad operators with different choice of damping parameters γ_n. The $\alpha > 0$ case corresponds to systems where H and the Lindblad terms $L_n^\dagger L_n$ exhibit non-trivial commutation relations (e.g. bosonic systems with finite particle number). For these systems, the time dependence can be reduced to $K_{m,\vec{k}}(t) \sim \sqrt{1/m!}$, which means that the Kraus series converges much faster and exhibits much lower truncation error than the general case. For additional details regarding the derivation of the general form of the time-perturbative Kraus operators in Eq. (8.7), we refer the reader to reference [229].

8.2.3.1 Mapping to quantum circuits. Upon inspection of Eq. (8.6) and Eq. (8.7), we observe that if the effective time evolution operator $T(t)$ and the Lindblad operators L_n can be efficiently realized as quantum circuits acting on a pure state $|\psi\rangle$ (or a purified superposition $\sum_i \sqrt{p_i}\,|\psi_i\rangle$), then one can straightforwardly simulate the time-evolution of an arbitrary density matrix $\rho = \sum_i p_i\,|\psi_i\rangle\langle\psi_i|$ on a quantum computer. This is achieved by first applying the sequence of Lindblad operators $\prod_i L_{\vec{k}_i}$ followed by $T(t)$, and then multiplying by the scalar term

$$a_{m,\vec{k}} = \sqrt{\frac{f(t)^m}{m!}} \prod_{i=1}^{m} \sqrt{\gamma_{\vec{k}_i}}. \tag{8.12}$$

This must be repeated for each Kraus term $K_{m,\vec{k}}(t)$ up to a desired cutoff order $m = M$. Applying each of these operators through quantum gates, however, is a non-trivial task and requires additional discussion.

First, we consider how the Lindblad operators L_n can be expanded into unitary operations. Since it may generally be the case that $\|L_n\| \neq 1$, the Lindblad operators must first be normalized using the transformation

$$\begin{aligned} L_n &\mapsto L_n/b_n \\ \gamma_n &\mapsto b_n^2 \gamma_n \end{aligned} \tag{8.13}$$

where $b_n = \|L_n\|$ is the normalization constant. This transformation has no effect on the dynamics of Eq. (8.1), and ensures that each L_n can be encoded in a unitary of norm 1. Once the L_n operators are normalized, they must be expanded to unitaries that can be implemented with quantum gates. This is most commonly achieved through a unitary block-encoding scheme in which the transformed Lindblad operators are embedded in a block unitary U_{L_n} that requires at least one clean ancilla qubit per application in a quantum circuit [230]. One such block form is given by the unitary Sz.-Nagy dilation [202] of L_n:

$$U_{L_n} = \begin{pmatrix} L_n & \sqrt{I - L_n L_n^\dagger} \\ \sqrt{I - L_n^\dagger L_n} & -L_n^\dagger \end{pmatrix}. \tag{8.14}$$

This block encoding is such that $U_{L_n}(|0\rangle \otimes |\psi\rangle) = (L_n |\psi\rangle) \otimes |0\rangle + |\phi\rangle \otimes |1\rangle$, where $|\phi\rangle = \sqrt{I - L_n^\dagger L_n}\,|\psi\rangle$ is a state that must be discarded when the ancilla qubit measures $|1\rangle$. Each unitary U_{L_n} can be decomposed efficiently as sequences of quantum gates using algorithms such as the linear combination of unitaries (LCU) method [231, 232] or the quantum Shannon decomposition (QSD) [203]. Due to the constraints of NISQ quantum hardware, implementing arbitrary unitary operators without accumulating noise is a significant challenge; thus the implementation of complex Lindblad operators (for example the bosonic annihilation operator $\hat{a}$) requires

careful circuit design and hardware-specific optimization [228]. We give additional discussion regarding circuit design in Appendix E.1.

Next, we briefly discuss how the operator $T(t)$ can be implemented. $T(t)$ is a contraction, meaning it can be expanded to a unitary operator without normalization. This expansion can be achieved through a unitary block-encoding scheme, however we recall that the effective Hamiltonian H_{eff} is diagonalizable in the same basis as H. As a result, if a diagonal form of H is known, then a diagonal block-encoding of the the operator $T(t)$ can be implemented efficiently in this diagonalized basis using the methods discussed in Appendix E.1.

8.2.3.2 Measurement and readout of circuits. For each operator Eq. (8.7) in the Kraus series expansion, the unitaries U_{L_n} (implementing the Lindblad operators) and $T(t)$ (implementing the effective Hamiltonian evolution) can be combined to produce a circuit $U_{K_{m,\vec{k}}}(t)$ corresponding to each Kraus operator, as shown in Figure 8.1. When each circuit is run on a quantum device, both the system qubits and all ancillas are measured. When the ancillas all read $|0\rangle$, the the resulting system qubits are measured in a desired basis and the estimated outcome probabilities for each circuit are added on a classical computer with scalar weights $|a_{m,\vec{k}}|^2$.

Measurements in multiple bases can be combined to estimate the expectation value of the observable for the state $\rho(t)$. The state density matrix $\rho(t)$ itself can also be estimated by performing quantum state tomography in a complete basis [233]. On NISQ devices, the Pauli basis is often chosen because transformations into this basis require only single-qubit Pauli gates. Although the Kraus operator method requires many circuits to be evaluated per measurement basis (in the worst case, $M!$ circuits),

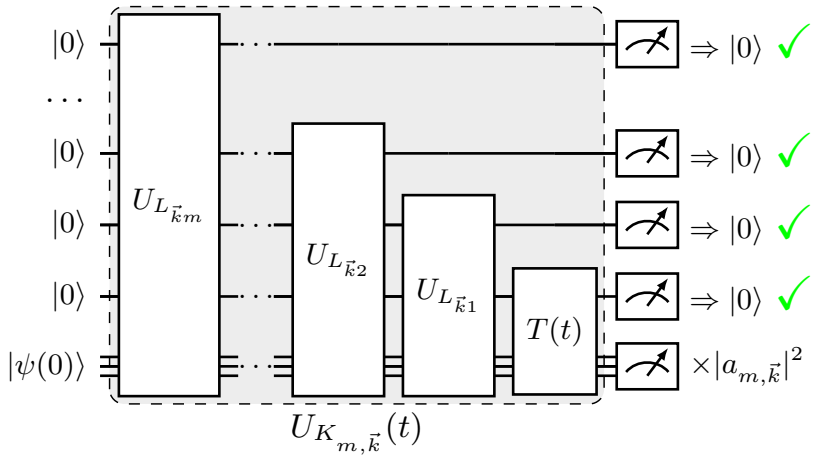

Figure 8.1: Diagram of the quantum circuit $U_{K_{m,\vec{k}}}(t)$ corresponding to a single Kraus operator in the time-perturbative Kraus series Eq. (8.6). This circuit applies a set of unitary dilations of the Lindblad operators ($U_{L_{\vec{k}}}$) indexed by the vector $\vec{k}$. Then, the effective Hamiltonian evolution $T(t)$ is applied. When all of the ancillas are measured to read $|0\rangle$, the resulting measurement probabilities on the system qubits are re-scaled by the factor $|a_{m,\vec{k}}|^2$ and added on a classical computer.

we note that the depth of each order m circuit is at most $O(m)$. Furthermore, the Kraus series terms that contribute the most probability mass to the total system trajectory tend to correspond to the shortest circuits. As a result, this approach naturally mitigates the detrimental effect of noise on NISQ devices by distributing the computational cost of the simulation over many short circuits.

8.2.4 Special Cases

Although the time-perturbative Kraus series method is limited to systems that satisfy conditions (i)-(iv) and can require many circuits in the most general case, we note that there are some important classes of systems where our method not only applies, but can be made very efficient. In some cases, the time-perturbative Kraus series can even be reduced to a finite number of terms, allowing for an exact simulation

distributed over only a finite number of circuits. In the following subsections, we provide further analysis of these noteworthy special cases:

8.2.4.1 Systems with unitary symmetry. First, we examine the special case of systems that exhibit unitary symmetries. These systems have a Hamiltonian H and unitary operators $U_1, U_2, ..., U_N$ that satisfy $U_n H U_n^\dagger = H$. The unitaries U_n can be used as Lindblad operators $L_n = U_n$ with appropriate choice of damping parameters γ_n to phenomenologically model decoherence that drives the system toward a steady state that is invariant under these symmetries. For these systems it can be shown that conditions (i)-(iv) are satisfied with constants $\nu = \lambda = 0$, resulting in $f(t) = t$. The effective Hamiltonian evolution operator $T(t)$ takes the form

$$T(t) = e^{-\Gamma t/2} U_H(t) \tag{8.15}$$

where $\Gamma = \sum_{n=1}^{N} \gamma_n$ and $U_H(t) = \exp(-itH/\hbar)$ is the time evolution operator under the regular system Hamiltonian H. The scalar prefactor $e^{-\Gamma t/2}$ can be computed classically, which means that one only needs to apply the sequence of unitaries $U_H(t) U_{\vec{k}_1} U_{\vec{k}_2} ... U_{\vec{k}_N} |\psi\rangle$ to simulate each Kraus term on a quantum computer.

8.2.4.2 Bosonic systems. Another special case of systems we consider here are bosonic systems. The Hamiltonian of a free bosonic system with a single mode can be modeled as a single-particle harmonic oscillator with Hamiltonian

$$H = \hbar\omega \hat{a}^\dagger \hat{a} \tag{8.16}$$

where $\hat{a}$ is the mode annihilation operator and ω is the mode frequency. To model particle loss in the mode, we apply the Lindblad operator $L_1 = \hat{a}$. It can be shown that this system satisfies conditions (i)-(iv) with constants $\nu = \hbar\omega$ and $\lambda = -\gamma_1$,

which means that $\alpha = \gamma_1 \geq 0$. Thus, the time-dependence of the the derived time-perturbative Kraus series is of the form $f(t) = (1 - e^{-\alpha t})/\alpha$, meaning the Kraus series converges rapidly as claimed earlier. We can generalize the single mode case to a quadratic bosonic Hamiltonian with n coupled modes, which can be written as

$$H = \sum_{i,j} M_{ij} \hat{a}_i^\dagger \hat{a}_j \tag{8.17}$$

where M is an $n \times n$ Hermitian matrix and $\hat{a}_i$ is the annihilation operator for mode i. In a manner similar to the Hamiltonian, Lindblad operators can be introduced as a matrix quadratic form

$$L_n = \sum_{i,j} (V_n)_{ij} \hat{a}_i^\dagger \hat{a}_j \tag{8.18}$$

where V_n is an arbitrary matrix. As a direct consequence of the Jordan-Schwinger map, the commutator algebra of H and the Lindblad operators L_n is isomorphic to that of the matrices M and V_n. As a result, one only needs to consider the commutation relations of these matrices when verifying that conditions (i)-(iv) above apply. An important case to consider is when the matrices V_n are diagonal or correspond to unitary symmetries of M (i.e. $V_n M V_n^\dagger = M$). These kinds of systems are relevant in the theory of quantum angular momentum [234], coupled optical cavities [215], and phonons in solids [235].

We would like to emphasize that the damped harmonic oscillator systems that can be simulated using the Trotterless method can be generalized to model the decay of quadratic bosonic systems under a Bogoliubov transformation [18]. This would allow for a Hamiltonian of the form

$$H = \sum_{i,j} M_{ij} \hat{a}_i^\dagger \hat{a}_j + \frac{1}{2} \left(\Delta_{jk} \hat{a}_j^\dagger \hat{a}_k^\dagger + \Delta_{jk}^* \hat{a}_j \hat{a}_k \right) \tag{8.19}$$

where M is a Hermitian matrix, Δ is a symmetric matrix, and the Lindblad operators are in terms of the transformed bosonic annihilation operator $L_i = \hat{a}_i$. This Hamiltonian in Eq. (8.19) can realize the four wave-mixing/spontaneous parametric down conversion Hamiltonian [236] in a lossy cavity. This is particularly relevant in quantum imaging when trying to obtain strong squeezing in leaky cavities [237]. There has also been interest in using Lindbladians with dissipative couplings to engineer non-abelian states in photonic lattices [238]. The case where $\Delta_{jk} = 0$ can be related to the dissipative evolution of various spin and fermionic Hamiltonians in the single-excitation subspace [239], as long as the Lindblad operators are particle/excitation number conserving.

8.2.4.3 Lindblad operators with finite group and semigroup structure. For systems with Lindblad operators that form a finite group or semigroup, this infinite series can be simplified down to a finite number of Kraus operators, yielding an exact solution. Of these types of systems, the most important case to consider is when the action of the Lindblad operators L_n on some initial density matrix ρ form finite abelian groups or finite abelian nilpotent semigroups, meaning they satisfy

$$L_{n'} L_n \rho L_n^\dagger L_{n'}^\dagger = L_n L_{n'} \rho L_{n'}^\dagger L_n^\dagger \tag{8.20}$$

(i.e. the action of the Lindblad operators on ρ is commutative) and

$$(L_n)^{\ell_n} \rho (L_n^\dagger)^{\ell_n} = \theta_n \rho \tag{8.21}$$

for real constants $\theta_n \geq 0$ and integers $\ell_n > 0$. If $\theta_n = 0$, the group generated by the action of L_n on ρ is a finite nilpotent semigroup; otherwise it forms a finite abelian group (up to multiplication by some constant). In either case, it can be shown that the resulting time-perturbative Kraus series can be reduced to the finite sum of Kraus

operators

$$\rho(t) = \sum_{m_1=0}^{\ell_1-1} \sum_{m_2=0}^{\ell_2-1} ... \sum_{m_N=0}^{\ell_N-1} K_{\vec{m}}(t)\rho(0)K_{\vec{m}}(t)^\dagger \tag{8.22}$$

where

$$K_{\vec{m}}(t) = T(t) \prod_{n=1}^{N} \sqrt{F^{\theta_n}_{\ell_n,m_n}(\gamma_n t)}(L_n)^{m_n} \tag{8.23}$$

and $F^\theta_{\ell,m}$ is the generalized hyperbolic function

$$F^\theta_{\ell,m}(x) = \frac{1}{\ell}\theta^{-m/\ell} \sum_{k=0}^{\ell-1} \omega_\ell^{-mk} \exp\left(\omega_\ell^k \theta^{\frac{1}{\ell}} x\right) \tag{8.24}$$

with $\omega_\ell = e^{i2\pi/\ell}$. In the nilpotent case ($\theta = 0$), we define $F^0_{\ell,m}(x) = f(x)^m/(m!)$.

Although the finite sum of Kraus operators in Eq. (8.22) contains many individual terms, we observe that this series can be factored as the product of polynomial functions of the Lindblad operators. Specifically, in the finite abelian case with $\theta_n \neq 0$, we can re-scale the Lindblad operators via the transformation in Eq. (8.13) with $b_n = \theta_n^{1/(2\ell_n)}$. This allows us to write the time evolution of the system as $\rho(t) = S(t)\rho S(t)^\dagger$ where $S(t)$ is given by

$$S(t) = e^{\Gamma' t/2} T(t) \prod_{n=1}^{N} \left(\sum_{m=0}^{\ell_n-1} \sqrt{e^{-\gamma'_n t} F^1_{\ell_n,m}(\gamma'_n t)}(L'_n)^m \right) \tag{8.25}$$

where $L'_n = L_n/\theta^{1/(2\ell_n)}$, $\gamma'_n = \theta^{1/\ell_n}\gamma_n$, and $\Gamma' = \sum_{n=1}^N \gamma'_n$. If one can efficiently apply a controlled form of the gates $U_{L'_n}$ that implement each re-scaled Lindblad operator L'_n, then each factor in Eq. (8.25) can be applied using circuits of the form shown in Figure 8.2, provided that the distribution

$$|\phi_{F_n}(t)\rangle = \sum_{m=0}^{\ell_n-1} \sqrt{e^{-\gamma'_n t} F^1_{\ell_n,m}(\gamma'_n t)}\, |m\rangle \tag{8.26}$$

can be efficiently prepared for each time evolution period $t > 0$ on a set of $\log_2(\ell_n)$ ancilla qubits. It can be shown that a circuit preparing the distribution $|\phi_{F_n}(t)\rangle$ can

be compiled and executed efficiently, requiring at most $O(\ell_n \log(\ell_n))$ quantum gates and classical preprocessing steps [240].

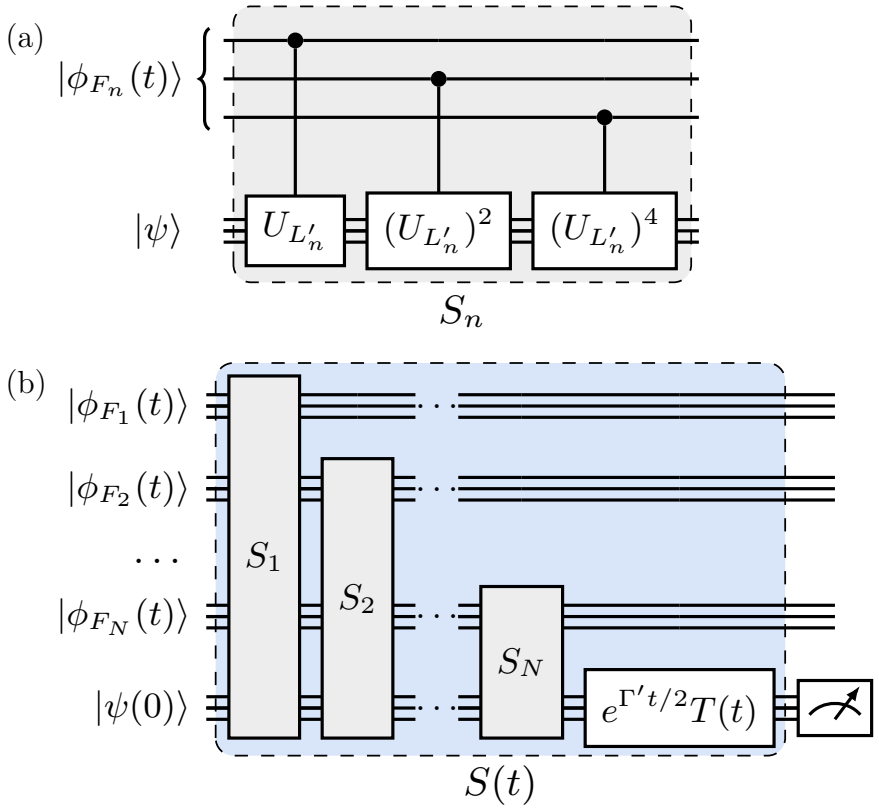

Figure 8.2: Quantum circuit diagrams for the circuits S_n (a) and $S(t)$ (b), which can be used to efficiently simulate systems where the action of the Lindblad operators that form a finite abelian group. Simulating these systems requires the hyperbolic distributions $\phi_{F_n}(t)$ to be prepared on sets of ancilla qubits.

8.3 Results

To demonstrate the effectiveness of our proposed framework on NISQ hardware, we present results of simulations of basic open quantum systems on noisy trapped ion quantum computers, specifically on the IonQ Harmony [241] and Quantinuum H1-1 [242] devices. These devices have slightly higher noise in comparison to state-of the-art devices, but the noise syndromes of these devices are well-understood [243],

making them a good platform for demonstrating NISQ-friendly algorithms and error mitigation strategies in high-throughput simulation tasks. Quantum circuits were designed and compiled via the qiskit software framework [244] for all IonQ results, and via the pytket framework [245] for all Quantinuum device results.

To start, we present simulation results for the Pauli channel and damped Schwinger oscillator models, and discuss how these models satisfy the criterion of the special cases discussed in Section 8.2.4. We then present results obtained from simulating the trajectory of the system's density matrix on real quantum hardware. Next, we focus on simulating a damped quantum harmonic oscillator (i.e. a single bosonic mode occupied by many particles), and demonstrate how error mitigation techniques can be applied to suppress the effect of noise on NISQ devices.

8.3.1 Continuous-Time Pauli Channel

The continuous-time Pauli channel is a type of open quantum system used to model the noise induced in a quantum computing device as a function of time [184], or to model decoherence in degenerate spin-1/2 states coupled to a spin bath [213]. In most cases, it is common to assume a trivial Hamiltonian $H = 0$, with unitary Lindblad operators L_n with respective error rates γ_n that generate an error syndrome on N_q-qubit states in the device. These Lindblad operators assume the form of a Pauli string

$$L_n = \sigma_{n1} \otimes \sigma_{n2} \otimes ... \otimes \sigma_{nN_q}, \tag{8.27}$$

where each $\sigma_{ij} \in \{I, X, Y, Z\}$ is a single-qubit Pauli operator and N_q is the number of qubits. Since $H = 0$ and the operators L_n are unitary, we have $[H, L_n] = 0$, which means that the system satisfies conditions (i)-(iv) as discussed in Section 8.2.4.1. In

addition, it can be shown that Pauli string Lindblad operators form a finite abelian group that satisfies Eq. (8.20) and Eq. (8.21) with $\ell_n = 2$ for each L_n. This means that the system admits a finite Kraus representation of the form (8.22) with Kraus operators

$$K_{\vec{m}}(t) = e^{-\Gamma t/2} \prod_{n=1}^{N} \sqrt{F^1_{2,m_n}(\gamma_n t)} (L_n)^{m_n} \tag{8.28}$$

which can be combined in superposition to obtain a factored time evolution operator

$$S(t) = \frac{1}{\sqrt{2^N}} \prod_{n=1}^{N} \left(\sqrt{1 + e^{-\gamma_n t}} I + \sqrt{1 - e^{-\gamma_n t}} L_n \right). \tag{8.29}$$

This operator can be efficiently simulated using a quantum circuit of the form shown in Figure 8.2. Since $\ell_n = 2$, only a single ancilla qubit is needed to prepare the distribution $|\phi_{F_n}(t)\rangle$ corresponding to each Lindblad operator and each evolution period t. This circuit requires one two-qubit gate for each Pauli component $\sigma_{ij} \neq I$ in the set of Lindblad operators. Alternatively, one can also simulate a Pauli channel using the non-factored Kraus operators in Eq. (8.28), making the observation that the time-dependent terms $e^{-\Gamma t/2} \prod_n \sqrt{F^1_{2,m_n}(\gamma_n t)}$ can be applied on a classical computer after measuring the initial state under the action of the powers L_n^m of each Kraus operator. This requires performing partial (or in the worst case, a full) tomography on the initial state; however, the result of these measurements can be re-combined classically to find the continuous system trajectory for all values of $t > 0$.

As a motivating demonstration, we simulate a two-qubit Pauli channel under strongly correlated XX and ZZ interactions. This specific Pauli channel model is relevant in modeling qubit crosstalk in trapped ion quantum processors. For this

system we use the Lindblad operators

$$\begin{aligned} L_1 = I \otimes X, \quad L_2 = X \otimes I, \\ L_3 = Z \otimes Z, \quad L_4 = X \otimes X \end{aligned} \tag{8.30}$$

with single-qubit error rates $\gamma_1, \gamma_2 = 0.1$ and two-qubit error rates $\gamma_3, \gamma_4 = 1.0$. We simulated the trajectory of the initial unentangled pure state

$$|\psi(0)\rangle = -3/5\,|01\rangle - 4/5\,|11\rangle \tag{8.31}$$

on the noisy Quantinuum H1-1 device, using both the $S(t)$ and the full Kraus series method. In Figure 8.3, we plot the trajectory of the diagonal populations of the density matrix and the Z-basis Pauli operators estimated with 2048 shots per circuit. These results are compared with an exact numerical solution of the Lindblad equation.

Due to the strongly correlated nature of the simulated Pauli errors, we observe in Figure 8.3(a) that the system is driven from an unentangled pure state at $t = 0$ to a mixed, yet weakly correlated state at $t = 2.0$. While the single-qubit polarizations in the Z-basis decay to zero in Figure 8.3(b), the ZZ-correlation between the qubits decays at a much slower rate. In Figure 8.3, we also note the strong agreement between the Kraus series solution and the exact solution to the Lindblad equation. We also observe strong agreement between the results of the factored $S(t)$ method and the exact solution, but with increased noise due to the larger number of two-qubit gates used in the S_n circuits depicted in Figure 8.2.

8.3.2 Damped Schwinger Oscillator Model

The Schwinger Oscillator model is a 2D quantum harmonic oscillator system that can be equivalently viewed as a degenerate bosonic mode with two degrees of freedom. This model was first proposed by Julian Schwinger in 1952 to explain

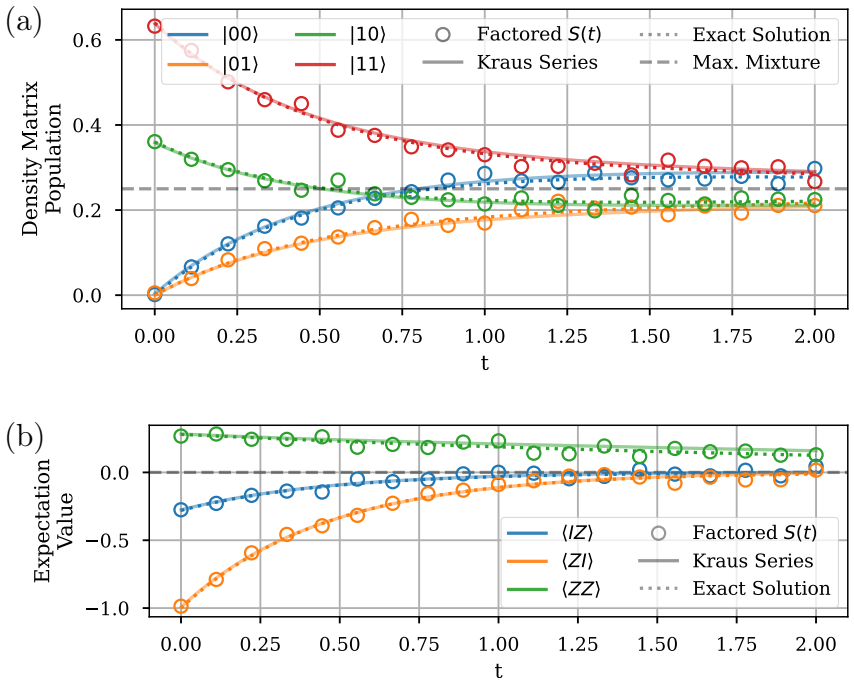

Figure 8.3: Trajectory of a two-qubit Pauli channel obtained from the noisy Quantinuum H1-1 device. Results are shown for (a) the populations along the diagonal of the density matrix and (b) the expectation values of Pauli strings in the Z-basis. For each plot, trajectories using the standard Kraus operator method (solid lines) and the factored $S(t)$ operator (circles) are shown.

the theory of quantum angular momentum with two uncoupled oscillators [234]. A degenerate 2D oscillator has the diagonalized Hamiltonian

$$H = \hbar\omega(1 + \hat{a}_1^\dagger \hat{a}_1 + \hat{a}_2^\dagger \hat{a}_2). \tag{8.32}$$

Up to addition of scalar terms, this Hamiltonian can be written as a quadratic product of bosonic creation and annihilation operators of the form in Eq. (8.17) with the matrix $M = \hbar\omega I$, where I is the 2×2 identity. Following the analysis in Section 8.2.4.2, we can efficiently simulate quadratic Lindblad operators of the form in Eq. (8.18), provided that the associated matrices M and V_n satisfy conditions (i)-(iv) in place of H and L_n respectively. For the 2-mode case, a natural choice of V_n is the set of Pauli operators (X, Y, Z), which generate $SU(2)$. In their bosonic quadratic form, these are proportional to the quantum angular momentum operators, as first proposed by Schwinger:

$$\hat{J}_x = (\hat{a}_1^\dagger \hat{a}_2 + \hat{a}_2^\dagger \hat{a}_1)/2 \tag{8.33}$$

$$\hat{J}_y = (\hat{a}_1^\dagger \hat{a}_2 - \hat{a}_2^\dagger \hat{a}_1)/2i \tag{8.34}$$

$$\hat{J}_z = (\hat{a}_1^\dagger \hat{a}_1 - \hat{a}_2^\dagger \hat{a}_2)/2 \tag{8.35}$$

Since $M = \hbar\omega I$ and the Pauli operators $V_n \in \{X, Y, Z\}$ meet the conditions for satisfying (i)-(iv), we can simulate the evolution of the system under a subset of the Lindblad operators Eqs. (8.33)-(8.35) using a time-perturbative Kraus series of the form in Eq. (8.6) with $f(t) = t$.

In accordance with convention, the $\hat{J}_z$ operator is chosen to be diagonal, though we recall that an arbitrary angular momentum operator $\hat{J}_{\hat{n}}$ about an axis $\hat{n}$ is similar to $\hat{J}_z$ under the unitary transformation $\exp(-i\pi J_{\hat{v}})$, where $\hat{v} = (\hat{n} + \hat{z})/\|\hat{n} + \hat{z}\|$. As a result, one can always choose at least one Lindblad operator to have the diagonalized

representation $\hat{J}_{\hat{z}}$, which can be applied more efficiently on a quantum computer. When a particular angular momentum operator (e.g. $\hat{J}_z$) is applied as a Lindblad operator, it produces a depolarization in the angular momentum of the oscillator about the given axis while conserving the total energy of the system. This can model, for example, the decoherence experienced by a charged particle occupying degenerate quantized Landau levels under a strong uniform magnetic field. (In this particular example, one recovers the 2D oscillator representation when the magnetic field is treated in the symmetric gauge.) The quantization of these degenerate Landau levels is relevant in explaining the Shubnikov-de Haas effect [246] and the integer quantum Hall effect [247].

To demonstrate the application of our methods to this system, we simulate a 2D oscillator with a single occupied doubly-degenerate Landau orbit level with energy $E_L = 2\hbar\omega$. Simulating up to the first degenerate level requires two qubits, where the degenerate subspace is spanned by the logical states $|01\rangle$ and $|10\rangle$. We apply angular damping $L_1 = J_z$ to an initially pure combination of the ground and excited states

$$|\psi(0)\rangle = (|00\rangle + i\,|01\rangle + |10\rangle)/\sqrt{3}. \tag{8.36}$$

and estimate the trajectory of the full density matrix by performing complete Pauli-basis tomography on the system qubits. Quantum simulations were carried out on the Quantinuum H1-1 device to reconstruct the full system density matrix at five different points in time with 1024 shots per circuit.

In Figure 8.4 we plot the expectation value of the trajectory of the system in mass-independent coordinates using the operators $x_0 = (\hat{a}_1^\dagger + \hat{a}_1)/\sqrt{2}, y_0 = (\hat{a}_2^\dagger + \hat{a}_2)/\sqrt{2}$ and their momentum analogues $p_{x0} = i(\hat{a}_1^\dagger - \hat{a}_1)/\sqrt{2}, p_{y0} = i(\hat{a}_2^\dagger - \hat{a}_2)/\sqrt{2}$,

where $\hat{a}_1, \hat{a}_2$ are restricted to act on only the lowest modes $|0\rangle$ and $|1\rangle$. For both the position and momentum coordinates, we observe strong agreement between the exact solution and the estimated values (for both the noisy H1-1 device and a simulated ideal noiseless device).

As $t \to \infty$, the position and momentum coordinates asymptotically approach zero, which is consistent with the depolarization of angular momentum in the z-direction. The depolarization can be visualized by reconstructing the full position and momentum distributions from the density matrix estimated on the noisy device. We expand the density matrix $\rho_{in,i'n'}$ in the position and momentum basis as a sum of products of Hermite-Gauss polynomials, given by

$$\rho(\vec{x_0}) = \sum_{n,n'} \prod_{i,i'} \left(\rho_{in,i'n'} \frac{\exp(-(\vec{x_{0i}})^2/2)}{\sqrt{2^{n+n'}\pi(n!)(n'!)}} H_n(\vec{x_{0i}}) H_{n'}(\vec{x_{0i'}}) \right) \tag{8.37}$$

where i, i' are the coordinate (i.e. mode) indices and n, n' are the coordinate eigenstate indices. The analogous momentum density can be obtained through a Fourier transform of Eq. (8.37). In Figure 8.5 we plot the position and momentum density estimated on the Quantinuum H1-1 device for the first four points in time.

8.3.3 Damped Harmonic Oscillator with Error Mitigation

So far, we have simulated relatively small systems on NISQ hardware, where the effect of noise is on the same order of magnitude as the sampling error. However, when simulating much larger systems on NISQ devices, the effect of noise is more problematic and must be combated through techniques such as hardware-native gate optimization and error mitigation [248, 249]. In this section, we focus on error-mitigation techniques, as circuit optimization is more hardware specific. In particular,

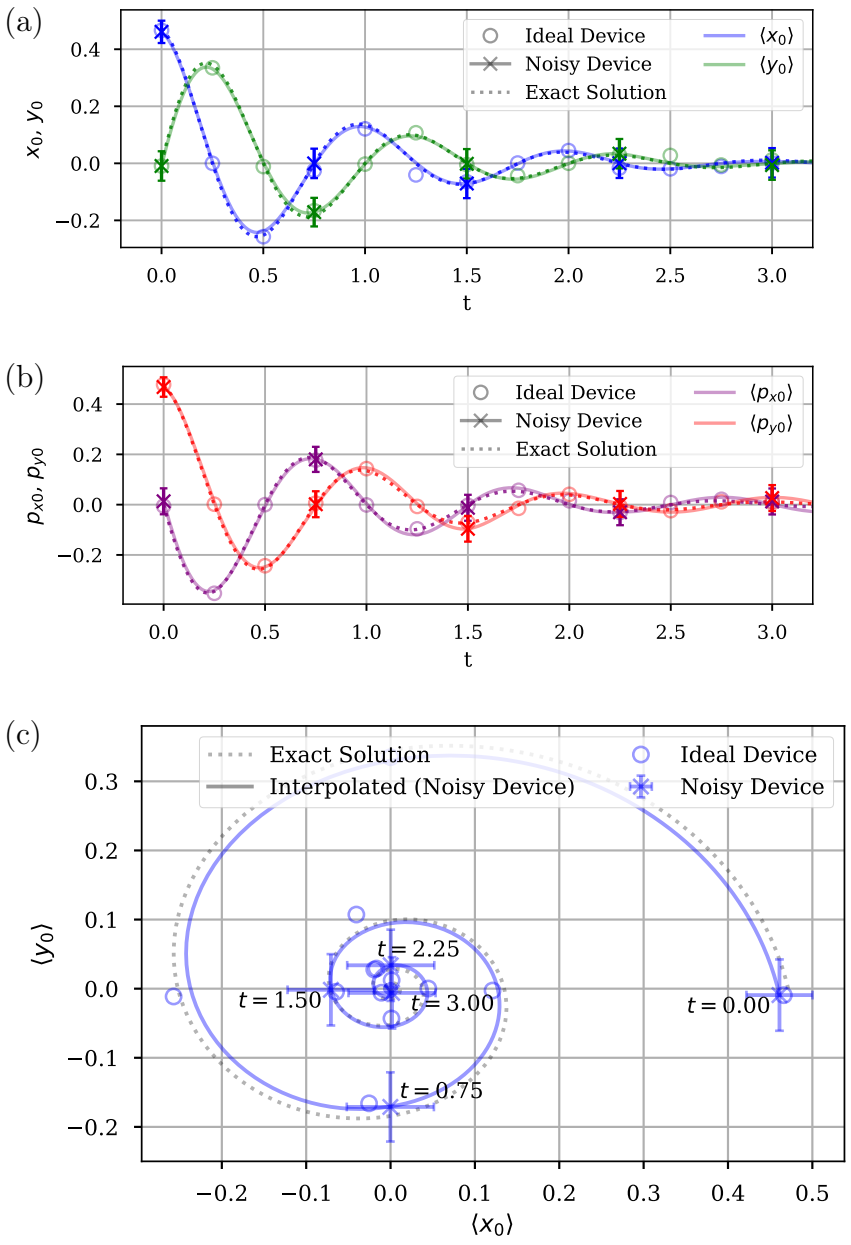

Figure 8.4: Trajectory of the expectation values of position and momentum for oscillator 1 (x_0, p_{x0}) and oscillator 2 (y_0, p_{y0}) computed on the noisy Quantinuum H1-1 device. Trajectories are shown for (a) the scaled position operators (b) the scaled momentum operators, and (c) for (x_0, y_0) joint spatial coordinates. The noisy results are compared to both an ideal quantum computation (containing only sampling error) and the exact solution to the Lindblad equation. The solid lines in plots (a)-(c) show an interpolation of the noisy data corresponding to the trajectory of a damped classical harmonic oscillator model with frequency ω.

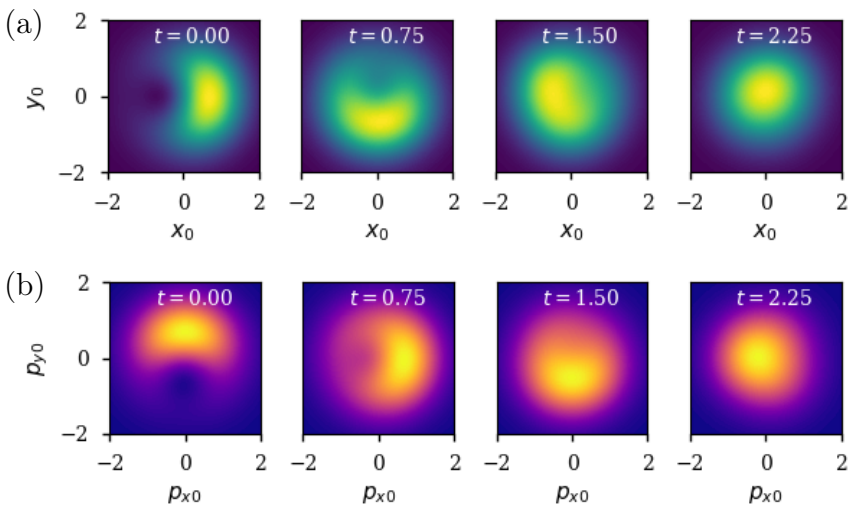

Figure 8.5: Snapshots of the trajectory of the joint position (a) and momentum (b) density of the Schwinger oscillator system. These densities were estimated directly from the density matrices obtained from the noisy Quantinuum H1-1 device.

we propose hardware-agnostic error mitigation techniques that can be applied to improve the accuracy of the density matrix trajectories of bosonic systems simulated using the Kraus series method. We apply these techniques to a damped harmonic oscillator system with Hamiltonian Eq. (8.16) and Lindblad operator $L_1 = \hat{a}$ with damping parameter γ_1. We restrict the representation of $\hat{a}$ to incorporate occupied eigenstates up to $n = 3$ in a dense two-qubit binary encoding. This allows for a Kraus representation of the form in Eq. (8.6) with $f(t) = (1 - e^{-\gamma_1 t}))/\gamma_1$.

As with the Schwinger oscillator system, we estimate the trajectory of the two-qubit oscillator density matrix by performing full tomography in the Pauli basis. While this process of reconstructing the density matrix can be expensive (requiring measurement in nine different bases for each Kraus operator circuit), it provides a significant amount of data that can be used for error mitigation both at and above the hardware level. However, when simulating open quantum systems, the system density

matrix is not always pure, which is guaranteed to be the case when considering the unitary quantum evolution of a closed system on a noiseless device. Furthermore, the decoherence of the system qubits due to entanglement with ancilla qubits combines with the device noise acting on the system qubits to produce a state that is often more decoherent and has a larger entropy than desired. This problem of distinguishing the simulated noise from the device noise poses a unique challenge that remains to be satisfactorily addressed in the literature.

One simple method for mitigating device noise is to model the noise as some CPTP channel $\mathcal{E}(\rho)$ and attempt to partially mitigate device noise by inverting the channel. The inverted channel is applied to the noisy density matrix ρ to obtain a mitigated density matrix ρ_{mit}. As we have seen in Section 8.3.1, one such model is given by a Pauli channel. The evolution of a density matrix ρ for a constant time period under a Pauli channel with Lindblad operators L_n of the form in Eq. (8.30) can be always written in the simplified form

$$\mathcal{E}_{\text{PC}}(\rho) = \varepsilon_0 \rho + \sum_i \varepsilon_i P_i \rho P_i^\dagger, \tag{8.38}$$

where $P_i \in \{I, X, Y, Z\}^{\otimes N} \setminus \{I^{\otimes N}\}$ are non-trivial Pauli strings and $\varepsilon_i \in [0, 1]$ are the associated error probabilities that sum to unity. An important special case of the general Pauli channel is the quantum depolarizing channel (QDC), where the error probabilities ε_i for $i > 0$ are identical across the set of all P_i. Letting $\lambda = 1 - \varepsilon_0$, it can be shown that the Pauli channel simplifies to

$$\mathcal{E}_{\text{QDC}}(\rho) = (1 - \lambda)\rho + \lambda \left(\sum_{P_i \in \mathcal{P}} P_i \rho P_i^\dagger \right) \tag{8.39}$$

$$= (1 - \lambda)\rho + \frac{\lambda}{2^N} I^{\otimes N}. \tag{8.40}$$

One important property of the QDC is that the noise induced by this channel commutes with any simulated CPTP noise channel $\mathcal{E}_{\text{sim}}$, that is

$$\mathcal{E}_{\text{sim}}(\mathcal{E}_{\text{QDC}}(\rho)) = \mathcal{E}_{\text{QDC}}(\mathcal{E}_{\text{sim}}(\rho)) \tag{8.41}$$

for all density matrices ρ. This means that one can correct for device-induced depolarizing errors that occur before and during the simulation by applying an inverse depolarization channel to the measured density matrix after simulation is complete. Moreover, inverting $\mathcal{E}_{\text{QDC}}$ can be carried out quite efficiently, requiring only a rescaling of the diagonal and off-diagonal components of ρ to obtain the mitigated density matrix ρ_{mit}. Although the Pauli channel model in Eq. (8.38) is a much more flexible noise model, it does not exhibit the same commutativity as in Eq. (8.41) except in the case when $\mathcal{E}_{\text{sim}}$ is also a Pauli noise model. For more information on the process of fitting QDCs and Pauli channels to measurements on quantum hardware, we refer the reader to Appendix E.2. In the remainder of this section, we discuss the results of two experiments that apply the error mitigation techniques discussed above to the damped harmonic oscillator system.

8.3.3.1 Noisy oscillating state. First, we consider the simulation of the oscillating non-stationary state

$$|\psi(0)\rangle = \frac{1}{\sqrt{2}}(i\,|2\rangle + |3\rangle). \tag{8.42}$$

which we simulated on the trapped ion IonQ Harmony device with damping rate $\gamma_1 = 1.0$ and $\hbar\omega = 1.0$ using the Kraus series method truncated to order $m = 3$. The full density matrices were estimated via tomography in the Pauli bases for 19 time steps uniformly spaced in the interval $t \in [0, 3]$ with 1024 shots per circuit.

To mitigate the effect of noise decoherence on the Harmony device, we fit a Pauli channel model and a QDC model to the initial state and select the respective parameters ε_i and λ such that the fidelity of the density matrix reconstructed on the $t = 0$ density matrix is maximized. The results of the estimated Pauli channel and QDC parameters are shown in Figure 8.6(a) and 8.6(b) respectively. Since the QDC satisfies Eq. (8.40), an estimate of the parameter λ that is sufficient to at least correct for decoherence during state preparation and measurement can be obtained by selecting the smallest value of λ that maximizes the fidelity of the initial density matrix. As shown in Figure 8.6(b), estimating λ in this manner approximately maximizes the average fidelity of the mitigated density matrices across the entire trajectory. In Figure 8.6(c) and 8.6(d), we plot the fidelity and von Neumann entropy of the density matrices estimated on the noisy device. In these figures, the gap between the noisy device results and a simulated ideal device are shaded in red, illustrating the effect of device noise decoherence. We observe in Figure 8.6(d) that both the Pauli channel and QDC error models decrease the density matrix entropy to closer resemble that of an ideal device. These methods also increase the average fidelity of the reconstructed density as seen in Figure 8.6(d), especially in the regime where the entropy of the density matrix is low. On the other hand, in the high-entropy regime (where $t \approx 0.5$) both error mitigation techniques reduce the entropy below that of the exact solution, resulting in a lower density matrix fidelity than the raw unmitigated data. This is illustrative of a key trade-off that exists between the fidelity of the mitigated density matrix in the high-entropy regime and low-entropy regimes: as the degree of mitigation is increased (i.e. the noise channel parameters ε_i and λ are increased), the fidelity in the high-entropy regime increases at the cost of some fidelity of the

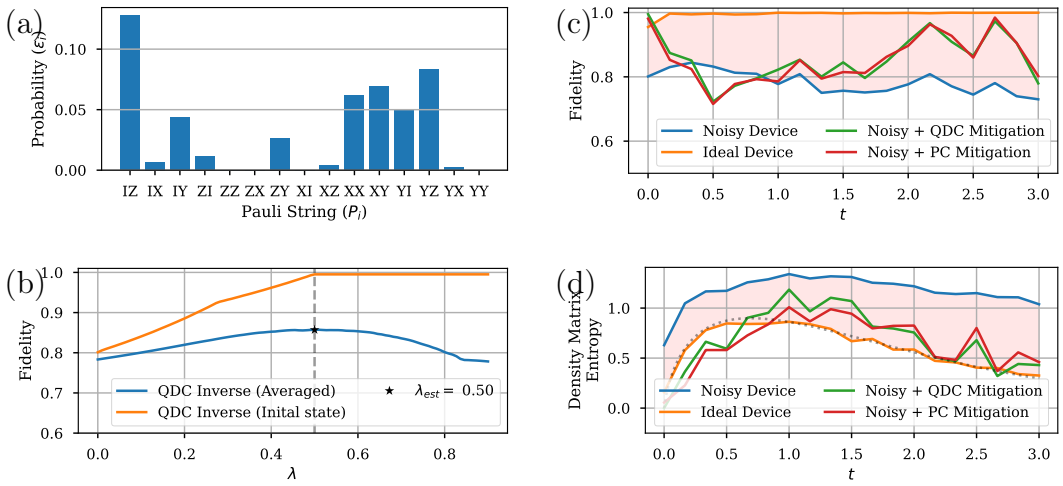

Figure 8.6: (a): Pauli channel model fit to the device-induced noise on the IonQ Harmony device. (b): Effect of the quantum depolarizing channel (QDC) parameter λ on the fidelity of the error-mitigated density matrix obtained from inverting a QDC fit to measurements obtained from the IonQ Harmony device. The average fidelity across the entire trajectory and only the prepared initial state are shown as a function of λ, suggesting an optimal value of $\lambda \approx 0.5$ for this device and system.
(c,d): Fidelity (c) and von Neumann entropy (d) of the density matrix trajectory simulated on the IonQ Harmony device. For each time value t, results are shown for an ideal device (only sampling error), the noisy device, and the noisy device with error mitigation via inversion of a general Pauli channel and a QDC fit to the initial state only.

lower-entropy states. We remark that the exact nature of this trade-off, however, is likely to be specific to both the system being simulated and the quantum hardware it is simulated on.

In Figure 8.7 we plot the error-mitigated expectation value of position and momentum of the damped oscillator system, where we see reasonable agreement with the exact solution. This agreement can be made stronger by interpolating the mitigated position and momentum values using the model of a damped classical harmonic oscillator with a variable damping coefficient. This interpolation is represented by the solid lines in Figure 8.7.

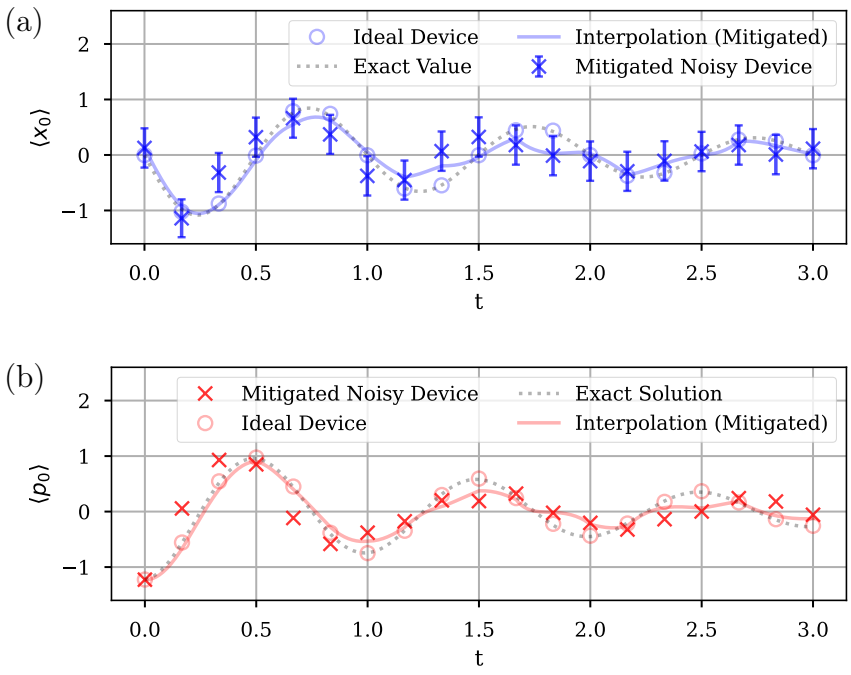

Figure 8.7: Trajectory of error-mitigated expectation values of oscillator position (a) and momentum (b) computed on the noisy IonQ Harmony device. The results for the mitigated density matrix, an ideal device, and the exact solution to the Lindblad equation are plotted for comparison. An interpolating trajectory of a classical harmonic oscillator with variable damping is also shown (solid line). This interpolation represents the least-squares error fit of the mitigated data.

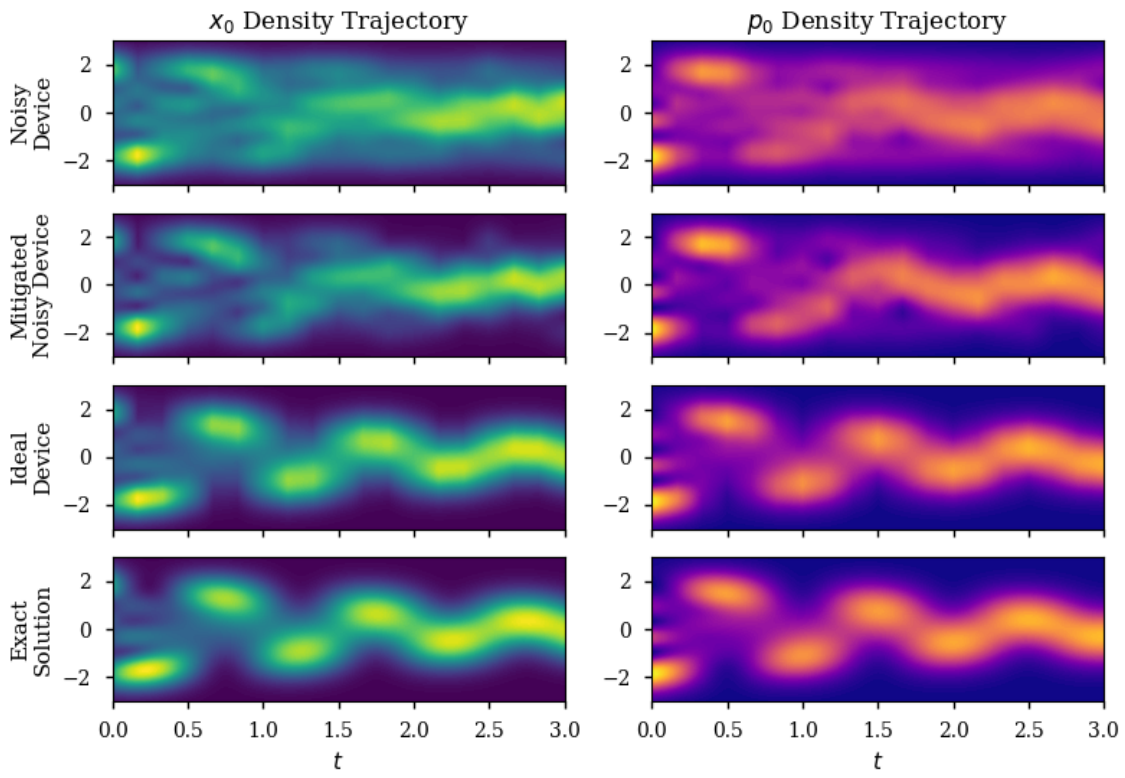

Figure 8.8: Trajectory of the position density (left column) and momentum density (right column) of the harmonic oscillator prepared in a non-stationary state. Densities are plotted for the noisy IonQ Harmony device results, the error-mitigated noisy results, an ideal quantum device trajectory, and finally the exact solution to the Lindblad equation. For the device results, the distributions between discrete time steps are estimated through Gaussian interpolation.

In Figure 8.11, we plot the trajectory of the position density and momentum density of the harmonic oscillator system interpolated as function of time. For each discrete time sample, the position and momentum densities were computed using Eq. (8.37) and interpolated between time steps with Gaussian smoothing. For comparison, the exact solution to the Lindblad equation is plotted in the last row of Figure 8.11. Although the unmitigated results appear to be degraded with noise, we see that much of this noise is suppressed after the inverse QDC error mitigation is applied. Nonetheless, the effect of noise is still prominent in the high-entropy regime at $t \approx 1.0$.

As the oscillator evolves, the dynamics of the oscillator transition from behaving like a quantum-mechanical object, where measurements of position and momentum are governed by the uncertainty principle, to a classical statistical object, where measurement outcomes can be described by a joint probability function of position and momentum. This transition can be observed by computing the Wigner quasi-probability distribution of the harmonic oscillator as a function of time. The Wigner quasi-probability distribution of a one-dimensional system with density operator $\hat{\rho}$ is given by:

$$W(x_0, p_0) = \frac{1}{\pi} \int \langle x_0 + s| \, \rho \, |x_0 - s\rangle \, e^{i2p_0 s} \, ds \tag{8.43}$$

The Wigner distribution plays a crucial role in the phase-space formulation of quantum mechanics and can be interpreted as a joint distribution over the position and momentum phase space of a system. Unlike a standard probability distribution, this quasi-probability distribution can sometimes take on negative values, which are attributed to inherently quantum-mechanical effects arising from the non-commutativity of x_0 and p_0. Furthermore, when the quasi-probability distribution is entirely positive, it is indicative of classical statistical behavior (e.g., an ensemble of Gaussian states) [250].

For a harmonic oscillator with density matrix $\rho_{n,n'}$ (where n, n' are eigenstate indices), the Wigner quasi-probability distribution can be computed in closed form using the upper-triangular entries of ρ [251]. This is achieved with the expansion

$$W(x_0, p_0) = \frac{e^{-(x_0^2+p_0^2)}}{\pi} \sum_n \sum_{n' \geq n} c_{n,n'} \, \mathrm{Re}\{\rho_{n,n'} [\sqrt{2}(x_0 + ip_0)]^{n'-n}\} \tilde{L}_n^{(n'-n)}(x_0^2 + p_0^2), \tag{8.44}$$

where $\tilde{L}_n^{(k)}$ are the generalized Laguerre polynomials and $c_{n,n'} = (-1)^n (2 - \delta_{n,n'}) \sqrt{n!/n'!}$.

In Figure 8.9, we plot the Wigner quasi-probability distributions reconstructed from

the density matrices obtained on the IonQ Harmony device (both with and without mitigation) and an ideal device. On the ideal device, we observe that the negative regions of the Wigner distribution (shown in blue) decay rapidly, vanishing at roughly $t \approx 0.6$, yet these regions are present in the noisy device results as late as $t \approx 1.6$. This suggests that applying mitigation to account for device noise may effectively blur the quantum-to-classical transition of the system. From comparing the mitigated and ideal Wigner distributions, we also note the existence of phase mismatching with respect to the ideal distribution, meaning the regions of high quasi-probability mass are rotated either clockwise or counter-clockwise relative to the ideal distribution. This same phase mismatching is observed in Figure 8.7, and can be partially attributed to the Harmony device's less precise execution of arbitrary-angle phase gates needed to apply the effective Hamiltonian evolution $T(t)$.

8.3.3.2 Noisy cat state. In this section, we briefly consider the simulation of a bosonic "Schrodinger cat" harmonic oscillator state [252]. We give special consideration to the simulation of this state due to the fact that it exhibits parity symmetry that is preserved under the evolution of the system with damping. Due to this property, cat states have been proposed as a platform for realizing discrete qubit states in superconducting and quantum optical systems, and they are instrumental to the implementation of quantum bosonic codes for fault-tolerant computation [217]. Here, we examine the dynamics of the same two-qubit truncated harmonic oscillator representing a single bosonic mode. This mode is prepared in the approximate two-qubit odd cat state

$$|\psi(0)\rangle = \frac{1}{\sqrt{|\alpha|^2 + |\alpha|^6/3!}} \left(\frac{\alpha^1}{\sqrt{1!}} |1\rangle + \frac{\alpha^3}{\sqrt{3!}} |3\rangle \right). \tag{8.45}$$

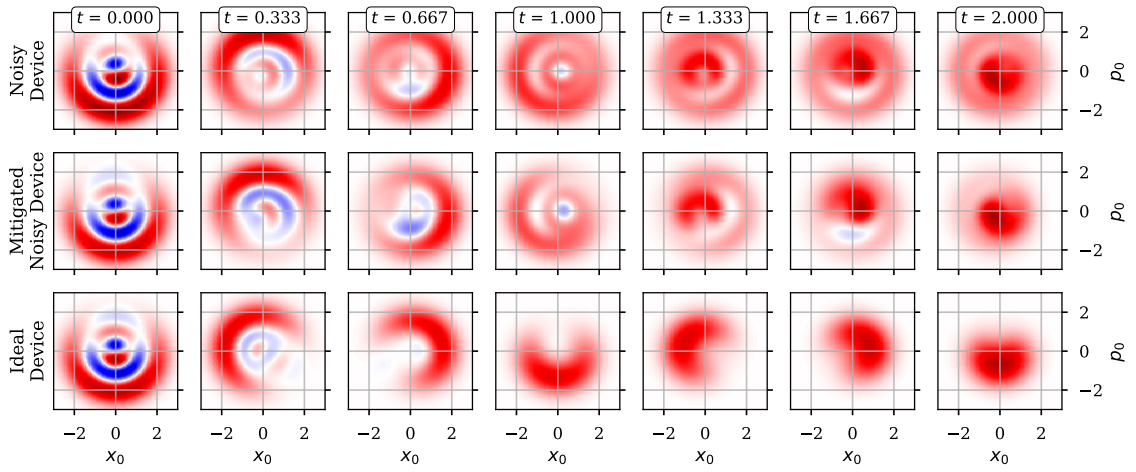

Figure 8.9: Estimated Wigner quasi-probability distribution of the harmonic oscillator at selected points t along the trajectory. For comparison, the results for the raw IonQ Harmony device (first row), the error-mitigated IonQ Harmony device (second row), and an ideal noiseless simulator (third row) are shown. The red shaded regions indicate where $W > 0$, while the blue shaded regions indicate the regions where $W < 0$. These regions where W is negative correspond to phase space configurations of the system which cannot be explained under a purely classical treatment of the system.

where $\alpha \in \mathbb{C}$ is a coherent state number. This state is invariant under the parity operator $\hat{\tau} = \sum_n (-1)^n |n\rangle \langle n|$, and remains invariant when evolved with the Lindblad operator $L_1 = \hat{a}$. In the context of bosonic systems, the Lindblad operator $L_1 = \hat{a}$ can be applied to model bosonic particle loss with decay rate γ_1.

As the system is simulated on a quantum device, both device noise and sampling error can result in this symmetry being violated. By incorporating this symmetry directly into error mitigation protocols, however, we can impose this symmetry upon the mitigated result, producing results with much higher fidelity than if symmetry is ignored. For example, to impose parity symmetry on the density matrix, we can apply the twirling operation of the group generated by the parity operator $\hat{\tau}$,

given by

$$\mathcal{T}_{\hat{\tau}}(\rho) = \frac{1}{2}(\rho + \hat{\tau}\rho\hat{\tau}^{\dagger}). \tag{8.46}$$

After the $\mathcal{T}_{\hat{\tau}}$ operation is applied to symmetrize the density matrix, error-mitigation schemes such as QDC inversion can be applied to the system to correct for additional device noise.

To provide a more concrete overview of how this can be incorporated into the error mitigation methods presented in the previous section, we present results obtained from a noisy emulation of the Quantinuum H1-1 device for the initial state in Eq. (8.45) with $\alpha = 1.2$ and $\gamma_1 = 0.6$. We simulate this system using the Kraus series method for nine uniform time steps from $t = 0.0$ to $t = 2.0$. In Figure 8.10 we plot the fidelity and entropy of the cat state trajectory estimated from the noisy device emulation and consider the effects of applying the twirling operation in Eq. (8.46) and subsequent QDC inverse error mitigation. From these results, we note that twirling alone provides only a modest increase in overall state fidelity; however, when it is combined with the QDC error mitigation scheme, it provides a significant boost to the overall state fidelity, especially for the initial prepared state at $t = 0$.

In Figure 8.11 we plot the position and momentum densities reconstructed from the noisy simulation results. In the context of bosonic systems, we note that the position and momentum density can be interpreted in the context of the over-complete basis of coherent state numbers $\alpha \in \mathbb{C}$, where $x_0 = \mathrm{Re}[\alpha]$ and $p_0 = \mathrm{Im}[\alpha]$. In the raw noisy position and momentum data (first row) we observe a significant bias that is present in the regions where $x_0 > 0$ and $p_0 > 0$, which is consistent with a strong phase bias in the system qubits, resulting in the appearance of oscillations

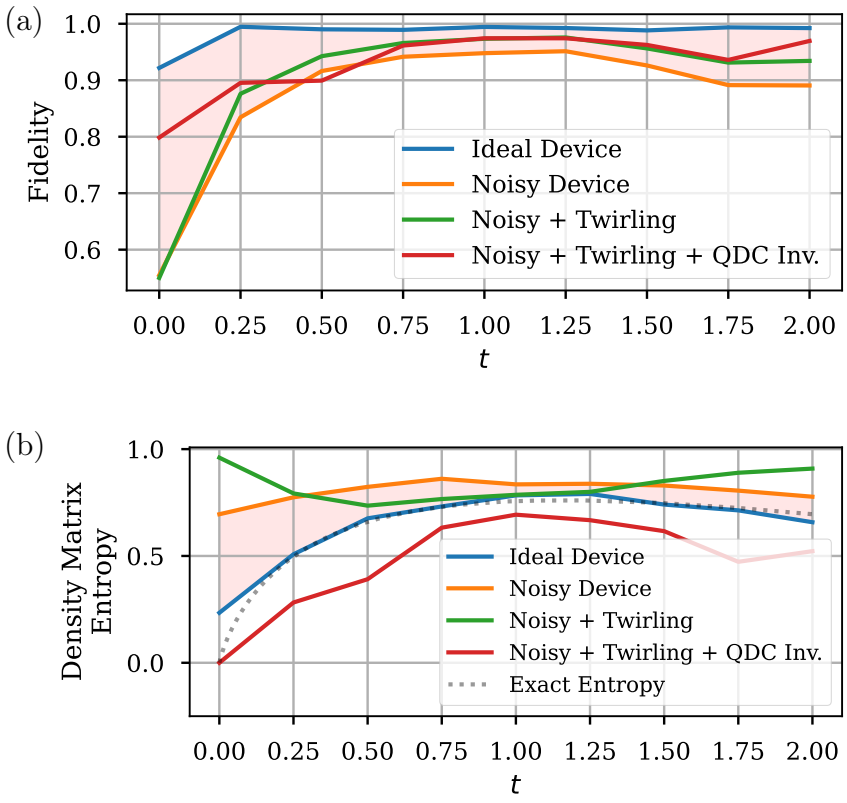

Figure 8.10: Fidelity (a) and von Neumann Entropy (b) of the density matrix trajectory obtained through a noise model emulation of the Quantinuum H1-1 device. Results are shown for an ideal noiseless device and the simulated noisy device with no mitigation, with twirling, and with combined twirling and error mitigation. Error mitigation was applied using parity twirling and QDC inversion with a value of $\lambda_{\text{est}} = 0.48$, which was obtained by fitting a QDC to the first state. The ideal trajectory of the system was obtained through a noiseless simulator with the same number of measurements as the noisy device.

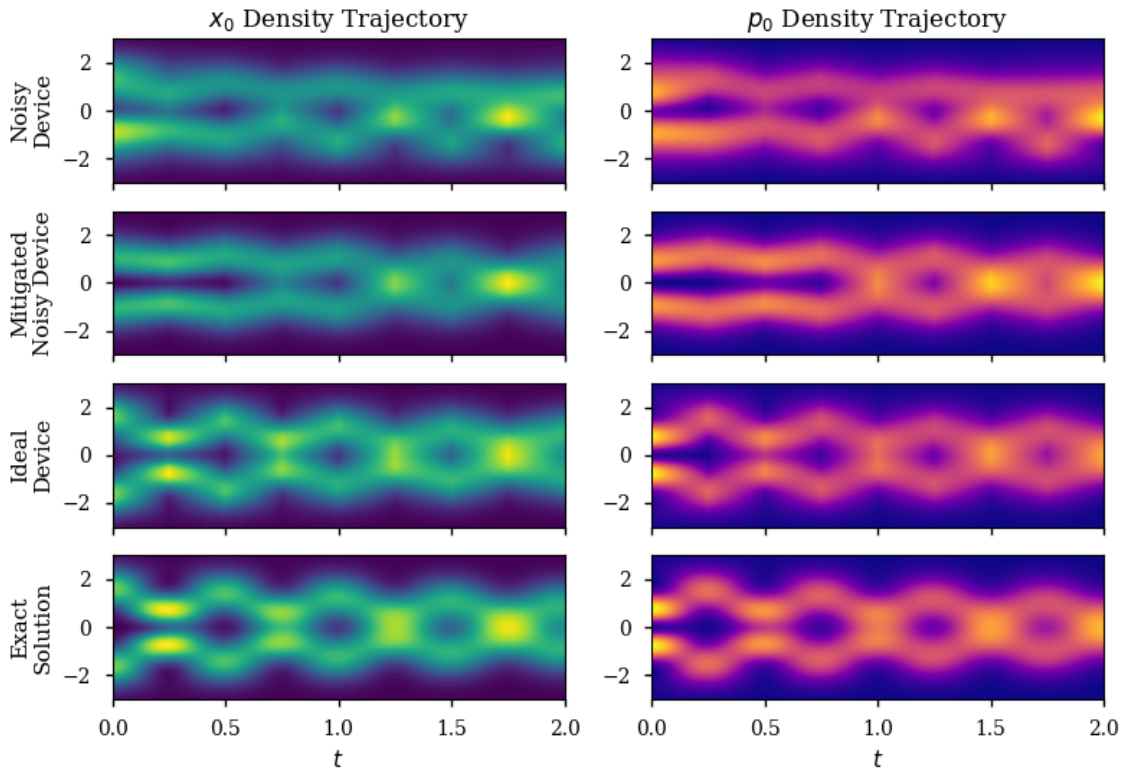

Figure 8.11: Trajectory of the position density (left column) and momentum density (right column) of the harmonic oscillator prepared in an approximate two-qubit cat state. Densities are shown for a noisy emulated Quantinuum H1-1 device trajectory, the error-mitigated noisy trajectory, an ideal quantum device trajectory, and finally the exact solution to the Lindblad equation.

when $x_0, p_0 < 0$, and no oscillations when $x_0, p_0 > 0$. Since we expect the trajectory to exhibit parity symmetry, we observe that the application of twirling and device noise mitigation produces a more faithful representation of the system dynamics.

In Figure 8.12 we plot the estimated Wigner quasi-probability distributions of the emulated noisy device trajectory, and we compare these distributions to those of an ideal device. In Figure 8.10), we observed that even the initial state at $t = 0$ was reproduced with low fidelity. From inspecting the Wigner distributions, we observe that this low fidelity partially manifests as broken parity symmetry. When parity twirling is applied to a harmonic oscillator state, the Wigner distribution of that state

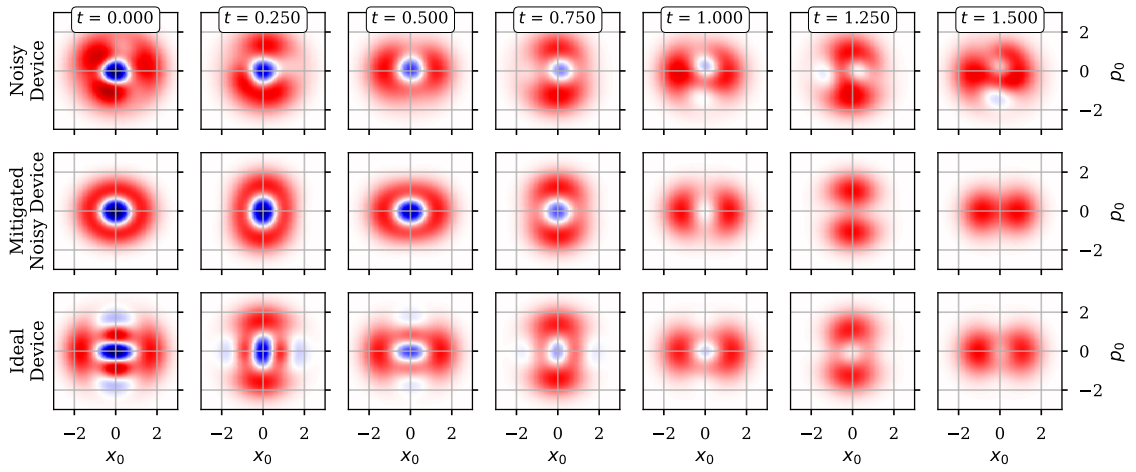

Figure 8.12: Wigner quasi-probability distribution of the harmonic oscillator modeling a single bosonic mode at selected points t along the trajectory. Results are shown for the emulated Quantinuum H1-1 device (first row), the error-mitigated Quantinuum H1-1 device (second row), and an ideal noiseless simulator (third row). The red shaded regions indicate where $W > 0$, while the blue shaded regions indicate the regions where $W < 0$. These regions where W is negative correspond to phase space configurations of the system which cannot be explained under a purely classical treatment of the system.

is averaged with its own 180° rotation. It follows that an oscillator state exhibits parity symmetry if its Wigner distribution is invariant under 180° rotations. We observe that this is the case for both the mitigated and ideal Wigner distributions, however the mitigated results at around $t = 0$ exhibit distributions that appear to be more consistent with a single excited quantum state (i.e. $\rho \approx |1\rangle\langle 1|$) than with the approximate cat state in Eq. (8.45). The Wigner distributions of cat states are characterized by the interference pattern of alternating positive and negative regions in phase space which separate two regions of high quasi-probability mass, which are not discernible in the mitigated results for $t < 1.0$. Although we observe these deviations when the system is in the quantum mechanical regime at $t < 1.0$, we also see that in the classical statistical regime at times $t > 1.0$, there is strong agreement between the mitigated results and the ideal results. This suggests that parity twirling

and subsequent error mitigation is very effective when applied to noisy simulations of systems with more classical character, but may not be able to restore all of the important quantum-mechanical effects in systems with a strong quantum character.

8.4 Conclusion

We introduced novel methods for efficiently simulating open quantum systems on NISQ devices, to overcome the challenges posed by noise and limited qubit resources. Our approach used Kraus operators to represent the time evolution of open quantum systems. Unlike traditional Trotter-based methods, which can be computationally expensive and noisy due to the large number of quantum gates needed, our method allowed for efficient simulation by avoiding costly Trotterization for certain classes of systems. This is particularly useful for systems that satisfy the commutation relations (i)-(iv), enabling the simulation to be carried out in closed form without requiring diagonalization. We extended the Kraus operator method by deriving a time-perturbative Kraus series that can be applied to simulate the system's evolution under various conditions, such as systems with unitary Lindblad operators, bosonic Lindblad operators, and systems where the Lindblad operators exhibit finite group or semigroup structure. The series was designed to converge rapidly (especially for bosonic systems), making it practical for NISQ devices. We proposed an efficient way to map the Kraus operator series to quantum circuits, making it feasible to simulate these systems on NISQ hardware. The mapping reduced quantum circuit depth and gate count, directly addressing the limitations of NISQ devices in terms of noise and qubit fidelity. We also presented novel error mitigation strategies tailored for open quantum systems simulated on NISQ devices. By fitting noise models such as

Pauli and quantum depolarizing channels and applying error mitigation techniques, we demonstrated a significant improvement in the accuracy of the simulations. This approach is particularly relevant for large systems where noise significantly degrades the fidelity of results. Finally, we simulated open quantum systems on real NISQ devices (IonQ Harmony and Quantinuum H1-1), showcasing the practicality and effectiveness of our methods. The novel use of these error mitigation techniques with actual quantum hardware is a critical step in bridging theoretical methods and real-world quantum computing applications. The combination of Kraus-based methods with efficient circuit mapping and error mitigation strategies represents a significant step forward in the simulation of open quantum systems on NISQ devices.

Our work opens up several avenues for future research. As quantum hardware continues to improve, the methods developed here can be further refined and scaled to tackle more complex quantum systems with larger qubit counts and deeper circuits. An important direction is the extension of these techniques to more diverse classes of open quantum systems, including non-Markovian dynamics, where system-environment correlations persist over time [253, 254]. Developing tailored algorithms for these scenarios could unlock more realistic modeling of quantum processes in chemistry, biology, and materials science.

Furthermore, enhancing the synergy between quantum hardware and software through hardware-aware optimizations is crucial. Circuit depth remains a limitation, but advances in error correction, gate synthesis, and qubit connectivity could allow for more efficient implementations of the Kraus method on future quantum architectures. Collaborative efforts between quantum hardware developers and algorithm designers will be key to pushing the boundaries of quantum simulations on NISQ devices.

As quantum error mitigation techniques continue to evolve, integrating them with dynamic learning algorithms or hybrid quantum-classical approaches may further improve the reliability of quantum simulations. As we approach fault-tolerant quantum computing, these methods could evolve into robust tools for simulating highly entangled, dissipative systems that are beyond the reach of classical computation.

8.4.1 Data Availability Statement

The code, data, and demo notebooks necessary to reproduce these results can be found in the associated Github repository [255].

8.4.2 Acknowledgments

N. B. and G. S. acknowledge support by the National Science Foundation under award DGE-2152168 and the Department of Energy under awards DE-SC0024325 and DE-SC0024328. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. This research was also supported by compute credits from IonQ via the Microsoft Azure Quantum Credits program.

8.4.3 Author Contributions

C.B. and E.P.B conceptualized the work. C.B. developed the theory and code. C.B., N.B., and G.S. conducted the quantum hardware experiments. All authors contributed to the writing and review of the manuscript.

CHAPTER NINE

Conclusion and Future Work

9.1 Summary of this Dissertation

This dissertation has investigated non-classical computational paradigms for the simulation and design of quantum materials. Motivated by the limitations of conventional mean-field electronic structure methods and the exponential scaling of rigorous many-body approaches, this work explored two complementary directions: (i) machine learning models for accelerated materials discovery and electronic structure prediction, and (ii) quantum computational algorithms for the simulation of open quantum systems relevant to condensed matter and materials physics.

The central thesis of this work is that machine learning and quantum computational methods can meaningfully reduce the gap between computational tractability and physical rigor in quantum materials research. Rather than replacing established *ab initio* approaches, these methods augment and extend them, enabling scalable approximation, structural optimization, and simulation of systems that are otherwise classically intractable.

9.1.1 Review of Objectives

The primary objectives outlined in Chapter One were:

(1) To develop machine learning architectures capable of accelerating high-throughput materials discovery and electronic structure prediction while preserving key physical symmetries.

(2) To construct a quantum-informed neural network framework capable of predicting ground-state electronic band structure in periodic solid-state systems.

(3) To design and analyze quantum algorithms for simulating open quantum systems with improved circuit depth and noise resilience relative to conventional Trotter-based approaches.

(4) To experimentally validate these quantum simulation methods on contemporary noisy intermediate-scale quantum (NISQ) hardware.

These objectives were pursued through the development of machine learning frameworks, new models, open-source software infrastructure, quantum algorithmic approaches, and quantum hardware demonstrations.

9.1.2 Review of Research Contributions

9.1.2.1 Machine learning. In the machine learning domain, this dissertation made three principal contributions.

First, a high-throughput screening framework for superconducting materials was developed using graph neural network architectures, including ALIGNN-based models. This framework demonstrated strong predictive performance for superconducting critical temperature and enabled the identification of candidate materials for further experimental investigation. By integrating structural optimization, defect modeling, and supervised learning, this work illustrated a scalable pathway

for accelerating materials discovery. Second, this dissertation introduced Quantum-Informed Neural Networks (QuINNs), a class of equivariant models designed to directly parameterize electronic structure quantities. Unlike purely data-driven regressors, QuINNs embed physically motivated structure—such as Gaussian orbital parameterizations, equivariant representations under $E(3)$ symmetry, and Hamiltonian construction—into the architecture itself. This allows for direct prediction of electronic band structure while bypassing self-consistent field loops typical of density functional theory (DFT). Third, the PyQuINN software framework was developed to unify data structures, equivariant neural network primitives, Gaussian orbital Hamiltonian construction, and backend-agnostic compilation. By integrating database workflows and automatic differentiation through modern machine learning backends, PyQuINN provides a modular platform for scalable electronic structure learning and optimization.

Collectively, these contributions demonstrate that quantum-informed ML architectures can bridge the gap between empirical regression and physics-based simulation, offering both computational efficiency and interpretability.

9.1.2.2 Quantum computing. In the quantum computing domain, this dissertation introduced a novel framework for the simulation of Markovian open quantum systems using a time-perturbative Kraus series representation.

The principal theoretical contribution was the derivation of a Trotterless simulation method based on superoperator algebra and structured commutation relations of Lindblad operators. By transforming the Lindblad equation into an environmental

interaction picture and constructing a Kraus operator series with bounded truncation error, this approach distributes computational complexity across many shallow circuits rather than a single deep circuit.

This framework was shown to reduce effective circuit depth relative to standard Trotter-Suzuki methods, making it particularly well-suited to NISQ hardware where noise accumulation is a limiting factor. Explicit constructions were provided for multi-qubit continuous-time Pauli channels and damped quantum harmonic oscillator systems, along with rigorous error and resource analysis.

Finally, these methods were experimentally implemented on trapped-ion quantum hardware. Through circuit compilation, noise modeling, and error mitigation techniques, open quantum system trajectories were reconstructed and compared to analytical solutions of the Lindblad equation. These demonstrations provide empirical evidence that structured Kraus-series methods can enhance the practical feasibility of open-system simulation on near-term quantum devices.

9.1.3 Significance of Contributions

The contributions of this dissertation lie at the intersection of computational physics, machine learning, and quantum information science. Their significance is best understood in terms of both methodological innovation and long-term impact on quantum materials research.

9.1.3.1 Machine learning. The machine learning contributions advance the state of scientific ML in three key ways.

First, they demonstrate that symmetry-aware and physics-informed neural architectures can achieve predictive performance while maintaining interpretability and

physical consistency. The explicit incorporation of equivariance under $E(3)$ transformations and the structured construction of Hamiltonians distinguish this work from black-box regression approaches.

Second, the QuINN framework establishes a paradigm in which neural networks do not merely approximate material properties but directly parameterize physically meaningful intermediate quantities—such as orbital coefficients and effective potentials—thereby preserving a clear connection to first-principles theory.

Third, by integrating high-throughput data generation, scalable training pipelines, and modular software infrastructure, this work lays the foundation for data-driven exploration of complex material families, including strongly correlated and topological systems. In the long term, such frameworks may enable inverse design workflows in which desired electronic features are engineered directly through differentiable band structure models.

9.1.3.2 Quantum computing. The quantum computing contributions are significant both theoretically and practically.

Theoretically, the time-perturbative Kraus series approach expands the toolkit for digital quantum simulation beyond Hamiltonian-only dynamics. By providing a constructive and analyzable framework for open-system simulation with bounded error, this work contributes to the broader understanding of how dissipative and non-unitary processes can be simulated efficiently on quantum computers.

Practically, the reduction of circuit depth through Trotterless decomposition addresses one of the central bottlenecks of NISQ-era computation: noise accumulation. The experimental demonstrations on trapped-ion hardware illustrate that structured algorithm design, when guided by operator algebra and system symmetries, can meaningfully extend the class of physically relevant systems that are accessible to current devices. More broadly, this work reinforces the perspective that quantum computing is not a long-term replacement for classical simulation, but a tool whose utility depends critically on algorithmic structure, hardware-aware optimization, and hybrid classical-quantum workflows.

In summary, this dissertation advances non-classical computational methods for quantum materials by developing symmetry-aware machine learning architectures and depth-efficient quantum simulation algorithms. Together, these contributions help close the gap between theoretical rigor and computational feasibility, providing new pathways for the simulation, understanding, and design of quantum materials.

9.2 Future Directions

The work presented in this dissertation establishes a foundation for symmetry-aware machine learning and structured quantum simulation methods for quantum materials. However, it also opens several new avenues of inquiry. The following research directions build directly upon the theoretical, algorithmic, and software contributions developed herein, and align with a broader vision of unifying electronic structure theory, machine learning, and quantum computation into a coherent computational framework.

9.2.1 Machine Learning

9.2.1.1 Existing perspectives. Machine learning has become increasingly prominent in computational materials science, particularly in high-throughput screening and surrogate modeling of electronic structure calculations. Graph neural networks, equivariant architectures, and message-passing models have demonstrated strong predictive performance for energies, forces, and materials properties. However, most existing approaches treat electronic structure as a regression problem, directly predicting scalar quantities such as total energy, band gaps, or Hamiltonian matrix elements.

While successful in narrow domains, such approaches often lack strong generalization across material classes and geometries. Moreover, directly regressing Hamiltonians or energies can obscure the underlying physical structure of the electronic wave function. Fixed basis sets in traditional electronic structure methods similarly limit adaptability and may scale poorly for heterogeneous or strongly correlated systems. As a result, there remains a conceptual and methodological gap between data-driven models and the symmetry-constrained structure of quantum mechanics.

9.2.1.2 Key challenges and limitations. Several fundamental challenges must be addressed to advance machine learning for electronic structure:

First, the representation problem: electronic wave functions must respect rotational, translational, and gauge symmetries. Although $E(3)$-equivariant networks partially address geometric symmetry, few methods explicitly model wave-function structure in a way that integrates seamlessly with established solvers.

Second, scalability and low-rank structure: practical materials modeling requires representations that compress electronic structure information without sacrificing fidelity. Learning adaptive basis functions that generalize across chemical environments remains a nontrivial problem.

Third, integration with physics-based solvers: ML models must interface robustly with density functional theory (DFT), tight-binding, and Green's function methods. Without principled integration, learned models risk remaining disconnected from physically interpretable workflows.

Finally, multiscale coupling remains largely unresolved. Bridging atomistic electronic structure to device-level quantum transport requires representations that remain stable and informative across scales.

9.2.1.3 Future directions of this work. Building upon the QuINN framework and the symmetry-aware models developed in this dissertation, future work will focus on learning symmetry-preserving representations of electronic wave functions rather than directly regressing energies or Hamiltonians.

The first objective is to construct $E(3)$-equivariant neural architectures that produce adaptive, local basis functions tailored to atomic environments. These learned basis functions will serve as low-rank, physically consistent building blocks for electronic structure calculations. Rather than replacing established solvers, the learned representations will be embedded within them as flexible alternatives to fixed Gaussian or plane-wave bases.

The second objective is to demonstrate that these learned bases generalize across material families and provide measurable computational acceleration. This includes benchmarking low-rank representations for complex systems such as correlated materials and topological phases.

The third objective is multiscale integration. By coupling learned electronic structure representations to non-equilibrium Green's function (NEGF) transport formalisms, future work will enable quantum-accurate simulations of nanoscale devices. In this setting, machine learning will serve not only as an accelerator but as a structural component of a unified atom-to-device modeling pipeline.

Finally, sustained effort will be directed toward open-source infrastructure, including expansion of the PyQuINN ecosystem, standardized benchmarking datasets, and integration layers connecting ML models with DFT and NEGF workflows. These tools will ensure reproducibility and broad community adoption.

9.2.2 Quantum Computing

9.2.2.1 Existing perspectives. Quantum computing offers a principled framework for simulating quantum many-body systems, with algorithms such as phase estimation, variational quantum eigensolvers, and Hamiltonian simulation providing theoretical advantages over classical methods. Significant effort has focused on ground-state energy estimation and unitary dynamics.

However, realistic quantum materials frequently involve open-system dynamics, dissipation, and finite-temperature effects. Simulation of such systems using Lindblad master equations or quantum channels remains resource-intensive, particularly

when implemented through Trotter-Suzuki decompositions that incur substantial circuit depth.

In the noisy intermediate-scale quantum (NISQ) regime, hardware constraints limit achievable circuit depth and coherence times, restricting the class of tractable simulations. Thus, practical quantum advantage depends not only on asymptotic complexity but on algorithmic structure that minimizes depth and maximizes robustness to noise.

9.2.2.2 Key challenges and limitations. Three major challenges define the frontier of quantum materials simulation on quantum hardware. First, *resource reduction*: mapping realistic materials Hamiltonians to qubit representations often requires large Hilbert spaces and deep circuits. Efficient encodings that exploit locality and symmetry are essential. Second, *hybridization*: purely quantum approaches remain limited by noise and scale. Effective workflows must leverage classical preprocessing, compression, and conditioning to reduce quantum resource demands. Third, open-system simulation requires *depth-efficient* methods that avoid excessive Trotterization while maintaining bounded error.

9.2.2.3 Future directions of this work. Building on the Trotterless Kraus-series framework developed in this dissertation, future work will extend structured open-system simulation methods to larger and more physically realistic systems, including correlated lattice models and quantum transport simulation. The primary direction of this work will be the integration of learned electronic structure representations with quantum algorithms. Machine learning models can be used to construct reduced-complexity Hamiltonians, compress basis representations, and precondition

systems prior to quantum execution. Such hybrid classical–quantum workflows aim to reduce qubit counts and circuit depth for phase estimation protocols, variational algorithms, and steady-state solvers.

Further work will also explore symmetry-adapted encodings that exploit conserved quantities and locality to produce hardware-efficient mappings of atomic structure to variational circuits and simulation methods. Physically structured ansatz circuits, guided by wave-function representations learned in the classical setting, may mitigate barren plateaus and improve trainability in simulations, especially for solid-state quantum materials.

In the longer term, these efforts will contribute to a hybrid computational paradigm in which classical machine learning organizes, compresses, and structures the problem, while quantum hardware executes the components that are known to be intractable classically. Such a synergistic approach moves beyond viewing quantum computing as a wholesale replacement for classical computation and instead positions it as a purposefully integrated component of the scientific computational stack.

9.3 Conclusion

This dissertation has advanced two complementary methodological frontiers: symmetry-aware machine learning for electronic structure and depth-efficient quantum algorithms for open quantum systems. Together, these contributions form the foundation of a broader research program aimed at unifying electronic structure theory, machine learning, quantum simulation, and multiscale modeling.

The future directions outlined above extend this foundation toward adaptive wave-function representations, hybrid classical–quantum workflows, and multiscale

atom-to-device modeling. By integrating principled symmetry constraints, scalable computational architectures, and hardware-aware algorithm design, this research aims to close the gap between physical rigor and computational feasibility.

Ultimately, the long-term vision of the work presented here is the development of a unified computational framework for quantum materials: one that is symmetry-aware, data-informed, quantum-enabled, and capable of addressing complex materials systems across scales. Such a framework would not only accelerate materials modeling but also deepen our understanding of quantum matter in regimes that remain beyond the reach of existing classical methods.

APPENDICES

APPENDIX A

ALIGNN Model Hyperparameter Optimization

In this appendix section, we give an overview of how hyperparameter search an optimization was applied to estimate the optimal configuration of the ALIGNN model as discussed in Chapter 3.

A.1 ALIGNN Model Hyperparameter Optimization

In training the ALIGNN model, we identified a total of 6 hyperparameters that we applied DOE to optimize. Two of these hyperparameters (the learning rate and the weight decay parameter) were training hyperparameters, whereas the remaining four parameters (the number of ALIGNN layers, CGCNN layers, hidden features and embedding features) were model hyperparameters. From training previous instances of the ALIGNN model on Baylor's Kodiak cluster GPU, we knew that the time required to train each model was roughly on the order of 8 hours using the default hyperparameters. Since we could effectively train up to two models concurrently, we figured that we could train the ALIGNN model with up to 20 distinct hyperparameter configurations without exceeding a budget of 80 hours (roughly one business week) of compute time. To suggest model hyperparameter configurations, we used the open-source PyDOE2 software package. We selected this package due to its transparent documentation and ease of integration with the Python3 programming language, which allowed us to fully automate the training procedure without having to manually copy data back and forth from a spreadsheet. The package also integrates nicely with code notebook platforms (such as JupyterLab and Juyter Notebooks), which

makes for easy and customizable data visualizations.

Since the space of factors we wanted to investigate was continuous and six-dimensional, we needed a sampling scheme that would effectively explore this high-dimensional space using relatively few samples. Since a full-factorial design would require at least 64 different experiments, we decided that a fractional factorial design without factor discretization would be the best option. The design scheme we decided upon was the Latin Hypercube Sampling (LHS) scheme. This scheme selects experimental configurations (whether continuous or discrete numeric) in a randomized fashion that minimizes the correlation between selected factors. This yielded the experimental design shown in Table A.1.

A.2 Results

A.2.1 Experiment Results

For each of the experimental configurations in Table A.1, we trained an instance of of the ALIGNN model. During each experiment, we recorded the number of training epochs needed for the model to converge (i.e. the number of times the model needs to see each example in the training set). To evaluate the performance of the model once it converged, we also reported the accuracy and the area under the receiver operating characteristic curve (ROCAUC) to measure the performance of the model for both the training and validation sets.

We note that the ROCAUC is the area under model receiver operating characteristic curve, which describes the trade-off between the true positive rate and false

Table A.1: A table of the 20 experiments we conducted, with recommended hyperparameters selected according to PyDOE2's Latin Hypercube Sampling (LHS) design scheme. The LHS scheme is ideal for designing experiments with many continuous and discrete numerical factors, but where relatively few experiments can be performed.

	Training Hyperparameters		Model Hyperparameters			
#	Weight Decay	Learning Rate	ALIGNN Layers	CGCNN Layers	Hidden Features	Embedding Features
0	0.000073	0.009738	7	6	135	74
1	0.000598	0.007298	1	5	167	205
2	0.000355	0.004050	3	4	195	223
3	0.000531	0.004785	3	6	100	162
4	0.000871	0.002788	4	3	141	135
5	0.000316	0.006193	5	1	250	91
6	0.000025	0.001143	6	4	160	179
7	0.000682	0.000968	6	6	75	243
8	0.000649	0.008000	7	2	172	251
9	0.000115	0.006608	2	4	185	67
10	0.000417	0.008260	5	5	103	121
11	0.000485	0.008562	2	6	208	236
12	0.000169	0.005525	4	5	212	190
13	0.000903	0.002065	2	2	91	212
14	0.000734	0.000334	3	7	72	96
15	0.000298	0.002444	5	1	244	156
16	0.000953	0.003555	4	4	226	175
17	0.000805	0.005783	2	3	130	145
18	0.000781	0.009087	4	3	112	104
19	0.000215	0.004221	6	2	229	129

Table A.2: A table of the experiment results, reporting the number of training epochs to achieve convergence, the model accuracy and receiver operating characteristics area under curve (ROCAUC) for both the training and validation set. Note that the true accuracy and ROCAUC of the model is best estimated by the validation set values, not the training set values. For reference, an ROCAUC value of 0.5 indicates random guessing, a value of 0.75 indicates good performance (for hard classification tasks), and an ROCAUC of 1.0 indicates perfect classification.

#	Training Epochs	Accuracy (Training)	ROCAUC (Training)	Accuracy (Validation)	ROCAUC (Validation)
0	11	0.907210	0.940193	0.881076	0.854032
1	27	0.954245	0.987186	0.882812	0.870306
2	43	0.987415	0.999332	0.881076	0.865370
3	42	0.986561	0.999235	0.887153	0.864158
4	49	0.984428	0.998939	0.873264	0.854509
5	40	0.985068	0.998902	0.880208	0.855024
6	47	0.984535	0.998832	0.885417	0.877391
7	19	0.937287	0.973232	0.879340	0.867015
8	37	0.986241	0.999177	0.882812	0.858009
9	41	0.987521	0.999298	0.884549	0.855724
10	14	0.911689	0.950151	0.880208	0.864090
11	39	0.986561	0.999259	0.885417	0.859053
12	46	0.987095	0.999234	0.885417	0.871529
13	32	0.974509	0.995912	0.878472	0.863161
14	50	0.987628	0.999417	0.880208	0.845117
15	49	0.986988	0.999286	0.880208	0.853893
16	17	0.952005	0.983124	0.880208	0.871654
17	41	0.984855	0.998878	0.884549	0.859946
18	22	0.927474	0.967336	0.881076	0.868636
19	42	0.987521	0.999216	0.887153	0.846419

positive rates achievable by the model when the decision threshold (i.e. for superconducting versus nonsuperconducting) is varied. Since it is independent of the model decision threshold selected, it is typically used instead of accuracy, which depends on a selected decision threshold. An ROCAUC of 0.5 or less corresponds to model that is classifying by random guessing, whereas an ROCAUC of 1.0 indicates perfect classification. Typically a value of 0.75 or higher indicates good model performance. We summarize all of the recorded experiment outcomes in Table A.2.

In Table A.2, we report both the accuracy and ROCAUC for both the training and validation sets. As expected the accuracy and ROCAUC values are higher for the training data, since these are the data the model was fit to. Since the validation data was not used in training the model, the ROCAUC and accuracy more accurately reflect the performance of the model on unseen data. Note that we do not report any data for the test set, as the test set should be reserved to provide a final estimate of model performance that is not contaminated by selection bias. In this instance, we are only concerned with optimizing the ROCAUC of the validation set; however we will also perform some regression analysis to determine the estimate number of epochs needed to train a model with a given hyperparameter configuration, as a reliable estimate of this number can be used to estimate the computational cost of training future models.

A.2.2 Statistical Significance of Hyperparameters

After obtaining our experimental data, we first performed some basic analysis of the statistical significance of each hyperparameter's contribution to both the observed number of training epochs and the observed validation set ROCAUC. To accomplish this, we performed one-way univarite linear Analysis of Variance (ANOVA) for these observed quantities. In Table A.3, we report the F-statistics and p-values of each hyperparameter.

For the p-values reported in Table A.3, we set the threshold of rough statistical significance to be $p < 0.2$. Per this threshold, we find that the number of CGCNN layers and the number of hidden features have a statistically significant contribution on the number of epochs required for the model to converge. Since the CGCNN

Table A.3: Table of F-statistics obtained from one-way univariate linear ANOVA for the number of training epochs and the validation set ROCAUC. From this test, we discover that the number of ALIGNN layers and hidden features have a statistically significant contribution ($p < 0.2$) to the number of training epochs needed for the model to converge, while the learning rate and number of embedding features have the most statistically significant contribution to the accuracy of the classifier.

Feature	Training Epochs F-statistic	Training Epochs p-value	ROCAUC F-statistic	ROCAUC p-value
Weight Decay	0.085695	0.773071	0.899791	0.355400
Learning Rate	0.202776	0.657866	2.293826	0.147252
ALIGNN Layers	0.064816	0.801926	0.007865	0.930312
CGCNN Layers	3.704885	0.070206	0.769419	0.391954
Hidden Features	2.375701	0.140633	0.401508	0.534281
Embedding Features	0.844992	0.370128	2.112193	0.163342

layers occur quite deep in the model, it makes sense that increasing the number of these layers may have a significant impact on the number of training steps (and hence training epochs) needed for the model to converge. Furthermore, the number of hidden features are the parameter that contributes most to the number of total weights in the model, and thus changes in this parameter greatly affect model complexity and training time. For the ROCAUC, we find that the learning rate and the number of embedding features have the greatest impact on the model performance. Since the number of embedding features corresponds to the number of weights in the first layer of the model, it should come as no surprise that it directly affects the number of features that can be extracted from the crystal lattice and thus model performance. The learning rate itself is also known to play a significant role in model performance, which is reflected in its statistical significance of $p = 0.14$. To summarize, we find that number of hidden features and CGCNN layers are most likely to affect how long the ALIGNN model will take to train; however, the learning rate and number of embedding features are most likely to affect the model performance.

Table A.4: Table of estimated optimal hyperparameters according to the quadratic ROCAUC response surface. The default values and range of experimental values are also given.

Feature	Default Value	Experimental Range	Optimal value
Weight Decay	10^{-5}	$[10^{-7}, 10^{-3}]$	10^{-7}
Learning Rate	10^{-3}	$[10^{-4}, 10^{-2}]$	10^{-2}
ALIGNN Layers	4	$[1, 7]$	7
GCN Layers	4	$[1, 7]$	3
Hidden Features	128	$[64, 256]$	145
Embedding Features	128	$[64, 256]$	170

A.2.3 Optimal Response Surface

Using the experimental data, we also generated a best fit of the ROCAUC response surface, and used this best fit to estimate the hyperparameter configuration that maximizes the classifier ROCAUC (and thus indirectly, the model accuracy). Since we conducted few experiments, we decided that a multivariate quadratic ridge regression fit ($\lambda = 10^{-4}$) was most appropriate for the response surfaces of both the number of training epochs and the ROCAUC. In Figure A.1, we plot the response surfaces of the ROCAUC and estimated optimal parameters for each hyperparameter.

A.2.4 Evaluating the Optimal Model

After we computed the ROCAUC response surface, we found the optimal value of each hyperparameter that maximized the ROCAUC on the response surface within the range of experimental values. This produced optimal the hyperparameters summarized in Table A.4.

In Table A.4, we note that all of the predicted optimal values vary from the default values recommended for the ALIGNN model. We note that the optimal value of the weight decay parameter is 10^{-7} at the lower extreme, and the optimal learning rate is 10^{-2} at the upper extreme, which likely reflects the fact that weight decay

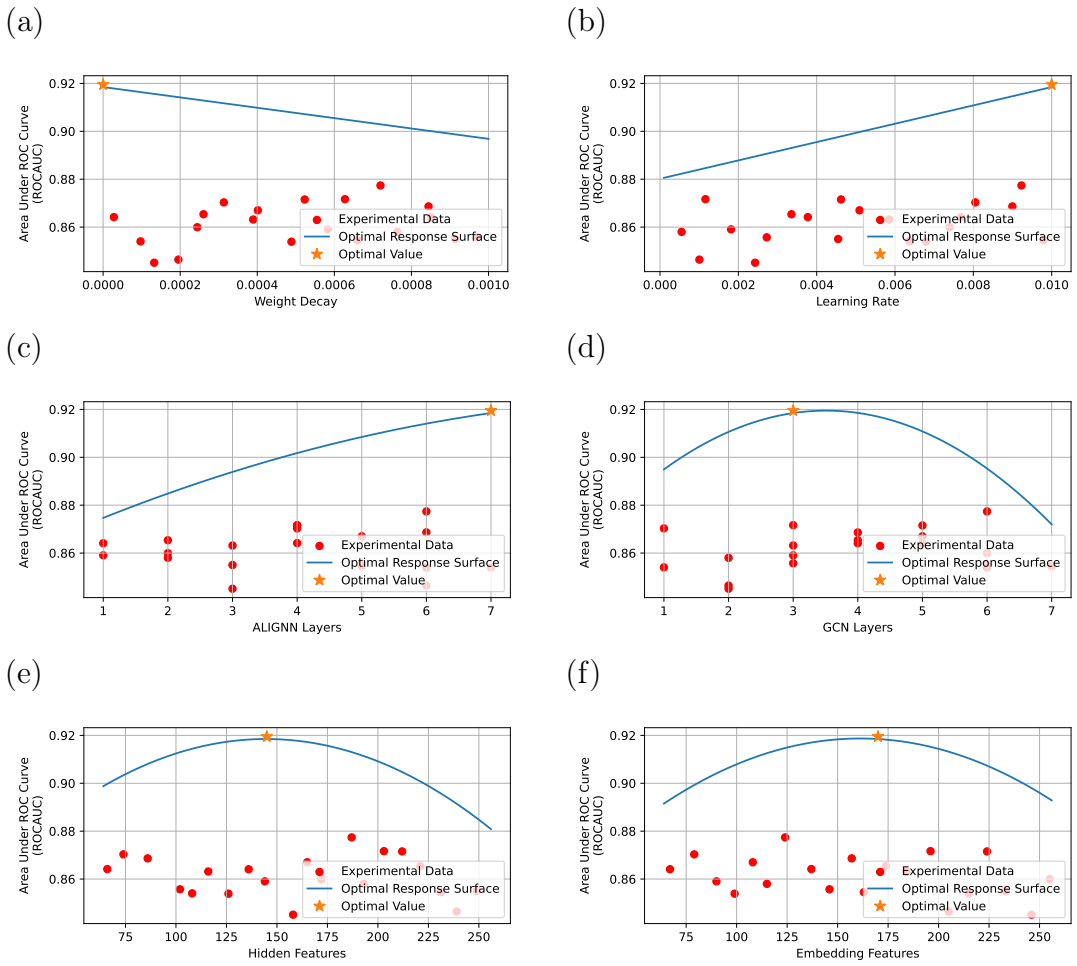

Figure A.1: Plots of the experimental data and estimated response surfaces of each factor in the designed experiment. For the response surfaces, the factors not shown were held constant at the predicted optimal value.

regularization may not be needed when training the ALIGNN model on our dataset, and that the learning rate may need to be increased past the threshold of 10^{-2} to achieve better validation accuracy. We also find that the model performs best with many ALIGNN layers and relatively few CGCNN layers. Finally, we estimate that the optimal number of hidden features and embedding features are 145 and 170 respectively, which is slightly higher than the default value of 128.

Table A.5: Table of predicted and actual training epochs and ROCAUC under optimal hyperparameter configuration. After training the ALIGNN model with the optimal parameters from Table A.4, we obtained a validation ROCAUC of 0.88, which is less than the predicted value of 0.92. The ROCAUC of the model on the reserved test set is also given as a final estimate of model performance. The reserved test set ROCAUC (0.86) being slightly less than that of the validation set (0.88) reflects the effect of selection bias when using the validation ROCAUC as an estimate of model performance.

Predicted/Actual Outcome	Outcome Range	Value
Estimated Optimal Training Epochs	$[1, 50]$	46
Estimated Optimal ROCAUC	$[0.5, 1.0]$	0.92
Actual Optimal Training Epochs	$[1, 50]$	48
Actual Optimal ROCAUC	$[0.5, 1.0]$	0.88
Reserved Test Set ROCAUC	$[0.5, 1.0]$	0.86

After determining the optimal parameter configuration, we found that the response surface indicated an optimal validation set ROCAUC of 0.92 could be achieved in only 46 epochs, as summarized below in Table A.5. After training one more instance of the ALIGNN model on these optimal parameters, we found that the actual validation set ROCAUC was only 0.88. Nonetheless, we found that the number of epochs needed for convergence was 48, which was approximately equal to the predicted value of 46. To obtain a final estimate of the model performance, we evaluated the model on the reserved test set, which was kept independent of the training and validation sets, and obtained an ROCAUC of 0.86, which is still lower than the reported validation ROCAUC of 0.88. This discrepancy in the test set versus validation set ROCAUC accounts for the selection bias introduced in using the validation set to choose the best of multiple competing models.

A.3 Conclusion

In this report, we applied Design of Experiments (DOE) to optimize the hyperparameters of the ALIGNN deep learning model, which we applied to the problem

of classifying materials superconductors as superconducting or non-superconducting based on their raw atomic structure. We used the PyDOE2 software package to generate 20 different computational experiments, each of which used different configurations of six model hyperparameters selected according to the Latin Hypercube Sampling (LHS) scheme. An instance of the ALIGNN model was trained with each of the hyperparameter configurations in order to optimize the overall model performance, which we measured using the receiver operating characteric area under curve (ROCAUC). After generating a quadratic fit of the response surface to the observed values, we estimated the optimal model hyperparameters, which we summarized in Table A.4. Finally, we trained trained the ALIGNN model using these optimal parameters and obtained a validation ROCAUC of approximately 0.88, which was higher than any of the values obtained in the preceding experiments. However, this value did not quite attain the more optimistic ROCAUC of 0.92 predicted by the response surface. This suggests that additional work may need to be conducted to fine-tune the response surface close to the optimal values.

APPENDIX B

Computing Gaussian Orbital Hamiltonians

In this section we discuss how overlap integrals between Gaussian orbitals and pseudopotentials are computed in a way that is continuous and differentiable with respect to the parameters.

B.1 Overlap Integrals of Gaussian Orbitals

First, we remark that even though the majority of the analysis of overlap integrals and equivariant convolutions discussed above was carried out in the basis of spherical harmonics, in the computational implementation, the real basis of spherical harmonics $\bar{Y}_{\ell m}$ (and respective regular solid harmonics $\bar{R}_{\ell m}$) is used. This contributes to greater computational efficiency and ensures all overlap integral quantities are real-valued. In this real basis, each periodic orbital over the unit cell can be written as a sum of gaussians times real solid harmonic functions $\bar{R}_{\ell m}$. For convenience, we will denote this here using the form

$$\bar{u}_{n\ell m}(\mathbf{x}-\mathbf{r}) = \sum_{\mathbf{a}} \sum_{b=1}^{N_b} \bar{c}_{nb}\; g_{n\ell mb}(\mathbf{x}-(\mathbf{r}+\mathbf{a})) \tag{B.1}$$

where $g_{n\ell mb}$ is the product of a non-contracted 3D Gaussian function times a solid harmonic function:

$$g_{n\ell m,b}(\mathbf{x}) = \exp\left(-\alpha_{n,b}\mathbf{x}^2\right) \bar{R}_{\ell m}(\mathbf{x}). \tag{B.2}$$

Each $\bar{u}_{n\ell m}$ orbital consists of a linear combination of N_b of the $g_{n\ell m,b}$ evaluated at each periodic image (associated with a lattice vector $\mathbf{a}$) if the system of interest is periodic. The real solid harmonics (much like the more familiar cubic harmonics, which form

the traditional set of orbitals in quantum chemistry), can be further decomposed as a linear combination of independent powers of Cartesian orbitals, denoted $A_{nijkb}(x)$:

$$g_{n\ell mb}(\mathbf{x}) = \sum_{\substack{i,j,k\geq 0 \\ i+j+k=\ell}} X_{ijk}^{\ell m} A_{nijkb}(\mathbf{x}) \tag{B.3}$$

where $X_{ijk}^{\ell m}$ the coefficients of a non-unitary linear transformation between the basis of Cartesian and solid harmonic orbitals. These Cartesian orbital consist of simple 3D Gaussian times integer powers of each dimension (x^i, y^j, and z^k):

$$A_{nijkb}(\mathbf{x}) = N_{nijkb}\ x^i y^j z^k \exp\left(-\alpha_{nb}\mathbf{x}^2\right). \tag{B.4}$$

The Cartesian orbitals are also normalized by the coefficient N_{nijkb} such that $1 = \int |A(\mathbf{x})|^2\ d\mathbf{x}$. This coefficient is dependent on both α and the integer values of i, j, and k, given by:

$$N_{nijkb} = \left(\frac{2\alpha_{nb}}{\pi}\right)^{3/4} \sqrt{\frac{(8\alpha_{nb})^{i+j+k} i!j!k!}{(2i)!(2j)!(2k)!}} \tag{B.5}$$

This transformation from regular harmonics into the basis of Cartesian orbitals is quite useful, since the integral between two Cartesian orbitals can be evaluated over each dimension independently.

B.1.1 Orbital Overlap Integrals

Now that we have introduced the transformation into the basis of Cartesian orbitals, we will discuss how overlap integrals between neighboring sites $(\mathbf{r}, \mathbf{r}')$ can be evaluated. To simplify our analysis, we will only consider the non-periodic case. Non-periodic overlap integrals can be extended into periodic overlap integrals by simply summing over the non-periodic pairwise overlap integrals between the periodic image of each site.

We start by considering the overlap of two orbitals centered at positions $\mathbf{r}$ and $\mathbf{r}'$ respectively. This integral can be decomposed as a sum over the pairwise overlaps of each uncontracted Gaussian function $g_{n\ell mb}$ times products of the orbital coefficients:

$$\langle \bar{u}_{n,\ell,m}(\mathbf{x}-\mathbf{r})|\bar{u}_{n',\ell',m'}(\mathbf{x}-\mathbf{r}')\rangle = \sum_{b=1}^{N_b}\sum_{b'=1}^{N_b} \bar{c}_{nb}\bar{c}'_{nb}\,\langle g_{n\ell mb}(\mathbf{x}-\mathbf{r})|g_{n'\ell'm'b'}(\mathbf{x}-\mathbf{r}')\rangle \quad \text{(B.6)}$$

Each of the pairwise integrals between uncontracted Gaussians can be further decomposed as a sum of pairwise overlap integrals with respect to Cartesian Gaussian functions A_{nijkb}:

$$\begin{aligned}\langle g_{n\ell mb}(\mathbf{x}-\mathbf{r})|g_{n'\ell'm'b'}(\mathbf{x}-\mathbf{r}')\rangle = \sum_{\substack{i,j,k\geq 0\\ i+j+k=\ell}}\sum_{\substack{i',j',k'\geq 0\\ i'+j'+k'=\ell'}} \Big[X^{\ell m}_{ijk}X^{\ell' m'}_{i'j'k'} \\ \times \langle A_{nijkb}(\mathbf{x}-\mathbf{r})|A_{n'i'j'k'b'}(\mathbf{x}-\mathbf{r}')\rangle \Big]\end{aligned} \quad \text{(B.7)}$$

Exploiting the fact that the Cartesian functions can be written as a product of independent functions along the x, y, and z, axes, the overlap integrals between a pair of such functions can be decomposed as the product of three similar integrals:

$$\begin{aligned}\langle A_{nijkb}(\mathbf{x}-\mathbf{r})|A_{n'i'j'k'b'}(\mathbf{x}-\mathbf{r}')\rangle =& N_{nijkb}N_{n'i'j'k'b'} \prod_{\substack{d=1,2,3\\ (i_d,i'_d=i,j,k)}} \Big[\\ & \int (x-\mathbf{r}_d)^{i_d}(x-\mathbf{r}_d)^{i'_d} e^{-\alpha_{nb}(x-\mathbf{r}_d)^2} e^{-\alpha_{n'b'}(x-\mathbf{r}_d)^2}\, dx\Big]\end{aligned} \quad \text{(B.8)}$$

These three integrals essentially consist of the overlap integral of 1-dimensional primitive Gaussians, which we denote as

$$\langle A_{nijkb}(\mathbf{x}-\mathbf{r})|A_{n'i'j'k'b'}(\mathbf{x}-\mathbf{r}')\rangle = N_{nijkb}N_{n'i'j'k'b'} \prod_{\substack{d=1,2,3\\ (i_d,i'_d=i,j,k)}} \langle a_{ni_db}(s)|a_{ni_db}(s-(\mathbf{r}_d-\mathbf{r}'_d))\rangle\,, \quad \text{(B.9)}$$

where each primitive Gaussian a_{nib} takes the form

$$a_{nib}(x-r) = (x-r)^i e^{-\alpha_{nb}(x-r)^2}. \quad \text{(B.10)}$$

Remarkably, each of these integrals of the form $\langle a_{ni_db}(s)|a_{ni_db}(s-(\mathbf{r}_d-\mathbf{r}'_d))\rangle$ can be evaluated in closed form using a result called the *Gaussian product theorem*, which states that the product of two Gaussian functions (in one or more dimensions) is itself a Gaussian function. Specifically, for the one-dimensional case of two Gaussians centered at r and r', it can be shown that:

$$e^{-\alpha(x-r)} \times e^{-\alpha'(x-r')} = (e^{-\eta(r-r')})e^{-\gamma(x-r'')} \tag{B.11}$$

where the quantities η, γ, and r'' are

$$\eta = \frac{\alpha\alpha'}{\alpha+\alpha'}, \qquad \gamma = \alpha+\alpha', \qquad r'' = \frac{\alpha r + \alpha' r'}{\gamma}. \tag{B.12}$$

. The integral of the product of two Gaussians then reduces to a single Gaussian integral:

$$\int e^{-\alpha(x-r)}e^{-\alpha'(x-r')}\,dx = e^{-\eta(r-r')}\int e^{-\gamma(x-r)^2}\,dx = e^{-\eta(r-r')}\sqrt{\frac{\pi}{\gamma}}. \tag{B.13}$$

We can apply the above trick, in combination with integration by parts to compute the overlap integral of two arbitrary gaussian primitives. (We will not show the details of this derivation, but instread state the result of recursive application of integration by parts to reduce all integrals to standard Gaussians). With some work, it can be shown that the integral of two primitives separated by a distance r has the closed form

$$\langle a_{nib}(s)|a_{n'i'b'}(s-r)\rangle = \frac{i!i'!(\sqrt{\alpha})^i(\sqrt{\alpha'})^{i'}}{2^{i+1'}}\sum_{p=0}^{\lfloor i/2\rfloor}\sum_{p'=1}^{\lfloor i'/2\rfloor}\frac{(\sqrt{\alpha})^{-(i-2p)}(\sqrt{\alpha'})^{-(i'-2p')}}{p!p'!(i-2p)!(i'-2p')!}\mathcal{H}_{i-2p,i'-2p'} \tag{B.14}$$

where $\mathcal{H}_{m,m'}$ is a matrix of Hermite-Gauss polynomials times powers of $\sqrt{\eta}$:

$$\mathcal{H}_{m,m'} = (-1)^m(\sqrt{\eta})^{m+m'}H_{m+m'}(r\sqrt{\eta})e^{-(r\sqrt{\eta})^2} \tag{B.15}$$

Above, $H_n(x)$ denotes the (physicist's) Hermite polynomial of degree n. Although this integral is somewhat complicated to evaluate, it is continuous and entirely differentiable with respect to both the α parameters and the distances between sites r. It is this result that enables the evaluation of the overlap between Gaussian orbitals in closed form.

B.1.2 Overlap Matrices

Previously, we showed that the overlap integrals between two sites could be computed in in a closed, differentiable form with respect to the distance $(\mathbf{r} - \mathbf{r}')$ between sites. However, computing each overlap integral individually using the formulas above would result in a significant amount of redundant work. Instead, it is better to compute the full overlap matrix all at once. The entries of this matrix correspond to the overlaps between all pairs of orbitals at either site. To show how this computation can be performed, it is helpful to use Einstein tensor notation, which shows the tensors and their respective indices that must be contracted without prescribing a specific order in which the index contractions occur.

Computationally, this is very helpful and saves a significant amount of computational work by contracting product of Einstein-like tensors in an optimal order. Indeed, many numerical libraries like PyTorch have support for both statically and dynamically determining the optimal contraction order to tensors, enabling significant computational speedups. We will proceed by denoting the overlap matrix between adjacent Cartesian orbitals using the notation

$$\langle A_{nijkb}(\mathbf{x} - \mathbf{r}) | A_{n'i'j'k'b'}(\mathbf{x} - \mathbf{r}') \rangle \Leftrightarrow \boldsymbol{\mathcal{A}}_{nijkb}^{(nijkb)'} \tag{B.16}$$

Table B.1: Table of Einstein tensor indices used in this appendix section.

Index symbol(s)	Meaning
n	Harmonic orbital principal quantum number
ℓ	Harmonic orbital angular quantum number
m	Harmonic orbital magnetic quantum number
b	Basis index (values from $1, 2, ..., N_b$)
d	Dimension (independent $x, y, z / i, j, k$ components)
$i, j, k / f, g, h$	Cartesian orbital powers ($x^i y^j z^k$ / $x^f y^g z^h$)
p	Polynomial power of a radial quantity (e.g. r^p)
s	Degree of a polynomial value (e.g. $H_s(x)$)
t	Potential basis index

and we will proceed to write down sequence of tensor contractions that compute $\boldsymbol{\mathcal{A}}_{nijkb}^{(nijkb)'}$ and the overlap matrix tensor $\boldsymbol{\mathcal{S}}_{n\ell m}^{(n\ell m)'}$ in the basis of solid harmonic functions. In Table B.1, we document the letters used to denote each axis used in Einstein tensor contractions from this appendix section.

The overlap matrix $\boldsymbol{\mathcal{A}}_{nijkb}^{(nijkb)'}$ in the basis of Cartesian functions can be written as the product of a sequence of contractions of Gaussian primitives corresponding to each dimension. This sequence is given by

$$\boldsymbol{\mathcal{A}}_{nijkb}^{(nijkb)'} = \prod_{\substack{d=1,2,3 \\ (i_d=i,j,k)}} \left(\boldsymbol{\mathcal{N}}_{i_d}^{m} \, \boldsymbol{\alpha}_{nmb} \, \boldsymbol{\mathcal{H}}_{dnmb}^{(nmb)'} \, \boldsymbol{\alpha'}^{(nmb)'} \, \boldsymbol{\mathcal{N}}_{m'}^{i'_d} \right) \tag{B.17}$$

where $\boldsymbol{\mathcal{N}}_{i}^{m}$ is a constant tensor of integral re-normalization constants and $\boldsymbol{\mathcal{H}}_{nijkb}^{(nijkb)'}$ is a Hermite-Gauss tensor, whose values depend on both the α_{nb} values and the distance $(\mathbf{r}' - \mathbf{r})$ between sites. The entries of $\boldsymbol{\mathcal{N}}_{i}^{m}$ are constants, which combine the contributions of N_{nijkb} corresponding to each dimension (i, j, k) and the integral coefficients obtained from the formula (B.14). The tensors $\boldsymbol{\alpha}_{nmb}$, $\boldsymbol{\alpha'}_{nmb}$ consist of increasing half-integer powers of the width parameters α_{nb}:

$$(\boldsymbol{\alpha}_{nmb})_{nmb} = (\alpha_{nb})^{m/2} \tag{B.18}$$

The Hermite-Gauss tensor $\boldsymbol{\mathcal{H}}_{nijkb}^{(nijkb)'}$ can also be expanded as

$$\boldsymbol{\mathcal{H}}_{dnmb}^{(nmb)'} = \boldsymbol{\sigma}_{m}^{m's}\ \boldsymbol{\eta}_{dsnb}^{(nb)'}\ \boldsymbol{\mathcal{P}}_{p}^{s}\ \boldsymbol{\eta}_{dpnb}^{(nb)'}\ \boldsymbol{r}_{dp} \tag{B.19}$$

Above, the tensors $\boldsymbol{\eta}_{dpnb}^{(nb)}$ and $\boldsymbol{r}_{dp}$ consist of half-integer and integer powers of η_{nb} and the displacement vector components r_d respectively, such that

$$(\boldsymbol{\eta}_{dpnb}^{(nb)})_{dp,nb,n'b'} = (\eta_{nb,n'b'})^{p/2} = \left(\frac{\alpha_{nb}\alpha'_{n'b'}}{\alpha_{nb}+\alpha'_{n'b'}}\right)^{p/2} \tag{B.20}$$

and

$$(\boldsymbol{r}_{dp})_{dp} = (r_d)^p. \tag{B.21}$$

Finally, the tensors $\boldsymbol{\sigma}_{m}^{m's}$ and $\mathcal{P}_{p}^{s}$ are constant summation and Hermite polynomial tensors, such that

$$(\boldsymbol{\sigma}_{ms}^{m'})_{mm's} = \begin{cases} (-1)^m, & m_1 + m_2 = s \\ \\ 0, & \text{otherwise} \end{cases} \tag{B.22}$$

and

$$(\boldsymbol{\mathcal{P}}_{p}^{s})_{ps} = h_{ps} \tag{B.23}$$

where h_{ps} is the pth coefficient of the degree s Hermite polynomial (i.e., $H_s(x) = \sum_{p=0}^{s}(h_{sp})x^p$).

Now that we have computed the overlap tensor $\boldsymbol{\mathcal{A}}_{nijkb}^{(nijkb)'}$, the tensor can be transformed into the basis of regular solid harmonics using the transformation

$$\boldsymbol{\mathcal{S}}_{n\ell m}^{(n\ell m)'} = \boldsymbol{c}_{n}^{b}\boldsymbol{\mathcal{X}}_{\ell m}^{ijk}\boldsymbol{\mathcal{A}}_{nijkb}^{(nijkb)'}\boldsymbol{\mathcal{X}}_{(ijk)'}^{(\ell m)'}\boldsymbol{c}_{b'}^{n'} \tag{B.24}$$

where $(\boldsymbol{\mathcal{X}}_{lm}^{ijk})_{lmijk} = X_{lm}^{ijk}$ are simply the solid harmonic basis transformation coefficients. The resulting tensor $\boldsymbol{\mathcal{S}}_{n\ell m}^{(n\ell m)'}$ corresponds to the full overlap matrix between

pairs of real solid harmonic orbitals:

$$\langle \bar{u}_{n,\ell,m}(\mathbf{x}-\mathbf{r})|\bar{u}_{n',\ell',m'}(\mathbf{x}-\mathbf{r}')\rangle \Leftrightarrow \boldsymbol{\mathcal{S}}_{n\ell m}^{(n\ell m)'}. \tag{B.25}$$

B.1.3 Hamiltonian Matrices

We have discussed how to compute the matrix of overlap integrals between orbitals; however, this is insufficient for computing the full Hamiltonian matrix $h(\mathbf{k})$ corresponding to a (possibly periodic) system of interest. To do this, we must consider how the (a) potential operator, (b) kinetic energy operator, and (c) the momentum operator can be evaluated between pairs of orbitals. The potential operator is evaluated through integrals of the form

$$\langle \bar{u}_{n,\ell,m}(\mathbf{x}-\mathbf{r})|\, V(\mathbf{x}-\mathbf{r})\, |\bar{u}_{n',\ell',m'}(\mathbf{x}-\mathbf{r}')\rangle\,, \tag{B.26}$$

while the kinetic energy operator is evaluated via integrals of the form

$$-\frac{1}{2}\,\langle \bar{u}_{n,\ell,m}(\mathbf{x}-\mathbf{r})|\, \nabla^2\, |\bar{u}_{n',\ell',m'}(\mathbf{x}-\mathbf{r}')\rangle\,. \tag{B.27}$$

Finally, the momentum operator requires integrals of the form

$$-i\,\langle \bar{u}_{n,\ell,m}(\mathbf{x}-\mathbf{r})|\, \nabla\, |\bar{u}_{n',\ell',m'}(\mathbf{x}-\mathbf{r}')\rangle\,. \tag{B.28}$$

B.1.3.1 Potential energy terms. The potential energy integals require integrating the product of the wavefunctions with Gaussians expanded in the form

$$V(\mathbf{x}-\mathbf{r}) = \sum_{t=1}^{n_t} v_t e^{-\gamma_t(\mathbf{x}-\mathbf{r})^2} = \sum_{t=1}^{n_t} v_t \prod_{d=1,2,3} V_t(x-\mathbf{r}_d) \tag{B.29}$$

where $V_t(x) = e^{-\gamma_t x^2}$. The expansion of $V(\mathbf{x})$ as a product of 1D potential Gaussians V_t facilitates the evaluation of the integrals of primitive Gaussians times V_t:

$$\langle a_{nib}(s)|\, V_t(x)\, |a_{n'i'b'}(s-r)\rangle = \langle \nu_{nibt}(s)|a_{n'i'b'}(s-r)\rangle \tag{B.30}$$

However, we note that this product essentially reduces to the product of V_t and the first wave function results in a modified wave function

$$\nu_{nibt}(x-r) = (x-r)^i e^{-(\alpha_{nb}\gamma_k)(x-r)^2}. \tag{B.31}$$

This means that the potential overlap integrals can be evaluated in matrix form using the tensor product sequence

$$\boldsymbol{\mathcal{V}}_{n\ell m}^{(n\ell m)'} = \boldsymbol{v}^t \boldsymbol{c}_n^b \boldsymbol{\mathcal{X}}_{\ell m}^{ijk} \boldsymbol{\mathcal{A}}[V_t]_{nijkbt}^{(nijkb)'} \boldsymbol{\mathcal{X}}_{(ijk)'}^{(\ell m)'} \boldsymbol{c'}_{b'}^{n'} \tag{B.32}$$

where $\boldsymbol{v}_t$ is the tensor of potential coefficients and $\boldsymbol{\mathcal{A}}[V_t]_{nijkbt}^{(nijkb)'}$ is the Cartesian potential overlap tensor.

B.1.3.2 Kinetic energy terms. The kinetic energy integrals can also be evaluated by considering how the Laplace operator ∇^2 acts on the basis of primitive Gaussians a_{nib} in each dimension. We find that

$$\begin{aligned} -\frac{1}{2}\left\langle a_{nib}(s)\right| \nabla^2 \left|a_{n'i'b'}(s-r)\right\rangle = \; & \alpha_{nb}(2i+1)\left\langle a_{nib}(s)|a_{n'i'b'}(s-r)\right\rangle \\ & - 2\alpha_{nb}^2 \left\langle a_{n(i+2)b}(s)\middle|a_{n'i'b'}(s-r)\right\rangle \\ & - \frac{1}{2} i(i-2)\left\langle a_{n(i-2)b}(s)\middle|a_{n'i'b'}(s-r)\right\rangle, \end{aligned} \tag{B.33}$$

which means that the kinetic integrals can be expressed in terms of the regular Cartesian overlap tensor. Specifically, we write:

$$\boldsymbol{\mathcal{T}}_{n\ell m}^{(n\ell m)'} = \boldsymbol{c}_n^b \; \boldsymbol{\mathcal{X}}_{\ell m}^{ijk} \; \boldsymbol{\Delta}_{nijkb}^{nfghb} \; \boldsymbol{\mathcal{A}}_{nfghb}^{(nijkb)'} \; \boldsymbol{\mathcal{X}}_{(ijk)'}^{(\ell m)'} \; \boldsymbol{c'}_{b'}^{n'} \tag{B.34}$$

where $\boldsymbol{\Delta}^{nfghb}_{nijkb}$ is the Laplace tensor, given by:

$$(\boldsymbol{\Delta}^{nfghb}_{nijkb})_{n,ijk,fghmb} = \sum_{\substack{d=1,2,3 \\ (i_d=i,j,k) \\ (f_d=f,g,h)}} \begin{cases} \alpha_{nb}(2i_d+1) & i_d = f_d \\ -2\alpha_{nb}^2 & i_d+2 = f_d \\ -i_d(i_d-2)/2 & i_d-2 = f_d \\ 0 & \text{otherwise} \end{cases} \tag{B.35}$$

B.1.3.3 Momentum terms. In a similar manner, we can expand the momentun operator $\mathbf{p} = -i\nabla$ in terms and see how it acts on the primitive Gaussian functions in each dimension. We observe that

$$\begin{aligned} \langle a_{nib}(s)| \frac{\partial}{\partial s} |a_{n'i'b'}(s-r)\rangle = & -2\alpha_{nb} \left\langle a_{n(i+1)b}(s) \middle| a_{n'i'b'}(s-r) \right\rangle \\ & - i \left\langle a_{n(i-1)b}(s) \middle| a_{n'i'b'}(s-r) \right\rangle, \end{aligned} \tag{B.36}$$

so we can expand the momentum tensor via a sequence of the form

$$\boldsymbol{p}^{(n\ell m)'}_{n\ell md} = -i \left[\boldsymbol{c}^{b}_{n} \, \boldsymbol{\mathcal{X}}^{ijk}_{\ell m} \, \boldsymbol{\nabla}^{nfghb}_{nijkbd} \, \boldsymbol{\mathcal{A}}^{(nijkb)'}_{nfghb} \, \boldsymbol{\mathcal{X}}^{(\ell m)'}_{(ijk)'} \, \boldsymbol{c'}^{n'}_{b'} \right] \tag{B.37}$$

where $\boldsymbol{\nabla}^{nfghb}_{nijkbd}$ is the gradient tensor (indexed by dimension d), given by:

$$(\boldsymbol{\nabla}^{nfghb}_{nijkbd})_{n,ijk,fgh,bd} = \begin{cases} 2\alpha_{nb} & i_d+1 = f_d \\ -i_d & i_d-1 = f_d \\ 0 & \text{otherwise} \end{cases}, \tag{B.38}$$

where we use $i_d = i, j, k$ and $f_d = f, g, h$ for $d = 1, 2, 3$ respectively.

APPENDIX C

PyQuinn Usage and Documentation

In this section, we present the content of the Installation and Quickstart guide found in the official PyQuinn documentation. We reproduce this content here to give basic and informative examples showing how the PyQuinn API is used. For more information, see the official repository: `https://github.com/cburdine/pyquinn` [].

C.1 Introduction

PyQuinn is a Pytorch-based Python Library for the development of Quantum-Informed Neural Networks (QuINNs).

What is a Quantum-Informed neural network?

QuINNs are a type of Physics-Informed Neural Network (PINN), which learn mean-field solutions to the electronic time-independent Schrodinger equation for molecules and solid state materials. The PyQuinn framework provides a simple and extensible framework for rapidly building and training QuINN models, as well as more traditional graph neural networks. QuINNs can be trained to predict atom-wise properties, force fields, and even macroscopic material properties from raw atomic structures.

C.2 Quickstart Guide

C.2.1 Installation

To use PyQuinn, you will need to install the necessary dependencies. Since this project is still under development, you can see the most up-to-date installation instructions in the `README.md` file.

C.2.1.1 Installing from source. To install PyQuinn from source, you can use pip:

```
pip install -e .
```

C.2.2 Optional Dependencies

Installing PyQuinn ensures that all core dependencies are installed, including the default Pytorch backend. However, there may be some additional optional dependencies you may want to install:

C.2.2.1 Visualization packages. To visualize atomic structures and plot predicted electronic bands, charge densities, etc., you will need to install either `matplotlib` (default) or `plotly`.

```
pip install matplotlib plotly
```

We recommend using `plotly` for interactive visualizations inside Python notebooks.

C.2.2.2 Additional backend packages. Currently, only the default Pytorch (`torch`) backend is supported. However, in the future we aim to add support for other popular frameworks, such as JAX and Tensorflow.

C.2.3 Examples of Basic Usage

- **Building and compiling an `AtomGraph`**

```
from pyquinn.types import AtomGraph

# create a 2D periodic graph of carbon atoms
graph = AtomGraph(
    atoms=['C', 'C'],
    positions=[
        [0.0,   0.0,   0.0],
        [1.23,  0.71,  0.0]],
    unit_cell=[
        [2.46,  0.0,  0.0],
        [-1.23, 2.13, 0.0],
        [0.0,   0.0,  0.0]],
```

```
    periodic=(True, True, False))

# compile graph for PyTorch with a C cutoff radius of 2.0 Angstroms
torch_graph = graph.compile(backend='torch', cutoffs={'C' : 2.0})

# access Pytorch tensors associated with the compiled graph
print(torch_graph.atomic_numbers)
print(torch_graph.positions)
print(torch_graph.neighbors)
```

- **Retrieving and Visualizing Structures from Databases**

```
from pyquinn.io.databases import COD
import plotly

# connect to the Crystallography Open Database (COD)
with COD() as cod:

    # query the COD for structures
    ids = cod.get_atomgraph_ids_with_elements(['Si', 'O'])
    print('# of results:', len(ids))

    # retrieve an example atomgraph
    graph = cod.get_atomgraph_by_id(ids[0])

# visualize structure
fig = graph.plot()
```

- **Training a Basic Atomwise Model:**

```
from pyquinn.types import AtomGraph, AtomGraphDataset
from pyquinn.torch.models import TorchAtomwiseModel

import pyquinn.torch.nn as pqnn

# H2 bond lengths (Angstroms) and energies (eV)
bond_lengths =  [ 0.4, 0.7, 1.0, 1.5, 2.0, 3.0 ]
energies = [ 20.0, -4.4, -3.2, -1.2, -0.3, 0.0 ]

# build the dataset
dataset = AtomGraphDataset()
for d, E in zip(bond_lengths, energies):
    dataset.append(AtomGraph(
        atoms=['H', 'H'],
        positions=[[0,0,0],[0,0,d]],
        graph_outputs=[E],
    ))

# compile the dataset with an interaction cutoff of 4.0 Angstroms
```

```
torch_dataset = dataset.compile(backend='torch', cutoffs={'H': 4.0})

# make an Atomwise model to predict total energy:
model = TorchAtomwiseModel(
    atomwise_pool='sum',
    out_features=1,
)

# fit model to the dataset with a learning rate of 0.1
history = model.fit(torch_dataset, lr=0.1, verbosity=0)

# Print the final training loss:
print('Final training loss:', history['train_loss'][-1])
```

C.3 About PyQuinn

PyQuinn is a framework for the design, training, and deployment of neural networks for predicting material and chemical properties from atomic structure, with a heavy emphasis on efficiency, modularity, and physical interpretability of models.

PyQuinn also provides many useful tools to aid in the generation and curation of high-quality training datasets. PyQuinn is designed primarily to support Quantum-Informed Neural Networks (QuINNs), which are a type of Physics-Informed Neural Network (PINN), that combines aspects of first-principles quantum mechanical calculations with classical machine learning models, all through the use of end-to-end differentiable modules that allow for the training of neural networks and the optimization of materials at the atomic scale to have targeted properties.

PyQuinn provides a unified, backend-agnostic API that supports data structure primitives for representing materials, molecules, and other chemical systems. These data structure can be compiled to target various backends, allowing for rapid prototyping and development of models using popular frameworks like Pytorch. Result data types obtained from each backend can then be extracted to convert them back to the PyQuinn default type for easier manipulation.

At the moment, only the Pytorch backend is supported, but future support for more frameworks is planned.

C.3.1 The AtomGraph Object

At the core of PyQuinn is the `AtomGraph` class, which represents a 3D atomic structure of atoms with support for periodicity. By convention, all spatial positions must be given in Angstroms (1Å$= 10^{-10}$m). An `AtomGraph` instance can be created and visualized as follows:

```
from pyquinn.types import AtomGraph

# build a water molecule
graph = AtomGraph(
    atoms=['O', 'H', 'H'],
    positions=[
        [0.0, 0.0, 0.119262],
        [0.0, 0.763239, -0.477047],
        [0.0, -0.763239, -0.477047]]
    )

# visualize water molecule
graph.plot()
```

See Also: There are more instance variables in the AtomGraph which can be used for certain systems, such as periodic crystals or to incorporate atomwise or graphwise features to the graph. Visit the API reference page for `AtomGraph` to learn more.

C.3.2 Backend Compilation

`AtomGraph` instances are straightforward to manipulate, since each instance variable is stored as a NumPy Array (`ndarray`). One can modify existing atoms, add new atoms, or obtain subgraphs using NumPy-style indexing. Since they are intended to be efficiently modified, `AtomGraph` do not keep track of "edges" between neighboring atoms. These edges are only computed once the `AtomGraph` is compiled for a target backend using the `compile()` function. When compiling a graph, the user

must supply appropriate cutoff radii for each atom in the graph such that if distance between two atoms is less than the cutoff radii, then an edge is added between the two atoms. To compile the graph we created above for the PyTorch backend with specific cutoffs, we call the `compile()` function as follows:

```
# compile graph
torch_graph = graph.compile(
    backend='torch',
    cutoffs={'H' : 1.0, 'O': 2.0 }
)

# access computed neighbors:
print(torch_graph.neighbors)
```

The compiled graph has almost all of the same instance variables as the regular AtomGraph, except that these variables are now array or tensor data types created on the corresponding backend. Additionally, a neighbors variable is added, which contains the indices of all neighbors of a given atom (the index "-1" is designated as a padding value).

The `extract()` function recovers an `AtomGraph` from a `BackendAtomGraph`. Note, however, that this removes any backend-specific features (such as neighbor lists, or computed gradients):

```
recovered_graph = torch_graph.extract()
print(recovered_graph)
```

See Also: Visit the API reference page for `BackendAtomGraph` to learn about the functions and instance variables supported by all backends.

C.3.3 Other PyQuinn Types

In addition to `AtomGraph`, there are other types supported for compilation, such as:

- `BandStructure`: represents an electronic band structure object, with support for projections onto orbital bases.

- `AtomGraphDataset`: stores an entire dataset of `AtomGraph`s, supporting caching and loading from disk.

- `BandStructureDataset`: Stores a dataset of `AtomGraph`s and associated `BandStructure`s, supporting caching and loading from disk.

These primitives all support efficiently saveable and loadable file formats, making the generation and curation of large datasets simple.

C.3.4 Data I/O

PyQuinn also supports loading or exporting AtomGraphs and other primitive types to objects used in other popular atomic simulation frameworks, such as the Atoms object used in ASE, (the Atomic Simulation Environment). AtomGraphs can also be loaded and and exported to a variety of popular file formats (such as CIF format):

```
from ase.build import graphene
from pyquinn.types import AtomGraph

# create a graphene ase.Atoms object
atoms = graphene()

# build an AtomGraph object from an ase.Atoms object
graph = AtomGraph.from_ase(atoms)

# export an AtomGraph (stored in 'graph') to CIF format
graph.to_file('graphene.cif')

# load the exported file
loaded_graph = AtomGraph.from_file('graphene.cif')
```

Additionally, PyQuinn provides useful database connectors to popular open source atomic structure databases, such as the Crystallography Open Database, or the Materials Project. These database connnectors are found in the `pyquinn.io.databases submodule`. For example, the Materials Project database connector has support for loading both atomic structures and their associated electronic band structures, if available:

```
from pyquinn.io.databases import MaterialsProject

# retrieve an AtomGraph and BandStructure object for mp-66 (diamond)
with MaterialsProject(mp_api_key='<your api key here>') as mp:
    graph = mp.get_atomgraph_by_id('mp-66')
    bands = mp.get_bandstructure_by_id('mp-66')


# visualize BandStructure object:
bands.plot_band_path(atoms=graph)
```

APPENDIX D

Appendix of "Trotterless Methods for the Simulation of Open Quantum Systems"

This appendix published within: Burdine, Colin, and Enrique P. Blair. "Trotterless Simulation of Open Quantum Systems for NISQ Quantum Devices." *Advanced Quantum Technologies*, Oct. 2024, p. 2400240., doi: 10.1002/qute.202400240.

D.1 Parameters for Diagonal Operator Quantum Circuits

In this appendix section we provide additional details regarding the parameter mappings for the quantum circuit implementations of $W(t)$ (Figure 7.9) and $\Lambda(t)$ (Figure 7.10). We will also derive equations (7.73)-(7.75) and show how the transformation Q_N in (7.74) can be computed efficiently.

D.1.1 Parameter mappings for the diagonal unitary $W(t)$:

First, we consider the representation of an arbitrary N-qubit diagonal unitary $W(t) = \sum_{n=0}^{2^N-1} e^{-it\omega_n} |e_n\rangle \langle e_n|$ using the circuit in Figure 7.9 with parameter vector $\vec{\theta} = (\theta_0\ \theta_1\ ...\ \theta_{2^N-1})$. Adding up the phases accumulated by each diagonal state $|e_n\rangle$ from the circuit $W(t)$, we see that the parameters $\vec{\theta}(t)$ must be chosen such that:

$$\exp\left(i\theta_0(t) + i \sum_{B_s \subseteq B_n} \theta_s(t)\right) |e_n\rangle = e^{it\omega_n} |e_n\rangle \tag{D.1}$$

where $B_n = \{m_0, m_1, ...\}$ denotes the set of non-negative indices m of the "1" bits in the binary representation of n (i.e. B_n uniquely satisfies $\sum_{m \in B_n} 2^m = n$). Since θ_0 corresponds to the global phase induced by $W(t)$, this parameter can be ignored in

most situations, as it has no effect on measurement outcomes. Equating the left hand exponential to the right hand exponential in (D.1), we obtain the linear system:

$$\left\{ \theta_0(t) + \sum_{B_s \subseteq B_n} \theta_s(t) = t\omega_n \quad (n = 0, 1, ..., 2^N - 1) \right. \tag{D.2}$$

This system can be written in the matrix form:

$$S_N \vec{\theta}(t) = t\vec{\omega} \tag{D.3}$$

where the matrix S_N factors as an order N tensor power:

$$S_N = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\otimes N} \tag{D.4}$$

Solving for $\vec{\theta}(t)$, we get:

$$\vec{\theta}(t) = S_N^{-1} t\vec{\omega} \tag{D.5}$$

where

$$S_n^{-1} = \left[\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-1} \right]^{\otimes N} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}^{\otimes N} = Q_N \tag{D.6}$$

from which we immediately obtain (7.73) and (7.74).

D.1.2 Parameter mappings for the diagonal contraction operator $\Lambda(t)$:

Next, we will consider the representation of the circuit $U_\Lambda(t)$ corresponding to an arbitrary diagonal contraction operator $\Lambda(t) = \sum_{n=0}^{2^N-1} \vec{v}_n(t) |e_n\rangle \langle e_n|$, where $\|\vec{v}(t)\| \leq 1$ for all $t \geq 0$. To realize the Sz.-Nagy dilation of $\Lambda(t)$, we proposed the circuit in Figure 7.10, which consists of a sequence of controlled $\hat{R}_y$ gates applied to a single ancilla qubit prepared in the $|1\rangle_{\text{anc}}$ state. The amplitude of a filtered basis state $|e_n\rangle \otimes |0\rangle_{\text{anc}}$ after applying this realization of U_Λ with parameter vector $\vec{\beta}$ is

given by:

$$\begin{aligned}
&(\langle e_n| \otimes \langle 0|_{\text{anc}}) U_\Lambda(t) (|e_n\rangle \otimes |0\rangle_{\text{anc}}) \\
&\qquad = \langle 0|_{\text{anc}} \, e^{\frac{-i}{2}\left(\beta_0 + \sum_{B_s \subseteq B_n} \beta_s\right)\hat{Y}} \, |1\rangle_{\text{anc}} \\
&\qquad = -\sin\left[\frac{1}{2}\left(\beta_0 + \sum_{B_s \subseteq B_n} \beta_s\right)\right]
\end{aligned} \tag{D.7}$$

In order to represent a contraction operator with diagonal elements $\vec{v}(t)$, the parameters $\vec{\beta}(t)$ must then be selected such that they satisfy the linear system:

$$\begin{cases} \beta_0(t) + \sum_{B_s \subseteq B_n} \beta_s(t) = -2\arcsin(v_n(t)) \\ \\ (n = 0, 1, ..., 2^N - 1) \end{cases} \tag{D.8}$$

Following the same procedure as with $W(t)$, the solution to this system can be written in matrix form:

$$\vec{\beta}(t) = -2S_N^{-1} \arcsin(\vec{v}(t)) = -2Q_N \arcsin(\vec{v}(t)) \tag{D.9}$$

This agrees with the parameter mapping given in (7.75).

D.1.3 Computing the parameter transform Q_N efficiently

Above, we claimed that the product of Q_N and an arbitrary vector could be computed efficiently on a classical computer in $O(d\log(d))$ time, where $d = 2^N$ for an N-qubit Hilbert space. This is achieved by a modified form of the fast Walsh-Hadamard transform, as described in [204]. Computing the product of Q_N and a vector $\vec{u} = \vec{u}^{(0)}$ can be reduced to an iterative N-step procedure, where the N-th step intermediate result $\vec{u}^{(N)}$ contains the result of the product

$$\vec{u}^{(N)} = Q_N \vec{u}^{(0)}. \tag{D.10}$$

Since Q_N can be factored as an order N tensor power, we can apply the matrix Q_N by sequentially applying the component of Q_N corresponding to each qubit in

the Hilbert space. This gives us the recurrent formula

$$\vec{u}_n^{(t+1)} = \left(I^{\otimes(N-(t+1))} \otimes \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \otimes I^{\otimes t} \right) \vec{u}^{(t)}. \tag{D.11}$$

It follows that each entry $\vec{u}_n^{(t)}$ in $\vec{u}^{(t)}$ (for $n = 0, 1, ..., 2^N$) can be updated according to the procedure

$$\vec{u}_n^{(t+1)} = \begin{cases} \vec{u}_n^{(t)}, & t \notin B_n \\ \vec{u}_n^{(t)} - \vec{u}_{n-2^t}^{(t)}, & t \in B_n \end{cases}. \tag{D.12}$$

In total, applying (D.12) for N steps requires $N \times 2^{N-1}$ subtraction operations, which for a Hilbert space of dimension $d = 2^N$ corresponds to a time complexity of $O(d \log(d))$.

D.2 Bounds on Kraus Series Error

In this appendix section, we derive equations (7.79)-(7.84) by using properties of the Hilbert-Schmidt norm to bound the Kraus series error.

D.2.1 Error of the Kraus Series

For a complete orthonormal basis $|e_i\rangle$ in some Hilbert space, the Hilbert-Schmidt norm is given by:

$$\|A\|_{\text{HS}} = \sqrt{\text{Tr}(A^\dagger A)} = \sqrt{\sum_{i,j} |\langle e_i| A |e_j\rangle|^2} \tag{D.13}$$

Now, let $E_M = \|\vec{\rho}(t) - \vec{\rho}_M(t)\|$ be the error of a Kraus series evaluated up to order $m = M$. Expanding E_M with the Hilbert Schmidt norm and using (7.37) and

(7.76), we bound E_M as follows:

$$\begin{aligned}
E_M &= \|\vec{\rho}(t) - \vec{\rho}_M(t)\| \\
&\leq \left\| e^{t\mathcal{H}} h(t) e^{f(t)\mathcal{L}} \vec{\rho}(0) - \sum_{m=0}^{M} e^{t\mathcal{H}} h(t) \frac{(f(t)\mathcal{L})^m}{m!} \vec{\rho}(0) \right\| \\
&\leq \left\| e^{t\mathcal{H}} h(t) \left(e^{f(t)\mathcal{L}} - \sum_{m=0}^{M} \frac{(f(t)\mathcal{L})^m}{m!} \right) \right\|_{\text{HS}} \|\vec{\rho}(0)\| \\
&\leq \left\| e^{t\mathcal{H}} \right\| h(t) \left\| \sum_{m=(M+1)}^{\infty} \frac{(f(t)\mathcal{L})^m}{m!} \right\|_{\text{HS}} \\
&\leq \left\| e^{t\mathcal{H}} \right\| h(t) \sum_{m=(M+1)}^{\infty} \frac{(f(t)\left\|\mathcal{L}\right\|_{\text{HS}})^m}{m!} \\
&\leq \left\| e^{t\mathcal{H}} \right\| h(t) \left[\sum_{m=0}^{\infty} \left(\frac{f(t)\left\|\mathcal{L}\right\|_{\text{HS}}}{M+1} \right)^m \right] \frac{(f(t)\left\|\mathcal{L}\right\|_{\text{HS}})^{M+1}}{(M+1)!}. \qquad \text{(D.14)}
\end{aligned}$$

For $M > f(t)\left\|\mathcal{L}\right\|_{\text{HS}}$, we expand the geometric series in (D.14) and re-arrange terms to obtain

$$\frac{E_M}{\|e^{t\mathcal{H}}\| h(t)} \leq \left(\frac{1}{1 - \frac{f(t)\|\mathcal{L}\|_{\text{HS}}}{(M+1)}} \right) \frac{(f(t)\left\|\mathcal{L}\right\|_{\text{HS}})^{M+1}}{(M+1)!}, \qquad \text{(D.15)}$$

from which (7.79) and (7.80) immediately follow. From Stirling's inequality, we have

$$2\left(\frac{M}{e}\right)^{M+1} \leq \sqrt{2\pi(M+1)} \left(\frac{(M+1)}{e}\right)^{(M+1)} \leq (M+1)!, \qquad \text{(D.16)}$$

which means that if M is chosen such that (7.83) is satisfied, then we obtain

$$\begin{aligned}
\frac{E_M}{\|e^{t\mathcal{H}}\| h(t)} &\leq \left(\frac{1}{1 - (1/2)} \right) \frac{(f(t)\left\|\mathcal{L}\right\|_{\text{HS}})^{M+1}}{2(M/e)^{M+1}} \\
&\leq \frac{(f(t)\left\|\mathcal{L}\right\|_{\text{HS}})^{M+1}}{(f(t)\left\|\mathcal{L}\right\|_{\text{HS}})^{(1+\delta)(M+1)}} \\
&\leq (f(t)\left\|\mathcal{L}\right\|_{\text{HS}})^{-\delta(M+1)}. \qquad \text{(D.17)}
\end{aligned}$$

After re-arranging (D.17), we obtain the following bound of E_M for a desired error tolerance ε:

$$E_M \leq \frac{h(t)\left\|e^{t\mathcal{H}}\right\|}{(f(t)\left\|\mathcal{L}\right\|_{\text{HS}})^{\delta(M+1)}} \leq \varepsilon. \qquad \text{(D.18)}$$

Solving for M on the right hand side in terms of ε and the remaining factors, we obtain

$$\frac{\log\big(h(t)\left\|e^{t\mathcal{H}}\right\|_{\text{HS}}/\varepsilon\big)}{\delta\log(f(t)\left\|\mathcal{L}\right\|_{\text{HS}})} \leq M+1 \tag{D.19}$$

It immediately follows that if the second inequality (7.84) is satisfied, then the error bound (D.18) holds.

D.3 Results on Pauli Channels

In this section, we give additional discussion regarding the continuous-time multi-qubit Pauli channel.

D.3.1 Commutators of Pauli strings

Here, we prove that (7.88) holds by deriving the commutator formula for arbitrary Pauli strings $\Pi_n, \Pi_{n'}$. Writing (7.88) in superoperator form, we obtain

$$\Pi_n^{\overline{\otimes}2}\Pi_{n'}^{\overline{\otimes}2}\vec{\rho} = \Pi_{n'}^{\overline{\otimes}2}\Pi_n^{\overline{\otimes}2}\vec{\rho} \tag{D.20}$$

for an arbitrary vector $\vec{\rho}$. Thus, proving that (7.88) holds is equivalent to proving that $[\Pi_n^{\overline{\otimes}2}, \Pi_{n'}^{\overline{\otimes}2}] = 0$. First, we consider the commutator and anti-commutator of two arbitrary Pauli strings $\Pi_n, \Pi_{n'}$, both of which can be expanded as the Kronecker product of commutators and anti-commutators of the Pauli operators that comprise them. For the commutator, we observe that

$$\begin{aligned}[\Pi_n, \Pi_{n'}] &= [\Pi_n^{(1)}, \Pi_{n'}^{(1)}] \\ &= \frac{1}{2}[\sigma_{n1}, \sigma_{n'1}] \otimes \{\Pi_n^{(2)}, \Pi_{n'}^{(2)}\} \quad , \\ &\quad + \frac{1}{2}\{\sigma_{n1}, \sigma_{n'1}\} \otimes [\Pi_n^{(2)}, \Pi_{n'}^{(2)}]\end{aligned} \tag{D.21}$$

where we use $\Pi_n^{(j)} = \sigma_{nj} \otimes \sigma_{n(j+1)} \otimes ... \otimes \sigma_{nN}$ to denote the Pauli sub-string of Π_n starting at the jth operator. Likewise for the anti-commutator, we have

$$\begin{aligned} \{\Pi_n, \Pi_{n'}\} &= \{\Pi_n^{(1)}, \Pi_{n'}^{(1)}\} \\ &= \frac{1}{2}\{\sigma_{n1}, \sigma_{n'1}\} \otimes \{\Pi_n^{(2)}, \Pi_{n'}^{(2)}\} \\ &\quad + \frac{1}{2}[\sigma_{n1}, \sigma_{n'1}] \otimes [\Pi_n^{(2)}, \Pi_{n'}^{(2)}]. \end{aligned} \tag{D.22}$$

Since either $[\sigma_i, \sigma_j] = 0$ or $\{\sigma_i, \sigma_j\} = 0$ for all $\sigma_i, \sigma_j \in \{I, X, Y, Z\}$, exactly one of the two terms in (D.21) and (D.22) must vanish. Thus, if we apply (D.21) and (D.22) recursively to expand $[\Pi_n, \Pi_n']$ as the Kronecker product of commutators and anti-commutators of the corresponding Pauli operators, we obtain the formula

$$[\Pi_n, \Pi_n'] = \left(1 - \prod_{j=1}^{N} \tau_j\right)(S_1 \otimes S_2 \otimes ... \otimes S_N), \tag{D.23}$$

where

$$\tau_j = \begin{cases} +1, & [\sigma_{nj}, \sigma_{n'j}] = 0 \\ -1, & [\sigma_{nj}, \sigma_{n'j}] \neq 0 \end{cases} \tag{D.24}$$

and

$$S_j = \begin{cases} \frac{1}{2}\{\sigma_{nj}, \sigma_{n'j}\}, & [\sigma_{nj}, \sigma_{n'j}] = 0 \\ \frac{1}{2}[\sigma_{nj}, \sigma_{n'j}], & [\sigma_{nj}, \sigma_{n'j}] \neq 0 \end{cases}. \tag{D.25}$$

From inspecting (D.23) and (D.24), we deduce that $[\Pi_n, \Pi_{n'}] = 0$ if and only if the number of non-commuting pairs of Pauli operators $\sigma_{nj}, \sigma_{n'j}$ is even.

Finally, we note that the conjugated Kronecker product $\Pi_n^{\overline{\otimes}2}$ is itself a Pauli string with an added sign of ± 1, since the map $\sigma_i \mapsto \overline{\sigma_i}$ fixes all Pauli operators except for $Y \mapsto \overline{Y} = -Y$. For an N-qubit string, we write

$$\Pi_n^{\overline{\otimes}2} = \pm(\sigma_{n1} \otimes \sigma_{n2} \otimes ... \otimes \sigma_{nN}) \otimes (\sigma_{n1} \otimes ... \otimes \sigma_{n2}). \tag{D.26}$$

For any two such strings $\Pi_n^{\overline{\otimes}2}, \Pi_{n'}^{\overline{\otimes}2}$ every non-commuting Pauli operator pair $\sigma_{nj}, \sigma_{n'j}$ appears twice in the string, so there are an even number of these pairs. It follows from (D.23) that $[\Pi_n^{\overline{\otimes}2}, \Pi_{n'}^{\overline{\otimes}2}] = 0$ as desired.

APPENDIX E

Appendix of "Efficient Simulation of Open Quantum Systems on NISQ Trapped-Ion Hardware"

This appendix published in: C. Burdine, N. Bauer, G. Siopsis, E. P. Blair, "Efficient Simulation of Open Quantum Systems on NISQ Trapped-Ion Hardware." *Advanced Quantum Technologies*, Feb. 2025, 2400606., doi: 10.1002/qute.202400606

E.1 Mapping the Kraus Series to Quantum Circuits

In this appendix, we give additional details concerning the mapping of Kraus operators to quantum circuits and provide some examples of how this can be done in an efficient manner for systems satisfying the commutation relations (i)-(iv). In Section 8.2.3.1, we discussed briefly how these Kraus operators can be implemented using unitary dilations of the Lindblad operators U_{L_n} and a unitary dilation of the effective time-evolution operator $T(t)$. We also discussed in Section 8.2.4 how certain special cases (such as bosonic systems and systems that form finite abelian groups) can be simulated with greater efficiency. However, since representations of bosonic operators $\hat{a}^\dagger$ and $\hat{a}$ are infinite-dimensional, they must be made discrete by truncating the system to a basis of finite occupation numbers up to some maximum occupation number state $|n_{\max}\rangle$. For more information on the selection of appropriate cutoff numbers, we refer the reader to the accompanying supplemental information document.

First, we consider the problem of realizing the effective evolution operator $T(t)$. As discussed in Section 8.2.3.1, the commutation relations (i) and (ii) ensure that the effective Hamiltonian H_{eff} (given in Eq. (8.9)) can be diagonalized in the same basis

as the system Hamiltonian H. It follows that $T(t)$ can then be diagonalized in the form

$$T(t) = UW(t)\Lambda(T)U^\dagger \tag{E.1}$$

where $W(t)$ is diagonal unitary and $\Lambda(t)$ is diagonal positive semi-definite with time-dependent entries of the form $W_{ii}(t) = e^{-i\theta_i t}$ and $\Lambda_{ii}(t) = e^{-\beta_i t}$, where $\theta_i \in [0, 2\pi)$ and $\beta_i \geq 0$ correspond to the imaginary and real components of $-itE_{\text{eff}}/\hbar$, where E_{eff} (given in Eq. (8.10)) is an eigenvalue of H_{eff}. While one could in theory apply the diagonalizing unitary U in the quantum circuit, this is often impractical on NISQ devices, since one might as well represent the entire system in this diagonal basis and avoid the expense of applying U altogether.

Methods of efficiently implementing diagonal unitaries $W(t)$ have been discussed in the literature (e.g., see [210]). Among these methods, there exists a general trade-off between classical pre-processing complexity and quantum gate complexity, depending on whether the diagonal elements are encoded with parameterized rotation gates via a standard binary or a Gray code encoding. In a binary encoding, the elements $W_{ii}(t)$ of a d-dimensional unitary are encoded through a sequence of d multi-qubit controlled phase gates, where the control qubits for the nth phase gate are the 1 bits in the binary representation of n. In this encoding, the phase angles corresponding to the entries $W_{ii}(t)$ can be computed efficiently in time $O(d\log(d))$ and can be executed with $O(d\log(d))$ quantum gates [229]. In a Gray code encoding, controlled rotation gates are applied to single qubits such that a sequence of parity-controlled R_z gates are applied to recursive sub-blocks of W [210]. An important benefit of the Gray code encoding is that it requires roughly $d/2$ CNOT gates when d is a power

of 2; however, the mapping of the entries $W_i i(t)$ to rotation gate parameters requires solving a $d^2 \times d^2$ linear system, which has a classical complexity between $O(d^2)$ and $O(d^3)$ that dominates the quantum simulation time for large systems. For this reason, we consider the standard binary encoding more favorable when simulating the trajectory of large systems over many t values, and the Gray code encoding more favorable for small systems with few t values. In the results presented in Section 8.3. The non-unitary diagonal matrix $\Lambda(t)$ can also be mapped to quantum gates through methods similar to those for the unitary $W(t)$; however instead of applying phase or R_z gates, controlled R_y gates are instead applied on an ancilla qubit with the system qubit as a control. This implements a unitary block encoding of the real diagonal operator $\Lambda(t)$. Both the standard binary encoding and the Gray code encoding of this non-unitary operator exhibit similar asymptotic quantum gate complexity and classical parameter mapping complexity as the diagonal opertor $W(t)$. For additional details regarding the trade-off, and for circuit diagrams implementing $\Lambda(t)$ and $W(t)$ for simple systems (such as the damped quantum harmonic oscillator), we refer the reader to reference [229].

E.2 Pauli Noise Models on NISQ Hardware

In this appendix, we consider Pauli noise models, and how they can be applied to density matrix simulations of open quantum systems on NISQ hardware. In Section 8.3.3, we introduced the general Pauli channel noise model Eq. (8.38), which represents the Markovian evolution of a system under a continuous-time Pauli channel for some unknown time. We also considered a special case of the Pauli channel model, called the quantum depolarizing channel (QDC), given by Eq. (8.40). Although it

is a relatively naive noise model, a QDC can be applied to model depolarization of quantum states, which is a type of decoherence that commutes with all other types of decoherence described by a CPTP map. Specifically, we claimed that the QDC satisfies the commutativity property Eq. (8.41), where $\mathcal{E}_{\text{sim}}$ is the CPTP map being simulated on the quantum device. This can be shown by expanding $\mathcal{E}_{\text{sim}}$ as a sum of Kraus operators K_i, so that by the unitality of the Kraus operators we obtain

$$\begin{aligned}
\mathcal{E}_{\text{sim}}(\mathcal{E}_{\text{QDC}}(\rho)) &= \sum_i K_i \left[(1-\lambda)\rho + \frac{\lambda}{2^N} I\right] K_i^\dagger \\
&= \sum_i (1-\lambda) K_i \rho K_i^\dagger + \frac{\lambda}{2^N} K_i K_i^\dagger \\
&= (1-\lambda)\mathcal{E}_{\text{sim}}(\rho) + \frac{\lambda}{2^N} I \\
&= \mathcal{E}_{\text{QDC}}(\mathcal{E}_{\text{sim}}(\rho))
\end{aligned} \tag{E.2}$$

as claimed. In a similar manner, it can also be shown that any two Pauli channels commute. Letting $P_0 \equiv I$ for convenience of notation, we see that

$$\begin{aligned}
\mathcal{E}_{\text{PC}}(\mathcal{E}'_{\text{PC}}(\rho)) &= \sum_i \varepsilon_i P_i \left[\sum_j \varepsilon'_j P'_j \rho P'^\dagger_j\right] P_i^\dagger \\
&= \sum_i \sum_j \varepsilon_i \varepsilon'_j P_i P'_j \rho P'^\dagger_j P_i^\dagger \\
&= \sum_j \sum_i \varepsilon'_j \varepsilon_i P'_j P_i \rho P_i^\dagger P'^\dagger_j \\
&= \mathcal{E}'_{\text{PC}}(\mathcal{E}_{\text{PC}}(\rho))
\end{aligned} \tag{E.3}$$

as claimed. This result should come as no surprise, because we have already shown that the Lindblad operators L_n of a continuous-time Pauli channel form a commutative group, hence the CPTP maps they generate should also commute.

Next, we consider the problem of fitting and inverting Pauli channels to measured data. Given a set of "exact" density matrices $\rho_1, \rho_2, ..., \rho_n$ and "noisy" density matrices $\tilde{\rho}_1, \tilde{\rho}_2, ..., \tilde{\rho}_n$, we aim to find error parameters ε_i such that the mean Frobenius

norm error $\sum_{i=1}^{n} \|\mathcal{E}_{PC}(\rho_i) - \tilde{\rho}_i\|_F$ for all noisy density matrices is minimized. This corresponds to the non-negative least squares regression problem

$$\begin{aligned} &\text{Minimize: } \frac{1}{n}\sum_{j=1}^{n}\left\|\tilde{\rho}_j - \sum_i \varepsilon_i P_i \rho_j P_i^\dagger\right\|_F^2 \\ &\text{subject to: } \varepsilon_i \geq 0 \end{aligned} \tag{E.4}$$

which can be solved through standard quadratic programming algorithms. A similar method can be used for the QDC. An important feature of Pauli noise models is that they can be inverted. Specifically, the action of a channel $\mathcal{E}(\rho)$ can be written as a matrix A acting on the vectorized density matrix $\vec{\rho}$:

$$A_{\mathcal{E}}\vec{\rho} = \vec{\tilde{\rho}} \tag{E.5}$$

where A is the $d^2 \times d^2$ matrix with entries

$$A_{\mathcal{E}} = \sum_i \varepsilon_i (P_i \otimes \overline{P_i}) \tag{E.6}$$

where $\overline{P_i}$ denotes the complex conjugate of P_i.

Given a noisy estimate of a density matrix $\tilde{\rho}$, one can compute the mitigated density matrix ρ_{mit} by applying the channel inverse. This is computed as

$$\vec{\rho}_{\text{mit}} = A_{\mathcal{E}}^{+}\vec{\tilde{\rho}}. \tag{E.7}$$

where A^+ denotes the matrix pseudoinverse. In the case of the QDC, the inverse can be computed with the much simpler formula

$$\rho_{\text{mit}} = \frac{1}{(1-\lambda)}(\tilde{\rho} - (\lambda/2^N)I). \tag{E.8}$$

For both the QDC and the general Pauli channel, applying the channel inverse produces a density matrix ρ_{mit} that is Hermitian with unit trace. Sometimes, however, the resulting density matrices have small negative eigenvalues. These “unphysical"

density matrices can be corrected by fixing all negative eigenvalues to zero and re-normalizing ρ_{mit} to have unit trace.

Because computing the QDC inverse is computationally inexpensive, we also note that the parameter λ can be estimated based on minimizing the reconstruction error of ρ_{mit} versus a known state without device noise. This can be done by either minimizing the mean Frobenius error $\sum_{j=1}^{n} \|\rho_j - \rho_{\text{mit}j}\|/n$ or by maximizing the fidelity of the reconstructed density matrix, which is computed as:

$$F(\rho, \rho_{\text{mit}}) = \left(\text{Tr}\left[(\rho^{\frac{1}{2}}\rho_{\text{mit}}\rho^{\frac{1}{2}})^{\frac{1}{2}}\right]\right)^2 \tag{E.9}$$

In this paper, we maximize Eq. (E.9) to find the optimal estimated value of λ for the QDC as shown in Figure 8.6(b). For the general Pauli channel, however, we use the non-negative least squares approach that solves Eq. (E.4) for the optimal coefficients ε_i, as shown in Figure 8.6.

APPENDIX F

PyQuinn Stable Eigensolver Implementation

F.1 Torch Backend Implementation

```
import torch
import warnings
from abc import abstractmethod, ABC


import torch.nn as nn


class _StableEigh(torch.autograd.Function):

    @staticmethod
    def forward(ctx, A, eps):
        eigvals, eigvecs = torch.linalg.eigh(A)
        ctx.save_for_backward(eigvals, eigvecs)
        ctx.eps = eps
        return eigvals, eigvecs

    @staticmethod
    def backward(ctx, grad_eigvals, grad_eigvecs):
        eigvals, eigvecs = ctx.saved_tensors
        eps = ctx.eps

        V = eigvecs  # (..., N, N)
        VT = V.conj().transpose(-2, -1)
        dLdLambda = grad_eigvals      # (..., N)
        dLdV = grad_eigvecs           # (..., N, N)

        # 1. Compute symmetric part: V @ diag(dL/dLambda) @ V^T
        grad_lambda_term = V @ torch.diag_embed(dLdLambda).to(V.
    dtype) @ VT  # (..., N, N)

        # 2. Compute eigenvector term
        #    Compute F_ij = 1 / (lambda_i - lambda_j) with 0 on
    diagonal
        lambda_i = eigvals.unsqueeze(-1)  # (..., N, 1)
        lambda_j = eigvals.unsqueeze(-2)  # (..., 1, N)
        delta = lambda_i - lambda_j       # (..., N, N)

        # Avoid divide-by-zero on diagonal
        F = torch.where(delta.abs() > eps, 1.0 / delta, torch.
    zeros_like(delta))

        # Project grad onto eigenbasis
        Vt_dLdV = VT @ dLdV               # (..., N, N)
```

```
42
43          # Apply symmetric correction
44          F_prod = F * Vt_dLdV
45          grad_v_term = V @ (F_prod + F_prod.conj().transpose(-2, -1))
    * 0.5 @ VT  # (..., N, N)
46
47          # Combine
48          grad_input = grad_lambda_term + grad_v_term
49          return grad_input, None  # Second arg is for eps
50
51
52  SUPPORTED_EIGENSOLVERS = []
53  """The full list of eigensolvers supported by default."""
54
55
56  class Eigensolver(nn.Module, ABC):
57      """ Abstract Base class for PyTorch Eigensolvers  """
58
59      _solvers = {}
60
61      def __init__(self):
62          super().__init__()
63
64      @staticmethod
65      def register(name: str):
66          """Registers an eigensolver subclass of this abstract class.
    You can
67          add a custom eigensolver by using the python decorator
   syntax as
68          follows:
69
70          .. code-block:: python
71
72              from pyquinn.torch.eigensolver import Eigensolver
73
74              @Eigensolver.register('my_eigensolver_name')
75              class MyEigensolver(Eigensolver):
76
77                  def diagonalize(...):
78                      ...
79
80          The registered eigensolver will be added to
81          :attr:`SUPPORTED_EIGENSOLVERS
82          <pyquinn.torch.eigensolver.SUPPORTED_EIGENSOLVERS>`
83
84          Args:
85              name (str): Eigensolver class to be registered.
86          """
87          def decorator(subclass):
88              if name not in Eigensolver._solvers:
89                  SUPPORTED_EIGENSOLVERS.append(name)
90              Eigensolver._solvers[name] = subclass
91              return subclass
92          return decorator
```

```
93
94      @staticmethod
95      def from_name(
96              solver_name: str,
97              solver_kwargs: dict | None = None) -> 'Eigensolver':
98          """Loads an eigensolver with the given name from the set of
99          supported eigensolvers in :attr:`SUPPORTED_EIGENSOLVERS
100         <pyquinn.torch.eigensolver.SUPPORTED_EIGENSOLVERS>`
101
102
103         Args:
104             solver_name (str): Name of eigensolver
105             solver_kwargs (dict, optional): Keyword args passed to
    eigensolver.
106                                             Defaults to {}.
107
108         Raises:
109             ValueError: If eigensolver name is invalid
110
111         Returns:
112             Eigensolver: Eigensolver with given name
113         """
114         if solver_name not in Eigensolver._solvers:
115             raise ValueError(f'Unknown Eigensolver Type: {
    solver_name}')
116
117         solver_kwargs = {} if solver_kwargs is None else
    solver_kwargs
118
119         return Eigensolver._solvers[solver_name](**solver_kwargs)
120
121     @abstractmethod
122     def diagonalize(self, M: torch.Tensor,
123                     **kwargs) -> tuple[torch.Tensor, torch.Tensor]:
124         raise NotImplementedError(
125             "This method should be implemented in a "
126             "subclass. To create a subclass instance, "
127             "use Eigensolver.from_name('<solver_name>').")
128
129     def forward(self, M: torch.Tensor) -> torch.Tensor:
130         return self.diagonalize()[0]
131
132
133 @Eigensolver.register('torch_eigh')
134 class EighEigensolver(Eigensolver):
135     """ Dense Matrix Eigensolver that uses `pytorch.linalg.eigh`
    with
136         added support for stabilizing gradients of degenerate
    eigenvalues."""
137
138     def __init__(self, eps=1e-5):
139         """Creates an instance of an EighEigensolver"""
140         super().__init__()
141         self.eps = eps
```

```
142
143     def diagonalize(self, H: torch.Tensor
144                     ) -> tuple[torch.Tensor, torch.Tensor]:
145         """Diagonalizes a dense Hermitian matrix (:math:`H = (H^*)^{
    T}`).
146         Supports backward passes for gradients with respect to
147         eigenvectors and eigenvalues.
148
149         .. warning::
150             Gradients with respect to eigenvectors are only uniquely
     defined
151             up to a phase. It is recommended that you define loss
    functions
152             using the modulus or modulus squared of each eigenvector
     component.
153
154             See `torch.linalg.eigh` for more information.
155
156         Args:
157             H (torch.Tensor): Hermitian matrix (shape: [..., N, N])
158
159         Returns:
160             tuple[torch.Tensor, torch.Tensor]: Eigenvalues and
    Eigenvectors
161             of the matrix H. (shape: [..., N], [..., N, N])
162         """
163         if not hasattr(self, 'eps'):
164             self.eps = 1e-5
165
166         return _StableEigh.apply(H, self.eps)
167
168     def forward(self, H: torch.Tensor) -> torch.Tensor:
169         """Computes the Eigenvalues of a dense Hermitian matrix
170         (:math:`H = (H^*)^{T}`). Supports backward passes for
171         stable gradients with respect to eigenvalues.
172
173         Args:
174             H (torch.Tensor): Hermitian matrix (shape: [..., N, N])
175
176         Returns:
177             tuple[torch.Tensor, torch.Tensor]: Eigenvalues of
178             the matrix H. (shape: [..., N])
179         """
180
181         return torch.linalg.eigvalsh(H)
182
183 @Eigensolver.register('torch_lobpcg')
184 class LobpcgEigensolver(Eigensolver):
185
186     def __init__(self, solver_kwargs: dict = {}):
187         super().__init__()
188         self.solver_kwargs = solver_kwargs
189
190         if not hasattr(self.__class__, '_initialized'):
```

```
            warnings.warn("The Pytorch API for LOBPCG diagonalization is "
                          "still under development. Backward passes and "
                          "complex Hamiltonians are not yet fully supported.",
                          UserWarning, stacklevel=2)
            self.__class__._initialized = True

    def diagonalize(self,
                    H: torch.Tensor,
                    E_min: torch.Tensor = torch.tensor(0.0),
                    n: torch.Tensor = torch.tensor(1),
                    S: torch.Tensor | None = None,
                    C: torch.Tensor | None = None) -> torch.Tensor:
        """Diagonalizes a dense symmetric matrix by solving a generalized
           eigenvalue problem for the `n` least eigenvectors of a
           real symmetric Hamiltonian :math:`H` with eigenvalues bounded
           below by some constant `E_min`. The generalized eigenvalue problem
           takes the form

        .. math::
            H C = SCE

        where :math:`C` is a matrix of eigenvalue coefficients, `E` is a
        diagonal matrix containing the `n` least eigenvalues,
        and :math:`S` is a basis overlap matrix (equal to the
        identity by default).

        .. warning::
            This eigensolver uses the `torch.lobpcg` function, which is still
            under development. Certain feature like backward passes for
            non-trivial :math:`S` matrices and complex Hamiltonians
            are not yet fully supported. Use at your own risk.

        Args:
            H (torch.Tensor): Hermitian matrix (shape: [..., N, N])
            E_min (torch.Tensor, optional): Lower bound on eigenvalues of the
                Hamiltonian. Defaults to torch.tensor(0.0).
            n (torch.Tensor, optional): Number of lowest eigenvalues to
                solve for. Defaults to 1.
            S (torch.Tensor | None, optional): Overlap matrix :math:`S`.
                (shape: [..., N, N]). Defaults to None.
            C (torch.Tensor | None, optional): Initial guess of eigenvectors
```

```
232                 (shape: [...,N,n]). Defaults to None.
233
234         Returns:
235             tuple[torch.Tensor, torch.Tensor]: Eigenvalues and
    Eigenvectors
236             of the matrix `H`. (shape: [..., n], [..., N, n])
237         """
238
239         H_posdef = H - E_min*torch.eye(H.shape[-1])
240
241         eigvals, eigvects = torch.lobpcg(A=H_posdef, k=n, B=S, X=C,
242                                          largest=False,
243                                          **self.solver_kwargs)
244         eigvals += E_min
245
246         return eigvals, eigvects
247
248     def forward(self,
249                 H: torch.Tensor,
250                 E_min: torch.Tensor = torch.tensor(0.0),
251                 n: torch.Tensor = torch.tensor(1),
252                 S: torch.Tensor | None = None,
253                 C: torch.Tensor | None = None):
254         """Diagonalizes a dense symmetric matrix by solving a
    generalized
255             eigenvalue problem for the `n` least eigenvalues of a
256             real symmetric Hamiltonian :math:`H` with the eigenvalues
     bounded
257             below by some constant `E_min`. The generalized
    eigenvalue problem
258             takes the form
259
260         .. math::
261             H C = SCE
262
263         where :math:`C` is a matrix of eigenvalue coefficients, `E`
    is a
264         diagonal matrix containing the `n` least eigenvalues,
265         and :math:`S` is a basis overlap matrix (equal to the
266         identity by default).
267
268         .. warning::
269             This eigensolver uses the `torch.lobpcg` function, which
     is still
270             under development. Certain feature like backward passes
    for
271             non-trivial :math:`S` matrices and complex Hamiltonians
272             are not yet fully supported. Use at your own risk.
273
274         Args:
275             H (torch.Tensor): Hermitian matrix (shape: [..., N, N])
276             E_min (torch.Tensor, optional): Lower bound on
    eigenvalues of the
277                 Hamiltonian. Defaults to torch.tensor(0.0).
```

```
278             n (torch.Tensor, optional): Number of lowest eigenvalues
    to
279                 solve for. Defaults to 1.
280             S (torch.Tensor | None, optional): Overlap matrix :math
    :`S`.
281                 (shape: [..., N, N]). Defaults to None.
282             C (torch.Tensor | None, optional): Initial guess of
    eigenvectors
283                 (shape: [...,N,n]). Defaults to None.
284 
285         Returns:
286             torch.tensor: Eigenvalues of the matrix `H`. (shape:
    [..., n])
287         """
288 
289         eigvals, _ = self.diagonalize(H=H, E_min=E_min, n=n, S=S, C=
    C)
290 
291         return eigvals
```

BIBLIOGRAPHY

[1] G. Ho Gu, J. Noh, I. Kim, and Y. Jung, "Machine learning for renewable energy materials", *Journal of Materials Chemistry A* **7** (2019), no. 29, 17096–17117, doi: 10.1039/C9TA02356A.

[2] N. Mamidi et al., "Recent Advances in Designing Fibrous Biomaterials for the Domain of Biomedical, Clinical, and Environmental Applications", *ACS Biomater. Sci. Eng.* **8** (September, 2022) 3690–3716, doi: 10.1021/acsbiomaterials.2c00786.

[3] D. Raabe, "The Materials Science behind Sustainable Metals and Alloys", *Chem. Rev.* **123** (March, 2023) 2436–2608, doi: 10.1021/acs.chemrev.2c00799.

[4] E. Persky, I. Sochnikov, and B. Kalisky, "Studying Quantum Materials with Scanning SQUID Microscopy", *Annual Review of Condensed Matter Physics* **13** (March, 2022) 385–405, doi: 10.1146/annurev-conmatphys-031620-104226.

[5] A. Passian and G. Siopsis, "Quantum state atomic force microscopy", *Phys. Rev. A* **95** (April, 2017) 043812, doi: 10.1103/PhysRevA.95.043812.

[6] A. Zong et al., "Emerging ultrafast techniques for studying quantum materials", *Nat Rev Mater* **8** (April, 2023) 224–240, doi: 10.1038/s41578-022-00530-0.

[7] M. Mitrano, S. Johnston, Y.-J. Kim, and M. P. M. Dean, "Exploring Quantum Materials with Resonant Inelastic X-Ray Scattering", *Phys. Rev. X* **14** (December, 2024) 040501, doi: 10.1103/PhysRevX.14.040501.

[8] W. Kohn and L. J. Sham, "Self-Consistent Equations Including Exchange and Correlation Effects", *Phys. Rev.* **140** (November, 1965) A1133–A1138, doi: 10.1103/PhysRev.140.A1133.

[9] R. K. Goyal, S. Maharaj, P. Kumar, and M. Chandrasekhar, "Exploring quantum materials and applications: a review", *J Mater. Sci: Mater Eng.* **20** (January, 2025) 4, doi: 10.1186/s40712-024-00202-7.

[10] J. Bardeen, "Theory of the Meissner Effect in Superconductors", *Phys. Rev.* **97** (March, 1955) 1724–1725, doi: 10.1103/PhysRev.97.1724.

[11] C. L. Kane and E. J. Mele, "$\{Z\}_\{2\}$ Topological Order and the Quantum Spin Hall Effect", *Phys. Rev. Lett.* **95** (September, 2005) 146802, doi: 10.1103/PhysRevLett.95.146802.

[12] A. Szabo and N. S. Ostlund, "Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory". Courier Corporation, July, 1996. ISBN 978-0-486-69186-2.

[13] F. Verstraete et al., "Density matrix renormalization group, 30 years on", *Nat Rev Phys* **5** (May, 2023) 273–276, doi: 10.1038/s42254-023-00572-5.

[14] J. Olsen, "The CASSCF method: A perspective and commentary", *International Journal of Quantum Chemistry* **111** (2011), no. 13, 3267–3272, doi: 10.1002/qua.23107.

[15] J. J. Eriksen, "The Shape of Full Configuration Interaction to Come", *J. Phys. Chem. Lett.* **12** (January, 2021) 418–432, doi: 10.1021/acs.jpclett.0c03225.

[16] J. Preskill, "Quantum Computing in the NISQ era and beyond", *Quantum* **2** (August, 2018) 79, doi: 10.22331/q-2018-08-06-79.

[17] N. Srivastava, "The Complexity of Diagonalization", in *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*, ISSAC '23, pp. 1–6. Association for Computing Machinery, New York, NY, USA, July, 2023. doi: 10.1145/3597066.3597145.

[18] M.-w. Xiao, "Theory of transformation for the diagonalization of quadratic Hamiltonians", August, 2009. doi: 10.48550/arXiv.0908.0787, `http://arxiv.org/abs/0908.0787`.

[19] S. Sachdev, "Quantum magnetism and criticality", *Nature Phys* **4** (March, 2008) 173–185, doi: 10.1038/nphys894.

[20] B. Brandow, "Electronic structure of Mott insulators", *Advances in Physics* **26** (September, 1977) 651–808, doi: 10.1080/00018737700101443.

[21] M. Y. Amusia, K. G. Popov, V. R. Shaginyan, and V. A. Stephanovich, "Theory of Heavy-Fermion Compounds: Theory of Strongly Correlated Fermi-Systems", volume 182 of *Springer Series in Solid-State Sciences*. Springer International Publishing, Cham, 2015. doi: 10.1007/978-3-319-10825-4, ISBN 978-3-319-10824-7 978-3-319-10825-4.

[22] H. Luo, P. Yu, G. Li, and K. Yan, "Topological quantum materials for energy conversion and storage", *Nat Rev Phys* **4** (September, 2022) 611–624, doi: 10.1038/s42254-022-00477-9.

[23] K. Balzer and M. Bonitz, "Nonequilibrium Green's Functions Approach to Inhomogeneous Systems". Springer, December, 2012. ISBN 978-3-642-35082-5.

[24] K. R. Bryenton, A. A. Adeleke, S. G. Dale, and E. R. Johnson, "Delocalization error: The greatest outstanding challenge in density-functional theory", *WIREs Computational Molecular Science* **13** (2023), no. 2, e1631, doi: 10.1002/wcms.1631.

[25] L. Reining, "The GW approximation: content, successes and limitations", *WIREs Computational Molecular Science* **8** (2018), no. 3, e1344, doi: 10.1002/wcms.1344.

[26] K. Held et al., "Realistic investigations of correlated electron systems with LDA + DMFT", *physica status solidi (b)* **243** (2006), no. 11, 2599–2631, doi: 10.1002/pssb.200642053.

[27] F. Becca and S. Sorella, "Quantum Monte Carlo Approaches for Correlated Systems". Cambridge University Press, November, 2017. Google-Books-ID: 4phADwAAQBAJ. ISBN 978-1-108-54731-4.

[28] "The sign problem in quantum Monte Carlo simulations", doi: 10.1016/B978-0-323-90800-9.00095-0.

[29] A. A. Emery and C. Wolverton, "High-throughput DFT calculations of formation energy, stability and oxygen vacancy formation energy of ABO3 perovskites", *Sci Data* **4** (October, 2017) 170153, doi: 10.1038/sdata.2017.153.

[30] J. E. Saal et al., "Materials Design and Discovery with High-Throughput Density Functional Theory: The Open Quantum Materials Database (OQMD)", *JOM* **65** (November, 2013) 1501–1509, doi: 10.1007/s11837-013-0755-4.

[31] K. Li et al., "A critical examination of robustness and generalizability of machine learning prediction of materials properties", *npj Comput Mater* **9** (April, 2023) 55, doi: 10.1038/s41524-023-01012-9.

[32] G. E. Karniadakis et al., "Physics-informed machine learning", *Nat Rev Phys* **3** (June, 2021) 422–440, doi: 10.1038/s42254-021-00314-5.

[33] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, "A general-purpose machine learning framework for predicting properties of inorganic materials", *npj Comput Mater* **2** (August, 2016) 1–7, doi: 10.1038/npjcompumats.2016.28.

[34] K. Choudhary and B. DeCost, "Atomistic Line Graph Neural Network for improved materials property predictions", *npj Comput Mater* **7** (November, 2021) 1–8, doi: 10.1038/s41524-021-00650-1.

[35] V. Botu, R. Batra, J. Chapman, and R. Ramprasad, "Machine Learning Force Fields: Construction, Validation, and Outlook", *J. Phys. Chem. C* **121** (January, 2017) 511–522, doi: 10.1021/acs.jpcc.6b10908.

[36] A. S. Fuhr and B. G. Sumpter, "Deep Generative Models for Materials Discovery and Machine Learning-Accelerated Innovation", *Frontiers in Materials* **9** (2022).

[37] B. Sanchez-Lengeling and A. Aspuru-Guzik, "Inverse molecular design using machine learning: Generative models for matter engineering", *Science* **361** (July, 2018) 360–365, doi: 10.1126/science.aat2663.

[38] N. Andrejevic et al., "Machine learning spectral indicators of topology", April, 2022. arXiv:2003.00994 [cond-mat, physics:physics] version: 3. `http://arxiv.org/abs/2003.00994`.

[39] V. L. Deringer, M. A. Caro, and G. Csányi, "Machine Learning Interatomic Potentials as Emerging Tools for Materials Science", *Advanced Materials* **31** (2019), no. 46, 1902765, doi: 10.1002/adma.201902765.

[40] M. S. Scheurer and R.-J. Slager, "Unsupervised Machine Learning and Band Topology", *Phys. Rev. Lett.* **124** (June, 2020) 226401, doi: 10.1103/PhysRevLett.124.226401.

[41] Y. Tian, Y. Zhang, and H. Zhang, "Recent Advances in Stochastic Gradient Descent in Deep Learning", *Mathematics* **11** (January, 2023) 682, doi: 10.3390/math11030682.

[42] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization", January, 2019. arXiv:1711.05101 [cs, math]. doi: 10.48550/arXiv.1711.05101, `http://arxiv.org/abs/1711.05101`.

[43] L. Prechelt, "Early Stopping - But When?", in *Neural Networks: Tricks of the Trade*, G. B. Orr and K.-R. Müller, eds., pp. 55–69. Springer, Berlin, Heidelberg, 1998. doi: 10.1007/3-540-49430-8_3.

[44] N. Srivastava et al., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting",.

[45] V. Fortuin, "Priors in Bayesian Deep Learning: A Review", *International Statistical Review* **90** (2022), no. 3, 563–591, doi: 10.1111/insr.12502.

[46] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules", *J. Chem. Inf. Comput. Sci.* **28** (February, 1988) 31–36, doi: 10.1021/ci00057a005.

[47] K.-L. Du et al., "Exploring Kernel Machines and Support Vector Machines: Principles, Techniques, and Future Directions", *Mathematics* **12** (January, 2024) 3935, doi: 10.3390/math12243935.

[48] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions", *Journal of Mathematical Psychology* **85** (August, 2018) 1–16, doi: 10.1016/j.jmp.2018.03.001.

[49] S. De, A. P. Bartók, G. Csányi, and M. Ceriotti, "Comparing molecules and solids across structural and alchemical space", *Phys. Chem. Chem. Phys.* **18** (May, 2016) 13754–13769, doi: 10.1039/C6CP00415F.

[50] T. Szandała, "Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks", in *Bio-inspired Neurocomputing*, A. K. Bhoi, P. K. Mallick, C.-M. Liu, and V. E. Balas, eds., pp. 203–224. Springer, Singapore, 2021. doi: 10.1007/978-981-15-5495-7_11.

[51] P. Kidger and T. Lyons, "Universal Approximation with Deep Narrow Networks", in *Proceedings of Thirty Third Conference on Learning Theory*, pp. 2306–2327. PMLR, July, 2020. ISSN: 2640-3498.

[52] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, "Recent advances and applications of machine learning in solid-state materials science", *npj Comput Mater* **5** (August, 2019) 1–36, doi: 10.1038/s41524-019-0221-0.

[53] H. J. Kulik et al., "Roadmap on Machine learning in electronic structure", *Electron. Struct.* **4** (August, 2022) 023004, doi: 10.1088/2516-1075/ac572f.

[54] A. Jain et al., "Commentary: The Materials Project: A materials genome approach to accelerating materials innovation", *APL Materials* **1** (July, 2013) 011002, doi: 10.1063/1.4812323.

[55] S. Curtarolo et al., "AFLOW: An automatic framework for high-throughput materials discovery", *Computational Materials Science* **58** (June, 2012) 218–226, doi: 10.1016/j.commatsci.2012.02.005.

[56] S. Kirklin et al., "The Open Quantum Materials Database (OQMD): assessing the accuracy of DFT formation energies", *npj Comput Mater* **1** (December, 2015) 15010, doi: 10.1038/npjcompumats.2015.10.

[57] A. G. Kusne, T. Mueller, and R. Ramprasad, "Machine learning in materials science: Recent progress and emerging applications", *NIST* (May, 2016).

[58] R. Drautz, "Atomic cluster expansion for accurate and transferable interatomic potentials", *Phys. Rev. B* **99** (January, 2019) 014104, doi: 10.1103/PhysRevB.99.014104.

[59] H. Li et al., "Deep-learning density functional theory Hamiltonian for efficient ab initio electronic-structure calculation", *Nat Comput Sci* **2** (June, 2022) 367–377, doi: 10.1038/s43588-022-00265-6.

[60] Y. Wang et al., "DeepH-2: Enhancing deep-learning electronic structure via an equivariant local-coordinate transformer", January, 2024. arXiv:2401.17015 [physics]. doi: 10.48550/arXiv.2401.17015, `http://arxiv.org/abs/2401.17015`.

[61] A. Glielmo, P. Sollich, and A. De Vita, "Accurate interatomic force fields via machine learning with covariant kernels", *Phys. Rev. B* **95** (June, 2017) 214302, doi: 10.1103/PhysRevB.95.214302.

[62] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons", *Phys. Rev. Lett.* **104** (April, 2010) 136403, doi: 10.1103/PhysRevLett.104.136403.

[63] X. Dong, E. Gull, and L. Wang, "Equivariant neural network for Green's functions of molecules and materials", *Phys. Rev. B* **109** (February, 2024) 075112, doi: 10.1103/PhysRevB.109.075112.

[64] E. Sheridan et al., "Data-driven dynamical mean-field theory: An error-correction approach to solve the quantum many-body problem using machine learning", *Phys. Rev. B* **104** (November, 2021) 205120, doi: 10.1103/PhysRevB.104.205120.

[65] G. Corso et al., "Graph neural networks", *Nat Rev Methods Primers* **4** (March, 2024) 17, doi: 10.1038/s43586-024-00294-7.

[66] J. Gilmer et al., "Neural Message Passing for Quantum Chemistry", in *Proceedings of the 34th International Conference on Machine Learning*, pp. 1263–1272. PMLR, July, 2017. ISSN: 2640-3498.

[67] N. Thomas et al., "Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds", May, 2018. arXiv:1802.08219 [cs]. doi: 10.48550/arXiv.1802.08219, `http://arxiv.org/abs/1802.08219`.

[68] Z. Li et al., "Unifying O(3) Equivariant Neural Networks Design with Tensor-Network Formalism", *Mach. Learn.: Sci. Technol.* **5** (June, 2024) 025044, doi: 10.1088/2632-2153/ad4a04.

[69] X. Niu et al., "Chiral Materials: Progress, Applications, and Prospects", *Small* **19** (2023), no. 38, 2303059, doi: 10.1002/smll.202303059.

[70] F. Devínsky, "Chirality and the Origin of Life", *Symmetry* **13** (December, 2021) 2277, doi: 10.3390/sym13122277.

[71] E. Bedolla, L. C. Padierna, and R. Castañeda-Priego, "Machine learning for condensed matter physics", *J. Phys.: Condens. Matter* **33** (November, 2020) 053001, doi: 10.1088/1361-648X/abb895.

[72] P. Zhang, H. Shen, and H. Zhai, "Machine Learning Topological Invariants with Neural Networks", *Phys. Rev. Lett.* **120** (February, 2018) 066401, doi: 10.1103/PhysRevLett.120.066401.

[73] Y. Lvovsky, E. W. Stautner, and T. Zhang, "Novel technologies and configurations of superconducting magnets for MRI", *Supercond. Sci. Technol.* **26** (July, 2013) 093001, doi: 10.1088/0953-2048/26/9/093001.

[74] P. Bruzzone et al., "High temperature superconductors for fusion magnets", *Nucl. Fusion* **58** (August, 2018) 103001, doi: 10.1088/1741-4326/aad835.

[75] J. M. Gambetta, J. M. Chow, and M. Steffen, "Building logical qubits in a superconducting quantum computing system", *npj Quantum Inf* **3** (January, 2017) 1–7, doi: 10.1038/s41534-016-0004-0.

[76] C. L. Degen, F. Reinhard, and P. Cappellaro, "Quantum sensing", *Rev. Mod. Phys.* **89** (July, 2017) 035002, doi: 10.1103/RevModPhys.89.035002.

[77] M. Mirhosseini, A. Sipahigil, M. Kalaee, and O. Painter, "Superconducting qubit to optical photon transduction", *Nature* **588** (December, 2020) 599–603, doi: 10.1038/s41586-020-3038-6.

[78] J. Bardeen, L. N. Cooper, and J. R. Schrieffer, "Theory of Superconductivity", *Phys. Rev.* **108** (December, 1957) 1175–1204, doi: 10.1103/PhysRev.108.1175.

[79] X. Zhou et al., "High-temperature superconductivity", *Nat Rev Phys* **3** (July, 2021) 462–465, doi: 10.1038/s42254-021-00324-3.

[80] Q. Liu et al., "Robust and Clean Majorana Zero Mode in the Vortex Core of High-Temperature Superconductor (Li0.84Fe0.16)OHFeSe", *Phys. Rev. X* **8** (December, 2018) 041056, doi: 10.1103/PhysRevX.8.041056.

[81] "NIMS Materials Database(MatNavi) - DICE :: National Institute for Materials Science". `https://mits.nims.go.jp/en/`.

[82] V. Stanev et al., "Machine learning modeling of superconducting critical temperature", *npj Comput Mater* **4** (June, 2018) 1–14, doi: 10.1038/s41524-018-0085-8.

[83] L. Foppiano et al., "Automatic extraction of materials and properties from superconductors scientific literature", *Science and Technology of Advanced Materials: Methods* **3** (December, 2023) 2153633, doi: 10.1080/27660400.2022.2153633.

[84] T. Konno et al., "Deep learning model for finding new superconductors", *Phys. Rev. B* **103** (January, 2021) 014509, doi: 10.1103/PhysRevB.103.014509.

[85] C. Seegmiller, S. Baird, H. Sayeed, and T. Sparks, "Discovering Chemically Novel, High-Temperature Superconductors". March, 2023. doi: 10.26434/chemrxiv-2023-8t8kt.

[86] S. R. Xie et al., "Machine learning of superconducting critical temperature from Eliashberg theory", *npj Comput Mater* **8** (January, 2022) 1–8, doi: 10.1038/s41524-021-00666-7.

[87] K. Choudhary and K. Garrity, "Designing high-TC superconductors with BCS-inspired screening, density functional theory, and deep-learning", *npj Comput Mater* **8** (November, 2022) 1–12, doi: 10.1038/s41524-022-00933-1.

[88] D. Aoki et al., "Unconventional superconductivity in UTe2", *J. Phys.: Condens. Matter* **34** (April, 2022) 243002, doi: 10.1088/1361-648X/ac5863.

[89] G. Leibfried and N. Breuer, "Point Defects in Metals I: Introduction to the Theory". Springer, April, 2006. Google-Books-ID: w5d0DgAAQBAJ. ISBN 978-3-540-37201-1.

[90] L. Shen, P. A. Salvador, and T. O. Mason, "Point Defect Modeling of La2CuO4-Based Superconductors", *Journal of the American Ceramic Society* **77** (1994), no. 1, 81–88, doi: 10.1111/j.1151-2916.1994.tb06960.x.

[91] M. S. Daw, S. M. Foiles, and M. I. Baskes, "The embedded-atom method: a review of theory and applications", *Materials Science Reports* **9** (March, 1993) 251–310, doi: 10.1016/0920-2307(93)90001-U.

[92] P. Giannozzi et al., "QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials", *J. Phys.: Condens. Matter* **21** (September, 2009) 395502, doi: 10.1088/0953-8984/21/39/395502.

[93] J. P. Perdew, M. Ernzerhof, and K. Burke, "Rationale for mixing exact exchange with density functional approximations", *J. Chem. Phys.* **105** (December, 1996) 9982–9985, doi: 10.1063/1.472933.

[94] G. Prandini et al., "Precision and efficiency in solid-state pseudopotential calculations", *npj Comput Mater* **4** (December, 2018) 1–13, doi: 10.1038/s41524-018-0127-2.

[95] A. Dunn et al., "Benchmarking materials property prediction methods: the Matbench test set and Automatminer reference algorithm", *npj Comput Mater* **6** (September, 2020) 1–10, doi: 10.1038/s41524-020-00406-3.

[96] A. P. Drozdov et al., "Conventional superconductivity at 203 kelvin at high pressures in the sulfur hydride system", *Nature* **525** (September, 2015) 73–76, doi: 10.1038/nature14964.

[97] J.-F. Ge et al., "Superconductivity above 100 K in single-layer FeSe films on doped SrTiO3", *Nature Mater* **14** (March, 2015) 285–289, doi: 10.1038/nmat4153.

[98] M. Gelbrich, "On a Formula for the L2 Wasserstein Metric between Measures on Euclidean and Hilbert Spaces", *Mathematische Nachrichten* **147** (1990), no. 1, 185–203, doi: 10.1002/mana.19901470121.

[99] H.-Y. Lv et al., "Metallization and superconductivity in methane doped by beryllium at low pressure", *Phys. Chem. Chem. Phys.* **22** (January, 2020) 1069–1077, doi: 10.1039/C9CP06008A.

[100] Z.-J. Yao, J.-X. Li, and Z. D. Wang, "Spin fluctuations, interband coupling and unconventional pairing in iron-based superconductors", *New J. Phys.* **11** (February, 2009) 025009, doi: 10.1088/1367-2630/11/2/025009.

[101] F. Steglich and S. Wirth, "Foundations of heavy-fermion superconductivity: lattice Kondo effect and Mott physics", *Rep. Prog. Phys.* **79** (July, 2016) 084502, doi: 10.1088/0034-4885/79/8/084502.

[102] R. Ramprasad et al., "Machine learning in materials informatics: recent applications and prospects", *npj Comput Mater* **3** (December, 2017) 1–13, doi: 10.1038/s41524-017-0056-5.

[103] G. R. Schleder et al., "From DFT to machine learning: recent approaches to materials science–a review", *J. Phys. Mater.* **2** (May, 2019) 032001, doi: 10.1088/2515-7639/ab084b.

[104] C. Karpovich, E. Pan, and E. A. Olivetti, "Deep reinforcement learning for inverse inorganic materials design", *npj Comput Mater* **10** (December, 2024) 287, doi: 10.1038/s41524-024-01474-5.

[105] P. M. Gill, "Molecular integrals Over Gaussian Basis Functions", in *Advances in Quantum Chemistry*, volume 25, pp. 141–205. Elsevier, 1994. doi: 10.1016/S0065-3276(08)60019-2.

[106] S. Cuomo et al., "Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next", *J Sci Comput* **92** (July, 2022) 88, doi: 10.1007/s10915-022-01939-z.

[107] E. J. R. Coutinho et al., "Physics-informed neural networks with adaptive localized artificial viscosity", *Journal of Computational Physics* **489** (September, 2023) 112265, doi: 10.1016/j.jcp.2023.112265.

[108] W. Kohn, A. D. Becke, and R. G. Parr, "Density Functional Theory of Electronic Structure", *J. Phys. Chem.* **100** (January, 1996) 12974–12980, doi: 10.1021/jp960669l.

[109] J. Pan, "Scaling up system size in materials simulation", *Nat Comput Sci* **1** (February, 2021) 95–95, doi: 10.1038/s43588-021-00034-x.

[110] P. Hohenberg and W. Kohn, "Inhomogeneous Electron Gas", *Phys. Rev.* **136** (November, 1964) B864–B871, doi: 10.1103/PhysRev.136.B864.

[111] J. Hafner, "Ab-initio simulations of materials using VASP: Density-functional theory and beyond", *Journal of Computational Chemistry* **29** (2008), no. 13, 2044–2078, doi: 10.1002/jcc.21057.

[112] C. Hermann and C. Weisbuch, "$\stackrel{\ensuremath{\rightarrow}}{\mathrm{k}}\ifmmode\cdot\else\textperiodcentered\fi{}\s perturbation theory in III-V compounds and alloys: a reexamination", *Phys. Rev. B* **15** (January, 1977) 823–833, doi: 10.1103/PhysRevB.15.823.

[113] T. Xie and J. C. Grossman, "Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties", *Phys. Rev. Lett.* **120** (April, 2018) 145301, doi: 10.1103/PhysRevLett.120.145301.

[114] X. M. Feng, P. Wang, W. Yang, and G. R. Jin, "High-precision evaluation of Wigner's d matrix by exact diagonalization", *Phys. Rev. E* **92** (October, 2015) 043307, doi: 10.1103/PhysRevE.92.043307.

[115] M. Geiger and T. Smidt, "e3nn: Euclidean Neural Networks", July, 2022. arXiv:2207.09453. doi: 10.48550/arXiv.2207.09453, `http://arxiv.org/abs/2207.09453`.

[116] A. R. Edmonds, "Angular momentum in quantum mechanics". Princeton, N.J. : Princeton University Presss, 1960. ISBN 978-0-691-07912-7.

[117] L. Zhang, "Berry curvature and various thermal Hall effects", *New J. Phys.* **18** (October, 2016) 103039, doi: 10.1088/1367-2630/18/10/103039.

[118] F. M. Mueller, A. J. Freeman, J. O. Dimmock, and A. M. Furdyna, "Electronic Structure of Palladium", *Phys. Rev. B* **1** (June, 1970) 4617–4635, doi: 10.1103/PhysRevB.1.4617.

[119] P. Giannozzi et al., "Advanced capabilities for materials modelling with Quantum ESPRESSO", *J. Phys.: Condens. Matter* **29** (October, 2017) 465901, doi: 10.1088/1361-648X/aa8f79.

[120] X. Gong et al., "General framework for E(3)-equivariant neural network representation of density functional theory Hamiltonian", *Nat Commun* **14** (May, 2023) 2848, doi: 10.1038/s41467-023-38468-8.

[121] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library", December, 2019. arXiv:1912.01703 [cs, stat]. doi: 10.48550/arXiv.1912.01703, `http://arxiv.org/abs/1912.01703`.

[122] M. Abadi, "TensorFlow: learning functions at scale", in *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming*, ICFP 2016, p. 1. Association for Computing Machinery, New York, NY, USA, September, 2016. doi: 10.1145/2951913.2976746.

[123] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization", January, 2017. arXiv:1412.6980 [cs]. doi: 10.48550/arXiv.1412.6980, `http://arxiv.org/abs/1412.6980`.

[124] S. R. Hall, F. H. Allen, and I. D. Brown, "The crystallographic information file (CIF): A new standard archive file for crystallography",.

[125] J. D. Hunter, "Matplotlib: A 2D Graphics Environment", *Computing in Science & Engineering* **9** (May, 2007) 90–95, doi: 10.1109/MCSE.2007.55.

[126] "Plotly". `https://plotly.com/python/`.

[127] M. A. Nielsen and I. L. Chuang, "Quantum Computation and Quantum Information". Cambridge University Press, October, 2000. ISBN 978-0-521-63503-5.

[128] R. P. Feynman, "Simulating Physics with Computers", in *Feynman And Computation*. CRC Press, 2002. Num Pages: 21.

[129] C. M. Dawson and M. A. Nielsen, "The Solovay-Kitaev algorithm", August, 2005. arXiv:quant-ph/0505030. doi: 10.48550/arXiv.quant-ph/0505030, `http://arxiv.org/abs/quant-ph/0505030`.

[130] G. Kuperberg, "Breaking the cubic barrier in the Solovay-Kitaev algorithm", June, 2023. doi: 10.48550/arXiv.2306.13158, `http://arxiv.org/abs/2306.13158`.

[131] E. Knill, R. Laflamme, and G. J. Milburn, "A scheme for efficient quantum computation with linear optics", *Nature* **409** (January, 2001) 46–52, doi: 10.1038/35051009.

[132] S. Faisal, S. U. Khan, M. Kordasz, and K. A. Cyran, "Advancements and Challenges in Linear Quantum optics: A Comprehensive Review of Quantum Information processing", in *2025 11th International Conference on Control, Decision and Information Technologies (CoDIT)*, volume 1, pp. 1131–1136. July, 2025. ISSN: 2576-3555. doi: 10.1109/CoDIT66093.2025.11321337.

[133] Y. Suzuki, S. Endo, K. Fujii, and Y. Tokunaga, "Quantum Error Mitigation as a Universal Error Reduction Technique: Applications from the NISQ to the Fault-Tolerant Quantum Computing Eras", *PRX Quantum* **3** (March, 2022) 010345, doi: 10.1103/PRXQuantum.3.010345.

[134] J. S. Bell, "On the Einstein Podolsky Rosen paradox", *Physics Physique Fizika* **1** (November, 1964) 195–200, doi: 10.1103/PhysicsPhysiqueFizika.1.195.

[135] J. F. Clauser and A. Shimony, "Bell's theorem. Experimental tests and implications", *Rep. Prog. Phys.* **41** (December, 1978) 1881, doi: 10.1088/0034-4885/41/12/002.

[136] C. Budroni et al., "Kochen-Specker contextuality", *Rev. Mod. Phys.* **94** (December, 2022) 045007, doi: 10.1103/RevModPhys.94.045007.

[137] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits", *Phys. Rev. A* **70** (November, 2004) 052328, doi: 10.1103/PhysRevA.70.052328.

[138] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", *SIAM J. Comput.* **26** (October, 1997) 1484–1509, doi: 10.1137/S0097539795293172.

[139] A. J. Daley et al., "Practical quantum advantage in quantum simulation", *Nature* **607** (July, 2022) 667–676, doi: 10.1038/s41586-022-04940-6.

[140] M. Sipser, "The history and status of the P versus NP question", in *Proceedings of the twenty-fourth annual ACM symposium on Theory of Computing*, STOC '92, pp. 603–618. Association for Computing Machinery, New York, NY, USA, July, 1992. doi: 10.1145/129712.129771.

[141] O. Goldreich, "In a World of P=BPP", in *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation: In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, O. Goldreich, ed., pp. 191–232. Springer, Berlin, Heidelberg, 2011. doi: 10.1007/978-3-642-22670-0_20.

[142] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring", in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134. November, 1994. doi: 10.1109/SFCS.1994.365700.

[143] 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, pp. 212–219. Association for Computing Machinery, New York, NY, USA, July, 1996. doi: 10.1145/237814.237866.

[144] S. Aaronson, "BQP and the polynomial hierarchy", in *Proceedings of the forty-second ACM symposium on Theory of computing*, STOC '10, pp. 141–150. Association for Computing Machinery, New York, NY, USA, June, 2010. doi: 10.1145/1806689.1806711.

[145] J. Kempe, A. Kitaev, and O. Regev, "The Complexity of the Local Hamiltonian Problem", *SIAM J. Comput.* **35** (January, 2006) 1070–1097, doi: 10.1137/S0097539704445226.

[146] T. Cubitt and A. Montanaro, "Complexity Classification of Local Hamiltonian Problems", *SIAM J. Comput.* **45** (January, 2016) 268–316, doi: 10.1137/140998287.

[147] S. Aaronson, "On Perfect Completeness for QMA", August, 2008. arXiv:0806.0450 [quant-ph]. doi: 10.48550/arXiv.0806.0450, `http://arxiv.org/abs/0806.0450`.

[148] F. Barahona, "On the computational complexity of Ising spin glass models", *J. Phys. A: Math. Gen.* **15** (October, 1982) 3241, doi: 10.1088/0305-4470/15/10/028.

[149] C. Cade et al., "Improved Hardness Results for the Guided Local Hamiltonian Problem", in *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, K. Etessami, U. Feige, and G. Puppis, eds., volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 32:1–32:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2023. ISSN: 1868-8969. doi: 10.4230/LIPIcs.ICALP.2023.32.

[150] J. Tilly et al., "The Variational Quantum Eigensolver: A review of methods and best practices", *Physics Reports* **986** (November, 2022) 1–128, doi: 10.1016/j.physrep.2022.08.003.

[151] M. A. Nielsen, "The Fermionic canonical commutation relations and the Jordan-Wigner transform",.

[152] S. B. Bravyi and A. Y. Kitaev, "Fermionic Quantum Computation", *Annals of Physics* **298** (May, 2002) 210–226, doi: 10.1006/aphy.2002.6254.

[153] S. Bravyi, J. M. Gambetta, A. Mezzacapo, and K. Temme, "Tapering off qubits to simulate fermionic Hamiltonians", January, 2017. doi: 10.48550/arXiv.1701.08213, `http://arxiv.org/abs/1701.08213`.

[154] L. Leone, S. F. E. Oliviero, L. Cincio, and M. Cerezo, "On the practical usefulness of the Hardware Efficient Ansatz", *Quantum* **8** (July, 2024) 1395, doi: 10.22331/q-2024-07-03-1395.

[155] P. K. Barkoutsos et al., "Quantum algorithms for electronic structure calculations: Particle-hole Hamiltonian and optimized wave-function expansions", *Phys. Rev. A* **98** (August, 2018) 022322, doi: 10.1103/PhysRevA.98.022322.

[156] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, "Connecting Ansatz Expressibility to Gradient Magnitudes and Barren Plateaus", *PRX Quantum* **3** (January, 2022) 010313, doi: 10.1103/PRXQuantum.3.010313.

[157] A. Anand et al., "A quantum computing view on unitary coupled cluster theory", *Chem. Soc. Rev.* **51** (March, 2022) 1659–1684, doi: 10.1039/D1CS00932J.

[158] X. Liu et al., "Mitigating Barren Plateaus of Variational Quantum Eigensolvers", *IEEE Transactions on Quantum Engineering* **5** (2024) 1–19, doi: 10.1109/TQE.2024.3383050.

[159] A. M. Childs et al., "Theory of Trotter Error with Commutator Scaling", *Phys. Rev. X* **11** (February, 2021) 011020, doi: 10.1103/PhysRevX.11.011020.

[160] G. H. Low and I. L. Chuang, "Optimal Hamiltonian Simulation by Quantum Signal Processing", *Phys. Rev. Lett.* **118** (January, 2017) 010501, doi: 10.1103/PhysRevLett.118.010501.

[161] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics", in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pp. 193–204. Association for Computing Machinery, New York, NY, USA, June, 2019. doi: 10.1145/3313276.3316366.

[162] S. Lloyd et al., "Hamiltonian singular value transformation and inverse block encoding", May, 2021. doi: 10.48550/arXiv.2104.01410, http://arxiv.org/abs/2104.01410.

[163] Y. Jin et al., "Optimizing Quantum Fourier Transformation (QFT) Kernels for Modern NISQ and FT Architectures", in *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–15. November, 2024. doi: 10.1109/SC41406.2024.00074.

[164] U. Dorner et al., "Optimal Quantum Phase Estimation", *Phys. Rev. Lett.* **102** (January, 2009) 040403, doi: 10.1103/PhysRevLett.102.040403.

[165] A. Y. Kitaev, "Quantum measurements and the Abelian Stabilizer Problem", November, 1995. arXiv:quant-ph/9511026. doi: 10.48550/arXiv.quant-ph/9511026, http://arxiv.org/abs/quant-ph/9511026.

[166] H.-P. Breuer and F. Petruccione, "The Theory of Open Quantum Systems". Oxford University Press, 2002. ISBN 978-0-19-852063-4.

[167] L. Clinton, J. Bausch, and T. Cubitt, "Hamiltonian simulation algorithms for near-term quantum hardware", *Nat Commun* **12** (August, 2021) 4989, doi: 10.1038/s41467-021-25196-0.

[168] A. Smith, M. S. Kim, F. Pollmann, and J. Knolle, "Simulating quantum many-body dynamics on a current digital quantum computer", *npj Quantum Inf* **5** (November, 2019) 1–13, doi: 10.1038/s41534-019-0217-0.

[169] Z. Jiang et al., "Quantum Algorithms to Simulate Many-Body Physics of Correlated Fermions", *Phys. Rev. Applied* **9** (April, 2018) 044036, doi: 10.1103/PhysRevApplied.9.044036.

[170] A. Y. Kitaev, "Quantum computations: algorithms and error correction", *Russ. Math. Surv.* **52** (December, 1997) 1191, doi: 10.1070/RM1997v052n06ABEH002155.

[171] L. Lamata, A. Mezzacapo, J. Casanova, and E. Solano, "Efficient quantum simulation of fermionic and bosonic models in trapped ions", *EPJ Quantum Technol.* **1** (December, 2014) 1–13, doi: 10.1140/epjqt9.

[172] A. Macridin, P. Spentzouris, J. Amundson, and R. Harnik, "Digital quantum computation of fermion-boson interacting systems", *Phys. Rev. A* **98** (October, 2018) 042312, doi: 10.1103/PhysRevA.98.042312.

[173] F. Tacchino, A. Chiesa, S. Carretta, and D. Gerace, "Quantum Computers as Universal Quantum Simulators: State-of-the-Art and Perspectives", *Advanced Quantum Technologies* **3** (2020), no. 3, 1900052, doi: 10.1002/qute.201900052.

[174] Y. Wang et al., "Simulating Open Quantum System Dynamics on NISQ Computers with Generalized Quantum Master Equations", *J. Chem. Theory Comput.* **19** (August, 2023) 4851–4862, doi: 10.1021/acs.jctc.3c00316.

[175] H.-Y. Su and Y. Li, "Quantum algorithm for the simulation of open-system dynamics and thermalization", *Phys. Rev. A* **101** (January, 2020) 012328, doi: 10.1103/PhysRevA.101.012328.

[176] M. Müller, S. Diehl, G. Pupillo, and P. Zoller, "Engineered Open Systems and Quantum Simulations with Atoms and Ions", in *Advances In Atomic, Molecular, and Optical Physics*, P. Berman, E. Arimondo, and C. Lin, eds., volume 61 of *Advances in Atomic, Molecular, and Optical Physics*, pp. 1–80. Academic Press, July, 2012. doi: 10.1016/B978-0-12-396482-3.00001-6.

[177] H. Kamakari, S.-N. Sun, M. Motta, and A. J. Minnich, "Digital Quantum Simulation of Open Quantum Systems Using Quantum Imaginary–Time Evolution", *PRX Quantum* **3** (February, 2022) 010320, doi: 10.1103/PRXQuantum.3.010320.

[178] M. Kliesch et al., "Dissipative Quantum Church-Turing Theorem", *Phys. Rev. Lett.* **107** (September, 2011) 120501, doi: 10.1103/PhysRevLett.107.120501.

[179] A. W. Schlimgen et al., "Quantum simulation of the Lindblad equation using a unitary decomposition of operators", *Phys. Rev. Res.* **4** (June, 2022) 023216, doi: 10.1103/PhysRevResearch.4.023216.

[180] M. Jo and M. Kim, "Simulating open quantum many-body systems using optimised circuits in digital quantum simulation", March, 2022. arXiv:2203.14295 [cond-mat, physics:quant-ph]. `http://arxiv.org/abs/2203.14295`.

[181] M. Cygorek et al., "Simulation of open quantum systems by automated compression of arbitrary environments", *Nat. Phys.* **18** (June, 2022) 662–668, doi: 10.1038/s41567-022-01544-9.

[182] L. Del Re, B. Rost, A. F. Kemper, and J. K. Freericks, "Driven-dissipative quantum mechanics on a lattice: Simulating a fermionic reservoir on a quantum computer", *Phys. Rev. B* **102** (September, 2020) 125112, doi: 10.1103/PhysRevB.102.125112.

[183] J. D. Guimarães et al., "Noise-Assisted Digital Quantum Simulation of Open Systems Using Partial Probabilistic Error Cancellation", *PRX Quantum* **4** (November, 2023) 040329, doi: 10.1103/PRXQuantum.4.040329.

[184] I. J. David, I. Sinayskiy, and F. Petruccione, "Digital simulation of convex mixtures of Markovian and non-Markovian single qubit Pauli channels on NISQ devices", *EPJ Quantum Technology* **11** (February, 2024) doi: 10.1140/epjqt/s40507-024-00224-2.

[185] Z. Hu, R. Xia, and S. Kais, "A quantum algorithm for evolving open quantum dynamics on quantum computing devices", *Sci Rep* **10** (February, 2020) 3301, doi: 10.1038/s41598-020-60321-x.

[186] B. Dive, F. Mintert, and D. Burgarth, "Quantum simulations of dissipative dynamics: Time dependence instead of size", *Phys. Rev. A* **92** (September, 2015) 032111, doi: 10.1103/PhysRevA.92.032111.

[187] G. García-Pérez, M. A. C. Rossi, and S. Maniscalco, "IBM Q Experience as a versatile experimental testbed for simulating open quantum systems", *npj Quantum Inf* **6** (January, 2020) 1–10, doi: 10.1038/s41534-019-0235-y.

[188] S. Endo et al., "Variational Quantum Simulation of General Processes", *Phys. Rev. Lett.* **125** (June, 2020) 010501, doi: 10.1103/PhysRevLett.125.010501.

[189] K. Jacobs, "Quantum Measurement Theory and its Applications". Cambridge University Press, Cambridge, 2014. doi: 10.1017/CBO9781139179027, ISBN 978-1-107-02548-6.

[190] H. F. Trotter, "On the product of semi-groups of operators", *Proceedings of the American Mathematical Society* **10** (1959), no. 4, 545–551, doi: 10.1090/S0002-9939-1959-0108732-6.

[191] N. Hatano and M. Suzuki, "Finding Exponential Product Formulas of Higher Orders", in *Quantum Annealing and Other Optimization Methods*, A. Das and B. K. Chakrabarti, eds., Lecture Notes in Physics, pp. 37–68. Springer, Berlin, Heidelberg, 2005. doi: 10.1007/11526216_2.

[192] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, "Efficient Quantum Algorithms for Simulating Sparse Hamiltonians", *Commun. Math. Phys.* **270** (March, 2007) 359–371, doi: 10.1007/s00220-006-0150-x.

[193] T. F. Havel, "Robust procedures for converting among Lindblad, Kraus and matrix representations of quantum dynamical semigroups", *Journal of Mathematical Physics* **44** (February, 2003) 534–557, doi: 10.1063/1.1518555.

[194] E. Andersson, J. D. Cresser, and M. J. W. Hall, "Finding the Kraus decomposition from a master equation and vice versa", *Journal of Modern Optics* **54** (August, 2007) 1695–1716, doi: 10.1080/09500340701352581.

[195] Y.-x. Liu, S. K. Özdemir, A. Miranowicz, and N. Imoto, "Kraus representation of a damped harmonic oscillator and its application", *Phys. Rev. A* **70** (October, 2004) 042308, doi: 10.1103/PhysRevA.70.042308.

[196] M.-D. Choi, "Completely positive linear maps on complex matrices", *Linear Algebra and its Applications* **10** (June, 1975) 285–290, doi: 10.1016/0024-3795(75)90075-0.

[197] L. Dupays and J.-C. Pain, "Closed forms of the Zassenhaus formula", *J. Phys. A: Math. Theor.* **56** (June, 2023) 255202, doi: 10.1088/1751-8121/acc8a3.

[198] F. Casas, A. Murua, and M. Nadinic, "Efficient computation of the Zassenhaus formula", *Computer Physics Communications* **183** (November, 2012) 2386–2391, doi: 10.1016/j.cpc.2012.06.006.

[199] M. E. Peskin and D. V. Schroeder, "An Introduction To Quantum Field Theory". Avalon Publishing, October, 1995. ISBN 978-0-8133-4543-7.

[200] M. E. Muldoon, "Generalized hyperbolic functions, circulant matrices and functional equations", *Linear Algebra and its Applications* **406** (September, 2005) 272–284, doi: 10.1016/j.laa.2005.04.011.

[201] W. F. Stinespring, "Positive functions on ??*-algebras", in *Proceedings of the American Mathematical Society*, volume 6, pp. 211–216. 1955. doi: 10.1090/S0002-9939-1955-0069403-4.

[202] J. J. Schäffer, "On Unitary Dilations of Contractions", *Proceedings of the American Mathematical Society* **6** (1955), no. 2, 322–322, doi: 10.2307/2032368.

[203] V. V. Shende, S. S. Bullock, and I. L. Markov, "Synthesis of Quantum Logic Circuits", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **25** (June, 2006) 1000–1010, doi: 10.1109/TCAD.2005.855930.

[204] J. Johnson and M. Puschel, "In search of the optimal Walsh-Hadamard transform", in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 6, pp. 3347–3350. IEEE, Istanbul, Turkey, 2000. doi: 10.1109/ICASSP.2000.860117.

[205] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental realization of any discrete unitary operator", *Phys. Rev. Lett.* **73** (July, 1994) 58–61, doi: 10.1103/PhysRevLett.73.58.

[206] F. Barratt et al., "Parallel quantum simulation of large systems on small NISQ computers", *npj Quantum Inf* **7** (May, 2021) 1–7, doi: 10.1038/s41534-021-00420-3.

[207] D. Aharonov, D. Gottesman, S. Irani, and J. Kempe, "The Power of Quantum Systems on a Line", *Commun. Math. Phys.* **287** (April, 2009) 41–65, doi: 10.1007/s00220-008-0710-3.

[208] M. G. Davis et al., "Towards Optimal Topology Aware Quantum Circuit Synthesis", in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 223–234. IEEE, Denver, CO, USA, October, 2020. doi: 10.1109/QCE49297.2020.00036.

[209] R. Vale et al., "Decomposition of Multi-controlled Special Unitary Single-Qubit Gates", February, 2023. http://arxiv.org/abs/2302.06377.

[210] S. S. Bullock and I. L. Markov, "Smaller Circuits for Arbitrary n-qubit Diagonal Computations", August, 2003. doi: 10.48550/arXiv.quant-ph/0303039, http://arxiv.org/abs/quant-ph/0303039.

[211] R. Harper, W. Yu, and S. T. Flammia, "Fast Estimation of Sparse Quantum Noise", *PRX Quantum* **2** (February, 2021) 010322, doi: 10.1103/PRXQuantum.2.010322.

[212] S. Chen et al., "The learnability of Pauli noise", *Nat Commun* **14** (January, 2023) 52, doi: 10.1038/s41467-022-35759-4.

[213] S. Bhattacharya, A. Misra, C. Mukhopadhyay, and A. K. Pati, "Exact master equation for a spin interacting with a spin bath: Non-Markovianity and negative entropy production rate", *Physical Review A* **95** (January, 2017) 012122, doi: 10.1103/PhysRevA.95.012122.

[214] T. Giurgica-Tiron et al., "Digital zero noise extrapolation for quantum error mitigation", in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 306–316. October, 2020. doi: 10.1109/QCE49297.2020.00045.

[215] C. Fabre and N. Treps, "Modes and states in quantum optics", *Reviews of Modern Physics* **92** (September, 2020) 035005, doi: 10.1103/RevModPhys.92.035005.

[216] M. Mirhosseini et al., "High-dimensional quantum cryptography with twisted light", *New J. Phys.* **17** (March, 2015) 033033, doi: 10.1088/1367-2630/17/3/033033.

[217] M. H. Michael et al., "New Class of Quantum Error-Correcting Codes for a Bosonic Mode", *Physical Review X* **6** (July, 2016) 031006, doi: 10.1103/PhysRevX.6.031006.

[218] X. Li et al., "A class of efficient quantum incrementer gates for quantum circuit synthesis", *Int. J. Mod. Phys. B* **28** (January, 2014) 1350191, doi: 10.1142/S0217979213501919.

[219] N. Bogoliubov, "On the theory of superfluidity", *J. Phys* **11** (1947), no. 1, 23.

[220] R. Cleve and C. Wang, "Efficient Quantum Algorithms for Simulating Lindblad Evolution", in *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, I. Chatzigiannakis, P. Indyk, F. Kuhn, and A. Muscholl, eds., volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 17:1–17:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2017. ISSN: 1868-8969. doi: 10.4230/LIPIcs.ICALP.2017.17.

[221] J. Han et al., "Experimental Simulation of Open Quantum System Dynamics via Trotterization", *Physical Review Letters* **127** (July, 2021) doi: 10.1103/physrevlett.127.020504.

[222] H. Chen, N. Gomes, S. Niu, and W. A. d. Jong, "Adaptive variational simulation for open quantum systems", *Quantum* **8** (February, 2024) 1252, doi: 10.22331/q-2024-02-13-1252.

[223] J. Mellak, E. Arrigoni, and W. v. d. Linden, "Deep neural networks as variational solutions for correlated open quantum systems", *Communications Physics* **7** (2024) 1–8.

[224] H. J. Carmichael, "Quantum trajectory theory for cascaded open systems", *Phys. Rev. Lett.* **70** (April, 1993) 2273–2276, doi: 10.1103/PhysRevLett.70.2273.

[225] M. Cech, I. Lesanovsky, and F. Carollo, "Thermodynamics of Quantum Trajectories on a Quantum Computer", *Phys. Rev. Lett.* **131** (September, 2023) 120401, doi: 10.1103/PhysRevLett.131.120401.

[226] Z. Hu, R. Xia, and S. Kais, "A quantum algorithm for evolving open quantum dynamics on quantum computing devices", *Scientific Reports* **10** (2019).

[227] C. W. Kim, J. M. Nichol, A. N. Jordan, and I. Franco, "Analog Quantum Simulation of the Dynamics of Open Quantum Systems with Quantum Dots and Microelectronic Circuits", *PRX Quantum* **3** (October, 2022) 040308, doi: 10.1103/PRXQuantum.3.040308.

[228] N. P. D. Sawaya et al., "Resource-efficient digital quantum simulation of d-level systems for photonic, vibrational, and spin-s Hamiltonians", *npj Quantum Information* **6** (June, 2020) 1–13, doi: 10.1038/s41534-020-0278-0.

[229] C. Burdine and E. P. Blair, "Trotterless Simulation of Open Quantum Systems for NISQ Quantum Devices", *Advanced Quantum Technologies* (October, 2024) 2400240, doi: 10.1002/qute.202400240.

[230] D. Camps and R. Van Beeumen, "FABLE: Fast Approximate Quantum Circuits for Block-Encodings", in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 104–113. September, 2022. doi: 10.1109/QCE53715.2022.00029.

[231] C. Zheng, "Universal quantum simulation of single-qubit nonunitary operators using duality quantum algorithm", *Scientific Reports* **11** (February, 2021) 3960, doi: 10.1038/s41598-021-83521-5.

[232] A. M. Childs and N. Wiebe, "Hamiltonian Simulation Using Linear Combinations of Unitary Operations", doi: 10.48550/ARXIV.1202.5822.

[233] T. Cai et al., "Optimal large-scale quantum state tomography with Pauli measurements", *The Annals of Statistics* **44** (April, 2016) 682–712, doi: 10.1214/15-AOS1382.

[234] J. Schwinger, "On Angular Momentum", Technical Report NYO-3071, Harvard Univ., Cambridge, MA (United States); Nuclear Development Associates, Inc. (US), January, 1952. doi: 10.2172/4389568.

[235] D. Bolmatov et al., "Unified phonon-based approach to the thermodynamics of solid, liquid and gas states", *Annals of Physics* **363** (December, 2015) 221–242, doi: 10.1016/j.aop.2015.09.018.

[236] C. Vendromin and M. M. Dignam, "Simple way to incorporate loss when modeling multimode-entangled-state generation", *Phys. Rev. A* **105** (June, 2022) 063707, doi: 10.1103/PhysRevA.105.063707.

[237] H. Seifoory, S. Doutre, M. Dignam, and J. Sipe, "Squeezed thermal states: the result of parametric down conversion in lossy cavities", *Journal of the Optical Society of America B* **34** (July, 2017) 1587, doi: 10.1364/JOSAB.34.001587.

[238] M. Parto et al., "Non-Abelian effects in dissipative photonic topological lattices", *Nature Communications* **14** (March, 2023) doi: 10.1038/s41467-023-37065-z.

[239] T. J. Sturges et al., "Quantum simulations with multiphoton Fock states", *npj Quantum Information* **7** (June, 2021) doi: 10.1038/s41534-021-00427-w.

[240] I. F. Araujo, D. K. Park, F. Petruccione, and A. J. da Silva, "A divide-and-conquer algorithm for quantum state preparation", *Scientific Reports* **11** (March, 2021) 6329, doi: 10.1038/s41598-021-85474-1.

[241] "IonQ Harmony", July, 2024. Publication Title: https://ionq.com/quantum-systems/harmony. `https://ionq.com/quantum-systems/harmony`.

[242] "Quantinuum H1-1", September, 2024. Publication Title: https://quantinuum.com/. `https://quantinuum.com/`.

[243] K. Wright et al., "Benchmarking an 11-qubit quantum computer", *Nature Communications* **10** (November, 2019) 5464, doi: 10.1038/s41467-019-13534-2.

[244] A. Javadi-Abhari et al., "Quantum computing with Qiskit", June, 2024. doi: 10.48550/arXiv.2405.08810, `http://arxiv.org/abs/2405.08810`.

[245] S. Sivarajah et al., "t|ket?: a retargetable compiler for NISQ devices", *Quantum Science and Technology* **6** (November, 2020) 014003, doi: 10.1088/2058-9565/ab8e92.

[246] B. Laikhtman and E. L. Altshuler, "Quasiclassical Theory of Shubnikov-de Haas Effect in 2D Electron Gas", *Annals of Physics* **232** (June, 1994) 332–374, doi: 10.1006/aphy.1994.1056.

[247] Y. Hatsugai, "Chern number and edge states in the integer quantum Hall effect", *Physical Review Letters* **71** (November, 1993) 3697–3700, doi: 10.1103/PhysRevLett.71.3697.

[248] J. Sun et al., "Mitigating Realistic Noise in Practical Noisy Intermediate-Scale Quantum Devices", *Physical Review Applied* **15** (March, 2021) 034026, doi: 10.1103/PhysRevApplied.15.034026.

[249] Z. Cai et al., "Quantum error mitigation", *Reviews of Modern Physics* **95** (December, 2023) 045005, doi: 10.1103/RevModPhys.95.045005.

[250] A. Mandilara, E. Karpov, and N. J. Cerf, "Extending Hudson's theorem to mixed quantum states", *Physical Review A* **79** (June, 2009) 062302, doi: 10.1103/PhysRevA.79.062302.

[251] M. S. Bartlett and J. E. Moyal, "The exact transition probabilities of quantum-mechanical oscillators calculated by the phase-space method", *Mathematical Proceedings of the Cambridge Philosophical Society* **45** (October, 1949) 545–553, doi: 10.1017/S030500410002524X.

[252] P. T. Cochrane, G. J. Milburn, and W. J. Munro, "Macroscopically distinct quantum-superposition states as a bosonic code for amplitude damping", *Physical Review A* **59** (April, 1999) 2631–2634, doi: 10.1103/PhysRevA.59.2631.

[253] A. W. Schlimgen et al., "Quantum simulation of the Lindblad equation using a unitary decomposition of operators", *Phys. Rev. Res.* **4** (June, 2022) 023216, doi: 10.1103/PhysRevResearch.4.023216.

[254] Y. Guo and X. Gao, "Quantum Simulation of Open Quantum Dynamics via Non-Markovian Quantum State Diffusion", 2024. _eprint: 2404.10655. `https://arxiv.org/abs/2404.10655`.

[255] C. Burdine, 2024. `https://github.com/cburdine/nisq-open-quantum-systems`.