Experimental Quantum Kernels in NMR Applied to Machine Learning with Classical and Quantum Data

A Thesis

submitted to

Indian Institute of Science Education and Research Pune

in partial fulfilment of the requirements for the

BS-MS Dual Degree Programme

by

Vivek Sabarad

Indian Institute of Science Education and Research Pune

Dr. Homi Bhabha Road,

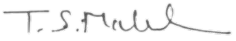
Pashan, Pune 411008, INDIA.

April, 2025

Supervisor: T. S. Mahesh

Certificate

This is to certify that this dissertation entitled Experimental Quantum Kernels in NMR Applied to Machine Learning with Classical and Quantum Data towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Vivek Sabarad at Indian Institute of Science Education and Research under the supervision of T. S. Mahesh, Professor, Department of Physics , during the academic year 2020-2025.



T. S. Mahesh

Sreejith G. J.

Vivek Sabarad

Committee:

T. S. Mahesh

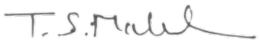
Sreejith G. J.

To the eternal Self whose nature is freedom and expression is love.

Declaration

I hereby declare that the matter embodied in the report entitled Experimental Quantum Kernels in NMR Applied to Machine Learning with Classical and Quantum Data are the results of the work carried out by me at the Department of Physics, Indian Institute of Science Education and Research, Pune, under the supervision of T. S. Mahesh and the same has not been submitted elsewhere for any other degree.

Vivek Sabarad



T. S. Mahesh

Acknowledgments

I would like to express my sincere gratitude to my thesis supervisor, Prof. T. S. Mahesh, for his unwavering guidance, patience, and support. His insightful observations and generous encouragement have shaped my approach to both research and problem-solving, and I am truly thankful for the intellectual freedom he granted throughout this work. I also extend my appreciation to my Thesis Expert, Prof. Sreejith G. J., whose expertise and valuable feedback have strengthened my research.

To my parents, Gurunath and Chitra Sabarad, I owe immeasurable thanks for their constant encouragement and emotional fortitude. Their unwavering faith in my abilities and compassionate presence during periods of uncertainty provided the foundation of resilience that sustained this journey. They have been instrumental in developing my strong self-confidence which stands as my key asset today and has played a pivotal role in making this thesis possible.

My heartfelt thanks go to Prof. Sachin Jain, whose mentorship was instrumental in cultivating my passion and knack for research. His guidance, spanning several semester projects and beyond, made an enduring impact on my academic journey.

I would also like to acknowledge all the PhD scholars in our lab, especially Vishal and Arijit, for their insightful discussions, constructive critiques, and willingness to share knowledge at every step. Their ever-helpful nature and positivity cultivated a collaborative environment that made my work both productive and enjoyable. I am equally grateful to the lab techni-

cians, who generously helped me learn to operate the spectrometer, always taking the time to ensure I understood the subtleties of each process.

My friends have been a constant source of motivation and camaraderie. Saransh has not only been a great friend and well-wisher, but also an inspiration—he consistently reminded me to pick the right path over the easy one. Shreya deserves special thanks for her role in shaping who I am today and supporting my emotional development. Naman, Satyam and Akilan have been an anchor through our deep and meaningful conversations, offering strong moral support when I needed it most. Special mentions go to Vijayalaxmi, Bhavik, Karthika, Rinku, and Asmi for the laughter, companionship, and constant encouragement.

A special word of gratitude goes to Keshav for his critical comments, ideas, and suggestions, which greatly influenced my project work. His thoughtful discussions kept me on track and challenged me to refine my thinking. I would also like to thank ChatGPT for providing valuable suggestions and assistance in shaping parts of the text. While my work reflects my own research and understanding, the tool's guidance smoothed out details and improved clarity.

I acknowledge the financial support from the Department of Science and Technology (DST), Government of India, through the INSPIRE Scholarship program, which significantly facilitated my academic pursuits and enabled me to undertake this research.

Finally, I thank everyone else who has helped me, in ways big or small, to reach this milestone. This thesis is as much a testament to their support as it is to my individual effort.

Abstract

Kernel methods enable the learning of nonlinear functions by mapping data into high-dimensional spaces, where linear techniques can then be applied effectively. In quantum kernel methods, classical data is encoded into quantum states, thereby leveraging the exponentially large Hilbert space available in quantum systems. This thesis implements quantum kernel methods using nuclear magnetic resonance (NMR) as a platform to control and measure nuclear spin systems. In this work, classical data is encoded through tailored pulse sequences that generate multiple quantum coherences, first in solid-state, followed by liquid-state NMR setups. We demonstrate the effectiveness of the resulting quantum kernels by applying them to standard machine learning tasks such as one-dimensional regression using a kernel ridge regression model and two-dimensional classification using Support Vector Machines (SVMs). In addition, we extend the method to process quantum data directly. For this, we develop a protocol to compute quantum kernels for unparameterized operator inputs and present experimental results for the classification of quantum operators based on their entangling properties. Overall, our results confirm that quantum kernels derived from NMR quantum systems can be successfully used for machine learning tasks involving both classical and quantum data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Quantum information processing and machine learning are two of the most rapidly evolving fields in recent years. They have opened up exciting new frontiers in computation, with the potential to revolutionize various aspects of technology. This thesis is motivated by the promise of quantum-enhanced techniques to achieve results and performances beyond the reach of classical approaches. The ability of algorithms with quantum core to handle quantum data which is not possible with classical algorithms is one such are where quantum advantage can be explicitly demonstrated. The central theme of this work is to explore the intersection of these two fields, focusing on quantum kernel methods implemented on nuclear magnetic resonance (NMR) quantum processors and their ability to process both classical and quantum data.

At its core, the thesis explores ways to embed classical data into the operator space of a quantum system using tailored feature maps that exploit multiple quantum coherences—a resource naturally available in nuclear spin systems that was first demonstrated in the context of solid-state NMR [1]. Such mappings are not only computationally intriguing but also pave the way for more efficient learning algorithms, as evidenced by recent advances in both classical kernel methods and their quantum counterparts.

The significance of this research is twofold. First, it provides a concrete *experimental platform*—based on NMR quantum processors—to implement and validate theoretical ideas from quantum machine learning. Second, by focusing on less-explored quantum kernel techniques that leverage multiple quantum coherences, this work uncovers novel pathways for encoding *quantum data* for machine learning. This approach may ultimately enable quantum processors to handle complex learning tasks more efficiently than classical systems, marking an important step toward realizing practical quantum advantage in machine learning.

In the chapters that follow, we briefly review the key concepts in quantum information processing and NMR quantum computing, followed by a discussion of quantum machine learning to provide a context and motivation of this work.

1.1 Quantum Information

A formalism and set of tools to describe information stored and processed in quantum systems is what what we refer to as quantum information science. Here we cover the basics of quantum information theory as described in Nielsen and Chuang's seminal textbook [2]. The fundamental unit of information, in classical information theory, is the bit. It can take values 0 or 1. In quantum information, this role is played by the quantum bit or *qubit*. It is essentially a two-level quantum system that can exist in a superposition of its basis states. The basis states can be represented as $|0\rangle$ and $|1\rangle$ and a general pure state of a single qubit can be written as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\ , \tag{1.1}$$

where α and β are complex coefficients and represent probability amplitudes. These satisfy the normalization given by $|\alpha|^2 + |\beta|^2 = 1$. It is this nature of superposition that allows the qubit to exist in a continuum of states between $|0\rangle$ and $|1\rangle$ and encode a continuum of corresponding values, a key resource in quantum computing.

Multiple-qubit systems are represented within the tensor product of their constituent single-qubit Hilbert spaces. Specifically, a two-qubit system resides in a four-dimensional Hilbert space with basis states given by $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. Crucially, multi-qubit systems can exhibit *entanglement* [2], a form of non-classical correlation with no analog in classical information. An entangled pair of qubits cannot be described as having independent states; instead, only the joint state is well-defined. A well-known instance is the Bell state for two qubits, defined as $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. This entangled state exhibits the property that performing a measurement on one qubit instantly fixes the outcome for the other, independent of the physical distance separating them [2].

In quantum computing, information processing is achieved through quantum logic gates, which act as reversible, unitary operations transforming the state of the qubits. For instance, single-qubit gates like the Pauli-X (which flips $|0\rangle \leftrightarrow |1\rangle$), Pauli-$Z$ (which phases $|1\rangle$ by -1 relative to $|0\rangle$), and the Hadamard gate (which creates superposition) are analogous to classical logic operations but act on quantum amplitudes. Entanglement between qubits is generated using multi-qubit gates like the controlled-NOT (CNOT) gate, which conditionally inverts the state of a target qubit based on the state of a control qubit. Through appropriate sequences of such gates, a quantum computer can perform complex computations.

Another essential component is *measurement*. When a qubit is measured, the result is a classical bit value (such as 0 or 1), and the qubit's state collapses onto the corresponding computational basis state. The outcome probabilities are given by the squared amplitudes (Born rule). In contrast to a classical bit, a qubit generally collapses upon measurement, losing any superposition or entanglement with other qubits. This inherent balance between acquiring information and causing state disturbance represents a core principle in quantum information theory.

In practical quantum information processing, especially on experimental platforms, not all states are perfectly isolated pure states. We often encounter *mixed states*, which are statistical ensembles of pure states. A practical depiction of a quantum state, whether pure

or mixed, is provided by the density operator ρ. For a pure state $|\psi\rangle$, it is defined as $\rho = |\psi\rangle\langle\psi|$, while for a statistical ensemble $\{p_i, |\psi_i\rangle\}$, it is given by $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$. This density matrix framework is especially advantageous in contexts such as NMR, where experiments involve large groups of spins rather than isolated quantum entities [3].

Quantum information science has motivated powerful algorithms (e.g., Shor's factoring algorithm, Grover's search) that theoretically outperform their classical counterparts [2, 4]. However, implementing these algorithms requires precisely controlled qubit registers and long coherence times. Various physical platforms have been explored for realizing qubits, including trapped ions, superconducting circuits, and nuclear spins in molecules. In this thesis, our attention is centered on nuclear spin qubits within an NMR system. Prior to examining how such systems can be leveraged for machine learning applications, we first introduce the fundamentals of quantum information processing as implemented in NMR.

1.2 NMR Quantum Information Processing

Nuclear Magnetic Resonance (NMR) has emerged as a pivotal platform for quantum information processing since the late 1990s [6, 7]. In NMR quantum computing, qubits are typically encoded in nuclear spin-$\frac{1}{2}$ particles, such as those of ^{1}H, ^{13}C, or ^{31}P nuclei in molecules. An ensemble of molecules is held in a strong static magnetic field, which defines a common quantization axis (conventionally the z-axis). Each nuclear spin experiences a Zeeman interaction with the field, causing its two spin states ($m_z = +\frac{1}{2}$ or $-\frac{1}{2}$) to have different energies. The energy level separation for a nuclear spin μ is given by $\hbar\omega_\mu$, where the Larmor frequency $\omega_\mu = -\gamma_\mu B_0$ depends linearly on the nuclear gyromagnetic ratio γ_μ and the strength of the applied magnetic field B_0. The single-spin Hamiltonian is

$$H_{\text{Zeeman}} = -\hbar\omega_\mu I_{z,\mu}\,, \tag{1.2}$$

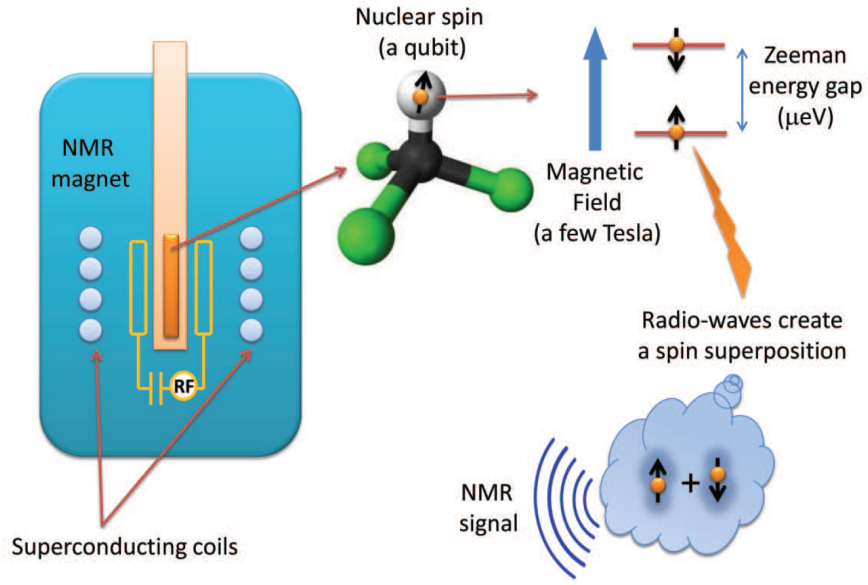


Figure 1.1: Schematic of an NMR quantum information processing setup. (Reproduced from Ref. [5] with permission from Springer Nature.)

where $I_{z,\mu}$ is the spin angular momentum operator in the z-axis for the μ^{th} spin. For a collection of n nuclear spins, the Zeeman Hamiltonian is expressed as $H_{\text{Zeeman}} = -\hbar \sum_{\mu=1}^{n} \omega_\mu I_{z,\mu}$. At thermal equilibrium and temperature T, the state of the spin ensemble is characterized by the Boltzmann density operator

$$\rho_{\text{th}} = \frac{1}{Z} e^{-\frac{H}{k_B T}} \approx \frac{1}{2^n}\left(I + \epsilon \sum_{\mu=1}^{n} I_{z,\mu}\right), \tag{1.3}$$

where $Z = \text{Tr}(e^{-H/k_B T})$ denotes the partition function, k_B is Boltzmann's constant, and $\epsilon = \frac{\hbar\omega}{k_B T} \ll 1$ is the thermal polarization factor, typically on the order of 10^{-5} at room temperature under standard NMR conditions [8]. The approximation in Eq. (1.3) arises from the high-temperature limit, where $k_B T \gg \hbar\omega$. In this regime, the thermal state is close to the maximally mixed state $I/2^n$, with a small polarization bias along the $+z$ axis for each individual spin.

In the context of NMR, the qubit states $|0\rangle$ and $|1\rangle$ are associated with the spin eigenstates $|\uparrow\rangle$ and $|\downarrow\rangle$, corresponding to magnetic quantum numbers $m_z = +1/2$ (spin-up) and

$m_z = -1/2$ (spin-down), respectively. Logical operations in an NMR quantum computer are performed using resonant radio-frequency (RF) pulses, which induce spin rotations to realize single-qubit gates. Two-qubit gates are achieved by harnessing the intrinsic interactions between coupled nuclear spins [3]. NMR systems are inherently ensemble quantum computers: rather than a single molecule, we have roughly 10^{18} identical copies undergoing the same operations. As a result, the measurement in NMR does not collapse a single quantum state, but instead yields an ensemble-averaged signal. Fig. 1.1 shows a schematic of an NMR quantum information processing setup.

1.2.1 Acquisition and Measurement in NMR

In NMR, readout is accomplished through inductive detection of the transverse magnetization of the ensemble. After manipulating the spins, one typically allows the spins to precess in the transverse (xy) plane, generating an oscillating voltage in a detection coil. The measured signal, referred to as the free induction decay (FID), arises from the collective coherent transverse magnetization of the spin ensemble [8]. Applying a Fourier transform to the FID yields the NMR spectrum, which exhibits resonance peaks corresponding to distinct nuclear spins identified by their respective Larmor frequencies and their associated coherences. The magnitude and phase of these spectral peaks reveal details about the populations and coherences of the underlying spin states. In quantum information terms, the measurement in liquid-state NMR is typically a collective measurement of the average magnetization, which can be related to expectation values of spin operators [3]. Through careful calibration and pulse sequences, one can infer the outcome of computational basis measurements (i.e., the effective populations of $|0\rangle$ and $|1\rangle$ states for each qubit) from the NMR signal.

Importantly, the observed FID is also shaped by two intrinsic relaxation processes. The T_1 relaxation time, known as spin-lattice relaxation, describes the process by which the longitudinal magnetization returns to equilibrium through energy exchange with the surrounding environment. In contrast, the T_2 relaxation time, or spin-spin relaxation, refers to the decay

of phase coherence among spins in the transverse plane due to mutual interactions. In most FID measurements, the decay of the signal is predominantly governed by T_2, which limits the duration over which the coherent oscillations can be observed and accurately transformed into the frequency domain. In liquid-state NMR, T_2 relaxation times can reach durations on the order of seconds, enabling extended coherence and the implementation of high-fidelity quantum operations [3]. Conversely, in solid-state NMR, T_2 times are generally shorter due to pronounced dipolar couplings, which accelerate the dephasing of spin states [9].

1.2.2 Spin–Spin Interactions: J-Coupling and Dipolar Coupling

Beyond the Zeeman splitting, qubits in an NMR quantum processor interact with one another via two primary mechanisms: scalar J-coupling and dipole-dipole coupling[8]. The *J-coupling*, also referred to as scalar or indirect coupling, is an interaction transmitted through chemical bonds, mediated by the electrons shared between neighboring nuclei. It is isotropic and typically described (in a two-spin system) by a Hamiltonian $H_J = 2\pi\hbar J\, \mathbf{I}_1 \cdot \mathbf{I}_2 = 2\pi\hbar J(I_{1x}I_{2x} + I_{1y}I_{2y} + I_{1z}I_{2z})$. In the presence of a strong magnetic field, the components of the interaction Hamiltonian that do not commute with the Zeeman term—known as non-secular terms—are effectively averaged out. Consequently, the primary form of spin-spin interaction in liquids is well-approximated by $H_J \approx 2\pi\hbar J\, I_{1z}I_{2z}$ [8]. The scalar coupling constant J, measured in hertz, varies depending on the nature of the chemical bonding and typically spans a range from a few hertz to several hundred hertz. In liquid-state NMR quantum computing, J-couplings between qubits are weak but nonzero, and they serve as the resource for implementing two-qubit gate operations (for example, a controlled-NOT can be realized through sequences that refocus unwanted single-spin evolutions and let the I_zI_z coupling generate entangling interactions) [10].

The *dipole-dipole coupling* arises from a direct through-space interaction between the magnetic dipole moments of two spins. In the regime of a strong external magnetic field B_0, the

relevant secular component of the dipolar Hamiltonian for spins i and j is expressed as [8]

$$H_{dd}^{(i,j)} = d_{ij}\Big(2I_{iz}I_{jz} - I_{ix}I_{jx} - I_{iy}I_{jy}\Big)\,, \tag{1.4}$$

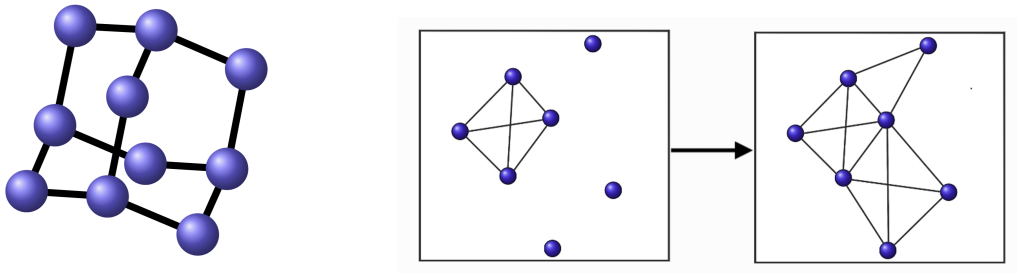
where the coupling strength

$$d_{ij} = \frac{\mu_0}{4\pi}\frac{\gamma_i\gamma_j\hbar^2}{r_{ij}^3}(1 - 3\cos^2\theta_{ij})\,, \tag{1.5}$$

It depends on the inter-spin distance r_{ij} and the angle θ_{ij} between the internuclear vector $\mathbf{r}_{ij}$ and the direction of the external magnetic field [8]. Unlike J-couplings, dipolar couplings do not require a chemical bond and can be much stronger (kilohertz range or more for spins that are nearby). However, in a liquid the rapid molecular tumbling averages d_{ij} to zero (because the angular factor averages to zero), effectively eliminating dipolar interactions. In solid samples, dipolar couplings remain active and can induce complex multi-spin dynamics. Dipolar interactions are inherently two-body and can create entanglement among nuclear spins in solid-state NMR systems [11].

1.2.3 Solid-State NMR for Quantum Computing

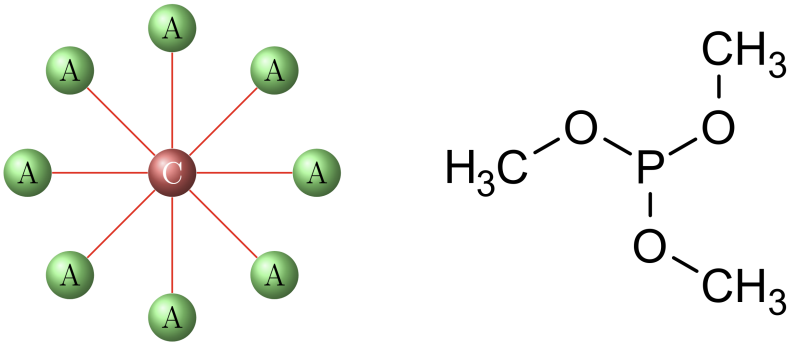
Solid-state NMR offers the advantage of naturally strong couplings between qubits, which can in principle generate entanglement and multi-qubit quantum gates more readily than in liquid-state systems. However, the presence of strong dipolar interactions among many spins also makes control more challenging, as spins do not remain isolated two-level systems. Techniques such as magic-angle spinning and decoupling pulses are often used in NMR to average out or selectively re-introduce interactions [9]. For quantum information processing, one approach is to use carefully designed pulse sequences to engineer an effective Hamiltonian that mimics the desired gate or interaction. For example, sequences based on average Hamiltonian theory can turn on or off certain terms of the dipolar Hamiltonian, enabling coherent control over coupled spins [12].



(a) (b)

Figure 1.2: (a) Structure of adamantane. (b) Intermolecular interactions in the crystal lattice that lead to growth of multiple quantum coherences (MQCs). Each molecule is represented by a blue sphere, and the connecting lines indicate the MQC developed between the spins of those molecules.

An interesting model system in solid-state NMR is a so-called *spin network* in a crystal. One example is the molecular plastic crystal **adamantane** ($C_{10}H_{16}$), which has a symmetric structure (see Fig. 1.2). Like other plastic crystals it shows a peculiar behaviour with respect to its orientation and the dipolar interactions between its ^{13}C nuclear spins. Due to its symmetric structure, molecules at every lattice site rapidly tumble in the crystal lattice, averaging out the intra-molecular dipolar interactions. This leads to long enough coherence times ($T_2^* \sim 1$ms) for the nuclear spins to be manipulated using NMR pulses. However, dipolar couplings between spins belonging to different molecules persist and are not averaged out, making them useful for generating multiple quantum coherences (MQCs) [13]. MQCs are collective excitations where multiple spins flip coherently, and their observation in NMR serves as a signature of multi-spin entanglement and correlated dynamics [11]. This ability to access correlated multi-spin states makes solid-state NMR a compelling platform for exploring quantum algorithms that require entanglement among several qubits. Chapter 2 will delve into the theory and experimental results on MQCs and discuss their relevance to quantum information processing.



(a) (b)

Figure 1.3: (a) Structure of the single layered star-topology register. (b) Molecular structure of trimethyl phosphite, where ^{31}P and ^{1}H spins form central (C) and (A) ancillary qubits, respectively.

1.2.4 Liquid-State NMR and Quantum Gates

Most early demonstrations of quantum algorithms with NMR employed liquid-state systems [14]. In liquid-state NMR, molecules are not only orientationally disordered but are positionally disordered too. Therefore they experience averaged Hamiltonian where dipolar couplings vanish and only J-couplings (and chemical shift differences) are significant [15]. Each distinct nuclear spin in a molecule can be treated as a qubit, and they can be individually addressed by frequency-selective RF pulses (since each spin type resonates at a different frequency). This selectivity, combined with precisely calibrated pulse sequences, allows for universal quantum control. Pseudopure states are used to emulate a pure initial state out of the thermal ensemble [6, 7]. Through techniques like temporal averaging or spatial averaging, one prepares an effective $\rho \approx |00\cdots0\rangle\langle00\cdots0|$ state (up to an identity component) from the thermal state.

Beyond the well-known quantum algorithms like Grover's search [16, 17] and Shor's factoring algorithm (demonstrating the factorization of 15) [18], liquid-state NMR has facilitated the implementation of several lesser-known algorithms. For instance, the quantum random walk algorithm has been experimentally realized using a three-qubit NMR system, demonstrating the unique quantum interference patterns absent in classical random walks [19].

Additionally, the quantum pattern recognition algorithm, which combines principles from Hopfield neural networks with adiabatic quantum computation, has been implemented on a two-qubit NMR quantum computer [20]. Another notable example is the realization of the quantum algorithm for solving linear systems of equations, known as the HHL algorithm, on a four-qubit NMR processor [21]. These implementations underscore the versatility of NMR quantum computing in exploring a diverse array of quantum algorithms. Single-qubit rotations in NMR are realized by resonant RF pulses that act as fast rotations about axes in the xy-plane of the Bloch sphere. Two-qubit gates are typically realized by coupling evolutions: for instance, a controlled-phase gate can be implemented by letting a two-spin system evolve under the $2\pi J I_{1z} I_{2z}$ interaction for a specific duration, up to single-qubit refocusing pulses [10]. By combining such operations, a CNOT gate can be achieved. The readout in liquid NMR yields the ensemble averages, from which the output of the algorithm (such as a spectral peak indicating the answer qubit) can be inferred.

Liquid-state NMR QIP benefited from extremely high quantum gate fidelities and long coherence times (on the order of seconds for T_2 in some systems), as well as the ability to fully characterize the quantum state through tomography [22]. Nonetheless, liquid-state NMR faces an inherent scalability limitation: as the number of qubits n increases, the signal-to-noise ratio diminishes exponentially. This is due to the fact that the deviation of the thermal state ρ_{th} (Eq. (1.3)) from the identity becomes vanishingly small. By about 10–12 qubits, the signal becomes too weak to be useful [23]. Moreover, preparing true entangled pure states is effectively impossible since the system remains mostly in a mixed state; NMR quantum computing operates without ever producing entanglement detectable by the usual measures [24]. Despite these limitations, NMR provided a crucial early platform to test quantum computing concepts and inspired many techniques in quantum control.

Additionally, one can build multi-qubit systems employing molecules with specific structural configurations. One such example is the *star-topology* register, where a central qubit is interactively linked to multiple ancillary qubits, forming a configuration reminiscent of a

star (see Fig. 1.2.4). The central qubit (^{31}P) can be used to mediate interactions between the ancillary qubits (^{1}H), enabling multi-qubit gates which allow generation of entanglement in as many as 20 qubits [25]. Such topological registers are a powerful tool for exploring multi-qubit dynamics and entanglement in NMR systems.

1.3 Quantum Machine Learning

By leveraging the unique properties of quantum mechanics, quantum machine learning offers the potential to boost the efficiency and power of learning algorithms. One prominent technique involves utilizing quantum systems to process data in high-dimensional feature spaces that are otherwise inaccessible to classical computers. Some of the earliest theoretical advancements in quantum machine learning include Harrow *et al.*'s quantum algorithm for solving linear systems of equations (2009) [26], quantum principal component analysis [27], and the quantum support vector machine [28]. Foundational reviews by Biamonte *et al.* [29] and Schuld *et al.* [30] provided comprehensive overviews of these developments, laying the theoretical foundation for the field. More recently, progress has been made in demonstrating quantum-enhanced feature spaces for classification tasks using superconducting hardware [31], as well as in advancing variational quantum algorithms, which have emerged as a prominent framework for near-term QML applications [32]. Moreover, studies on the power of data [33] have clarified conditions for quantum advantage in learning tasks, while recent theoretical work has expanded frameworks beyond kernel methods to uncover new avenues for QML [34]. This thesis focuses on *quantum kernel methods*, an approach that combines the strengths of classical kernel methods with quantum feature maps.

1.3.1 Quantum Kernel Methods

Supervised machine learning tasks often involve finding a function that maps input data to output labels. In linear regression or classification, the function is a linear combination of the input features. However, many real-world problems are nonlinear, and linear models may not capture the underlying structure of the data. Kernel methods offer a way to address this limitation by implicitly mapping the data into a high-dimensional feature space where linear models can be applied effectively. The kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ computes the inner product between the feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ in this high-dimensional space, allowing linear algorithms to operate in a nonlinear feature space without explicitly computing the high-dimensional mapping [35]. In classical machine learning there kernels are basically predefined kernel functions embedded in the model, such as the polynomial or Gaussian kernels, that compute the inner products in feature space [36]. In quantum kernel methods the classical data is encoded into quantum states or operators within a high-dimensional Hilbert space, where kernel functions are computed based on overlaps between these quantum representations. Quantum kernels can effectively represent complex data structures that might be difficult to capture using classical kernels [37]. These kernels are then compatible with classical supervised learning techniques, such as support vector machines [38]. Refer to the Chapter 3 for a detailed theoretical discussion on quantum kernels.

Quantum kernel methods offer several advantages:

- **Efficient feature mapping:** A quantum processor can generate complex feature maps that are difficult to simulate classically. This allows the quantum kernel to represent high-dimensional data correlations efficiently.
- **Hybrid implementation:** In this hybrid framework, the quantum device is responsible for evaluating the kernel—specifically, the inner products between quantum states—while the machine learning model itself, such as SVM or kernel ridge regression, is trained using classical computational resources. This division of tasks aligns

well with the capabilities of present-day quantum hardware.

- **Potential quantum advantage:** There exist theoretical proposals indicating that certain learning tasks that involve "quantum inputs" might be intractable for classical algorithms but can be efficiently handled using quantum kernels [39].

Experimental implementations of quantum kernels have demonstrated their effectiveness for regression and classification tasks, including studies utilizing nuclear spins in solid-state NMR [1] and photonic quantum circuits [40] and other near-term quantum devices [31]. Further research has focused on mitigating errors and optimizing circuit design to maintain kernel magnitudes as systems scale, exemplified by classification of complex cosmological datasets using Google's Sycamore quantum processor [41]. These experimental validations underscore the practicality of quantum kernels for diverse classical and quantum applications, thereby expanding possibilities for advanced quantum computing tasks.

In this work, we experimentally implement quantum kernel methods using a liquid-state NMR quantum register characterized by a star topology [25], by exploiting multiple quantum coherences (MQCs) as a means to access the full Hilbert space of the system. This approach is inspired by work presented by Kusumoto et al. [1]. Our approach surpasses prior methodologies by generalizing the kernel framework to handle quantum data and perform quantum-specific tasks. Initially, we validate our quantum kernel through experiments involving classical machine learning benchmarks. Subsequently, we numerically analyze the extended kernel by applying it to entanglement classification tasks, achieving promising experimental results [42].

1.3.2 Objectives

In this thesis we aim to achieve the following::

- **Design and Implement Quantum Feature Maps:** Designing new protocols for

encoding classical information into the quantum states of nuclear spin systems involves crafting customized pulse sequences. Specifically, our approach focuses on utilizing multiple quantum coherences (MQCs) in both liquid-state and solid-state NMR to implement feature maps capable of capturing higher-order spin correlations.

- **Compute and Evaluate Quantum Kernels:** Establish an experimental framework where the quantum kernel is evaluated by measuring the fidelity between quantum states generated by the feature maps. These quantum-derived kernels will subsequently be integrated into classical machine learning algorithms to carry out specific tasks.

- **Show a Clear Quantum Advantage In Machine Learning With Quantum Data:** Demonstrate that the quantum kernel methods implemented on NMR quantum processors can outperform classical kernel methods on learning tasks that involve quantum inputs. This will involve comparing the performance of quantum kernel models against classical models on specific tasks like entanglement classification.

In summary, this thesis seeks not only to advance the theoretical understanding of quantum kernel methods in terms of design and implementation of quantum feature maps but also to demonstrate their experimental viability by computing them using nuclear spin systems. The insights gained here could have far-reaching implications, potentially leading to new quantum algorithms that transform how we process and analyze data across a wide array of scientific and engineering disciplines.

The subsequent chapters expand upon the foundation provided in this chapter by exploring the generation and detection of multiple quantum coherences, the implementation of quantum feature maps in nuclear spin systems, and the experimental assessment of quantum kernel methods in various learning scenarios. Chapter 2 introduces the theory behind multiple quantum coherences and their relevance to quantum information processing. Chapter 3 delves into the theoretical framework of quantum kernel methods and their application to machine learning tasks. Chapter 4 outlines the experimental protocol for implementing

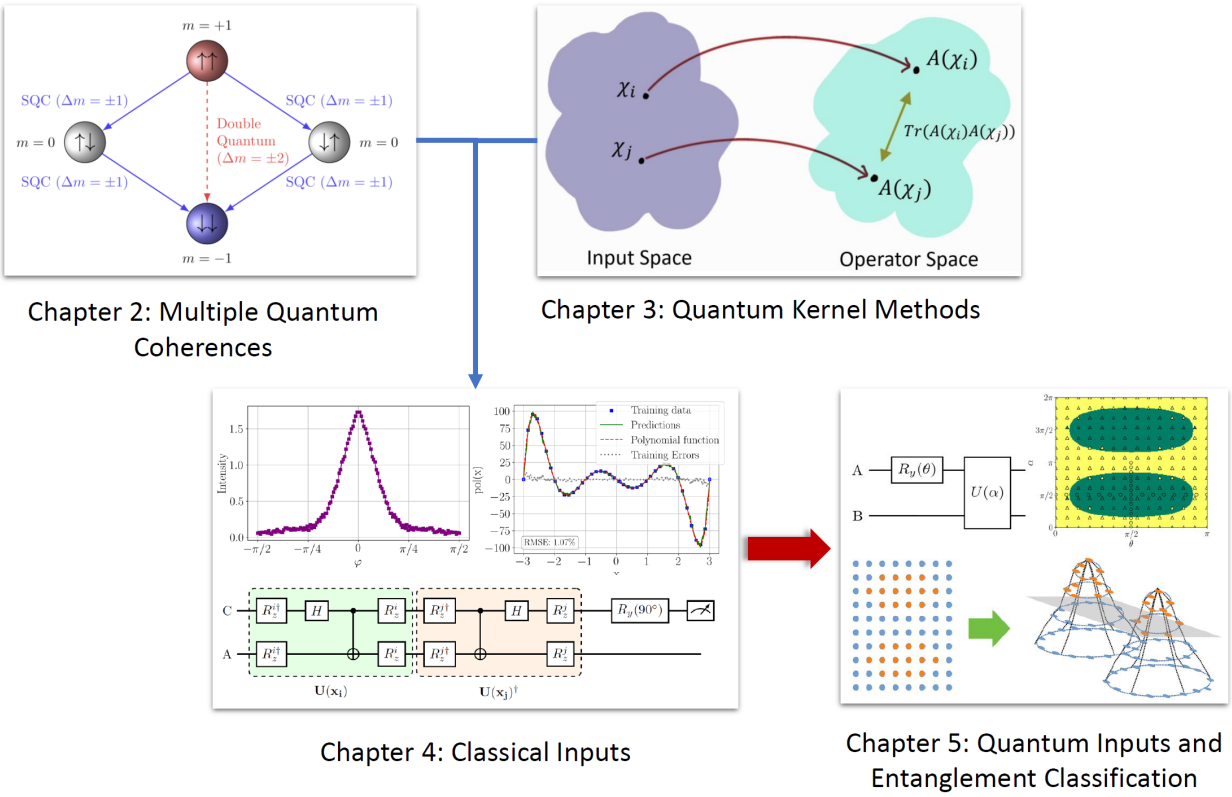


Figure 1.4: Structure of the thesis, highlighting the chapters and their interconnections.

quantum kernels with classical inputs in NMR systems and shows the results of applying the kernel to classical machine learning tasks. Chapter 5 introduces the quantum kernel with quantum inputs and discusses its implementation in NMR systems followed by application of such a kernel for entanglement classification task. See Fig. 1.4 for a visual representation of the thesis structure. Finally, Chapter 6 summarizes the key findings and discusses the implications of our work for quantum machine learning and quantum information processing.

Chapter 2

Multiple Quantum Coherences

2.1 What are Multiple Quantum Coherences (MCQs)?

Multiple quantum coherences (MQCs) in multiple spin-$\frac{1}{2}$ systems basically refer to coherent spin excitations that involve simultaneous flips of multiple spins. In NMR spin-systems that invovle multiple spins interaction with each other, one often observes single-quantum coherences (SQC) in a standard NMR experiment, which correspond to transitions flipping one spin at a time (e.g., $|\uparrow\downarrow\rangle \rightarrow |\downarrow\downarrow\rangle$). However, higher-order, multi-spin processes can occur under certain pulse sequences, giving rise to double-quantum (± 2) or even higher-order ($\pm 3, \pm 4, \dots$) coherences.

These MQCs provide valuable insights into the multi-spin correlation effects that can arise through strong dipole–dipole in solid-state NMR and other strongly coupled spin systems. We intend to utilize these MQCs to map our data onto the full Hilbert space of the quantum systems creating an expressive feature space and further compute the kernel functions corresponding to these maps. Refer to the next chpater for more details on the kernel functions. Additionally, MQCs have recently garnered attention in quantum information because their presence can signal nontrivial quantum correlations, including entanglement

among spins [11].

In this chapter, we present the theoretical framework underlying multiple quantum coherences, highlight their connection with quantum entanglement, describe an experimental protocol for their generation and detection, and report the corresponding experimental findings. We aim to establish the theoretical and practical foundation for utilizing MQCs in NMR quantum computing for quantum machine learning applications.

2.2 Theory of MQCs

This section provides an explanation of multiple quantum coherences from the density matrix viewpoint. We will begin with the notion of coherence order in an N-spin system, then illustrate it via a two-spin density matrix example.

2.2.1 Coherence Orders

Consider a system of N spin-$\frac{1}{2}$ nuclei in a strong static magnetic field $B_0\hat{z}$. Let $I_{z,\text{tot}} = \sum_{j=1}^{N} I_{jz}$ denote the total spin angular momentum operator along the z-axis, measured in units of $\hbar$. For a system of N spin-$\frac{1}{2}$ particles, the corresponding Hilbert space has dimension 2^N, and any density matrix ρ describing the system can be written as a linear combination of tensor product operators.

$$\rho \;=\; \sum_{\alpha} c_\alpha \, \Omega_\alpha, \tag{2.1}$$

where $\{\Omega_\alpha\}$ forms a convenient operator basis (e.g., product operators like $I_{1x}I_{2y}\cdots I_{Nz}$, etc.), and c_α are complex coefficients.

Multiple-quantum coherences are charecterized by the *coherence order*, which classifies the terms in ρ by how they change the total z-magnetization quantum number M_z. Formally,

an operator $O^{(p)}$ has coherence order p if

$$[I_{z,\mathrm{tot}}, O^{(p)}] \;=\; p\,\hbar\, O^{(p)}. \tag{2.2}$$

This means that $O^{(p)}$ connects states whose M_z values differ by p in the computational basis, M_z being the eigenvalue of $I_{z,\mathrm{tot}}$ for a given state.

In NMR, single-quantum operators are observable magnetic dipole transitions (one spin flips between $|\uparrow\rangle$ and $|\downarrow\rangle$). Higher-order coherences such as $p = \pm 2, \pm 3, \ldots$ that indicate simultaneous multi-spin flips are usually weak in standard experiments but can be selectively excited and detected by applying special multi-pulse sequences [13] or other entangling operations (refer to the Sec 2.3.3).

2.2.2 Density Matrix Expansion and Example

To see more concretely what multiple quantum coherences look like in the density matrix, let us consider a simplified two-spin ($N = 2$) example. The total z-magnetization quantum numbers for these basis states are:

$$\begin{aligned}
|00\rangle &: M_z = +\tfrac{1}{2} + \tfrac{1}{2} = +1,\\
|01\rangle &: M_z = +\tfrac{1}{2} - \tfrac{1}{2} = 0,\\
|10\rangle &: M_z = -\tfrac{1}{2} + \tfrac{1}{2} = 0,\\
|11\rangle &: M_z = -\tfrac{1}{2} - \tfrac{1}{2} = -1.
\end{aligned}$$

An operator $O^{(p)}$ of coherence order p changes M_z by p. To systematically see these coherence orders write ρ in the basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. Suppose

$$\rho = \sum_{\alpha,\beta} \rho_{\alpha,\beta}\; |\alpha\rangle\langle\beta|, \tag{2.3}$$

where $\alpha, \beta \in \{00, 01, 10, 11\}$ label the basis states. Each off-diagonal ($\alpha \neq \beta$) element $\rho_{\alpha,\beta}$ corresponds to a transition from $|\beta\rangle$ to $|\alpha\rangle$. To see the associated coherence order p, compute

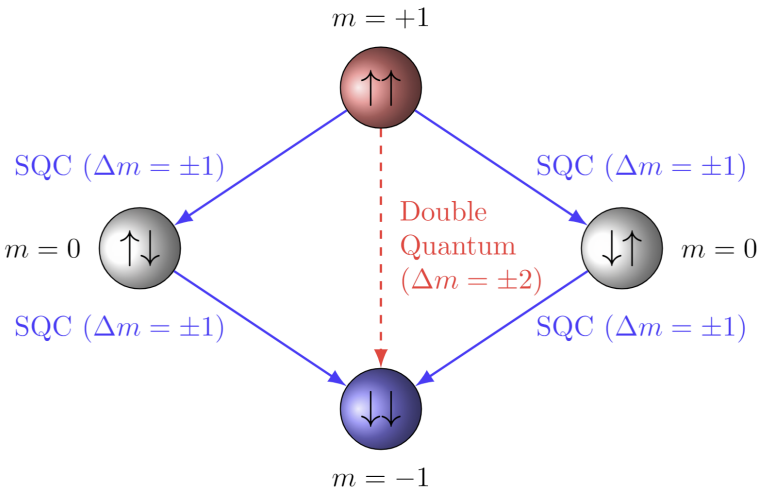


Figure 2.1: Coherence orders in a two-spin system.

the difference in M_z for $|\alpha\rangle$ and $|\beta\rangle$. For instance:

$$\rho_{01,00} \quad \text{connects } |00\rangle(M_z = +1) \text{ to } |01\rangle(M_z = 0),$$

so it has $p = (0) - (+1) = -1$. Likewise, $\rho_{11,00}$ connects $M_z = -1$ to $M_z = +1$, giving $p = -1 - 1 = -2$, and so on (see Fig. 2.1). Hence, each density matrix element $\rho_{\alpha,\beta}$ can be classified into a certain coherence order. Grouping these elements yields a partition of ρ into subspaces $\rho^{(p)}$ of coherence order p:

$$\rho = \sum_{p=-N}^{N} \rho^{(p)},$$

where $\rho^{(p)}$ contains exactly those matrix elements that connect states differing by p in M_z.

Thus, in a two-spin system, observing $p = \pm 2$ coherence indicates simultaneous flipping of both spins (e.g., $|\uparrow\uparrow\rangle \leftrightarrow |\downarrow\downarrow\rangle$). In general, an N-spin system can display coherence orders p ranging from $-N$ to $+N$. The presence of high-order coherences $|p| \gg 1$ implies strong multi-spin correlations, which can be selectively excited by suitable pulse sequences.

2.2.3 MQCs and Entanglement

Observation of multiple quantum coherences provide a unique window into multi-spin correlations, which are also at the heart of quantum entanglement. From the density matrix perspective, large off-diagonal terms connecting states with widely different M_z (e.g., $p = \pm 2, \pm 3, \dots$) mean that the system's wavefunction (or mixed state) has substantial coherence between macroscopically distinct spin configurations. Such coherences often imply nontrivial quantum correlations. In fact, for certain simple spin systems (e.g., a pair of dipolar-coupled spins at low temperature), the intensity of the $p = \pm 2$ coherence can act as an *entanglement witness* [11], revealing the presence of nonseparable states. Past studies have also have explored how the time evolution of MQC intensities correlates with quantitative measures of entanglement, such as concurrence, for various spin pairs [43]. MQCs can also act as robust indicators of multi-qubit entanglement in quantum information settings, such as verifying large Greenberger-Horne-Zeilinger states [44].

Hence, analyzing the density matrix in terms of coherence orders helps us connect multi-spin correlations (which yield higher-order coherences) to concepts like quantum entanglement. In more complex systems, multi-qubit states may develop large $|p|$ MQCs without necessarily implying pairwise entanglement among all spins, but strong MQCs are nonetheless a fingerprint of strongly correlated spin dynamics.

2.3 Methods

A typical MQC NMR experiment consists of a preparation period where a specific pulse sequence excites multiple-quantum coherences, an evolution period where these coherences evolve under the spin interactions, a mixing period to convert the higher order coherences to detectable single quantum coherences, and finally data acquisition (detection) [13]. The Fig. 2.2 illustrates the pulse sequence for a generic MQC experiment.

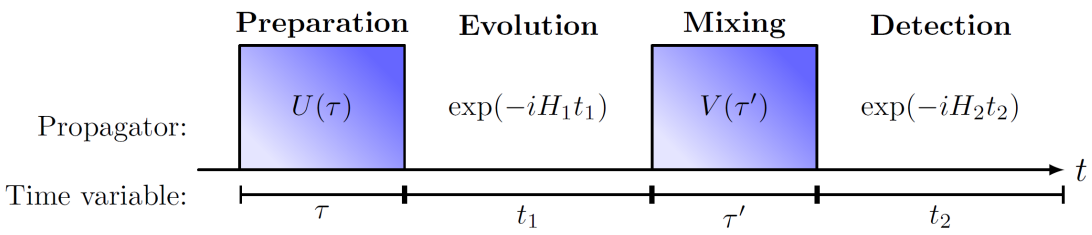


Figure 2.2: Schematic of a typical MQC NMR experiment. The pulse sequence is designed to generate multiple quantum coherences, which subsequently evolve under the influence of the spin Hamiltonian. These coherences are then transformed into single quantum coherences to enable their detection.

2.3.1 Generation of MQCs

The technique used to generate multiple quantum coherences is determined by the characteristics of the spin system in question, particularly the type and strength of spin-spin interactions present.

Solid-state systems

In solid-state NMR systems with strong dipolar interactions, MQCs can be excited using tailored pulse sequences that selectively create high-order coherences. For instance, an eight-pulse cycle, shown in can be used to engineer an effective Hamiltonian with flip-flip terms, as shown in Eq. (2.4), which preferentially excite ± 2-quantum transitions [13, 45]. We demostrate this taking polycrystalline adamantane as an example that has been introduced in the Sec 1.2. The spins naturally evolve under the dipolar interaction Hamiltonian, using which an effective Hamiltonian H_0 is engineered through the application of a periodic sequence of $\pi/2$ pulses [46]. These pulses are arranged in a cycle with a duration τ_0 and are designed to rotate the spin operators such that, over one cylce, the average Hamiltonian becomes

$$H_0 = \sum_{i<j} d_{ij} \left(I_{iy} I_{jy} - I_{ix} I_{jx} \right). \tag{2.4}$$

From Eq. (1.3), the initial state of the system is the high-temperature thermal equilibrium state, approximated by $\rho_0 \propto I_z$. Under evolution with the engineered Hamiltonian H_0, the density operator develops off-diagonal elements in the Zeeman basis. The density operator naturally takes on a product form when written as

$$\rho(\tau) \;=\; \rho(0) \;+\; i\tau\,[\,\rho(0), H_0\,] \;+\; i\tau^2\,[\,[\rho(0), H_0]\,, H_0\,]\cdots . \tag{2.5}$$

The nested commutators in this expansion generate terms like $I_{qz}I_{(q-1)z}\ldots I_{2z}I_{1x}$, representing correlated dynamics among q spins. These contributions scale with the $q/2$-th power of $d_{ij}\,\tau$ and are directly responsible for producing multiple quantum coherences. As a result, coherence orders $p = 0, \pm 2, \pm 4, \ldots$ emerge naturally in solid-state systems.

Liquid-state systems

Liquid-state NMR allows for more direct ways to generate MQCs within each molecule or spin systems as it allows for precise local as well as non-local gates [5]. As an illustrative example, consider a two-spin system: applying a Hadamard operation to one qubit, followed by a controlled-NOT interaction betweeen the two generates a Bell state. This entangled state exhibits a coherence order of 2.

Taking the example of star-topology spin systems introduced in the Sec 1.2 (refer to Fig. 1.2.4), one can generate MQCs of all orders in the following way. We apply the Hadamard gate on the central qubit to generate a superposition state. We then apply a controlled-NOT gate between the central and each of the outer qubits. If we start with the pure state of $|00\ldots\rangle$ with $M_z = -10$, this generates a W-state, which is a superposition of all possible spin configurations of the outer spins. The coherence order of the W-state is the highest possible coherence order in the system. However, in practice we have the thermal state of the system to begin with which contains a mixture of states with all possible M_z values ranging from -10 to 10. If we start with such a state, the MQCs generated will be a mixture of all

possible coherence orders.

2.3.2 Detecting MQCs

To detect MQCs, in a typical NMR experiment, high-order coherences must be transformed into observable single-quantum (SQ) coherences. The detected signal then reflects how each coherence order contributes to the measured magnetization. Here, we follow the treatment of Baum *et al.* [13] to describe the detection process. Assume the initial density operator is $\rho(0)$. After the preparation, evolution and mixing as shown in the Fig. 2.2, the detected signal is given by

$$S(\tau, t_1, \tau') = \mathrm{Tr}\Big\{ I_z \, V^\dagger(\tau') \, e^{-i\,H_1\,t_1} \, U^\dagger(\tau) \, I_z \, U(\tau) \, e^{i\,H_1\,t_1} \, V(\tau') \Big\} . \tag{2.6}$$

Letting the $t_1 = 0$ and expanding the operators in the Zeman eigenbasis $\{|i\rangle\}$, define the matrix elements

$$P_{ij}(\tau) = \langle i| \, U^\dagger(\tau) \, I_z \, U(\tau) |j\rangle \, , \tag{2.7}$$

$$Q_{ji}(\tau') = \langle j| \, V(\tau') \, I_z \, V^\dagger(\tau') |i\rangle \, . \tag{2.8}$$

Then the signal can be written as

$$S(\tau, \tau') = \sum_{i,j} P_{ij}(\tau) \, Q_{ji}(\tau'), \tag{2.9}$$

The idea is to use a mixing propagator that effectively undoes the effect of the preparation propagator up to a known phase factor. This can be imposed by choosing

$$V(\tau') = e^{-i\varphi\, I_z} \, U^\dagger(\tau') \, e^{i\varphi\, I_z} \, , \tag{2.10}$$

so that $V(\tau)$ is, apart from a phase shift by φ, the Hermitian conjugate of $U(\tau)$. This ensures

that all pathways corresponding to a given coherence order n acquire the same phase. Under the condition of Eq. (2.10), one can show that the signal separates into a discrete sum over coherence orders:

$$\begin{aligned} S(\tau) &= \sum_p \left[\sum_{i,j} \left| P_{ij}(\tau) \right|^2 \right] \exp\{i\, p\, \varphi\}\,, \\ &= \sum_p A_p(\tau) \exp\{i\, p\, \varphi\}\,. \end{aligned} \tag{2.11}$$

Here, each coherence order p is associated with an additional phase factor $\exp\{i\, p\, \varphi\}$. By systematically varying the phase φ, a Fourier transform with respect to φ can be employed to extract and distinguish the individual contributions corresponding to different coherence orders. The resulting spectrum A_p provides a direct measure of the intensity of each MQC in the system. Note that the amplitudes A_p depend directly on the ability of $U(\tau)$ to geenrate MQCs in the following way:

$$A_p = \sum_{i,j\,:\,|M_z(i) - M_z(j)| = p} |\langle i| U(\tau)\, I_z\, U^\dagger(\tau) |j\rangle|^2. \tag{2.12}$$

This theoretical framework forms the basis for detecting and analyzing MQCs in NMR experiments.

2.3.3 Experimental Details

All experiments were conducted at room temperature using a 500 MHz NMR spectrometer. A HX probe was utilized for solid-state measurements, while a QXI probe was employed for experiments in the liquid state. The pulse sequences were implemented using standard pulse programming techniques. The pulse lengths were calibrated using the standard $\pi/2$ and π pulse lengths for the spins under study.

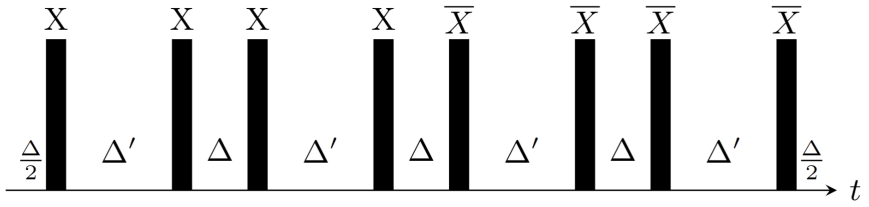


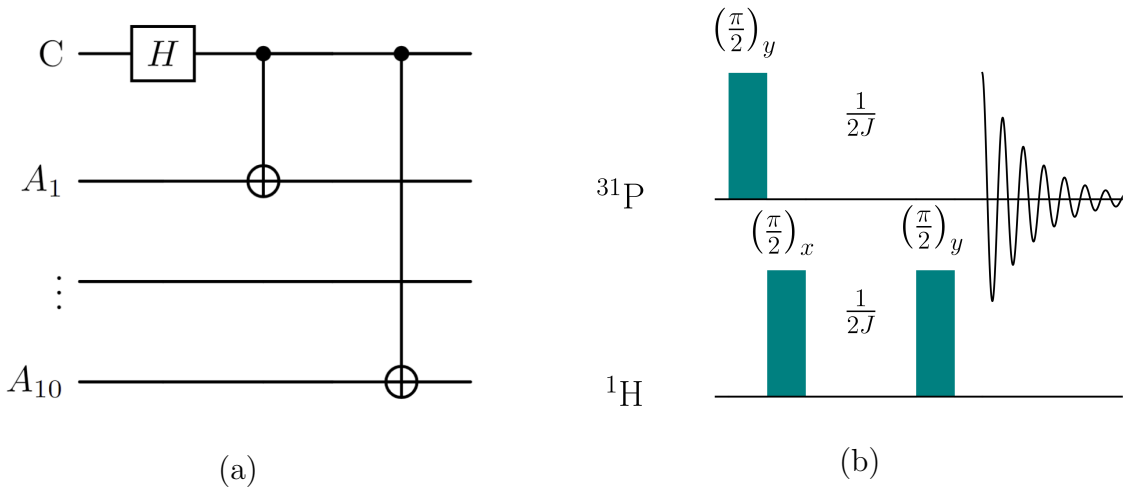
Figure 2.3: Pulse sequence for generating MQCs in adamantane. The effective Hamiltonian H_0 is realized through a specific sequence of $\pi/2$ pulses, as illustrated. In this setup, the timing parameter Δ' is defined as $\Delta' = 2\Delta + \tau_p$, where τ_p denotes the duration of a $\pi/2$ pulse. The pulses X and $\overline{X}$ refer to $\pi/2$ rotations applied along the $+x$ and $-x$ axes, respectively.

Adamantane

Preparation: For generation of MQCs the intermolecular dipolar interactions is essential. Hence, the experiments were performed on static polycrystalline adamantane introduced in the Sec 1.2 without any magic anngle spinning (MAS). The T_1 relaxation time for ^{1}H nulcei was observed to be 1.54 seconds. The MQC experiment employs the standard eight-pulse sequence originally developed by Warren *et al.* [47]. The pulse sequence consists of a series of $X \equiv (\frac{\pi}{2})_x$ and $\overline{X} \equiv (\frac{\pi}{2})_{-x}$ pulses with delays in between as shown in figure to generate the effective Hamiltonian H_0 as shown in Eq. (2.4). The delay Δ was set to 2μs and the pulse duration τ_p was set to 2.67μs. Noting that $\Delta' = 2\Delta + \tau_p$, the total duration of the pulse sequence can be calculated as $\tau_0 = 12\tau_p + 12\Delta = 56.04\mu$s. Looping over this pulse sequence N times amounts to applying the unitary $U(\tau)$ with $\tau = N\tau_0$.

Mixing: To implement $V(\tau)$, it is necessary to realize $U^\dagger(\tau)$, which can be achieved by applying a $\pi/2$ phase shift to each pulse in the original sequence. Therefore the mixing period consists of Y and $\overline{Y}$ pulses with an additional incremental phase shift of φ to implement full $V(\tau)$ as shown in Eq. (2.10).

Detection: After the mixing period a delay of about 0.5s is given to let all the single quantum coherences decay followed by a detection pulse - X that transfers all the z-magnetization



(a) (b)

Figure 2.4: (a) Circuit diagram illustrating the generation of multiple quantum coherences (MQCs) in TMP, representing the unitary evolution $U(\tau)$. A Hadamard gate is applied to the central qubit, while CNOT gates are applied between the central qubit and each of the outer qubits. (b) Corresponding pulse sequence for generating MQCs in TMP. The Hadamard gate is realized via a 90°_y pulse on the ^{31}P channel. The CNOT operation is implemented using a 90°_x pulse on the ^{31}P channel, followed by a delay of $1/(2J)$, and then a 90°_x pulse on the ^{1}H channel.

to X-Y plane for aquisition. The signal is obtained by varying φ over a range of values which is then fourier transformed to obtain the MQC spectrum.

TMP

Preparation: The J-couplings play a crucial role in the generation of MQCs in liquid-state systems. The experiments were performed on trimethyl phosphite (TMP) introduced in the Sec 1.2. J-coupling strenght between ^{31}P and ^{1}H nuclei was observed $J_{H-P} = 11$ Hz. The pseudo Hadamard gate is implemented on the central qubit by a 90°_y pulse on ^{31}P channel followed by a 90°_x-delay(1/2J)-90°_y sequence on ^{1}H channel which acts as a pseudo CNOT gate [48]. The pulse sequence used in the experiment is depicted in Fig. 2.4. Details regarding the calibration of pulse durations and relaxation times for TMP are provided in Table 2.1.

Nuclei	T_1 relaxation (in s)	$\pi/2$ pulse width (in μs)
^{1}H	5.6	11.02
^{31}P	9.7	18.25

Table 2.1: Relaxation times and pulse widths for TMP.

Mixing: The time reversed version of the preparation pulse sequence is used for implementing $U^{\dagger}(\tau)$. For the incremental phase shift φ each pulse is shifted by φ. This is possible because the phase rotation operator commuted with the interaction Hamiltonian of J-coupling.

Detection: A pulse field gradient is applied at the end of the mixing period to dephase any unwanted single-quantum coherences. Then a 90°_{y} pulse is applied on the ^{31}P channel to rotate all z-magnetization into the transverse plane, enabling detection on ^{31}P. Again, the signal is obtained by varying φ over a range of values which is then fourier transformed to obtain the MQC spectrum.

2.4 Results and Discussion

2.4.1 Solid-state NMR: Adamantane

The signal obtained in adamantane for different preparation times determined by N is shown in Fig. 2.4.1(a). Note that the signal for negative values of φ is obtained by reflecting the signal in the positive φ domain about $\varphi = 0$. This can be done keeping in mind the inherent symmetry in the signal about $\varphi = 0$. Additionally, to get the correct MQC spectrum, the signal has needs to be tiled multiple times in the φ domain. This is done keeping in mind the periodicity of the signal in the φ domain. Fig. 2.4.1(b) shows the spectrum for different vales of N. The intensity of the peaks at different orders gives a measure of the strength of the MQCs in the system.

N	τ(in μs)	K	Cluster Size ($\sqrt{K}$)
1	56.04	7.3	2.7
6	336.24	46.5	6.8
10	560.40	349.5	18.7

Table 2.2: Cluster size growth for different values of N in Adamantane.

Growth of Cluster Size

Longer the spin dynamics under tailored pulse sequences, higher will be the number of correlated spins and hence higher will be the coherence order of the MQCs generated. This is evident from the MQC spectrum obtained for different N values in adamantane. We can infer that the zero and double quantum coherences are present for $N = 1$ as the MQC spectrum shows peaks at $p = 0, \pm 2$. As the cluster size is increased to $N = 6$ and $N = 10$, the MQC spectrum shows peaks at higher orders $p = \pm 4, \pm 6, \ldots$ indicating the presence of higher order coherences. Such higher order coherences indicate that larger groups of spins have become coherently entangled which can be considered as forming a cluster. The growth in the size such clusters can be seen in the MQC spectrum as the appearance of higher order coherences. In a system of K spins, the number of transitions for a given $p = \Delta M_z$ follows a binomial distribution [46], which for $K \gg 1$, can be approximated by a Gaussian:

$$n(p, K) \propto \exp\left(-\frac{p^2}{K}\right). \tag{2.13}$$

The width of the MQC spectrum, $\sigma = \sqrt{K}$, corresponds to the cluster size. Table 2.4.1 displays the growth of cluster size for different values of N in Adamantane. The cluster size is determined as the square root of the number of spins in the cluster, as indicated in Eq. (2.13). This cluster size increases with the number of loops N in the pulse sequence, suggesting that the number of correlated spins in the system expands with the preparation time $N\tau_0$.

2.4.2 Liquid-state NMR: TMP

The signal obtained in TMP is shown in Fig. 2.4.2(a) and the corresponding fourier transform is shown in Fig. 2.4.2(b). The advantage of liquid state system is in precise control over the amplitudes of the MQCs generated. If instead of a Hadamard gate, a local y-rotation of some angle less than $\pi/2$ is applied, the amplitude of the MQCs generated can be controlled. However, the main limitation in liquid-state systemsis the number of spins that can be correlated simultaneously, which is constrained by the number of spins present in a single molecule. This restriction places an upper bound on the number of coherence orders that can be generated in liquid-state systems.

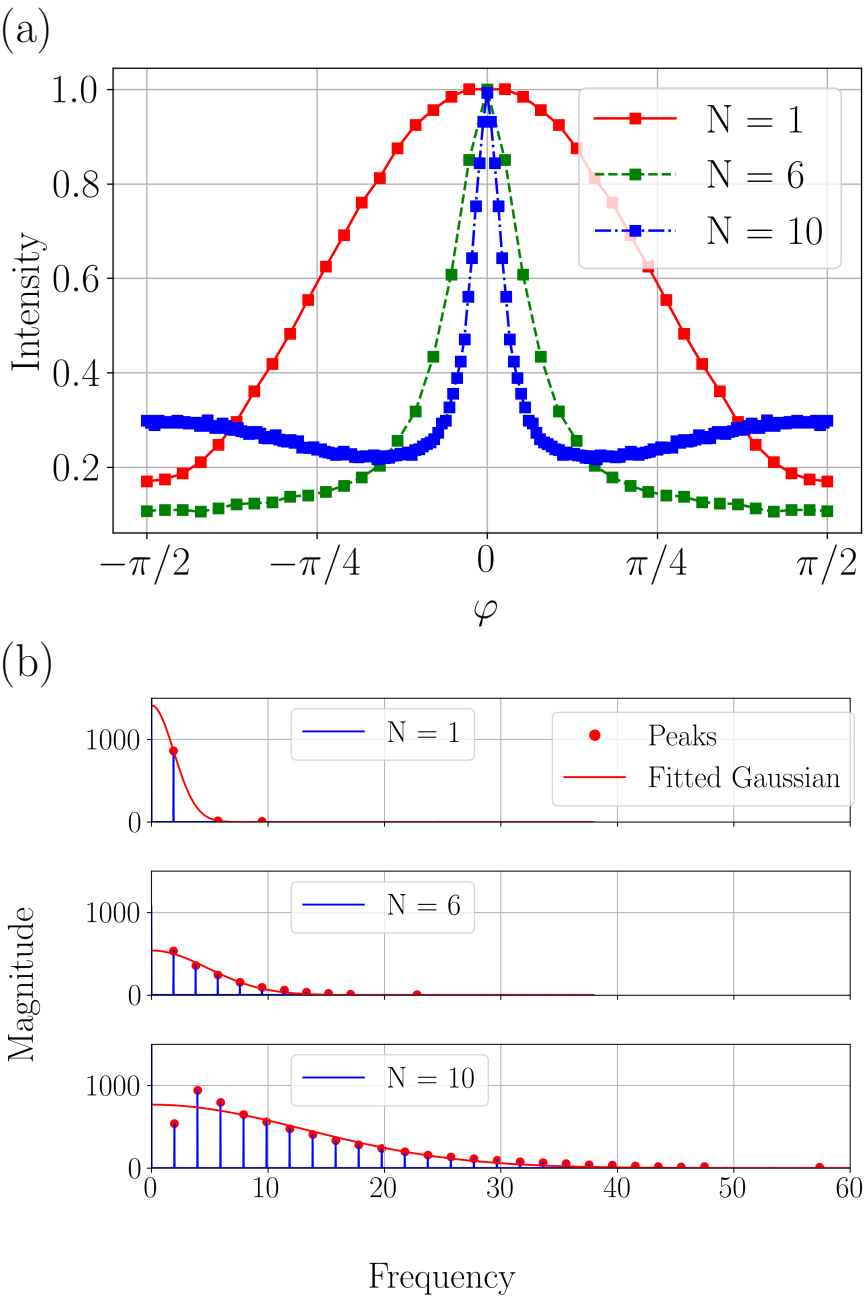


Figure 2.5: (a) The signal is obtained by varying φ over a range of values which is then fourier transformed to obtain the MQC spectrum. N corresponds to the number of loops the pulse sequence shown in Fig. 2.3 that determines the preperation time τ (which is also the mixing time) (b) The MQC spectrum obtained by fourier transforming the signal for different N values.

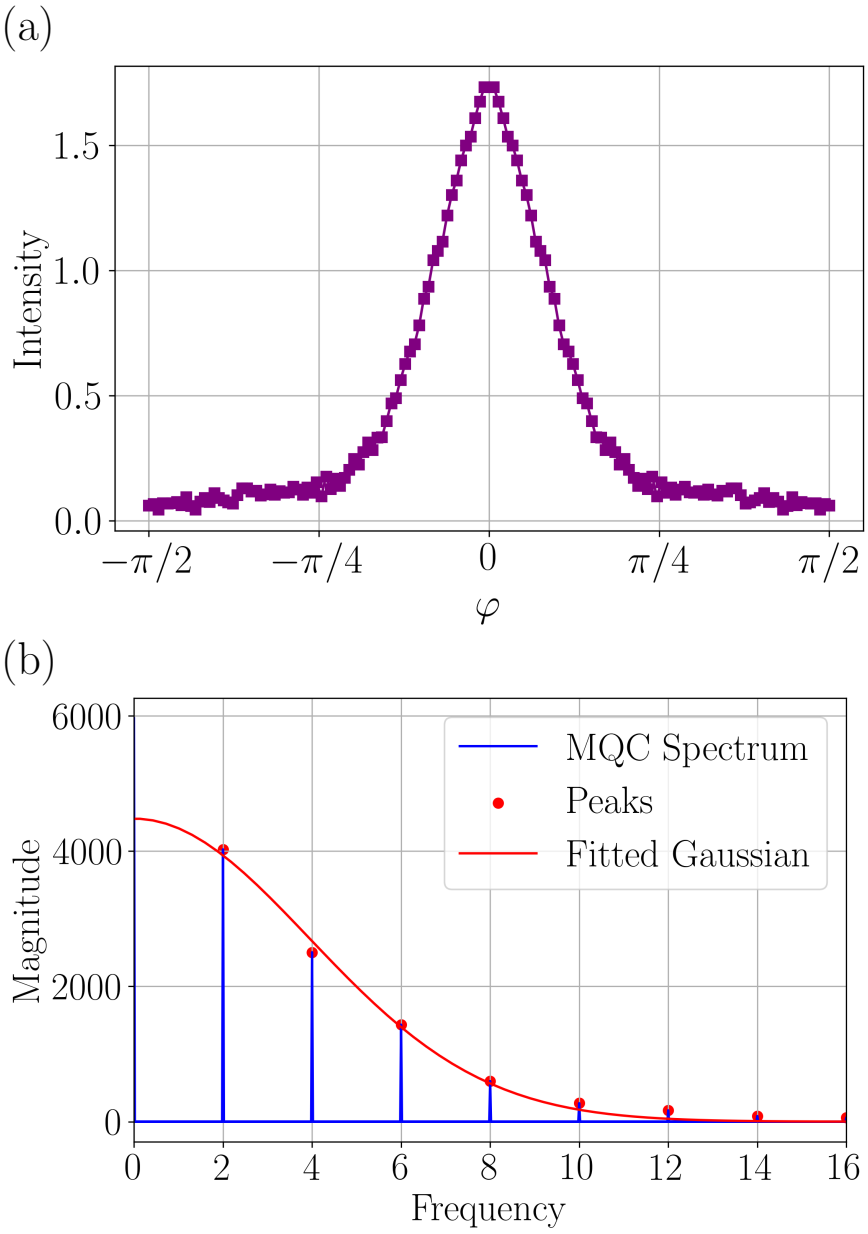


Figure 2.6: (a) MQC signal obtained for TMP. The signal is obtained by varying φ over a range of values which is then fourier transformed to obtain the MQC spectrum. (b) The MQC spectrum obtained by fourier transforming the signal tiled multiple times in the φ domain.

Chapter 3

Theory of Kernel Methods

3.1 Classical Kernel Methods

At the core of kernel methods lies the idea of transforming input data into a high-dimensional feature space—commonly known as the Reproducing Kernel Hilbert Space (RKHS)—where complex, non-linear relationships in the original input space become linear. This transformation is carried out implicitly through the evaluation of inner products between mapped data points using a kernel function. In this section, we present a concise overview of classical kernel-based techniques, with particular emphasis on linear regression and kernel ridge regression, which form the basis for their quantum counterparts.

3.1.1 Linear Regression

As one of the most foundational tools in machine learning, linear regression seeks to uncover a linear relationship between input features and a target variable. This is accomplished by fitting a hyperplane to the training data that minimizes the error between predicted and observed outcomes [49]. Rather than modeling complex structures directly, linear regression

assumes that the dependence of the target on the features can be effectively captured through a weighted sum, making it both interpretable and computationally efficient.

Model Formulation

Consider a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents the feature vector of the i-th observation, and $y_i \in \mathbb{R}$ is the corresponding target value. The model is trained on this set of information, which is referred to as the **training set**. The linear regression model *assumes* that the relationship between the input features and the output can be described by a linear function:

$$y_i = f^*(\mathbf{x}_i) + \epsilon_i, \tag{3.1}$$

and

$$f^*(\mathbf{x}_i) = \mathbf{w}^{*\top}\mathbf{x}_i + b \tag{3.2}$$

where $f^*(\mathbf{x})$ is the actual function from which the data points are sampled. $\mathbf{w}^* \in \mathbb{R}^d$ is the actual weight vector, $b \in \mathbb{R}$ is the bias term, and ϵ_i is the error term accounting for the noise in the sample data. For the purpose of clarity the bias, b, can be taken inside the weight vector by adding a dummy feature or an extra component to every feature vector which always takes the value 1. Formally, we transform $\mathbf{x}_i = [x_i^{(1)}, x_i^{(2)}, ..., x_i^{(d)}]$ into $\tilde{\mathbf{x}}_i = [x_i^{(0)}, x_i^{(1)}, x_i^{(2)}, ..., x_i^{(d)}]$ where $x_i^{(0)} = 1$ for all i. This allows us to write,

$$f^*(\mathbf{x}_i) = \tilde{\mathbf{w}}^{*\top}\tilde{\mathbf{x}}_i \tag{3.3}$$

where the transformed $\tilde{\mathbf{w}}^* = [b, w^{(1)}, w^{(2)}, ..., w^{(d)}]$. These transformed vectors $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{w}}^*$ are used in the rest of the discussion and tilde is dropped for the sake of brevity. The task now, is find the best estimate $f(\mathbf{x}_i)$ for the actual function $f^*(\mathbf{x}_i)$ or equivalently the best estimate $\mathbf{w}$ for the actual weight vector $\mathbf{w}^*$ from the limited training data.

Cost Function and Optimization

The above objective of linear regression can be achieved by finding the optimal value of $\mathbf{w}$ that minimizes the error between the predicted values $\hat{y}_i = f(\mathbf{x}_i) = \mathbf{w}^\top \mathbf{x}_i$ and the actual values y_i This is typically achieved by minimizing the Mean Squared Error (MSE) cost function, defined as:

$$J(\mathbf{w}, b) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 = \frac{1}{N} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2. \tag{3.4}$$

Minimizing this cost function with respect to $\mathbf{w}$ and b involves finding the gradient of $J(\mathbf{w}, b)$ and setting it to zero, which leads to the following normal equation:

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}, \tag{3.5}$$

where $\mathbf{X}$ is the design matrix formed by stacking the feature vectors $\mathbf{x}_i$ as rows, and $\mathbf{y}$ is the vector of target values [50]. Now we will show how a linear kernel function shows up in linear regression naturally and use the same construction to obtain the kernel function for non-linear regression tasks.

Regularization

To avoid overfitting and to make our model robust to noise in the sample, a regularized form of the linear regression cost function is considered, which includes a penalty term:

$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|^2, \tag{3.6}$$

where $\lambda > 0$ serves as a hyperparameter that balances the model's fit to the training examples against the magnitude of its weights. The solution $\mathbf{w}^*$, derived by optimizing this objective, admits a closed-form expression:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i, \tag{3.7}$$

where α_i are coefficients to be determined. Substituting this into the cost function and the prediction function, we get:

$$J\left(\{\alpha_i\}_{i=1}^{N}\right) = \frac{1}{N} \sum_{i=1}^{N} \left[y_i - \sum_{j=1}^{N} \alpha_j \mathbf{x}_j^\top \mathbf{x}_i \right]^2 + \lambda \left[\sum_{i,j} \alpha_i \alpha_j \mathbf{x}_i^\top \mathbf{x}_j \right]^2, \tag{3.8}$$

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i^\top \mathbf{x} \tag{3.9}$$

The task now reduces to optimizing the coefficients $\{\alpha_i\}$ such that they minimize the objective function defined in Eq. (3.8). Once determined, these $\{\alpha_i\}$ values are substituted into the predictive model given by Eq. (3.9) to estimate outputs for unseen data points.

The Linear Kernel

Observe that both Eq. (3.8) and Eq. (3.9) depend solely on inner products between feature vectors, not on the vectors themselves. This implies that neither the training process nor subsequent predictions require explicit knowledge of individual feature vectors. The usefulness of cost function and prediction function being independent of the individual feature vectors becomes evident in non-linear regression model. Here we define the linear kernel function,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j \tag{3.10}$$

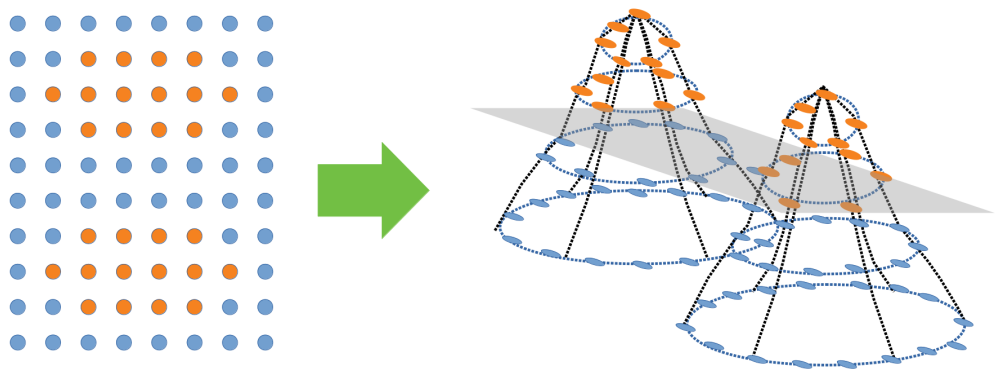


Figure 3.1: Illustrating data (left) and its feature space mapping using a higher-dimensional kernel (right).

3.1.2 Kernel Ridge Regression

We know that the assumptions made about the model in Eq. (3.1) and Eq. (3.2) do not hold in general. That is the dataset can be sampled from a non-linear function. A common strategy for handling such non-linear relationships involves mapping the input variables into an expanded feature space where a linear correlation with the target variable emerges, despite their non-linear interaction in the original space. However, instead of explicitly performing this transformation, kernel methods rely on computing inner products in the feature space directly using a kernel function (see Fig. 3.1) [51].

To better understand this, consider a training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents the feature vector of the i-th observation, and $y_i \in \mathbb{R}$ is the corresponding target value. This dataset has been sampled from $f^*(\mathbf{x})$, which we know to be non-linear.The objective becomes determining an optimal approximation $f(\mathbf{x})$ of the underlying function based on available training samples. To address nonlinear patterns, kernel techniques employ a transformation $\phi : \mathbb{R}^d \to \mathcal{F}$ that projects inputs into an enriched feature space $\mathcal{F}$, where the following assumption about the model becomes valid:

$$y_i = f^*(\mathbf{x}_i) + \epsilon_i, \tag{3.11}$$

and

$$f^*(\mathbf{x}_i) = \mathbf{w}^{*\top}\phi(\mathbf{x}_i). \tag{3.12}$$

The formulation in Eq. (3.12) reveals that $f^*(\curvearrowleft)$ behaves as a linear combination of the mapped features $\phi(\mathbf{x}_i)$ in $\mathcal{F}$. This establishes an important duality: linear operations in the high-dimensional feature space effectively capture nonlinear relationships present in the original $\mathbb{R}^d$ space. However, explicitly calculating $\phi(\mathbf{x})$ for every data point and then computing the inner product in $\mathcal{F}$ is computationally expensive and often infeasible. This is exactly where the linear regression model being independent of the individual feature vectors becomes significant. It enables the application of what is commonly called the *kernel trick*. Through this technique, we can efficiently evaluate inner products in $\mathcal{F}$ by means of a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$, bypassing the need for direct computation in the high-dimensional space. This becomes evident when we express the regularized objective function in terms of the feature space $\mathcal{F}$.

$$J(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w}^\top\phi(\mathbf{x}_i))^2 + \lambda\|\mathbf{w}\|^2, \tag{3.13}$$

with the positive parameter λ, serving as a regularization strength parameter, balances model complexity against training fit, mirroring its function in linear regression. Following the same derivation approach as before, the minimizing solution for $\mathbf{w}$ takes the form:

$$\mathbf{w} = \sum_{i=1}^{N}\alpha_i\phi(\mathbf{x}_i) \tag{3.14}$$

This result is also known by the name *Representer Theorem*. Substituting this into the cost function and the prediction function, and defining the kernel function as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j), \tag{3.15}$$

we get:

$$J\left(\{\alpha_i\}_{i=1}^N\right) = \frac{1}{N}\sum_{i=1}^{N}\left[y_i - \sum_{j=1}^{N}\alpha_j k(\mathbf{x}_j, \mathbf{x}_i)\right]^2 + \lambda\left[\sum_{i,j}\alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j)\right]^2, \tag{3.16}$$

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}) \tag{3.17}$$

Now the significance of cost function and prediction function being independent of the individual feature vectors and of the kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ becomes clear. The kernel function allows us to compute the inner product in the feature space $\mathcal{F}$ without explicitly computing the feature vectors $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$. And that further allows us to train our model and predict the output(or target) value for any new feature vector (or input vector). This is the essence of the kernel trick. To be more explicit, model training involves finding the optimal values of $\{\alpha_i\}$ that minimize the objective function defined in Eq. (3.16). Once optimized, these α_i can be used in the prediction function in Eq. (3.17) to predict the output for any new input.

Within this section we have examined the kernel trick for regression. The underlying principle remains equally applicable to classification problems. The theory for kernel trick in classification tasks is similar to that of regression tasks with the only difference being the cost function and the prediction function. For more details on the theory of kernel methods in classification tasks, refer to [51].

3.1.3 Common Classical Kernels

Several kernel functions are commonly used in classical machine learning and here are few examples of them:

- **Polynomial Kernel:** $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + c)^d$, where c and d are parameters. This kernel maps the data into a polynomial feature space.
- **Radial Basis Function (RBF) Kernel:** $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$, where σ controls the width of the Gaussian function. The RBF kernel maps data to an infinite-dimensional space, making it highly flexible.

- **Sigmoid Kernel:** $k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\alpha \mathbf{x}_i^\top \mathbf{x}_j + c)$, resembling the neural network activation functions.

The effectiveness of kernel techniques stems from their capacity to model nonlinear relationships through implicit high-dimensional embeddings. Despite being fundamental to conventional machine learning, their computational overhead for kernel matrix operations escalates rapidly with data size and dimensionality, driving research into quantum computing alternatives for these calculations.

3.2 Quantum Kernel Methods

These methods constitute a novel paradigm in machine learning that harnesses quantum information processing to boost the capabilities of traditional kernel-based models. The principal idea is to map the input data into the operator space of the quantum system that exploits the rich algebraic structure of quantum operators, without necessitating a full-scale quantum computer [52]. The inner products of these quantum states (or operators) are then computed using quantum algorithms, leading to the construction of quantum kernels that capture the underlying structure of the data. These quantum kernels can be used to train and predict in kernel-based models just like the classical kernels. Quantum kernel methods typically involve two main steps: Quantum Feature Mapping and Quantum Kernel Computation [53].

3.2.1 Quantum Feature Mapping

In quantum feature encoding, each input sample gets mapped to a quantum state or operator that encodes the data's inherent patterns and relationships. The selection of an appropriate quantum feature map is governed by both the problem requirements and the target char-

acteristics of the transformed feature space, while also being constrained by the practical limitations of available quantum hardware. Some common quantum feature mappings include quantum Fourier transform, quantum amplitude encoding, and quantum variational circuits and so on.

In this work we map the input data into the operator space of the quantum systems. Given an input data point χ_i, which can be classical or quantum in nature, we map it to an operator $A(\chi_i)$ in the operator space of the quantum system in the following way:

$$A(\chi_i) = U(\chi_i) A_0 U^\dagger(\chi_i), \tag{3.18}$$

where A_0 is a reference operator and is chosen to be Hermitian. The unitary transformation $U(\mathbf{x}_i)$ critically influences the feature space's capacity and complexity. Increased entangling capability in the unitary operator enhances the feature space's expressivity [31]. This operator $A(\chi_i)$ can be represented as a vector by expanding it in terms of Pauli operators. For an n-spin-$\frac{1}{2}$ system, this expansion results in $A(\chi_i)$ being a vector in a $\mathbb{R}^{4^n}$ dimensional space. The operator $A(\chi_i)$, when n is sufficiently large and $U(\chi_i)$ is expressive enough, becomes computationally intractable for classical computers due to the exponential growth in complexity. Particular examples of such quantum feature mappings are described in Chapter 4 and Chapter 5, for classical and quantum data respectively.

3.2.2 Quantum Kernel Computation

After mapping data points to quantum states, the quantum kernel is computed by evaluating inner products between state vectors. The resulting kernel matrix, built from these quantum state similarities, enables subsequent training and inference in kernel-based learning. Quantum kernel computation can be performed using various techniques, such as quantum state overlap estimation, quantum kernel alignment, or quantum kernel matrix inversion. These techniques leverage quantum algorithms and quantum circuits to efficiently compute

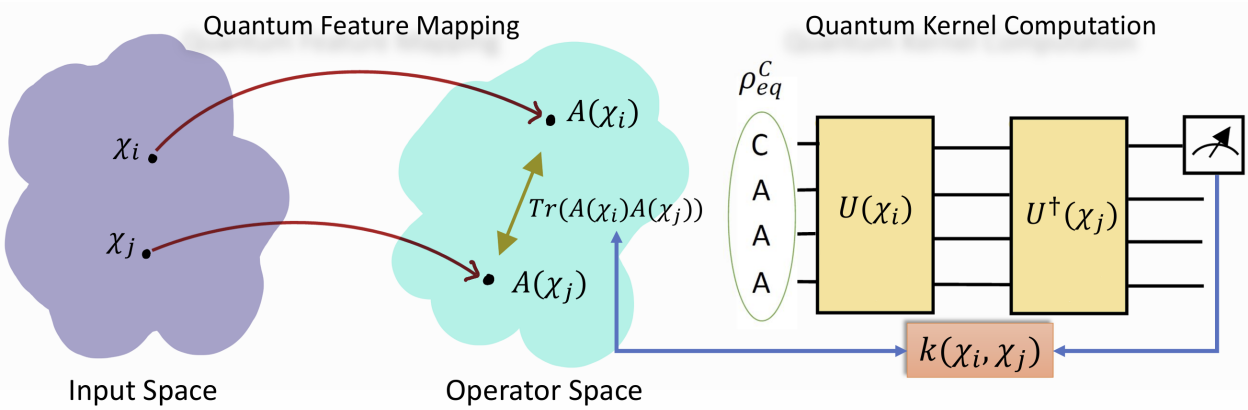


Figure 3.2: Quantum feature mapping and quantum kernel computation protocol. The quantum circuit starts with the initial state of ρ_{eq}^{C} in order to ensure the symmetry of the quantum kernel since only the C qubit is measured at the end. For more details refer to Sec 4.1

the kernel values. For the feature map defined in Eq. (3.18), the quantum kernel can defined as:

$$k(\chi_i, \chi_j) = \langle A(\chi_i), A(\chi_j) \rangle_F = \mathrm{Tr}\left(A(\chi_i)A(\chi_j)\right), \tag{3.19}$$

where the Frobenius inner product[1] has been selected as our operator inner product. This formulation defines the quantum kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ employed in our investigation.

If $A_0 \propto \rho_{eq}$ then $A(\chi_i)$ can be interpreted as the density matrix (quantum state), $\rho(\chi_i)$, that we get upon acting $U(\chi_i)$ on the system. The quantum kernel function $k(\chi_i, \chi_j)$ then measures the overlap between the quantum states corresponding to the input data χ_i and χ_j. This interpretation becomes useful in visualizing the quantum kernel function as a measure of similarity between the quantum states. The quantum feature mapping and quantum kernel computation protocol is illustrated in Fig. 3.2. Note that computation of kernel values does not require a universal quantum computer and can be done on a quantum system with a limitted control provided it allows the dynamics that can aid in implementing highly entangling $U(\chi)$ and computing the the Frobinious inner product between the operators.

[1]For operators A and B, the Frobenius inner product is $\langle A, B \rangle_F = \mathrm{Tr}(A^{\dagger}B)$, coinciding with the Hilbert-Schmidt inner product [54]. The Hermitian nature of $A(\mathbf{x}_i)$ allows omitting the dagger operation.

Chapter 4

Quantum Kernel for Classical Data

This chapter investigates the implementation and utilization of quantum kernels using an NMR spin register system, with particular emphasis on processing *classical* input data. Building upon the multiple quantum coherences (MQCs) introduced in Chapter 2, we encode real-valued classical data into operator representations on a quantum system. We then experimentally compute the resulting kernel function and demonstrate two standard machine-learning tasks. As mentioned earlier, these MQCs can be seen as a witness of the entanglement between the spins in the system. The goal in this chapter is to show that the kernel values computed using these MQCs can be used to train and predict in kernel-based models and thus hinting at a possible connection between the quantum entanglement and expressive quantum kernel functions. Parts of this chapter are inspired by the quantum kernel approach of Kusumoto *et al.* [1] and further extended in our own work [42].

4.1 Implementation

4.1.1 Quantum Feature Mapping in NMR

Recall from Sec 3.2 that, for a classical input $\mathbf{x}_i \in \mathbb{R}^n$, a *quantum feature map* can be constructed by associating to each $\mathbf{x}_i$ a unitary $U(\mathbf{x}_i)$. Then, by choosing a reference operator A_0 in the quantum system, we define

$$A(\mathbf{x}_i) \;=\; U(\mathbf{x}_i)\, A_0\, U^\dagger(\mathbf{x}_i). \tag{4.1}$$

In our NMR experiment, we use $A_0 = I_z^C$, the z-magnetization operator of the *central* spin in a star-topology spin register (see Sec 1.2). We utilize this operator $A(\mathbf{x}_i)$ as a feature map, denoted as $\phi_{\text{NMR}}(\mathbf{x}_i)$, for our quantum machine learning tasks. Given 10-spin-$\frac{1}{2}$ nuclear spins in the system, the operator $A(\mathbf{x}_i)$ becomes computationally intractable for classical computers when the $U(\mathbf{x}_i)$ involves entangling all the spins in the system. This is due to the exponential growth in complexity as the number of spins increases. Such entanglement can be accomplished by using the pulse sequence used for generation of MQCs shown in Fig. 2.4. The systems is shown to maintain coherence over tens of spins in Chapter 2, which implies that we can effectively work with feature vectors of dimension $4^{O(10)}$. The mere existence of high-dimensional feature spaces doesn't automatically translate to improved machine learning performance. However, the potential to access these computationally prohibitive spaces presents a valid rationale for examining their utility.

Encoding Classical Inputs. Following Refs. [1, 42], we encode each component of the data vector $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)})$ into a series of phase shifts on the ancillary qubits. Denoting a single "entangling block" by U_e, which is equivalent to the MQC generating unitary for star-topology systems in Sec 2.3, we then define, for each dimension $j = 1, \ldots, n$,

$$V(x_i^{(j)}) \;=\; e^{-i\,x_i^{(j)}\,I_z^A}\; U_e\; e^{+i\,x_i^{(j)}\,I_z^A},$$

where I_z^A acts collectively on the ancillary spins or one at a time, depending on the pulse implementation. Finally, we combine these steps for all j to form

$$U(\mathbf{x}_i) \;=\; \prod_{j=1}^{n} V\big(x_i^{(j)}\big). \tag{4.2}$$

4.1.2 NMR Kernel Computation

The kernel method offers a practical approach to untilize $A(\mathbf{x}_i)$ directly for machine learning applications. While element-wise evaluation of $A(\mathbf{x}_i)$ remains computationally prohibitive, we can directly determine the Frobenius inner product for feature vector pairs. This measurement forms the basis of our NMR kernel function $k_{\text{NMR}}(\mathbf{x}_i, \mathbf{x}_j)$ in this investigation. Now we can note that,

$$\begin{aligned} k_{\text{NMR}}(\mathbf{x}_i, \mathbf{x}_j) &= \text{Tr}\left(A(\mathbf{x}_i)A(\mathbf{x}_j)\right) \\ &= \text{Tr}\left(U(\mathbf{x}_i)I_z^C U^\dagger(\mathbf{x}_i)U(\mathbf{x}_j)I_z^C U^\dagger(\mathbf{x}_j)\right) \\ &= \text{Tr}\left(U^\dagger(\mathbf{x}_j)U(\mathbf{x}_i)I_z^C U^\dagger(\mathbf{x}_i)U(\mathbf{x}_j)I_z^C\right) \end{aligned} \tag{4.3}$$

The cyclic properties of the trace have been used in to get the last expression. If we destroy the magnatization of all the A spins in the thermal state in Eq. (1.3) and call the resulting state as ρ_{eq}^C, the above expression can be written as:

$$k_{\text{NMR}}(\mathbf{x}_i, \mathbf{x}_j) \propto \text{Tr}\left(U(\mathbf{x}_j)U(\mathbf{x}_i)\rho_{eq}^C U^\dagger(\mathbf{x}_i)U^\dagger(\mathbf{x}_j)I_z\right) \tag{4.4}$$

The physical interpretation becomes evident when examining $U(\mathbf{x}_i)U(\mathbf{x}_j)\rho_{eq}^C U^\dagger(\mathbf{x}_i)U^\dagger(\mathbf{x}_j)$: this operator represents the evolved quantum state after initializing the system in ρ_{eq}^C, subjecting it to unitary evolutions $U(\mathbf{x}_i)$ and $U(\mathbf{x}_j)$, and finally measuring the central spin's I_z magnetization of this state gives the kernel. In this text that follows we will equate $k_{\text{NMR}}(\mathbf{x}_i, \mathbf{x}_j)$ to the expression on the right of the above equation up to a constant of proportionality.

4.2 Experimental Details

The experimental measurements were conducted using a 500 MHz NMR spectrometer equipped with a QXI probe for liquid state samples. We use a star-topology molecule, namely *trimethyl phosphite* (TMP) where ^{31}P is the central spin and the three methyl ^{1}H groups collectively serve as ancillary spins [42].

1. **Initialization:** To destroy the z-magnatization of the A spins and prepare the thermal equilibrium state ρ_{eq}^{C}, we begin with a 90_y° excitation pulse applied to the A spins and then a gradient pulse to eliminate transverse magnetization components.

2. **Data Encoding:** For each classical input $\mathbf{x}_i$, we implement the unitary $U(\mathbf{x}_i)$ as a sequence of local rotation pulses on the ancillary spins to implement the input dependent z-rotation, interspersed with U_e. With this we prepare

$$\rho_{\text{fin}} \;=\; U^{\dagger}(\mathbf{x}_j)\, U(\mathbf{x}_i)\, \rho_{\text{eq}}^{C}\, U^{\dagger}(\mathbf{x}_i)\, U(\mathbf{x}_j).$$

3. **Kernel Value Readout:** We measure $\langle I_z^C \rangle$ by applying a 90_y° pulse on the central spin and then detecting the resulting free induction decay. This signal (normalized and scaled by constants) yields $k_{\text{NMR}}(\mathbf{x}_i, \mathbf{x}_j)$ up to an overall factor.

Iterating over $(\mathbf{x}_i, \mathbf{x}_j)$ pairs yields the full kernel matrix, which can then be used in standard kernel-based machine learning algorithms.

4.3 Results and Discussion

In what follows, we show how this NMR-based quantum kernel can handle typical regression and classification problems with classical data. Our results are consistent with and extend the solid-state demonstration by Kusumoto *et al.* [1].

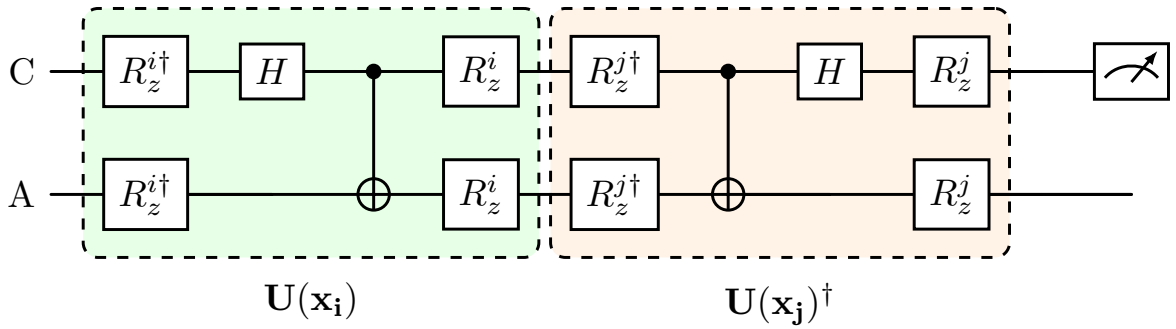


Figure 4.1: Quantum circuit for encoding classical inputs x_i and x_j in terms of unitaries $U(x_i)$ and $U(x_j)$, followed by measurement of z magnetization of C qubit. Here, helix represents the dephasing operations on x and y magnetizations C-qubit. The circuit assumes we start with the thermal equilibrium state ρ_{eq}^C.

4.3.1 One-Dimensional Regression

Consider a scalar input $x_i \in \mathbb{R}$ that we wish to regress onto a target function such as a trigonometric function or polynomial. We pick regularly spaced training points $\{x_i\}$, compute the kernel matrix $K_{ij} = k_{\text{NMR}}(x_i, x_j)$, and then fit a kernel ridge regression model (Sec 3.1.2). Writing the kernel explicitly, we get

$$\begin{aligned} k_{\text{NMR}}(x_i, x_j) &= \text{Tr}\left(U^\dagger(x_j)\, U(x_i)\, \rho_{\text{eq}}^C\, U^\dagger(x_i)\, U(x_j)\, I_z^C\right) \\ &= \text{Tr}\left(U_e^\dagger\, e^{i(x_i - x_j) I_z^A}\, U_e\, \rho_{\text{eq}}^C\, U_e^\dagger\, e^{-i(x_i - x_j) I_z^A}\, U_e\, I_z^C\right). \end{aligned} \tag{4.5}$$

Hence the kernel function depends only on the difference $x_i - x_j$. This symmetry can be exploited to reduce the number of experimental points needed to compute the kernel matrix. Considering $\varphi = x_i - x_j$ and noting that $e^{-i\varphi I_z^A}$ commutes with I_z^C, we can write this as

$$k_{\text{NMR}}(x_i, x_j) = \text{Tr}\left(e^{-i\varphi I_z^A}\, U_e^\dagger\, e^{i\varphi I_z^A}\, U_e\, \rho_{\text{eq}}^C\, U_e^\dagger\, e^{-i\varphi I_z^A}\, U_e\, e^{i\varphi I_z^A}\, I_z^C\right). \tag{4.6}$$

This right-hand term represents the MQC signal generated through the experimental sequence in Fig. 2.4 when applying a phase shift of φ. This means that the kernel function is basically the difference $x_i - x_j$ mapped to different fourier components of the MQC signal as can be seen fron Eq. (2.11). Consequently, the kernel $k(x_i - x_j)$ is identical to the results

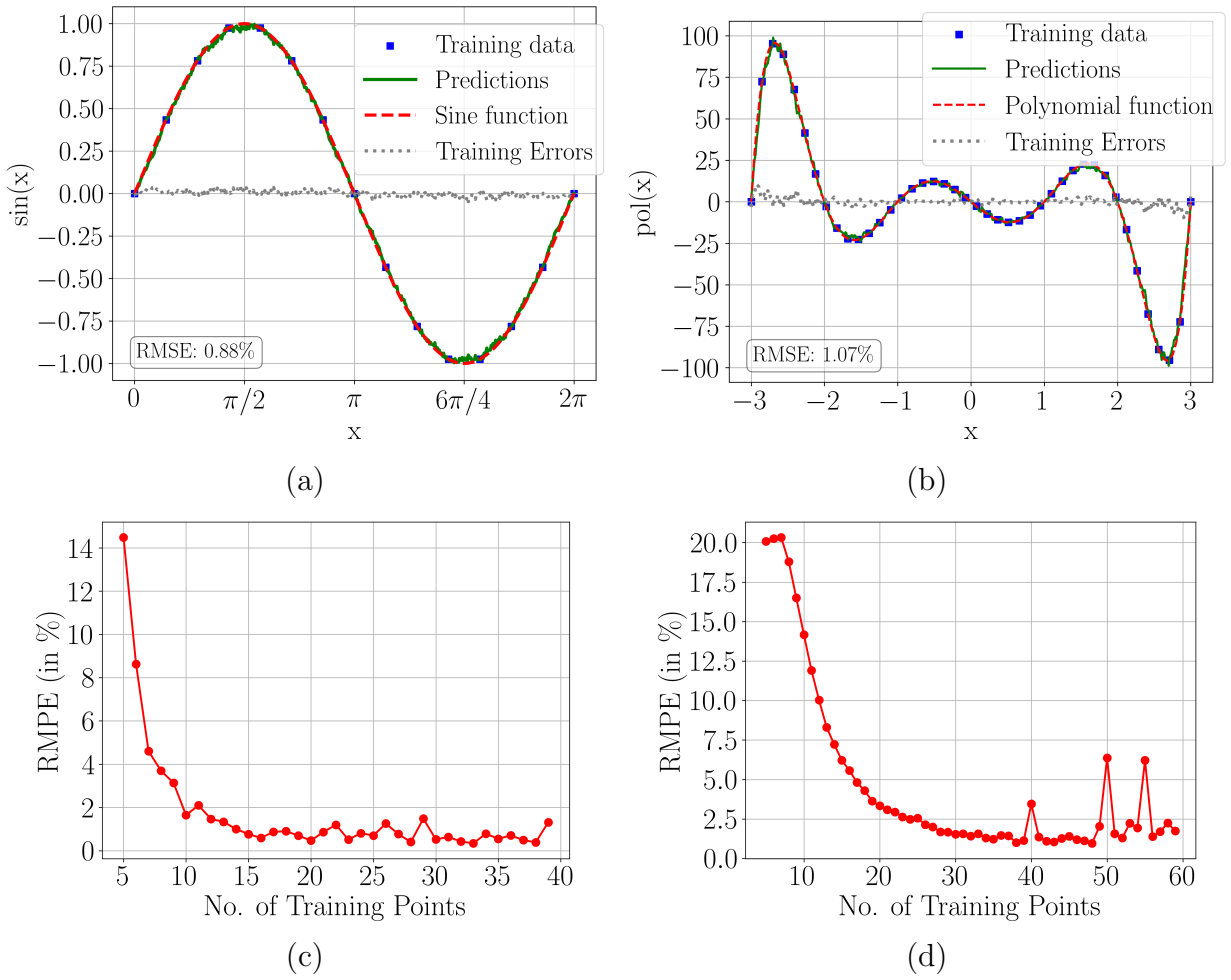


Figure 4.2: One-dimensional regression task with NMR Quantum Kernel. Regression of (a) sine function with 15 training data points and (b) a seventh-degree polynomial function with 42 training data points. (c) and (d) plot the percentage RMSE against the number of training samples for the sinusoidal and polynomial functions, correspondingly.

displayed in Fig. 2.4.2(a), with the abscissa representing the quantity $x_i - x_j$. The Fig. 4.1 shows the quantum circuit for encoding the classical inputs x_i and x_j in terms of unitaries $U(x_i)$ and $U(x_j)$, followed by measurement of z magnetization of the central spin.

As for the tasks, Fig. 4.2(a) illustrates one such example, where the target function is a sine function for one full period and we used 15 training points. Fig. 4.2(a) shows the same for an 8^{th} degree polynomial. Our trained model faithfully reproduces the sine function as well as the polynomial, with an RMS error well below 2%, underscoring the efficacy of the quantum

kernel approach for simple regression tasks.

4.3.2 Two-Dimensional Classification

We next extend the data input to $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}) \in \mathbb{R}^2$. We again construct

$$U(\mathbf{x}_i) \;=\; e^{-ix_i^{(1)} I_z^A}\, U_e\, e^{ix_i^{(1)} I_z^A}\; e^{-ix_i^{(2)} I_z^A}\, U_e\, e^{ix_i^{(2)} I_z^A}.$$

The resulting kernel $k_{\mathrm{NMR}}(\mathbf{x}_i, \mathbf{x}_j)$ is extracted experimentally for a grid of input points. Since this kernel satisfies the symmetry [1]

$$k(\{x_i^{(1)}, x_i^{(2)}\}, \{x_j^{(1)}, x_j^{(2)}\}) = k(\{x_i^{(1)} - x_j^{(2)}, x_i^{(2)} - x_j^{(2)}\}, \{x_i^{(1)} - x_i^{(2)}, 0\}), \tag{4.7}$$

we can set $x_j^{(2)} = 0$, and extract the kernel for two-dimensional inputs with only three independent parameters, $x_i^{(1)}$, $x_i^{(2)}$ and $x_j^{(1)}$. Fig. 4.3 shows the $x_j^{(1)}$ slices of the experimental kernel.

We then train, e.g., a support vector machine (SVM) to separate the data into two classes. Fig. 4.4 (a) and (b) show examples with the "circles" and "moons" dataset respectively, which are standard synthetic dataset in machine learning that is *not* linearly separable in the original space. In Figs. 4.4 (a,b), circles and squares show two classes of training datasets. The decision functions learned by the SVM are shown by the backgrounds of Figs. 4.4 (a,b). The hinge loss values, that quantify errors in these classifications, are 0.15 and 0.08 respectively. These results show that the choice of the protocol to compute the kernel is fairly reliable and has the potential to carry out standard ML tasks.

Overall, these classical-data tasks confirm that NMR-based quantum kernels can serve as powerful function approximators, with the dimension of the operator space acting as a "feature dimension" that is potentially very large. A direct classical simulation of large star-topology spins becomes intractable with increasing qubit number. This advantage hints at

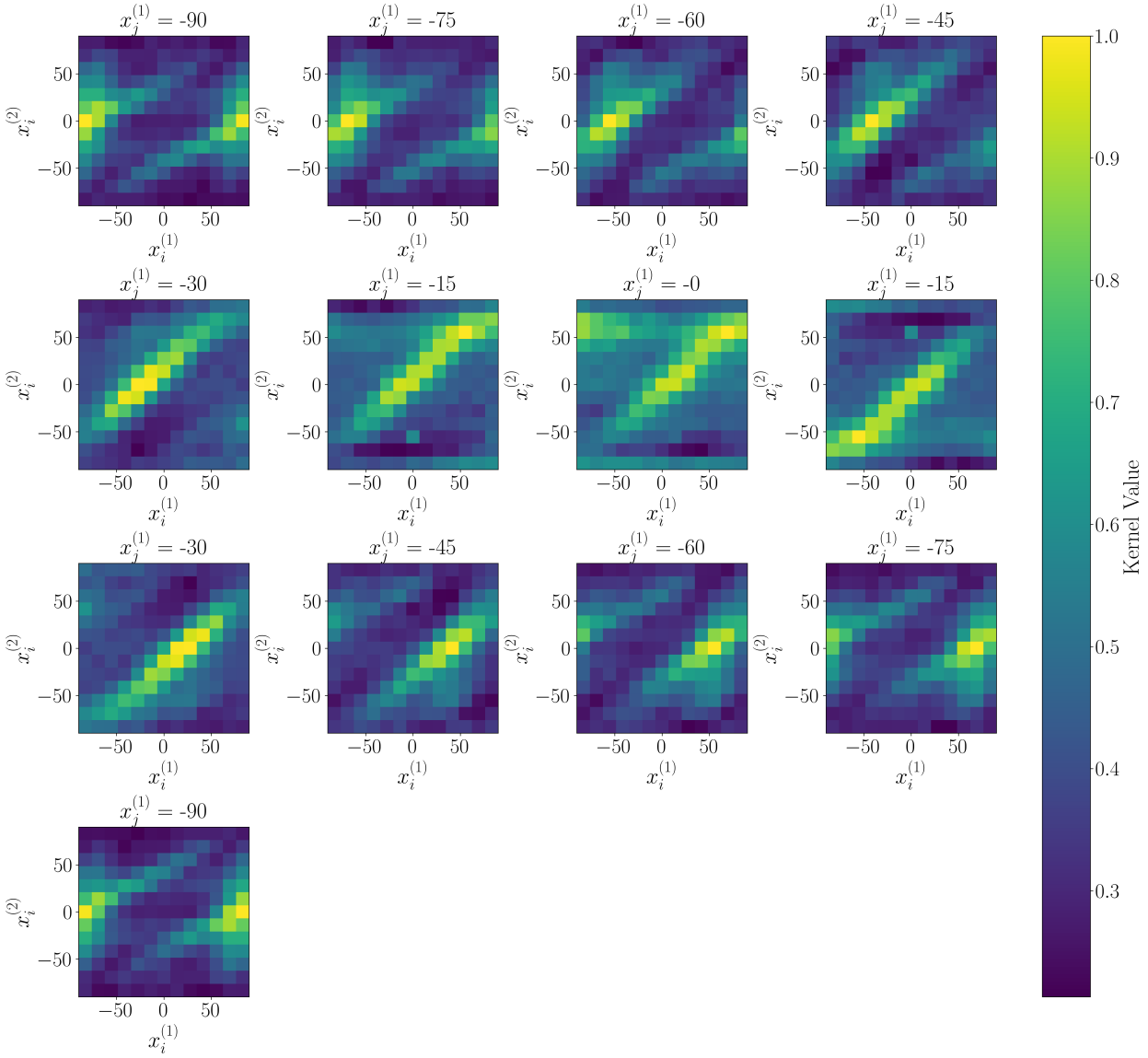


Figure 4.3: Two dimensional NMR quantum kernel and the corresponding classification tasks in 2D plane. The $x_j^{(1)}$ slices of the experimentally obtained kernel for two-dimensional input.

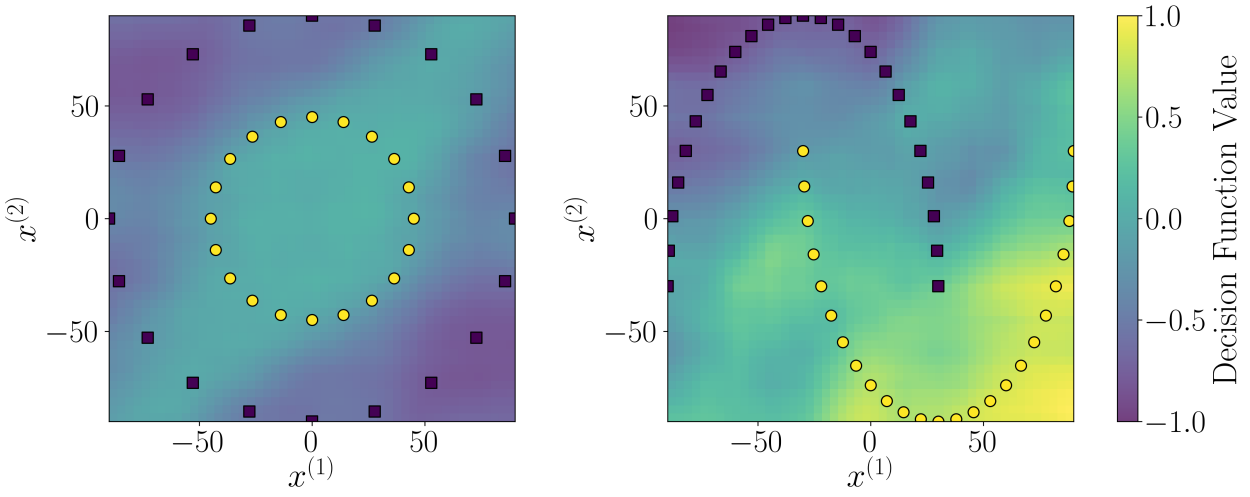


Figure 4.4: Two-dimensional classifications for (a) circles dataset and (b) moons dataset. In both classifications, the squares and circles represent the training data points belonging to two different classes. The background represents the decision function obtained after training.

the possibility of quantum speedup or at least a complexity-theoretic separation for certain classes of problems, although more theoretical analysis is needed to establish unconditional quantum advantage [42].

In this chapter, we demonstrated how one can encode *classical* real-valued inputs into quantum operators on an NMR spin register, measure their pairwise similarities (kernel values), and employ these kernel values to carry out regression and classification tasks. The success of these demonstrations reinforces the notion that *multiple quantum coherences* can provide an exponentially large effective feature space, with minimal overhead in hardware control. In the next chapter, we will shift our attention to *quantum* inputs—namely, unitaries and/or quantum states—and show that essentially the same kernel framework can be leveraged to discriminate entangling vs. non-entangling transformations, thus broadening the scope of quantum machine learning to fully quantum tasks.

Chapter 5

Quantum Kernel for Quantum Data

In the previous chapter, we used quantum kernels for classical data. However, a significant power of quantum machine learning arises when the input data themselves are quantum—for instance, quantum states or unitaries that one wants to classify or regress upon. This chapter generalizes kernel techniques to quantum data domains, demonstrating their application to quantum tasks like *entanglement cclassification*. Much of this discussion parallels our results in Ref. [42], where we proposed and numerically demonstrated a procedure to classify which unitaries become entangling on a given initial state.

5.1 Entanglement Classification Problem

We focus on a scenario described in [42] wherein we have:

1. A fixed initial quantum state ρ_0 of some number of spins (or qubits).

2. A family of candidate unitaries $\{U_i\}$, each of which *may* or may not generate entanglement in the final state $U_i \rho_0 U_i^\dagger$.

The goal is to label each U_i as *Entangling* or *Nonentangling* with respect to the initial state ρ_0. It is a nontrivial problem, which is discussed by Serrano-Ensástiga and Martin [55]. In many-body or multi-qubit settings, one typically has to check partial transposes or negativity measures to ascertain entanglement. Doing so repeatedly for a large set of U_i can be extremely costly. Instead, we propose a quantum kernel approach that can classify the unitaries more efficiently once the kernel is well-trained.

To be more concrete, consider a two-qubit system with an initial thermal state $\beta = 1.5$ given by

$$\rho_0 = e^{-\beta(I_{1z}+I_{2z})}/4,$$

We introduce a set of single-qubit unitary operations applicable to qubit 2, as depicted in Fig. 5.1(a). The operations combine a y-rotation $R_y(\theta)$ on qubit A with a composite pulse sequence $U(\alpha) = 90^\circ_x$-delay($\frac{\alpha}{2\pi J}$)-90°_y, where J represents the A-B qubit coupling constant. Some of these $U(\theta, \alpha)$ generate entanglement in $U\rho_0 U^\dagger$, some do not. Fig. 5.1(b) shows the range of θ and α values for which the unitaries are entangling.

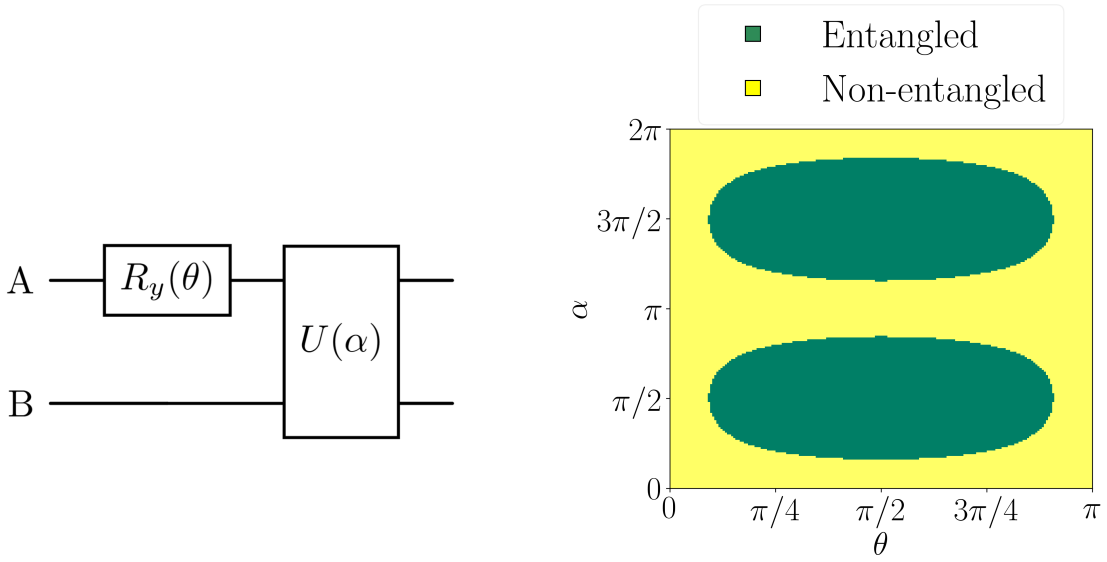
5.2 Encoding Quantum Data

The *quantum input* in this problem is an operator U_i rather than a classical vector $\mathbf{x}_i$. We can still define a quantum feature map $A(U_i)$ as follows. Suppose we already have an "entangling block" U_e, akin to the blocks in Chapter 4. We can *sandwich* U_i around U_e so that

$$V(U_i) \;=\; U_i\, U_e\, U_i^\dagger. \tag{5.1}$$

Next, we define

$$A(U_i) \;=\; V(U_i)\, A_0\, V^\dagger(U_i), \tag{5.2}$$



(a) (b)

Figure 5.1: (a) Quantum circuit depicting the class of input unitaries used for entanglement classification task. The unitary $U(\theta, \alpha)$ is a product of a local and a non-local unitary acting on qubit 2. (b) Entanglement labels. The range of θ and α values for which the unitaries are entangling are colored in green while the non-entangling region is colored in yellow.

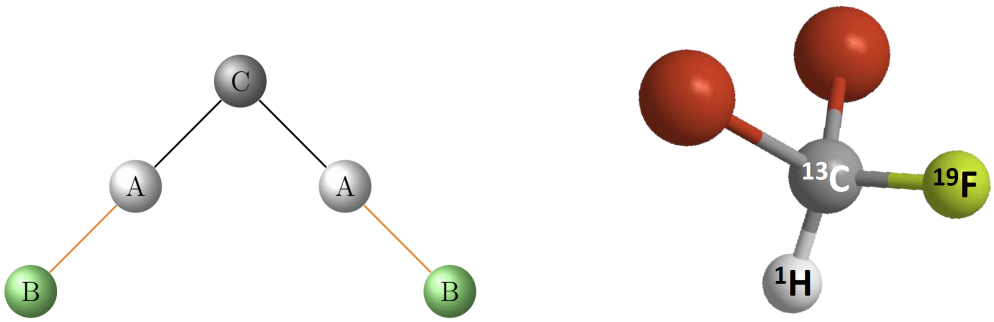
where A_0 is again chosen as some reference operator, typically the z-magnetization of a designated "central" qubit or set of qubits. In other words,

$$A_0 \;=\; I_z^C, \quad \text{where } C \text{ is the central qubit.}$$

Then

$$k(U_i, U_j) \;=\; \mathrm{Tr}\big[A(U_i)\, A(U_j)\big]$$

defines the *quantum kernel* between unitaries U_i and U_j. The physical interpretation is the same as before: If ρ_{eq}^C is the initial state for the central qubit, one effectively applies $V(U_i)$ followed by $V^\dagger(U_j)$ to measure the final z-magnetization on C. This gives the overlap of $A(U_i)$ and $A(U_j)$ in the operator space.



(a) (b)

Figure 5.2: (a) The geometry of the spin register used for the entanglement classification task. The central qubit C is coupled to two ancillary qubits A each of which is coupled to one B. The unitaries U_i act on the A and B qubits, while the z-magnetization of the central qubit is measured. (b) The structure of the Dibromofluoromethane (DBFM) molecule used to realize one branch of the register in (a).

5.2.1 Extended Star-Topology System

Whereas in Chapter 4 we considered a single-layer star system (one central qubit, plus n ancillary qubits), here we may optionally consider a *double-layer star geometry* to allow for additional registers. For instance, one layer of A qubits couples to C and a second layer of B qubits, one coupled to each A qubit, as shown schematically in [42]. This arrangement can be beneficial for tasks like entanglement classification, because the unitaries U_i might act predominantly on the outer (A–B) qubits, while leaving C for readout. Fig. 5.2(a) shows the geometry of such a system with 2 ancillary qubits A and 2 ancillary qubits B.

The entangling block U_e can then be implemented between C and A qubits, and the z-magnetization of C can be measured to compute the kernel. The experimental implementation of the entangling gate U_e follows the pulse sequence: $90^{\circ C}_{y}$-$90^{\circ A}_{y}$-$\tau(\frac{1}{2J_{AC}})$-$90^{\circ C}_{x}$, where τ denotes the evolution period under the J_{AC} coupling between qubits C and A. This sequence effectively realizes a pseudo-Hadamard operation on qubit C followed by a controlled-phase interaction between C and A.

J-coupling
$J_{HC} = 224.5$
$J_{FC} = -310.9$
$J_{HF} = 49.7$

	T_1	T_2^*
H	13.7	0.4
F	5.2	0.2
C	1.9	0.5

Table 5.1: Tabulated values include the spin-spin coupling constants (in Hz) and the characteristic relaxation times T_1 and T_2 (reported in seconds).

5.3 Experimental Details

The experiments have been done on a 500 MHz NMR spectrometer with QXI liquid state probe. Fig. 5.2(b) shows the structure of the Dibromofluoromethane (DBFM) [56] molecule used to realize one branch of the double layered star system. Dibromofluoromethane consists of three nuclear spin qubits ^{1}H, ^{13}C and ^{19}F. Since it is an all to all coupled system, the central qubit is chosen to be ^{31}C and the inner ancillary qubits are ^{1}H amd the outer ancillary qubits are ^{19}F. The solvant used to prepare the solution of DBFM is Acetone-d6 and the experiments are done at room temperature.

The same NMR hardware described in Chapter 4 can implement the circuit $V(U_i)$ in principle, provided we can physically synthesize U_i or $U_i^\dagger$ via standard J-couplings and single-spin gates between A and B qubits. The z-magnetization of the central qubit C is then measured to obtain the kernel value $k(U_i, U_j)$. The kernel matrix can be computed by iterating over all pairs of U_i and U_j in the training set. The readout simply measures $\langle I_z^C \rangle$ by a 90_y° pulse on the central spin followed by a free induction decay (FID) detection.

5.4 Results and Discussion

5.4.1 Classification Protocol

Our classification pipeline is as follows:

1. **Training set:** Identify a subset $\{U_i\}$ of unitaries for which we have labels (entangling vs. nonentangling) obtained by classical negativity measures like logarithmic negativity [57] or by prior knowledge.

2. **Compute kernel matrix:** For each pair (U_i, U_j) in the training set, measure or simulate $k(U_i, U_j)$ as described in Sec. 5.2.

3. **Train classifier:** Feed the kernel matrix K to a standard kernel SVM or kernel logistic regression, where the SVM hyperplane is in an abstract high-dimensional operator space.

4. **Inference on test set:** For a new unitary U^*, measure $k(U^*, U_j)$ for j in the training set, form the row vector $\big(k(U^*, U_1), \ldots, k(U^*, U_N)\big)$, and apply the learned classifier to predict whether U^* is entangling or not.

5.4.2 Numerical Results

We numerically simulate the computation of the kernel on the quantum register shown in Fig. 5.2(a). For the kernel computation, We take a number of these unitaries as a training set, measure (or simulate) $k(U_i, U_j)$, and train an SVM. Then we pick another set of 196 (14×14) unitaries (test set) to see how well the kernel classifies them. As shown in Fig. 5.3 below, we find accuracy of 95%, even for test unitaries *outside* the range of training parameters—an effect we refer to as *extrapolation in operator space*. In Fig. 5.3(b) we show the accuracy when tested on non-parametrized random unitaries which is 82%. This shows that our model is able to generalize well to unseen data. While the training is done on a restricted class of unitaries, the model is able to make correct predictions on catagorically different unitaries. The accuracy as a function of the number of training unitaries is shown in Fig. 5.4(a). The accuracy saturates at around 95% with 30 training unitaries. The accuracy is robust to variations in the delay time τ in the entangling unitary U_e used to construct the kernel, as shown in Fig. 5.4(b).

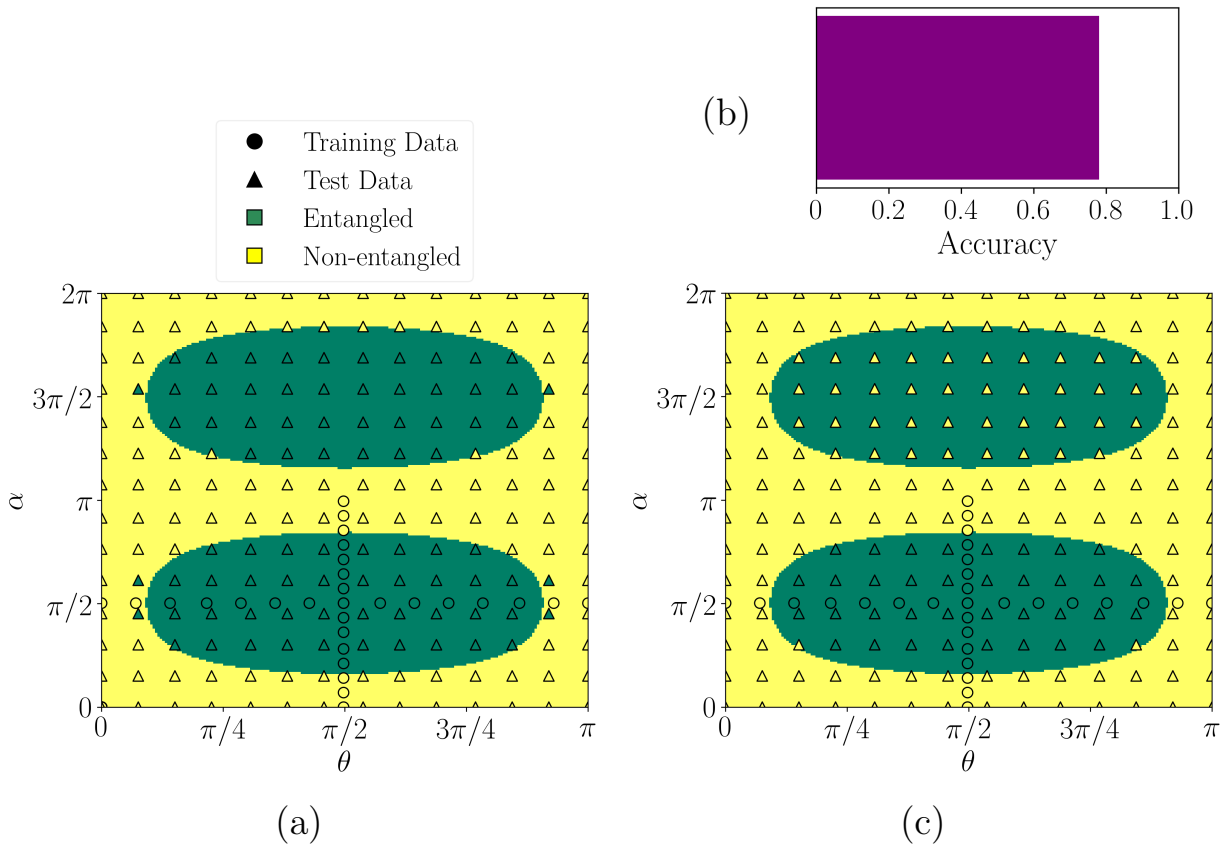


Figure 5.3: Numerical Quantum kernel classification of entangling vs. nonentangling unitaries. ding unitaries U_i parametrized by θ and α. (a) The background indicates the actual classification of entangling (inside elliptical patches) and nonentangling (outside elliptical patches) when acted on the thermal equilibrium state ρ_0. The circles represent 30 training unitary operators, and the triangles are the test unitary operators. (b) The accuracy when tested on non-parametrized random unitaries. (c) Same as (a), except with a classical Gaussian kernel trained on the parameter space (θ, α). A quantum kernel SVM can label new *test* unitaries (marked by triangles) with high accuracy (95%) using only a small training set. Notably, the classifier based on the quantum kernel can discover an upper region of entangling transformations even when no training data is present in that region (extrapolation), while the classical kernel in (c) completely fails to recognize this region.

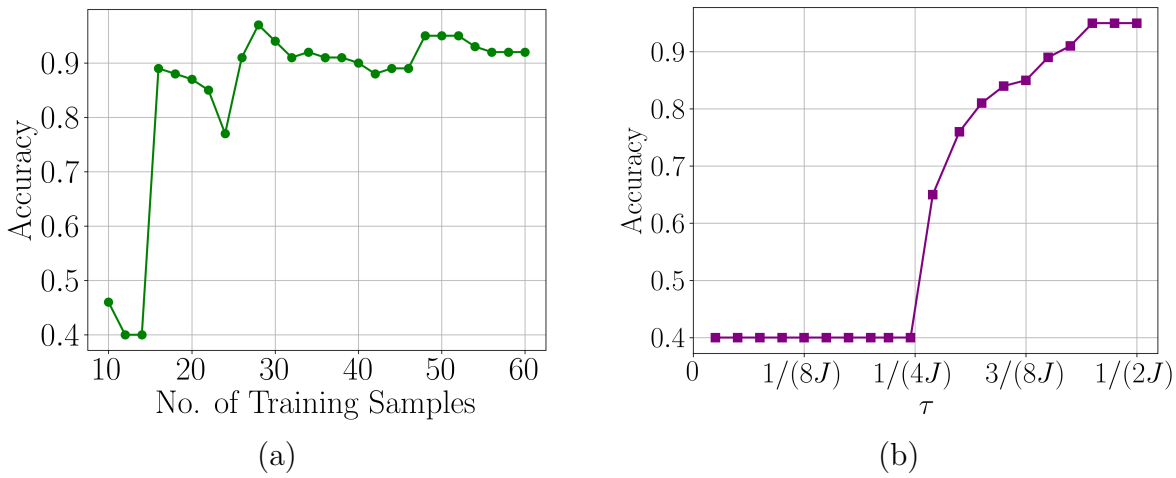


Figure 5.4: (a) Accuracy of the quantum kernel SVM vs. the number of training unitaries. (b) Accuracy vs. the delay time τ used in the entangling unitary U_e. Both saturate at around 95%.

5.4.3 Experimental results

We now turn to the experimental realization of the quantum kernel for entanglement classification. We use the DBFM molecule to realize one branch of the double-layered star system shown in Fig. 5.2(a). We prepare the thermal equilibrium state ρ_0 on the central qubit ^{31}C and the ancillary qubits ^{1}H and ^{19}F. We then apply the unitaries $U(\theta, \alpha)$ as the encoding unitaries on the ancillary qubits ^{1}H and ^{19}F whenever required. The entangling unitary U_e is applied between the central qubit ^{31}C and the ancillary qubit ^{1}H. The z-magnetization of the central qubit ^{31}C is then measured to obtain the kernel value $k(U_i, U_j)$. This readout is done by applying a 90°_y pulse on the central spin followed by a free induction decay (FID) detection. The kernel matrix is computed by iterating over all pairs of U_i and U_j in the training set. Again, 30 unitaries are used as the training set and 196 unitaries are used as the test set. The experimental results are shown in Fig. 5.5. The accuracy of the quantum kernel SVM is 91% which is slightly lower than the numerical results. This discrepancy can be attributed to the imperfections in the pulse sequences used to implement the unitaries $U(\theta, \alpha)$ and U_e.

We also show the heat maps of the kernel values of all testing unitaries with respect to

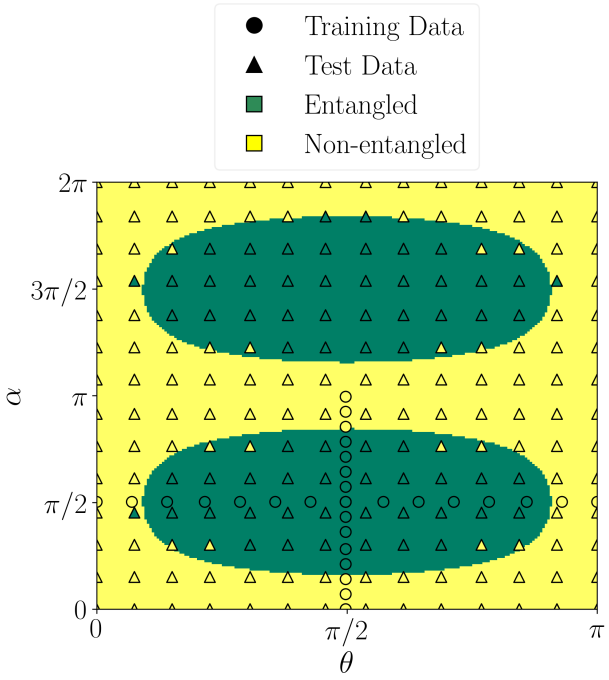


Figure 5.5: Experimental quantum kernel classification of entangling vs. nonentangling unitaries. The description is the same as in Fig. 5.3(a).

each training unitary in Fig. 5.6. The title of each subplot indicates the (θ, α) values of the training unitary. The heat maps show that the kernel values are high for the testing unitaries that are close to the training unitaries in the operator space. In particular, take a look at the training unitary with (θ, α) values of (90, 90). The heat map shows that the kernel values are high for the testing unitaries that are close to (90, 90) as well as (90, 270). This is because the unitaries $U(90, 90)$ and $U(90, 270)$ are equivalent up to a global phase and hence produce the same transformation on the register. This explains why our quantum kernel SVM can classify the test unitaries with high accuracy even when they are outside the range of training parameters. Such an inherent symmetry in the unitaries would be difficult to capture with a classical kernel which are trained on the parameter space.

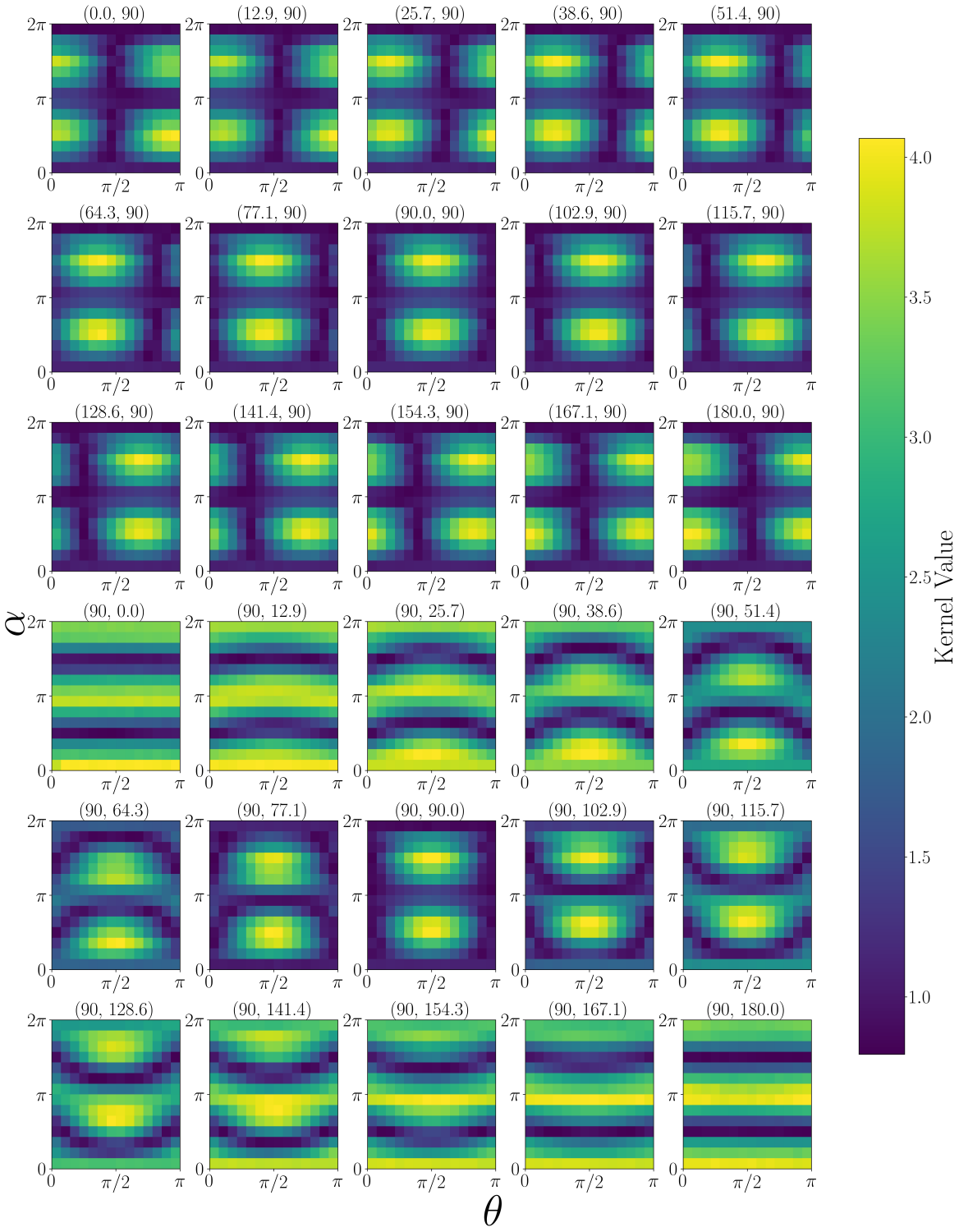


Figure 5.6: Heat maps of kernel values of all testing unitaries with respect to each training unitary. The title of each subplot indicated the (θ, α) values of the training unitary.

5.4.4 Advantages and Outlook

A classical kernel approach would require expressing the unitaries as parameter vectors and learning a decision boundary in that parameter space. But for arbitrary, nonparametric U_i, or for very high-dimensional families, a classical approach struggles. By contrast, the quantum kernel with $k(U_i, U_j) = \mathrm{Tr}[\, A(U_i)\, A(U_j)]$ operates directly in operator space—*without* any explicit parameterization which is evident from the model's ability to handle random unitaries. One can show that unitaries that produce similar transformations on the register yield large kernel values from Fig. 5.6. Hence, once the kernel is trained, it can efficiently classify new unitaries that may differ from the training set in many parameters, as long as they yield similar transformations in the extended star-topology system. This approach is promising for tasks such as verifying quantum gates, detecting quantum phase transitions, or diagnosing multi-qubit operations without performing full tomography [42].

5.5 Summary

In this chapter, we introduced and analyzed *quantum kernels* for *quantum inputs*: specifically, how to map each input unitary U into an operator $A(U)$ on an NMR spin register, and thereby define a kernel function $k(U_i, U_j)$. We argued that this can be realized on a star-topology or extended star-topology architecture, using readily available NMR pulse sequences. A key advantage over classical kernels is the ability to handle large or non-parameterized unitaries in a high-dimensional operator space. Our numerical simulations on entanglement classification indicate that the kernel-based classifier can learn to distinguish entangling transformations with high accuracy, even extrapolating beyond the parameter region of the training set. Future experimental work on an extended star-topology molecule will aim to implement this scheme directly and compare the classification success with standard measures of entanglement generation.

Chapter 6

Conclusion

This thesis has demonstrated the potential of integrating quantum machine learning with NMR quantum systems by developing and experimentally validating quantum kernel methods. The work advances the state-of-the-art in quantum information processing by leveraging multiple quantum coherences (MQCs) in NMR to create high-dimensional quantum feature spaces, thereby providing a novel approach to both classical and quantum learning tasks.

The research began with a comprehensive analysis of MQCs in both solid-state and liquid-state NMR systems, establishing a robust experimental framework to generate and detect these coherences. By mapping input data into the operator space of quantum systems via data-dependent unitary transformations that generate the MQCs, a practical realization of quantum feature mapping was achieved. This enabled the construction of quantum kernels that capture intricate correlations among the input data.

Building on this foundation, two main applications were explored. First, quantum kernels for classical data were implemented and tested on standard machine learning tasks such as one-dimensional regression and two-dimensional classification. The experimental results not only validate the theoretical underpinnings of kernel methods but also demonstrate that the quantum kernels, extracted from the dynamics of MQCs, can successfully reproduce

the performance of their classical counterparts while operating in an exponentially larger feature space. Second, the approach was extended to quantum data by designing quantum kernels capable of handling non-parametrized unitary inputs. This extension was applied to the entanglement classification problem, where the proposed kernel showed strong potential in distinguishing entangling from nonentangling unitaries with high accuracy. The experimental results on the NMR quantum register confirmed the feasibility of implementing quantum kernels for quantum data, paving the way for future applications in quantum machine learning. The work also highlighted the the *quantum advantage* in using quantum kernels over classical methods, particularly in terms of their ability to extrapolate and handle non-parametric data efficiently.

The quantum data used in this work is limitted to the unitary operators. A future direction in this regard could be developing a model that can handle non-unitary quantum maps as inputs. This is an ongoing project that we are currently working on. While the present study lays a strong foundation for the use of quantum kernels, scaling these techniques to larger qubit systems and exploring their integration with error-mitigation strategies is essential. Such efforts will be crucial for transitioning from proof-of-concept experiments to practical, real-world quantum learning applications.

Overall, this work contributes to the growing body of knowledge at the intersection of quantum information and machine learning, opening promising avenues for the development of fully quantum learning systems that leverage the natural high-dimensionality and entanglement present in quantum systems.

Bibliography

[1] Takeru Kusumoto, Kosuke Mitarai, Keisuke Fujii, Masahiro Kitagawa, and Makoto Negoro. Experimental quantum kernel trick with nuclear spins in a solid. *npj Quantum Information*, 7(1):94, 2021.

[2] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[3] Lieven M. K. Vandersypen and Isaac L. Chuang. NMR techniques for quantum control and computation. *Rev. Mod. Phys.*, 76:1037–1069, 2005.

[4] Ashley Montanaro. Quantum algorithms: an overview. *NPJ Quantum Information*, 2:15023, 2016.

[5] T. S. Mahesh. Quantum information processing by nmr. *Resonance*, 20(11):1053–1065, 2015. Published: 2015/11/01.

[6] Neil A. Gershenfeld and Isaac L. Chuang. Bulk spin-resonance quantum computation. *Science*, 275:350–356, 1997.

[7] D. G. Cory, A. F. Fahmy, and T. F. Havel. Ensemble quantum computing by nmr spectroscopy. *Proc. Natl. Acad. Sci. U.S.A.*, 94:1634–1639, 1997.

[8] Malcolm H. Levitt. *Spin Dynamics: Basics of Nuclear Magnetic Resonance (2nd ed.)*. John Wiley & Sons, Chichester, UK, 2008.

[9] Melinda J. Duer. *Introduction to Solid-State NMR Spectroscopy.* Blackwell Science, Oxford, UK, 2004.

[10] D. G. Cory, M. D. Price, W. Maas, E. Knill, R. Laflamme, W. H. Zurek, T. F. Havel, and S. S. Somaroo. Experimental quantum error correction. *Phys. Rev. Lett.*, 81:2152–2155, 1998.

[11] E. B. Fel'dman and A. N. Pyrkov. Evolution of spin entanglement and an entanglement witness in multiple-quantum nmr experiments. *JETP Letters*, 88:398–401, 2008.

[12] Jiangyu Cui, Jun Li, Xiaomei Liu, Xinhua Peng, and Riqiang Fu. Engineering spin hamiltonians using multiple pulse sequences in solid state nmr spectroscopy. *Journal of Magnetic Resonance*, 294:83–92, 2018.

[13] J. Baum, M. Munowitz, A. N. Garroway, and A. Pines. Multiple-quantum dynamics in solid state nmr. *J. Chem. Phys.*, 83:2015–2025, 1985.

[14] Yasushi Kondo. Liquid-state nmr quantum computer: Working principle and some examples. *Journal of Magnetic Resonance*, 200:123–134, 2009.

[15] John Cavanagh, Wayne J. Fairbrother, Arthur G. Palmer, and Nicholas J. Rance. *Protein NMR Spectroscopy: Principles and Practice.* Academic Press, San Diego, CA, 2nd edition, 2010.

[16] J. A. Jones, M. Mosca, and R. H. Hansen. Implementation of a quantum search algorithm on a nuclear magnetic resonance quantum computer. *Nature*, 393:344–346, 1998.

[17] Ranabir Das, T.S. Mahesh, and Anil Kumar. Experimental implementation of grover's search algorithm using efficient quantum state tomography. *Chemical Physics Letters*, 369(1):8–15, 2003.

[18] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang. Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414:883–887, 2001.

[19] C. A. Ryan, M. Laforest, J. C. Boileau, and R. Laflamme. Experimental implementation of discrete time quantum random walk on an nmr quantum information processor. *Physical Review A*, 72(6):062317, 2005.

[20] R. Neigovzen, J. L. Neves, R. Sollacher, and S. J. Glaser. Quantum pattern recognition with liquid-state nuclear magnetic resonance. *Physical Review A*, 77(6):062327, 2008.

[21] X.-D. Cai, C. Weedbrook, Z.-E. Su, M.-C. Chen, M. Gu, M.-J. Zhu, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan. Experimental quantum computing to solve systems of linear equations. *Physical Review Letters*, 110(23):230501, 2013.

[22] C. A. Ryan, C. Negrevergne, M. Laforest, E. Knill, and R. Laflamme. Liquid state nmr as a test-bed for developing quantum control methods. *Physical Review Letters*, 100(17):170501, 2008.

[23] Warren S. Warren. The usefulness of nmr quantum computing. *Science*, 277:1688–1689, 1997.

[24] Samuel L. Braunstein, Carlton M. Caves, Richard Jozsa, Noah Linden, Sandu Popescu, and Rüdiger Schack. Separability of very mixed states and implications for NMR quantum computing. *Phys. Rev. Lett.*, 83:1054–1057, 1999.

[25] T. S. Mahesh, Deepak Khurana, Krithika V R, Sreejith G J, and Sudheer Kumar. Star-topology registers: Nmr and quantum information perspectives. *Journal of Physics: Condensed Matter*, 33, 2021.

[26] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, 2009.

[27] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Phys.*, 10:631–633, 2014.

[28] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Phys. Rev. Lett.*, 113:130503, 2014.

[29] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549:195–202, 2017.

[30] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemp. Phys.*, 56:172–185, 2015.

[31] V. Havlíček, A. D. Córcoles, K. Temme, and et al. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567:209–212, 2019.

[32] Marco Cerezo, Alex Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3:625–644, 2021.

[33] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature Communications*, 12:2631, 2021.

[34] Sofiene Jerbi, Lukas J Fiderer, Hendrik Poulsen Nautrup, Jonas M Kübler, Hans J Briegel, and Vedran Dunjko. Quantum machine learning beyond kernel methods. *Nature Communications*, 14, 2023.

[35] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2002.

[36] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[37] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.*, 122:040504, 2019.

[38] Maria Schuld. Supervised quantum machine learning models are kernel methods. *Quantum*, 5:531, 2021.

[39] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Phys.*, 17:1013–1017, 2021.

[40] Karol Bartkiewicz et al. Experimental kernel-based quantum machine learning in finite feature space. *Scientific Reports*, 9:63, 2019.

[41] E. Peters et al. Machine learning of high dimensional data on a noisy quantum processor. *npj Quantum Information*, 7:86, 2021.

[42] Vivek Sabarad and T. S. Mahesh. Experimental machine learning with classical and quantum data via nmr quantum kernels, 2024.

[43] G. B. Furman, V. M. Meerovich, and V. L. Sokolovsky. Entanglement and multiple quantum coherence dynamics in spin clusters. *Quantum Inf Process*, 8:379–386, 2009.

[44] Ken X. Wei, Isaac Lauer, Srikanth Srinivasan, Neereja Sundaresan, Douglas T. McClure, David Toyli, David C. McKay, Jay M. Gambetta, and Sarah Sheldon. Verifying multipartite entangled greenberger-horne-zeilinger states via multiple quantum coherences. *Phys. Rev. A*, 101:032343, 2020.

[45] Gonzalo A. Álvarez and Dieter Suter. Localization effects induced by decoherence in superpositions of many-spin quantum states. *Phys. Rev. A*, 84:012320, Jul 2011.

[46] Gonzalo A. Álvarez, Dieter Suter, and Robin Kaiser. Supplementary materials for "localization-delocalization transition in the dynamics of dipolar-coupled nuclear spins". Supplementary material for Science, **349**(6250), 846–848, 2015.

[47] W. S. Warren, S. Sinton, D. P. Weitekamp, and A. Pines. Selective excitation of multiple-quantum coherence in nuclear magnetic resonance. *Phys. Rev. Lett.*, 43:1791–1794, Dec 1979.

[48] J.A. Jones, R.H. Hansen, and M. Mosca. Quantum logic gates and nuclear magnetic resonance pulse sequences. *Journal of Magnetic Resonance*, 135(2):353–360, 1998.

[49] George A. F. Seber and Alan J. Lee. *Linear Regression Analysis*. Wiley, 2 edition, 2012.

[50] Douglas C. Montgomery, Elizabeth A. Peck, and Geoffrey G. Vining. *Introduction to Linear Regression Analysis*. Wiley, 5 edition, 2012.

[51] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171 – 1220, 2008.

[52] Maria Schuld. Quantum models as kernel methods. In *Machine Learning with Quantum Computers*, pages 125–144. Springer, Cham, 2021.

[53] Q. L. Wang, Y. Jin, X. H. Li, et al. An advanced quantum support vector machine for power quality disturbance detection and identification. *EPJ Quantum Technology*, 11:70, 2024.

[54] John B. Conway. A course in functional analysis. *Springer-Verlag*, pages 267–268, 1990. Discusses Hilbert–Schmidt operators and their properties.

[55] Eduardo Serrano-Ensástiga and John Martin. Maximum entanglement of mixed symmetric states under unitary transformations. *SciPost Physics*, 15:120, 2023.

[56] Swathi Hegde, K. Rao, and T S Mahesh. Pauli decomposition over commuting subsets: Applications in gate synthesis, state preparation, and quantum simulations. *arXiv:1603.06867*, 03 2016.

[57] M. B. Plenio. Logarithmic negativity: A full entanglement monotone that is not convex. *Physical Review Letters*, 95(9):090503, 2005.