Quantum Machine Learning in Noisy Intermediate-Scale Quantum Era

Fangjun Hu

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of Electrical and Computer Engineering
Advisor: Hakan E. Türeci

September 2025

Abstract

With the rapid advancement of quantum technologies, the application of quantum computing to machine learning (ML) tasks has attracted growing interest, especially in the Noisy Intermediate-Scale Quantum (NISQ) era. The emerging field of Quantum Machine Learning (QML) explores how quantum resources can benefit ML, yet concrete advantages over classical methods on classical data remain elusive. In practice, QML faces several major obstacles: barren plateaus – a phenomenon where the loss function gradient vanishes – hinder effective training, while quantum shot noise – uncertainty from finite sampling – limits both training and generalization accuracy. Additionally, the limited coherence time of NISQ devices poses challenges for learning on temporal or streaming data. Among various QML approaches, Quantum Reservoir Computing (QRC) has emerged as a promising alternative. Inspired by classical recurrent neural networks and quantum kernel methods, QRC circumvents the need for extensive parameter training of deep quantum circuits, thereby avoiding barren plateaus. However, QRC remains affected by quantum noise and decoherence.

This thesis introduces theoretical frameworks – Resolvable Expressive Capacity and Eigentask Analysis – which rigorously quantify the expressive power of QRC under shot noise. By identifying the noise-resilient features of the function space of QRC, this framework provides practical guidelines for enhancing generalization in noisy quantum systems. It also reveals deep connections between QML, quantum metrology, and quantum dynamics, opening new avenues for interdisciplinary research. Furthermore, I present a hardware-compatible implementation, NISQ Reservoir Computing (NISQRC), which uses partial measurements and deterministic resets to realize QRC with a finite temporal memory that persists indefinitely. Finally, I explore a novel QML optimization strategy – Reservoir Gradient Descent – that provides a surrogate loss landscape derived from the reservoir in the presence of shot noise, by fully utilizing the convexity in output layers of QML. Most of these theoretical investigations have been experimentally validated on state-of-the-art superconducting quantum platforms.

These approaches enable efficient parameter training, stronger generalization ability, and longer memory persistence. They provide powerful tools to more robust and higher-performance QML algorithms suited for current and near-term quantum hardware.

Acknowledgements

"Seek knowledge even if you have to go as far as Princeton." My decision to apply for an overseas doctoral program during my final undergraduate year was a hasty one. Six years ago, I could not have imagined that I would spend such a significant portion of my life – nearly one-fifth of it – in this beautiful and quiet town. Over these years, I have constantly reflected on what I, as a physicist – particularly a quantum physicist – can contribute to the field of artificial intelligence, and conversely, how artificial intelligence can aid quantum computing. I am grateful that in the past few years, the answers to these questions have gradually come into focus.

Throughout this six-year journey, my advisor, Professor Hakan Türeci, has been a steadfast guide, leading me from one discovery to the next. His insight helps me identify open questions in quantum machine learning across various scenarios. Whenever I felt uncertain about the direction of my work, he was always there to patiently brainstorm and help me push forward. The diverse directions he encouraged ultimately led to the formation of different chapters in this dissertation. Beyond research, he also took the time and effort to cultivate my broader scientific skills. I am especially thankful to him and to Janet Kayfetz for their generous and invaluable mentorship in improving my writing and presentation abilities.

I am deeply grateful to Professor Xun Gao from the University of Colorado Boulder, whom I met during his visit to Princeton two years ago. Our collaboration has significantly broadened my perspective on the interdisciplinary intersections of quantum computing, statistical physics, and artificial intelligence. Working with him has been both intellectually enriching and thoroughly enjoyable. I also thank the members of my thesis committee – Sarang Gopalakrishnan, Dmitry Abamin, and Alejandro Rodriguez – for their time and feedback. In particular, I appreciate Sarang's insightful suggestions, many of which inspired several of the small projects included in this dissertation.

When I first joined Hakan's group, there were only two senior graduate students: Saeed Khan and Gerry Angelatos. They were more than colleagues – they were mentors. I learned

many essential skills from them, and their help was instrumental in advancing all the big projects I participated in. I would also like to thank the rest of the group – Marti Vives, Tianyang Chen, Theodoros Ilias, Elif Cuce, Wentao Fan, Dzung Pham, Tommy Maldonado, Zoe Zager, Leon Bello, Emily Linden, Ben Lienhard, and Kanu Sinha – for the stimulating discussions and collaborative spirit.

In the final year of my PhD, I had the precious opportunity to intern at QuEra Computing, where I experienced for the first time the synergy between quantum theory and experiment. I am especially thankful to my mentors there – Shengtao Wang, Milan Kornjača, Pedro Lopes, Fangli Liu – and to other QuEra colleagues, for offering both the freedom and resources to explore the applications of quantum computation to machine learning. During my time in Boston, I was fortunate to connect with many brilliant minds in the quantum community. Conversations with Soonwon Choi, Di Luo, Susanne Yelin, Hong-Ye Hu, Sitan Chen, and Weiyuan Gong opened my eyes to new directions and significantly expanded my understanding of quantum computing.

I am also very grateful to Professor Liang Jiang for inviting me to visit the University of Chicago this April. During my stay, I enjoyed the warm hospitality of Su-un Lee, Bikun Li, Guo Zheng, Pei Zeng, Yesun Huang, Allen Zang, and other group members. These exchanges inspired many new research ideas that I look forward to pursuing.

My sincere thanks go to my other collaborators: Quntao Zhuang, Bingzhi Zhang, and Runzhe Mo at the University of Southern California; Nicholas Bronn at IBM Quantum; Graham Rowlands and Guilhem Ribeill at RTX BBN Technologies; Yifan Zhang at Princeton University; and Guangkuo Liu at the University of Colorado Boulder. It was a pleasure working with all of you.

There are not many recreational activities in Princeton. Outside of academic life, I found great joy in weekly board game nights with Yuxiao Jiang, Jiayi Hu, Bingzheng Han, Yangsibo Huang, Fan Yi, Kehan Cai, Xiyuan Zhang, Tianhao Li, Jinxin Sun, Zhihan Liu, Hao Wang, Genyue Liu, Tianshu Wang, and others. My sincere thanks to Haoran Wan, Fan

Yi, Bingzheng Han, Yunxiang Chi, and Zhihan Liu for dragging me to the gym every day. I also appreciate every friend who kindly drove me to Edison for Chinese meals, ensuring that I could continue enjoying authentic food despite not owning a car during these six years. Thanks to all of my roommates in Princeton – especially Ming Lyu, who offered great help during my early days here. I am also thankful to Shiyuan Ma, Yiqing Zhou, Yikun Jiang, Yu Zhou, Yang Hu, Yuxiang Peng, and Yunchao Liu for making my stay in Boston this cold winter so enjoyable. In the end, I owe all my achievements to my family, whose unconditional support has sustained me through all the challenges of studying abroad.

To my grandma.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Machine Learning with Quantum Systems

The notion of creating artificial beings with human-like intelligence has ancient roots, appearing in myths and legends from civilizations such as ancient Greece, where stories tell of automata imbued with life and agency. While these early creations of human intellect were symbolic and aspirational in their nature, the scientific pursuit of artificial intelligence began earnestly in the mid-20th century, following foundational work in formal logic, computation, and cybernetics. In the decades since, especially over the past ten years, AI research has accelerated dramatically. Key breakthroughs, including transformers [130], diffusion models [47, 122], large language models [10], and scaling laws [68], have reshaped the field. These advances have given rise to transformative applications including AlphaGo [119], AlphaFold [65], and ChatGPT [97], revolutionizing both scientific inquiry and everyday life.

Despite the remarkable progress in AI, state-of-the-art systems demand substantial computational resources, including high volumes of floating-point operations and significant energy consumption. A pressing practical challenge is to develop more energy-efficient hardware to support AI workloads. One promising direction is to harness physical systems whose intrinsic dynamics exhibit temporal correlations that naturally align with statistical correla-

tions in the data extracted from the physical domain. This basic idea underlies the emerging paradigm of Physical Neural Networks (PNNs) [54, 80, 92, 137], in which diverse physical substrates are used to perform trainable transformations on input signals from the physical domain.

Quantum systems represent a compelling class of physical systems for implementing physical neural networks (PNNs), owing to the rich statistical correlation structure in their dynamics. Quantum systems should not be viewed merely as a distinct class of physical substrates, but rather as systems whose evolution and measurable outputs are fundamentally governed by the laws of quantum mechanics rather than classical physics. Leveraging quantum systems for computing has a history of more than 60 years [33]. Theoretical work on quantum algorithms leverage superposition and entanglement between quantum bits (qubits) to outperform classical algorithms in specific tasks, such as integer factorization [118] and unstructured search [42]. On the experimental side, over the past several years, quantum computers have been built on various platforms, including ion traps [20], superconducting [3], and neutral atom systems [7].

Quantum Neural Networks (QNNs) have received considerable early theoretical attention as a potential class of PNNs for scalable and resource-efficient machine learning [17, 22, 45, 58, 85, 104, 109, 136], due to their evolution taking place in the Hilbert space that scales exponentially with the number of nodes [19, 34, 66, 89, 96, 101, 140]. In the past decade, many topics relating to quantum machine learning (QML) algorithms [4, 109] have been extensively studied, including, for example, quantum kernel methods [45, 108], parameterized quantum circuits [23, 31, 32, 111] and quantum generative models [76].

Despite extensive research, the advantages of quantum computing for machine learning tasks on classical data remain uncertain. On current noisy intermediate-scale quantum (NISQ) hardware, quantum machine learning has largely been limited to training and inference on low-dimensional, static datasets. Practical implementation is hindered by several challenges, including the phenomenon of barren plateaus in training [84], circuit noise [124,

132], gate errors [25], fundamental measurement noise [2], and limited coherence times [140]. This dissertation focuses on a principled analysis of limitations posed by measurement noise and decoherence, and establishment of rigorous metrics and methods of analysis to design PNNs for supervised machine learning on classical data. This theoretical progress culminates in the development of the NISQRC algorithm, presented in Chapter 4, which recently enabled a proof-of-principle demonstration [52] of inference on streaming data exceeding the coherence time of a superconducting quantum processor.

Moreover, the limited coherence time of NISQ devices also raises a challenge for QML in applications when tackling online learning task that has indefinitely long streaming data. In Chapter 2, 3 and 4, I will show how I contribute to the theoretical tools, including Resolvable Expressive Capacity, Eigentask Learning [54], and NISQ Reservoir Computing [52], to understand and resolve these problems in a quantitative way.

1.2 Supervised Quantum Machine Learning and Quantum Reservoir Computing

The idea of a supervised quantum machine learning algorithm is very similar to the generic supervised machine learning in classical cases. Therefore, before introducing the supervised QML, I will begin with a brief introduction to the background of supervised classical ML.

A generic supervised machine learning algorithm typically utilizes a complicated map called a *neural network* to map the inputs to output an outcome that approximates a target value that is associated with this input. In this thesis, we will represent the input by a high-dimensional vector $\boldsymbol{u}$, which can be static in time (e.g., in Chapter 2, 3, and 5) or time-dependent (Chapter 4). And the collection of different inputs constitutes the training, validation, and testing datasets. For simplicity, in this dissertation, we only consider the training and testing datasets. For example, suppose we have a dataset of images of cats and dogs, and we want to classify a given image selected from this dataset. Then, we feed

the input $\boldsymbol{u}$ into a deep neural network which is parametrized by some internal parameter $\boldsymbol{\theta}$. The neural network should process the information and output an outcome y to fit the target $y^\star$ (e.g., the target in this case of cats and dogs is the label of images). A loss function $\mathcal{L}(y, y^\star)$ is defined to quantify the performance of such an approximation, and the internal parameters are trained by some method of optimization, like the gradient descent method through the Backpropagation algorithm, to optimize the performance of the approximation.

In supervised quantum machine learning, one just needs to replace the classical neural network in generic supervised machine learning with a parametrized quantum circuit (PQC, some papers also refer to it as quantum neural network or QNN). We feed input $\boldsymbol{u}$ into the PQC. This circuit depends not only on the $\boldsymbol{u}$ but also on some internal parameters $\boldsymbol{\theta}$, like the rotational angles in a quantum circuit or the couplings and drives in a quantum spin network. The information is then accessible from the parametrized quantum circuit via measuring some quantum observables. The readout information is classically post-processed to execute some desired tasks like fitting functions and classification. In analogy to classical machine learning, QML also trains internal parameters and output weights by using gradient descent to optimize the behavior of the parametrized quantum circuit. Fig. 1.1 shows a simplified schematic of quantum machine learning.

As we pointed out in Sec. 1.1, the first fundamental restriction of the optimization landscape for training quantum systems often features "barren plateaus" [84, 133], which are regions where optimization becomes exponentially difficult. Heuristically, the phenomenon of barren plateaus claims that if the quantum circuits are complicated enough, then the loss function of them will have a landscape whose gradient covariance vanishes exponentially. This means we need huge measurement shots or state copies to train the parameters theta, which is completely infeasible in NISQ devices.

One scheme to avoid such difficulty is *Quantum Reservoir Computing* (QRC) [19, 26, 34, 35, 67, 136], which is inspired by classical recurrent neural networks and kernel methods. In QRC, we can randomly draw the internal parameters $\boldsymbol{\theta}$ inside the quantum circuits and

keep them fixed. Then we only need to train the classical post-processing, usually the output weights $\boldsymbol{w}$. The expressive ability may decrease, but we can use a larger qubit number to compensate for it. Usually, the classical optimization in QRC is convex, and no feedback loop is needed. It is believed to be one of the easiest realizable quantum machine learning algorithms in NISQ devices. A simplified schematic of quantum reservoir computing is also depicted in Fig. 1.1.

Additionally, despite the absence of barren plateaus, QRC still suffers from the Quantum Shot Noise (QSN) even on fault-tolerant hardware. These still present a significant challenge to implementing QRC at scales relevant to practical applications. Moreover, when dealing with the machine learning task with temporal input-output, the inputs are so long that embedding them in the quantum system already requires very deep quantum circuits. When the time for encoding and processing is far longer than the quantum computer's coherence time, you will lose all information by the time you measure it. One basic question is whether quantum machine learning algorithms can maintain the memory persistency of a quantum computer when we need to retrieve information from the computer, even if we don't have a perfect fault-tolerant quantum computer yet. Now, let me elaborate on these two main challenges more thoroughly in Sec. 1.3 and Sec. 1.4.

1.3 Shot Noise in Learning with Physical Systems

1.3.1 Physical learning

A physical system receiving an input stimulus typically evolves in response to it, such that its degrees of freedom become dependent on said input after a certain period of interaction with it. This everyday observation has a profound implication: any dynamical system can be viewed as performing a transformation of its input, realizing an input-output map [8]. This functional map can in principle be optimized, inspiring an emerging approach to learning with analog physical systems, which we will collectively refer to as Physical Neural Networks

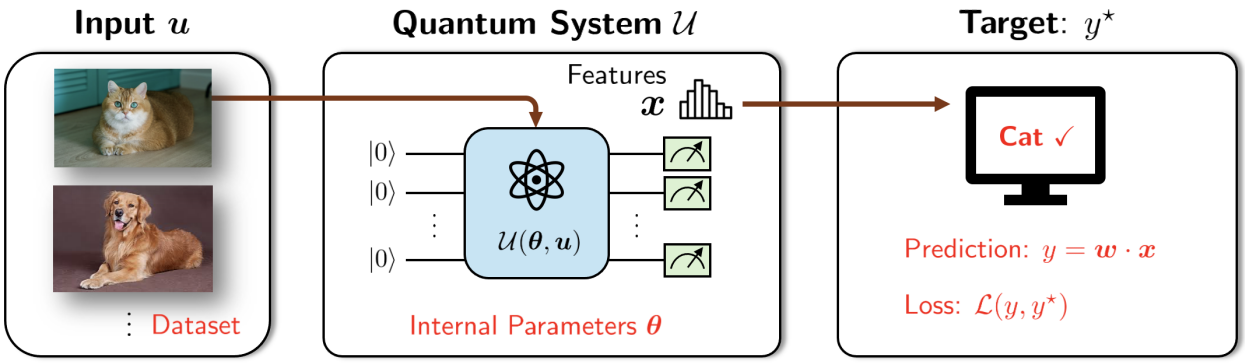

Figure 1.1: **The schematic of quantum machine learning and quantum reservoir computing**. Both QML and QRC consist of classical inputs, a quantum system, and a target. The only difference between QML and QRC is the parameters that need to be optimized. Unlike the ordinary machine learning architecture, where all internal parameters $\boldsymbol{\theta}$ and output layer weights $\boldsymbol{w}$ are trained, QRC only trains the weights $\boldsymbol{w}$ in the output layer or classical post-processing, and the optimization is usually convex and therefore efficient. We specify that there is a benefit in dividing parameters of a learning machine into two types, $\boldsymbol{\theta}$ and $\boldsymbol{w}$. Chapter 2, 3 and 4 are geared towards understanding the capacity of a given physical system (not only in a QRC algorithm, but also in *any* schemes where internal parameters are already optimized with some unspecified method), whereas Chapter 5 will investigate a novel method for the efficient optimization of internal parameters, referred to as "reservoir-based optimization"

(PNN) [80, 92, 138]. PNNs employ a wide variety of analog physical systems to compute a trainable transformation on an input signal [13, 17, 35, 45, 74, 77, 89, 99, 100, 103, 117, 127, 134]. More precisely, the role of an idealized (*i.e.* completely deterministic, noise-free) physical system in these approaches is that of a high-dimensional feature generator. Given inputs $\boldsymbol{u}$, the measured degrees of freedom $x_j(\boldsymbol{u})$ for $j \in [K]$ with total feature number K, generated by the system, act as an input-dependent vector of features. These features are used to approximate a function $f(\boldsymbol{u})$ via a learned linear projection with sufficient accuracy, as dictated by a chosen loss function (See Fig. 1.2). Different characteristics of the physical system, described by a set of hyperparameters $\boldsymbol{\theta}$, may determine its ability to approximate a particular function. Consequently, the relationship between a specific physical system and the classes of functions it can express with high accuracy is a fundamental question in this paradigm of machine learning [26, 100, 110, 114, 136, 139].

No physical system however exists in isolation, and is therefore necessarily subject to noise. Noise can enter at the input, whereby it evolves under the same dynamical law governing the evolution of the physical system. There may also be variability in this very dynamics of the physical system itself. Finally, there is typically noise associated with the measurement of output features from the physical system. As a consequence of these noise sources, the resulting feature map is stochastic: even under identical preparations and inputs $\boldsymbol{u}$, the outcome of a measurement $X_j^{(s)}(\boldsymbol{u})$ of a feature j can vary between repetitions, each of which is referred to as a "shot" s. By empirically averaging the outcomes of S shots, one can generally reduce this stochasticity. We will refer to the resulting noise as "shot noise". And the actually readout feature should be the stochastic version of $\boldsymbol{x}_j(\boldsymbol{u})$, which we denoted as $\boldsymbol{X}_j(\boldsymbol{u})$.

The general input-output relationship $\boldsymbol{u} \to X_j(\boldsymbol{u})$ above can be made concrete by considering three example physical systems, depicted in Fig. 1.2. For an *optical system*, the input $\boldsymbol{u}$ could for instance be embedded as a collection of pixel values on a spatial light modulator (SLM) in the path of a propagating beam of light. The individual single-shot features $\{X_j^{(s)}\}$ could be generated by integrating the photocurrent from each pixel of a number-resolving CCD camera for a certain hold time. For a *biological neural circuit*, the input $\boldsymbol{u}$ might be a static-in-time visual stimulus, representing the electromagnetic field intensity incident on photoreceptors in the eye, and $\{X_j^{(s)}(\boldsymbol{u})\}$ can be the action potential of the kth neuron integrated over a certain time-period, e.g. measured through Ca^{2+} imaging [41]. Finally, for a *superconducting quantum processor*, inputs may be embedded via a suitable quantum channel, implemented for example via parameterized quantum gates. The single-shot features are simply the indicator functions of the possible outcome labels after quantum measurement. In all cases, the measured features $X_j(\boldsymbol{u})$ may be obtained by repeating each experiment S times with the same $\boldsymbol{u}$ and constructing the S-shot histogram.

Theoretical analysis and experimental implementations of PNNs have already demonstrated that shot noise can have a substantial role in the ultimate performance of a physical