An Introduction to Quantum Machine Learning

Jeongbin Jo[1,*]

[1]*Department of Physics, Yonsei University, Seoul, 03722, Republic of Korea*

(Dated: June 12, 2026)

Quantum machine learning (QML) sits at the intersection of quantum information science and modern artificial intelligence, offering a promising route toward computational advantages that are inaccessible by classical means alone. This report provides a self-contained introduction to the theoretical foundations necessary to understand and critically evaluate QML proposals. Starting from the postulates of quantum mechanics—state, dynamics, and measurement—we develop the quantum circuit model with single- and multi-qubit gates, establish the universality of quantum gate sets, and examine the no-cloning theorem and elementary quantum communication protocols. We then discuss reversible computation, Landauer's principle, the Hadamard and swap tests with complete circuit derivations, and the basics of quantum computational complexity, culminating in a tutorial treatment of the HHL linear-systems algorithm, the classical SVM and least-squares SVM formulations that reduce training to a linear system, their connection to quantum speedups, and the variational quantum machine learning framework—covering variational quantum eigensolvers, quantum convolutional neural networks, the universal approximation theorem for quantum circuits, the parameter-shift rule for exact gradient estimation, and quantum state discrimination via the Helstrom bound. We then turn to quantum approaches to discrete optimization, developing the QUBO framework and its Ising Hamiltonian encoding. The quantum adiabatic theorem is derived in detail: expanding in the instantaneous eigenbasis and performing integration by parts yields the quantitative condition $T \gg \hbar \max_s |\langle E_m(s)|\partial_s H|E_0(s)\rangle|/g_{\min}^2$. The quantum approximate optimization algorithm (QAOA) is then derived from adiabatic evolution via the first-order Lie–Trotter and second-order symmetric Suzuki–Trotter product formulas; Taylor expansions establish $O(\Delta t^2)$ and $O(\Delta t^3)$ local truncation errors respectively, and the global error bounds $O(t^2/N)$ and $O(t^3/N^2)$ are proved via the telescoping identity. Throughout, formal derivations are presented at a level of detail beyond standard textbook treatments, with the aim of equipping the reader with both intuition and rigorous mathematical foundations.

CONTENTS

I. Introduction 2
A. Motivation: The Age of Quantum Computing 2
B. Classical vs. Quantum Information 2
C. Machine Learning as an Optimization Problem 2
D. Scope and Organization of This Report 3

II. Mathematical Preliminaries 3
A. Dirac Notation and Hilbert Space 3
B. Asymptotic Complexity Notation 3
C. Classical Information: Probabilistic States and Stochastic Matrices 4
D. Optimization Preliminaries 4

III. Postulates of Quantum Mechanics 5
A. Postulate I: State 5
B. Postulate II: Dynamics 5
C. Postulate III: Measurement 5

IV. Multi-Qubit Systems 5
A. Composite Systems and Tensor Products 5
B. Quantum Entanglement 6
C. Partial Measurement 6
D. Pauli Matrices 6
E. Density Matrix Formalism 6
F. Reduced Density Matrix and Partial Trace 6
G. Superposition vs. Statistical Mixture 7

V. Quantum Circuit Model 7
A. Motivation: Classical Circuits 7
B. Single-Qubit Gates 7
C. Two-Qubit Gates and Entanglement Generation 7
D. Universality of Quantum Gates 7
E. No-Cloning Theorem 8
F. Basic Quantum Protocols 8

VI. Reversible Computation and Complexity 8
A. Landauer's Principle 8
B. Toffoli Gate: Universal Reversible Classical Gate 8
C. Gate Decompositions and Computational Cost 9
D. Hadamard Test and Swap Test 9
E. Classical Complexity Classes 10
F. Quantum Complexity Classes 11

VII. Query Model of Computation and Algorithms 11
A. Query Model of Computation 11
B. The Phase Kick-back Trick 11
C. Quantum Query Algorithms 11

VIII. Quantum Fourier Transform and Phase Estimation 12
A. Quantum Fourier Transform (QFT) 12
B. Quantum Phase Estimation (QPE) 12

IX. Towards Quantum Machine Learning 13
A. Quantum linear systems: the HHL algorithm 13
B. Linear classifiers and support vector machines 14
C. Least-squares SVM as a linear system 15
D. Quantum least-squares SVM and classification 16
E. Quantum Advantage in Machine Learning 17
F. Dual SVM Formulation and Kernel Trick 17
G. Quantum Feature Maps and Quantum Kernels 18
H. Other landmark QML algorithms 19
I. Open Questions and Future Outlook 19

X. Variational Quantum Machine Learning 20
A. Variational Quantum Algorithms 20
1. Basic idea 20
2. Variational method in quantum mechanics 20

* jeongbin033@yonsei.ac.kr

B. Quantum Supervised Learning 21
1. Essence of QML: a linear model 21
C. Variational Quantum Classifier 21
1. Hyperplane picture 21
D. Quantum Convolutional Neural Networks 21
1. Parameterized quantum circuit examples 22
E. Universal Approximation Theorem 22
F. Analytical Gradients for Variational Quantum Circuits 22
1. Density-matrix form of the gradient 22
2. Schrödinger- and Heisenberg-picture interpretations 23
3. Parameter-shift rule 24
G. Quantum State Discrimination 24
1. Two-state discrimination problem 24
2. Success probability and trace distance 25
H. Variational Quantum Classifier from State Discrimination 25
I. Summary 26

XI. Quantum Approaches to Discrete Optimization 27
A. Discrete Optimization and QUBO 27
B. Famous Example: the Knapsack Problem 27
1. Handling constraints via the penalty method 27
C. From Optimization to Quantum Hamiltonians 28
D. Adiabatic Quantum Computing 28
1. The quantum adiabatic theorem 28
2. Adiabatic quantum optimization 29
E. QUBO Example: Max-Cut 29
F. Challenges of Adiabatic Quantum Computing 30
G. Quantum Approximate Optimization Algorithm (QAOA) 30
1. From adiabatic evolution to quantum circuits 30
2. Lie–Trotter and Suzuki–Trotter Product Formulas 30
3. QAOA circuit 32
4. Max-cut example with QAOA 32
5. Generalization: non-uniform schedules 32
H. Summary 32

XII. Conclusion 33

References 33

I. INTRODUCTION

A. Motivation: The Age of Quantum Computing

For over half a century, classical computing has followed Moore's Law: the number of transistors on a microprocessor doubles approximately every two years, delivering exponential growth in computational power at constant cost [1]. However, as transistor feature sizes approach atomic dimensions, quantum mechanical effects such as tunneling and thermal fluctuations undermine deterministic switching. This physical barrier signals the end of the classical scaling era and motivates the search for radically different computational paradigms.

Quantum computing exploits the principles of quantum mechanics—superposition, entanglement, and interference—to represent and process information in ways that classical computers cannot efficiently simulate. A classical register of n bits can store exactly one of 2^n binary strings at any given time. By contrast, an n-qubit quantum register can exist in a superposition of all 2^n computational basis states simultaneously (see Postulate I in Section III A for a formal definition). The amplitudes $\{\alpha_x\}$ encode correlations across an exponentially large state space, and unitary operations manipulate all 2^n amplitudes in parallel—a wave-like processing step. Measurement then collapses this superposition, yielding a bit-string x with probability $|\alpha_x|^2$—a particle-like readout step. This *analog-digital duality* underlies the power of quantum computing [2].

B. Classical vs. Quantum Information

Classical information theory studies the representation, compression, and transmission of information encoded in discrete, deterministic symbols [3]. A probabilistic extension replaces each bit state by a probability distribution $\mathbf{p} = (p_0, p_1)^\top$, $p_0 + p_1 = 1$, $p_i \geq 0$. Operations on probabilistic bits are described by stochastic matrices—matrices whose columns are probability vectors.

Quantum information theory is a mathematical generalization of classical probabilistic information. Specifically, whereas classical transitions are modeled by a *Markov chain* where a stochastic matrix M acts on a probability vector $\mathbf{x}$, quantum transitions are governed by *unitary evolution* where a unitary matrix U acts on a ket state $|\psi\rangle$. The key correspondences are summarized below:

	Classical (Markov)	Quantum (Unitary)
State	Prob. vector $\mathbf{x} \in \mathbb{R}_+^d$	Ket $\|\psi\rangle \in \mathbb{C}^d$
Normalization	L_1 norm ($\sum x_i = 1$)	L_2 norm ($\sum \|\alpha_i\|^2 = 1$)
Evolution	Stochastic M ($\sum_i M_{ij} = 1$)	Unitary U ($U^\dagger U = \mathbf{I}$)
Observation	Sampling from $\mathbf{x}$	Born-rule measurement

This structural analogy reveals that quantum mechanics is, in a precise sense, a *complex-amplitude* generalization of probability theory [4]. The crucial difference is that complex amplitudes can interfere constructively and destructively—classical probabilities cannot. Interference is the key resource exploited by quantum algorithms to suppress wrong answers and amplify correct ones [2].

C. Machine Learning as an Optimization Problem

Machine Learning (ML) aims to enable computers to learn underlying properties of data to perform tasks without explicit programming. Mathematically, ML represents an optimization problem where we search for a model $f(z, x)$ parameterized by z that minimizes an expected loss over the data distribution $\mathcal{D}$ [5]:

$$\min_{z\in\Omega} \mathbb{E}_{x\sim\mathcal{D}}\big[\ell\big(f(z,x),x\big)\big], \tag{1}$$

for some per-example loss ℓ (e.g., squared error or cross-entropy). Given a finite sample $\mathcal{S} = \{x_i\}_{i=1}^m$ drawn from $\mathcal{D}$, training minimizes the *empirical risk*

$$\min_{z\in\Omega} \frac{1}{m}\sum_{i=1}^{m} \ell\big(f(z,x_i),x_i\big). \tag{2}$$

Typical objective functions can be derived from the principle of *Maximum Likelihood Estimation* (MLE). For a dataset $Y =$

$\{y_1, \ldots, y_m\}$, the log-likelihood $L(\theta)$ is maximized:

$$\max_\theta L(\theta) = \sum_{i=1}^{m} \log(p(y_i|x_i, \theta)). \tag{3}$$

Under the assumption of i.i.d. Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ in observations $y_i = f(x_i, \theta) + \epsilon$, MLE is equivalent to minimizing the sum of squared errors (SSE/MSE):

$$L(\theta) \propto -\sum_{i=1}^{m} (y_i - f(x_i, \theta))^2 + \text{const.} \tag{4}$$

Similarly, for binary labels $y_i \in \{0, 1\}$, if the model predicts $p_i := p(y{=}1 \mid x_i, \theta)$ (e.g. $p_i = \sigma(f(x_i, \theta))$ with a sigmoid σ), then maximizing the Bernoulli log-likelihood is equivalent to minimizing the *binary cross-entropy*

$$L(\theta) = -\sum_{i=1}^{m} \big[y_i \log p_i + (1 - y_i) \log(1 - p_i)\big]. \tag{5}$$

D. Scope and Organization of This Report

This report follows STA4121 (through Week 11) at a level of mathematical detail intended to complement and, in key derivations, surpass standard textbook treatments. The organization is as follows. Section I introduces the motivation and the machine learning optimization framework. Section II reviews prerequisites including Dirac notation, computational complexity, and classical information theory. Section III develops the three postulates of quantum mechanics. Section IV treats entanglement and density matrices. Section V introduces the quantum circuit model and elementary protocols including Superdense Coding and Teleportation. Section VI discusses reversible computation, Landauer's principle, the Hadamard test, and a pure-state swap test (density-matrix swap test and Pauli kernel coordinates appear with quantum kernels in Sec. IX). Section VII presents the quantum query model and Grover search. Section VIII develops the quantum Fourier transform and phase estimation as subroutines underpinning exponential quantum speedups. Section IX connects these tools to machine learning: HHL, classical margin and least-squares SVMs, the quantum LS-SVM pipeline, criteria for quantum advantage, dual SVMs and the kernel trick, then quantum encodings and feature-map kernels, followed by a concise survey of other QML ideas. Section X develops the variational quantum machine learning framework: the variational quantum eigensolver (VQA), linearity of QML models in Pauli feature space, variational quantum classifiers with hyperplane decision boundaries, quantum convolutional neural networks (QCNN), the universal approximation theorem for quantum models, the parameter-shift rule for exact gradient computation, and fundamental limits from quantum state discrimination (Helstrom bound). Section XI turns to quantum approaches to combinatorial optimization: the QUBO framework and Ising Hamiltonian encoding, adiabatic quantum computing (AQC) with a self-contained derivation of the adiabatic condition $T \gg \hbar \max |\langle E_m|\dot{H}|E_0\rangle|/g_{\min}^2$ via the instantaneous eigenbasis and integration by parts, the max-cut problem as a canonical QUBO example, and the quantum approximate optimization algorithm (QAOA). QAOA is derived from adiabatic evolution via first-order Lie–Trotter and second-order symmetric Suzuki–Trotter product formulas, with detailed Taylor expansions proving $O(\Delta t^2)$ and $O(\Delta t^3)$ local truncation errors respectively and corresponding global error bounds via the telescoping identity. Section XII concludes.

II. MATHEMATICAL PRELIMINARIES

A. Dirac Notation and Hilbert Space

All quantum states live in a complex Hilbert space $\mathcal{H}$, a complete inner-product space over $\mathbb{C}$. Following Dirac's notation:

- A *ket* $|\psi\rangle \in \mathcal{H}$ is a column vector representing a quantum state.
- A *bra* $\langle\psi| = |\psi\rangle^\dagger$ is the conjugate-transpose row vector.
- The *inner product* $\langle\phi|\psi\rangle \in \mathbb{C}$ satisfies linearity in the second argument and $\langle\psi|\psi\rangle \geq 0$, with equality iff $|\psi\rangle = 0$.
- The *outer product* $|\psi\rangle\langle\phi|$ is a linear operator on $\mathcal{H}$.

For an n-qubit system, $\mathcal{H} = \mathbb{C}^{2^n}$ with the standard inner product $\langle\phi|\psi\rangle = \sum_x \phi_x^* \psi_x$. The *computational basis* $\{|x\rangle\}_{x\in\{0,1\}^n}$ is the set of standard basis vectors, which forms a complete orthonormal set satisfying $\langle x|y\rangle = \delta_{xy}$ and $\sum_x |x\rangle\langle x| = \mathbf{I}$.

B. Asymptotic Complexity Notation

In the analysis of both classical and quantum algorithms, it is essential to characterize how the computational cost (time or space) scales with the input size N as $N \to \infty$ [6].

1. **Big-O** (O): $f(N) = O(g(N))$ if there exist constants $c, N_0 > 0$ such that $f(N) \leq c \cdot g(N)$ for all $N \geq N_0$. This provides an *upper bound* on growth.
2. **Big-Omega** (Ω): $f(N) = \Omega(g(N))$ if $f(N) \geq c \cdot g(N)$ for all $N \geq N_0$. This provides a *lower bound*.
3. **Big-Theta** (Θ): $f(N) = \Theta(g(N))$ if $f(N) = O(g(N))$ and $f(N) = \Omega(g(N))$. This represents a *tight bound*.
4. **Little-o** (o) and **Little-omega** (ω): Used for strict upper and lower bounds, respectively.

An algorithm is generally considered *efficient* if its complexity is polynomial in N, i.e., $O(N^k)$ for some constant k. Key examples include:

- Inner product $\mathbf{x}^\top \mathbf{y}$ costs $O(N)$.
- Matrix-vector product $A\mathbf{x}$ costs $O(MN)$ for $A \in \mathbb{R}^{M\times N}$.
- Classical matrix inversion costs $O(N^3)$, while the HHL algorithm offers an exponential speedup to $O(\text{poly}\log N)$ for sparse, well-conditioned systems [7].

C. Classical Information: Probabilistic States and Stochastic Matrices

Let $\mathcal{S} = \{0, 1\}$ denote the classical state set of a bit X. A *probabilistic state* of X is a column vector $\mathbf{p} = (p_0, p_1)^\top \in \mathbb{R}^2$ where $p_a = \Pr(X = a)$, so $p_0, p_1 \geq 0$ and $p_0 + p_1 = 1$.

A *deterministic operation* mapping $a \mapsto f(a)$ is represented by the matrix M satisfying $M|a\rangle = |f(a)\rangle$ for all $a \in \mathcal{S}$. A *probabilistic operation* introducing randomness is represented by a *stochastic matrix*: a matrix M satisfying $M_{ij} \geq 0$ and $\sum_i M_{ij} = 1$ for all j. Stochastic matrices are exactly those that map probability vectors to probability vectors. Composition of operations:

$$\mathbf{p} \xrightarrow{M_1} M_1\mathbf{p} \xrightarrow{M_2} M_2M_1\mathbf{p}, \tag{6}$$

and the product M_2M_1 is again stochastic.

D. Optimization Preliminaries

Many machine learning algorithms, including Support Vector Machines (SVM), rely heavily on mathematical optimization to find the best model parameters. We briefly review the essential multivariable calculus and linear algebra concepts that underpin these optimization methods.

a. Gradients and Jacobians. For a scalar-valued function $f : \mathbb{R}^n \to \mathbb{R}$, the *gradient* $\nabla f(\mathbf{x})$ is the vector of its first-order partial derivatives:

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right)^\top. \tag{7}$$

For a vector-valued function $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^m$, the *Jacobian matrix* $J \in \mathbb{R}^{m\times n}$ generalizes the gradient, where $J_{ij} = \frac{\partial F_i}{\partial x_j}$.

b. Hessian Matrix and Taylor Expansion. The *Hessian matrix* H of a scalar function f contains its second-order partial derivatives, $H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$. By Clairaut's theorem, H is a symmetric matrix for continuously twice-differentiable functions. The Hessian allows us to approximate the function near a point $\mathbf{x}^*$ using the second-order Taylor expansion:

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^\top(\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^\top H(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + O(\|\mathbf{x} - \mathbf{x}^*\|^3). \tag{8}$$

c. Conditions for a Local Minimum. Using Eq. (8), we can rigorously derive the conditions for $\mathbf{x}^*$ to be a local minimum. First, the *first-order necessary condition* requires $\nabla f(\mathbf{x}^*) = 0$. If $\nabla f(\mathbf{x}^*) \neq 0$, one could choose a small displacement $\mathbf{x} - \mathbf{x}^* = -\epsilon\nabla f(\mathbf{x}^*)$ for some $\epsilon > 0$, yielding $f(\mathbf{x}) - f(\mathbf{x}^*) \approx -\epsilon\|\nabla f(\mathbf{x}^*)\|^2 < 0$, which contradicts the assumption that $\mathbf{x}^*$ is a local minimum. A point satisfying $\nabla f(\mathbf{x}^*) = 0$ is called a *stationary point*.

Second, the *second-order necessary condition* requires the Hessian $H(\mathbf{x}^*) \equiv \nabla^2 f(\mathbf{x}^*)$ to be *positive semi-definite* ($H \succeq 0$). At a stationary point, the first-order term vanishes, leaving $f(\mathbf{x}) - f(\mathbf{x}^*) \approx \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^\top H(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*)$. If $H(\mathbf{x}^*)$ had a negative eigenvalue, choosing $\mathbf{x} - \mathbf{x}^*$ along its corresponding eigenvector would make the quadratic form negative, again contradicting local minimality.

d. Spectral Decomposition and Sufficient Conditions. A symmetric matrix $A \in \mathbb{R}^{n\times n}$ admits a *spectral decomposition* $A = P\Lambda P^\top$, where P is an orthogonal matrix of eigenvectors and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ contains the real eigenvalues. A matrix is *positive definite* ($A \succ 0$) if all its eigenvalues are strictly positive ($\lambda_i > 0$).

We can now prove the *sufficient condition for a strict local minimum*: if $\nabla f(\mathbf{x}^*) = 0$ and $H(\mathbf{x}^*) \succ 0$, then $\mathbf{x}^*$ is a strict local minimum.

Proof. Let $\lambda_{\min} > 0$ be the smallest eigenvalue of $H(\mathbf{x}^*)$. For any displacement $\mathbf{d} = \mathbf{x} - \mathbf{x}^*$, the spectral decomposition guarantees that $\mathbf{d}^\top H(\mathbf{x}^*)\mathbf{d} \geq \lambda_{\min}\|\mathbf{d}\|^2$. From the Taylor expansion at the stationary point:

$$f(\mathbf{x}) - f(\mathbf{x}^*) \geq \frac{1}{2}\lambda_{\min}\|\mathbf{d}\|^2 + O(\|\mathbf{d}\|^3). \tag{9}$$

For sufficiently small $\|\mathbf{d}\| > 0$, the strictly positive quadratic term $\frac{1}{2}\lambda_{\min}\|\mathbf{d}\|^2$ strictly dominates the $O(\|\mathbf{d}\|^3)$ error term. Therefore, $f(\mathbf{x}) > f(\mathbf{x}^*)$ for all $\mathbf{x} \neq \mathbf{x}^*$ in a neighborhood of $\mathbf{x}^*$, proving it is a strict local minimum. ◊

e. Lagrangian and Duality Theory. Constrained optimization problems form the core of Support Vector Machines. Consider the general primal problem:

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad h_i(\mathbf{x}) = 0, \quad g_j(\mathbf{x}) \leq 0. \tag{10}$$

We define the *Lagrangian* by introducing multipliers $\lambda_i \in \mathbb{R}$ and $\mu_j \geq 0$:

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_i \lambda_i h_i(\mathbf{x}) + \sum_j \mu_j g_j(\mathbf{x}). \tag{11}$$

The *primal optimal value* p^* can be rewritten as $p^* = \min_{\mathbf{x}} \max_{\boldsymbol{\lambda}, \boldsymbol{\mu}\geq 0} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$. The *dual problem* reverses the order of optimization:

$$d^* = \max_{\boldsymbol{\lambda}, \boldsymbol{\mu}\geq 0} \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{12}$$

Theorem (Weak Duality). *For any optimization problem, $d^* \leq p^*$.*

Proof. By definition, for any chosen $\tilde{\boldsymbol{\lambda}}$ and $\tilde{\boldsymbol{\mu}} \geq 0$, the minimum over $\mathbf{x}$ provides a lower bound:

$$\min_{\mathbf{x}} L(\mathbf{x}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}) \leq L(\tilde{\mathbf{x}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}) \quad \text{for all } \tilde{\mathbf{x}}. \tag{13}$$

Taking the supremum over $\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}} \geq 0$ on the left side, and the infimum over $\tilde{\mathbf{x}}$ on the right side yields $d^* \leq p^*$. ◊

The difference $p^* - d^* \geq 0$ is the *duality gap*. When $p^* = d^*$, we say *strong duality* holds. Slater's condition guarantees strong duality for convex optimization problems with strictly feasible points, which is why SVMs can be reliably solved via their dual formulation.

f. Karush–Kuhn–Tucker (KKT) template. At a constrained local minimum, the KKT system combines stationarity of the Lagrangian with primal feasibility, dual feasibility for inequality multipliers ($\mu_j \geq 0$), and complementary slackness $\mu_j g_j(\mathbf{x}) = 0$. Soft-margin SVMs exhibit the slackness structure (Sec. IX B), whereas equality-constrained LS-SVM collapses to linear equalities only (Sec. IX C).

III. POSTULATES OF QUANTUM MECHANICS

A. Postulate I: State

Postulate I (State). *Any isolated physical system is completely described by a unit vector $|\psi\rangle$ in a complex Hilbert space $\mathcal{H}$, called the state space. For an n-qubit system $\mathcal{H} = \mathbb{C}^{2^n}$ and $\langle\psi|\psi\rangle = 1$.*

For a single qubit, $\mathcal{H} = \mathbb{C}^2$ and the most general state is

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \alpha,\beta \in \mathbb{C}, \quad |\alpha|^2 + |\beta|^2 = 1. \tag{14}$$

a. Bloch sphere parameterization. Writing $\alpha = e^{i\gamma}\cos(\theta/2)$ and $\beta = e^{i(\gamma+\phi)}\sin(\theta/2)$, the global phase $e^{i\gamma}$ is physically unobservable (it cancels in every Born-rule probability $p(m) = |\langle m|\,|\psi\rangle|^2$). Absorbing it, the canonical form is

$$|\psi\rangle = \cos\tfrac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\tfrac{\theta}{2}\,|1\rangle\,, \quad \theta \in [0,\pi],\ \phi \in [0,2\pi). \tag{15}$$

The pair (θ,ϕ) defines a point on the unit sphere—the *Bloch sphere*—giving a bijective correspondence between physically distinct pure single-qubit states and S^2. Antipodal points correspond to orthogonal states: the poles $|0\rangle$ and $|1\rangle$ are mutually orthogonal.

Sketch of proof. We verify Eq. (15) represents all and only unit vectors (up to global phase). Any $(\alpha,\beta) \in \mathbb{C}^2$ with $|\alpha|^2+|\beta|^2 = 1$ can be written as $\alpha = r_0 e^{i\phi_0}$, $\beta = r_1 e^{i\phi_1}$ with $r_0^2 + r_1^2 = 1$, $r_i \geq 0$. Setting $r_0 = \cos(\theta/2)$, $r_1 = \sin(\theta/2)$ (unique for $\theta \in [0,\pi]$), and factoring out $e^{i\phi_0}$ as the global phase gives $\phi = \phi_1 - \phi_0$. The parameterization is surjective onto the sphere and two-to-one only at the poles ($\theta = 0,\pi$), where ϕ is irrelevant—consistent with $|0\rangle, |1\rangle$ being pole states. ◊ ◊

B. Postulate II: Dynamics

Postulate II (Dynamics). *The time evolution of a closed quantum system is governed by the Schrödinger equation $i\hbar\,\partial_t\,|\psi(t)\rangle = H(t)\,|\psi(t)\rangle$, where $H(t)$ is the Hermitian Hamiltonian of the system. Equivalently, $|\psi(t)\rangle = U(t)\,|\psi(0)\rangle$ for some unitary $U(t)$.*

a. General solution via Dyson series. Formally integrating the Schrödinger equation gives

$$|\psi(t)\rangle = \mathcal{T}\exp\!\left(-\frac{i}{\hbar}\int_0^t H(t')\,dt'\right)|\psi(0)\rangle\,, \tag{16}$$

where $\mathcal{T}$ denotes time-ordering. For a *time-independent* Hamiltonian $H(t) = H$, the series resums exactly:

$$|\psi(t)\rangle = e^{-iHt/\hbar}\,|\psi(0)\rangle =: U(t)\,|\psi(0)\rangle\,. \tag{17}$$

Sketch of proof. Split $[0,t]$ into L equal intervals of width $\delta = t/L$. For small δ, $U(\delta) \approx I - iH\delta/\hbar$. The total evolution is $U(t) \approx (I - iH\delta/\hbar)^L \xrightarrow{L\to\infty} e^{-iHt/\hbar}$ by the definition of the matrix exponential. Hermiticity of H gives $U^\dagger = e^{iH^\dagger t/\hbar} = e^{iHt/\hbar}$, hence $U^\dagger U = e^{iHt/\hbar}e^{-iHt/\hbar} = I$, confirming unitarity. ◊ ◊

b. Spectral form and inner-product preservation. Since H is Hermitian, its spectral decomposition reads $H = \sum_k E_k\,|E_k\rangle\langle E_k|$, giving

$$U(t) = \sum_k e^{-iE_k t/\hbar}\,|E_k\rangle\langle E_k|\,. \tag{18}$$

Because $|e^{-iE_k t/\hbar}| = 1$, the evolution merely rotates phases; it preserves the norm and inner product: $\langle\psi(t)|\phi(t)\rangle = \langle\psi(0)|\,U^\dagger U\,|\phi(0)\rangle = \langle\psi(0)|\phi(0)\rangle$.

c. Circuit model. In quantum computing we set $\hbar = 1$. A quantum circuit decomposes $U = U_L\cdots U_2U_1$ where each U_k acts on one or two qubits. Because a product of unitaries is unitary ($(VU)^\dagger(VU) = U^\dagger V^\dagger VU = I$), every circuit realizes a valid unitary.

C. Postulate III: Measurement

Postulate III (Measurement). *Quantum measurements are described by a collection $\{M_m\}$ of measurement operators satisfying the completeness relation $\sum_m M_m^\dagger M_m = \mathbf{I}$. If the pre-measurement state is $|\psi\rangle$, the outcome m occurs with probability $p(m) = \langle\psi|\,M_m^\dagger M_m\,|\psi\rangle$, and the post-measurement state is $M_m\,|\psi\rangle\,/\sqrt{p(m)}$.*

a. Projective measurements. A *projective* (von Neumann) measurement is associated with a Hermitian observable $O = \sum_m \mu_m P_m$, where μ_m are real eigenvalues and $P_m = |\mu_m\rangle\langle\mu_m|$ are orthogonal projectors: $P_mP_{m'} = \delta_{mm'}P_m$, $\sum_m P_m = \mathbf{I}$. Setting $M_m = P_m$:

$$p(\mu_m) = \langle\psi|P_m|\psi\rangle\,, \qquad |\psi'\rangle = \frac{P_m\,|\psi\rangle}{\sqrt{p(\mu_m)}}. \tag{19}$$

b. Expectation value.

$$\langle O\rangle = \sum_m \mu_m p(\mu_m) = \langle\psi|\,O\,|\psi\rangle = \mathrm{Tr}(O\,|\psi\rangle\langle\psi|). \tag{20}$$

The second equality follows from $\sum_m \mu_m\langle\psi|P_m|\psi\rangle = \langle\psi|\,(\sum_m \mu_m P_m)\,|\psi\rangle$. The third uses the cyclic property of the trace.

c. Computational basis measurement. Setting $M_m = |m\rangle\langle m|$ for $m \in \{0,1\}^n$ and $|\psi\rangle = \sum_x \alpha_x\,|x\rangle$, outcome m occurs with probability $|\alpha_m|^2$, and the post-measurement state collapses to $|m\rangle$.

IV. MULTI-QUBIT SYSTEMS

A. Composite Systems and Tensor Products

Postulate IV (Composite Systems). *The state space of a composite physical system is the tensor product $\mathcal{H}_1\otimes\cdots\otimes\mathcal{H}_n$ of the component Hilbert spaces. If subsystem k is in state $|\psi_k\rangle$, the joint state is $|\psi_1\rangle\otimes\cdots\otimes|\psi_n\rangle$.*

For two qubits with $|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ and $|\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$, the joint state is

$$|\psi_1\rangle \otimes |\psi_2\rangle = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle, \tag{21}$$

which coincides with the Kronecker product of the column vectors. An n-qubit Hilbert space has dimension 2^n, growing exponentially—the root of both the power and the classical-simulation difficulty of quantum computation.

B. Quantum Entanglement

A bipartite state $|\Psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$ is *separable* if $|\Psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$; otherwise it is *entangled*. The four *Bell states*,

$$|\Phi^{\pm}\rangle = \tfrac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle), \tag{22}$$

$$|\Psi^{\pm}\rangle = \tfrac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle), \tag{23}$$

are maximally entangled two-qubit states and form an orthonormal basis for $\mathbb{C}^4$ (the Bell basis).

a. Schmidt decomposition and separability criterion. Any $|\Psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$ admits the *Schmidt decomposition*

$$|\Psi\rangle = \sum_k \sqrt{\lambda_k}\,|a_k\rangle|b_k\rangle, \quad \lambda_k \geq 0, \quad \sum_k \lambda_k = 1, \tag{24}$$

where $\{|a_k\rangle\}$ and $\{|b_k\rangle\}$ are orthonormal. The Schmidt rank (number of non-zero λ_k) characterizes entanglement: a state is separable iff its Schmidt rank is 1.

Sketch of proof. Represent $|\Psi\rangle$ as a matrix $M_{ij} = \alpha_{ij}$ via $|\Psi\rangle = \sum_{i,j}\alpha_{ij}|a_i\rangle|b_j\rangle$. Applying the singular value decomposition (SVD) $M = U\Sigma V^\dagger$ gives $|\Psi\rangle = \sum_k \sigma_k|\tilde{a}_k\rangle|\tilde{b}_k\rangle$ where $|\tilde{a}_k\rangle = \sum_i U_{ik}|a_i\rangle$ and $|\tilde{b}_k\rangle = \sum_j V^*_{jk}|b_j\rangle$ are new orthonormal sets. Setting $\sqrt{\lambda_k} = \sigma_k$ and normalizing completes the proof. ◊ ◊

C. Partial Measurement

Consider a two-qubit state $|\Psi\rangle = \sum_{i,j}\alpha_{ij}|ij\rangle$ with $\sum_{i,j}|\alpha_{ij}|^2 = 1$. Measuring only the first qubit with $M_0 = |0\rangle\langle 0| \otimes \mathbf{I}$ and $M_1 = |1\rangle\langle 1| \otimes \mathbf{I}$:

- Outcome 0 with probability $p(0) = |\alpha_{00}|^2 + |\alpha_{01}|^2$; post-state $|0\rangle \otimes \frac{\alpha_{00}|0\rangle + \alpha_{01}|1\rangle}{\sqrt{p(0)}}$.
- Outcome 1 with probability $p(1) = |\alpha_{10}|^2 + |\alpha_{11}|^2$; post-state $|1\rangle \otimes \frac{\alpha_{10}|0\rangle + \alpha_{11}|1\rangle}{\sqrt{p(1)}}$.

Notably, if $|\Psi\rangle = |\Phi^+\rangle$, measuring the first qubit and obtaining 0 instantly collapses the second qubit to $|0\rangle$, regardless of distance. This is *not* superluminal signaling: the outcome probabilities are always $1/2$.

D. Pauli Matrices

The Pauli matrices

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{25}$$

together with $I \equiv \sigma_0$ satisfy $P^2 = I$, $[X, Y] = 2iZ$ (and cyclic), and $\{P, Q\} = 2\delta_{PQ}I$. The set $\mathcal{P}_n = \{I, X, Y, Z\}^{\otimes n}$ (with phases $\{\pm 1, \pm i\}$) forms the n-qubit Pauli group and provides a basis for $2^n \times 2^n$ Hermitian matrices. This makes Pauli operators the natural language for quantum error correction and VQA cost functions.

E. Density Matrix Formalism

When the state of a system is only known probabilistically—system is in $|\psi_k\rangle$ with probability p_k—the *density operator*

$$\rho = \sum_k p_k |\psi_k\rangle\langle\psi_k| \tag{26}$$

provides a complete description. Key properties: (i) $\rho \geq 0$, (ii) $\mathrm{Tr}(\rho) = 1$, (iii) $\mathrm{Tr}(\rho^2) \leq 1$, with equality iff ρ is pure. Under unitary evolution, $\rho \to U\rho U^\dagger$. Under measurement $\{M_m\}$:

$$p(m) = \mathrm{Tr}(M_m^\dagger M_m \rho), \qquad \rho' = \frac{M_m \rho M_m^\dagger}{p(m)}. \tag{27}$$

The expectation value generalizes Eq. (20) to $\langle O\rangle = \mathrm{Tr}(O\rho)$.

a. Quantum channels (CPTP). Beyond unitary evolution $\rho \mapsto U\rho U^\dagger$, noisy or open-system dynamics is modeled by a *completely positive trace-preserving* map Φ. By Kraus' theorem [2], $\Phi(\rho) = \sum_k K_k \rho K_k^\dagger$ for operators $\{K_k\}$ with $\sum_k K_k^\dagger K_k = I$. Such maps preserve positivity and trace; QML slides encoding classical x in a state $\rho(x)$ implicitly allow ρ to arise from arbitrary valid quantum processing of some earlier pure state.

F. Reduced Density Matrix and Partial Trace

A central problem in quantum information is describing a subsystem A of a composite system AB when we lack access to system B. If the joint system is in state ρ_{AB}, the *reduced density matrix* ρ_A is defined via the *partial trace* over B:

$$\rho_A \equiv \mathrm{Tr}_B(\rho_{AB}) = \sum_j \langle j|_B \rho_{AB} |j\rangle_B, \tag{28}$$

where $\{|j\rangle_B\}$ is any orthonormal basis for $\mathcal{H}_B$.

a. Proof of Physical Consistency. The reduced density matrix ρ_A is the unique operator that correctly predicts the expectation values of all local observables O_A on system A.

Sketch of proof. Let $O = O_A \otimes I_B$ be an observable acting only on system A. Its expectation value in state ρ_{AB} is:

$$\begin{aligned}
\langle O_A \otimes I_B\rangle &= \mathrm{Tr}\big((O_A \otimes I_B)\rho_{AB}\big) \\
&= \sum_i \sum_j \langle i|_A \langle j|_B (O_A \otimes I_B)\rho_{AB}|i\rangle_A |j\rangle_B \\
&= \sum_i \langle i|_A O_A \underbrace{\left(\sum_j \langle j|_B \rho_{AB}|j\rangle_B\right)}_{\mathrm{Tr}_B(\rho_{AB})} |i\rangle_A \\
&= \sum_i \langle i|_A O_A \rho_A |i\rangle_A = \mathrm{Tr}(O_A\rho_A).
\end{aligned}$$

This identity proves that ρ_A contains all information necessary to describe local experiments on A [3]. ◊ ◊

b. Entanglement and Mixedness. Consider the maximally entangled Bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Its joint density matrix is $\rho_{AB} = |\Phi^+\rangle\langle\Phi^+|$. Computing the partial trace over B:

$$\begin{aligned}\rho_A &= \langle 0|_B \rho_{AB} |0\rangle_B + \langle 1|_B \rho_{AB} |1\rangle_B \\ &= \tfrac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) = \tfrac{1}{2}\mathbf{I}. \end{aligned} \tag{29}$$

While the joint state ρ_{AB} is pure ($\mathrm{Tr}(\rho_{AB}^2) = 1$), the reduced state ρ_A is *maximally mixed* ($\mathrm{Tr}(\rho_A^2) = 1/2$). This conversion of global entanglement into local classical uncertainty (entropy) is a hallmark of quantum correlations [2].

G. Superposition vs. Statistical Mixture

The density matrix formalism cleanly distinguishes a coherent quantum *superposition* from a classical *statistical mixture*. Consider a system that can be in eigenstates $|u_1\rangle$ or $|u_2\rangle$ of an observable with eigenvalues λ_1, λ_2.

a. Framework 1: Superposition State. A coherent superposition is written as $|\psi\rangle = c_1|u_1\rangle + c_2|u_2\rangle$, with $|c_1|^2 + |c_2|^2 = 1$. Its density matrix is:

$$\begin{aligned}\rho_{\text{pure}} &= |\psi\rangle\langle\psi| \\ &= |c_1|^2 |u_1\rangle\langle u_1| + |c_2|^2 |u_2\rangle\langle u_2| \\ &\quad + c_1 c_2^* |u_1\rangle\langle u_2| + c_1^* c_2 |u_2\rangle\langle u_1|. \end{aligned} \tag{30}$$

The off-diagonal terms represent quantum *interference*. If we measure an arbitrary observable with eigenstates $|\phi_m\rangle$, the probability $P(\phi_m) = \langle\phi_m|\rho_{\text{pure}}|\phi_m\rangle$ includes the cross-terms $2\mathrm{Re}[c_1 c_2^* \langle\phi_m|u_1\rangle\langle u_2|\phi_m\rangle]$, which can lead to constructive or destructive interference.

b. Framework 2: Statistical Mixture. If instead the system is in state $|u_1\rangle$ with probability $p_1 = |c_1|^2$ and $|u_2\rangle$ with probability $p_2 = |c_2|^2$ (e.g., due to classical ignorance), the mixed state density matrix is:

$$\rho_{\text{mixed}} = |c_1|^2 |u_1\rangle\langle u_1| + |c_2|^2 |u_2\rangle\langle u_2|. \tag{31}$$

Here, the off-diagonal interference terms are strictly zero. The measurement probability $P(\phi_m) = |c_1|^2 |\langle\phi_m|u_1\rangle|^2 + |c_2|^2 |\langle\phi_m|u_2\rangle|^2$ is simply a classical weighted sum of independent probabilities, fundamentally differentiating it from the coherent quantum behavior.

V. QUANTUM CIRCUIT MODEL

A. Motivation: Classical Circuits

Classical digital computation is described by Boolean circuits: directed acyclic graphs of *gates* with *wires* carrying bits. The set $\{\mathrm{NAND}, \mathrm{FANOUT}\}$ is universal. However, NAND is *irreversible* (many-to-one), which by Landauer's principle implies thermodynamic energy dissipation (Sec. VI A).

B. Single-Qubit Gates

Single-qubit gates are 2×2 unitary matrices. The most important examples:

$$H = \tfrac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \quad X = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \quad Z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}, \tag{32}$$

$$S = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}, \quad T = \begin{pmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{pmatrix}, \quad T^2 = S,\ S^2 = Z. \tag{33}$$

Rotation gates about Bloch-sphere axes:

$$R_{\hat{n}}(\theta) \coloneqq e^{-i\theta\hat{n}\cdot\vec{\sigma}/2} = \cos\tfrac{\theta}{2}\, I - i\sin\tfrac{\theta}{2}\,\hat{n}\cdot\vec{\sigma}, \tag{34}$$

where $\vec{\sigma} = (X, Y, Z)$. Geometrically, $R_{\hat{n}}(\theta)$ rotates the Bloch-sphere vector by angle θ about the $\hat{n}$-axis.

a. Euler decomposition. Every single-qubit unitary $U \in SU(2)$ can be written as

$$U = e^{i\alpha} R_z(\beta)\, R_y(\gamma)\, R_z(\delta) \tag{35}$$

for some real $\alpha, \beta, \gamma, \delta$.

Sketch of proof. Since U is 2×2 unitary and $\det U = 1$ (up to global phase), write $U = \left(\begin{smallmatrix} a & -b^* \\ b & a^* \end{smallmatrix}\right)$ with $|a|^2 + |b|^2 = 1$. Set $a = e^{i(\alpha-\beta/2-\delta/2)}\cos(\gamma/2)$ and $b = e^{i(\alpha+\beta/2-\delta/2)}\sin(\gamma/2)$. Matching this to $R_z(\beta)R_y(\gamma)R_z(\delta)$ computed explicitly via Eq. (34) confirms the decomposition for all $U \in SU(2)$. ◊ ◊

A hardware-convenient parameterization for superconducting qubits is

$$U(\theta, \phi, \lambda) = R_z(\phi)\, R_x\left(-\tfrac{\pi}{2}\right)\, R_z(\theta)\, R_x\left(\tfrac{\pi}{2}\right)\, R_z(\lambda). \tag{36}$$

C. Two-Qubit Gates and Entanglement Generation

Product operators $U_A \otimes U_B$ cannot generate entanglement. The canonical entangling gates are:

$$CX = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X, \tag{37}$$

$$CZ = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z. \tag{38}$$

CX flips the target conditioned on the control being $|1\rangle$. Applying CX to $(H|0\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$:

$$CX \cdot \tfrac{1}{\sqrt{2}}(|00\rangle + |10\rangle) = \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\Phi^+\rangle, \tag{39}$$

demonstrating Bell-state generation from a product state.

D. Universality of Quantum Gates

Theorem (Universality). *A gate set $\mathcal{G}$ is* universal for quantum computation *if any n-qubit unitary can be approximated to precision ε by a circuit of size $O\left(\mathrm{poly}\left(\log\frac{1}{\varepsilon}\right)\right)$ from $\mathcal{G}$. The set $\{H, T, CX\}$ is universal [2].*

Sketch of proof. The argument proceeds in two steps. (1) *Single-qubit universality*: H and T together generate a dense subgroup of $SU(2)$—this can be shown by computing HTH and checking that the resulting rotations are irrational multiples

of π, ensuring density. (2) *Multi-qubit reduction*: any n-qubit unitary can be decomposed into two-qubit unitaries via a sequence of controlled-U gates, each decomposable into CX and single-qubit gates (Gray-code argument). The Solovay-Kitaev theorem then guarantees that approximation within ε requires only $O(\mathrm{poly}\log(1/\varepsilon))$ gates. ◊ ◊

E. No-Cloning Theorem

Theorem (No-Cloning, Wootters & Zurek 1982). *There is no unitary operation U such that $U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$ for all $|\psi\rangle$ [8].*

Proof. Suppose such U exists and let $|\psi\rangle, |\phi\rangle$ be arbitrary. Then $U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$ and $U|\phi\rangle|0\rangle = |\phi\rangle|\phi\rangle$. Taking the inner product of both equations and using unitarity ($U^\dagger U = I$):

$$\langle\psi|\phi\rangle \underbrace{\langle 0|0\rangle}_{=1} = \langle\psi|\phi\rangle\langle\psi|\phi\rangle = \langle\psi|\phi\rangle^2 . \tag{40}$$

Hence $\langle\psi|\phi\rangle(\langle\psi|\phi\rangle - 1) = 0$, forcing $\langle\psi|\phi\rangle \in \{0, 1\}$. Any two states that U clones must be either identical or orthogonal, contradicting the assumption that U clones *all* states. ◊

a. Implication for QML. No-cloning forbids direct copying of unknown quantum data, restricting classical-style data augmentation. It simultaneously motivates quantum-native protocols (superdense coding, teleportation) that exploit entanglement instead of copying.

F. Basic Quantum Protocols

a. Superdense coding. By sharing the Bell pair $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ in advance, Alice can transmit *2 classical bits* to Bob by sending only a *single qubit* [9]. By applying a local unitary $X^a Z^b$ ($a, b \in \{0, 1\}$) to her qubit, Alice can transform the globally shared entangled state into any of the four orthogonal Bell states. Bob then performs a Bell-basis measurement to distinguish these states with 100% certainty.

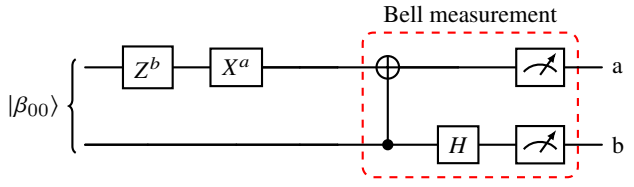


FIG. 1. Superdense coding circuit. Alice applies local operators Z^b and X^a to her half of a shared Bell pair (q_A). She then sends q_A to Bob, who performs a Bell-basis measurement (CNOT and H) to recover both bits a, b.

The protocol succeeds because the set $\{(E_i \otimes I)|\Phi^+\rangle\}$ forms an orthonormal basis for the four-dimensional Hilbert space $(\mathbb{C}^2)^{\otimes 2}$. Alice effectively picks which basis vector the system resides in, and Bob performs the change-of-basis back to the computational basis for sampling.

b. Quantum teleportation. Alice transmits an unknown qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ to Bob using only a shared Bell pair and two classical bits [10]. The circuit (Fig. 2) evolves as follows. The three-qubit initial state is

$$|\psi\rangle \otimes |\Phi^+\rangle = \tfrac{1}{\sqrt{2}}(\alpha|0\rangle + \beta|1\rangle)(|00\rangle + |11\rangle). \tag{41}$$

After Alice applies CX then H on her qubits, the state becomes

$$\tfrac{1}{2}\big[\,|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)\big]. \tag{42}$$

Alice measures qubits 1 and 2 and sends outcomes (a, b) classically. Bob applies $Z^a X^b$ to recover $\alpha|0\rangle + \beta|1\rangle = |\psi\rangle$ exactly.

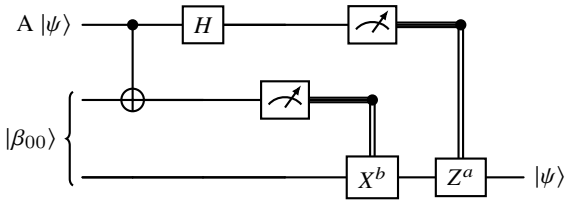


FIG. 2. Quantum teleportation circuit. Alice performs a Bell measurement on her state A and her half of a shared Bell pair A. The outcomes a and b are used to apply gates X^b and Z^a to Bob's qubit B, recovering $|\psi\rangle$.

Importantly, the "teleportation" is not instantaneous. Although the system collapses into one of the four states in Table II immediately upon Alice's measurement, Bob cannot know which gate to apply without the two classical bits. This preserves causality and satisfies the *no-communication theorem*: entanglement alone cannot transmit information faster than light.

VI. REVERSIBLE COMPUTATION AND COMPLEXITY

A. Landauer's Principle

Landauer's Principle (1961). *Erasing one bit of information irreversibly dissipates at least $k_B T \ln 2$ of energy as heat, where k_B is Boltzmann's constant and T is the temperature of the environment [11].*

Since NAND (and most classical gates) are many-to-one, they are thermodynamically irreversible and incur this minimum energy cost. In contrast, unitary quantum evolution is always reversible ($U^{-1} = U^\dagger$), satisfying Landauer's bound with equality in principle.

B. Toffoli Gate: Universal Reversible Classical Gate

The Toffoli (CCX) gate maps $(a, b, c) \mapsto (a, b, c \oplus ab)$. It is universal for reversible classical computation: both NAND and FANOUT can be simulated using Toffoli gates with ancilla bits. As a quantum gate on three qubits, it is a 8×8 permutation matrix.

TABLE I. Alice's encoding and resulting Bell states for Superdense Coding. The bit b (determined by the wire with the H gate) controls the Z gate.

Requested Bits (ba)	Alice's Gate Logic ($X^a Z^b$)	Resulting Bell State (on wire A)	Bob's Measurement Result (ba)
00	I	$\|\Phi^+\rangle = \frac{1}{\sqrt{2}}(\|00\rangle + \|11\rangle)$	00
01	X	$\|\Psi^+\rangle = \frac{1}{\sqrt{2}}(\|10\rangle + \|01\rangle)$	01
10	Z	$\|\Phi^-\rangle = \frac{1}{\sqrt{2}}(\|00\rangle - \|11\rangle)$	10
11	XZ	$\|\Psi^-\rangle = \frac{1}{\sqrt{2}}(\|10\rangle - \|01\rangle)$	11

TABLE II. Alice's measurement outcomes and Bob's corresponding recovery gates. The bit a (from the wire with H) controls the Z gate.

Outcome (ba)	Alice's Projector	Bob's Received State	Correction Logic ($Z^a X^b$)
00	$\|\Phi^+\rangle\langle\Phi^+\|$	$\alpha\|0\rangle + \beta\|1\rangle$	I
10	$\|\Psi^+\rangle\langle\Psi^+\|$	$\alpha\|1\rangle + \beta\|0\rangle$	X
01	$\|\Phi^-\rangle\langle\Phi^-\|$	$\alpha\|0\rangle - \beta\|1\rangle$	Z
11	$\|\Psi^-\rangle\langle\Psi^-\|$	$\alpha\|1\rangle - \beta\|0\rangle$	ZX

a. Quantum oracle for arbitrary functions. Any $f : \{0,1\}^n \to \{0,1\}^m$ can be embedded into the reversible unitary

$$U_f : |x\rangle\,|y\rangle \mapsto |x\rangle\,|y \oplus f(x)\rangle\,, \tag{43}$$

where $\oplus$ denotes bitwise XOR. Applying U_f to a uniform superposition:

$$U_f\left(\frac{1}{\sqrt{2^n}}\sum_x |x\rangle\right)|0\rangle = \frac{1}{\sqrt{2^n}}\sum_x |x\rangle\,|f(x)\rangle\,, \tag{44}$$

evaluating f on all 2^n inputs simultaneously—the key mechanism underlying quantum speedups via interference.

C. Gate Decompositions and Computational Cost

In practice, we do not have direct access to arbitrary multi-qubit gates. Complex gates must be decomposed into a set of "native" gates supported by the specific quantum hardware. Typically, this universal set consists of single-qubit rotations and a two-qubit entangling gate such as CNOT (CX) or CZ. The overall computational cost is heavily influenced by the number of required two-qubit gates.

a. SWAP Gate Decomposition. The SWAP gate, which exchanges the quantum states of two qubits, can be elegantly decomposed into three consecutive CNOT gates alternating their control and target qubits: This identity is easily verified by

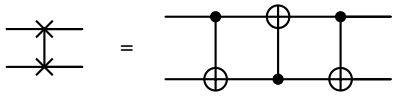


FIG. 3. Decomposition of the SWAP gate into three CNOT gates.

evaluating the action of the three CNOTs on the computational basis states $|00\rangle, |01\rangle, |10\rangle, |11\rangle$.

b. Toffoli (CCX) Gate Decomposition. The Toffoli gate is a three-qubit controlled-controlled-NOT gate. While universal for reversible classical computation, it is not native to most quantum hardware and requires decomposition. A standard decomposition into the Clifford+T gate set requires 6 CNOT gates and several single-qubit T, $T^\dagger$, and H gates: Because the Toffoli gate is foundational, this 6-CNOT cost serves as a critical benchmark for the overhead in fault-tolerant quantum algorithms.

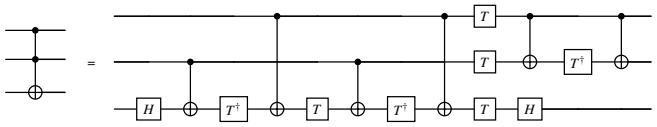


FIG. 4. Decomposition of the Toffoli (CCX) gate using 6 CNOT gates and single-qubit operations.

c. Fredkin (CSWAP) Gate Decomposition. The Fredkin gate is a controlled-SWAP gate. If the control qubit is $|1\rangle$, it swaps the two target qubits. It is heavily utilized in algorithms evaluating state overlaps, such as the swap test. Using the fact that a SWAP gate is three CNOTs, a CSWAP can be decomposed into two CNOTs and one Toffoli gate: Given that

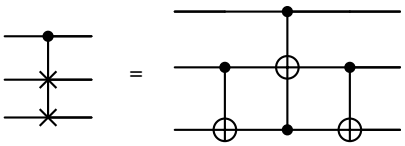


FIG. 5. Decomposition of the Fredkin (CSWAP) gate into two CNOT gates and one Toffoli gate.

a Toffoli gate costs 6 CNOTs, the CSWAP gate requires an overall cost of 8 CNOT gates.

D. Hadamard Test and Swap Test

a. Hadamard test. Given a unitary U and a state $|\psi\rangle$, the Hadamard test (Fig. 6) estimates $\langle\psi|U|\psi\rangle \in \mathbb{C}$. A full step-by-step derivation follows.

Complete derivation of Hadamard test. Step 1. Initialize ancilla and system: $|0\rangle\,|\psi\rangle$.

Step 2. Apply H to ancilla: $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,|\psi\rangle$.

Step 3. Apply controlled-U (if ancilla is $|1\rangle$, apply U to system):

$$\tfrac{1}{\sqrt{2}}\big(|0\rangle\,|\psi\rangle + |1\rangle\,U\,|\psi\rangle\big). \tag{45}$$

Step 4. Apply final H to ancilla:

$$\begin{aligned}&\tfrac{1}{2}\big[(|0\rangle + |1\rangle)\,|\psi\rangle + (|0\rangle - |1\rangle)U\,|\psi\rangle\big]\\&= \tfrac{1}{2}\,|0\rangle\,(|\psi\rangle + U\,|\psi\rangle) + \tfrac{1}{2}\,|1\rangle\,(|\psi\rangle - U\,|\psi\rangle).\end{aligned} \tag{46}$$

Step 5. Measure ancilla. The outcome probabilities are:

$$P(0) = \tfrac{1}{4}\|\,|\psi\rangle + U\,|\psi\rangle\,\|^2 = \tfrac{1}{4}(2 + 2\,\mathrm{Re}\,\langle\psi|U|\psi\rangle), \tag{47}$$

$$P(1) = \tfrac{1}{4}(2 - 2\,\mathrm{Re}\,\langle\psi|U|\psi\rangle). \tag{48}$$

Hence $P(0) - P(1) = \mathrm{Re}\,\langle\psi|U|\psi\rangle$.

Imaginary part. Insert an $S^\dagger$ phase gate before the final H; a parallel argument gives $P(0) - P(1) = \mathrm{Im}\,\langle\psi|U|\psi\rangle$. Together, both runs fully specify $\langle\psi|U|\psi\rangle \in \mathbb{C}$. ◊ ◊

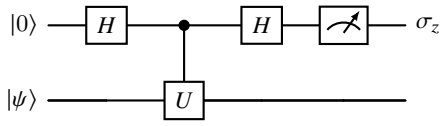


FIG. 6. Hadamard test circuit. The ancilla qubit $|0\rangle$ controls the unitary U on the system register $|\psi\rangle$. A final Hadamard on the ancilla followed by σ_z measurement yields $P(0) - P(1) = \mathrm{Re}\,\langle\psi|U|\psi\rangle$. Inserting $S^\dagger$ before the final H yields the imaginary part.

b. Swap test (pure states). The swap test compares two *pure* states $|\psi_1\rangle$ and $|\psi_2\rangle$ on separate registers using one ancilla. The circuit is shown in Fig. 7. Set $U = \mathrm{SWAP}$ and $|\psi\rangle = |\psi_1\rangle\,|\psi_2\rangle$ in the Hadamard test; then $P(0) - P(1) = \mathrm{Re}\,\langle\psi|U|\psi\rangle$ gives

$$\begin{aligned}P(0) - P(1) &= (\langle\psi_1| \otimes \langle\psi_2|)\,\mathrm{SWAP}\,(|\psi_1\rangle \otimes |\psi_2\rangle)\\&= (\langle\psi_1| \otimes \langle\psi_2|)(|\psi_2\rangle \otimes |\psi_1\rangle) = |\,\langle\psi_1|\psi_2\rangle\,|^2.\end{aligned} \tag{49}$$

Since $P(0) + P(1) = 1$,

$$P(0) = \tfrac{1}{2}\big(1 + |\,\langle\psi_1|\psi_2\rangle\,|^2\big), \qquad |\,\langle\psi_1|\psi_2\rangle\,|^2 = 2P(0) - 1. \tag{50}$$

The same circuit estimates $|\,\langle\psi_1|\psi_2\rangle\,|^2$ by repeated shots on the ancilla. A *density-matrix* formulation (mixed inputs ρ_1, ρ_2), explicit 8×8 unitaries, Hilbert–Schmidt kernels $K_{ij} = \mathrm{Tr}(\rho(x_i)\rho(x_j))$, and a Pauli-basis coordinate picture appear in Sec. IX G 0 g (quantum kernels).

c. Matrix representation (three-qubit unitary). Label wires top-to-bottom ancilla a, first data j, second data k, and use the computational ordering $|ajk\rangle \in \{|000\rangle, \ldots, |111\rangle\}$ (eight-dimensional). The single-qubit Hadamard is

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}. \tag{51}$$

Hadamard on the ancilla only is the Kronecker product

$$H_a \equiv H \otimes I_4 = \frac{1}{\sqrt{2}}\begin{pmatrix}I_4 & I_4\\ I_4 & -I_4\end{pmatrix} \in \mathbb{C}^{8\times 8}. \tag{52}$$

The two-qubit SWAP on the data wires, in the $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ basis (second qubit j, third k), is

$$S = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}, \qquad S\,|jk\rangle = |kj\rangle\,. \tag{53}$$

The Fredkin (controlled-SWAP) gate fixes the data when $a = 0$ and applies S when $a = 1$, hence in the $|ajk\rangle$ ordering it is the block matrix

$$U_{\mathrm{CSWAP}} = \begin{pmatrix}I_4 & 0\\ 0 & S\end{pmatrix}, \tag{54}$$

because the first four basis kets have $a = 0$ and the last four have $a = 1$. The full swap-test unitary (before measurement) is

$$\begin{aligned}U_{\mathrm{SW}} &= H_a\,U_{\mathrm{CSWAP}}\,H_a\\&= (H \otimes I_4)\,U_{\mathrm{CSWAP}}\,(H \otimes I_4).\end{aligned} \tag{55}$$

For $|\Psi_{\mathrm{in}}\rangle = |0\rangle\,|\psi_1\rangle\,|\psi_2\rangle$, expanding $U_{\mathrm{SW}}\,|\Psi_{\mathrm{in}}\rangle$ in the computational basis and summing Born weights on the ancilla branch $|0\rangle_a$ reproduces (50); equivalently, insert $\rho_i = |\psi_i\rangle\,\langle\psi_i|$ into (126).

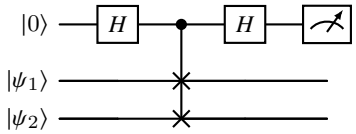


FIG. 7. Swap test for *pure* states: ancilla prepared in $|0\rangle$, two data registers in $|\psi_1\rangle$ and $|\psi_2\rangle$. Controlled-SWAP (Fredkin) sandwiched between Hadamards yields $P(0) = \frac{1}{2}(1 + |\,\langle\psi_1|\psi_2\rangle\,|^2)$ (50). Matrix form (54)–(55).

E. Classical Complexity Classes

To understand the potential advantages of quantum computing, it is essential to review the foundations of classical complexity theory. Complexity theory categorizes problems based on the resources (time or space) required to solve them.

a. Decision Problems. A decision problem asks a yes-or-no question. Many complex optimization problems (e.g., "minimize $f(x)$") can be recast as decision problems (e.g., "is there an x such that $f(x) \leq L$?").

b. The Classes P and NP.

- **P** (Polynomial time): The class of decision problems that can be solved efficiently by a deterministic classical computer in polynomial time $O(n^k)$, where n is the input size. Examples include sorting or finding a substring.

- **NP** (Nondeterministic Polynomial time): The class of decision problems for which a "yes" instance has a proof (or witness) that can be verified efficiently in polynomial time. For example, finding a subset of numbers that sums to zero is difficult, but verifying a proposed subset takes only linear time.

While clearly $\mathrm{P} \subseteq \mathrm{NP}$, determining whether $\mathrm{P} = \mathrm{NP}$ is widely considered the most important open question in theoretical computer science.

c. NP-Hard and NP-Complete. A problem is **NP-hard** if every problem in NP can be reduced to it in polynomial time. It is **NP-complete** if it is both NP-hard and in NP. Classic NP-complete problems include Boolean satisfiability (SAT) and graph coloring.

F. Quantum Complexity Classes

- **BQP** (Bounded-error Quantum Polynomial time): decision problems solvable by a polynomial-time quantum circuit with error $\leq 1/3$. $\mathrm{BPP} \subseteq \mathrm{BQP} \subseteq \mathrm{PSPACE}$.
- **QMA** (Quantum Merlin-Arthur): the quantum analogue of NP; YES instances have a polynomial-size quantum witness verifiable by a BQP verifier.

Whether $\mathrm{BQP} \not\subseteq \mathrm{NP}$ or $\mathrm{BQP} \neq \mathrm{BPP}$ remains open [6]. The believed picture is $\mathrm{BPP} \subsetneq \mathrm{BQP}$, with factoring as a candidate separation (Shor's algorithm).

VII. QUERY MODEL OF COMPUTATION AND ALGORITHMS

A. Query Model of Computation

One of the potential advantages of quantum computers is to provide faster solutions to computational problems. To rigorously analyze computational complexity and demonstrate quantum advantages, we introduce the *query model of computation*. In this model, the input is provided via an oracle or black box that evaluates a Boolean function $f : \{0,1\}^n \to \{0,1\}$. The complexity of an algorithm is measured by the number of queries made to the oracle.

Classically, the oracle is evaluated as $x \mapsto f(x)$. In the quantum query model, the oracle is represented as a unitary operator U_f that acts on a superposition of states:

$$U_f \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle |y\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle |y \oplus f(x)\rangle . \quad (56)$$

This allows the quantum computer to query all possible inputs simultaneously, leveraging quantum parallelism.

B. The Phase Kick-back Trick

A recurring and foundational principle in many quantum algorithms (such as Deutsch-Jozsa, Simon's, Grover's, and Phase Estimation) is the *phase kick-back* trick. In classical reversible computation, an oracle U_f evaluates a Boolean function $f(x)$ by flipping a target bit y based on the result: $U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$.

However, in quantum computation, if we prepare the target register y in the specific superposition state $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, applying the oracle yields a remarkable effect. Consider the action of U_f on a single basis state $|x\rangle$ and the target $|-\rangle$:

$$\begin{aligned} U_f |x\rangle |-\rangle &= U_f \left(|x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \\ &= \frac{1}{\sqrt{2}} \left(U_f |x\rangle |0\rangle - U_f |x\rangle |1\rangle \right) \\ &= \frac{1}{\sqrt{2}} \left(|x\rangle |f(x)\rangle - |x\rangle |1 \oplus f(x)\rangle \right) . \end{aligned} \quad (57)$$

If $f(x) = 0$, the state becomes $|x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |x\rangle |-\rangle$. If $f(x) = 1$, the state becomes $|x\rangle \frac{|1\rangle - |0\rangle}{\sqrt{2}} = -|x\rangle |-\rangle$. We can succinctly write this as:

$$U_f |x\rangle |-\rangle = (-1)^{f(x)} |x\rangle |-\rangle . \quad (58)$$

Notice that the target state $|-\rangle$ remains entirely unchanged. Instead, the function evaluation $f(x)$ is "kicked back" as a global phase factor $(-1)^{f(x)}$ onto the control register $|x\rangle$. When $|x\rangle$ is in a superposition, these relative phases create destructive and constructive interference patterns, which quantum algorithms exploit to efficiently extract global properties of f.

More generally, if a unitary operator U has an eigenvector $|\psi\rangle$ with eigenvalue $e^{i\theta}$, a controlled-U gate targeting $|\psi\rangle$ will kick back the phase $e^{i\theta}$ to the control qubit. This generalized phase kick-back is the core mechanism behind the Quantum Phase Estimation algorithm.

C. Quantum Query Algorithms

Several foundational algorithms demonstrate the power of the quantum query model, exhibiting separations in complexity between classical and quantum computation.

a. Deutsch-Jozsa Algorithm. The Deutsch-Jozsa algorithm solves a specialized problem: given a boolean function $f : \{0,1\}^n \to \{0,1\}$ that is promised to be either constant (same output for all inputs) or balanced (outputs 0 for exactly half of the inputs and 1 for the other half), determine which it is. Classically, this requires $O(2^{n-1} + 1)$ queries in the worst case. Quantumly, it requires only a single query. The quantum circuit is as follows:

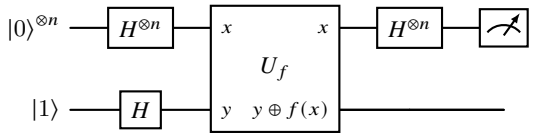


FIG. 8. Quantum circuit for the Deutsch-Jozsa and Bernstein-Vazirani algorithms.

The algorithm proceeds by initializing the system to $|0\rangle^{\otimes n} |1\rangle$ and applying Hadamard gates to all qubits, preparing the state:

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes |-\rangle . \quad (59)$$

Querying the oracle U_f applies a phase kick-back effect, encoding $f(x)$ into the amplitude:

$$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle \otimes |-\rangle . \quad (60)$$

Applying the final n-qubit Hadamard gate $H^{\otimes n}$ transforms the state using the identity $H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_z (-1)^{x\cdot z} |z\rangle$:

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} \sum_{z\in\{0,1\}^n} (-1)^{x\cdot z + f(x)} |z\rangle \otimes |-\rangle . \quad (61)$$

If f is constant, the amplitude of $|z = 0\rangle$ is ± 1, so measuring the first register yields 0^n with probability 1. If f is balanced, the amplitude of $|z = 0\rangle$ evaluates to $\sum_x (-1)^{f(x)}/2^n = 0$. Hence, a single measurement perfectly distinguishes the two cases.

b. Bernstein-Vazirani Algorithm. This algorithm finds a hidden n-bit string a for a linear function $f(x) = a \cdot x$. Classically, identifying a requires $O(n)$ queries by probing each

basis vector. Quantumly, the identical circuit (Fig. 8) finds a in 1 query. Substituting $f(x) = a \cdot x$ into $|\psi_3\rangle$, we get:

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} \sum_{z\in\{0,1\}^n} (-1)^{(a+z)\cdot x} |z\rangle \otimes |-\rangle . \quad (62)$$

The amplitude is non-zero if and only if $z = a$, yielding the state $|a\rangle |-\rangle$. Thus, a single measurement reveals the hidden string a exactly.

c. Simon's Algorithm. Simon's algorithm solves a problem with an exponential separation between quantum and randomized classical algorithms. We are given $f : \{0,1\}^n \to \{0,1\}^m$, promised that there is a hidden string s such that $f(x) = f(y)$ if and only if $x = y \oplus s$ or $x = y$. Classically, finding s requires $\Omega(2^{n/2})$ queries. The quantum algorithm finds s in $O(n)$ queries.

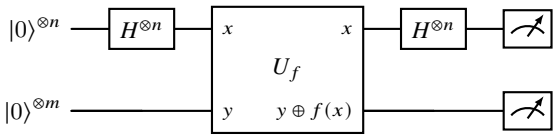


FIG. 9. Quantum circuit for Simon's algorithm.

The state before the oracle is $|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle |0\rangle^{\otimes m}$. The oracle yields $|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle |f(x)\rangle$. Measuring the second register gives some $f(x)$ and collapses the first register into an equal superposition of the two pre-images:

$$|\psi_2'\rangle = \frac{1}{\sqrt{2}}(|x\rangle + |x \oplus s\rangle) \otimes |f(x)\rangle . \quad (63)$$

Applying the final $H^{\otimes n}$ to the first register yields:

$$\begin{aligned} |\psi_3\rangle &= \frac{1}{\sqrt{2^{n+1}}} \sum_{z\in\{0,1\}^n} \left[(-1)^{x\cdot z} + (-1)^{(x\oplus s)\cdot z}\right] |z\rangle \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{z\in\{0,1\}^n} (-1)^{x\cdot z} \left[1 + (-1)^{s\cdot z}\right] |z\rangle . \end{aligned} \quad (64)$$

The amplitude is non-zero only when $s \cdot z = 0 \pmod 2$. By repeating the algorithm $O(n)$ times, we obtain a system of linear equations $z_i \cdot s = 0$, which can be solved efficiently on a classical computer to find s.

VIII. QUANTUM FOURIER TRANSFORM AND PHASE ESTIMATION

A. Quantum Fourier Transform (QFT)

The Fourier Transform is a fundamental mathematical tool for transforming signals between domains. The Discrete Fourier Transform (DFT) maps a vector $(x_0, \ldots, x_{N-1})$ to $(y_0, \ldots, y_{N-1})$ where $y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \exp(2\pi i jk/N)$. This requires $O(N \log N)$ operations classically using the Fast Fourier Transform (FFT).

The *Quantum Fourier Transform* (QFT) is the quantum analogue, performing the transformation on the amplitudes of a quantum state. For an n-qubit system with $N = 2^n$ basis states, the QFT is defined as:

$$\mathrm{QFT}\,|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i jk/N} |k\rangle . \quad (65)$$

Since it is unitary, the inverse QFT ($\mathrm{QFT}^\dagger$) is straightforwardly defined by taking the complex conjugate of the phase.

a. Product State Representation. To build the quantum circuit, it is highly instructive to rewrite the output state in a factored tensor product form. Let the integer j be represented in binary as $j = j_1 j_2 \ldots j_n$, and define the binary fraction $0.j_l \ldots j_n = \sum_{m=1}^{n-l+1} j_{l+m-1} 2^{-m}$. The QFT transformation can be elegantly factored as:

$$\begin{aligned} \mathrm{QFT}\,|j_1 j_2 \ldots j_n\rangle = \frac{1}{\sqrt{2^n}} &\left(|0\rangle + e^{2\pi i 0.j_n} |1\rangle\right) \\ &\otimes \left(|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle\right) \\ &\otimes \cdots \otimes \left(|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n} |1\rangle\right) . \end{aligned} \quad (66)$$

b. Quantum Circuit for QFT. This product form implies that the QFT can be implemented using single-qubit Hadamard gates (H) and two-qubit controlled-phase gates (R_k), where $R_k = \mathrm{diag}(1, e^{2\pi i/2^k})$. The Hadamard gate creates the equal superposition and the leading phase bit, while the controlled-R_k gates incrementally add the fractional phase contributions from the other qubits. A standard circuit for a 3-qubit QFT is shown below. Note that the output qubits are in reverse order (k_3, k_2, k_1), requiring a final set of SWAP gates if the original ordering is strictly needed.

The QFT requires $O(n^2) = O(\log^2 N)$ gates, offering an exponential speedup over the classical FFT's $O(N \log N)$ operations.

B. Quantum Phase Estimation (QPE)

Quantum Phase Estimation is one of the most important subroutines in quantum computing, forming the basis for Shor's algorithm (integer factoring) and the HHL algorithm (solving linear systems). Given a unitary operator U and its eigenvector $|\psi\rangle$ with an unknown eigenvalue $e^{2\pi i\theta}$, the goal of QPE is to estimate the phase $\theta \in [0, 1)$.

a. Algorithm Steps. The QPE algorithm uses two registers. The first (estimation) register contains t qubits initialized to $|0\rangle^{\otimes t}$, where t determines the precision of the phase estimate. The second (target) register is initialized to the eigenvector $|\psi\rangle$. The algorithm proceeds in three main steps:

1. **Superposition**: Apply Hadamard gates to all t qubits in the first register to create an equal superposition of all computational basis states.
2. **Controlled Unitaries**: Apply a sequence of controlled-U^{2^k} operations, controlled by the k-th qubit of the first register and targeting the second register. Because $U|\psi\rangle = e^{2\pi i\theta} |\psi\rangle$, applying U^{2^k} kicks back a phase of $e^{2\pi i\theta 2^k}$ to the control qubit. After all controlled unitaries, the state of the first register is:
$$\frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e^{2\pi i\theta k} |k\rangle . \quad (67)$$
3. **Inverse QFT**: The state of the first register is precisely the Quantum Fourier Transform of the state $|2^t\theta\rangle$. By applying the Inverse QFT ($\mathrm{QFT}^\dagger$) to the first register, the state transforms into the basis state $|2^t\theta\rangle$. Measuring the first register yields a binary string representing the best t-bit approximation of the phase θ.

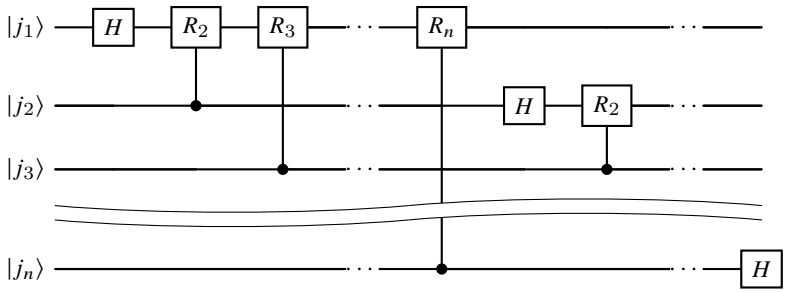


FIG. 10. Quantum circuit for an n-qubit Quantum Fourier Transform. SWAP gates at the end are omitted for brevity.

b. Quantum Circuit for QPE. The schematic circuit for Quantum Phase Estimation is illustrated below:

If the true phase θ cannot be exactly represented with t bits, the measured result will be the closest t-bit string with high probability. The required number of qubits t is chosen based on the desired accuracy and the allowed failure probability.

IX. TOWARDS QUANTUM MACHINE LEARNING

A. Quantum linear systems: the HHL algorithm

Many QML constructions reduce training or inference to solving a linear system $A\mathbf{x} = \mathbf{b}$. The quantum algorithm of Harrow, Hassidim, and Lloyd (HHL) [7] targets this problem when $A \in \mathbb{C}^{N\times N}$ is Hermitian and *s-sparse* (at most s nonzero entries per row). The idealized output is a normalized quantum state

$$|x\rangle \;\propto\; A^{-1}\,|b\rangle\,, \tag{68}$$

where $|b\rangle$ encodes $\mathbf{b}$ in an amplitude vector of dimension $N = 2^n$. Typical classical solvers for sparse, well-conditioned systems (such as conjugate gradient) cost

$$T_{\mathrm{cl}} \;=\; \tilde{O}\big(Ns\,\kappa\log(1/\varepsilon)\big) \tag{69}$$

up to logarithmic factors, where $\kappa = \lambda_{\max}/\lambda_{\min}$ is the spectral condition number and ε the accuracy. Under an ideal matrix-entry query model and QRAM-like state preparation, HHL achieves

$$T_{\mathrm{q}} \;=\; \mathrm{poly}\big(\log N,\, s,\, \kappa,\, 1/\varepsilon\big), \tag{70}$$

i.e., polynomial time in $\log N$ rather than in the dimension N [5, 7].

a. Spectral reduction. For Hermitian invertible A,

$$A = \sum_{j=0}^{N-1} \lambda_j\,|u_j\rangle\,\langle u_j|\,, \tag{71}$$

$$A^{-1} = \sum_{j=0}^{N-1} \lambda_j^{-1}\,|u_j\rangle\,\langle u_j|\,. \tag{72}$$

Writing $|b\rangle = \sum_j \beta_j\,|u_j\rangle$,

$$A^{-1}\,|b\rangle = \sum_{j=0}^{N-1} \frac{\beta_j}{\lambda_j}\,|u_j\rangle\,. \tag{73}$$

HHL prepares (73) coherently. Quantum phase estimation (Sec. VIII B) with $U = e^{iAt}$ entangles each $|u_j\rangle$ with a binary encoding of λ_j. Let $C > 0$ satisfy $|C/\lambda_j| \le 1$ on the relevant eigenvalue support. A single-qubit rotation on an *ancilla*, controlled on that encoding, can load amplitudes proportional to C/λ_j. After inverting the QPE unitary, unmeasured data registers hold a coherent superposition whose renormalization is the target (68).

b. Non-Hermitian systems. If A is square but not Hermitian, embed into the Hermitian dilation

$$\tilde{A} = \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}. \tag{74}$$

Then, with block vectors $\mathbf{0}, \mathbf{x}, \mathbf{b} \in \mathbb{C}^N$,

$$\tilde{A}\begin{pmatrix}\mathbf{0}\\ \mathbf{x}\end{pmatrix} = \begin{pmatrix} A\mathbf{x}\\ \mathbf{0}\end{pmatrix} = \begin{pmatrix}\mathbf{b}\\ \mathbf{0}\end{pmatrix} \quad\Longleftrightarrow\quad A\mathbf{x} = \mathbf{b}. \tag{75}$$

c. Circuit outline (conceptual). Figure 12 summarizes the core flow with wires q_1 (data), q_2 (clock), q_3 (ancilla), matching the lecture slides:

1. Prepare the amplitude-encoded state $|b\rangle = \sum_j \beta_j\,|u_j\rangle$.
2. Apply QPE with $U = e^{iAt}$ so that
$$|b\rangle \;\mapsto\; \sum_j \beta_j\,|u_j\rangle\,|\tilde{\lambda}_j\rangle\,, \tag{76}$$
where $|\tilde{\lambda}_j\rangle$ is a t-bit approximation of λ_j.
3. **Conditioned rotation on q_3.** Work in the computational basis of the ancilla and adopt the usual y-rotation
$$R_y(\theta) \;=\; e^{-i\theta Y/2} \;=\; \begin{pmatrix}\cos(\theta/2) & -\sin(\theta/2)\\ \sin(\theta/2) & \cos(\theta/2)\end{pmatrix}, \tag{77}$$
where the Pauli Y matrix is
$$Y = \begin{pmatrix} 0 & -i\\ i & 0\end{pmatrix}. \tag{78}$$
Hence $R_y(\theta)\,|0\rangle = \cos(\theta/2)\,|0\rangle + \sin(\theta/2)\,|1\rangle$. Fix a scale $C > 0$ such that $|C/\lambda_j| \le 1$ on every eigenvalue that receives amplitude from $|b\rangle$ (for signed λ_j, one first infers $|\lambda_j|$ coherently and absorbs any phase into an additional z-type rotation—the idealized positive case suffices for the amplitude logic). Define
$$\theta_j \;:=\; 2\arcsin\!\Big(\frac{C}{\lambda_j}\Big), \tag{79}$$
so that the *uncontrolled* rotation would map
$$R_y(\theta_j)\,|0\rangle = \sqrt{1-\frac{C^2}{\lambda_j^2}}\;|0\rangle + \frac{C}{\lambda_j}\;|1\rangle\,. \tag{80}$$

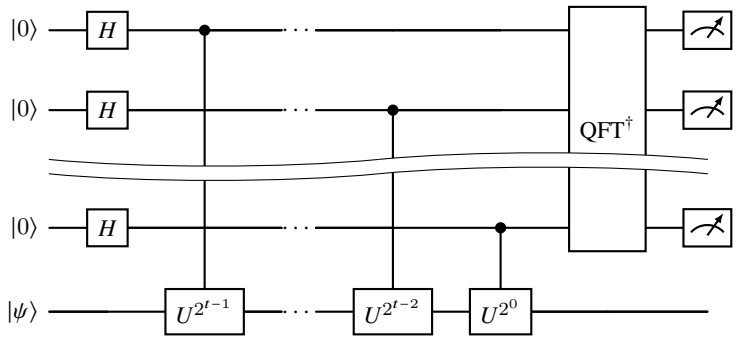


FIG. 11. Quantum circuit for the Quantum Phase Estimation algorithm using an arbitrary t-qubit precision register.

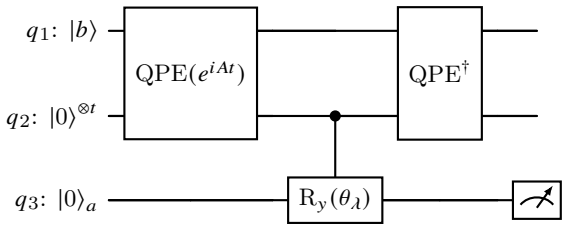


FIG. 12. Schematic HHL subroutine in the same qubit stacking as the course slides (top: amplitude loading of $|b\rangle$ on q_1, middle: t-qubit phase register q_2, bottom: workspace ancilla q_3 for eigenvalue inversion). The $\mathrm{QPE}(e^{iAt})$ block entangles q_1 (eigenstate register) with the encoded eigenvalues on q_2 (cf. Fig. 11; bit order within q_2 follows that figure's QFT convention). The gate $\mathrm{R}_y(\theta_\lambda)$ on q_3 is controlled by the phase register q_2 (filled control dot). After $\mathrm{QPE}^\dagger$ clears q_2, post-selecting q_3 on $|1\rangle$ yields $q_1 \propto A^{-1}|b\rangle$ [7].

The controlled gate applies $R_y(\theta_j)$ (or the QPE-limited analogue with $\tilde{\lambda}_j$) on q_3 whenever q_2 encodes branch j.

4. **Inverse QPE.** Applying $\mathrm{QPE}^\dagger$ disentangles the clock, leaving

$$|\Psi_{\mathrm{pre}}\rangle = \sum_{j=0}^{N-1} \beta_j |u_j\rangle_{q_1} \otimes |0\rangle_{q_2}^{\otimes t} \otimes \left(\sqrt{1-\frac{C^2}{\lambda_j^2}}\,|0\rangle_{q_3} + \frac{C}{\lambda_j}|1\rangle_{q_3}\right), \tag{81}$$

(Again, replace $\lambda_j \to \tilde{\lambda}_j$ to track finite-precision QPE.)

5. **Post-select $|1\rangle$ on q_3.** Measuring q_3 in the computational basis gives outcome probabilities

$$\Pr(\text{outcome } 1) = \sum_{j=0}^{N-1} |\beta_j|^2 \frac{C^2}{\lambda_j^2} =: p_{\mathrm{succ}}, \tag{82}$$

$$\Pr(\text{outcome } 0) = \sum_{j=0}^{N-1} |\beta_j|^2 \left(1-\frac{C^2}{\lambda_j^2}\right). \tag{83}$$

The joint Born weight on "1" for branch j is the product of $|\beta_j|^2$ from q_1 and $\sin^2(\theta_j/2) = C^2/\lambda_j^2$ from (80). Conditioning on outcome 1 collapses q_1 to

$$|x_{\mathrm{post}}\rangle = \frac{1}{\sqrt{p_{\mathrm{succ}}}} \sum_{j=0}^{N-1} \beta_j \frac{C}{\lambda_j} |u_j\rangle = \frac{C}{\sqrt{p_{\mathrm{succ}}}} A^{-1}|b\rangle, \tag{84}$$

which is exactly the target direction (73), up to the known global factor $C/\sqrt{p_{\mathrm{succ}}}$. Large κ makes some $|\lambda_j|^{-2}$ terms dominate (82), so naive post-selection can cost $\Omega(\kappa^{-2})$ attempts; amplitude amplification reduces the query overhead to $\tilde{O}(\kappa)$ in the analysis of Ref. [7].

The procedure assumes efficient oracles for preparing $|b\rangle$ and simulating e^{-iAt} (Hamiltonian simulation for sparse Hermitian A). The speedup is stated relative to this *quantum input model*; quantitative comparisons to classical numerical linear algebra require care [5].

B. Linear classifiers and support vector machines

Binary classifiers often seek a hyperplane in feature space $\mathbb{R}^d$ that splits two classes. Any affine hyperplane can be written as $H = \{\mathbf{x} : \langle \mathbf{w}, \mathbf{x}\rangle + b = 0\}$ with normal $\mathbf{w} \in \mathbb{R}^d$ and offset $b \in \mathbb{R}$. The *linear score* or *logit*

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x}\rangle + b \tag{85}$$

is positive on one side of H (predict class +1) and negative on the other (predict −1). For labeled training pairs $\{(\mathbf{x}_k, y_k)\}$ with $y_k \in \{+1, -1\}$, correct orientation means $y_k f(\mathbf{x}_k) > 0$ on every example used for training.

a. Geometric margin. The Euclidean distance from a point $\mathbf{x}$ to H is $|f(\mathbf{x})|/\|\mathbf{w}\|$. The quantity $\gamma_k := y_k f(\mathbf{x}_k)$ is the *functional margin* (up to sign it is the "score confidence"). Dividing by $\|\mathbf{w}\|$ gives the *geometric margin* $\gamma_k/\|\mathbf{w}\|$—the signed distance to the decision boundary in direction $\mathbf{w}$. Rescaling $(\mathbf{w}, b) \mapsto (c\mathbf{w}, cb)$ leaves the hyperplane fixed but changes $\|\mathbf{w}\|$; SVM fixes the scale by imposing $y_k f(\mathbf{x}_k) \geq 1$ on support vectors so that the slab between the two hyperplanes $f = \pm 1$ has half-width $1/\|\mathbf{w}\|$.

b. Maximum-margin (hard) SVM. For linearly separable data, the support vector machine (SVM) finds the unique hyperplane that maximizes the smallest geometric margin—equivalently, it shrinks $\|\mathbf{w}\|$ while keeping all training points

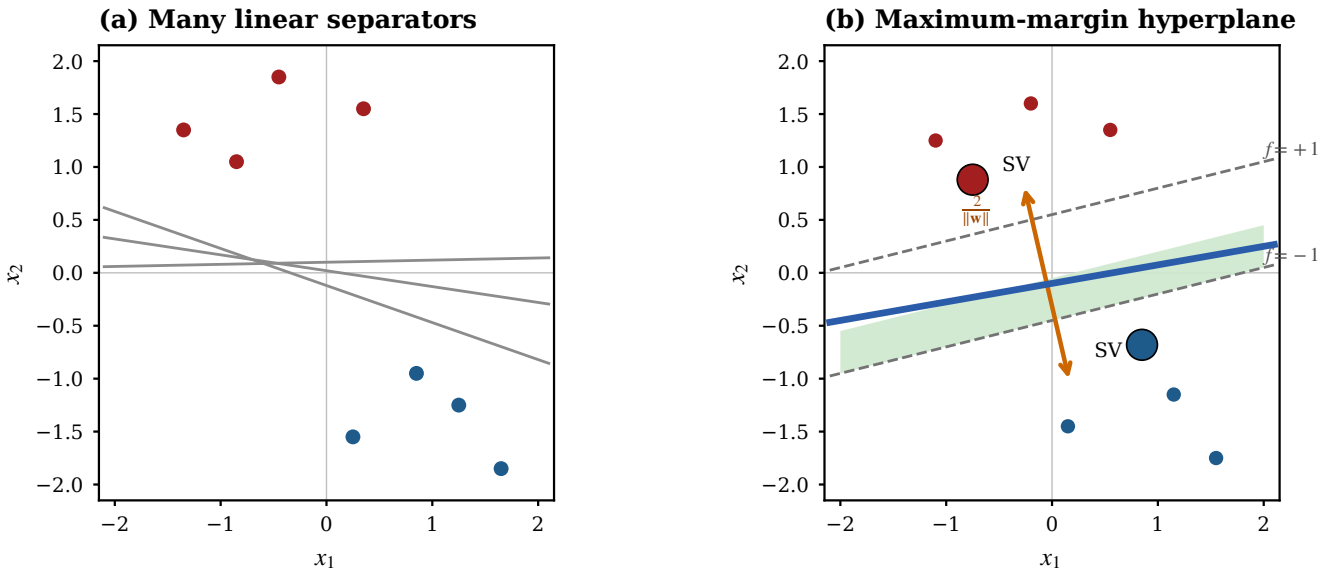


FIG. 13. Two-dimensional schematic of linear classification (vector PDF from `scripts/plot_svm_figures.py`). **(a)** A perceptron only needs *some* hyperplane that separates the classes, so many slopes are admissible (gray). **(b)** A hard-margin SVM picks the separator that maximizes the distance between class clouds; the shaded band is between $f = \pm 1$. Support vectors (marked "SV") lie on the dashed margin lines and have active constraints in (87).

outside the $f = \pm 1$ slab. The primal problem is

$$\min_{\mathbf{w},b} \quad \tfrac{1}{2}\|\mathbf{w}\|^2 \tag{86}$$

$$\text{s.t.} \quad y_k\big(\langle \mathbf{w}, \mathbf{x}_k\rangle + b\big) \geq 1, \quad k = 1, \ldots, M. \tag{87}$$

Any constraint satisfied with equality defines a *support vector*; non-support points can be removed without changing the optimal $(\mathbf{w}, b)$. Introducing multipliers for the margin inequalities yields a convex *dual* quadratic program whose coefficients are inner products $\langle \mathbf{x}_i, \mathbf{x}_j\rangle$—this is the classical route to the kernel trick, spelled out after LS-SVM in Sec. IX F. For quantum linear-algebra pipelines (HHL-class), it is often more transparent to first use *least-squares* SVM (Sec. IX C), where inequalities are replaced by equalities and training collapses to a single Hermitian linear system in $(b, \boldsymbol{\alpha})$.

c. Soft-margin SVM. Real data often overlap. Nonnegative slack variables ξ_k tolerate points inside the margin or even on the wrong side of H, while a penalty $C\sum_k \xi_k$ limits violations:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{k=1}^{M} \xi_k \tag{88}$$

$$\text{s.t.} \quad y_k\big(\langle \mathbf{w}, \mathbf{x}_k\rangle + b\big) \geq 1 - \xi_k, \quad \xi_k \geq 0. \tag{89}$$

Large C enforces a stricter margin (risk of over-fitting); small C allows more misclassification slack but fattens the effective margin.

We next treat least-squares SVM because its KKT conditions form a *linear* system—the classical object coupled to HHL in the Rebentrost–Mohseni–Lloyd quantum LS-SVM construction—before recording NISQ-oriented feasibility constraints, then the standard dual SVM and kernel trick (a different convex QP).

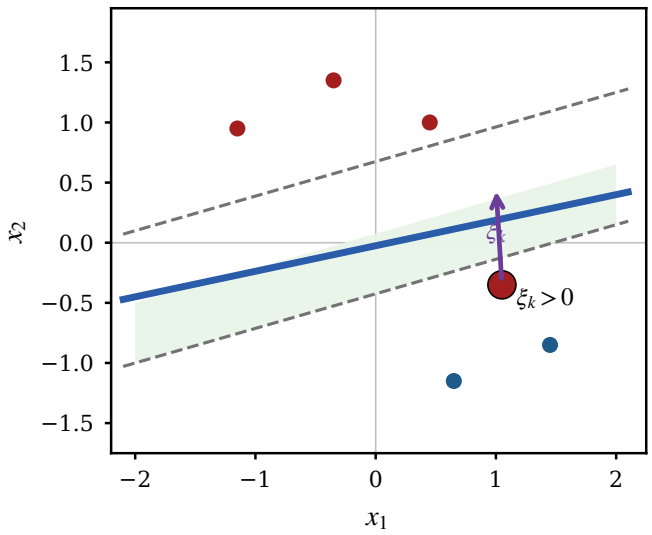


FIG. 14. Soft-margin notion: most positive-class points satisfy $y_k f(\mathbf{x}_k) \geq 1$, but an outlier (red) lies inside the margin tube or on the wrong side. Its slack ξ_k measures how much the margin constraint (89) is relaxed.

C. Least-squares SVM as a linear system

Least-squares SVM (LS-SVM) replaces margin inequalities by *equality* constraints with squared loss on slack [12]. One seeks

$$\min_{\mathbf{w},b,\mathbf{e}} \quad \tfrac{1}{2}\|\mathbf{w}\|^2 + \tfrac{\gamma}{2}\sum_{k=1}^{M} e_k^2 \tag{90}$$

$$\text{s.t.} \quad y_k\big(\langle \mathbf{w}, \mathbf{x}_k\rangle + b\big) = 1 - e_k, \quad k = 1, \ldots, M, \tag{91}$$

with regularization parameter $\gamma > 0$. Introduce Lagrange multipliers α_k via the Lagrangian (sign chosen so that α_k

coincides with the usual LS-SVM dual unknowns [12])

$$\mathcal{L} = \tfrac{1}{2}\|\mathbf{w}\|^2 + \tfrac{\gamma}{2}\sum_{k=1}^{M} e_k^2 - \sum_{k=1}^{M} \alpha_k \Big[y_k\big(\langle \mathbf{w}, \mathbf{x}_k\rangle + b\big) + e_k - 1 \Big]. \quad (92)$$

This is a *convex* quadratic program: the objective $\frac{1}{2}\|\mathbf{w}\|^2 + \frac{\gamma}{2}\sum_k e_k^2$ is strictly convex in $(\mathbf{w}, \mathbf{e})$, and the constraints are affine equalities in $(\mathbf{w}, b, \mathbf{e})$. For such equality-constrained convex problems, the Karush–Kuhn–Tucker (KKT) conditions—stationarity of the Lagrangian with respect to all primal variables, together with primal feasibility—are *necessary and sufficient* for a global minimizer (there is no primal–dual gap to close by separate duality theory at this step).

Stationarity gives the KKT conditions:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{0} \;\Rightarrow\; \mathbf{w} = \sum_{k=1}^{M} \alpha_k y_k \mathbf{x}_k, \quad (93)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \;\Rightarrow\; \sum_{k=1}^{M} \alpha_k y_k = 0, \quad (94)$$

$$\frac{\partial \mathcal{L}}{\partial e_k} = 0 \;\Rightarrow\; e_k = \frac{\alpha_k}{\gamma}. \quad (95)$$

Because every constraint is an equality, there are no inequality multipliers and no complementary slackness conditions of the form $\mu_k g_k(\cdot) = 0$ as in the soft-margin SVM; once (93)–(95) hold, the only remaining KKT content is primal feasibility, i.e. enforcing $y_k(\langle \mathbf{w}, \mathbf{x}_k\rangle + b) = 1 - e_k$ for all k.

Substituting (93)–(95) into the equality constraints yields, for each k,

$$y_k\left(\sum_{j=1}^{M} \alpha_j y_j \langle \mathbf{x}_j, \mathbf{x}_k\rangle + b\right) = 1 - \frac{\alpha_k}{\gamma}. \quad (96)$$

Define the *kernel matrix* $K_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j\rangle$ (linear kernel) and $\Omega_{ij} = y_i y_j K_{ij}$. Vectorially,

$$\Omega\boldsymbol{\alpha} + b\,\mathbf{y} = \mathbf{1}_M - \gamma^{-1}\boldsymbol{\alpha} \quad\Longleftrightarrow\quad (\Omega + \gamma^{-1} I_M)\boldsymbol{\alpha} + b\,\mathbf{y} = \mathbf{1}_M, \quad (97)$$

together with $\mathbf{y}^\top\boldsymbol{\alpha} = 0$ from (94). Stacking bias and multipliers gives the symmetric $(M+1)\times(M+1)$ system

$$\begin{pmatrix} 0 & \mathbf{y}^\top \\ \mathbf{y} & \Omega + \gamma^{-1} I_M \end{pmatrix} \begin{pmatrix} b \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{1}_M \end{pmatrix}, \quad (98)$$

a. Reading the KKT system. The unknown vector is $\mathbf{z} = (b, \boldsymbol{\alpha})^\top$, so (98) is an $(M+1)\times(M+1)$ linear system. The *first row* reproduces bias stationarity (94): $\mathbf{y}^\top\boldsymbol{\alpha} = 0$, i.e. the label-weighted dual coefficients must sum to zero. The *remaining M rows* are the training constraints after eliminating $\mathbf{w}$ and $\mathbf{e}$: each row states that the signed margin $y_k(\sum_j \alpha_j y_j K_{jk} + b)$ equals $1 - \alpha_k/\gamma$, with $K_{jk} = \langle \mathbf{x}_j, \mathbf{x}_k\rangle$. The diagonal shift $\gamma^{-1} I_M$ is the algebraic shadow of the squared slack penalty $\frac{\gamma}{2}\sum_k e_k^2$ via $e_k = \alpha_k/\gamma$. The coefficient matrix is real symmetric (hence Hermitian), but the leading 2×2 block pattern means it is generally *indefinite*; invertibility and the condition number of F depend on the data and on γ, and must be checked before invoking linear-system solvers (classical or HHL-class).

The key point for QML is that (98) is *a linear system with structured Hermitian data*, so its solution vector can in principle be prepared via HHL-style methods given suitable oracle access to the matrix.

D. Quantum least-squares SVM and classification

Rebentrost, Mohseni, and Lloyd [13] promote the least-squares SVM linear system to a *quantum* subroutine while keeping the same classical optimality conditions (Sec. IX C). The construction separates into (i) preparing the optimal KKT vector in a quantum register and (ii) turning the SVM decision score into overlaps measurable on a quantum device. Figure 15 summarizes the workflow.

a. Quantum formulation of the KKT system. The vector $\mathbf{z} = (b, \alpha_1, \ldots, \alpha_M)^\top$ solving (98) is exactly the collection of Lagrange multipliers (plus bias) that satisfy the KKT stationarity and feasibility equations derived above; quantum algorithms aim to prepare $\mathbf{z}^\star$ coherently rather than to solve for it by direct Gaussian elimination on a classical machine. Writing (98) as a single matrix equation,

$$F\,\mathbf{z} = \mathbf{r}, \qquad F := \begin{pmatrix} 0 & \mathbf{y}^\top \\ \mathbf{y} & \Omega + \gamma^{-1} I_M \end{pmatrix}, \qquad \mathbf{r} := \begin{pmatrix} 0 \\ \mathbf{1}_M \end{pmatrix}, \quad (99)$$

where $\Omega_{ij} = y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle$ for the linear kernel. The block F is real symmetric, hence Hermitian, so it falls inside the scope of the HHL input model once suitable sparsity or block-encoded access is assumed [5, 7].

Let $|r\rangle$ denote a normalized amplitude encoding of $\mathbf{r}$ (after possibly appending dummy components if one prefers a power-of-two dimension). HHL with $A = F$ produces a state proportional to

$$|z\rangle \;\propto\; F^{-1}\,|r\rangle\,, \quad (100)$$

matching the solution $\mathbf{z}^\star = (b^\star, \boldsymbol{\alpha}^\star)^\top$ up to global normalization and finite-error phase estimation of e^{iFt}. In practice one does not always *read* every $\alpha_k^\star$ classically; instead $|z\rangle$ is kept coherent for subsequent controlled rotations or as an input to overlap-based inference, trading full tomography for the specific linear functional needed at test time.

b. Decision score and swap-test overlaps. The classical LS-SVM classifier uses the affine score

$$s(\tilde{\mathbf{x}}) \;:=\; b^\star + \sum_{k=1}^{M} \alpha_k^\star\, y_k\, \langle \mathbf{x}_k, \tilde{\mathbf{x}}\rangle, \quad (101)$$

and predicts $\tilde{y} = \operatorname{sign} s(\tilde{\mathbf{x}})$. Using $\mathbf{w}^\star = \sum_k \alpha_k^\star y_k \mathbf{x}_k$ recovers the equivalent form $s(\tilde{\mathbf{x}}) = \langle \mathbf{w}^\star, \tilde{\mathbf{x}}\rangle + b^\star$.

c. Lecture-style state construction and Hadamard test. The lecture notes implement the score evaluation by embedding the bias and training coefficients into a single "index" superposition state produced by HHL. Writing the HHL output for the KKT solution schematically as an amplitude encoding of $(b^\star, \alpha_1^\star, \ldots, \alpha_M^\star)$,

$$|b, \boldsymbol{\alpha}\rangle \;:=\; \frac{1}{\sqrt{N_b}}\Big(b^\star\,|0\rangle + \sum_{k=1}^{M} \alpha_k^\star\,|k\rangle\Big), \qquad N_b := |b^\star|^2 + \sum_{k=1}^{M} |\alpha_k^\star|^2, \quad (102)$$

one then entangles each training index k with its feature vector $\mathbf{x}_k \in \mathbb{R}^d$ by a data-loading unitary (QRAM model):

$$|k\rangle\,|0\rangle \;\mapsto\; |k\rangle\,|\mathbf{x}_k\rangle \;\approx\; |k\rangle \sum_{j=1}^{d} (x_k)_j\,|j\rangle\,, \quad (103)$$

and similarly for the test point $\tilde{\mathbf{x}}$. With this convention, the lecture constructs the joint "model" state

$$|u\rangle \;\coloneqq\; \frac{1}{\sqrt{N_u}}\Big(b^\star\,|0\rangle\,|0\rangle + \sum_{k=1}^{M}\alpha_k^\star y_k\;|k\rangle\,|\mathbf{x}_k\rangle\Big),$$
$$N_u \coloneqq |b^\star|^2 + \sum_{k=1}^{M}|\alpha_k^\star|^2\|\mathbf{x}_k\|^2, \tag{104}$$

and the "query" state for the new data

$$|\tilde{x}\rangle \;\coloneqq\; \frac{1}{\sqrt{N_{\tilde{x}}}}\Big(|0\rangle\,|0\rangle + \sum_{k=1}^{M}|k\rangle\,|\tilde{\mathbf{x}}\rangle\Big),$$
$$N_{\tilde{x}} \coloneqq 1 + M\|\tilde{\mathbf{x}}\|^2. \tag{105}$$

Taking the inner product and using orthogonality $\langle k'|k\rangle = \delta_{kk'}$ gives

$$\langle\tilde{x}|u\rangle = \frac{1}{\sqrt{N_u N_{\tilde{x}}}}\left(b^\star + \sum_{k=1}^{M}\alpha_k^\star\, y_k\,\langle\mathbf{x}_k,\tilde{\mathbf{x}}\rangle\right), \tag{106}$$

which agrees with the LS-SVM score (101) up to the positive prefactor $1/\sqrt{N_u N_{\tilde{x}}}$ (for real feature vectors, $\langle\tilde{\mathbf{x}}|\mathbf{x}_k\rangle = \langle\mathbf{x}_k,\tilde{\mathbf{x}}\rangle$). Thus the lecture-style rule

$$\tilde{y} = \operatorname{sign}\big(\langle\tilde{x}|u\rangle\big) = \operatorname{sign}\big(s(\tilde{\mathbf{x}})\big) \tag{107}$$

matches classical LS-SVM classification. The overlap $\langle\tilde{x}|u\rangle$ is estimated by the Hadamard test (Sec. VI D), which returns the real (and, with a phase shift, imaginary) part of $\langle\tilde{x}|u\rangle$ rather than only its magnitude.

Quantumly, each term $\langle\mathbf{x}_k,\tilde{\mathbf{x}}\rangle$ may be estimated by preparing feature states $|\psi_k\rangle \approx |\mathbf{x}_k\rangle/\|\mathbf{x}_k\|$, $|\tilde{\psi}\rangle \approx |\tilde{\mathbf{x}}\rangle/\|\tilde{\mathbf{x}}\|$ and invoking the pure-state swap test (Sec. VI D, (50); mixed-state extension Sec. IX G 0 g). Denoting the control outcome probability of seeing $|0\rangle_a$ on the ancilla by p_0, one has the standard relation

$$p_0 = \frac{1+|\langle\psi_k|\tilde{\psi}\rangle|^2}{2}, \qquad \text{hence} \qquad |\langle\psi_k|\tilde{\psi}\rangle|^2 = 2p_0 - 1 \tag{108}$$

(in the ideal, unmitigated protocol). Recovering the *signed* inner product needed in (101) may require additional phase information or data preprocessing so that all overlaps share a known global phase convention; the lecture-level takeaway is that the bottleneck becomes accurate quantum evaluation of similarities $\langle\mathbf{x}_k,\tilde{\mathbf{x}}\rangle$, possibly in superposition over k, before combining them with the trained weights $\alpha_k^\star y_k$ and bias $b^\star$.

d. Kernelized QSVM. Nothing in (99) is specific to dot products: replacing $\Omega_{ij} = y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)$ with a positive semidefinite kernel embeds nonlinear margins while preserving the Hermitian KKT matrix for HHL-class solvers. Quantum feature maps $K(\mathbf{x},\mathbf{x}') = \langle\phi(\mathbf{x})|\phi(\mathbf{x}')\rangle$ connect this picture to modern kernel-based QML (Sec. IX G) [14].

e. Assumptions and caveats. The advertised polylogarithmic scaling presupposes efficient preparation of $|r\rangle$ (QRAM-like or structured data access), sparse or block-encoded Hamiltonian simulation of e^{-iFt}, and a moderate condition number κ of F so that HHL post-selection (82) remains tractable after amplitude amplification. Classical *dequantization* shows that comparable sampling models can erase the exponential separation on some linear-algebra problems [5], and constant-depth overlap readouts on NISQ devices inherit hardware noise not reflected in idealized complexity displays. These issues are discussed in depth in Refs. [13, 15].

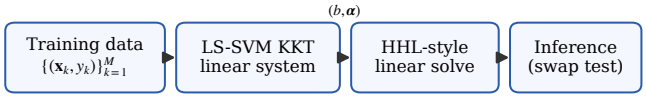


FIG. 15. Block view of Rebentrost–Mohseni–Lloyd quantum LS-SVM [13]: encode the training problem as a structured Hermitian system, prepare the solution amplitudes with HHL-class primitives, then classify new features using estimated overlaps (Sec. VI D).

E. Quantum Advantage in Machine Learning

Quantum machine learning (QML) investigates whether quantum computers can accelerate or qualitatively improve machine learning tasks [15]. Three criteria define meaningful quantum advantage:

1. *Physical feasibility* (with or without QEC),
2. *Computational advantage* over the best known classical algorithm,
3. *Industry relevance* for real-world ML tasks.

Meeting all three simultaneously is an open problem. Quantum speedups in supervised learning often rely on quantum RAM (QRAM) for efficient data loading—a hardware resource whose practical feasibility remains uncertain.

a. Near-term QML families (lecture landscape). Introductory decks group practical approaches as follows: *(i)* quantum kernel methods, which estimate Gram-matrix entries (or related overlaps) and pair with classical convex solvers; *(ii)* variational QML, optimizing continuous circuit parameters without assembling the full kernel matrix; *(iii)* quantum annealing and QAOA for combinatorial structure; and *(iv)* quantum-enhanced sampling in Monte Carlo workflows. These occupy different regions of the "NISQ-feasible / provably faster / industrially relevant" diagram emphasized in Week 11-style surveys [1, 15].

F. Dual SVM Formulation and Kernel Trick

Having completed the HHL–LS-SVM thread and summarized hurdles to practical quantum speedups (Sec. IX E), we return to the classical hinge-loss SVM from Sec. IX B. Its *dual* is the formulation implemented by most kernel SVM solvers and by hybrid pipelines that feed a quantum kernel matrix into a classical optimizer.

While the primal soft-margin SVM formulation optimizes over the weight vector $\mathbf{w}$ and bias b, its dimensionality grows linearly with the number of features d. For high-dimensional data, solving the *dual problem* becomes computationally favorable because the number of optimization variables scales with the number of training examples M rather than the feature dimension.

By introducing Lagrange multipliers $\alpha_i \geq 0$, the primal problem can be converted into the dual formulation:

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{M}\alpha_i - \tfrac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M}\alpha_i\alpha_j y_i y_j\langle\mathbf{x}_i,\mathbf{x}_j\rangle$$
$$\text{s.t.} \quad \sum_{i=1}^{M}\alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad \forall i. \tag{109}$$

Once the optimal multipliers $\boldsymbol{\alpha}^\star$ are found, the optimal weight vector is recovered as $\mathbf{w}^\star = \sum_{i=1}^{M} \alpha_i^\star y_i \mathbf{x}_i$.

a. Recovering the bias. Complementary slackness implies that any training index with $0 < \alpha_i^\star < C$ lies on the margin, so $y_i(\langle \mathbf{w}^\star, \mathbf{x}_i \rangle + b^\star) = 1$ and hence

$$b^\star = y_i - \sum_{j=1}^{M} \alpha_j^\star y_j \, \langle \mathbf{x}_j, \mathbf{x}_i \rangle. \tag{110}$$

In practice one averages (110) over several such points for stability.

Crucially, the training data vectors $\mathbf{x}_i$ appear in Eq. (109) *only* through their inner products $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$. Likewise, the classification rule for a new data point $\tilde{\mathbf{x}}$ becomes:

$$\tilde{y} = \operatorname{sign}\left(\sum_{i=1}^{M} \alpha_i^\star y_i \langle \mathbf{x}_i, \tilde{\mathbf{x}} \rangle + b^\star \right). \tag{111}$$

This reliance exclusively on inner products invites the *kernel trick*. If we map our data to a higher-dimensional space via a non-linear feature map $\phi(\mathbf{x})$, we do not need to compute the coordinates in that space explicitly. Instead, we only need to compute a kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. The Gram matrix (or kernel matrix) $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ encapsulates all the necessary similarity measures for the dual SVM.

b. Worked polynomial kernel. For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$, the function $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^2$ is a valid kernel because it is the inner product of the six-dimensional feature vector

$$\phi(\mathbf{x}) = \big(1, \ \sqrt{2}\,x_1, \ \sqrt{2}\,x_2, \ \sqrt{2}\,x_1 x_2, \ x_1^2, \ x_2^2\big)^\top, \tag{112}$$

matching the explicit expansion used in standard slides on nonlinear separation.

G. Quantum Feature Maps and Quantum Kernels

Machine-learning inputs must be represented inside a Hilbert space before any quantum model or quantum kernel is evaluated. Lecture treatments allow a *general* encoding $\rho(\mathbf{x}) = \sum_k p_k \, |\psi_k(\mathbf{x})\rangle \langle \psi_k(\mathbf{x})|$ (convex combinations of pure states); pure feature maps $|\phi(\mathbf{x})\rangle$ are the special case $\rho(\mathbf{x}) = |\phi(\mathbf{x})\rangle \langle \phi(\mathbf{x})|$. Two recurring patterns in that setting are *amplitude encoding* (information in the 2^n amplitudes of n qubits) and *computational / angle encoding* (information in bit strings or local rotation angles), followed by optional *parameterized feature-map* unitaries $U_\Phi(\mathbf{x})$ that entangle qubits so that $|\phi(\mathbf{x})\rangle \coloneqq U_\Phi(\mathbf{x}) \, |0\rangle^{\otimes n}$ is a genuine quantum feature vector [5, 14].

a. Computational (basis) encoding. Let $\mathbf{x} = (x_1, \ldots, x_d) \in \{0,1\}^d$. The *basis-encoded* state is

$$|\psi_{\text{bas}}(\mathbf{x})\rangle \ = \ |x_1\rangle \otimes \cdots \otimes |x_d\rangle \ = \ \bigotimes_{j=1}^{d} X^{x_j} \, |0\rangle, \tag{113}$$

so each classical bit occupies one qubit in the computational basis. Figure 17(top) sketches the idea for $d = 3$.

b. Amplitude encoding ("ket vector"). Let $\mathbf{v} \in \mathbb{C}^N$ satisfy $\sum_{j=0}^{N-1} |v_j|^2 = 1$ with $N = 2^n$. The *amplitude-encoded* n-qubit state is

$$|\psi_{\text{amp}}(\mathbf{v})\rangle \ = \ \sum_{j=0}^{N-1} v_j \, |j\rangle, \tag{114}$$

where $|j\rangle$ denotes the n-qubit computational basis kets indexed by binary strings of length n (pad $\mathbf{v}$ with zeros if the data length is not a power of two). For real data $\mathbf{x} \in \mathbb{R}^N$ one first forms a normalized vector $\mathbf{v} = \mathbf{x}/\|\mathbf{x}\|_2$ (global scale drops out of many overlap-based kernels). By the Solovay–Kitaev picture there always exists a unitary $U_\mathbf{v} \in U(2^n)$ such that

$$U_\mathbf{v} \ |0\rangle^{\otimes n} = |\psi_{\text{amp}}(\mathbf{v})\rangle, \tag{115}$$

but constructing $U_\mathbf{v}$ from arbitrary $\mathbf{v}$ may require $\Theta(2^n)$ elementary gates in the worst case unless the amplitudes have additional structure; algorithms such as HHL assume *efficient* amplitude preparation as part of the quantum input model [7]. Course notes on QML often stress that exponential compression via amplitude encoding must be weighed against this state-preparation overhead, which is easy to gloss over when quoting only query-complexity speedups.

c. Indexed batch of vectors (matrix amplitude encoding). Given training vectors $\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(M-1)} \in \mathbb{R}^N$, one can entangle a sample index with a feature register,

$$|\Psi\rangle_{\text{batch}} \ \propto \ \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_n^{(m)} \ |n\rangle \, |m\rangle, \tag{116}$$

after suitable normalization (cf. stacked amplitude-encoding slides). The qubit count is $\lceil \log_2 N \rceil + \lceil \log_2 M \rceil$, but preparing $|\Psi\rangle_{\text{batch}}$ from classical tables still incurs a cost analyzed separately from the algorithm that consumes it.

The minimal $n = 1$ example illustrates the idea completely: given $\mathbf{v} = (v_0, v_1)^\top$ with $v_0, v_1 \geq 0$ and $v_0^2 + v_1^2 = 1$, choose θ such that $v_0 = \cos(\theta/2)$ and $v_1 = \sin(\theta/2)$; then

$$R_y(\theta) \, |0\rangle \ = \ \cos\tfrac{\theta}{2} \, |0\rangle + \sin\tfrac{\theta}{2} \, |1\rangle \ = \ v_0 \, |0\rangle + v_1 \, |1\rangle, \tag{117}$$

using $R_y(\theta) = e^{-i\theta Y/2}$ as in (77). Figure 17(bottom-left) shows this single-qubit loader. For $n > 1$, a standard construction uses a tree of controlled R_y (and, for complex amplitudes, R_z) rotations on successive qubits [2]; Figure 18 shows the first split for $n = 2$ in a binary tree (angles $\theta_0, \theta_{00}, \theta_{01}$ depend on the target $(v_{00}, v_{01}, v_{10}, v_{11})^\top$).

Two encoded vectors $\mathbf{v}, \mathbf{w}$ satisfy $\langle \psi_{\text{amp}}(\mathbf{v}) | \psi_{\text{amp}}(\mathbf{w}) \rangle = \langle \mathbf{v}, \mathbf{w} \rangle$ (Hermitian inner product in $\mathbb{C}^N$), so overlaps $|\langle \cdots \rangle|^2$ used in kernels recapture classical cosine-similarity structure at the level of amplitudes.

d. Angle (rotational / tensor-product) encoding. A lightweight alternative stores each coordinate in a *separate* qubit's rotation angle:

$$|\psi_{\text{ang}}(\mathbf{x})\rangle \ = \ \bigotimes_{j=1}^{d} R_y\big(\varphi(x_j)\big) \, |0\rangle, \tag{118}$$

where φ rescales features (e.g. $\varphi(x_j) = \pi x_j$ for $x_j \in [0,1]$). The state is a product state, so entanglement—if desired for expressivity—must be introduced by *subsequent* entangling layers inside a feature map $U_\Phi(\mathbf{x})$ or by a variational ansatz [14]. Figure 17(bottom-right) shows $d = 3$ parallel R_y gates.

e. Hamiltonian and IQP-style feature maps. A widely used non-product template applies data-dependent *Hamiltonian evolution* before measurement-based kernels:

$$\begin{aligned} |\phi(\mathbf{x})\rangle \ &= \ U_\Phi(\mathbf{x}) \ |0\rangle^{\otimes n}, \\ U_\Phi(\mathbf{x}) \ &= \ \prod_{\ell=1}^{L} \exp\Big(-i \, \eta_\ell \, H_\ell(\mathbf{x})\Big), \end{aligned} \tag{119}$$

where each $H_\ell(\mathbf{x})$ is a Hermitian operator built from tensor products of Pauli matrices with coefficients (entangling "ZZ", "XXX", etc. structures in lecture notes and in IBM Qiskit-style feature circuits). Taking all $H_\ell(\mathbf{x}) \propto H_{\text{tot}}(\mathbf{x})$ and $\eta_\ell = t$ recovers the single-time "Hamiltonian encoding"

$$|\phi(\mathbf{x})\rangle = \exp\big(-iH(\mathbf{x})\,t\big)\,|0\rangle^{\otimes n}, \tag{120}$$

often drawn schematically as a block $\mathrm{U}(\mathbf{x}, t)$ on $|0\rangle^{\otimes n}$.

f. Example: two-qubit ZZ entangling layer (schematic). Many annotated lecture slides illustrate a *Hadamard sandwich* followed by single-qubit R_Z rotations and controlled-X gates implementing effective ZZ couplings. A prototypical one-layer pattern acts on two qubits as Repeating such layers *interleaved* with independent trainable unitaries yields *data re-uploading* circuits discussed in the QML literature [14].

The generic *quantum feature map* statement is then

$$\mathbf{x} \mapsto |\phi(\mathbf{x})\rangle \in \mathcal{H}, \qquad |\phi(\mathbf{x})\rangle \coloneqq U_\Phi(\mathbf{x})\,|0\rangle^{\otimes n}, \tag{121}$$

with U_Φ built from any combination of the ingredients above (basis load, amplitude preparation, local rotations, and Hamiltonian/ZZ entanglers).

The corresponding *quantum kernel* is given by the state overlap (or the Hilbert-Schmidt inner product of their pure-state density matrices):

$$k(\mathbf{x}, \mathbf{y}) = |\,\langle\phi(\mathbf{x})|\phi(\mathbf{y})\rangle\,|^2 = \mathrm{Tr}(\rho(\mathbf{x})\rho(\mathbf{y})). \tag{122}$$

For M training points, assembling (122) into a Gram matrix yields K_{ij} in (127). In the NISQ (Noisy Intermediate-Scale Quantum) era, calculating distances in exponentially large Hilbert spaces offers a concrete pathway to provable quantum advantage. If the unitary evolution $\exp(-iH(\mathbf{x})t)$ is hardware-efficient but classically intractable to simulate, the resulting quantum kernel matrix can be fed into a classical SVM solver. This hybrid approach enables non-linear classification boundaries that are difficult or impossible to capture with purely classical algorithms [14].

g. Density-matrix swap test, Pauli coordinates, and quantum kernel matrices. For *mixed* states ρ_1, ρ_2 on the two data registers, the same Fredkin circuit (Fig. 19) estimates the Hilbert–Schmidt overlap $\mathrm{Tr}(\rho_1\rho_2)$. One initializes

$$\rho_{\text{in}} = |0\rangle\langle 0|_a \otimes \rho_1 \otimes \rho_2, \tag{123}$$

applies $H_a = H \otimes I_4$ so that

$$\rho^{(1)} = \frac{1}{2}\Big(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + |1\rangle\langle 1|\Big)_a \otimes \rho_1 \otimes \rho_2, \tag{124}$$

then U_{CSWAP} from (54), then H_a again. Writing $M \coloneqq \rho_1 \otimes \rho_2$ and using $U_{\text{CSWAP}}(|a\rangle\langle b| \otimes M)U_{\text{CSWAP}}^\dagger = |a\rangle\langle b| \otimes (V_a M V_b^\dagger)$ with $V_0 = I_4, V_1 = S$,

$$\begin{aligned}\rho^{(2)} &= U_{\text{CSWAP}}\,\rho^{(1)}\,U_{\text{CSWAP}}^\dagger \\ &= \frac{1}{2}\Big[\,|0\rangle\langle 0| \otimes M + |0\rangle\langle 1| \otimes MS^\dagger \\ &\qquad + |1\rangle\langle 0| \otimes SM + |1\rangle\langle 1| \otimes SMS^\dagger\Big].\end{aligned} \tag{125}$$

Since $SMS^\dagger = S$ and $S(\rho_1 \otimes \rho_2)S = \rho_2 \otimes \rho_1$, the last term is $|1\rangle\langle 1| \otimes (\rho_2 \otimes \rho_1)$. Measuring the ancilla in the computational basis gives

$$P(0) = \frac{1}{2}\big(1 + \mathrm{Tr}(\rho_1\rho_2)\big), \qquad \mathrm{Tr}(\rho_1\rho_2) = 2P(0) - 1, \tag{126}$$

by the same trace bookkeeping as in the pure-state derivation, now with $\mathrm{Tr}(S(\rho_1 \otimes \rho_2)) = \mathrm{Tr}(\rho_1\rho_2)$.

For sample data $\{x_1, \ldots, x_M\}$ and a quantum feature map $x \mapsto \rho(x)$ (e.g. $\rho(x) = |\phi(x)\rangle\langle\phi(x)|$ for pure encodings), the *quantum kernel matrix* $K \in \mathbb{R}^{M\times M}$ with entries

$$K_{ij} \;\coloneqq\; \mathrm{Tr}\big(\rho(x_i)\,\rho(x_j)\big) \tag{127}$$

collects all pairwise overlaps estimable via swap tests (or tomography). When $\rho(x_i)$ are pure, $K_{ij} = |\,\langle\phi(x_i)|\phi(x_j)\rangle\,|^2$ and agrees with (50).

Pauli basis and real coordinates. Let $\{P_\mu\}_{\mu=1}^{4^n}$ enumerate Hermitian Pauli strings $P_\mu \in \{I, X, Y, Z\}^{\otimes n}$ (fixed ordering). They obey

$$\mathrm{Tr}(P_\mu P_\nu) = 2^n\,\delta_{\mu\nu}, \tag{128}$$

since single-qubit factors satisfy $\mathrm{Tr}(\sigma_a\sigma_b) = 2\delta_{ab}$ for $\sigma_a, \sigma_b \in \{I, X, Y, Z\}$ (e.g. $\mathrm{Tr}(X) = \mathrm{Tr}(Y) = \mathrm{Tr}(Z) = 0$, $\mathrm{Tr}(I) = 2$, and distinct Pauli products yield a traceless Pauli). A convenient multiplication table for one qubit is

$\cdot$	I	X	Y	Z
I	I	X	Y	Z
X	X	I	iZ	$-iY$
Y	Y	$-iZ$	I	iX
Z	Z	iY	$-iX$	I

Every n-qubit Hermitian operator (in particular any density matrix ρ) expands as

$$\rho = \frac{1}{2^n}\sum_{\mu=1}^{4^n}\mathrm{Tr}(P_\mu\rho)\,P_\mu. \tag{129}$$

Define a real vector $\vec{\phi}(\rho) \in \mathbb{R}^{4^n}$ by $\phi_\mu(\rho) \coloneqq \mathrm{Tr}(P_\mu\rho)/\sqrt{2^n}$. Then

$$\mathrm{Tr}(\rho\sigma) = \frac{1}{2^n}\sum_{\mu=1}^{4^n}\mathrm{Tr}(P_\mu\rho)\,\mathrm{Tr}(P_\mu\sigma) = \big\langle\vec{\phi}(\rho),\,\vec{\phi}(\sigma)\big\rangle_{\mathbb{R}^{4^n}}, \tag{130}$$

so estimating $K_{ij} = \mathrm{Tr}(\rho_i\rho_j)$ is equivalent to a Euclidean inner product of 4^n-dimensional classical feature vectors once the Pauli expectations $\mathrm{Tr}(P_\mu\rho)$ are known.

H. Other landmark QML algorithms

Beyond the LS-SVM construction, we briefly recall two widely cited directions.

a. Quantum principal component analysis. Lloyd, Mohseni, and Rebentrost [16] estimate principal components of a density matrix ρ in time $O(\log N)$ under a strong QRAM input model—exponentially faster than naive classical diagonalization in matrix dimension.

b. Variational quantum algorithms. VQAs use short parameterized circuits $U(\boldsymbol{\theta})$ and optimize $\boldsymbol{\theta}$ classically to minimize $C(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|O|\psi(\boldsymbol{\theta})\rangle$ [17]. VQAs are the primary paradigm for NISQ-era QML [1].

I. Open Questions and Future Outlook

Despite significant theoretical progress, fundamental questions remain:

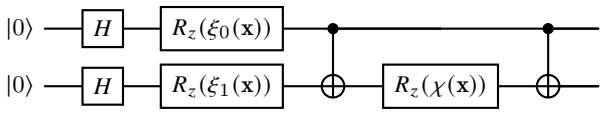


FIG. 16. Schematic two-qubit feature-map layer used in lecture demonstrations of quantum kernels: Hadamards create $|{+}{+}\rangle$, local R_z embed coordinates, and the middle CX–R_z–CX block introduces effective $\exp(-i\chi Z\otimes Z)$ coupling (exact angles ξ_0, ξ_1, χ depend on the chosen map and normalization of $\mathbf{x}$).

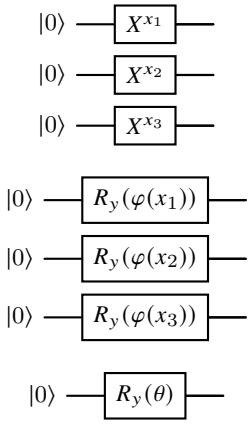


FIG. 17. **(Top)** Basis encoding (113): X^{x_j}. **(Middle)** Angle encoding (118): one R_y per feature. **(Bottom)** Amplitude encoding on one qubit (117).

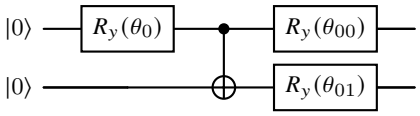


FIG. 18. First branching step in a standard binary tree for preparing a two-qubit amplitude-encoded state $v_{00}|00\rangle + v_{01}|01\rangle + v_{10}|10\rangle + v_{11}|11\rangle$: the top $R_y(\theta_0)$ splits weight between the $|0\cdot\rangle$ and $|1\cdot\rangle$ branches; conditional rotations on the lower wire complete the $|00\rangle, |01\rangle$ (and similarly $|10\rangle, |11\rangle$ elsewhere) subspaces [2]. Angles $(\theta_0, \theta_{00}, \theta_{01}, \ldots)$ are fixed by the target amplitudes (v_{jk}).

- *Dequantization*: Can classical sampling-based algorithms match quantum speedups [5]?
- *Barren plateaus*: Do VQA gradients vanish exponentially with system size, making training intractable?
- *End-to-end advantage*: Is a provable, QRAM-free quantum speedup achievable on relevant ML tasks?

Progress on these questions will determine whether QML transcends academic interest to become a practical technology.

X. VARIATIONAL QUANTUM MACHINE LEARNING

Variational quantum algorithms (VQAs) constitute one of the most promising paradigms for near-term quantum machine learning. They combine *shallow-depth* parameterized quantum circuits with classical optimization loops, sidestepping the need for deep coherent evolution while still exploiting quantum superposition and entanglement. This chapter develops the key building blocks: the variational quantum eigensolver (VQE) as the prototypical VQA, variational quantum classifiers (VQC), quantum convolutional neural networks (QCNN),

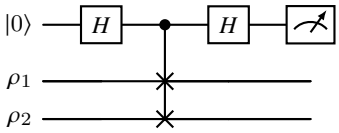


FIG. 19. Density-matrix swap test: data lines carry (generally mixed) states ρ_1, ρ_2. The ancilla marginal obeys $P(0) = \frac{1}{2}(1 + \mathrm{Tr}(\rho_1\rho_2))$ (126). Same U_{CSWAP} as in (54).

the universal approximation theorem for quantum models, analytical gradient formulas, and the connection to quantum state discrimination.

A. Variational Quantum Algorithms

1. Basic idea

The central object in a variational quantum algorithm is a parameterized quantum circuit $U(\boldsymbol{\theta})$ acting on an initial state, typically $|0\rangle^{\otimes n}$. One defines a cost function that depends on the expectation value of a Hermitian observable $H = H^\dagger$:

$$L(\boldsymbol{\theta}) = f\big(h(\boldsymbol{\theta})\big), \tag{131}$$

$$h(\boldsymbol{\theta}) = \mathrm{Tr}\Big(H\,U(\boldsymbol{\theta})\,|\psi\rangle\,\langle\psi|\,U^\dagger(\boldsymbol{\theta})\Big). \tag{132}$$

The function f is typically the identity or a simple post-processing map. The optimization task is

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}'} f\big(h(\boldsymbol{\theta}')\big). \tag{133}$$

The circuit $U(\boldsymbol{\theta})$ is chosen to be of *shallow depth* so that it can be executed on noisy hardware, while the classical optimizer iteratively updates $\boldsymbol{\theta}$ to reduce $L(\boldsymbol{\theta})$.

2. Variational method in quantum mechanics

The word "variational" originates from the *variational method* in quantum mechanics. Consider a Hermitian operator $H = H^\dagger$ with spectral decomposition

$$H\,|h_i\rangle = \lambda_i\,|h_i\rangle\,, \qquad \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{N-1}. \tag{134}$$

The variational principle states that for *any* normalized state $|\psi\rangle$,

$$\langle\psi|H|\psi\rangle \;\ge\; \lambda_{\min}, \tag{135}$$

with equality if and only if $|\psi\rangle$ is the ground state. Thus, minimizing the expectation value $\langle\psi|H|\psi\rangle$ over the Hilbert space yields the ground-state energy $\lambda_{\min}$.

Proof. Expand $|\psi\rangle$ in the eigenbasis of H:

$$|\psi\rangle = \sum_{i=0}^{N-1} c_i |h_i\rangle, \qquad \sum_{i=0}^{N-1} |c_i|^2 = 1. \tag{136}$$

Then

$$\begin{aligned}\langle\psi|H|\psi\rangle &= \sum_{i,j} c_i^* c_j \langle h_i|H|h_j\rangle = \sum_{i,j} c_i^* c_j \lambda_j \langle h_i|h_j\rangle \\ &= \sum_{i=0}^{N-1} |c_i|^2 \lambda_i \;\geq\; \sum_{i=0}^{N-1} |c_i|^2 \lambda_{\min} \;=\; \lambda_{\min}, \end{aligned} \tag{137}$$

where the inequality uses $\lambda_i \geq \lambda_{\min}$ for all i. Equality holds if and only if $c_i = 0$ for all i with $\lambda_i > \lambda_{\min}$, i.e., $|\psi\rangle$ lies entirely in the ground-state subspace. □

In practice, searching over the entire Hilbert space is intractable. Instead, one restricts to a *parametrized subspace*—called the *ansatz*—spanned by states of the form $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|0\rangle^{\otimes n}$. The variational quantum eigensolver (VQE) solves

$$\min_{\boldsymbol{\theta}} \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle, \qquad |\psi(\boldsymbol{\theta})\rangle \;=\; U(\boldsymbol{\theta})\,|0\rangle^{\otimes n}. \tag{138}$$

The quality of the solution depends on whether the true ground state lies within (or near) the ansatz manifold.

B. Quantum Supervised Learning

Quantum supervised learning considers labeled training data $\{(\rho(\mathbf{x}_1), y_1), \ldots, (\rho(\mathbf{x}_N), y_N)\}$ drawn i.i.d. from a distribution $\mathcal{D}$, where $\rho(\mathbf{x})$ is a quantum encoding of the classical feature vector $\mathbf{x}$ and $y_i \in \{-1, +1\}$ is a binary label. The goal is to find a prediction function $f(\mathbf{x}, \boldsymbol{\theta})$ that minimizes the expected risk

$$\mathbb{E}_{(\rho(\mathbf{x}),y)\sim\mathcal{D}} \big| f(\mathbf{x}, \boldsymbol{\theta}) - y \big|. \tag{139}$$

Two major approaches exist:

1. **Quantum kernel methods.** One defines a quantum kernel

$$k(\mathbf{x}, \mathbf{x}') \;=\; \mathrm{Tr}\big(\rho(\mathbf{x})\,\rho(\mathbf{x}')\big), \tag{140}$$

and builds a classical linear model

$$f(\mathbf{x}, \boldsymbol{\alpha}) \;=\; \sum_{i=1}^{N} \alpha_i\, k(\mathbf{x}_i, \mathbf{x}) \;=\; \mathrm{Tr}\left(\sum_{i=1}^{N} \alpha_i\, \rho(\mathbf{x}_i)\,\rho(\mathbf{x})\right), \tag{141}$$

where the coefficients $\boldsymbol{\alpha}$ are determined by a classical solver (cf. Sec. IX G).

2. **Variational quantum algorithms.** One directly parameterizes the predictor as

$$f(\mathbf{x}, \boldsymbol{\theta}) \;=\; \mathrm{Tr}\Big(M\,U(\boldsymbol{\theta})\,\rho(\mathbf{x})\,U^\dagger(\boldsymbol{\theta})\Big), \tag{142}$$

where $M = M^\dagger$ is a measurement observable and $U(\boldsymbol{\theta})$ is a trainable circuit. The parameters $\boldsymbol{\theta}$ are optimized to minimize an empirical loss.

1. Essence of QML: a linear model

A key structural insight is that every QML model of the form $f(\mathbf{x}, \boldsymbol{\theta}) = \mathrm{Tr}(M(\boldsymbol{\theta})\,\rho(\mathbf{x}))$ is a *linear model* in the space of Pauli operators. Let $\{\sigma_i\}_{i=1}^{4^n}$ enumerate all n-qubit Pauli strings $\sigma_i \in \{I, X, Y, Z\}^{\otimes n}$. Since both $M(\boldsymbol{\theta})$ and $\rho(\mathbf{x})$ are Hermitian, they expand as

$$M(\boldsymbol{\theta}) = \sum_{i=1}^{4^n} \alpha_i(\boldsymbol{\theta})\,\sigma_i, \tag{143}$$

$$\rho(\mathbf{x}) = \sum_{j=1}^{4^n} \beta_j(\mathbf{x})\,\sigma_j, \tag{144}$$

with real coefficients α_i, β_j (Hermiticity forces the coefficients to be real because $\mathrm{Tr}(\sigma_i\sigma_j) \propto \delta_{ij}$ and $\mathrm{Tr}(\sigma_i) = 0$ for non-identity Paulis). Using $\mathrm{Tr}(\sigma_i\sigma_j) = 2^n\delta_{ij}$,

$$\begin{aligned} f(\mathbf{x}, \boldsymbol{\theta}) &= \sum_{i=1}^{4^n}\sum_{j=1}^{4^n} \alpha_i(\boldsymbol{\theta})\,\beta_j(\mathbf{x})\;\mathrm{Tr}(\sigma_i\sigma_j) \\ &= 2^n \sum_{i=1}^{4^n} \alpha_i(\boldsymbol{\theta})\,\beta_i(\mathbf{x}). \end{aligned} \tag{145}$$

Thus QML models are linear combinations of the Pauli-feature functions $\beta_i(\mathbf{x})$, with weights $\alpha_i(\boldsymbol{\theta})$ determined by the trainable circuit and measurement. This is the quantum analogue of a linear model in a 4^n-dimensional feature space.

C. Variational Quantum Classifier

1. Hyperplane picture

The variational quantum classifier (VQC) uses the predictor $f(\mathbf{x}, \boldsymbol{\theta})$ from (142) to define a decision boundary. For a bias parameter $b \in \mathbb{R}$, the decision rule for a new data point $\tilde{\mathbf{x}}$ is

$$\tilde{y} \;=\; \mathrm{sign}\Big(\mathrm{Tr}\big(M\,U(\boldsymbol{\theta})\;|\tilde{\mathbf{x}}\rangle\langle\tilde{\mathbf{x}}|\;U^\dagger(\boldsymbol{\theta})\big) \;-\; b\Big). \tag{146}$$

Equivalently,

$$\tilde{y} = +1 \iff f(\tilde{\mathbf{x}}, \boldsymbol{\theta}) > b, \qquad \tilde{y} = -1 \iff f(\tilde{\mathbf{x}}, \boldsymbol{\theta}) < b. \tag{147}$$

The surface $f(\mathbf{x}, \boldsymbol{\theta}) = b$ defines a *hyperplane* in the quantum feature space, generalizing the classical linear classifier (Sec. IX B) to the Hilbert space of quantum states.

D. Quantum Convolutional Neural Networks

Quantum convolutional neural networks (QCNNs) extend the classical CNN architecture to the quantum domain. The framework consists of three stages applied sequentially:

1. **Data encoding.** Classical features are loaded into a quantum state $|\mathbf{x}\rangle$ using one of the encoding schemes from Sec. IX G.

2. **Convolutional layers.** Local parameterized unitaries act on neighboring qubit patches, analogous to classical convolutional filters.

3. **Pooling layers.** Qubits are measured or discarded, reducing the effective qubit number while preserving the most relevant information.

For n input qubits, the circuit depth scales as $O(\log n)$ and the number of parameters as $O(n)$, making QCNNs hardware-efficient. The overall prediction is

$$h(\boldsymbol{\theta}, |\mathbf{x}\rangle) \;=\; \langle \mathbf{x}|Z|\mathbf{x}\rangle \;\in\; [-1,+1], \tag{148}$$

and the loss (e.g., mean-squared error over m training samples) is

$$L(\boldsymbol{\theta}) \;=\; \frac{1}{m}\sum_{i=1}^{m}\big(y_i - h(\boldsymbol{\theta}, |\mathbf{x}_i\rangle)\big)^2. \tag{149}$$

1. *Parameterized quantum circuit examples*

A variety of convolutional circuit templates have been proposed. Common building blocks include single-qubit rotations $R_x(\theta)$, $R_y(\theta)$, $R_z(\theta)$ and entangling gates (e.g., CNOT, CZ). Figure 20 shows representative two-qubit convolutional unitaries.

Pooling is implemented by measuring a subset of qubits and conditioning subsequent rotations on the classical outcome, or by simply tracing out (reducing) qubits. Stacking multiple convolution–pooling layers yields a hierarchical feature extractor whose effective receptive field grows with depth.

E. Universal Approximation Theorem

A fundamental result by Schuld *et al.* [18] establishes that variational quantum models are universal function approximators.

[Universal Approximation Theorem for QML] Let $f : \Omega \to \mathbb{R}$ be a continuous function on a compact set $\Omega \subset \mathbb{R}^n$. A quantum model of the form

$$f(\mathbf{x}, \boldsymbol{\theta}) \;=\; \langle 0|\mathcal{U}^\dagger(\mathbf{x}, \boldsymbol{\theta})\, O\, \mathcal{U}(\mathbf{x}, \boldsymbol{\theta})|0\rangle\rangle\,, \tag{150}$$

where the parameterized unitary is a product of alternating data-encoding and trainable layers,

$$\mathcal{U}(\mathbf{x}, \boldsymbol{\theta}) \;=\; \prod_{\ell=1}^{L} U_{\mathrm{tr}}(\boldsymbol{\theta}_\ell)\, V_{\mathrm{emb}}(\mathbf{x}), \tag{151}$$

can approximate f to arbitrary accuracy provided the number of layers L is sufficiently large.

Proof sketch. The key idea is to show that the quantum model can represent any truncated Fourier series, and then invoke the density of Fourier series in $C(\Omega)$.

Step 1: Single-frequency encoding. Consider the angle encoding $V_{\mathrm{emb}}(\mathbf{x}) = \bigotimes_{j=1}^{n} R_y(x_j)$ applied to $|0\rangle^{\otimes n}$. Each qubit state is $R_y(x_j)\,|0\rangle \;=\; \cos(x_j/2)\,|0\rangle + \sin(x_j/2)\,|1\rangle$. The n-qubit product state is

$$V_{\mathrm{emb}}(\mathbf{x})\,|0\rangle^{\otimes n} = \sum_{\mathbf{z}\in\{0,1\}^n}\Big[\prod_{j=1}^{n}\cos^{1-z_j}\frac{x_j}{2}\,\sin^{z_j}\frac{x_j}{2}\Big]\,|\mathbf{z}\rangle\,. \tag{152}$$

Expanding the trigonometric products yields terms of the form $\prod_{j\in S}\sin(x_j/2)\prod_{j\notin S}\cos(x_j/2)$ for each subset $S \subseteq \{1,\dots,n\}$. Using $\sin(\alpha)\cos(\beta) = \frac{1}{2}[\sin(\alpha+\beta)+\sin(\alpha-\beta)]$ and $\cos(\alpha)\cos(\beta) = \frac{1}{2}[\cos(\alpha-\beta)+\cos(\alpha+\beta)]$, each such term is a finite linear combination of functions $e^{i\boldsymbol{\omega}\cdot\mathbf{x}}$ with $\boldsymbol{\omega} \in \{-1/2, 0, +1/2\}^n$.

Step 2: Entangling layers generate higher frequencies. A trainable layer $U_{\mathrm{tr}}(\boldsymbol{\theta})$ with entangling gates (e.g., CNOT, CZ) mixes the computational basis states. After L alternating layers, the expectation value $\langle 0|\mathcal{U}^\dagger(\mathbf{x},\boldsymbol{\theta})\, O\, \mathcal{U}(\mathbf{x},\boldsymbol{\theta})|0\rangle$ is a multivariate trigonometric polynomial in $\mathbf{x}$ whose frequencies are integer linear combinations of the base frequencies from Step 1. Increasing L enlarges the set of accessible frequencies.

Step 3: Density of Fourier series. For any continuous $f : \Omega \to \mathbb{R}$ and any $\varepsilon > 0$, the Weierstrass approximation theorem (via Fejér's theorem for Fourier series) guarantees the existence of a finite Fourier series $\tilde{f}(\mathbf{x}) = \sum_{\boldsymbol{\omega}\in F} c_{\boldsymbol{\omega}}\, e^{i\boldsymbol{\omega}\cdot\mathbf{x}}$ with $|F| < \infty$ such that $\sup_{\mathbf{x}\in\Omega}|f(\mathbf{x}) - \tilde{f}(\mathbf{x})| < \varepsilon$. By choosing L large enough and appropriate $\boldsymbol{\theta}$, the quantum model can represent each term $e^{i\boldsymbol{\omega}\cdot\mathbf{x}}$ for $\boldsymbol{\omega} \in F$, and hence approximate f within ε. □

F. Analytical Gradients for Variational Quantum Circuits

Training variational quantum models requires computing gradients of the cost function with respect to the circuit parameters. Consider a parameterized circuit

$$U(\boldsymbol{\theta}) \;=\; \prod_{k=1}^{K} U_k(\theta_k), \qquad U_k(\theta_k) = e^{-i\theta_k G_k/2}, \tag{153}$$

where each G_k is a Hermitian generator (e.g., a Pauli string with $G_k^2 = I$). Acting on an input state—which in general is the mixed state ρ_0 (for a pure input $\rho_0 = |0\rangle\langle 0|^{\otimes n}$)—the circuit prepares

$$\rho(\boldsymbol{\theta}) \;=\; U(\boldsymbol{\theta})\,\rho_0\,U^\dagger(\boldsymbol{\theta}), \tag{154}$$

and the cost function is the expectation value of a Hermitian observable $M = M^\dagger$,

$$C(\boldsymbol{\theta}) \;=\; \mathrm{Tr}\big(M\,\rho(\boldsymbol{\theta})\big) \;=\; \langle 0|U^\dagger(\boldsymbol{\theta})\,M\,U(\boldsymbol{\theta})|0\rangle\,. \tag{155}$$

The density-matrix form on the left of (155) is the most convenient starting point for differentiation, because it makes the *linearity* of the cost in the state manifest and treats pure and mixed inputs on the same footing.

1. *Density-matrix form of the gradient*

To differentiate (155) with respect to a single parameter θ_k, split the circuit around the k-th gate into a *pre-k* segment (all gates before index k) and a *post-k* segment (all gates after index k):

$$U(\boldsymbol{\theta}) = U_{\mathrm{post},k}\,U_k(\theta_k)\,U_{\mathrm{pre},k}, \qquad U_{\mathrm{pre},k} = \prod_{j<k} U_j, \quad U_{\mathrm{post},k} = \prod_{j>k} U_j, \tag{156}$$

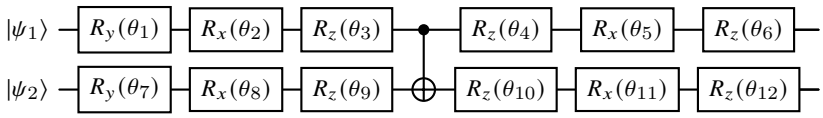


FIG. 20. Representative two-qubit convolutional layer: alternating single-qubit rotations (R_y, R_x, R_z) and entangling CNOT gates. The 12 continuous parameters $\{\theta_1, \ldots, \theta_{12}\}$ are optimized during training. Various templates in the literature differ in the ordering and choice of rotation axes, but all follow this general pattern of local rotations interleaved with entanglers.

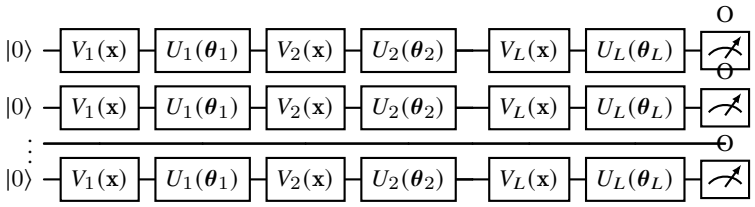


FIG. 21. Ansatz for the universal approximation theorem: alternating data-encoding unitaries $V_\ell(\mathbf{x})$ and trainable unitaries $U_\ell(\boldsymbol{\theta}_\ell)$ for $\ell = 1, \ldots, L$. The observable O is measured on the final state. Increasing L expands the expressivity of the model, enabling approximation of arbitrary continuous functions [18].

and define the state prepared just *before* the k-th gate together with the state prepared just *after* it,

$$\rho_{\mathrm{pre},k} \coloneqq U_{\mathrm{pre},k}\,\rho_0\,U_{\mathrm{pre},k}^\dagger, \qquad \rho_k \coloneqq U_k(\theta_k)\,\rho_{\mathrm{pre},k}\,U_k^\dagger(\theta_k). \tag{157}$$

Only $U_k(\theta_k)$ depends on θ_k, and because G_k commutes with $U_k(\theta_k)$,

$$\frac{\partial U_k}{\partial \theta_k} = -\frac{i}{2}\,G_k\,U_k = -\frac{i}{2}\,U_k\,G_k. \tag{158}$$

Differentiating ρ_k therefore produces a commutator with the generator,

$$\begin{aligned}\frac{\partial \rho_k}{\partial \theta_k} &= -\frac{i}{2}G_k\,U_k\rho_{\mathrm{pre},k}U_k^\dagger + U_k\rho_{\mathrm{pre},k}U_k^\dagger\,\frac{i}{2}G_k \\ &= -\frac{i}{2}\,[\,G_k,\,\rho_k\,].\end{aligned} \tag{159}$$

Propagating this through the remaining post-k gates $U_{\mathrm{post},k}$ (which are independent of θ_k) and tracing against M gives the density-matrix gradient

$$\begin{aligned}\frac{\partial C}{\partial \theta_k} &= \mathrm{Tr}\Big(M\,U_{\mathrm{post},k}\,\frac{\partial \rho_k}{\partial \theta_k}\,U_{\mathrm{post},k}^\dagger\Big) \\ &= -\frac{i}{2}\,\mathrm{Tr}\big(M_k\,[\,G_k,\,\rho_k\,]\big) = -\frac{i}{2}\,\mathrm{Tr}\big([\,M_k,\,G_k\,]\,\rho_k\big),\end{aligned} \tag{160}$$

where in the last step we used the cyclicity of the trace and defined the *back-propagated observable*

$$M_k \;\coloneqq\; U_{\mathrm{post},k}^\dagger\,M\,U_{\mathrm{post},k}. \tag{161}$$

Equation (160) expresses the gradient as the expectation value of the Hermitian commutator-observable $-\frac{i}{2}[M_k, G_k]$ (note $-\frac{i}{2}[M_k, G_k]$ is Hermitian because M_k and G_k are) in the intermediate state ρ_k. This is the exact, hardware-independent form of the gradient; the parameter-shift rule below is simply a way of measuring it with the *same* ansatz circuit rather than a modified one.

2. *Schrödinger- and Heisenberg-picture interpretations*

The single result (160) can be read in two dual ways, exactly mirroring the Schrödinger and Heisenberg pictures of quantum dynamics, where here the "time" is the variational parameter θ_k.

a. Heisenberg picture (back-propagate the observable). Keep the intermediate state ρ_k fixed and push the observable backward through the later gates. Writing (160) as

$$\frac{\partial C}{\partial \theta_k} = -\frac{i}{2}\,\mathrm{Tr}\big([M_k, G_k]\,\rho_k\big) = \Big\langle -\tfrac{i}{2}[M_k, G_k]\Big\rangle_{\rho_k}, \tag{162}$$

the gradient is the expectation, in the state ρ_k reached right after gate k, of the commutator between the generator G_k and the Heisenberg-evolved observable $M_k = U_{\mathrm{post},k}^\dagger M\,U_{\mathrm{post},k}$. All the θ_k-dependence has been absorbed into the local generator, while the effect of the downstream circuit is encoded entirely in how it dresses the measurement operator.

b. Schrödinger picture (forward-propagate the state). Alternatively, keep the physical observable M fixed at the end of the circuit and evolve the *differentiated state* forward. Defining the generator transported to the circuit output,

$$\widetilde{G}_k \;\coloneqq\; U_{\mathrm{post},k}\,G_k\,U_{\mathrm{post},k}^\dagger, \tag{163}$$

and using $U_{\mathrm{post},k}\,[G_k, \rho_k]\,U_{\mathrm{post},k}^\dagger = [\widetilde{G}_k, \rho(\boldsymbol{\theta})]$, equation (160) becomes

$$\frac{\partial C}{\partial \theta_k} = -\frac{i}{2}\,\mathrm{Tr}\big(M\,[\widetilde{G}_k, \rho(\boldsymbol{\theta})]\big) = \Big\langle -\tfrac{i}{2}[\widetilde{G}_k, M]\Big\rangle_{\rho(\boldsymbol{\theta})}. \tag{164}$$

Here the full output state $\rho(\boldsymbol{\theta})$ is retained and the derivative acts as an infinitesimal generator $-\frac{i}{2}[\widetilde{G}_k, \cdot\,]$ inserted into the output state, with $\widetilde{G}_k$ the generator carried forward to the end of the circuit. The two pictures are related by the unitary similarity transformation $U_{\mathrm{post},k}$ and give identical gradients, just as expectation values agree in the Schrödinger and Heisenberg pictures of ordinary time evolution.

3. Parameter-shift rule

The commutator gradient (160) cannot be measured directly, because $-\frac{i}{2}[M_k, G_k]$ is generally not an observable that the hardware can read out. The *parameter-shift rule* [19, 20] resolves this by rewriting the commutator as a *difference of two physical expectation values* of the original observable, evaluated on the same ansatz with the single parameter θ_k shifted by $\pm\pi/2$:

$$\frac{\partial C}{\partial \theta_k} = \frac{1}{2}\Big[C\big(\boldsymbol{\theta} + \tfrac{\pi}{2}\mathbf{e}_k\big) - C\big(\boldsymbol{\theta} - \tfrac{\pi}{2}\mathbf{e}_k\big)\Big], \tag{165}$$

where $\mathbf{e}_k$ is the unit vector in the k-th direction. This formula is *exact* (not a finite-difference approximation) and requires only two evaluations of the unmodified circuit per parameter, making it ideal for hardware implementation.

Figure 22 summarizes the hardware procedure: for each trainable parameter θ_k, run the *same* $\rho_0 \to U \to M$ circuit twice with the single gate angle shifted by $\pm\pi/2$, then take their difference. All other parameters and the measurement observable M remain unchanged.

Proof. The key is a "commutator-to-shift" identity for the generator with $G_k^2 = I$. From (153),

$$U_k(s) = e^{-isG_k/2} = \cos\frac{s}{2}\, I - i \sin\frac{s}{2}\, G_k, \tag{166}$$

which uses $G_k^2 = I$ so that the eigenvalues of G_k are ± 1. For the choice $s = \pm\pi/2$ this gives the "$\pi/2$ pulses" $U_k(\pm\pi/2) = \frac{1}{\sqrt{2}}(I \mp iG_k)$. Conjugating an arbitrary state ρ and using $G_k^\dagger = G_k$, $G_k^2 = I$,

$$\begin{aligned} U_k(\pm\tfrac{\pi}{2})\,\rho\,U_k^\dagger(\pm\tfrac{\pi}{2}) &= \tfrac{1}{2}(I \mp iG_k)\,\rho\,(I \pm iG_k) \\ &= \tfrac{1}{2}\big(\rho + G_k\rho G_k\big) \;\mp\; \tfrac{i}{2}\,[\,G_k,\,\rho\,]. \end{aligned} \tag{167}$$

Subtracting the two cases, the symmetric $\frac{1}{2}(\rho + G_k\rho G_k)$ pieces cancel and the commutator survives:

$$-\frac{i}{2}\,[\,G_k,\,\rho\,] = \frac{1}{2}\Big[U_k(\tfrac{\pi}{2})\,\rho\,U_k^\dagger(\tfrac{\pi}{2}) - U_k(-\tfrac{\pi}{2})\,\rho\,U_k^\dagger(-\tfrac{\pi}{2})\Big]. \tag{168}$$

Apply (168) with $\rho = \rho_k$. Because $U_k(\pm\pi/2)\rho_k\,U_k^\dagger(\pm\pi/2) = U_k(\theta_k \pm \frac{\pi}{2})\,\rho_{\mathrm{pre},k}\,U_k^\dagger(\theta_k \pm \frac{\pi}{2})$, the right-hand side is exactly the state ρ_k with the parameter shifted to $\theta_k \pm \pi/2$. Substituting into the density-matrix gradient (160),

$$\begin{aligned} \frac{\partial C}{\partial \theta_k} &= \mathrm{Tr}\Big(M_k\,\big(-\tfrac{i}{2}[G_k, \rho_k]\big)\Big) \\ &= \frac{1}{2}\,\mathrm{Tr}\Big(M_k\,U_k(\theta_k{+}\tfrac{\pi}{2})\rho_{\mathrm{pre},k}U_k^\dagger(\theta_k{+}\tfrac{\pi}{2})\Big) \\ &\quad - \frac{1}{2}\,\mathrm{Tr}\Big(M_k\,U_k(\theta_k{-}\tfrac{\pi}{2})\rho_{\mathrm{pre},k}U_k^\dagger(\theta_k{-}\tfrac{\pi}{2})\Big) \\ &= \frac{1}{2}\Big[C\big(\boldsymbol{\theta} + \tfrac{\pi}{2}\mathbf{e}_k\big) - C\big(\boldsymbol{\theta} - \tfrac{\pi}{2}\mathbf{e}_k\big)\Big], \end{aligned} \tag{169}$$

where the last line re-attaches the back-propagated observable $M_k = U_{\mathrm{post},k}^\dagger M U_{\mathrm{post},k}$ to its full circuit form via $\mathrm{Tr}(M_k\,\cdot) = \mathrm{Tr}(M\,U_{\mathrm{post},k}\,\cdot\,U_{\mathrm{post},k}^\dagger)$, recognizing each term as the cost evaluated at the shifted parameter. This proves (165). □

A useful cross-check is to compute the cost explicitly as a function of θ_k. Inserting (166) into $C = \mathrm{Tr}(M_k\rho_k)$ and abbreviating $a = \mathrm{Tr}(M_k\rho_{\mathrm{pre},k})$, $b = \mathrm{Tr}(G_kM_kG_k\rho_{\mathrm{pre},k})$, $d = \mathrm{Tr}([M_k, G_k]\rho_{\mathrm{pre},k})$ yields the single-parameter dependence

$$C(\theta_k) = \frac{a+b}{2} + \frac{a-b}{2}\cos\theta_k - \frac{i}{2}\,d\,\sin\theta_k, \tag{170}$$

i.e. a pure sinusoid of period 2π in θ_k. Its exact derivative $\partial_{\theta_k}C = -\frac{a-b}{2}\sin\theta_k - \frac{i}{2}d\cos\theta_k$ coincides with the symmetric difference $\frac{1}{2}[C(\theta_k + \frac{\pi}{2}) - C(\theta_k - \frac{\pi}{2})]$, confirming (165). (A shift of $\pi/4$ would instead return $1/\sqrt{2}$ times the true gradient, which is why the shift must be $\pi/2$ for the generator convention $U_k = e^{-i\theta_k G_k/2}$.)

When the generator does not satisfy $G_k^2 = I$—so that G_k has more than two distinct eigenvalues—the cost $C(\theta_k)$ becomes a finite Fourier series with several frequencies, and a *generalized parameter-shift rule* reconstructs the gradient from a correspondingly larger number of shifted evaluations [21]; the essential feature—exact gradients from a finite set of circuit evaluations at shifted parameters—persists.

G. Quantum State Discrimination

Quantum state discrimination provides the theoretical foundation for understanding the fundamental limits of quantum classification. We consider the simplest case: distinguishing between two known quantum states.

1. Two-state discrimination problem

[Binary quantum state discrimination] An unknown quantum state is provided: it is either ρ_0 with prior probability p_0 or ρ_1 with prior probability $p_1 = 1 - p_0$. Determine the measurement that maximizes the success probability of correctly identifying the state.

A quantum measurement is described by a Positive Operator-Valued Measure (POVM) $\{M_k\}$ satisfying

$$\sum_k M_k^\dagger M_k = I, \qquad E_k := M_k^\dagger M_k \geq 0. \tag{171}$$

For binary discrimination, the POVM has two elements $\{E_0, E_1\}$ with $E_0 + E_1 = I$. The success probability is

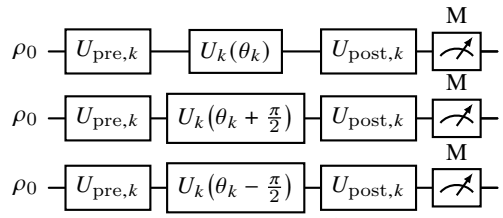
$$\begin{aligned} p_{\mathrm{s}} &= p_0\ \mathrm{Tr}(E_0\rho_0) + p_1\ \mathrm{Tr}(E_1\rho_1) \\ &= p_0\ \mathrm{Tr}\big((I - E_1)\rho_0\big) + p_1\ \mathrm{Tr}(E_1\rho_1) \\ &= p_0 - \mathrm{Tr}\big((p_0\rho_0 - p_1\rho_1)E_1\big). \end{aligned} \tag{172}$$

Maximizing p_{s} is equivalent to *minimizing* $\mathrm{Tr}\big((p_0\rho_0 - p_1\rho_1)E_1\big)$. Let

$$\Gamma := p_0\rho_0 - p_1\rho_1 = \sum_j \lambda_j\,|\eta_j\rangle\langle\eta_j| \tag{173}$$

be the spectral decomposition of the Hermitian operator Γ. The optimal E_1 projects onto the subspace spanned by eigenvectors with *negative* eigenvalues:

$$E_1^\star = \sum_{j:\,\lambda_j<0} |\eta_j\rangle\langle\eta_j|, \qquad E_0^\star = I - E_1^\star. \tag{174}$$



(a) reference $C(\boldsymbol{\theta})$ (b) shift $+\pi/2 \to C^+$ (c) shift $-\pi/2 \to C^-$

$$\frac{\partial C}{\partial \theta_k} = \frac{1}{2}\big(C^+ - C^-\big) = \frac{1}{2}\Big[C\big(\boldsymbol{\theta} + \tfrac{\pi}{2}\mathbf{e}_k\big) - C\big(\boldsymbol{\theta} - \tfrac{\pi}{2}\mathbf{e}_k\big)\Big], \qquad C = \mathrm{Tr}\big(M\,U(\boldsymbol{\theta})\rho_0 U^\dagger(\boldsymbol{\theta})\big).$$

FIG. 22. Parameter-shift rule in circuit form. Each row is the same variational ansatz $\rho(\boldsymbol{\theta}) = U(\boldsymbol{\theta})\rho_0 U^\dagger(\boldsymbol{\theta})$ followed by a measurement of the observable M. The circuit is split at the k-th gate into fixed blocks $U_{\mathrm{pre},k}$ and $U_{\mathrm{post},k}$; only the parameterized gate U_k changes between rows (a)–(c). Shifting θ_k by $\pm\pi/2$ and combining the outcomes C^+ and C^- yields the exact gradient (165) without modifying M or any other parameter.

The measurement $\{E_0^\star, E_1^\star\}$ is the *Helstrom measurement*.

Proof of optimality. We must minimize $\mathrm{Tr}(\Gamma E_1)$ over all POVM elements $0 \le E_1 \le I$. Expanding in the eigenbasis of Γ,

$$\mathrm{Tr}(\Gamma E_1) = \sum_j \lambda_j \langle \eta_j | E_1 | \eta_j \rangle = \sum_j \lambda_j\, p_j, \qquad p_j \coloneqq \langle \eta_j | E_1 | \eta_j \rangle \in [0,1], \tag{175}$$

where $0 \le p_j \le 1$ because $0 \le E_1 \le I$, and $\sum_j p_j = \mathrm{Tr}(E_1)$ is unconstrained (any value in $[0, \mathrm{Tr}(I)]$ is admissible since $E_0 = I - E_1 \ge 0$ only requires $E_1 \le I$).

The objective $\sum_j \lambda_j p_j$ is linear in each p_j with $0 \le p_j \le 1$. To minimize it, we set $p_j = 1$ (maximal weight) when $\lambda_j < 0$ and $p_j = 0$ when $\lambda_j > 0$. For $\lambda_j = 0$ the choice of p_j is immaterial. This is achieved precisely by

$$E_1^\star = \sum_{j:\,\lambda_j < 0} |\eta_j\rangle \langle \eta_j|, \tag{176}$$

which satisfies $0 \le E_1^\star \le I$ and yields the minimum value $\sum_{j:\,\lambda_j<0} \lambda_j$. □

2. *Success probability and trace distance*

Substituting the optimal POVM into (172),

$$\begin{aligned} p_{\mathrm{s}}^\star &= p_0 - \sum_j \lambda_j \,\mathrm{Tr}\big(|\eta_j\rangle\langle\eta_j|\, E_1^\star\big) \\ &= p_0 + \sum_{j:\,\lambda_j<0} |\lambda_j|. \end{aligned} \tag{177}$$

Equivalently, by symmetry,

$$p_{\mathrm{s}}^\star = p_1 + \sum_{j:\,\lambda_j \ge 0} \lambda_j. \tag{178}$$

Adding these two expressions and dividing by 2,

$$p_{\mathrm{s}}^\star = \frac{1}{2}\Big(1 + \sum_j |\lambda_j|\Big) = \frac{1}{2}\Big(1 + \|\Gamma\|_1\Big), \tag{179}$$

where $\|\Gamma\|_1 = \mathrm{Tr}\,|\Gamma| = \sum_j |\lambda_j|$ is the *trace norm*. The *error probability* is therefore

$$p_{\mathrm{e}}^\star = 1 - p_{\mathrm{s}}^\star = \frac{1}{2}\Big(1 - \|\Gamma\|_1\Big) = \frac{1}{2} - \frac{1}{2}\,\mathrm{Tr}\,|p_0\rho_0 - p_1\rho_1|. \tag{180}$$

Derivation of the trace-norm form. By definition, $|\Gamma| = \sqrt{\Gamma^\dagger\Gamma} = \sqrt{\Gamma^2}$ (since Γ is Hermitian). In the eigenbasis, $|\Gamma| = \sum_j |\lambda_j|\,|\eta_j\rangle\langle\eta_j|$, so $\mathrm{Tr}\,|\Gamma| = \sum_j |\lambda_j|$. Substituting $\Gamma = p_0\rho_0 - p_1\rho_1$ into $p_{\mathrm{s}}^\star = \frac{1}{2}(1 + \|\Gamma\|_1)$ gives

$$p_{\mathrm{s}}^\star = \frac{1}{2} + \frac{1}{2}\,\mathrm{Tr}\,|p_0\rho_0 - p_1\rho_1| = \frac{1}{2} + D_{\mathrm{tr}}(p_0\rho_0,\, p_1\rho_1), \tag{181}$$

where the last equality uses the definition $D_{\mathrm{tr}}(\rho,\sigma) = \frac{1}{2}\,\mathrm{Tr}\,|\rho - \sigma|$. The error probability follows immediately: $p_{\mathrm{e}}^\star = 1 - p_{\mathrm{s}}^\star = \frac{1}{2} - D_{\mathrm{tr}}(p_0\rho_0,\, p_1\rho_1)$. □

The *trace distance* between two states is defined as

$$D_{\mathrm{tr}}(\rho,\sigma) \;\coloneqq\; \frac{1}{2}\,\mathrm{Tr}\,|\rho - \sigma|. \tag{182}$$

In terms of the trace distance, the optimal success and error probabilities become

$$p_{\mathrm{s}}^\star = \frac{1}{2} + D_{\mathrm{tr}}(p_0\rho_0,\, p_1\rho_1), \qquad p_{\mathrm{e}}^\star = \frac{1}{2} - D_{\mathrm{tr}}(p_0\rho_0,\, p_1\rho_1). \tag{183}$$

The trace distance thus quantifies the fundamental distinguishability of the two quantum states: the larger the trace distance, the lower the minimum error probability.

H. Variational Quantum Classifier from State Discrimination

The connection between VQCs and quantum state discrimination provides a powerful interpretive framework. Consider a binary VQC with training data

$$\mathcal{S} = \big\{\mathbf{x}_i^{(+)}, +1\big\}_{i=1}^{M_+} \cup \big\{\mathbf{x}_j^{(-)}, -1\big\}_{j=1}^{M_-}. \tag{184}$$

Define the *average* quantum states for each class after the parameterized circuit:

$$\bar{\rho}_+ \coloneqq \frac{1}{M_+}\sum_{i=1}^{M_+} U(\boldsymbol{\theta})\ |\mathbf{x}_i^{(+)}\rangle\langle\mathbf{x}_i^{(+)}|\ U^\dagger(\boldsymbol{\theta}), \tag{185}$$

$$\bar{\rho}_- \coloneqq \frac{1}{M_-}\sum_{j=1}^{M_-} U(\boldsymbol{\theta})\ |\mathbf{x}_j^{(-)}\rangle\langle\mathbf{x}_j^{(-)}|\ U^\dagger(\boldsymbol{\theta}). \tag{186}$$

The VQC loss function (probability of measuring the incorrect label) is

$$L(\mathcal{S},\boldsymbol{\theta}) = \frac{1}{M_+ + M_-}\left[\sum_{i=1}^{M_+}\mathrm{Tr}\big(E_-\,U(\boldsymbol{\theta})\,|\mathbf{x}_i^{(+)}\rangle\langle\mathbf{x}_i^{(+)}|\,U^\dagger(\boldsymbol{\theta})\big) + \sum_{j=1}^{M_-}\mathrm{Tr}\big(E_+\,U(\boldsymbol{\theta})\,|\mathbf{x}_j^{(-)}\rangle\langle\mathbf{x}_j^{(-)}|\,U^\dagger(\boldsymbol{\theta})\big)\right], \tag{187}$$

where $\{E_+, E_-\}$ is the POVM implementing the classification measurement. In terms of the average states,

$$L(\mathcal{S},\boldsymbol{\theta}) = p_+\ \mathrm{Tr}(E_-\,\tilde{\rho}_+) + p_-\ \mathrm{Tr}(E_+\,\tilde{\rho}_-), \tag{188}$$

where $p_\pm = M_\pm/(M_+ + M_-)$ and $\tilde{\rho}_\pm = U^\dagger(\boldsymbol{\theta})\,E_\pm\,U(\boldsymbol{\theta})$ are the effective states in the measurement basis.

From the Helstrom bound (Sec. X G 1), the loss is *lower-bounded* by

$$L(\mathcal{S},\boldsymbol{\theta})\ \geq\ \frac{1}{2} - D_{\mathrm{tr}}(p_+\tilde{\rho}_+,\ p_-\tilde{\rho}_-). \tag{189}$$

The minimum is achieved when the measurement $\{U^\dagger(\boldsymbol{\theta})E_y U(\boldsymbol{\theta})\}_{y=0,1}$ coincides with the Helstrom measurement for the pair $(p_+\tilde{\rho}_+,\ p_-\tilde{\rho}_-)$.

Derivation of the Helstrom lower bound. The loss (188) has the same mathematical form as the error probability in binary state discrimination with prior probabilities p_+, p_- and states $\tilde{\rho}_+, \tilde{\rho}_-$. Explicitly,

$$L(\mathcal{S},\boldsymbol{\theta}) = p_-\ \mathrm{Tr}(E_+\,\tilde{\rho}_-) + p_+\ \mathrm{Tr}(E_-\,\tilde{\rho}_+) = p_-\ \mathrm{Tr}(E_+\,\tilde{\rho}_-) + p_+\ \big(1 - \mathrm{Tr}(E_+\,\tilde{\rho}_+)\big). \tag{190}$$

Rearranging,

$$L(\mathcal{S},\boldsymbol{\theta}) = p_+ + \mathrm{Tr}\big(E_+\,(p_-\tilde{\rho}_- - p_+\tilde{\rho}_+)\big). \tag{191}$$

This is minimized when E_+ projects onto the negative eigenspace of $p_-\tilde{\rho}_- - p_+\tilde{\rho}_+$, i.e., the positive eigenspace of $p_+\tilde{\rho}_+ - p_-\tilde{\rho}_-$. The minimum value is

$$L_{\min} = p_+ - \sum_{j:\,\lambda_j^{(+)}>0}\lambda_j^{(+)} = \frac{1}{2} - D_{\mathrm{tr}}(p_+\tilde{\rho}_+,\ p_-\tilde{\rho}_-), \tag{192}$$

where $\lambda_j^{(+)}$ are the eigenvalues of $p_+\tilde{\rho}_+ - p_-\tilde{\rho}_-$, and the last equality follows from the trace-norm identity proved above. □

This result has an important implication: *no variational circuit can reduce the classification error below the Helstrom limit determined by the trace distance between the class-averaged quantum states*. The trainable circuit $U(\boldsymbol{\theta})$ can at best rotate the states to maximize this trace distance, but the fundamental bound is set by the data encoding itself. This provides a quantum analogue of the Bayes-error bound in classical statistical classification.

I. Summary

This chapter developed the variational quantum machine learning framework from the ground up:

- **VQAs** minimize a cost $L(\boldsymbol{\theta}) = \mathrm{Tr}(HU(\boldsymbol{\theta})\,|\psi\rangle\,\langle\theta|\,U^\dagger(\boldsymbol{\theta}))$ by alternating quantum expectation estimation with classical parameter updates. The variational principle guarantees that the minimum cost bounds the ground-state energy.
- **QML models are linear** in the 4^n-dimensional Pauli feature space: $f(\mathbf{x},\boldsymbol{\theta}) = 2^n\sum_i \alpha_i(\boldsymbol{\theta})\beta_i(\mathbf{x})$.
- **VQCs** implement a hyperplane decision boundary in quantum feature space via $\tilde{y} = \mathrm{sign}(\mathrm{Tr}(MU(\boldsymbol{\theta})\,|\tilde{\mathbf{x}}\rangle\,\langle\tilde{\mathbf{x}}|\,U^\dagger(\boldsymbol{\theta})) - b)$.
- **QCNNs** generalize classical CNNs with convolutional layers (local parameterized unitaries), pooling (measurement / qubit reduction), and $O(\log n)$ depth scaling.
- **Universal approximation**: alternating encoding and trainable layers can approximate any continuous function to arbitrary accuracy.
- **Analytical gradients**: in density-matrix form the derivative of the cost is the commutator expectation $\partial_{\theta_k}C = -\frac{i}{2}\mathrm{Tr}([M_k, G_k]\rho_k)$, which admits dual Heisenberg- (back-propagated observable $M_k = U_{\mathrm{post},k}^\dagger M U_{\mathrm{post},k}$) and Schrödinger-picture (forward-transported generator $\widetilde{G}_k = U_{\mathrm{post},k}G_kU_{\mathrm{post},k}^\dagger$) readings. The **parameter-shift rule** turns this commutator into a difference of two physical evaluations,

$\partial_{\theta_k} C = \frac{1}{2}[C(\boldsymbol{\theta} + \frac{\pi}{2}\mathbf{e}_k) - C(\boldsymbol{\theta} - \frac{\pi}{2}\mathbf{e}_k)]$, exact for Pauli generators ($G_k^2 = I$).

- **Quantum state discrimination**: the Helstrom measurement achieves the minimum error probability $p_e^\star = \frac{1}{2} - D_{\mathrm{tr}}(p_0\rho_0, p_1\rho_1)$, which lower-bounds the VQC loss.

These results establish both the capabilities and the fundamental limits of variational quantum classifiers, and connect the NISQ-era algorithmic toolkit to the broader landscape of quantum information theory.

XI. QUANTUM APPROACHES TO DISCRETE OPTIMIZATION

Discrete (combinatorial) optimization problems ask for the configuration that minimizes (or maximizes) some objective function over a finite search space. Because the search space grows exponentially with the problem size, these problems are intractable for classical computers in the worst case. Quantum computing offers several paradigms for tackling them: *adiabatic quantum computing* (AQC), which leverages continuous Hamiltonian evolution, and the *quantum approximate optimization algorithm* (QAOA), which discretizes the adiabatic idea into a gate-based variational circuit. We develop both approaches and illustrate them with the canonical QUBO (Quadratic Unconstrained Binary Optimization) framework.

A. Discrete Optimization and QUBO

Consider a function $f : \{0,1\}^n \to \mathbb{R}$. The goal is to find a bit string $\mathbf{z}^\star$ that globally minimizes f:

$$\mathbf{z}^\star \;=\; \arg\min_{\mathbf{z}\in\{0,1\}^n} f(\mathbf{z}). \tag{193}$$

a. Quadratic Unconstrained Binary Optimization. A prototypical class of discrete problems is QUBO, where the objective is a quadratic form in binary variables:

$$f(\mathbf{z}) \;=\; \sum_i b_i z_i \;+\; \sum_{i<j} w_{ij} z_i z_j, \qquad b_i \in \mathbb{R},\;\; w_{ij} \in \mathbb{R}. \tag{194}$$

Because $z_i \in \{0,1\}$ implies $z_i^2 = z_i$, this can be rewritten without the linear term as

$$f(\mathbf{z}) \;=\; \sum_{i,j} Q_{ij}\, z_i z_j, \qquad Q_{ij} \in \mathbb{R}, \tag{195}$$

a compact quadratic expression.

b. Ising Hamiltonian. Alternatively, mapping $z_i \in \{0,1\}$ to spin variables $s_i \in \{-1,+1\}$ via $z_i = (1 - s_i)/2$ gives the *Ising Hamiltonian*:

$$H_{\mathrm{Ising}} \;=\; \sum_i h_i\, s_i \;+\; \sum_{i<j} J_{ij}\, s_i s_j, \tag{196}$$

which is the standard form for quantum annealing hardware.

B. Famous Example: the Knapsack Problem

The *knapsack problem* illustrates the difficulty introduced by discreteness. Given items with values v_i and weights w_i, and a capacity W, we wish to

$$\max_{\mathbf{z}\in\{0,1\}^n} \quad \sum_{i=1}^n v_i z_i \tag{197}$$

$$\text{s.t.} \quad \sum_{i=1}^n w_i z_i \le W. \tag{198}$$

The continuous relaxation (allowing $z_i \in [0,1]$) is easily solved by greedy methods, but the binary constraint makes the problem NP-hard.

1. Handling constraints via the penalty method

The penalty method converts constrained problems into unconstrained ones by adding a penalty term to the objective.

a. Equality constraint. For $\min_{\mathbf{z}} f(\mathbf{z})$ subject to $h(\mathbf{z}) = C$, the penalized objective is

$$f'(\mathbf{z}) \;=\; f(\mathbf{z}) \;+\; \lambda\,\big(h(\mathbf{z}) - C\big)^2, \tag{199}$$

where $\lambda > 0$ is a sufficiently large penalty parameter. When $h(\mathbf{z}) \neq C$, the penalty drives the cost up, so the minimizer satisfies the constraint.

b. Inequality constraint. For $g(\mathbf{z}) \le W$, we introduce a *slack variable* $S \ge 0$ satisfying $g(\mathbf{z}) + S = W$. Representing S in binary with K qubits,

$$S \;=\; \sum_{k=0}^{K-1} 2^k x_k, \qquad x_k \in \{0,1\}, \tag{200}$$

the penalized problem becomes

$$\min_{\mathbf{z},\mathbf{x}} f(\mathbf{z}) \;+\; \lambda\left(g(\mathbf{z}) + \sum_{k=0}^{K-1} 2^k x_k - W\right)^2, \tag{201}$$

where the number of slack qubits K must satisfy $2^K \ge \lceil \max_{\mathbf{z}} g(\mathbf{z}) \rceil - W$.

c. Knapsack as QUBO. Applying the penalty method to the knapsack problem yields

$$\min_{\mathbf{z},\mathbf{x}} \; -\sum_{i=1}^n v_i z_i \;+\; \lambda\left(\sum_{i=1}^n w_i z_i + \sum_{k=0}^{K-1} 2^k x_k - W\right)^2. \tag{202}$$

The quadratic penalty term naturally fits the QUBO form (194).

C. From Optimization to Quantum Hamiltonians

Every discrete optimization problem can be mapped to a quantum algorithm by defining a *problem Hamiltonian*:

$$H_P \;=\; \sum_{\mathbf{z}\in\{0,1\}^n} f(\mathbf{z})\;|\mathbf{z}\rangle\,\langle\mathbf{z}|\,. \tag{203}$$

Since H_P is diagonal in the computational basis, its ground state is the bit string $\mathbf{z}^\star$ that minimizes f. For QUBO, we replace classical variables z_i with quantum spin operators $\frac{1}{2}(I - Z_i)$ (or equivalently, identify z_i with the eigenvalue of $\frac{I-Z_i}{2}$ in state $|z_i\rangle$), giving

$$\begin{aligned} H_P \;&=\; \sum_{i,j} Q_{ij}\,\frac{I-Z_i}{2}\,\frac{I-Z_j}{2} \\ &=\; \sum_{i,j} Q_{ij}\,Z_iZ_j \;+\; \text{local terms} \;+\; \text{constant}. \end{aligned} \tag{204}$$

Preparing the ground state of H_P therefore solves the optimization problem.

D. Adiabatic Quantum Computing

1. The quantum adiabatic theorem

Suppose we have a time-varying Hamiltonian $H(t)$ with $H(0) = H_I$ (initial) and $H(T) = H_F$ (final), where $[H_I, H_F] \neq 0$. The *quantum adiabatic theorem* states:

[Adiabatic Theorem] If the system is initially in the ground state of H_I, and the Hamiltonian evolves sufficiently slowly from $t = 0$ to $t = T$, the state remains in the instantaneous ground state throughout the evolution. At $t = T$, the state is the ground state of H_F.

---

a. Mathematical formulation. Let

$$H(t) \;=\; \big(1-\ell(t)\big)\,H_I \;+\; \ell(t)\,H_F, \tag{205}$$

where $\ell(t)$ is a smooth, monotonic function with $\ell(0) = 0$ and $\ell(T) = 1$. The time evolution satisfies the Schrödinger equation

$$i\hbar\,\frac{d}{dt}\,|\psi(t)\rangle \;=\; H(t)\,|\psi(t)\rangle\,, \tag{206}$$

and $H(t)\,|E_j(t)\rangle = E_j(t)\,|E_j(t)\rangle$ with $E_0(t) \le E_1(t) \le \cdots$ (ordered eigenvalues).

Define the *minimum energy gap*

$$g_{\min} \;:=\; \min_{t\in[0,T]}\;\min_{j\neq 0}\,\big(E_j(t) - E_0(t)\big). \tag{207}$$

The adiabatic theorem implies that if

$$T \;\gg\; O\big(g_{\min}^{-2}\big)\,, \tag{208}$$

then

$$|\langle E_0(T)|\psi(T)\rangle| \;\approx\; 1, \tag{209}$$

i.e., the final state has high overlap with the ground state of H_F.

b. Derivation of the Adiabatic Condition. To derive the adiabatic condition quantitatively, we expand the system's state vector $|\psi(t)\rangle$ in the instantaneous eigenbasis $\{|E_n(t)\rangle\}$ of the Hamiltonian $H(t)$, incorporating the dynamical phase:

$$|\psi(t)\rangle \;=\; \sum_n c_n(t)e^{-\frac{i}{\hbar}\theta_n(t)}\,|E_n(t)\rangle\,, \tag{210}$$

where $\theta_n(t) = \int_0^t E_n(t')\,dt'$. Differentiating this expansion with respect to time t and substituting it into the Schrödinger equation $i\hbar\frac{d}{dt}\,|\psi(t)\rangle = H(t)\,|\psi(t)\rangle$, we obtain:

$$\begin{aligned} i\hbar\sum_n\left[\dot{c}_n\,|E_n\rangle - \frac{i}{\hbar}E_nc_n\,|E_n\rangle + c_n\,|\dot{E}_n\rangle\right]e^{-\frac{i}{\hbar}\theta_n(t)} \\ =\; \sum_n c_nE_n\,|E_n\rangle\,e^{-\frac{i}{\hbar}\theta_n(t)}. \end{aligned} \tag{211}$$

The energy terms on both sides cancel, leaving:

$$\sum_n\left[\dot{c}_n\,|E_n\rangle + c_n\,|\dot{E}_n\rangle\right]e^{-\frac{i}{\hbar}\theta_n(t)} \;=\; 0. \tag{212}$$

Projecting this equation onto the instantaneous state $\langle E_m(t)|$ yields the differential equations for the amplitudes:

$$\dot{c}_m(t) \;=\; -\sum_n c_n(t)\,\langle E_m(t)|\dot{E}_n(t)\rangle\,e^{-\frac{i}{\hbar}[\theta_n(t)-\theta_m(t)]}. \tag{213}$$

To compute the coupling coefficient $\langle E_m|\dot{E}_n\rangle$ for $m \neq n$, we differentiate the instantaneous eigenvalue equation $H(t)\,|E_n(t)\rangle = E_n(t)\,|E_n(t)\rangle$ with respect to time:

$$\dot{H}\,|E_n\rangle + H\,|\dot{E}_n\rangle \;=\; \dot{E}_n\,|E_n\rangle + E_n\,|\dot{E}_n\rangle\,. \tag{214}$$

Projecting this onto $\langle E_m(t)|$ for $m \neq n$ gives:

$$\langle E_m|\,\dot{H}\,|E_n\rangle + E_m\,\langle E_m|\dot{E}_n\rangle \;=\; E_n\,\langle E_m|\dot{E}_n\rangle\,, \tag{215}$$

which allows us to express the transition coupling as:

$$\langle E_m(t)|\dot{E}_n(t)\rangle \;=\; \frac{\langle E_m(t)|\,\dot{H}(t)\,|E_n(t)\rangle}{E_n(t) - E_m(t)}. \tag{216}$$

Substituting this back into Eq. (213) for $m \neq 0$, and assuming the system starts in the ground state $c_0(0) = 1$ and $c_n(0) \approx 0$ for $n \neq 0$, the first-order transition amplitude to any excited state $|E_m(t)\rangle$ is given by:

$$\begin{aligned} c_m(t) \;\approx\; -\int_0^t \frac{\langle E_m(t')|\,\dot{H}(t')\,|E_0(t')\rangle}{E_0(t') - E_m(t')} \\ \times\exp\!\left(\frac{i}{\hbar}\int_0^{t'}\big[E_m(\tau) - E_0(\tau)\big]\,d\tau\right)dt'. \end{aligned} \tag{217}$$

To understand the scaling with total time T, we change to the normalized time variable $s = t'/T \in [0,1]$, where $dt' = Tds$ and $\dot{H}(t') = \frac{1}{T}\frac{dH}{ds}$. Differentiating the phases and coordinates, we write:

$$\begin{aligned} c_m(T) \;\approx\; -\int_0^1 \frac{\langle E_m(s)|\,\frac{dH}{ds}\,|E_0(s)\rangle}{E_0(s) - E_m(s)} \\ \times\exp\!\left(iT\int_0^s \omega_{m0}(s')\,ds'\right)ds, \end{aligned} \tag{218}$$

where $\omega_{m0}(s) = [E_m(s) - E_0(s)]/\hbar$ is the transition frequency. Using integration by parts for a large T:

$$c_m(T) \approx \left[i\hbar \frac{\langle E_m(s)| \frac{dH}{ds} |E_0(s)\rangle}{T[E_m(s) - E_0(s)]^2} e^{iT\Theta_m(s)} \right]_0^1 - \int_0^1 \frac{d}{ds} \left(i\hbar \frac{\langle E_m(s)| \frac{dH}{ds} |E_0(s)\rangle}{T[E_m(s) - E_0(s)]^2} \right) \times e^{iT\Theta_m(s)}\, ds, \tag{219}$$

where $\Theta_m(s) = \int_0^s \omega_{m0}(s')\, ds'$. This expression shows that the transition amplitude scales as $O(1/T)$. For the probability of transitioning to any excited state to be bounded by a small parameter $\epsilon \ll 1$, the first term must satisfy:

$$\frac{\hbar \left| \langle E_m(s)| \frac{dH}{ds} |E_0(s)\rangle \right|}{[E_m(s) - E_0(s)]^2} \ll T. \tag{220}$$

Applying this bound over the entire evolution $s \in [0, 1]$ and taking the minimum energy gap $g_{\min}$ over all states, we arrive at the standard adiabatic condition:

$$T \gg \frac{\hbar \max_{s\in[0,1]} \left| \langle E_m(s)| \frac{dH}{ds} |E_0(s)\rangle \right|}{g_{\min}^2}. \tag{221}$$

This rigorous derivation was pioneered by Kato [22] and is discussed in detail in Messiah [23] (Vol. II). Its application to universal quantum computing and optimization was formalized by Farhi et al. [24] (see also the review by Albash and Lidar [25]).

2. Adiabatic quantum optimization

To solve a discrete optimization problem, we define a *beginning Hamiltonian* whose ground state is easy to prepare:

$$H_B = -\sum_{j=0}^{n-1} X_j, \tag{222}$$

whose unique ground state is $|\psi_B\rangle = |+\rangle^{\otimes n} = \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)^{\otimes n}$, an equal superposition of all 2^n bit strings. Here, X_j is the Pauli-X operator acting on the j-th qubit.

The time-dependent Hamiltonian

$$H(t) = \big(1 - \ell(t)\big)\, H_B + \ell(t)\, H_P \tag{223}$$

evolves from H_B at $t = 0$ to H_P at $t = T$. If the evolution is adiabatically slow, the final state is the ground state of H_P, which encodes the optimal bit string $\mathbf{z}^\star$.

E. QUBO Example: Max-Cut

The *max-cut* problem asks for the partition of vertices in a graph that maximizes the number of edges crossing between the two sets.

a. Unweighted max-cut. Consider a graph $G = (V, E)$ with vertices $V = \{0, 1, \ldots, n-1\}$. Assign each vertex to one of two sets via $z_i \in \{0, 1\}$. The objective is to maximize

$$C(\mathbf{z}) = \sum_{(i,j)\in E} z_i(1 - z_j) + z_j(1 - z_i), \tag{224}$$

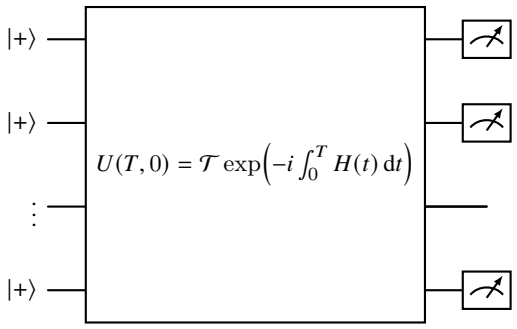


FIG. 23. Schematic adiabatic quantum optimization: start from $|+\rangle^{\otimes n}$ (ground state of H_B), evolve under $H(t) = (1 - \ell(t))H_B + \ell(t)H_P$, and measure in the computational basis. The measurement outcome with highest probability approximates the ground state of H_P.

which counts the edges (i, j) where $z_i \neq z_j$.

Using the mapping $z_i = \frac{1-s_i}{2}$ with $s_i \in \{-1, +1\}$, we have $z_i(1 - z_j) + z_j(1 - z_i) = \frac{1-s_i s_j}{2}$. The problem Hamiltonian becomes

$$H_P = \sum_{(i,j)\in E} Z_i Z_j + \text{constant}, \tag{225}$$

where $Z_i Z_j$ has eigenvalue $+1$ when spins are aligned (vertex i, j in the same set) and -1 when they are anti-aligned (different sets). Minimizing the expectation of H_P maximizes the cut.

b. Five-vertex example. For a path graph 0–1–2–3–4,

$$H_P = Z_0Z_1 + Z_1Z_2 + Z_2Z_3 + Z_3Z_4, \qquad H_B = -\sum_{j=0}^{4} X_j. \tag{226}$$

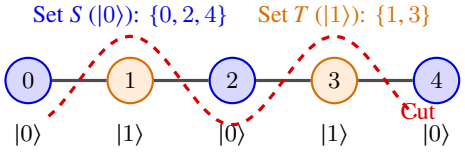


FIG. 24. Unweighted Max-Cut on a 5-vertex path graph. The vertices are partitioned into Set S (blue nodes, state $|0\rangle$) and Set T (orange nodes, state $|1\rangle$), corresponding to the state $|01010\rangle$. All 4 edges are cut (indicated by the dashed red line), achieving the maximum possible cut size of 4.

The operators act on the full 5-qubit Hilbert space as tensor products:

$$\begin{aligned} X_0 &= \sigma_x \otimes I \otimes I \otimes I \otimes I, \\ X_1 &= I \otimes \sigma_x \otimes I \otimes I \otimes I, \\ &\vdots \\ Z_0Z_1 &= \sigma_z \otimes \sigma_z \otimes I \otimes I \otimes I, \\ Z_1Z_2 &= I \otimes \sigma_z \otimes \sigma_z \otimes I \otimes I, \end{aligned} \tag{227}$$

and the interpolated Hamiltonian is

$$H(t) = \left(1 - \frac{t}{T}\right)H_B + \frac{t}{T}H_P. \tag{228}$$

c. Weighted max-cut and clustering. When edges carry weights $w_{ij} \geq 0$, the objective becomes

$$\max_{\mathbf{z}} \sum_{(i,j)\in E} w_{ij} \left[z_i(1 - z_j) + z_j(1 - z_i) \right], \tag{229}$$

and the Hamiltonian is

$$H_P = \sum_{i<j} w_{ij} Z_i Z_j. \tag{230}$$

If w_{ij} represents the distance between vertices i and j, the weighted max-cut is equivalent to *clustering*—partitioning similar points together and separating dissimilar ones.

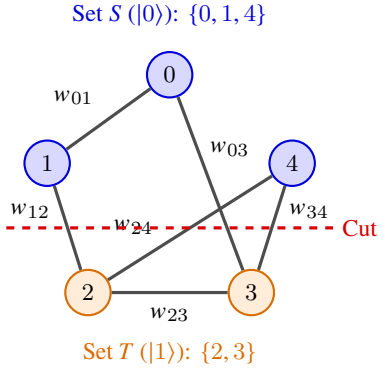


FIG. 25. Weighted Max-Cut and its clustering analogy. The 5 vertices are partitioned by a horizontal cut line (dashed red) into Set S (blue nodes, state $|0\rangle$) and Set T (orange nodes, state $|1\rangle$). This partition cuts four edges: $(1, 2)$, $(3, 4)$, $(0, 3)$, and $(2, 4)$, yielding a total cut weight of $w_{12} + w_{34} + w_{03} + w_{24}$. If the weights w_{ij} represent the distance between points, maximizing the cut isolates dissimilar points into different clusters.

F. Challenges of Adiabatic Quantum Computing

Several practical and theoretical challenges limit the applicability of AQC:

1. **Gap scaling.** The minimum gap $g_{\min}$ can scale exponentially with system size n. For example, if $g_{\min} \in O(2^{-n})$, the adiabatic condition (208) requires $T \gg O(2^{2n})$, which defeats the purpose of quantum speedup. Even when $T \in O(\text{poly}(n))$, the absolute time can still be too long for practical applications.
2. **Noise and imperfections.** The continuous Hamiltonian evolution in AQC is vulnerable to environmental noise and control errors. Fault-tolerant AQC remains an open problem.
3. **Annealing schedule.** The choice of $\ell(t)$ affects the effective gap and runtime. Optimizing the schedule is a nontrivial problem.

G. Quantum Approximate Optimization Algorithm (QAOA)

QAOA discretizes the adiabatic idea into a *gate-based* quantum algorithm, suitable for both fault-tolerant and NISQ-era devices.

1. *From adiabatic evolution to quantum circuits*

Implementing the continuous time evolution of a time-dependent Hamiltonian $H(t) = (1-\ell(t))H_B + \ell(t)H_P$ directly on a quantum circuit is difficult because $[H_B, H_P] \neq 0$.

2. *Lie–Trotter and Suzuki–Trotter Product Formulas*

Implementing the continuous-time evolution of a time-dependent Hamiltonian $H(t) = (1 - \ell(t))H_B + \ell(t)H_P$ directly on a quantum circuit is difficult because $[H_B, H_P] \neq 0$. In practice, the total evolution is discretized into small time steps, and the evolution over each step is approximated using product formulas. To analyze these approximations, we study the case of a time-independent Hamiltonian $H = A + B$, where A and B are Hermitian operators that do not commute.

a. First-Order Lie–Trotter Formula. The first-order Lie–Trotter product formula, originally introduced in the context of operator semi-groups by Trotter [26] and popularized for quantum simulation by Lloyd [27], approximates the joint evolution operator $e^{-i(A+B)\Delta t}$ for a small time step Δt as:

$$U_{\text{LT}}(\Delta t) = e^{-iA\Delta t} e^{-iB\Delta t}. \tag{231}$$

To derive the local truncation error, we perform a Taylor expansion of both the exact evolution $U(\Delta t) = e^{-i(A+B)\Delta t}$ and the Lie–Trotter operator $U_{\text{LT}}(\Delta t)$ up to second order in Δt. The exact evolution is given by:

$$\begin{aligned} e^{-i(A+B)\Delta t} &= I - i(A+B)\Delta t - \frac{1}{2}(A+B)^2\Delta t^2 \\ &\quad + O(\Delta t^3) \\ &= I - i(A+B)\Delta t - \frac{1}{2}(A^2 + AB + BA + B^2)\Delta t^2 \\ &\quad + O(\Delta t^3). \end{aligned} \tag{232}$$

Expanding the Lie–Trotter product, we obtain:

$$\begin{aligned} U_{\text{LT}}(\Delta t) &= \left(I - iA\Delta t - \frac{1}{2}A^2\Delta t^2 + O(\Delta t^3)\right) \\ &\quad \times \left(I - iB\Delta t - \frac{1}{2}B^2\Delta t^2 + O(\Delta t^3)\right) \\ &= I - i(A+B)\Delta t - \left(\frac{1}{2}A^2 + AB + \frac{1}{2}B^2\right)\Delta t^2 \\ &\quad + O(\Delta t^3). \end{aligned} \tag{233}$$

Taking the difference between the exact evolution and the Lie–Trotter approximation yields:

$$\begin{aligned} U(\Delta t) - U_{\text{LT}}(\Delta t) &= \frac{1}{2}(AB - BA)\Delta t^2 + O(\Delta t^3) \\ &= \frac{1}{2}[A, B]\Delta t^2 + O(\Delta t^3). \end{aligned} \tag{234}$$

Thus, the local truncation error is of order $O(\Delta t^2)$, and its magnitude is directly proportional to the commutator $[A, B]$.

For a total evolution time t, we divide the interval into N steps of size $\Delta t = t/N$. The global evolution operator is approximated by:

$$U(t) \approx (U_{\text{LT}}(t/N))^N = \left(e^{-iAt/N} e^{-iBt/N}\right)^N. \tag{235}$$

To find the global error bound, we use the telescoping sum identity for unitary operators. Let $V = U(\Delta t)$ and $W = U_{\text{LT}}(\Delta t)$. Since both V and W are unitary, their operator norms are $\|V\| = \|W\| = 1$. The difference of their N-th powers satisfies:

$$V^N - W^N = \sum_{j=0}^{N-1} V^{N-1-j}(V - W)W^j. \tag{236}$$

Taking the operator norm on both sides and applying the triangle inequality:

$$\begin{aligned}\|V^N - W^N\| &\leq \sum_{j=0}^{N-1} \|V\|^{N-1-j}\|V-W\|\|W\|^j \\ &= N\|V-W\|. \end{aligned} \tag{237}$$

Substituting the local error bound $\|V-W\| \leq \frac{1}{2}\|[A,B]\|\Delta t^2 + O(\Delta t^3)$ with $\Delta t = t/N$, we obtain the global error bound:

$$\begin{aligned}&\left\| e^{-i(A+B)t} - \left(e^{-iAt/N}e^{-iBt/N}\right)^N \right\| \\ &\leq \frac{t^2}{2N}\|[A,B]\| + O\left(\frac{t^3}{N^2}\right). \end{aligned} \tag{238}$$

The leading term scales as $O(t^2/N)$; the $O(t^3/N^2)$ contribution is *subleading* and comes from the $O(\Delta t^3)$ Taylor remainder in each step (via $N\,\Delta t^3 = t^3/N^2$). Thus the Lie–Trotter formula is *first-order accurate*: with fixed total time t and $N \to \infty$, the dominant global error decays as $O(t^2/N)$, equivalently $O(t\,\Delta t)$ in the step size $\Delta t = t/N$.

b. Second-Order Symmetric Suzuki–Trotter Formula. To eliminate the first-order error term, Suzuki proposed a symmetric product formula [28], defined for a single time step Δt as:

$$S_2(\Delta t) = e^{-iA\Delta t/2}e^{-iB\Delta t}e^{-iA\Delta t/2}. \tag{239}$$

By symmetrizing the operator order, the local error can be pushed to $O(\Delta t^3)$. To prove this, we expand $S_2(\Delta t)$ using the Taylor series. Let $x = -i\Delta t$. We expand each factor:

$$e^{Ax/2} = I + \frac{1}{2}Ax + \frac{1}{8}A^2x^2 + \frac{1}{48}A^3x^3 + O(x^4), \tag{240}$$

$$e^{Bx} = I + Bx + \frac{1}{2}B^2x^2 + \frac{1}{6}B^3x^3 + O(x^4). \tag{241}$$

First, multiplying the first two factors:

$$\begin{aligned}e^{Ax/2}e^{Bx} &= I + x\left(\frac{1}{2}A + B\right) + x^2\left(\frac{1}{8}A^2 + \frac{1}{2}AB + \frac{1}{2}B^2\right) \\ &\quad + x^3\left(\frac{1}{48}A^3 + \frac{1}{8}A^2B + \frac{1}{4}AB^2 + \frac{1}{6}B^3\right) \\ &\quad + O(x^4). \end{aligned} \tag{242}$$

Now, multiplying by the third factor $e^{Ax/2}$ on the right:

$$\begin{aligned}S_2(x) &= e^{Ax/2}e^{Bx}e^{Ax/2} \\ &= I + x(A+B) + \frac{1}{2}x^2(A+B)^2 \\ &\quad + x^3\left(\frac{1}{6}A^3 + \frac{1}{6}B^3 + \frac{1}{8}A^2B + \frac{1}{4}ABA \right. \\ &\quad \left. + \frac{1}{8}BA^2 + \frac{1}{4}AB^2 + \frac{1}{4}B^2A\right) + O(x^4). \end{aligned} \tag{243}$$

Comparing this term-by-term with the exact evolution expansion $e^{(A+B)x}$:

$$e^{(A+B)x} = I + x(A+B) + \frac{1}{2}x^2(A+B)^2 + \frac{1}{6}x^3(A+B)^3 + O(x^4). \tag{244}$$

We observe that the terms of order x and x^2 match exactly, proving that the local error is at least $O(x^3) = O(\Delta t^3)$. To find the exact coefficient of the x^3 term, we subtract the x^3 coefficient of $S_2(x)$ from that of $e^{(A+B)x}$. The coefficient of x^3 in $e^{(A+B)x}$ is:

$$\begin{aligned}\frac{1}{6}(A+B)^3 &= \frac{1}{6}\Big(A^3 + B^3 + A^2B + ABA \\ &\quad + BA^2 + AB^2 + BAB + B^2A\Big). \end{aligned} \tag{245}$$

Taking the difference $\Delta_3 = \frac{1}{6}(A+B)^3 - (\text{coeff of } x^3 \text{ in } S_2(x))$, we get:

$$\begin{aligned}\Delta_3 &= \left(\frac{1}{6} - \frac{1}{8}\right)A^2B + \left(\frac{1}{6} - \frac{1}{4}\right)ABA \\ &\quad + \left(\frac{1}{6} - \frac{1}{8}\right)BA^2 + \left(\frac{1}{6} - \frac{1}{4}\right)AB^2 \\ &\quad + \frac{1}{6}BAB + \left(\frac{1}{6} - \frac{1}{4}\right)B^2A \\ &= \frac{1}{24}\left(A^2B - 2ABA + BA^2\right) \\ &\quad - \frac{1}{12}\left(B^2A - 2BAB + AB^2\right) \\ &= \frac{1}{24}[A,[A,B]] - \frac{1}{12}[B,[B,A]]. \end{aligned} \tag{246}$$

Reintroducing $x = -i\Delta t$, the leading-order local truncation error is:

$$\begin{aligned}&U(\Delta t) - S_2(\Delta t) \\ &= i\left(\frac{1}{24}[A,[A,B]] - \frac{1}{12}[B,[B,A]]\right)\Delta t^3 \\ &\quad + O(\Delta t^4). \end{aligned} \tag{247}$$

Applying the telescoping sum identity again, the global error after N steps is bounded by:

$$\begin{aligned}&\left\| e^{-i(A+B)t} - \left(e^{-iAt/2N}e^{-iBt/N}e^{-iAt/2N}\right)^N \right\| \\ &\leq \frac{t^3}{12N^2}\left(\frac{1}{2}\|[A,[A,B]]\| + \|[B,[B,A]]\|\right) + O\left(\frac{t^4}{N^3}\right). \end{aligned} \tag{248}$$

This proves that the symmetric Suzuki–Trotter formula is *second-order accurate*: the leading local error is $O(\Delta t^3)$ rather than $O(\Delta t^2)$, so the dominant global error scales as $O(t^3/N^2)$—one power of t higher and one power of N lower than first-order Trotter.

c. Higher-Order Formulas via Fractal Decomposition. Suzuki showed that higher-order product formulas can be constructed recursively [29]. For instance, a fourth-order product formula $S_4(\Delta t)$ is defined by combining five second-order steps:

$$S_4(\Delta t) = S_2(p\,\Delta t)^2\, S_2((1-4p)\,\Delta t)\, S_2(p\,\Delta t)^2, \tag{249}$$

where $p = 1/(4 - 4^{1/3}) \approx 1.3512$. This recursive structure, known as fractal decomposition, can be generalized to construct $2k$-th order product formulas with global error bounds of $O(t^{2k+1}/N^{2k})$.

d. Application to Time-Dependent Evolution. Assuming a piecewise-constant schedule with time step $\Delta t \ll T$, the time-dependent evolution operator $U(T,0) =$

$\mathcal{T}\exp(-i\int_0^T H(t)dt)$ is approximated by a product of step evolutions:

$$U(T,0) \;\approx\; \prod_{j=1}^{N} \exp\big(-i\,\Delta t\,H(j\Delta t)\big), \tag{250}$$

where $N = T/\Delta t$. For QAOA, each step $\exp\big(-i\Delta t[(1-\ell)H_B + \ell H_P]\big)$ is decomposed using the first-order Lie–Trotter formula as:

$$\begin{aligned}&\exp\big(-i\Delta t[(1-\ell)H_B + \ell H_P]\big)\\ &\quad\approx\; \exp\big(-i\beta H_B\big)\,\exp\big(-i\gamma H_P\big) \;+\; O(\Delta t^2),\end{aligned} \tag{251}$$

where $\beta = \Delta t\,(1-\ell)$ and $\gamma = \Delta t\,\ell$.

3. QAOA circuit

The *depth-p QAOA* circuit applies p alternating layers of $U_B = \exp(-i\beta H_B)$ and $U_P = \exp(-i\gamma H_P)$:

$$|\boldsymbol{\beta},\boldsymbol{\gamma}\rangle \;=\; \prod_{j=1}^{p} \exp\big(-i\beta_j H_B\big)\,\exp\big(-i\gamma_j H_P\big)\;|+\rangle^{\otimes n}, \tag{252}$$

where $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_p)$ and $\boldsymbol{\gamma} = (\gamma_1,\ldots,\gamma_p)$ are *variational parameters*. The QAOA algorithm optimizes these parameters to minimize the expectation value:

$$(\boldsymbol{\beta}^\star,\boldsymbol{\gamma}^\star) \;=\; \arg\min_{\boldsymbol{\beta},\boldsymbol{\gamma}} \langle\boldsymbol{\beta},\boldsymbol{\gamma}|\,H_P\,|\boldsymbol{\beta},\boldsymbol{\gamma}\rangle\,. \tag{253}$$

The optimization is performed by a classical optimizer (e.g., gradient descent or Bayesian optimization) that iteratively updates $\boldsymbol{\beta},\boldsymbol{\gamma}$ based on repeated circuit evaluations.

4. Max-cut example with QAOA

For the max-cut problem on a graph $G = (V,E)$ with $H_P = \sum_{(i,j)\in E} Z_iZ_j$ and $H_B = -\sum_{k=0}^{n-1} X_k$, the QAOA layers decompose as

$$U_B(\beta) = \exp\Big(i\beta\sum_{k=0}^{n-1} X_k\Big) = \bigotimes_{k=0}^{n-1} e^{i\beta X_k} = \bigotimes_{k=0}^{n-1} R_x(2\beta), \tag{254}$$

$$U_P(\gamma) = \exp\Big(-i\gamma\sum_{(i,j)\in E} Z_iZ_j\Big) = \prod_{(i,j)\in E} e^{-i\gamma Z_iZ_j}, \tag{255}$$

where each factor $e^{-i\gamma Z_iZ_j}$ is implemented by a two-qubit gate sequence:

$$e^{-i\gamma Z_iZ_j} \;=\; \mathrm{CNOT}\,\big(I\otimes R_z(2\gamma)\big)\,\mathrm{CNOT}, \tag{256}$$

where $\mathrm{CNOT} = |0\rangle\langle 0|\otimes I + |1\rangle\langle 1|\otimes X$ is the controlled-NOT gate. These decompositions make QAOA hardware-efficient: each layer uses only single-qubit rotations and local two-qubit entangling gates.

5. Generalization: non-uniform schedules

Beyond the linear schedule $H(t) = (1-t/T)H_B + (t/T)H_P$, QAOA generalizes to arbitrary non-uniform schedules with independent β_j,γ_j at each layer j. The depth parameter p controls the trade-off between expressivity and circuit complexity:

- $p = 1$ gives a shallow circuit with limited expressivity but low hardware demands.
- $p \to \infty$ asymptotically converges to adiabatic evolution and can find the exact optimum for large enough T, at the cost of deep circuits.

H. Summary

This chapter covered three quantum approaches to discrete optimization:

- **QUBO** reformulates combinatorial problems as minimizing a quadratic form $\sum_{i,j} Q_{ij}z_iz_j$ over binary variables. Constraints are handled by the penalty method, converting inequalities to unconstrained QUBO via slack variables and quadratic penalty terms.
- **Adiabatic quantum computing** evolves from a simple starting Hamiltonian $H_B = -\sum X_j$ (ground state $|+\rangle^{\otimes n}$) to the problem Hamiltonian H_P encoding the objective function. The adiabatic theorem guarantees convergence to the ground state of H_P if the evolution time $T \gg O(g_{\min}^{-2})$, where $g_{\min}$ is the minimum spectral gap.
- **QAOA** discretizes adiabatic evolution into a variational quantum circuit of depth p, alternating $U_B(\beta) = \exp(-i\beta H_B)$ and $U_P(\gamma) = \exp(-i\gamma H_P)$ layers. The $2p$ parameters are optimized by a classical outer loop, making QAOA suitable for both NISQ-era and fault-tolerant devices.

The max-cut problem serves as the canonical example, where the Hamiltonian $H_P = \sum_{(i,j)\in E} Z_iZ_j$ directly encodes the graph structure. QAOA's hardware-efficient decomposition into single-qubit rotations and local entangling gates

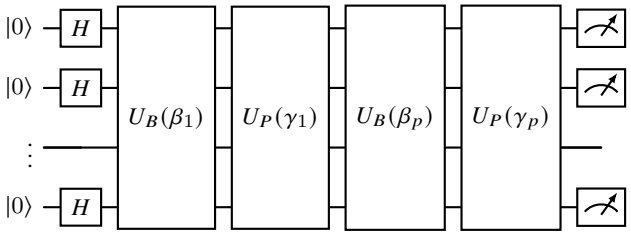


FIG. 26. Depth-p QAOA circuit. The initial Hadamard layer prepares $|+\rangle^{\otimes n}$, the ground state of H_B. p alternating layers of $U_B(\beta_j) = \exp(-i\beta_j H_B)$ and $U_P(\gamma_j) = \exp(-i\gamma_j H_P)$ are applied, followed by measurement in the computational basis. The $2p$ parameters $\boldsymbol{\beta}, \boldsymbol{\gamma}$ are optimized by a classical optimizer to minimize $\langle H_P \rangle$.

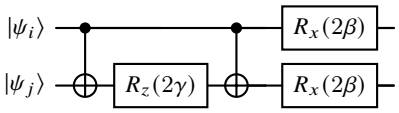


FIG. 27. Circuit decomposition of the QAOA step for a single edge (i, j): the $U_P(\gamma)$ layer implements $e^{-i\gamma Z_i Z_j}$ via the CNOT–R_z–CNOT sequence, followed by the $U_B(\beta)$ layer implementing local $R_x(2\beta)$ rotations on both qubits.

makes it one of the most promising near-term quantum optimization algorithms.

XII. CONCLUSION

This report has developed the theoretical foundations of quantum machine learning from first principles. Beginning with the postulates of quantum mechanics and the mathematical framework of Hilbert spaces and Dirac notation, we derived the quantum circuit model, established universality, proved the no-cloning theorem, and analyzed reversible computation and Landauer's principle. Detailed circuit derivations for the Hadamard test and a *pure-state* swap test (including explicit 8×8 Fredkin structure) showed how quantum expectation values and state overlaps are estimated; Sec. IX G 0 g later packaged the density-matrix swap test, Hilbert–Schmidt kernel matrices, and Pauli coordinates for QML kernels. Building toward algorithms, we reviewed the query model, Grover search, QFT, and quantum phase estimation—ingredients that feed the Harrow–Hassidim–Lloyd linear-systems solver.

Section IX synthesizes the machine-learning block of the course in the same progression: HHL, classical margin and least-squares SVMs, the Rebentrost *et al.* quantum LS-SVM pipeline (HHL plus overlap readout), hurdles to practical quantum advantage, dual SVMs and the kernel trick, and quantum encodings with feature-map / swap-test kernels. We briefly recalled quantum PCA and variational QML and highlighted open questions around dequantization, barren plateaus, and hardware assumptions.

Section X then developed the variational quantum machine learning framework in detail: the variational quantum eigensolver as the prototypical VQA, the linearity of QML models in Pauli feature space, variational quantum classifiers and their hyperplane decision boundaries, quantum convolutional neural networks with explicit circuit templates, the universal approximation theorem for quantum models, the parameter-shift rule for exact gradient computation, and the fundamental limits imposed by quantum state discrimination and the Helstrom bound.

Section XI covered quantum approaches to combinatorial optimization. The QUBO framework and Ising Hamiltonian encoding were developed together with the penalty method for handling constraints. A self-contained derivation of the adiabatic condition—expanding the state in the instantaneous eigenbasis, applying Schrödinger's equation, and using integration by parts— yielded the quantitative bound

$$T \gg \frac{\hbar \max_{s\in[0,1]} \left| \langle E_m(s) | \partial_s H | E_0(s) \rangle \right|}{g_{\min}^2},$$

as originally proved by Kato [22] and applied to quantum computing by Farhi *et al.* [24]. Max-cut was presented as the canonical QUBO example. QAOA was derived from adiabatic evolution by approximating each infinitesimal step with the first-order Lie–Trotter formula (Trotter [26]; Lloyd [27]), which was proved to have a local truncation error $\frac{1}{2}[A, B]\Delta t^2$ and a global error bound $O(t^2/N)$ via the telescoping identity. The second-order symmetric Suzuki–Trotter formula [28] was shown to cancel the leading error term, with the $O(\Delta t^3)$ residual expressed through double commutators $\frac{1}{24}[A, [A, B]] - \frac{1}{12}[B, [B, A]]$, giving a global bound $O(t^3/N^2)$. Higher-order formulas via Suzuki's fractal decomposition [29] were sketched.

Natural extensions include kernelized dual forms in fault-tolerant settings, deeper Hamiltonian-simulation oracles, and end-to-end resource estimates for QML pipelines.

[1] J. Preskill, Quantum computing in the nisq era and beyond, Quantum **2**, 79 (2018).

[2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. (Cambridge University Press, Cambridge, UK, 2010).

[3] M. M. Wilde, *Quantum Information Theory*, 2nd ed. (Cambridge University Press, Cambridge, UK, 2017).

[4] J. Watrous, *The Theory of Quantum Information* (Cambridge University Press, Cambridge, UK, 2018).

[5] M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers*, Quantum Science and Technology (Springer, Cham, Switzerland, 2021).

[6] E. Bernstein and U. Vazirani, Quantum complexity theory, SIAM Journal on Computing **26**, 1411 (1997).

[7] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Physical Review Letters **103**, 150502 (2009).
[8] W. K. Wootters and W. H. Zurek, A single quantum cannot be cloned, Nature **299**, 802 (1982).
[9] C. H. Bennett and S. J. Wiesner, Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states, Physical Review Letters **69**, 2881 (1992).
[10] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels, Physical Review Letters **70**, 1895 (1993).
[11] R. Landauer, Irreversibility and heat generation in the computing process, IBM Journal of Research and Development **5**, 183 (1961).
[12] J. A. K. Suykens and J. Vandewalle, Least squares support vector machine classifiers, Neural Processing Letters **9**, 293 (1999).
[13] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification, Physical Review Letters **113**, 130503 (2014).
[14] M. Schuld and N. Killoran, Quantum machine learning in feature hilbert spaces, Physical Review Letters **122**, 040504 (2019).
[15] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature **549**, 195 (2017).
[16] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nature Physics **10**, 631 (2014).
[17] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nature Reviews Physics **3**, 625 (2021).
[18] M. Schuld, R. Sweke, and J. J. Meyer, Effect of data encoding on the expressive power of variational quantum-machine-learning models, Physical Review A **103**, 032430 (2021).
[19] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Physical Review A **98**, 032309 (2018).
[20] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Physical Review A **99**, 032331 (2019).
[21] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, General parameter-shift rules for quantum gradients, Quantum **6**, 677 (2022).
[22] T. Kato, On the adiabatic theorem of quantum mechanics, Journal of the Physical Society of Japan **5**, 435 (1950).
[23] A. Messiah, *Quantum Mechanics*, Vol. II (North-Holland Publishing Company, Amsterdam, 1961).
[24] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, arXiv preprint quant-ph/0001106 (2000).
[25] T. Albash and D. A. Lidar, Adiabatic quantum computation, Reviews of Modern Physics **90**, 015002 (2018).
[26] H. F. Trotter, On the product of semi-groups of operators, Proceedings of the American Mathematical Society **10**, 545 (1959).
[27] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).
[28] M. Suzuki, Relationship between d-dimensional quantal spin systems and $(d+1)$-dimensional ising systems: Equivalence, critical exponents and systematic approximants of the partition function and spin correlations, Progress of Theoretical Physics **56**, 1454 (1976).
[29] M. Suzuki, Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations, Physics Letters A **146**, 319 (1990).