Scientific Research Publishing

Journal of Quantum Information Science, 2026, 16(1), 16-74
https://www.scirp.org/journal/jqis
ISSN Online: 2162-576X
ISSN Print: 2162-5751

The Grover Dilemma and Three Fundamental Barriers to Oracle-Based Quantum Search Algorithms

Ying Liu

Department of Engineering Technology, Savannah State University, Savannah, GA, USA
Email: liuy@savannahstate.edu

How to cite this paper: Liu, Y. (2026) The Grover Dilemma and Three Fundamental Barriers to Oracle-Based Quantum Search Algorithms. *Journal of Quantum Information Science*, **16**, 16-74.
https://doi.org/10.4236/jqis.2026.161002

Received: December 23, 2025
Accepted: March 1, 2026
Published: March 4, 2026

Abstract

Grover's algorithm is widely celebrated as providing quadratic quantum speedup for unsorted database search, forming the theoretical foundation for numerous claimed quantum advantages in machine learning, optimization, and computational applications. We demonstrate that Grover's algorithm and oracle-based quantum search algorithms face three fundamental barriers that severely limit their practical applicability. First, we identify the Grover Dilemma: when the computational space exceeds the valid data space, Grover's algorithm must choose between 1) creating uniform superposition over all computational states—including invalid states that lead to incorrect results, 2) restricting superposition to only valid states containing solutions—requiring prior knowledge that reduces the problem to trivial $O(1)$ complexity, or 3) constructing superposition over valid states through classical preprocessing—requiring $O(N)$ cost that eliminates the claimed $O\left(\sqrt{N}\right)$ quantum advantage. We extend this analysis to establish a unified framework of three independent barriers affecting oracle-based quantum search: 1) the Grover Dilemma (superposition construction for structured problems), 2) the Setup Cost Dilemma (oracle construction and data loading costs), and 3) Oracle Circularity (oracle specification requiring solution of the target problem). These barriers are logically independent—a quantum algorithm must avoid all three to achieve genuine advantage. Through systematic analysis of major problem classes and specific quantum algorithms, we found that oracle-based quantum search provides genuine computational advantage only in an exceptionally narrow class of problems. Our contributions are formalized in ten theorems: Theorems 6.1-6.5 establish the general framework (Grover Dilemma Barrier, Setup Cost Barrier, Oracle Circularity Barrier, Composite Barrier, and general conditions for no quantum advantage); Theorems 6.6-6.8 prove no quantum advantage for Deutsch's Algorithm, Deutsch-Jozsa Algorithm, and Simon's Algorithm in practical scenarios; Theorems 6.9-6.10 prove no quantum advantage for NP-complete problems and learning/optimization problems subject to Oracle Circularity. The three-barrier framework provides a systematic method for evaluating quantum search algorithm claims and distinguishing genuine advantages from artifacts of incomplete cost analysis.

Keywords

Quantum Algorithms, Grover's Algorithm, Quantum Search, Oracle-Based Algorithms, Amplitude Amplification, Deutsch-Jozsa Algorithm, Simon's Algorithm, NP-Complete Problems

1. Introduction

Quantum computing promises revolutionary computational advantages across numerous domains, with Grover's algorithm [1] standing as one of the most celebrated examples of quantum speedup. Since its introduction in 1996, Grover's algorithm has been hailed as providing a quadratic advantage for searching unsorted lists, reducing the classical $O(N)$ search complexity to quantum $O\left(\sqrt{N}\right)$. This foundational result has spawned many papers, claiming quantum advantages in machine learning [2] [3], optimization [4] [5], cryptography [6], and database applications [7] [8].

The theoretical elegance of Grover's algorithm lies in its use of quantum superposition to simultaneously explore all possible search states, combined with amplitude amplification to enhance the probability of measuring the correct answer [9]. This approach has been generalized through amplitude amplification techniques [10] and extended to numerous problem domains where quantum speedups are claimed [11] [12]. Recent surveys suggest that Grover-based algorithms form the backbone of quantum machine learning [13] and are central to the promise of quantum advantage in near-term applications [14].

However, despite decades of theoretical development and extensive experimental validation of Grover's algorithm in controlled quantum systems [15] [16], practical quantum advantages remain elusive. The gap between theoretical promise and practical reality has been attributed to hardware limitations [17], decoherence effects [18], and the challenges of quantum error correction [19]. More fundamentally, recent work has questioned whether quantum machine learning algorithms provide genuine advantages over their classical counterparts [20] [21], with several claimed quantum speedups being "dequantized" through improved classical algorithms [22] [23].

1.1. The Fundamental Problem

In this paper, we identify a deeper, more fundamental issue that challenges the practical applicability of Grover's algorithm and its variants. We demonstrate that

Grover's algorithm faces an inescapable dilemma when applied to realistic search problems: it must either 1) include invalid states in its superposition (leading to incorrect results), or 2) restrict itself to trivial problems where the solution space is already known (eliminating any quantum advantage), or 3) create a nonuniform initial superposition over only valid states (*i.e.*, anything other than the uniform superposition in Equation 1) with $O(N)$ classical preprocessing cost, eliminating the claimed $O\left(\sqrt{N}\right)$ advantage. This dilemma is not merely a technical limitation but a fundamental incompatibility between the uniform superposition assumption underlying quantum search algorithms and the structured nature of real-world computational problems.

Our analysis reveals that most claimed applications of Grover's algorithm fall into one of several categories:

- Those that produce wrong answers due to improper problem formulation.
- Those that solve artificially simplified problems where quantum advantage is meaningless.
- Those that provide no genuine quantum advantage when all costs are properly accounted for.

This critique is formalized through the Grover Dilemma. Through concrete examples—including searching for non-existent elements in structured datasets, database queries with data loading costs, and applications to NP-complete problems—we show that the dilemma applies broadly to quantum algorithms that rely on Grover-like speedups, including many quantum machine learning algorithms. Our analysis reveals that the uniform superposition assumption—fundamental to most quantum search algorithms—is incompatible with the structured nature of real-world problems where not all computational states correspond to valid problem instances.

1.2. Complete Quantum Complexity

It is important to establish that quantum algorithm analysis must account for all computational costs, not just quantum gate operations.

DEFINITION 1.1 (Complete Quantum Complexity): The total complexity of a quantum algorithm must include:

1) Classical preprocessing: Oracle construction and initial problem setup.
2) Quantum execution: Quantum gates and measurements.
3) Classical postprocessing: Result extraction and verification.

Theoretical quantum algorithm analysis often treats oracles and prepared quantum states as "free" resources, ignoring the classical computation required to construct them. When these costs are included in complete complexity analysis, many claimed quantum advantages disappear.

1.3. The Three Barriers Framework

The Grover Dilemma, when examined comprehensively, extends to three distinct and independent barriers that quantum search algorithms must overcome to achieve genuine computational advantage:

BARRIER 1 (Grover Dilemma, Section 3): For problems where the computational space S exceeds the valid data space D, constructing superposition over only valid states requires $O(|D|)$ or $O(|S|)$ classical preprocessing. This barrier affects database search, structured data queries, and any problem where not all computational basis states represent valid instances.

BARRIER 2 (Setup Cost Dilemma, Section 4): Oracle construction or data loading may require classical computation with cost, C_{setup}, that equals or exceeds the cost of classical algorithms for solving the problem. This barrier affects problems with black-box function specifications, data-dependent algorithms, and cases requiring expensive oracle construction.

BARRIER 3 (Oracle Circularity, Section 5): Oracle construction or specification may require solving the target problem, creating circular dependency where $C\text{(oracle)} \geq C\text{(problem)}$. This barrier affects NP-complete problems, learning and optimization problems, and any problem where the oracle must identify optimal or satisfying solutions.

Key property: These barriers are logically independent—a problem can suffer from one, two, or all three. A quantum algorithm must avoid all three barriers to achieve genuine advantage.

1.4. Main Results

We systematically analyze major problem classes and found:

Algorithms affected by Setup Cost Dilemma:

- Deutsch's Algorithm (general case): $O(2^n)$ oracle construction eliminates advantage (Section 4, Theorem 6.6).
- Deutsch-Jozsa Algorithm (black-box): $O(2^n)$ oracle construction eliminates advantage (Section 4, Theorem 6.7).
- Simon's Algorithm (black-box): $O(2^n)$ oracle construction eliminates advantage (Section 4, Theorem 6.8).

Special cases with limited applicability: There are special cases where oracles can be efficiently constructed from explicit problem specifications (e.g., Boolean formulas, algebraic expressions). However, when problems have sufficient structure to enable efficient quantum oracle construction, classical algorithms can typically exploit the same structure equally efficiently:

- Deutsch-Jozsa (explicit formulas): Both quantum oracle construction and classical evaluation are $O(n)$.
- Simon's Algorithm (linear functions): Classical can solve $as = 0$ (Section 4) directly via Gaussian elimination in $O(n^3)$.

In these cases, classical algorithms are more than a match—they achieve similar or better complexity using the same structural information.

Problems affected by Oracle Circularity:

- Subset Sum: Oracle must identify which subsets sum to target T, requiring solution of the problem itself (Theorem 6.9).

- NP-complete problems (general): Oracle must mark satisfying assignments or optimal configurations, requiring solution of the NP-complete problem (Theorem 6.9).
- Learning and optimization: Oracle must identify optimal policies or best parameters, requiring solution of the learning/optimization problem (Theorem 6.10).

Critical Finding: Under rigorous evaluation—where 1) all costs including oracle construction, data loading, and verification are included, 2) quantum algorithms are compared against optimal classical algorithms, and 3) structured cases are compared against classical algorithms exploiting the same structure—oracle-based quantum search provides computational advantage only for an exceptionally narrow problem class: cryptographic primitives. For all other analyzed problem classes (database search, Deutsch-type algorithms, NP-complete problems, and learning/optimization), no quantum advantage exists.

Formal Results: We establish ten theorems (Section 6) characterizing conditions under which quantum advantage cannot exist:

General Framework (Theorems 6.1-6.5):

- Theorem 6.1: Grover Dilemma Barrier
- Theorem 6.2: Setup Cost Barrier
- Theorem 6.3: Oracle Circularity Barrier
- Theorem 6.4: Composite Barrier (any single barrier eliminates advantage)
- Theorem 6.5: Oracle-Based Search Limitations (general conditions)

Specific Algorithms (Theorems 6.6-6.8):

- Theorem 6.6: No advantage for Deutsch's Algorithm
- Theorem 6.7: No advantage for Deutsch-Jozsa Algorithm
- Theorem 6.8: No advantage for Simon's Algorithm

Problem Classes (Theorems 6.9-6.10):

- Theorem 6.9: No advantage for NP-complete problems (general case)
- Theorem 6.10: No advantage for learning and optimization problems

Systematic Categorization: Section 7 identifies five categories of misleading quantum advantage claims:

Category A: Incorrect results (ignoring validity constraints)

Category B: Circular assumptions (assuming solutions known)

Category C: Hidden classical overhead (incomplete cost accounting)

Category D: Oracle abstraction (oracle requires solving problem)

Category E: Special structured problems (classical equally efficient)

This categorization provides a systematic method for evaluating future quantum algorithm claims and identifying which category of flawed analysis applies.

1.5. Related Work and Prior Critiques

The limitations of quantum computing have been the subject of ongoing scholarly debate since the field's inception. Our three-barrier framework synthesizes and extends several distinct lines of prior critique while providing a unified formal structure for evaluating oracle-based quantum search algorithms.

The Input Problem and Data Loading Costs: The challenge of loading classical data into quantum states—sometimes called the "input problem"—has been recognized since the early days of quantum algorithm development [20]. Ambainis [24] provides a comprehensive overview of quantum search algorithms and their fundamental assumptions. Montanaro [25] further surveys quantum algorithmic paradigms, distinguishing search-based approaches from structure-exploiting methods. Aaronson [20] emphasized that quantum algorithms must account for the cost of preparing quantum states encoding classical data, noting that "read the fine print" applies to many quantum speedup claims. The data loading bottleneck particularly affects algorithms claiming exponential speedups for machine learning tasks, where classical data must first be encoded in quantum-accessible form. The data loading cost cannot be ignored if it has an exponential time complexity. Implementing quantum algorithms requires careful construction of quantum circuits from elementary gates [26]. Quantum Random Access Memory (QRAM) has been proposed as a potential solution [27], enabling quantum algorithms to query classical databases in superposition. However, QRAM itself faces substantial implementation challenges. Giovannetti *et al.* [27] showed that QRAM requires complex quantum circuits with substantial overhead, and subsequent analyses have revealed that QRAM query costs may scale logarithmically or worse depending on architecture. [Arunachalam and de Wolf's survey on guest quantum search algorithms [28] provides a comprehensive review of data loading challenges and their impact on quantum algorithm complexity. These costs, often omitted from theoretical complexity analyses, can eliminate claimed quantum advantages when properly accounted for—a phenomenon we formalize as the Setup Cost Dilemma in Section 4.

Oracle Construction and the Black-Box Assumption: The oracle model in quantum complexity theory assumes oracles can be queried efficiently without accounting for construction costs. While this abstraction enables clean theoretical analysis, it can obscure practical limitations. Bennett *et al.* [7] identified early concerns about quantum algorithm practicality, noting that "strengths and weaknesses of quantum computing" depend critically on problem structure and oracle assumptions. Zalka [29] proved that Grover's algorithm is optimal for unstructured search, establishing a fundamental lower bound on quantum query complexity. However, this optimality result assumes the oracle is available without cost—an assumption that breaks down when oracle construction requires substantial classical computation. Our Setup Cost Dilemma (Section 4) formalizes when oracle construction costs eliminate quantum advantages. For structured problems admitting efficient classical solutions, quantum search may provide no benefit. Jozsa [21] observed that quantum algorithms exploiting problem structure (like Shor's factoring) differ fundamentally from unstructured search algorithms. When problems have sufficient structure for efficient quantum oracle construction, classical algorithms can typically exploit the same structure—a pattern we document systematically in Section 4.

Dequantization and Classical Algorithm Improvements: Recent years have seen several claimed quantum machine learning advantages "dequantized" through improved classical algorithms. Tang [22] developed a classical recommendation system matching the performance of a claimed quantum algorithm, demonstrating that sampling-based classical techniques could replicate quantum speedups. Gilyén *et al.* [23] extended this approach to other machine learning problems, showing that quantum advantages often depend on unrealistic assumptions about data access or problem structure. These dequantization results reveal a common pattern: quantum algorithms claiming exponential speedups for machine learning often assume quantum access to classical data structures (QRAM) or oracles encoding problem solutions. When these assumptions are made explicit and their costs accounted for, the quantum advantages vanish. Our framework extends this line of work by identifying three distinct barriers that systematically eliminate such claimed advantages.

Quantum Complexity Theory and Separations: The theoretical foundations of quantum advantage rest on complexity-theoretic separations between classical and quantum computation. Aaronson and Ambainis [12] identified problems like Forrelation that provably separate quantum from classical complexity, demonstrating that genuine quantum advantages exist for certain structured problems. However, these separations typically rely on oracle access models and worst-case complexity analysis. The gap between worst-case theoretical separations and practical algorithmic advantage is substantial. Preskill [14] discussed the challenges of achieving quantum advantage in the NISQ (Noisy Intermediate-Scale Quantum) era, noting that decoherence, gate errors, and limited connectivity constrain practical implementations. While our analysis focuses on asymptotic complexity rather than hardware limitations, it identifies fundamental barriers that persist even assuming perfect quantum hardware.

Structure-Exploiting vs Search-Based Quantum Algorithms: Early quantum algorithms—including Deutsch's algorithm [30] and the Deutsch-Jozsa algorithm [31] for function properties, Simon's algorithm [32] for hidden periods, and related approaches [33]—demonstrated quantum advantages for specific structured problems. These foundational results, along with advances in quantum algorithms [34] and applications to reinforcement learning [35]-[37], established the field's potential. A crucial distinction exists between quantum algorithms that exploit mathematical structure and those based on unstructured search. Shor's factoring algorithm [38] achieves exponential speedup by exploiting the periodic structure of modular exponentiation through the Quantum Fourier Transform—not through amplitude amplification over exponential search spaces. Similarly, quantum simulation algorithms [39]-[41] achieve exponential advantages by directly implementing quantum dynamics rather than searching solution spaces. Montanaro's overview [25] distinguishes these paradigms, noting that quantum advantages arise from "exponential quantum parallelism" in structured settings rather than from search speedups alone. Our framework makes this distinction precise: algorithms avoiding oracle-based search over exponential spaces can achieve genuine quantum advantage (Shor's, quantum simulation), while search-based algorithms face the three barriers we identify.

NP-Complete Problems and Oracle Limitations: The application of quantum algorithms to NP-complete problems has been extensively studied. Garey and Johnson [42] established the foundational theory of NP-completeness, and subsequent work has explored whether quantum computation could provide advantages for these problems. The consensus among complexity theorists is that quantum computers likely cannot solve NP-complete problems in polynomial time [43] [44]. However, Grover's algorithm does provide a quadratic speedup for unstructured search over NP-complete solution spaces, reducing complexity from $O(2^n)$ to $O(2^{n/2})$. Our analysis goes further, demonstrating that even this quadratic speedup becomes illusory when oracle construction costs are included (Section 5). For NP-complete problems, the oracle must identify which candidate solutions satisfy the problem constraints—but determining this is precisely the computational challenge we're trying to solve. This Oracle Circularity represents a fundamental barrier beyond query complexity lower bounds.

Quantum Machine Learning Critiques: The promise of quantum machine learning has been subject to increasing scrutiny. Biamonte *et al.* [13] surveyed quantum machine learning algorithms, noting both potential advantages and significant challenges. Subsequent work has revealed that many claimed advantages rely on unrealistic assumptions about quantum data access or disappear when compared against optimal classical algorithms rather than naive baselines. Aaronson [20] particularly emphasized the importance of "reading the fine print" in quantum speedup claims, noting that data loading costs, sample complexity, and oracle construction requirements often eliminate apparent advantages. Our Oracle Circularity barrier (Section 5) formalizes this critique for learning and optimization problems: the oracle must encode knowledge of optimal solutions, but discovering optimal solutions is the learning problem itself.

Positioning of This Work: Our contribution synthesizes and extends these prior critiques through a unified formal framework identifying three independent barriers:

- Grover Dilemma (Section 3): Extends the input problem to structured search, formalizing when superposition construction requires $O(N)$ classical preprocessing.
- Setup Cost Dilemma (Section 4): Generalizes data loading and oracle construction costs into a unified barrier, with systematic analysis across problem classes.
- Oracle Circularity (Section 5): Identifies a fundamental logical barrier beyond computational costs—when oracle specification requires solving the target problem.

Novel contributions:

- Unified framework: Previous critiques addressed specific issues (data loading, oracle costs, specific algorithms). We provide a systematic framework covering all oracle-based quantum search.
- Independence of barriers: We demonstrate that the three barriers are logically independent—problems may suffer from one, two, or all three. A quantum algorithm must avoid all three to achieve advantage.
- Formal theorems: Sections 6 establishes ten theorems (6.1-6.10) providing precise conditions for when quantum advantage cannot exist, going beyond informal critiques to rigorous mathematical results.
- Comprehensive classification: Section 7 systematically categorizes all analyzed problems by which barriers they face, revealing patterns across problem classes.
- Oracle Circularity formalization: While data loading costs and oracle complexity have been discussed informally, Oracle Circularity—where oracle construction logically requires solving the problem—represents a distinct barrier we formalize and prove eliminates quantum advantage.

This work does not claim quantum computing has no future—Shor's algorithm and quantum simulation provide genuine advantages. Rather, we clarify that oracle-based quantum search faces systematic fundamental barriers for most practical problems, and that quantum computing's future lies in paradigms exploiting mathematical structure rather than exhaustive search over exponential spaces.

1.6. Scope and What This Work Does NOT Claim

This analysis specifically targets oracle-based quantum search algorithms using amplitude amplification over exponential search spaces. It does not claim:

- Quantum computing has no advantages (Shor's factoring and quantum simulation provide genuine advantages).
- All quantum algorithms fail (non-search paradigms succeed through different mechanisms).
- Future developments cannot help (new paradigms avoiding barriers may emerge).
- What this analysis DOES claim:
- Oracle-based quantum search faces three fundamental barriers.
- Most claimed search-based quantum advantages are artifacts of incomplete analysis.
- No practical advantage found for problems analyzed when all costs are included.
- The framework provides systematic evaluation method for quantum search algorithms.

Successful quantum paradigms like Shor's algorithm (exploiting mathematical structure through period finding and Quantum Fourier Transform) and quantum simulation (direct physical implementation of quantum dynamics) succeed precisely because they avoid the oracle-based search paradigm and its associated barriers.

1.7. Paper Organization

Section 2 (Background):

- Reviews Grover's algorithm, its assumptions, and the standard oracle model.
- Establishes notation and presents the uniform superposition assumption (Equation 1) that underlies most quantum search algorithms.

Section 3 (Grover Dilemma):

- Introduces the fundamental dilemma facing quantum search on structured problems.
- Defines Type A problems (computational space exceeds valid data space) and Type B problems (all computational states valid).
- Shows that Type A problems require $O(N)$ classical preprocessing to identify valid states, eliminating quantum advantage.
- Presents Type B problems as an exception that avoids this particular barrier but typically faces others.

Section 4 (Setup Cost Dilemma):

- Analyzes oracle construction and data loading costs.
- Examines Deutsch's algorithm, Deutsch-Jozsa algorithm, and Simon's algorithm.
- Shows that black-box functions require exponential oracle construction, while structured functions enable efficient construction but classical algorithms achieve similar efficiency.
- Includes analysis of special cases and their limitations.

Section 5 (Oracle Circularity):

- Identifies the fundamental barrier where oracle specification requires solving the target problem.
- Provides detailed case studies: Subset Sum (showing how Type B problems avoiding the first two barriers still fails due to circularity), reinforcement learning for stock trading (showing circularity plus existence of superior classical dynamic programming solution), and generalization to other NP-complete problems.
- Introduces Oracle Impossibility for problems at infeasible scales.

Section 6 (Unified Framework):

- Synthesizes the three barriers into a formal framework.
- States and proves ten theorems characterizing when quantum advantage cannot exist.
- Provides general conditions (Theorems 6.1-6.5), specific algorithm results (Theorems 6.6-6.8), and problem class results (Theorems 6.9-6.10).

Section 7 (Discussion):

- Summarizes findings across all problem classes with comprehensive classification table.
- Identifies five categories of misleading claims.

- Discusses future work and additional physical implementation barriers beyond the three computational barriers.
- Acknowledges scope and limitations of the analysis.

Section 8 (Conclusion):

- Synthesizes main contributions, discusses implications for quantum computing research, and suggests productive future directions focusing on paradigms that naturally avoid the three barriers.

2. Grover's Algorithm and Its Assumptions

Grover's algorithm [1] provides a quantum search method with $O\left(\sqrt{N}\right)$ query complexity, offering a quadratic speedup over classical $O(N)$ linear search. However, as we demonstrate in this section, the algorithm relies on several critical assumptions that are often violated in practical applications.

2.1. Algorithm Structure

The Grover algorithm is:

Input: Search space of size N, Oracle O

Output: Target item x^*

1) Initialization: Uniform superposition in Equation (1)

2) Grover Iteration (repeat $O\left(\sqrt{N}\right)$ times):

- Oracle: Apply Equation (4)
- Amplitude Amplification: Apply diffusion operator

3) Measurement: Measure to obtain target x^*

The total query complexity is $O\left(\sqrt{N}\right)$, providing a quadratic speedup over classical search requiring $O(N)$ queries [1] [27].

2.2. Quantum Superposition Preparation

The algorithm begins by preparing a uniform superposition over all possible states [26]:

$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle \tag{1}$$

where $N = 2^n$. Applying the Hadamard gate [26]:

$$H|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) = |+\rangle \tag{2}$$

to all n qubits:

$$|\psi\rangle = H^{\otimes n}|0\rangle^{\otimes n} \tag{3}$$

The preparation requires $n = \log_2(N)$ Hadamard gates and can be performed in constant depth $O(1)$ with parallel gate operations, or $O(n) = O(\log N)$ sequential gate operations [9] [24]. However, this $O(\log N)$ cost only applies to creating the uniform superposition over computational basis states. As we show in Example 2.1, practical search problems require additional setup to encode problem-specific data, which can incur $O(N)$ overhead.

2.3. Oracle and Amplitude Amplification

The core of Grover's algorithm consists of repeated application of the Grover operator $G = D \circ O$, where O is the oracle operator and D is the diffusion operator.

Oracle Operator: The oracle O marks target states by phase inversion:

$$O|x\rangle = (-1)^{f(x)}|x\rangle \tag{4}$$

where

$$f(x) = (x == \text{target} ? 1 : 0) \tag{5}$$

Amplitude Amplification is implemented by the diffusion operator $D = 2|\psi_0\rangle\langle\psi_0| - I$. After $O(\sqrt{N})$ Grover iterations, the probability (measuring a marked state) approaches one [1].

REMARK 1 (Decision vs Search Problems): There is a crucial distinction between decision problems and search/optimization/learning problems: decision problems ask "Does a solution exist?" and require only recognition of valid solutions; search problems ask "Find a solution and its location" and require returning the solution itself; and optimization/learning problems ask "Find the best solution" and require evaluating solution quality.

Grover's algorithm, as typically presented, solves the decision problem: it amplifies the amplitude of states satisfying $f(x) = 1$, then measures to obtain a candidate solution. The algorithm does not inherently provide the location or index of the solution in a classical data structure—this requires additional $O(N)$ overhead to map quantum states to classical indices or to verify the solution's location. More fundamentally, for learning and optimization problems, defining the oracle function $f(x)$ itself requires knowing what constitutes a "good" solution—precisely the knowledge the algorithm claims to discover. This circularity is examined in Section 5.

2.4. Comparison with Classical Search

EXAMPLE 2.1 (Data Loading Overhead):

Approach 1: Classical Preprocessing

1) Load all N index into a classical array, index ($O(N)$)

2) Oracle find state $|i\rangle$

3) Return index $[i]$

Cost: $O(N)$ preprocessing eliminates the $O(\sqrt{N})$ quantum advantage—we've already touched all N data items classically before the quantum algorithm even begins.

Approach 2: Quantum RAM (QRAM)

- Add index information: $|i\rangle \to |i\rangle|0\rangle$, where $|0\rangle$ will hold index
- Preprocessing or Postprocessing: $|i\rangle|0\rangle \to |i\rangle|\text{index}[i]\rangle$

Problem: QRAM queries have significant overhead: Each QRAM query may require $O(\log N)$ classical operations [27]. Even with QRAM, the query access overhead can eliminate practical quantum speedup.

The fundamental issue: Quantum superposition provides "parallelism" over indices, but the data values must still be accessed. There is no way to avoid the cost of making indices available to the quantum oracle—either through $O(N)$ classical preprocessing (Approach 1) or through more expensive QRAM (Approach 2). This data loading bottleneck is distinct from the $O(\log N)$ cost of creating the superposition over indices and represents a fundamental limitation of quantum search for single-shot problems.

2.5. Instance-Independent Oracle

The oracle, O, in Grover's algorithm is problem-specific (different problems have different oracles) but instance-independent (the same oracle works for all instances of a given problem). The oracle, O, must work for ALL instances—it cannot be "tuned" or specialized to particular data values (instance). This instance-independence has profound implications (Section 3).

EXAMPLE 2.2 (Instance Independence): Consider searching for the value 2 in an unsorted list: $D_1 = \{0, 1, 3, 2\}$, $D_2 = \{0, 1, 2, 2\}$, and $D_3 = \{5, 7, 2, 9\}$. The oracle implementation must be fixed before seeing any specific data instance. For our three instances, the target positions are 3, {2, 3}, and 2. The oracle cannot distinguish D_1, D_2, and D_3. Instead, the oracle implements:

$$f(x) = (x == 2 ? 1 : 0) \tag{6}$$

Why this matters: The oracle needs runtime access to the actual data values, otherwise, the Oracle is not solving a particular instance (Instance Independence). There are only two ways to provide this access: Option 1. Classical Preprocessing, and Option 2. Quantum RAM.

The fundamental constraint: Instance-independence forces the oracle to be a general-purpose data accessor rather than an instance-specific circuit. The oracle cannot "know" in advance where target values are located (that's what we're searching for!) or what values the dataset contains (that's what makes it instance-independent). Therefore, the oracle must access the data during algorithm execution, and this data access cost—whether through classical memory queries or QRAM operations—eliminates the quantum speedup. This is not a technological limitation that better quantum computers might overcome—it's a logical requirement of the problem structure. Any oracle that works for all instances of a problem (e.g., searching any database with any data values) must have a mechanism to access the specific instance, and such data access has unavoidable costs.

2.6. Critical Assumptions

Grover's algorithm, as typically presented, relies on several assumptions that are often violated in practical applications:

1) Complete search space: All N computational basis states $|0\rangle, |1\rangle, \ldots, |N-1\rangle$ represent valid problem instances.

2) Oracle availability: A quantum oracle can efficiently evaluate $f(x)$ for any in-

put x without classical preprocessing overhead.

3) Uniform prior distribution: All valid states are equally likely to contain solutions a priori, justifying the uniform superposition initialization in Equation (1).

4) Problem-data separation: The algorithm structure is independent of instance-specific data, with data access cost not dominating the quantum speedup.

In Section 3, we demonstrate that these assumptions lead to a fundamental dilemma, the Grover Dilemma.

3. The Grover Dilemma

While Grover's algorithm provides a theoretical $O\left(\sqrt{N}\right)$ speedup for unsorted search, its application to practical problems faces fundamental obstacles. The unsorted list search problem used in Section 2, despite its pedagogical value, represents an exception rather than the rule in two critical ways:

1) **Oracle is well-defined before solving:** For example, Equation (6) is clearly specified without requiring knowledge of the solution.

2) **Simple oracle evaluation:** For example, Equation (6) only requires O(1) operations.

However, even for this simple case, the $O(N)$ data loading overhead (Example 2.1, 2.2) already eliminates quantum advantage for single-shot queries. More critically, most practical search problems either require expensive oracle construction (Section 4) or face oracle circularity where the oracle cannot be specified without solving the problem (Section 5), creating fundamental barriers for Oracle-based search algorithms.

3.1. Structured Search Problems

Most practical search problems have structure that violates Grover's assumption that "all N states $|0\rangle, |1\rangle, \ldots, |N-1\rangle$ represent valid problem instances."

DEFINITION 3.1 (Structured Search Problem): A structured search problem consists of:

- A problem instance with data $D = \{d_1, d_2, \ldots, d_m\}$
- A computational search space $S = \{s_1, s_2, \ldots, s_N\}$, where $N \geq m$
- A validity function $V: S \to \{0, 1\}$, where $V(s) = (s \in D?1:0)$
- A target predicate $T: D \to \{0, 1\}$, where $T(d)$ = (search criteria?1:0)

The key issue: The computational space S (what quantum superposition naturally represents) is larger than the valid data space D (what we actually want to search).

EXAMPLE 3.1 (False Positive from Invalid States): Consider searching for element 3 in a 2-bit representation:

- Data: $D = \{0, 1, 2, 2\}$
- Computational space: $S = \{0, 1, 2, 3\}$
- Validity: $V(0) = 1,\ V(1) = 1,\ V(2) = 1,\ V(3) = 0$ (3 is not in D)
- Target predicate from Grover's Algorithm: $T(3) = 1$ (we're searching for 3)

The problem: Standard Grover superposition in Equation (1) is:

$$|\psi\rangle = \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right) = \frac{1}{2}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle\right)$$

This superposition includes $|3\rangle$, which represents an invalid state ($3 \notin D$ or $V(3) = 0$). If the oracle marks states satisfying $T(x) = 1$, it will mark $|3\rangle$, causing Grover's algorithm to amplify and measure this state—producing a false positive result.

Critical observation: Both the initial superposition (Equation (1)) and the oracle, "$D[i]$ == 3?1:0", are "blind" to the validity function, V. The algorithm has no way to distinguish between computational states representing valid data, $V(i) = 1$, and those representing invalid states, $V(i) = 0$, where V is the Validity function in the above definition. This blindness to problem instances leads directly to the Grover Dilemma.

3.2. Type A vs Type B Problems

We will first classify structured search problems into two categories based on their Validity function, V, in Definition 3.1:

DEFINITION 3.2 (Type A Problems - Computational Space Exceeds Valid Space): A structured search problem is Type A, when $|S| > |D'|$, where D' = distinct(D).

For Type A problems, the computational space S (all possible bit strings) is larger than the valid data space D' and not all computational basis states represent valid problem instances. For example, searching for 3 from $D = \{0, 1, 2, 2\}$, where $|3\rangle$ is invalid.

DEFINITION 3.3 (Type B Problems - Natural Computational Encoding): A structured search problem is Type B, when $|S| = |D'|$, where D' = distinct(D).

For Type B problems, every computational basis state represents a valid problem instance, and all 2^n states in the computational space correspond to valid problem inputs. For examples: Subset Sum, SAT, Knapsack [42].

Table 1 summarizes the key distinctions between Type A and Type B problems.

Table 1. Type A vs Type B problems.

Problem type	*All states valid?*	*Superposition cost*	*Grover dilemma?*
Type A	No	$O(\lvert D\rvert)$ or $O(\lvert S\rvert)$ preprocessing	Yes
Type B	Yes	$O(n)$ Hadamard gates	No

Type A problems face the Grover Dilemma: constructing superposition over only valid states requires $O(|D|)$ or $O(|S|)$ classical preprocessing, eliminating quantum advantage.

Type B problems avoid the Grover Dilemma at the superposition construction stage. However, as we show in Section 5, these problems typically face Oracle Circularity—the oracle must identify which states in Equation (1) are solutions, which requires solving the problem itself. Thus, avoiding Grover Dilemma does not guarantee quantum advantage.

The distinction between Type A and Type B problems clarifies which barriers affect which problem classes, and demonstrates that the three barriers (Grover Dilemma for Type A, Setup Costs in Section 4 and Oracle Circularity in Section 5 for Type B) are independent obstacles that must each be overcome for quantum advantage to exist.

3.3. The Grover Dilemma: Three Problematic Options

When applying Grover's algorithm to structured search problems (Type A), we face a fundamental trilemma. The algorithm must choose one of three approaches, each of which is problematic.

Option 1. Complete Superposition Over All Computational States: Use the standard superposition (Equation 1) over all $N = 2^n$ computational states, ignoring validity function, V, in Definition 3.1:

Problem 1 - False Positives: When searching for a target that doesn't exist in the valid data D, the algorithm may return invalid states. In Example 3.1, searching for 3 in $D = \{0, 1, 2, 2\}$ would return $|3\rangle$ even though $3 \notin D$.

Problem 2 - If only some valid states exist among N computational states, there are many different ways to fix it, see below.

Consequence: The algorithm produces incorrect results—quantum speedup is meaningless, if the answers are wrong.

Option 2. Solution-Included Superposition: For example, all N states $|0\rangle$, $|1\rangle$, ..., $|N-1\rangle$ in Equation (1) represent valid problem instances:

Problem - Requires Knowing the Answer: If this is true, the decision answer always exists in $O(1)$ time—the quantum algorithm, $O\left(\sqrt{N}\right)$, becomes unnecessary. This approach is logically circular—it requires solving the decision problem before running the algorithm designed to solve the problem. It also has the problem of instance-independent (Section 2.4, 2.5).

Option 3: Valid-States Superposition with Classical Preprocessing: Restrict the superposition to valid states only, determined through classical preprocessing:

$$|\psi\rangle = \frac{1}{\sqrt{|D|}} \sum_{x=0}^{|D|-1} |x\rangle \tag{7}$$

Then apply Grover's algorithm to search among valid states for those satisfying $T(x) = 1$.

Problem - Classical Preprocessing Cost: To determine which states are valid, $V(x) = 1$, we must:

- Examine all N computational states or all m data items.
- Build a mapping from valid states to data elements.
- Construct the restricted superposition.

Three possible implementations:

Approach 3a: Classical Index Array

- For data $D = \{0, 1, 2, 2\}$, create classical array: index = $\{0, 1, 2, -1\}$.
- Oracle finds state $|3\rangle$, returns index [3] = −1 (not found).

- **Total cost: $O(N)$ classical + $O(\sqrt{m})$ quantum = $O(N)$.**

Approach 3b: Quantum RAM (QRAM) with Index Array

- Encode data with indices in superposition: $|\psi\rangle = (1/2)(|00\rangle|0\rangle + |01\rangle|0\rangle + |10\rangle|0\rangle + |11\rangle|0\rangle)$, where the second ket-vector, $|0\rangle$, indicates validity.
- Query to check validity: $|i\rangle|0\rangle \rightarrow |i\rangle|\text{valid}(i)\rangle$.
- **Total cost: Exceeds $O(N)$** [26] [27].

Approach 3c: Direct Valid Superposition

- Create superposition only over valid states (Equation 7).
- For $D = \{0, 1, 2\}$ (with no repeated numbers):

$$|\psi\rangle = \frac{1}{3}\left(|0\rangle + |1\rangle + |2\rangle\right) \tag{8}$$

- **Cost: $O(m)$ classical preprocessing** (where $m = |D|$).

For data with duplicates (Example 3.3), the situation becomes even more complex.

Consequence: All three implementations require $O(N)$ or $O(m)$ classical preprocessing, eliminating the $O\left(\sqrt{N}\right)$ quantum advantage.

EXAMPLE 3.3 (Repeated Elements): Consider $D = \{3, 3, 0, 0, 4, 2, 0, 0\}$, searching for 7, The restricted superposition must account for element frequencies:

$$|\psi\rangle = \frac{1}{\sqrt{8}}\left[\sqrt{2}\cdot|3\rangle + \sqrt{4}\cdot|0\rangle + |4\rangle + |2\rangle\right] \tag{9}$$

where coefficients reflect how many times each value appears in D.

Cost: $O(m)$ classical operations to analyze data structure, eliminating any quantum advantage over classical search with $O(m)$ complexity.

Summary of the Trilemma for Type A:

Option	*Approach*	*Problem*	*Consequence*
1	Complete superposition	False positives, wrong answers	**Correctness**
2	Solution-only superposition	Requires knowing solutions	**Triviality**
3	Valid-states superposition	Classical preprocessing $O(m)$ or $O(N)$	**Efficiency**

3.4. The Grover Dilemma Formalized

DEFINITION 3.4 (The Grover Dilemma - Type A): For structured search problems where the computational space S exceeds the valid data space D ($N > m$), Grover's algorithm must choose between:

1) **Correctness:** Restricting to valid states, which requires $O(m)$ or $O(N)$ classical preprocessing, eliminating quantum advantage.

2) **Efficiency:** Including all computational states, which produces false positives and incorrect results.

3) **Triviality:** Restricting to solution-known states only with all computational

states, which requires already knowing the solutions, making the algorithm unnecessary.

No intermediate option exists that simultaneously:

- Guarantees correct results for all inputs.
- Avoids requiring classical knowledge of data structure.
- Achieves better than $O(m)$ or $O(N)$ complexity.

3.5. Type B Problems: The Exception

While the Grover Dilemma affects most practical search problems, a special class of problems avoids this barrier entirely. These problems have the property that their natural computational encoding uses all possible computational basis states as valid problem instances.

DEFINITION 3.5 (Natural Computational Encoding): A problem has natural computational encoding when:

1) The search space can be represented using n qubits.

2) Every one of the 2^n computational basis in Equation (1) represents a valid problem instance.

3) The uniform superposition in Equation (1) requires only $O(n)$ operations to create.

4) No classical preprocessing is needed to identify or filter valid states.

Problems with this property avoid the Grover Dilemma's superposition construction cost.

EXAMPLE 3.4 Subset Sum Problem [42]: Given set $\{a_1, a_2, \cdots, a_n\}$ and target T, does a subset $S \subseteq \{a_1, \cdots, a_n\}$ exist where

$$\sum_{i \in S} a_i = T \tag{10}$$

Type B problems feature:

- Encoding: Use n qubits, one per element: qubit $i = |0\rangle \rightarrow a_i \notin S$ and Qubit $i = |1\rangle \rightarrow a_i \in S$.
- **Initial superposition construction Cost:** $O(n)$ (just n Hadamard gates).
- **No preprocessing needed:** Unlike Example 3.1 (searching $D = \{0, 1, 2, 2\}$ where state $|3\rangle$ is invalid), Subset Sum has no invalid states requiring classical filtering. Subset Sum avoids the Grover Dilemma entirely at the superposition construction stage.

Several important problem classes share this property: Boolean Satisfiability (SAT), Graph Coloring (with fixed encoding), Traveling Salesman (with certain encodings), and Knapsack Problem [42]. While these problems avoid the Grover Dilemma through natural computational encoding, this does **not** imply they achieve quantum advantage. Problems with all-valid-states encoding (Type B) typically face a different fundamental barrier: **Oracle Circularity**. The oracle must identify which states are solutions—but determining this is precisely the problem we're trying to solve. For Subset Sum, the oracle must know which subsets sum to target T; for SAT, which assignments satisfy the formula; for optimization problems, which configurations are optimal. Section 5 examines this Oracle Circularity problem in detail, showing it represents an independent barrier that eliminates quantum advantage even when Grover Dilemma and Setup Costs are avoided.

Example preview: Subset Sum avoids both Grover Dilemma ($O(n)$ superposition) and Setup Cost ($O(\text{poly}(n))$ oracle construction from formula), yet still provides no quantum advantage due to Oracle Circularity—the oracle cannot be properly specified without solving the problem.

3.6. Summary

The Grover Dilemma presents a fundamental barrier for many quantum search algorithms:

Type A Problems (Computational Space > Valid Space):

- Not all computational states represent valid problem instances.
- Constructing superposition over valid states requires $O(|D|)$ or $O(|S|)$ preprocessing.
- **Verdict:** Grover Dilemma eliminates quantum advantage.

Type B Problems (Natural Computational Encoding):

- All computational states represent valid problem instances.
- Initial superposition requires only O(n) operations.
- **Verdict:** Avoids Grover Dilemma, but faces other barriers (Sections 4 and 5).

The Grover Dilemma is the first of three independent barriers to quantum computational advantage. Even problems that avoid this barrier (Type B) typically face fundamental obstacles in oracle construction (Section 4) or oracle specification (Section 5). The complete picture emerges only after examining all three barriers.

4. The Quantum Setup Cost Dilemma

Section 3 examined the Grover Dilemma, which arises from the difficulty of constructing superpositions over only valid problem states. This section examines a related but broader problem: **setup costs** in quantum algorithm implementation. Even when superposition preparation appears efficient in Type B problems, practical implementation requires various forms of classical preprocessing which may eliminate quantum speedups. These costs include:

1) **Oracle construction:** Building quantum circuits which implement problem-specific functions.

2) **Data loading:** Transferring classical data into quantum-accessible memory.

3) **Superposition preparation:** Creating problem-specific quantum states beyond uniform superposition.

4) **Architecture overhead:** Error correction, connectivity constraints, and physical implementation costs.

The fundamental issue: Theoretical quantum algorithm analysis often treats oracles and prepared states as "free" resources, ignoring the classical computation required to construct them. When these costs are included, many claimed quantum advantages disappear.

4.1. The Setup Cost Dilemma

DEFINITION 4.1 (Quantum Setup Cost Dilemma): A quantum algorithm suffers from the Quantum Setup Cost Dilemma when the total classical preprocessing costs (oracle construction, data loading, superposition preparation, and other setup requirements) meet or exceed the complexity of the best classical algorithm for solving the problem directly.

Formally, let:

- C_{setup} = total classical preprocessing cost
- $C_{quantum}$ = quantum algorithm execution cost
- $C_{classical}$ = best classical algorithm cost for the same problem

The algorithm suffers from Setup Cost Dilemma if:

$$C_{\text{total}} = C_{\text{setup}} + C_{\text{quantum}} \geq C_{\text{classical}}$$

In this case, no genuine quantum computational advantage exists.

THEOREM 4.1 (Setup Cost Impossibility): Any quantum algorithm with $C_{setup} \geq C_{classical}$ cannot provide genuine computational advantage over classical approaches.

Proof: The total computational cost is:

$$C_{\text{total}} = C_{\text{setup}} + C_{\text{quantum}}$$

Since $C_{setup} \geq C_{classical}$ by hypothesis:

$$C_{\text{total}} = C_{\text{setup}} + C_{\text{quantum}} \geq C_{\text{classical}} + C_{\text{quantum}} > C_{\text{classical}}$$

The classical approach alone has lower complexity. Therefore, no quantum advantage exists.

Ubiquity of Setup Costs: The Setup Cost Dilemma affects any quantum algorithm that:

1) Requires problem-specific oracle construction through classical computation.

2) Needs to load classical data into quantum-accessible memory.

3) Must prepare non-uniform superposition states based on problem structure.

4) Competes against classical algorithms that can process data directly without quantum encoding overhead.

This includes many foundational quantum algorithms that claim exponential or quadratic speedups in theoretical analyses.

4.2. Case Study: Deutsch's Algorithm

Deutsch's algorithm [30] is often cited as providing the first demonstration of quantum computational advantage, claiming exponential speedup for a simple function evaluation problem.

Problem: Given a Boolean function $f:\{0,1\} \to \{0,1\}$, determine if the function, f, is constant ($f(0) = f(1)$) or balanced ($f(0) \neq f(1)$).

Classical Approach: Worst-case complexity: 2 function evaluations, which must evaluate $f(0)$ and $f(1)$ to distinguish constant from balanced.

Claimed Quantum Approach:

- Quantum complexity: $O(1)$ Oracle query
- Uses quantum superposition and interference to determine the answer in a single query;
- Claimed speedup: Exponential ($2^1 \rightarrow 1$)

Oracle Construction Analysis: To implement the quantum Oracle, O:

$$O|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle \tag{11}$$

we must:

- Classical evaluation: Determine $f(0)$ and $f(1)$ to construct the Oracle circuit.
- Circuit construction: Build quantum gates that implement the function mapping.

Realistic Complexity Analysis:

- Classical preprocessing: "$O(2)$" to evaluate $f(0)$ and $f(1)$
- Quantum execution: $O(1)$ Oracle query
- Total complexity: > 2
- Classical complexity: 2
- Actual quantum advantage: None

REMARK 4.1 (The Evaluation Paradox): To construct the oracle that tells us whether f is constant or balanced, we must evaluate $f(0)$ and $f(1)$—which immediately reveals whether f is constant or balanced! The "quantum advantage" exists only if we ignore the oracle construction cost and assume the oracle is provided for free.

REMARK 4.2 (When Setup Costs Don't Apply): If the function f is already given as a quantum circuit (rather than a black-box function requiring evaluation), then no construction cost is incurred. However, this fundamentally changes the problem specification—we are no longer comparing quantum vs classical evaluation of an arbitrary function, but rather assuming the function is already in quantum form. This scenario rarely occurs in practice.

4.3. Case Study: Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm [31] [33] extends Deutsch's approach to n-bit Boolean functions, with dramatically larger claimed speedups.

Problem Statement: Given $f:\{0,1\}^n \rightarrow \{0,1\}$, promised to be either constant or balanced, determine which.

- **Constant:** $f(x) = 0$ for all x, or $f(x) = 1$ for all x
- **Balanced:** $f(x) = 0$ for exactly half the inputs, $f(x) = 1$ for the other half

Classical Approach: Query f at $(2^{n-1} + 1)$ different inputs to guarantee distinguishing constant from balanced. Worst-case complexity is $O(\mathbf{2^n})$.

Claimed Quantum Approach:

- Prepare uniform superposition over all 2^n inputs
- Apply quantum oracle once
- Perform Hadamard transform and measure
- Claimed complexity: 1 oracle query

- Claimed speedup: $2^{n-1} \to 1$ (exponential!)

General Black-Box Function Setup Cost Analysis: To construct an oracle for arbitrary function $f:\{0,1\}^n \to \{0,1\}$, we must determine f's behavior:

- Evaluate $f(x)$ for all $x \in \{0, 1\}^n$ to know function behavior
- Only then can we construct quantum circuit implementing f
- Cost: 2^n function evaluations

Complete Complexity:

- Classical preprocessing: 2^n evaluations
- Quantum execution: 1 oracle query
- **Total:** $O(2^n)$
- Classical worst-case: $O(2^n)$
- Actual advantage: None

4.4. Deutsch-Jozsa Algorithm with Explicit Formula

Some functions can be specified compactly and have efficient quantum oracle implementations.

EXAMPLE 4.1 (Parity Function): Consider $f(x) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$ (returns 1 if odd number of bits are 1, else 0) [9] [31] [33]. This function is **balanced** (exactly half the 2^n inputs give 0, half give 1).

Quantum oracle construction from formula:

- Formula: $f(x) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$
- Quantum circuit: Chain of CNOT gates
- Construction: $O(n)$ gates
- **Oracle construction cost: $O(n)$**

Quantum algorithm:

- Oracle construction: $O(n)$
- Superposition preparation: $O(n)$
- Oracle query: $O(1)$
- Measurement: $O(1)$
- **Total: $O(n)$**

Classical algorithm from same formula with $O(n)$:

- result = 0
- for $i = 1$ to n
- result = result XOR x_i
- return result

Comparison:

- Quantum (from formula): $O(n)$
- Classical (from formula): $O(n)$
- No quantum advantage when formula is given!

The key insight: When the function has compact representation (like an explicit formula) that enables efficient quantum oracle construction, classical algorithms can also evaluate the function efficiently from the same representation.

The Artificial Promise Problem: The Deutsch-Jozsa problem includes a promise that f is either constant or balanced—no other possibilities exist. In practice, most functions are neither constant nor balanced, and we rarely have such guarantees. The promise itself requires prior knowledge about f.

Modified classical algorithm with promise:

1) Evaluate f at two inputs: x_1 and x_2
2) If $f(x_1) = f(x_2)$ → likely constant (check one more to be sure)
3) If $f(x_1) \neq f(x_2)$ → definitely balanced (promise guarantees only two options)
4) Cost: $O(1)$ evaluations (constant number)

With the promise, classical is also $O(1)$ in the average case!

Key Insight: Deutsch-Jozsa has exponential speedup only for functions with structure allowing efficient oracle construction. In these cases, classical algorithms are more than a match—they achieve similar or better complexity using the same structural information. For the general black-box case typically presented in theoretical analyses [31] [33], setup costs eliminate the advantage entirely.

REMARK 4.3 (Problem Specification Matters): The practical value of Deutsch-Jozsa depends critically on:

1) How the function f is specified (black box vs formula vs quantum circuit).
2) Whether f's structure allows efficient oracle synthesis.
3) What comparison baseline is used (naive classical vs optimal classical).

Most presentations omit these crucial details, leading to inflated claims of quantum advantage.

4.5. Case Study: Simon's Algorithm

Simon's algorithm [32] [34] is sometimes cited as an example of oracle-based quantum algorithms that avoid setup costs. However, careful analysis reveals that Simon's algorithm faces the Setup Cost Dilemma in the general case, with only narrow exceptions for specifically structured functions. $f:\{0,1\}^n \to \{0,1\}^n$

Simon's Problem: Given a function, with a hidden period s satisfying $f(x) = f(x \oplus s)$, find s.

Quantum claim: $O(n)$ queries to the oracle for f, providing exponential speedup over classical $O(^{n/2})$ query complexity.

The oracle requirement: Must implement $O_f|x\rangle|0\rangle \to |x\rangle|f(x)\rangle$ for all $x \in \{0, 1\}^n$.

Arbitrary Black-Box Functions specification requires:

- For each of 2^n possible inputs x, specify output $f(x)$
- Total specification: 2^n output values, each requiring n bits
- Information content: $O(n \cdot 2^n)$ bits

Oracle construction from specification:

- Must encode the mapping $x \to f(x)$ for all 2^n inputs
- Classical lookup table: $O(2^n)$ memory
- Quantum circuit encoding: $O(2^n)$ gates in general case
- Construction cost: $O(2^n)$

Verdict: For arbitrary black-box functions, Setup Cost Dilemma applies. Oracle construction requires $O(2^n)$ resources, eliminating any quantum query advantage.

4.6. Simon's Algorithm for Functions with Compact Representation

Functions can be specified with $O(\text{poly}(n))$ parameters instead of $O(2^n)$.

EXAMPLE 4.2 (Linear Function): Consider $f(x) = Ax \pmod 2$, where A is an $(n-1) \times n$ binary matrix [9] [32] [34]. A specific instance ($n = 3$) is:

$$f(x) = Ax(\text{mod}\,2) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \text{mod}\,2 \quad (12)$$

Equation (12) is also: $f(x_1, x_2, x_3) = (x_1 \oplus x_3, x_2 \oplus x_3)$. The hidden period is: $s = 111$: $f(000) = f(111) = (0, 0)$, $f(001) = f(110) = (1, 1)$, $f(010) = f(101) = (0, 1)$, and $f(011) = f(100) = (1, 0)$. The period, s, is in the kernel of A, which can be obtained by solving

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} = 0 \quad (13)$$

System of equations:

$$s_1 \oplus s_3 = 0 \rightarrow s_1 = s_3$$

$$s_2 \oplus s_3 = 0 \rightarrow s_2 = s_3$$

Using Gaussian elimination, the classical complexity is $O(n^3)$.

EXAMPLE 4.3 Let $n = 4$ bits; $f(x) = Ax \pmod 2$, where A is a 3×4 binary matrix, $A = \{[1\ 1\ 0\ 0], [0\ 1\ 1\ 0], [0\ 0\ 1\ 1]\}$; then $s = 1011$ (in the kernel of A).

The oracle circuit for $f(x) = A \cdot x$ is hardwired for this particular matrix. This is not a general-purpose quantum advantage—it's a solution for a single, pre-specified function instance.

Quantum approach:

- Build oracle: $O(n^2)$
- Query oracle: $O(n)$ times
- Solve linear system: $O(n^3)$
- Total quantum: $\boldsymbol{O(n^3)}$

Classical algorithms for structured functions: $O(n^3)$.

No quantum advantage: When function structure is explicitly known (enabling efficient oracle construction), classical algorithms can also exploit that structure directly. Table 2 illustrates this fundamental trade-off for Simon's algorithm.

Table 2. The dilemma persists: Simon's algorithm faces a fundamental trade-off.

Function type	*Oracle construction*	*Classical complexity*	*Quantum advantage?*
Arbitrary black-box	$O(2^n)$ - expensive	$O(^{n/2})$ queries	No - Setup cost kills it
Structured (known)	$O(\text{poly}(n))$ - cheap	$O(\text{poly}(n))$ direct solution	No - Classical also efficient

In summary:

1) Simon's algorithm suffers from Setup Cost Dilemma in the general case.

2) There are particular cases where an oracle can be constructed for a single, specific function in polynomial time, allowing that instance to escape the exponential Setup Cost. However, these special cases have equally efficient classical solutions with the same complexity.

3) These exceptions come at the cost of working for only one pre-specified function, not arbitrary functions, and provides no advantage over classical algorithms that can also exploit the same known structure.

4.7. Restricted Problem Class

Approach: Limit applicability to special cases where C_{setup} is small.

Strategy:

- Focus on problems where oracles can be efficiently constructed.
- Explicitly specify structure required for quantum advantage.
- Accept narrow applicability in exchange for avoiding general setup costs

Examples: Functions given as explicit formulas:

$$f(x) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$$

$$f(x) = Ax \bmod 2, \quad A = \{[1\,0\,1],[0\,1\,1]\}, \quad s = 111$$

Problem: When problems have sufficient structure to enable efficient quantum oracle construction, classical algorithms can typically exploit the same structure efficiently. The algorithms suffer from Setup Cost Dilemma, when $C_{\text{setup}} \geq C_{\text{classical}}$, so the reference point is the best classical algorithms.

Result: Classical algorithms are more than a match for these cases.

Why this occurs:

- Efficient oracle construction requires compact problem representation.
- Compact representation enables classical algorithms to work directly with structure.
- Both quantum and classical end up with similar complexity.
- Quantum advantage disappears when function structure is explicit.

Examples:

- Parity function: Both quantum and classical are $O(n)$.
- Linear functions (Simon): Classical can solve As = 0 directly in $O(n^3)$.
- Explicit Boolean formulas: Classical can evaluate efficiently.

4.8. True Black-Box Functions

DEFINITION 4.2 (True Black-Box Function): A function $f:\{0,1\}^n \to \{0,1\}$ is a true black-box if it has no compact representation and requires $O(2^n)$ bits to specify.

THEOREM (No Quantum Advantage for True Black-Box Functions): Oracle-based quantum search provides no computational advantage for true black-box functions.

Proof:

- Setup cost: $O(2^n)$ to specify which of 2^n inputs map to which outputs (symmetric for classical and quantum).
- Classical total: $O(2^n)$ setup + $O(2^n)$ search = $O(2^n)$.
- Quantum total: $O(2^n)$ setup + $O(2^{n/2})$ search = $O(2^n)$.
- Setup cost dominates; no quantum advantage.

Example: Specifying which of 2^n database items is "marked" requires $O(2^n)$ information. Both classical and quantum face identical $O(2^n)$ setup cost, eliminating quantum's $O(2^{n/2})$ query advantage.

4.9. The Setup Cost Dilemma Formalized

The table of Setup Cost Analysis of Foundational Quantum Algorithms can be found in Section 7.

DEFINITION 4.3 (The Setup Cost Dilemma): Any quantum algorithm claiming computational advantage must choose between:

Option 1: Complete Cost Accounting

- Often $C_{total} \geq C_{classical}$, eliminating claimed advantage.

Option 2: Incomplete Cost Accounting

- Use only $C_{quantum}$ and ignore C_{setup}, missing real cost C_{setup}.

Option 3: Restricted Problem Class

- Limit applicability to special cases where C_{setup} is small.
- Reduced practical applicability.
- Classical algorithms are more than a match for these cases.

The Grover Dilemma and Setup Cost Dilemma are independent barriers that can occur separately or together. While this section examined practical setup costs that often eliminate quantum advantage, the next section explores an even more fundamental problem: **Oracle Circularity**.

5. The Oracle Circularity Problem

Sections 3 and 4 examined two independent barriers to quantum computational advantage, this section examines a third, more fundamental barrier: **Oracle Circularity**. For certain problem classes—particularly optimization and learning problems—oracle construction is not merely expensive (Setup Cost issue), but logically circular. The oracle must encode the solution to the problem, but finding that solution is precisely what the algorithm claims to do.

Key distinction from Section 4:

- **Setup Cost Dilemma:** Oracle construction is expensive but possible (e.g., evaluating f on all 2^n inputs costs 2^n, which is expensive).
- **Oracle Circularity:** Oracle construction requires solving the target problem (e.g., oracle must identify optimal policy π^*, but finding π^* IS the problem).

While setup costs are a practical/economic barrier, oracle circularity is a logical/definitional barrier that reveals a fundamental flaw in how certain quantum algorithms are formulated.

5.1. Oracle Circularity Defined

DEFINITION 5.1 (Oracle Circularity): A quantum algorithm exhibits Oracle Circularity when constructing the required oracle necessitates solving the same problem (or a problem of equivalent difficulty) that the quantum algorithm claims to solve.

Formally, let:

- P = the problem the quantum algorithm claims to solve
- O = the oracle required by the quantum algorithm
- $C(P)$ = complexity of solving problem P classically
- $C(O)$ = complexity of constructing oracle O

Oracle Circularity exists when:

$$C(O) \geq C(P)$$

where the equality or near-equality arises because constructing O requires solving P.

The circular dependency:

1) To use the quantum algorithm to solve P, we need oracle O.
2) To construct oracle O, we must solve problem P.
3) But solving P is what the quantum algorithm claims to do!
4) **Result:** Circular reasoning that eliminates quantum advantage.

THEOREM 5.1 (Oracle Circularity and Impossibility): Any quantum algorithm with Oracle Circularity cannot provide computational advantage over directly solving the problem classically.

Proof:

- Total quantum cost: $C_{\text{total}} = C(O) + C_{\text{quantum}}$.
- By definition of Oracle Circularity: $C(O) \geq C(P)$.
- Therefore: $C_{\text{total}} \geq C(P) + C_{\text{quantum}} > C(P)$.
- Classical direct solution: $C(P)$.
- **Conclusion:** Quantum approach has higher total cost.

5.2. Subset Sum

The Subset Sum problem (Section 3.5) provides an ideal case study because it is a **Type B problem** that avoids the Grover Dilemma, allowing us to isolate Oracle Circularity as the primary barrier to quantum advantage.

Classical Approaches:

- Exhaustive search: $O(n \cdot 2^n)$ to check all subsets
- Dynamic programming: $O(n \cdot T)$ if T is polynomial in n
- Best classical (general case): $O(n \cdot 2^n)$

Claimed Quantum Approach:

- Grover's algorithm on 2^n subsets
- Claimed complexity: $O\left(\sqrt{2^n}\right) = O\left(2^{n/2}\right)$
- Each iteration: Oracle evaluation $O(n)$
- **Total claimed:** $O(n \cdot 2^{n/2})$

- **Claimed speedup:** $O(2^n) \to O(2^{n/2})$ (quadratic)

Barrier 1. Grover Dilemma (from Section 3): Subset Sum is Type B, thus avoiding Grover Dilemma.

EXAMPLE 5.1 Oracle for Subset Sum: $\{a_1 = 1, a_2 = 2, a_3 = 1, T = 3\}$, the solutions are $|110\rangle$ (subset {1, 2}) and $|001\rangle$ (subset {3}). The oracle must mark these specific states. Oracle requirement is:

$$f(S) = (S == \{S_1, S_2, \cdots, S_K\} ? 1 : 0) \tag{14}$$

and in this example

$$f(S) = (S == \{110, 001\} ? 1 : 0) \tag{15}$$

If we know the solution, $\{S_1, S_2, \ldots, S_K$, the oracle can be constructed as a quantum circuit directly in $O(\text{poly}(n))$ time from the solution.

Barrier 2. Setup Cost Dilemma (from Section 4):

Option A: Solve First, Then Build Oracle

Step 1: Solve Subset Sum classically

- For our example: Identify that $|110\rangle$ and $|001\rangle$ are solutions
- **Cost:** $O(n \cdot 2^n)$

Step 2: Build oracle circuit that marks these specific states

- Oracle function: $f(S) = (S == 110 \text{ or } S == 011? \ 1 : 0)$
- Circuit compares input $|S\rangle$ against $|110\rangle$ and $|011\rangle$
- **Cost:** $O(\text{poly}(n))$ to build comparison circuits

Total cost: $O(n \cdot 2^n) + O(\text{poly}(n)) \approx O(n \cdot 2^n)$

Analysis: Oracle construction requires $O(\text{poly}(n))$ once solutions are known, BUT finding solutions requires $O(n \cdot 2^n)$. The setup cost is dominated by solving the problem, not by circuit construction. This is Oracle Circularity (Barrier 3), not Setup Cost Dilemma.

Option B: Build Evaluation Circuit Without Solutions: Build a circuit for any input that evaluates

$$f(S) = \left(\sum_{a_i \in S} a_i == T ? 1 : 0 \right) \tag{16}$$

Circuit construction from $\{a_1, \ldots, a_n, T\}$:

- Quantum adder: computes $(\text{sum} = s_1 \cdot a_1 + s_2 \cdot a_2 + \ldots + s_n \cdot a_n) == T$
- **Cost:** $O(\text{poly}(n))$ to build circuit structure

This looks promising! We built a circuit in $O(\text{poly}(n))$ without knowing solutions.

The problem: This circuit doesn't know which states to mark! For our example. what this circuit does:

- Input: $|S\rangle = |s_1 s_2 s_3\rangle$ (in superposition over all 2^n states)
- Determine if, $(s_1 \cdot a_1 + s_2 \cdot a_2 + s_3 \cdot a_3) == T$, for each $|S\rangle$ in superposition

This determination—evaluating which of the 2^n subsets sum to T—IS solving the Subset Sum problem! The circuit CAN be built in $O(\text{poly}(n))$, but it doesn't yet "know" which states are solutions. To use this circuit as a Grover oracle requires executing it to determine which states to mark, but executing this circuit performs the full computational work of solving Subset Sum—defeating the purpose of the quantum algorithm.

Verdict on Setup Cost: The circuit construction itself is $O(\text{poly}(n))$ in both approaches, so Subset Sum does NOT suffer from Setup Cost Dilemma in the sense that the circuit structure can be built efficiently. However, this is misleading because neither approach provides a working oracle without solving the problem—that's Oracle Circularity (Barrier 3).

Barrier 3: Oracle Circularity

Option A. Oracle Circularity via "Solutions Must Be Known": To build the solution-marking oracle, we must know which states are solutions: $\{|110\rangle, |011\rangle\}$ in our example.

The circularity:

1) We want to find which subsets sum to T (solve Subset Sum).

2) To run Grover, we need an oracle that marks solution states.

3) To build this oracle, we must know which states are solutions.

4)To know which states are solutions, we must solve Subset Sum.

5) We need to solve the problem to build the oracle to solve the problem!

This is pure circularity: We cannot build the oracle without solving what the algorithm claims to solve.

Cost: $O(n \cdot 2^n)$ to find solutions + $O(\text{poly}(n))$ to build oracle = $O(n \cdot 2^n)$ total

Verdict: Oracle Circularity eliminates any quantum advantage. We must solve the problem before we can run the quantum algorithm.

Option B: Oracle Circularity via "Execution Requires Solving": In the evaluation circuit approach, we build oracle circuit in Equation (16). Using this circuit in Grover's algorithm requires it to identify which states are solutions, but making this identification IS Subset Sum. For each $|s_1 s_2 s_3\rangle$ in superposition, check if $s_1 \cdot 1 + s_2 \cdot 2 + s_3 \cdot 3 == 3$.

Computational work: Over all 2^n states in superposition, this oracle evaluates all subsets.

The fundamental issue is that the circuit evaluates sums to determine which subsets satisfy T. This evaluation—whether done in precomputation (Option A) or during execution (Option B)—IS the Subset Sum problem.

Verdict: Option B also suffers from Oracle Circularity. The circuit can be built efficiently, but using it requires solving Subset Sum through evaluation work. The Two Faces of Oracle Circularity are:

Conceptual circularity (Option A):

- Need solutions to build oracle.
- Need oracle to find solutions.
- Circular dependency in specification.

Computational circularity (Option B):

- Can build evaluation circuit.
- But circuit must solve the problem when executed.
- Circular dependency in execution.

Key Insight: Subset Sum demonstrates that even when:

- Grover Dilemma is avoided (Type B encoding).
- Circuit construction is cheap ($O(\text{poly}(n))$).

Oracle Circularity remains fatal because:

- Solution-marking oracle requires solving the problem (Option A).
- Evaluation oracle requires solving the problem during execution (Option B).

5.3. Reinforcement Learning for Stock Trading

We now examine a learning problem [35]-[37] where Oracle Circularity is even more severe, and where the problem has both an exponential search space AND an efficient polynomial-time solution.

Why this example is valuable: The existence of an $O(T)$ dynamic programming solution provides ground truth for verifying quantum algorithm claims, while the exponential search space (3^T) is what quantum algorithms attempt to exploit. This is an exceptionally good test ground for quantum's exhaustive search approach with efficient checking.

Stock Trading RL Problem [35]-[37]**:** Consider an automated trading agent that must decide actions based on recent stock price history, Given:

- **Price history:** $D = [p_0, p_1, \cdots, p_{T+N-1}]$
- **Price windows:** $t_i = [p_i, p_{i+1}, \cdots, p_{i+N-1}]$ for $i = 0, \cdots, T-1$
- **Actions:** A = {Buy, Sell, Hold}
- **State:** $h \in \{0, 1\}$ (holding 0 or 1 share)
- **Policy** π**:** Decision rule mapping $a_i = \pi(t_i)$ for $i = 0, \cdots, T-1$
- **Objective:** Maximize total profit over trading period

Optimal policy π^*: *Achieves highest expected profit*:

$$\pi^* = \arg\max_\pi \{E(D, \pi)\} \tag{17}$$

where $E(D, \pi)$ is total profit following policy π on data D from one share of stock.

Solution 1. (Classical Solutions). Dynamic Programming-$O(T)$: Using Bellman recursion [34]-[37] with value function, $V_t(h)$ = max expected profit from time t onward, given holdings $h \in \{0, \}$:

$$V_t(h) = \max_{\{a_t\}} \{r_t(h, a_t) + V_{\{t+1\}}(h')\} \tag{18}$$

where:

- $r_t(h, a_t)$ = immediate profit from action a_t given holdings h
- h' = new holdings after action a_t

Backward recursion:

- Initialize: $V_T(h) = 0$ (terminal condition)
- For $t = T - 1$ down to 0: Compute $V_t(h)$ for state: $h \in \{0,1\}$ and action: $a_t \in$ {Buy, Sell, Hold}
- Optimal policy: $\pi^*(t_i) = \arg\max_a \{r_i(h, a) + V_{i+1}(h')\}$

Complexity: $O(T \times S \times A) = O(T \times 2 \times 3) = O(T)$, where

- T time steps

- $S = 2$ states (holding 0 or 1 share)
- $A = 3$ actions

This provides ground truth for optimal policy π in polynomial time!

Solution 2. Exhaustive Search-$O(T{\cdot}3^T)$: Search all 3^T possible action sequences $\{a_0, a_1, \cdots, a_{T-1}\}$ [34]-[37]:

- Each action: 3 choices (Buy, Sell, Hold)
- T time steps
- Total sequences: 3^T
- Evaluate each: $O(T)$ to simulate and compute profit

Complexity: $O(T \times 3^T)$

On Comparing Quantum vs Classical Exhaustive Search:

To ensure a fair comparison, we compare quantum exhaustive search against classical exhaustive search over the same solution space. Both approaches search through all 3^T possible action sequences:

- Classical exhaustive search: $O(T \times 3^T)$ - enumerate and evaluate all action sequences.
- Quantum exhaustive search (Grover): $O(T \times 3^{T/2})$ - amplitude amplification over all action sequences.
- Quantum advantage (in theory): Quadratic speedup from $O(T \times 3^T)$ to $O(T \times 3^{T/2})$.

This is a conceptually equivalent comparison—both algorithms perform exhaustive search over the same 3^T solution space, with quantum providing the theoretically expected quadratic speedup.

Why we use this comparison: Exhaustive search represents a natural baseline for evaluating quantum search algorithms. Grover's algorithm is designed for unstructured search over solution spaces, and comparing it against classical exhaustive search over the same space provides an apples-to-apples comparison of the search paradigm itself.

The existence of alternative classical solutions: We note that this particular problem also admits an $O(T)$ Dynamic Programming solution exploiting sequential structure. However, this fact is not used in our analysis of quantum advantage. The existence of DP does not invalidate the quantum-classical exhaustive search comparison—it simply demonstrates that for this problem class, structured approaches exist alongside search-based approaches.

Our focus: The analysis that follows focuses on whether quantum exhaustive search provides genuine advantage over classical exhaustive search when oracle construction costs are properly accounted for. The comparison is fair: both use the same search paradigm (exhaustive enumeration), with quantum claiming quadratic speedup. The question is whether this theoretical speedup translates to practical advantage when Oracle Circularity (analyzed below) is considered.

Quantum RL using Grover's search is given in below:

1) Initialization: Create superposition over 3^T policies: Equation (20)

2) Grover Iteration (repeat $O\left(\sqrt{3^T}\right)$ times):

a) Oracle: Equation (21)

b) Amplitude amplification

3) Measurement: Obtain optimal policy π^*

Claimed complexity: $O\left(\sqrt{3^T}\right)$ iterations × $O(T)$ per oracle = $O(T{\cdot}3^{T/2})$

Claimed speedup: $O(T{\cdot}3^T) \to O(T{\cdot}3^{T/2})$ (quadratic in search space)

Encoding: Use T triplets (base-3 digits), one per time step:

- Action $a_i \in \{\text{Hold}, \text{Buy}, \text{Sell}\}$
- All 3^T computational states represent valid action sequences

Initial Superposition:

$$|\psi\rangle = \otimes_{i=0}^{T-1} \frac{1}{\sqrt{3}}\left[|\text{Hold}_i\rangle + |\text{Buy}_i\rangle + |\text{Sell}_i\rangle\right] \tag{19}$$

Expanding:

$$|\psi\rangle = \frac{1}{\sqrt{3^T}} \sum_{\pi \in \Pi} |\pi\rangle \tag{20}$$

where each $|\pi\rangle = |a_0, a_1, \cdots, a_{T-1}\rangle$ represents one of 3^T action sequences.

The Three Barriers:

Barrier 1. Grover Dilemma (from Section 3): This is NOT a barrier because the similarity to the Subset Sum problem.

Barrier 2. Setup Cost Dilemma (from Section 4): This is NOT a barrier because the similarity to the Subset Sum problem.

Barrier 3. Oracle Circularity (Section 5): Even though Trading RL avoids Barriers 1 and 2, it faces severe Oracle Circularity.

The oracle implements

$$O_f |\pi\rangle = (-1)^{f(\pi)} |\pi\rangle \tag{21}$$

where $f(\pi)$ marks the optimal policy.

Option 1. Mark Optimal Policy: Function is:

$$f(\pi) = \left(\pi == \pi^* \,?\, 1 : 0\right) \tag{22}$$

The circularity:

1) To run Grover, we need an oracle that marks π^*.

2) To build this oracle, we must know which policy is π^*.

3) To know which policy is π^*, we must solve the RL problem.

4) We need to solve the problem to build the oracle to solve the problem!

This is exactly Option A from Subset Sum: Must solve problem first, then build an oracle.

Option 2. Threshold Profit: Function is:

$$f(\pi, \tau) = \left(E(D, \pi) > \tau \,?\, 1 : 0\right) \tag{23}$$

for some threshold τ.

The circularity:

- **Question:** What should τ be?
- **Answer:** Should be slightly below optimal profit $E(D, \pi^*)$.

- **Problem:** We don't know $E(D, \pi^*)$ without solving the problem!

If τ too low:

- Many suboptimal policies marked.
- Grover amplifies mixture of good and bad policies.
- Measurement returns suboptimal policy with high probability.
- Grover Dilemma-style issue (multiple marked states).

If τ too high:

- Optimal policy not marked ($E(D, \pi^*) < \tau$).
- Grover amplifies nothing.
- Algorithm fails to find solution.

Setting correct τ requires knowing optimal profit, which requires solving the problem → Oracle Circularity; otherwise, it is an approximation.

Option 3. Evaluate and Compare: Oracle evaluates profit $E(D, \pi)$ for all policies in superposition, marks the maximum. The Function is:

$$f(\pi) = \left(E(D,\pi) == \max\left\{E(D,\pi') \text{ for all } \pi'\right\}?1:0\right) \tag{24}$$

The circularity:

- To determine which policy has maximum profit, oracle must:
- Evaluate $E(D, \pi)$ for all 3^T policies
- Find the maximum
- Mark policies achieving maximum

This is exactly Option B from Subset Sum: Evaluation circuit requires solving the problem during execution.

Option 4: Why DP Doesn't Help Quantum Approach

Tempting argument: "Use DP to find π^*, then build oracle, then..., wait, why do we need quantum? The fatal flaw is that DP already solved the problem.

Verdict: Oracle Circularity applies critically. Whether through specification (must know π^*) or execution (must evaluate all policies), the oracle cannot be properly constructed or used without solving the RL problem. Table 3 summarizes how each of the three barriers applies to the trading RL problem.

Table 3. Analysis of three fundamental barriers for reinforcement learning in stock trading.

Barrier	*Applies to trading RL?*	*Reason*	*Severity*
Grover dilemma (Section 3)	**NO**	Type B: All 3^T sequences valid, $O(T)$ superposition	N/A
Setup cost (Section 4)	**NO**	Data loading $O(T)$, circuit construction $O(\text{poly}(T))$	N/A
Oracle circularity (Section 5)	**YES**	Oracle must know π^* or evaluate all 3^T policies	Critical

Final verdict: Trading RL provides no quantum computational advantage. Oracle Circularity prevents proper oracle construction, and the existence of $O(T)$ DP solution makes quantum's $O(T \cdot 3^{T/2})$ approach exponentially inferior.

5.4. Universality of the Analysis: Beyond the Specific Example

The stock trading reinforcement learning example in Section 5.3 serves as a concrete case study, but the analysis and conclusions are universal and apply to any oracle-based quantum search problem. This section clarifies the generality of our findings.

The Universal Comparison Framework: Section 5.3 establishes a comparison based on:

1) Classical exhaustive search over solution space of size N: Complexity $O(N)$.

2) Quantum exhaustive search (Grover) over the same solution space: Complexity $O\left(\sqrt{N}\right)$.

3) Oracle Circularity analysis: Does oracle construction require solving the problem?

This comparison framework is completely general and applies to any search problem with solution space N, regardless of:

- The specific problem domain (trading, optimization, learning, satisfiability, etc.).
- Whether alternative classical algorithms exist.
- The structure or properties of the solution space.

Universality Principle 1. Fair Exhaustive Search Comparison: The comparison in Section 5.3 is between quantum and classical exhaustive search over identical solution spaces:

- Trading RL: Search space = 3^T action sequences.
- Classical exhaustive: $O(T \times 3^T)$.
- Quantum Grover: $O(T \times 3^{T/2})$.
- Comparison basis: Both enumerate the same 3^T sequences.

This comparison generalizes immediately to any search problem:

- Subset Sum: Search space = 2^n subsets.
- SAT: Search space = 2^n truth assignments.
- Graph Coloring: Search space = k^V colorings.
- General search: Search space = N configurations.

The analysis method is identical: Compare classical $O(N)$ exhaustive search against quantum $O\left(\sqrt{N}\right)$ exhaustive search, then determine whether Oracle Circularity eliminates the apparent quantum advantage.

Universality Principle 2. Oracle Circularity is Problem-Independent: The Oracle Circularity analysis in Section 5.3 asks: "Can the oracle be constructed/specified without solving the problem?" This question applies universally to all oracle-based search algorithms:

- **Trading RL:** Can we mark optimal policies without solving for optimal policies? No → Oracle Circularity.

- **Subset Sum:** Can we mark subsets summing to T without identifying which subsets sum to T? No → Oracle Circularity.
- **SAT:** Can we mark satisfying assignments without solving SAT? No → Oracle Circularity.
- **General optimization:** Can we mark optimal solutions without finding optimal solutions? No → Oracle Circularity.

The logical structure is identical across all these problems. Oracle Circularity is a universal barrier affecting any problem where the oracle must identify solutions that are the target of the search.

Universality Principle 3. Alternative Classical Solutions Don't Invalidate the Analysis: The trading RL problem admits an $O(T)$ Dynamic Programming solution. This fact is not used in the Section 5.3 analysis. The existence of DP does not:

- Invalidate the quantum-classical exhaustive search comparison.
- Make the Oracle Circularity analysis specific to this problem.
- Prevent generalization to other problems.

Why this matters: Some problems have efficient classical structured algorithms (like DP for trading RL), while others do not (like general NP-complete problems). **Oracle Circularity applies universally to both categories:**

- **Category A (Efficient classical solution exists):** Example: Trading RL with $O(T)$ DP.
- **Category B (No known efficient classical solution):** Example: General SAT, Subset Sum.

The key insight: Oracle Circularity eliminates quantum advantage regardless of whether efficient classical alternatives exist. The existence of DP for trading RL is mentioned for completeness but is not part of the Oracle Circularity argument.

Universality Principle 4. The RL Example is Representative, Not Special: The trading RL example could be replaced by any search problem:

Alternative Example 1 (Subset Sum):

- Search space: 2^n subsets
- Classical exhaustive: $O(n \times 2^n)$
- Quantum Grover: $O(n \times 2^{n/2})$
- Oracle Circularity: Oracle must identify which subsets sum to T
- Analysis: Identical logical structure to Section 5.3
- **Result:** No quantum advantage due to Oracle Circularity

Alternative Example 2 (SAT):

- Search space: 2^n truth assignments
- Classical exhaustive: $O(m \times 2^n)$ (m clauses)
- Quantum Grover: $O(m \times 2^{n/2})$
- Oracle Circularity: Oracle must identify satisfying assignments
- Analysis: Identical logical structure to Section 5.3
- **Result:** No quantum advantage due to Oracle Circularity

Alternative Example 3 (Traveling Salesman):

- Search space: n! tours

- Classical exhaustive: $O(n \times n!)$
- Quantum Grover: $O\left(n \times \sqrt{n!}\right)$
- Oracle Circularity: Oracle must identify optimal or near-optimal tours
- Analysis: Identical logical structure to Section 5.3
- **Result:** No quantum advantage due to Oracle Circularity

The pattern: The analysis structure is **universal**:

1) Define search space N
2) Compare classical $O(N)$ vs quantum $O\left(\sqrt{N}\right)$ exhaustive search
3) Analyze whether oracle can be constructed without solving the problem
4) If Oracle Circularity applies → no quantum advantage
5) If efficient classical structured algorithm exists → additional barrier

Universality Principle 5. Generalization to Complex Environments: Even when classical structured solutions become intractable (large state spaces, partial observability, unknown dynamics), **Oracle Circularity persists**:

Complex RL (no tractable DP):

- Oracle must still mark optimal policies
- Finding optimal policies still requires solving the RL problem
- Oracle Circularity: $C(\text{oracle}) \geq C(\text{problem})$
- **Modern classical methods** (function approximation, policy gradients) exploit structure through generalization
- Quantum exhaustive search over $|S|^T \times |A|^T$ space cannot exploit this structure
- **Verdict:** No quantum advantage

High-dimensional optimization:

- Oracle must mark optimal or near-optimal configurations
- Identifying these requires solving the optimization problem
- Oracle Circularity applies
- Classical methods exploit gradient information, smoothness, locality
- Quantum exhaustive search cannot exploit this structure
- **Verdict:** No quantum advantage

The universal pattern: Oracle Circularity is independent of problem complexity and independent of whether tractable classical solutions exist. It applies whenever the oracle must encode knowledge of solutions that are the target of the search.

Universal Applicability: The analysis in Section 5.3 establishes a universal framework for evaluating oracle-based quantum search:

- **Comparison basis:** Exhaustive search (quantum vs classical) over identical solution spaces—applicable to any search problem.
- **Oracle Circularity test:** Does oracle construction require solving the target problem?—applicable universally.
- **Independence from alternatives:** Analysis does not depend on whether efficient classical structured algorithms exist.
- **Logical structure:** Same reasoning applies to trading RL, Subset Sum, SAT, TSP, and all optimization/learning problems.

- **Scalability:** Applies to simple problems (small state spaces) and complex problems (large state spaces, partial observability).

The trading RL example in Section 5.3 is not a special case—it is a representative example demonstrating barriers that apply universally to oracle-based quantum search for optimization and learning problems. The existence of an O(T) DP solution for this particular example is noted but not used in the core Oracle Circularity analysis. The results of Section 5.3 generalize to all problems where oracles must encode solution knowledge, regardless of problem domain, complexity, or existence of alternative classical approaches.

5.5. NP-Complete Problems Summary

Section 5.4 established that Oracle Circularity applies universally to optimization and learning problems where the oracle must encode solution knowledge. This pattern extends systematically to NP-complete problems [42]. **Table 4** summarizes how Oracle Circularity affects major NP-complete problem classes.

Table 4. Oracle circularity in Np-complete problems.

Problem	*Search space*	*Oracle must identify*	*Construction requires*	*Quantum advantage*
Subset sum	2^n subsets	Subsets summing to T	Solving Subset Sum	**Eliminated**
SAT	2^n assignments	Satisfying assignments	Solving SAT	**Eliminated**
Graph 3-coloring	3^V colorings	Valid 3-colorings	Solving coloring	**Eliminated**
Traveling salesman	$n!$ tours	Shortest tour	Solving TSP	**Eliminated**
Knapsack	2^n item sets	Optimal item sets	Solving knapsack	**Eliminated**

Universal pattern: For all NP-complete problems, the oracle must mark optimal or satisfying solutions, but identifying these solutions requires solving the NP-complete problem itself. Oracle Circularity therefore eliminates quantum computational advantage across the entire class of NP-complete problems when using oracle-based quantum search algorithms.

COROLLARY 5.1 (NP-Complete Universality): Oracle Circularity eliminates quantum computational advantage for the entire class of NP-complete problems when using oracle-based quantum search algorithms.

Proof: NP-complete problems are polynomial-time reducible to each other [42]. If one NP-complete problem suffers from Oracle Circularity (as demonstrated for Subset Sum in Section 5.2), and all NP-complete problems require oracles that identify solutions to NP-complete predicates, then Oracle Circularity applies to all.

5.6. Oracle Impossibility

Oracle Circularity becomes Oracle Impossibility when the computational cost makes oracle construction not merely circular but practically infeasible.

DEFINITION 5.2 (Oracle Impossibility): An algorithm faces Oracle Impossibility when it exhibits:

1) **Oracle Circularity:** Oracle construction requires solving the target problem.

2) **Computational Infeasibility:** The required computation is categorically impossible due to excessive computational cost.

The distinction is one of scale:

- **Oracle Circularity:** The logical problem that oracle construction requires solving the target problem. This applies regardless of problem size.
- **Oracle Impossibility:** When the scale makes this circular requirement not just expensive, but computationally infeasible or physically impossible.

The fundamental issue: Whether the problem size makes oracle construction merely expensive (Circularity) or completely infeasible (Impossibility), the quantum algorithm provides no advantage. Both represent the same circular dependency problem at different scales.

Relationship to other barriers:

- **Grover Dilemma (Section 3):** Superposition construction cost.
- **Setup Cost Dilemma (Section 4):** Oracle construction and data loading costs.
- **Oracle Circularity (Section 5):** Oracle requires solving the problem (any scale).
- **Oracle Impossibility (Section 5):** Oracle Circularity at infeasible scales.

All quantum search algorithms claiming advantage must avoid all of these barriers. Oracle Impossibility is simply the most severe manifestation of Oracle Circularity, occurring when problem size makes the circular requirement not just problematic but completely infeasible.

5.7. The Narrow Exception: When All Barriers Are Avoided

Sections 3, 4, and 5 established three independent barriers. A natural question: Do problems exist that avoid all three barriers simultaneously?

THEOREM (Quantum Advantage Conditions): Quantum advantage can exist when ALL four conditions hold simultaneously:

1) **Type B encoding** (Section 3): All computational states valid, avoiding Grover Dilemma.

2) **Symmetric polynomial setup** (Section 4): $C_{\text{setup}} = O(\text{poly}(n))$ for both classical and quantum.

3) **No classical shortcut:** No classical algorithm solves the problem in less than $O(2^n)$ time.

4) **No Oracle Circularity** (Section 5): Oracle can be specified without solving the target problem.

Proof:

- Classical total: $O(\text{poly}(n)) + O(2^n) = O(2^n)$

- Quantum total: $O(\text{poly}(n)) + O(2^{n/2}) = O(2^{n/2})$
- Quantum advantage exists because setup is polynomial and all barriers are avoided.

REMARK (Necessary but Not Sufficient): The four conditions in Theorem are necessary for quantum advantage but not sufficient. Even when all four computational/logical barriers are avoided, physical implementation barriers remain [45]. Quantum algorithms require sustained coherence over sequential operations, accumulating errors that grow with circuit depth. For cryptographic key search, the number of sequential Grover iterations $2^{k/2}$ ($k \geq 128$) for key length k far exceeds physically achievable coherence times and gate fidelities. Thus, asymptotic computational advantage does not guarantee practical implement ability—a barrier of physical constraints must also be overcome [45].

EXAMPLE 5.2 (Cryptographic Key Search - AES):

Problem: Find key k where AES_k (plaintext) = ciphertext, given plaintext-ciphertext pair.

Verification of four conditions:

1) **Type B encoding (avoids Grover Dilemma):** All 2^k keys are valid computational states. No invalid states to filter. Superposition construction: $O(k)$ Hadamard gates.

2) **Symmetric polynomial setup (avoids Setup Cost Dilemma):** AES has explicit public specification. Both classical and quantum implement the same algorithm with $O(\text{poly}(k))$ circuit complexity.

3) **No classical shortcut:** AES designed to resist structural attacks. Best known classical attack is brute-force $O(2^k)$. No polynomial-time classical solution exists.

4) **No Oracle Circularity (avoids Section 5 barrier):** Oracle evaluates "Does this key produce correct ciphertext?" This evaluation does not require knowing which key is correct. The oracle checks a condition, it doesn't encode the solution.

Complexity analysis:

- Classical: $O(\text{poly}(k))$ setup + $O(2^k)$ search = $O(2^k)$.
- Quantum: $O(\text{poly}(k))$ setup + $O(2^{k/2})$ search = $O(2^{k/2})$.
- Quantum advantage exists in total complexity.

Despite asymptotic advantage, physical barriers remain [45]:

- AES-128: Requires 2^{64} sequential Grover iterations, likely beyond decoherence limits.
- AES-256: Requires 2^{128} iterations, physically impossible with current understanding.
- Query complexity advantage may not translate to physical implementation.

Rarity of this exception: Problems satisfying all four conditions simultaneously are **extremely rare**. Cryptographic primitives are specifically engineered to be such exceptions—they are designed to have:

- Explicit polynomial specifications (condition 2)
- No structural weaknesses (condition 3)
- Simple verification without solution knowledge (condition 4)

Most practical problems fail at least one condition. This narrow exception does not invalidate our three-barrier framework. Rather, it validates the framework's completeness: we have identified necessary conditions for quantum advantage (avoid all three barriers), and this identification reveals how exceptionally rare such problems are. The framework correctly predicts both failure (most problems) and potential success (rare cryptographic cases), though even these successes face physical implementation barriers [45].

Conclusion: Quantum advantage requires simultaneously avoiding Grover Dilemma, Setup Cost Dilemma, and Oracle Circularity. Even these rare exceptions face additional physical barriers limiting practical implementation [45].

5.8. Summary

Oracle Circularity represents a third fundamental barrier to quantum computational advantage, distinct from both the Grover Dilemma (Section 3) and Setup Cost Dilemma (Section 4).

Key findings:

Definition: Oracle Circularity occurs when constructing the required oracle necessitates solving the same problem (or a problem of equivalent difficulty) that the quantum algorithm claims to solve. Formally, $\mathcal{C}(\text{oracle}) \geq \mathcal{C}(\text{problem})$.

Severity spectrum: Ranges from Oracle Circularity (circular but computationally feasible) to Oracle Impossibility (circular and computationally infeasible at scale).

Affected problems: Learning, optimization, and NP-complete problems where the oracle must identify unknown optimal solutions or satisfying assignments.

Independence from other barriers: Oracle Circularity is independent of Grover Dilemma and Setup Costs. Problems may suffer from one, two, or all three barriers. Subset Sum (Section 5.2) avoids Grover Dilemma and Setup Cost but still faces Oracle Circularity.

Universality: The analysis in Section 5.3 applies universally to all oracle-based quantum search problems (Section 5.4). Oracle Circularity affects the entire class of NP-complete problems (Section 5.5) and extends to learning/optimization problems where oracles must encode solution knowledge.

No technological solution: Unlike Setup Costs (which might be reduced with better implementation), Oracle Circularity is a fundamental logical issue. The oracle cannot be properly specified without solving what the algorithm claims to solve—no technological advancement can overcome this logical impossibility.

The narrow exception (Section 5.7): Problems avoiding all three barriers simultaneously are extremely rare, essentially limited to cryptographic primitives with polynomial-size specifications and no structural weaknesses. These problems satisfy: 1) Type B encoding, 2) symmetric polynomial setup, 3) no classical shortcuts, and 4) no Oracle Circularity. Even these rare cases face additional physical implementation barriers [45] that may prevent practical advantage.

Conclusion: For problems exhibiting Oracle Circularity, quantum search algorithms do not provide—and cannot provide—genuine computational advantage. The claimed quantum speedup is illusory, arising from the false assumption that oracles encoding problem solutions are available "for free." When oracle construction costs are properly accounted for, the quantum approach has equal or higher complexity than classical solutions.

Combined with Grover Dilemma and Setup Cost barriers, the three-barrier framework establishes that oracle-based quantum search faces systematic fundamental limitations across virtually all practical problem classes. Only an exceptionally narrow class of problems—cryptographic primitives satisfying three simultaneous restrictive conditions—may retain theoretical quantum advantage, and even these face physical barriers limiting practical implementation [45].

6. A Unified Framework: The Three Barries

This section synthesizes the three independent barriers presented in Sections 3-5 and establishes formal conditions under which oracle-based quantum search algorithms cannot achieve computational advantage.

6.1. The Three Independent Barriers

Sections 3, 4, and 5 identified three distinct barriers:

BARRIER 1 (Grover Dilemma, Section 3): For Type A problems where the computational space S exceeds the valid data space D, constructing superposition over only valid states requires $O(|D|)$ or $O(|S|)$ classical preprocessing.

BARRIER 2 (Setup Cost Dilemma, Section 4): Oracle construction or data loading may require classical computation with cost C_{setup} that equals or exceeds the cost of classical algorithms solving the problem.

BARRIER 3 (Oracle Circularity, Section 5): Oracle construction or specification may require solving the target problem, creating circular dependency where $C(\text{oracle}) \geq C(\text{problem})$.

Key properties:

1) **Independence:** These barriers are logically independent—a problem can suffer from one, two, or all three.

2) **Necessity:** A quantum algorithm must avoid ALL three barriers to achieve advantage.

3) **Sufficiency:** Avoiding all three is necessary but not sufficient (e.g., may still be beaten by efficient classical algorithms and more barriers [45]).

6.2. General Conditions for No Quantum Advantage

We now state precise conditions under which oracle-based quantum search algorithms cannot achieve computational advantage.

THEOREM 6.1 (Grover Dilemma Barrier): For Type A structured search problems where the computational space S exceeds the valid data space D ($|S| > |D|$), if constructing superposition over only valid states requires classical preprocessing with cost $C_{\text{super}} \geq C_{\text{classical}}$, then no quantum advantage exists.

Proof: Total complexity is $C_{\text{total}} = C_{\text{super}} + C_{\text{quantum}} \geq C_{\text{classical}} + C_{\text{quantum}} > C_{\text{classical}}$.

THEOREM 6.2 (Setup Cost Barrier): For any quantum algorithm, if oracle construction or data loading requires cost $C_{\text{setup}} \geq C_{\text{classical}}$, then no quantum advantage exists for single-query scenarios.

Proof: Total complexity is $C_{\text{total}} = C_{\text{setup}} + C_{\text{quantum}} \geq C_{\text{classical}} + C_{\text{quantum}} > C_{\text{classical}}$.

THEOREM 6.3 (Oracle Circularity Barrier): For any quantum algorithm where oracle construction requires solving the target problem P with cost $C(\text{oracle}) \geq C(P)$, then no quantum advantage exists.

Proof: Total complexity includes solving P to build oracle: $C_{\text{total}} = C(\text{oracle}) + C_{\text{quantum}} \geq C(P)$. The quantum portion adds cost without reducing the requirement to solve P.

THEOREM 6.4 (Composite Barrier): If a quantum algorithm suffers from ANY of the three barriers (Grover Dilemma, Setup Cost, or Oracle Circularity), it cannot achieve computational advantage over the best classical algorithm for the problem.

Proof: Each barrier independently ensures $C_{\text{total}} \geq C_{\text{classical}}$. If any barrier applies, quantum advantage is eliminated.

THEOREM 6.5 (Oracle-Based Search Limitations): For oracle-based quantum search algorithms on problems that:

1) Are Type A (suffer Grover Dilemma), OR
2) Require expensive oracle construction (suffer Setup Cost), OR
3) Require solving the problem to specify the oracle (suffer Oracle Circularity)

No quantum computational advantage exists over optimal classical algorithms.

Proof: By Theorems 6.1, 6.2, 6.3, and 6.4.

Note: This does NOT claim all quantum algorithms fail—algorithms like Shor's factoring avoid all three barriers through mathematical structure rather than exhaustive search.

6.3. Specific Algorithm Results

Having established the general framework, we now state results for specific well-known quantum algorithms analyzed in Sections 4 and 5.

THEOREM 6.6 (Deutsch's Algorithm): No quantum computational advantage exists for Deutsch's Algorithm in practical scenarios.

Proof: See Section 4. For general black-box functions, Setup Cost Dilemma applies (Section 4.8 and Theorem 6.2). For structured functions with explicit formulas, classical algorithms are equally efficient.

THEOREM 6.7 (Deutsch-Jozsa Algorithm): No quantum computational advantage exists for Deutsch-Jozsa Algorithm in practical scenarios.

Proof: See Section 4. For arbitrary black-box functions, Setup Cost Dilemma applies (Section 4.8 and Theorem 6.2). For structured functions with compact representation, classical algorithms achieve similar complexity. The promise that f is constant or balanced is artificial and enables efficient classical solutions.

THEOREM 6.8 (Simon's Algorithm): No quantum computational advantage exists for Simon's Algorithm in practical scenarios.

Proof: See Section 4. For general black-box functions, Setup Cost Dilemma applies (Section 4.8 and Theorem 6.2)—oracle construction requires $O(2^n)$ function evaluations. For structured functions with compact representation (e.g., $f(x) = Ax$), classical algorithms can solve directly (e.g., solve $As = 0$ via Gaussian elimination in $O(n^3)$), achieving similar or better complexity.

THEOREM 6.9 (NP-Complete Problems): No quantum computational advantage exists for oracle-based quantum search algorithms on NP-complete problems in general.

Clarification on "No Quantum Advantage": This statement means no *practical* computational advantage exists when all costs are properly accounted for. Specifically:

1) **Theoretical query complexity:** Grover's algorithm provides quadratic speedup in oracle queries—from $O(2^n)$ classical queries to $O(2^{n/2})$ quantum queries. In the oracle model, this is a genuine complexity-theoretic improvement.

2) **Total computational complexity:** When oracle construction costs $C(\text{oracle})$ are included, the total complexity becomes:

- Classical: $O(2^n)$
- Quantum: $C(\text{oracle}) + O(2^{n/2})$

3) **Oracle Circularity:** For NP-complete problems, $C(\text{oracle}) \geq O(2^n)$ because the oracle must identify which candidate solutions satisfy the problem constraints—which requires solving the NP-complete problem. Therefore:

- Total quantum complexity: $\geq O(2^n) + O(2^{n/2}) = O(2^n)$
- No improvement over classical $O(2^n)$

4) **Why quadratic speedup doesn't translate to practical advantage:** While $O(2^{n/2})$ is asymptotically faster than $O(2^n)$, this speedup applies only to the oracle query phase. The quadratic speedup in queries is swamped by the exponential cost in oracle construction.

We therefore conclude: Oracle Circularity eliminates *practical* quantum computational advantage for NP-complete problems, even though theoretical query complexity analysis shows quadratic speedup. The distinction between query complexity (queries to a given oracle) and total complexity (including oracle construction) is crucial.

Proof: See Section 5. NP-complete problems suffer from Oracle Circularity (Theorem 6.3). The oracle must identify which states satisfy the NP-complete predicate (e.g., which subsets sum to T for Subset Sum, which assignments satisfy a Boolean formula for SAT). Determining which states to mark requires solving the NP-complete problem itself, creating circular dependency where $C(\text{oracle}) \geq C(\text{problem}) \geq O(2^n)$. Therefore, by Theorem 6.3, total computational complexity provides no advantage over classical approaches: $C_{\text{total}} = C(\text{oracle}) + C_{\text{quantum}} \geq O(2^n) + O(2^{n/2}) = O(2^n) \geq C_{\text{classical}}$.

Examples: Subset Sum, SAT, Graph Coloring, TSP, Knapsack (Section 5).

THEOREM 6.10 (Learning and Optimization Problems): No quantum computational advantage exists for oracle-based quantum search algorithms on learning and optimization problems where the oracle must identify optimal solutions.

Proof: See Section 5. These problems suffer from Oracle Circularity (Theorem 6.3). The oracle must identify which states represent optimal or high-quality solutions (e.g., optimal policy in reinforcement learning). Determining which states to mark requires solving the learning/optimization problem. Therefore, by Theorem 6.3, no quantum advantage exists.

Example: Reinforcement learning for stock trading (Section 5).

6.4. Summary

The three-barrier framework provides necessary conditions for oracle-based quantum search to achieve computational advantage:

1) **Must be Type B** or efficiently overcome Grover Dilemma.

2) **Must have efficient oracle construction** (avoid Setup Cost).

3) **Must avoid Oracle Circularity** (oracle doesn't require solving problem).

All three conditions must be satisfied. Satisfying all three is necessary but not sufficient—classical algorithms exploiting the same structure may still be competitive and more barrier will be introduced in [45].

Key results are:

General framework (Theorems 6.1-6.5):

- Three independent barriers identified and formalized.
- Conditions for no quantum advantage established.
- Composite barrier theorem shows any single barrier eliminates advantage.

Specific algorithms (Theorems 6.6-6.8):

- Deutsch's Algorithm: No practical advantage (Setup Cost).
- Deutsch-Jozsa Algorithm: No practical advantage (Setup Cost or classical equivalence).
- Simon's Algorithm: No practical advantage (Setup Cost or classical equivalence).

Problem classes (Theorems 6.9-6.10):

- NP-complete problems: No advantage (Oracle Circularity).
- Learning/optimization problems: No advantage (Oracle Circularity).

Section 7 applies this framework to analyze the broader implications, provide guidance for algorithm design, and discuss the future direction of quantum computing research.

7. Discussion

This section synthesizes the findings from Sections 3-6, categorizes common patterns in misleading quantum advantage claims, and acknowledges the limitations of this analysis.

7.1. Summary of Findings

Having established the three-barrier framework (Section 6), we now summarize how it applies across all problem classes analyzed in Sections 3-5. **Table 5** provides a comprehensive classification showing which barriers affect each problem type.

Table 5. Comprehensive problem classification by barriers.

Problem	*Grover dilemma?*	*Setup cost dilemma?*	*Oracle circularity?*	*Quantum advantage?*
Database search	Yes (O(N) preprocessing)	Yes (O(N) loading)	No	**No**
Deutsch (general)	N/A	Yes ($O(2^n)$ construction)	No	**No**
Deutsch-Jozsa (black-box)	N/A	Yes ($O(2^n)$ construction)	No	**No**
Deutsch-Jozsa (formula)	N/A	No (O(n) from formula)	No	**No** (classical also O(n))
Simon (black-box)	No (Type B)	Yes ($O(2^n)$ construction)	No	**No**
Simon (structured)	No (Type B)	No ($O(n^2)$ from matrix)	No	**No** (classical also $O(n^3)$)
Subset sum	No (Type B)	No (O(poly(n)) from formula)	Yes	**No**
SAT (general)	No (Type B)	No (O(poly(n)) from formula)	Yes	**No**
Trading RL	No (Type B)	No (O(T) data loading)	Yes	**No**
NP-complete (general)	No (Type B typically)	Varies	Yes	**No**

Key observations:

1) Type B problems avoid Grover Dilemma but typically suffer from Oracle Circularity.

2) Black-box functions suffer Setup Cost Dilemma across all algorithms.

3) Structured functions avoid Setup Cost but classical algorithms also exploit the same structure.

4) Oracle Circularity is pervasive in learning, optimization, and NP-complete problems.

5) No problem examined provides practical quantum advantage when all costs are properly accounted for [45].

Critical finding: Oracle-based quantum search provides genuine computational advantage only for an exceptionally narrow problem class: cryptographic primitives with polynomial-size specifications and no structural weaknesses (Section 5.7). For all other analyzed problem classes (database search, Deutsch-type algorithms, NP-complete problems, and learning/optimization), no quantum advantage exists.

This finding does not imply quantum computing has no advantages: Algorithms like Shor's factoring [38] and quantum simulation [39] [40] provide genuine exponential advantages through fundamentally different paradigms: exploiting mathematical structure rather than exhaustive search, and direct physical implementation rather than oracle-based queries. The limitation is specific to the oracle-based search paradigm using amplitude amplification.

7.2. Categories of Misleading Quantum Advantage Claims in Oracle-Based Algorithms

The three-barrier framework reveals that many claimed quantum search advantages fall into identifiable categories of flawed analysis. Understanding these categories helps evaluate future quantum algorithm claims.

Category A: Incorrect Results (Ignoring Validity)

Characteristics:

- Algorithms using complete superposition over computational states.
- Ignore problem structure and validity constraints.
- Report speedups while producing false positives or wrong answers.

Example: Searching for element 3 in data $D = \{0, 1, 2, 2\}$ using 2-bit encoding. The superposition includes $|11\rangle$ (representing 3), which is not in D. If the algorithm searches for 3, it may return $|11\rangle$—an invalid state producing incorrect results.

Which barrier: Grover Dilemma (Type A problems)

Why misleading: Claimed speedup is meaningless if answers are incorrect. Any algorithm, quantum or classical, can be "fast" if correctness is not required.

Category B: Artificial Problem Restrictions (Circular Assumptions)

Characteristics:

- Algorithms assuming solutions are known or easily identifiable.
- Solve artificially simplified problems where quantum advantage is meaningless.
- Claim $O(\sqrt{N})$ complexity while hiding solution knowledge in problem definition.

Example: A "search" algorithm that requires knowing solution locations in advance. If we already know which states are solutions, we can return one in $O(1)$ time—the quantum algorithm is unnecessary.

Which barrier: Oracle Circularity (solution-marking oracle requires knowing solutions)

Why misleading: If solutions are known, the problem is already solved. The algorithm is circular: it uses the solution to find the solution.

Category C: Hidden Classical Overhead (Incomplete Cost Accounting)

Characteristics:

- Algorithms hiding classical preprocessing costs in complexity analysis.
- Report $O(\sqrt{N})$ quantum complexity while ignoring $O(N)$ classical setup.
- Total complexity is $O(N) + O(\sqrt{N}) = O(N)$, no advantage.

Example: Database search with $O(N)$ data loading and $O(\sqrt{N})$ quantum search. Reporting only the quantum portion as "quadratic speedup" while ignoring setup cost that dominates total complexity.

Which barrier: Setup Cost Dilemma

Why misleading: When all costs are included, total complexity equals or exceeds classical approaches. The apparent advantage disappears under complete accounting.

Category D: Oracle Abstraction (Circular Oracle Construction)

Characteristics:

- Algorithms treating oracles as "free" without accounting for construction costs.
- Oracle construction requires solving the problem.
- Oracle Circularity eliminates advantage.

Example: Subset Sum algorithm where the oracle must mark all subsets summing to target T. But determining which subsets sum to T requires solving Subset Sum—the oracle cannot be built without solving what the algorithm claims to solve.

Which barrier: Oracle Circularity

Why misleading: The oracle cannot be properly constructed or specified without solving the target problem. The "speedup" assumes access to an oracle that requires solving the problem to build, creating circular dependency.

Category E: Special Structured Problems

Characteristics:

- Algorithms restricted to special cases where setup costs are small.
- Problems have sufficient structure to enable efficient oracle construction.
- Classical algorithms exploit the same structure equally efficiently.

Examples:

- Deutsch-Jozsa for explicit formulas (parity function $f(x) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$): Both quantum oracle construction and classical evaluation are $O(n)$.
- Simon's algorithm for known linear functions $f(x) = Ax$: Classical can solve $As = 0$ via Gaussian elimination in $O(n^3)$.

Which barrier: Avoids Setup Cost Dilemma for special cases, but classical achieves similar complexity

Why misleading: Classical algorithms are more than a match for these cases. When problems have enough structure to enable efficient quantum oracle construction, that same structure allows classical algorithms to solve the problem efficiently. The quantum advantage vanishes because both approaches have similar complexity. Table 6 categorizes the five types of misleading quantum advantage claims identified throughout this analysis.

Table 6. Categories of misleading claims.

Category	*Primary issue*	*Barrier*	*Classical reality*
A: Incorrect results	Ignores validity	Grover dilemma	Would also produce wrong answers
B: Circular assumptions	Assumes solutions known	Oracle circularity	Already solved if known

Continued

C: Hidden overhead	Ignores setup costs	Setup cost	O(N) preprocessing kills advantage
D: Oracle abstraction	Oracle requires solving problem	Oracle circularity	Can't build oracle without solution
E: Special structured	Restricted to easy cases	None (special structure)	Classical exploits same structure

Identifying Misleading Claims: Any quantum search algorithm claiming speedups must be examined for:

1) How is the initial superposition constructed? What is the true cost? → Check for Category A or C.

2) Does the algorithm handle invalid or non-existent targets correctly? → Check for Category A.

3) What classical preprocessing or problem knowledge is assumed? → Check for Category B or C.

4) How is the oracle constructed, and at what cost? → Check for Category C or D.

5) Does oracle specification require solving the problem? → Check for Category D.

6) If the problem has special structure enabling efficient oracle construction, can classical algorithms exploit the same structure? → Check for Category E.

Failure to address these questions indicates the algorithm likely falls into one of the five problematic categories.

The three-barrier framework provides a systematic method for this evaluation. By checking whether an algorithm suffers from Grover Dilemma (Type A), Setup Cost Dilemma, or Oracle Circularity, and by verifying complete cost accounting, researchers can identify which category of misleading claim applies—or confirm that genuine quantum advantage exists.

7.3. Limitations of This Analysis

1) Focus on oracle-based search: This analysis specifically targets quantum algorithms using amplitude amplification over search spaces. It does not address all quantum computing paradigms. Algorithms exploiting mathematical structure (Shor's factoring [38]), quantum simulation of physical systems, quantum annealing, and other non-search paradigms use fundamentally different approaches and are not covered by the three-barrier framework.

2) Asymptotic complexity: The analysis focuses on worst-case asymptotic complexity. It does not comprehensively address constant factor improvements, average-case complexity for specific problem distributions, or problem instances where constant factors dominate asymptotic terms. For problems where classical algorithms have large constants and quantum has small constants, quantum might provide practical benefit even without asymptotic advantage.

3) Current classical algorithms: Comparisons are against best-known classical algorithms as of 2025. Future classical algorithmic breakthroughs could change the landscape. Classical algorithms continue to improve (e.g., advances in SAT solvers, matrix multiplication, optimization heuristics), and such improvements could strengthen the case against quantum advantage or reveal new contexts where quantum helps.

4) Theoretical framework: The analysis is theoretical and mathematical. Experimental validation on scaled quantum computers with realistic noise and error rates remains pending. Practical considerations including hardware limitations, decoherence, gate fidelity, and connectivity constraints are not fully addressed. The framework establishes fundamental barriers but does not predict exact crossover points where quantum might become practical.

7.3.1. Assumptions

1) Complete cost accounting: The analysis assumes all classical preprocessing must be counted in total complexity. In contexts where costs can be amortized over many queries (e.g., $M >> \sqrt{N}$ database queries on persistent data structures), conclusions might differ. The acknowledged exception for repeated queries does not invalidate the framework but shows that specific scenarios may avoid Setup Cost Dilemma through amortization.

2) Oracle model: The analysis assumes oracles must be explicitly constructed or specified with associated costs. For some mathematical problems, efficient implementations might exist without exhaustive enumeration. The burden of proof is on those claiming such efficient implementations exist to demonstrate construction methods and verify costs.

3) Classical-quantum separation: The analysis assumes classical preprocessing cannot be parallelized quantumly in ways that fundamentally reduce costs. For data loading in particular, this seems fundamental: classical data requires classical measurement and cannot be "directly" accessed quantumly without classical processing. Quantum mechanics' measurement postulates enforce this boundary.

4) Problem formulations: The analysis uses standard problem formulations from computer science and optimization literature. Alternative formulations might change conclusions for specific cases. However, the burden of proof is on those proposing alternative formulations to show they represent the same computational problem and to verify that reformulation doesn't smuggle in additional structure or information.

7.3.2. What This Analysis Does NOT Claim

1) Quantum computing has no advantages:

FALSE. Shor's algorithm provides genuine exponential advantage for factoring and discrete logarithm problems. Quantum simulation provides genuine exponential advantage for simulating quantum physical systems. Other quantum algorithms exploiting mathematical structure (e.g., solving linear systems via HHL [41], certain applications of quantum walks) may provide advantages in specific contexts.

TRUE. Oracle-based quantum search doesn't provide practical advantage for the problems analyzed when all costs are properly accounted for.

2) All quantum algorithms fail:

FALSE. Shor's factoring (exploits mathematical structure, no search), quantum simulation (direct physical implementation, no oracles), variational quantum algorithms (different paradigm), and other non-search approaches may succeed.

TRUE. Oracle-based quantum search algorithms face fundamental barriers for problems analyzed in this work.

3) Quantum search is completely useless:

FALSE. Oracle-based quantum search provides genuine asymptotic computational advantage for an exceptionally narrow problem class: cryptographic primitives with polynomial-size specifications, no structural weaknesses, and avoiding all three barriers (Section 5.7). Example: AES key search achieves $O(2^{k/2})$ vs classical $O(2^k)$ when all computational costs are included.

CAVEAT. Even this narrow exception faces physical implementation barriers [45] that may prevent practical advantage. AES-128 requires ~2^{64} sequential iterations, likely beyond decoherence limits.

TRUE. No advantage found for: single-query database scenarios, NP-complete problems (Oracle Circularity), learning problems (Oracle Circularity), optimization problems (Oracle Circularity), or structured problems with efficient classical algorithms. Most claimed advantages are artifacts of incomplete cost analysis.

4) Future developments can't change anything:

FALSE. New algorithmic paradigms might emerge. Better problem characterization might reveal narrow exceptions. Problems naturally avoiding all three barriers might be discovered (as Shor's algorithm demonstrated for factoring).

TRUE. Oracle Circularity for NP-complete problems is a logical impossibility that cannot be overcome through technological or algorithmic advances. Setup Cost and Grover Dilemma stem from fundamental requirements (classical data processing, valid state identification) that appear immutable.

7.3.3. What This Analysis DOES Claim

1) Three fundamental barriers exist for oracle-based search: The Grover Dilemma (superposition construction for Type A problems), Setup Cost Dilemma (oracle construction and data loading), and Oracle Circularity (oracle must solve target problem) represent distinct, independent barriers. Each is sufficient to eliminate quantum advantage.

2) Most claimed quantum search advantages are artifacts: Common flaws include incomplete cost accounting (ignoring setup), circular assumptions (assuming solutions known), wrong baselines (comparing against naive classical), and ignoring correctness issues (false positives). The five categories (A-E) in Section 7.2 characterize these flaws systematically.

3) Oracle-based search faces systematic challenges: For many important problem classes—including NP-complete problems, learning, optimization, and data-dependent search—oracle-based quantum search faces fundamental obstacles. When all costs are properly accounted for, no practical advantage was found in our analysis.

4) Rigorous evaluation is essential: Complete cost accounting is required (setup + execution + verification). Comparison must be against best classical algorithms, not strawman baselines. All three barriers must be checked. Correctness must be verified, not just speed measured.

7.3.4. Clarification on "Oracle-Based"

In this work, "oracle-based" refers to algorithms that query black-box functions without knowing their structure, use query complexity as the primary measure, and apply amplitude amplification for search. Examples include Grover's algorithm, Deutsch-Jozsa, and Simon's algorithm (for black-box functions).

This does NOT include algorithms like Shor's, which implement specific mathematical functions (modular exponentiation) as explicit quantum circuits rather than treating them as black-box oracles. While Shor's uses subroutines, these are not oracles in the technical complexity-theoretic sense. The key distinction is between searching exponential spaces (oracle-based search) and exploiting mathematical structure (Shor's paradigm).

7.3.5. Future Work Directions

1) Experimental validation: Test predictions on scaled quantum systems. Measure end-to-end performance including all costs. Compare against optimal classical implementations. Document cases where quantum fails or succeeds.

2) Refined characterization: Better define boundaries between problems that can/cannot benefit from quantum search. Identify mathematical structures enabling quantum advantage. Develop theory for when search-based quantum helps (if ever).

3) Alternative paradigms: Explore non-search quantum approaches for NP-complete problems. Develop quantum algorithms that avoid oracle circularity. Investigate hybrid quantum-classical methods.

4) Positive results: Identify and rigorously prove cases where quantum search genuinely helps. If such cases exist, characterize them precisely. Develop theory explaining why those cases work while others don't.

5) Classical algorithm development: Continue improving classical algorithms. Develop better baselines for quantum comparison. Recognize that classical optimization remains a productive research direction.

7.3.6. Summary of Limitations

This work provides a systematic framework for evaluating oracle-based quantum search algorithms and demonstrates that such algorithms face fundamental barriers for many important problem classes. The analysis is:

- **Scoped:** Focuses on search-based quantum algorithms, not all quantum computing.
- **Theoretical:** Based on asymptotic complexity analysis, not experimental re-

sults.

- **Conservative:** Makes specific, defensible claims rather than broad generalizations.
- **Honest:** Acknowledges what is and isn't claimed.

The framework does not claim quantum computing has no future. Rather, it clarifies that the oracle-based search paradigm faces fundamental obstacles for most practical problems. The future of quantum advantage lies in paradigms that exploit mathematical structure (Shor's), natural physical mappings (simulation), or yet-to-be-discovered approaches that avoid all three barriers—not in incremental improvements to oracle-based search for problems with inherent Oracle Circularity.

7.4. Future Work: Additional Barriers

The three barriers identified in this work—Grover Dilemma (Section 3), Setup Cost Dilemma (Section 4), and Oracle Circularity (Section 5)—are not exhaustive. Additional fundamental barriers exist that further constrain oracle-based quantum search algorithms.

Physical Implementation Barriers: Even when an algorithm avoids all three barriers identified in this work, physical implementation faces fundamental constraints including:

- **Coherence time limits:** Quantum states decay exponentially with time constant T_2, limiting achievable circuit depth.
- **Gate fidelity requirements:** Errors accumulate over sequential operations, requiring impossibly high fidelity for deep circuits.
- **Operator depth constraints:** Each Grover iteration requires substantial circuit depth, multiplying the sequential operation count.

These physical barriers are analyzed systematically in a companion paper [45], which demonstrates that even algorithms avoiding the three logical/computational barriers identified here face insurmountable physical constraints. The combination of computational barriers (this work) and physical barriers [45] suggests that oracle-based quantum search faces fundamental limitations from multiple independent directions.

The complete picture: A quantum algorithm claiming advantage must simultaneously avoid:

1) Grover Dilemma (superposition construction for Type A problems).

2) Setup Cost Dilemma (oracle construction and data loading costs).

3) Oracle Circularity (oracle specification requiring problem solution).

4) Physical implementation barriers (coherence time, gate fidelity, operator depth) [45].

Failure to satisfy any one of these requirements eliminates practical quantum advantage. The narrow window for genuine quantum search advantage—if it exists at all for practical problems—requires avoiding all four independent barrier categories.

Through systematic analysis of major problem classes and specific quantum algorithms, we found no practical cases where oracle-based quantum search provides genuine computational advantage over optimal classical approaches when all four barriers are properly accounted for. Even the narrow exception of cryptographic primitives (Section 5.7)—which avoids the first three computational/logical barriers—faces the fourth barrier of physical implementation constraints [45], making practical advantage unlikely despite theoretical asymptotic improvements.

8. Conclusion

This work has presented a systematic framework for evaluating oracle-based quantum search algorithms through the identification of three independent fundamental barriers: the Grover Dilemma, the Setup Cost Dilemma, and Oracle Circularity. Our analysis demonstrates that these barriers eliminate claimed quantum computational advantages for a broad class of important problems when all costs are properly accounted for.

8.1. Summary of Main Findings

The Three-Barrier Framework: We established three independent barriers that oracle-based quantum search algorithms must overcome to achieve genuine computational advantage:

Barrier 1 (Grover Dilemma, Section 3): For Type A problems where the computational space exceeds the valid data space, constructing superposition over valid states requires O(|D|) or O(|S|) classical preprocessing.

Barrier 2 (Setup Cost Dilemma, Section 4): Oracle construction and data loading may require classical computation with cost C_setup ≥ C_classical.

Barrier 3 (Oracle Circularity, Section 5): Oracle construction or specification may require solving the target problem, creating circular dependency where C(oracle) ≥ C(problem).

Key property: These barriers are logically independent. A quantum algorithm must avoid **all three** to achieve genuine advantage.

Comprehensive Problem Analysis: We systematically analyzed major problem classes and specific quantum algorithms:

Database search (Section 4): Suffers from Grover Dilemma (Type A) and Setup Cost (O(N) data loading).

Deutsch's Algorithm (Section 4): Suffers from Setup Cost Dilemma for black-box functions; classical algorithm is equally efficient for structured functions.

Deutsch-Jozsa Algorithm (Section 4): Suffers from Setup Cost Dilemma for black-box functions; classical algorithm achieves similar complexity for structured functions.

Simon's Algorithm (Section 4): Suffers from Setup Cost Dilemma for black-box functions; classical solutions are equally efficient for structured functions.

Subset Sum (Section 5): Avoids Grover Dilemma and Setup Cost but suffers critically from Oracle Circularity.

NP-Complete Problems (Section 5.5): Generally, suffer from Oracle Circularity—oracle must identify solutions, requiring solution of the problem itself.

Learning and Optimization (Section 5.4): Suffer from Oracle Circularity—oracle must identify optimal solutions, which is the problem being solved.

Formal Results (Section 6): We established ten theorems (6.1-6.10) characterizing when quantum advantage cannot exist:

- **Theorems 6.1-6.5:** General conditions under which the three barriers eliminate quantum advantage
- **Theorems 6.6-6.8:** No quantum advantage for Deutsch's, Deutsch-Jozsa, and Simon's algorithms in practical scenarios
- **Theorems 6.9-6.10:** No quantum advantage for NP-complete problems or learning/optimization problems with oracle circularity

Critical Finding: When all costs are properly accounted for, oracle-based quantum search provides genuine computational advantage only for an exceptionally narrow problem class: cryptographic primitives with polynomial-size specifications, no structural weaknesses, and avoiding all three barriers simultaneously (Section 5.7). Even these face physical implementation barriers [45] that likely prevent practical advantage. Every other problem examined either suffers from one or more of the three barriers or has efficient classical algorithms exploiting the same structure.

8.2. Implications and Future Directions

The three-barrier framework has important implications for quantum computing research.

Problem Classification: Section 7.1 provides comprehensive classification of all problems analyzed, showing which barriers affect which problem classes. This classification reveals systematic patterns: Type B problems avoid Grover Dilemma but typically face Oracle Circularity; black-box functions universally suffer Setup Cost Dilemma; structured functions avoid setup costs but classical algorithms exploit the same structure equally well.

Common Patterns in Failed Claims: Section 7.2 identifies five categories (A-E) of misleading quantum advantage claims, providing a systematic method for evaluating future algorithm proposals. These categories—incorrect results, circular assumptions, hidden overhead, oracle abstraction, and special structured problems—encompass most flawed analyses in the literature.

Scope and Limitations: Section 7.3 acknowledges the scope and limitations of this analysis. The framework applies specifically to oracle-based search using amplitude amplification, not to all quantum computing. Successful paradigms like Shor's factoring and quantum simulation use fundamentally different approaches and provide genuine advantages.

Directions for Future Research: The analysis suggests several productive research directions:

1) Structure-exploiting algorithms: Like Shor's, which avoid exhaustive search by exploiting mathematical properties (periodicity, group structure). Success comes from discovering hidden structure through quantum interference, not searching exponential spaces.

2) Quantum simulation: Direct implementation of quantum dynamics for chemistry, materials science, and fundamental physics. Natural mapping between quantum systems provides genuine exponential advantage without oracles or search.

3) Better problem characterization: Formal theory distinguishing problems that naturally avoid all three barriers from those that inherently face them. Understanding which mathematical structures enable quantum advantage guides algorithm development toward promising directions.

4) Hybrid quantum-classical approaches: Algorithms where quantum provides advantage for specific subroutines within classical frameworks. Variational algorithms (VQE, QAOA) represent near-term applications that may provide practical benefit even without full exponential advantage.

5) Honest evaluation standards: Complete cost accounting, comparison against best classical algorithms, explicit oracle specifications, and acknowledgment of negative results. Rigorous evaluation strengthens the field by focusing effort on approaches that genuinely work.

What should be avoided: The analysis also clarifies what should not be pursued:

Oracle-based search for NP-complete problems faces Oracle Circularity—a logical impossibility that cannot be overcome through algorithmic refinement or technological improvement. Resources devoted to this direction would be better spent on alternative approaches.

Search-based learning and optimization similarly face Oracle Circularity. The oracle must identify optimal solutions, but finding optimal solutions is the problem. Classical alternatives (dynamic programming, gradient methods) often provide superior solutions.

Incremental improvements to setup costs cannot change fundamental asymptotic complexity. While technologies like QRAM may reduce constants, they do not eliminate the O(N) data loading requirement that underlies Setup Cost Dilemma.

Claims without complete cost accounting should be viewed skeptically. Any quantum algorithm paper must specify oracle construction methods and costs, account for all preprocessing, and compare against optimal classical algorithms—not just naive baselines.

The path forward: Quantum computing's future lies in paradigms that naturally avoid the three barriers:

Exploiting mathematical structure (Shor's approach) rather than searching exponential spaces. Success comes from discovering hidden properties through quantum interference and Fourier analysis, not from amplitude amplification over solution spaces.

Direct physical simulation (quantum simulation approach) where quantum systems naturally represent target dynamics. The exponential advantage arises from native Hilbert space representation, not from algorithmic speedup.

New paradigms yet to be discovered that avoid oracle-based search entirely. History suggests breakthroughs come from fundamentally new approaches, not incremental improvements to existing paradigms that face fundamental barriers.

Honest assessment of where quantum helps and where it doesn't. Acknowledging limitations strengthens the field by setting realistic expectations, focusing resources appropriately, and maintaining credibility with funders and the broader scientific community.

The three-barrier framework provides a systematic method for this evaluation. By checking whether algorithms suffer from Grover Dilemma, Setup Cost Dilemma, or Oracle Circularity, researchers can identify promising directions and avoid pursuing approaches that face insurmountable obstacles.

8.3. Final Remarks

Oracle-based quantum search algorithms face three fundamental barriers that eliminate practical computational advantage for the broad class of problems analyzed in this work. These barriers are not temporary obstacles but fundamental constraints rooted in logic, physics, and computational requirements.

This is not a negative result about quantum computing—it is clarity. Quantum computing has genuine potential for exponential advantage through paradigms like Shor's factoring and quantum simulation. The field advances faster by focusing on these successful paradigms rather than pursuing approaches that face insurmountable barriers.

Oracle Circularity in particular represents a logical barrier without resolution. For NP-complete problems, learning, and optimization, the oracle must identify solutions—but identifying solutions is the problem. This circularity cannot be overcome through innovation or technology because it is logically contradictory. It is as fundamental as the uncomputability of the halting problem.

The distinction between barriers matters. Oracle Circularity is logically impossible to overcome. Setup Cost and Grover Dilemma stem from physical and computational requirements (quantum measurement postulates, problem structure examination) that appear fundamental though perhaps less obviously so. Understanding which barriers are absolute and which might have narrow exceptions focuses research appropriately.

The path forward is clear:

- Identify problems naturally avoiding all three barriers.
- Develop new paradigms beyond amplitude amplification.
- Rigorously evaluate claims with complete cost accounting.
- Compare against optimal classical algorithms.
- Acknowledge limitations honestly.

- Focus on approaches that genuinely provide quantum advantage.

The three-barrier framework provides a systematic method for evaluating quantum search algorithms and distinguishing genuine advantages from artifacts of incomplete analysis. By clarifying where oracle-based search cannot work, we focus the quantum computing community's efforts on paradigms and problems where quantum advantage is practically achievable.

Quantum computing's future lies not in overcoming fundamental barriers, but in exploiting problems and paradigms that naturally avoid them. This work contributes to that future by providing clarity about which approaches work and which do not, enabling more productive allocation of research effort toward directions with genuine potential for quantum computational advantage.

Acknowledgements

We thank the anonymous AI reviewers for their careful analysis and constructive feedback. We thank Gina Porter for proof reading this paper.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Grover, L.K. (1996) A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of the Twenty-Eighth Annual ACM symposium on Theory of computing—STOC* '96, Philadelphia, 22-24 May 1996, 212-219. https://doi.org/10.1145/237814.237866

[2] Rebentrost, P., Mohseni, M. and Lloyd, S. (2014) Quantum Support Vector Machine for Big Data Classification. *Physical Review Letters*, **113**, Article 130503. https://doi.org/10.1103/physrevlett.113.130503

[3] Wiebe, N., Kapoor, A. and Svore, K.M. (2015) Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning. *Quantum Information and Computation*, **15**, 316-356. https://doi.org/10.26421/qic15.3-4-7

[4] Farhi, E., Goldstone, J. and Gutmann, S. (2014) A Quantum Approximate Optimization Algorithm. arXiv:1411.4028.

[5] Peruzzo, A., McClean, J., Shadbolt, P., Yung, M., Zhou, X., Love, P.J., *et al.* (2014) A Variational Eigenvalue Solver on a Photonic Quantum Processor. *Nature Communications*, **5**, Article No. 4213. https://doi.org/10.1038/ncomms5213

[6] Bernstein, D.J. (2009) Introduction to Post-Quantum Cryptography. In: Bernstein, D.J., Buchmann, J. and Dahmen, E., Eds., *Post-Quantum Cryptography*, Springer, 1-14. https://doi.org/10.1007/978-3-540-88702-7_1

[7] Bennett, C.H., Bernstein, E., Brassard, G. and Vazirani, U. (1997) Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing*, **26**, 1510-1523. https://doi.org/10.1137/s0097539796300933

[8] Ambainis, A. (2007) Quantum Walk Algorithm for Element Distinctness. *SIAM Journal on Computing*, **37**, 210-239. https://doi.org/10.1137/s0097539705447311

[9] Nielsen, M.A. and Chuang, I.L. (2010) Quantum Computation and Quantum Information. Cambridge University Press.

[10] Brassard, G., Høyer, P., Mosca, M. and Tapp, A. (2002) Quantum Amplitude Amplification and Estimation. *Contemporary Mathematics*, **305**, 53-74.

[11] Childs, A.M. and Goldstone, J. (2004) Spatial Search by Quantum Walk. *Physical Review A*, **70**, Article 022314. https://doi.org/10.1103/physreva.70.022314

[12] Aaronson, S. and Ambainis, A. (2018) Forrelation: A Problem That Optimally Separates Quantum from Classical Computing. *SIAM Journal on Computing*, **47**, 982-1038. https://doi.org/10.1137/15m1050902

[13] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N. and Lloyd, S. (2017) Quantum Machine Learning. *Nature*, **549**, 195-202. https://doi.org/10.1038/nature23474

[14] Preskill, J. (2018) Quantum Computing in the NISQ Era and Beyond. *Quantum*, **2**, Article 79. https://doi.org/10.22331/q-2018-08-06-79

[15] Figgatt, C., Maslov, D., Landsman, K.A., Linke, N.M., Debnath, S. and Monroe, C. (2017) Complete 3-Qubit Grover Search on a Programmable Quantum Computer. *Nature Communications*, **8**, Article 1918. https://doi.org/10.1038/s41467-017-01904-7

[16] Grover, L.K. (1998) Quantum Computers Can Search Rapidly by Using Almost Any Transformation. *Physical Review Letters*, **80**, 4329-4332. https://doi.org/10.1103/physrevlett.80.4329

[17] IBM Quantum Team (2023) IBM Quantum Backend Specifications. IBM Research.

[18] Shor, P.W. (1995) Scheme for Reducing Decoherence in Quantum Computer Memory. *Physical Review A*, **52**, R2493-R2496. https://doi.org/10.1103/physreva.52.r2493

[19] Gottesman, D. (1997) Stabilizer Codes and Quantum Error Correction. Ph.D. Thesis, California Institute of Technology.

[20] Aaronson, S. (2015) Read the Fine Print. *Nature Physics*, **11**, 291-293. https://doi.org/10.1038/nphys3272

[21] Jozsa, R. (1998) Quantum Algorithms and the Fourier Transform. *Proceedings of the Royal Society of London. Series A*: *Mathematical*, *Physical and Engineering Sciences*, **454**, 323-337. https://doi.org/10.1098/rspa.1998.0163

[22] Tang, E. (2019) A Quantum-Inspired Classical Algorithm for Recommendation Systems. *Proceedings of the* 51*st Annual ACM SIGACT Symposium on Theory of Computing*, Phoenix, 23-26 June 2019, 217-228. https://doi.org/10.1145/3313276.3316310

[23] Gilyén, A., Lloyd, S. and Tang, E. (2018) Quantum-Inspired Low-Rank Stochastic Regression with Logarithmic Dependence on the Dimension. arXiv:1811.04909.

[24] Ambainis, A. (2004) Quantum Search Algorithms. *ACM SIGACT News*, **35**, 22-35. https://doi.org/10.1145/992287.992296

[25] Montanaro, A. (2016) Quantum Algorithms: An Overview. *npj Quantum Information*, **2**, Article 15023. https://doi.org/10.1038/npjqi.2015.23

[26] Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., *et al.* (1995) Elementary Gates for Quantum Computation. *Physical Review A*, **52**, 3457-3467. https://doi.org/10.1103/physreva.52.3457

[27] Giovannetti, V., Lloyd, S. and Maccone, L. (2008) Quantum Random Access Memory. *Physical Review Letters*, **100**, Article 160501. https://doi.org/10.1103/physrevlett.100.160501

[28] Arunachalam, S. and de Wolf, R. (2017) Guest Column: A Survey of Quantum Learning Theory. *ACM SIGACT News*, **48**, 41-67.

https://doi.org/10.1145/3106700.3106710

[29] Zalka, C. (1999) Grover's Quantum Searching Algorithm Is Optimal. *Physical Review A*, **60**, 2746-2751. https://doi.org/10.1103/physreva.60.2746

[30] Deutsch, D. (1985) Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, **400**, 97-117. https://doi.org/10.1098/rspa.1985.0070

[31] Deutsch, D. and Jozsa, R. (1992) Rapid Solution of Problems by Quantum Computation. *Proceedings of the Royal Society of London. Series A*: *Mathematical and Physical Sciences*, **439**, 553-558. https://doi.org/10.1098/rspa.1992.0167

[32] Simon, D.R. (1997) On the Power of Quantum Computation. *SIAM Journal on Computing*, **26**, 1474-1483. https://doi.org/10.1137/s0097539796298637

[33] Cleve, R., Ekert, A., Macchiavello, C. and Mosca, M. (1998) Quantum Algorithms Revisited. *Proceedings of the Royal Society of London. Series A*: *Mathematical*, *Physical and Engineering Sciences*, **454**, 339-354. https://doi.org/10.1098/rspa.1998.0164

[34] Cormen, T.H., *et al.* (2022) Introduction to Algorithms. 4th Edition, MIT Press.

[35] Sutton, R.S. and Barto, A.G. (2018) Reinforcement Learning: An Introduction. MIT Press.

[36] Bellman, R.E. (1957) Dynamic Programming. Princeton University Press.

[37] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., *et al.* (2015) Human-Level Control through Deep Reinforcement Learning. *Nature*, **518**, 529-533. https://doi.org/10.1038/nature14236

[38] Shor, P.W. (1997) Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, **26**, 1484-1509. https://doi.org/10.1137/s0097539795293172

[39] Feynman, R.P. (1982) Simulating Physics with Computers. *International Journal of Theoretical Physics*, **21**, 467-488. https://doi.org/10.1007/bf02650179

[40] McArdle, S., Endo, S., Aspuru-Guzik, A., Benjamin, S.C. and Yuan, X. (2020) Quantum Computational Chemistry. *Reviews of Modern Physics*, **92**, Article 015003. https://doi.org/10.1103/revmodphys.92.015003

[41] Harrow, A.W., Hassidim, A. and Lloyd, S. (2009) Quantum Algorithm for Linear Systems of Equations. *Physical Review Letters*, **103**, Article 150502. https://doi.org/10.1103/physrevlett.103.150502

[42] Garey, M.R. and Johnson, D.S. (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman.

[43] Aaronson, S. (2004) Limits on Efficient Computation in the Physical World. Ph.D. Thesis, University of California.

[44] Fortnow, L. (2009) The Status of the P versus NP Problem. *Communications of the ACM*, **52**, 78-86. https://doi.org/10.1145/1562164.1562186

[45] Liu, Y. (2026) Physical Barriers to Quantum Loop Depth: Why Quantum Algorithms Cannot Use Deep Sequential Operations. *Journal of Quantum Information Science*, **16**, 1300512.