Best Arm Identification with Quantum Oracles

Xuchuang Wang[1], Yu-Zhen Janice Chen[1], Matheus Guedes de Andrade[1], Jonathan Allcock[2], Mohammad Hajiesmaili[1], John C.S. Lui[3], and Don Towsley[1]

[1]College of Information and Computer Sciences, University of Massachusetts, Amherst, Massachusetts, USA
[2]Tencent Quantum Laboratory, Tencent, Hong Kong
[3]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong
{xuchuangwang, yuzhenchen, mguedesdeand, hajiesmaili, towsley}@cs.umass.edu,
jonallcock@tencent.com, cslui@cse.cuhk.edu.hk

Abstract

Best arm identification (BAI) is a key problem in stochastic multi-armed bandits, where K arms each has an associated reward distribution, and the objective is to minimize the number of queries needed to identify the best arm with high confidence. In this paper, we explore BAI using *quantum oracles*. For the case where each query probes only one arm ($m = 1$), we devise a quantum algorithm with a query complexity upper bound of $\tilde{O}(K\Delta^{-1}\log(1/\delta))$, where δ is the confidence parameter and Δ is the reward gap between best and second best arms. This improves on the classical bound by a factor of Δ^{-1}. For the general case where a single query can probe m arms ($1 \leqslant m \leqslant K$) simultaneously, we propose an algorithm with an upper bound of $\tilde{O}((K/\sqrt{m})\Delta^{-1}\log(1/\delta))$, improving by a factor of $\sqrt{m}$ compared to the $m = 1$ case. We also provide query complexity lower bounds for both scenarios, which match the upper bounds up to logarithmic factors, and validate our theoretical results with Qiskit-based simulations.

1 Introduction

Best arm identification (BAI) is a fundamental problem in the bandits and online learning communities (Audibert, Bubeck, and Munos 2010; Bubeck, Munos, and Stoltz 2011; Mannor and Tsitsiklis 2004). Given $K \in \mathbb{N}^+$ arms, each arm k is associated with a reward distribution with unknown mean μ_k, and the goal of BAI is to identify the arm with the largest mean reward, with a confidence of $1 - \delta$, using as few queries as possible. The number of queries required is called the *query complexity*. Each query, in the classical setting, corresponds to the learner pulling (sampling) one arm and observing a reward drawn from the arm's reward distribution. As the learner only cares about finding the best arm, the BAI problem is a pure exploration problem. BAI has many real world applications, such as, clinical trials (Robbins 1952), network routing (Barrachina-Muñoz and Bellalta 2017), and crowdsourcing (Zhou, Chen, and Li 2014).

Recent progress in building quantum computers (Arute et al. 2019; Chow, Dial, and Gambetta 2021) and quantum networks (Wehner, Elkouss, and Hanson 2018; Azuma et al. 2022) has been encouraging, and wide applications of quantum systems are envisaged in the near future. In these quantum systems, BAI problems also emerge. For example, a quantum network may contain multiple channels between source and destination nodes. Among these channels, one may want to determine the "best" one, where "best" may refer, for example, to the channel with the highest fidelity (Liu et al. 2024) or with the lowest noise (Li, Deng, and Zhou 2008). Another example is in distributed quantum computing (Cacciapuoti et al. 2019), where different quantum computers may have different performances when applied to the same problem, and one wants to identify the quantum computer that provides the best performance for a given task. Although one can still apply classical BAI algorithms to address these problems, we aim to show that the quantum information feedback from these quantum systems can be leveraged to improve the learning efficiency.

In this paper, we study the BAI problem in quantum systems, where the learner can query the arms using quantum queries. More specifically, we study two key advantages of quantum feedback in BAI: (1) quantum parallelism (Chuang and Yamamoto 1995), and (2) quantum entanglement (Einstein, Podolsky, and Rosen 1935). The quantum Monte Carlo estimator (parallelism, Lemma 1) provides a more efficient estimator for the learner to estimate arm rewards. Additionally, multi-qubit oracles with entangled quantum superposition inputs enable the learner to query multiple arms simultaneously (coherently) within a single query. We model the former advantage by *weak quantum oracles*, one for each arm, and the latter by a *constrained quantum oracle* which can query several arms coherently (both detailed in Section 2.2). When a constrained oracle can query all arms coherently, we call it a *strong quantum oracle*.

The development of effective algorithms for both oracles necessitates the use of quantum computing to manage quantum information feedback and leverage quantum parallelism and entanglement. However, obtaining a valid output from a quantum computing subroutine, such as amplitude amplification (Brassard et al. 2002), typically demands multiple consecutive queries on the same arm or a subset of arms for the constrained oracle. This characteristic renders the classical BAI algorithm design and analysis ineffective for BAI with quantum oracles. Consequently, it is imperative to contemplate new algorithm designs and analyses for BAI with quantum oracles. On the other hand, to investigate the fundamental limit of quantum BAI problems, we need to establish query complexity lower bounds. However,

Oracle	**Lower Bound**	**Upper Bound**
Classical (2)	$\Omega\left(\sum_k \frac{1}{\Delta_k^2} \log \frac{1}{\delta}\right)$ (Mannor and Tsitsiklis 2004)	$O\left(\sum_k \frac{1}{\Delta_k^2} \log \frac{1}{\delta}\right)$ (Karnin, Koren, and Somekh 2013)
Strong quantum (5)	$\Omega\left(\sqrt{\sum_k \frac{1}{\Delta_k^2}}\right)$ (Wang et al. 2021)	$\tilde{O}\left(\sqrt{\sum_k \frac{1}{\Delta_k^2}} \log\left(\frac{1}{\delta}\right)\right)$ (Wang et al. 2021)
Weak quantum (3)	$\Omega\left(\sum_k \frac{1}{\Delta_k} \log\left(\frac{1}{\delta}\right)\right)$	$\tilde{O}\left(\sum_k \frac{1}{\Delta_k} \log\left(\frac{1}{\delta}\right)\right)$
Constrained quantum (4)	$\Omega\left(\sum_{\mathcal{S}\in\mathfrak{B}} \sqrt{\sum_{k\in\mathcal{S}} \frac{1}{\Delta_k^2}}\right)$	$\tilde{O}\left(\sum_{\mathcal{S}\in\mathfrak{B}} \sqrt{\sum_{k\in\mathcal{S}} \frac{1}{\Delta_k^2}} \log\left(\frac{1}{\delta}\right)\right)$

Table 1: Comparison of query complexity bounds with classical and quantum BAI

given that quantum information (including parallelism and entanglement) offers more informative and inherently different query feedback than classical BAI, the classical proofs of BAI complexity lower bound are not applicable. Instead, one needs to adapt quantum computation and quantum information approaches to examine quantum BAI problems. Additionally, to empirically validate the performance of devised algorithms for BAI with quantum oracles, one has to utilize quantum circuits (Nielsen and Chuang 2002) and implement the necessary quantum computation subroutines using basic quantum logic gates.

We summarize the key contributions of this paper as follows:

- For BAI with the weak quantum oracle, we derive a query complexity lower bound $\Omega\left(\sum_k (1/\Delta_k) \log(1/\delta)\right)$, showing that no quantum algorithm can achieve a smaller query complexity. Then, we propose an elimination-based quantum algorithm (Q-Elim) and derive its query complexity upper bound $\tilde{O}\left(\sum_k (1/\Delta_k) \log(1/\delta)\right)$, where the suboptimality gap $\Delta_k := \mu_1 - \mu_k$ is the difference in the mean rewards of the optimal arm and arm k, and $\tilde{O}(\cdot)$ hides poly-logarithmic factors. This implies that Q-Elim is near-optimal up to logarithmic factors for BAI with the weak quantum oracle (Section 3).
- For BAI with the m-constrained quantum oracle, we propose a partition-based quantum algorithm (Q-Part), and derive its query complexity upper bound $\tilde{O}(\sum_{\mathcal{S}\in\mathfrak{B}} \sqrt{\sum_{k\in\mathcal{S}} 1/\Delta_k^2} \log(1/\delta))$, where $\mathfrak{B}$ is a partition of the full arm set, i.e., a set of arm subsets, each subset $\mathcal{S}$ containing m arms. We also derive a query complexity lower bound of $\Omega(\sum_{\mathcal{S}\in\mathfrak{B}} \sqrt{\sum_{k\in\mathcal{S}} 1/\Delta_k^2})$ for the partition algorithm class, which matches the upper bound of Q-Part up to logarithmic factors (Section 4).
- We implement our quantum algorithms using the IBM Qiskit (Qiskit contributors 2023). We first corroborate the superiority of our quantum algorithms over classical BAI algorithms. We then evaluate our algorithms under simulated quantum noise (Section 5).

Related Works Prior works on multi-armed bandits (MAB) typically focus on regret minimization and BAI. This paper focuses on the BAI setting (Even-Dar, Mannor, and Mansour 2002; Even-Dar et al. 2006; Mannor and Tsitsiklis 2004). The BAI setting can be divided into two categories: (1) BAI with fixed confidence—find the best arm with a confidence of at least $1-\delta$ ($\delta \in (0,1)$) using as few samples as possible (Bubeck, Munos, and Stoltz 2011); and (2) BAI with fixed budget—given a fixed budget of Q queries, find the best arm with as high a probability as possible (Karnin, Koren, and Somekh 2013). In this paper, we focus on the former category which, for brevity, we will refer to simply as the BAI problem. BAI with the strong quantum oracle was first studied by Casalé et al. (2020); Wang et al. (2021), where they proposed a near-optimal quantum algorithm that enjoys a quadratic speedup in query complexity. We are the first to study the BAI problem with the weak and constrained quantum oracles. Besides BAI, another objective, regret minimization in bandit theory, has also been studied with quantum oracles, including Wan et al. (2023); Dai et al. (2023); Wu et al. (2023), etc. We defer a more detailed discussion of these works and other loosely related works to Appendix A.

In Table 1, we summarize the key results in this paper and compare them to prior works. Comparing the Δ_k-dependence of the complexities, we have

$$\underbrace{\sqrt{\sum_k \frac{1}{\Delta_k^2}}}_{\text{Strong oracle}} \leqslant \underbrace{\sum_{\mathcal{S}\in\mathfrak{B}} \sqrt{\sum_{k\in\mathcal{S}} \frac{1}{\Delta_k^2}}}_{m\text{-constrained oracle}} \leqslant \underbrace{\sum_k \frac{1}{\Delta_k}}_{\text{Weak oracle}} \leqslant \underbrace{\sum_k \frac{1}{\Delta_k^2}}_{\text{Classical oracle}} \tag{1}$$

All BAI problems with quantum oracles enjoy smaller query complexities than the classical one. The query complexity of the weak quantum oracle is worst among quantum oracles, which is due to the fact that the weak oracle cannot exploit quantum entanglement to probe multiple arms in parallel. The query complexity of the m-constrained quantum oracle lies between that of strong and weak oracles, and when $m = 1$ (resp., $m = K$) the complexity coincides with that of weak (resp., strong) oracles.

2 Model

2.1 Preliminaries

Best arm identification (BAI). Consider a multi-armed bandit (MAB) consisting of K arms, where each arm $k \in \mathcal{K} := \{1, 2, \ldots, K\}$ is associated with a Bernoulli distribution $\mathcal{B}(\mu_k)$ with mean $\mu_k \in (0,1)$.[1] An MAB instance is determined by the mean rewards of its arms, and we denote an instance $\mathcal{I}$ with means $\mu_1, \ldots, \mu_K$ as $\mathcal{I} := \{\mu_1, \ldots, \mu_K\}$.

[1] More general distributions, such as sub-Gaussian or bounded distributions, have also been considered in the MAB literature (Auer and Ortner 2010; Lattimore and Szepesvári 2020).

For simplicity, we assume the K arms are labeled in descending order of their means: $\mu_1 > \mu_2 \geqslant \ldots \geqslant \mu_K$, unknown to the learner, and denote the mean reward (suboptimality) gap as $\Delta_k \coloneqq \mu_1 - \mu_k$ for suboptimal arms $k > 1$ and $\Delta_1 \coloneqq \Delta_2$ for the optimal arm. We assume a unique optimal arm for the simplicity of the later presentation of the algorithms and analysis. One could extend the results to multiple optimal arms with techniques in bandits literature, e.g., find an ϵ-optimal arm (Even-Dar, Mannor, and Mansour 2002). Then, given confidence parameter $\delta \in (0, 1)$, the best arm identification (`BAI`) problem is to correctly output the best arm with a probability of at least $1 - \delta$ using as few queries as possible, noted as the query complexity Q.

Next, we present some basics notation from quantum computation and information (Nielsen and Chuang 2002).

Bra-ket notation. We make use of bra-ket notation to represent quantum states, where the "ket" $|x\rangle \coloneqq (x_1, x_2, \ldots, x_n)^T \in \mathbb{C}^n$ denotes a column vector of n complex numbers, while the "bra" $\langle x| \coloneqq |x\rangle^\dagger = (x_1^*, x_2^*, \ldots, x_n^*)$, a row vector, is the conjugate transpose of $|x\rangle$. For two quantum states $|x\rangle, |y\rangle \in \mathbb{C}^n$, their inner product is denoted as $\langle x|y\rangle \coloneqq \sum_{i=1}^n x_i^* y_i \in \mathbb{C}$, and given another quantum state $|z\rangle \in \mathbb{C}^m$, the tensor product between $|x\rangle$ and $|z\rangle$ is denoted as $|x\rangle |z\rangle = |x\rangle \otimes |z\rangle \coloneqq (x_1 z_1, x_1 z_2, \ldots, x_n z_m) \in \mathbb{C}^n \otimes \mathbb{C}^m$.

Qubit. A "qubit" is a two-level quantum system $|\phi\rangle = (\alpha, \beta) \in \mathbb{C}^2$, often written as $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $|0\rangle = (1, 0)^T$ and $|1\rangle = (0, 1)^T$ are two basis states, and $\alpha, \beta \in \mathbb{C}$ are complex numbers, called amplitudes, satisfying $|\alpha|^2 + |\beta|^2 = 1$. A measurement of the qubit in the $\{|0\rangle, |1\rangle\}$ basis will give a '0' with probability $|\alpha|^2$ and a '1' with probability $|\beta|^2$.

Quantum query model. In the quantum query model, one has access to a black-box unitary operator (i.e., oracle) which implements a given transformation. The objective is to study the *query complexity*, i.e., the number of calls Q to the oracle needed to solve a given task; all other possible costs, e.g., gate complexity, are ignored. This is a commonly used model for studying quantum algorithms (Childs 2017, §20) and can be used, for instance, to obtain algorithmic running time lower bounds (Klauck, Špalek, and De Wolf 2007). In this paper, we study the query complexity of best arm identification with fixed confidence under weak and constrained quantum oracles.

2.2 Quantum Oracles

Before introducing the quantum oracles, we first recall the classical oracle for the `BAI` problem. That is, when querying an arm k, one obtains a reward drawn from a Bernoulli distribution $\mathcal{B}(\mu_k)$ with *unknown* mean μ_k, i.e.,

$$X_k \sim \mathcal{B}(\mu_k). \tag{2}$$

We refer to (2) as the *classical oracle*.

In the quantum setting, the Bernoulli distributions can be mapped to oracles $O_{\text{weak}}^{(k)}$ (one for each k) that act as follows,

$$O_{\text{weak}}^{(k)} : |0\rangle_R \mapsto \sqrt{1-\mu_k}\,|0\rangle_R + \sqrt{\mu_k}\,|1\rangle_R\,, \tag{3}$$

where the register $|\cdot\rangle_R$ represents a single-qubit "bandit reward" register with basis states $|0\rangle$ and $|1\rangle$. The output qubit encodes the Bernoulli reward, meaning that if one measures the output in the basis $\{|0\rangle, |1\rangle\}$, the probability of observing $|1\rangle$ is μ_k, while the probability of observing $|0\rangle$ is $1-\mu_k$. We refer to (3) as the *weak quantum oracle*.

Note that directly measuring the output qubits reduces the weak oracle to a Bernoulli distribution. However, aside from direct measurement, the output qubits enable efficient quantum parallelism through quantum computing algorithms, which we elaborate in Section 3.

To harness the entanglement properties of quantum information in real-world quantum systems, we consider a more general quantum oracle that allows simultaneous querying of multiple arms. In addition to the reward register $|\cdot\rangle_R$, we introduce an "arm index" register $|\cdot\rangle_I$, which has K orthogonal basis states $\{|k\rangle_I\}_{k=1}^K$, each corresponding to an arm. A quantum state in the $|\cdot\rangle_I$ register can be expressed as $\sum_{k=1}^K a_k |k\rangle_I$, where $a_k \in \mathbb{C}$ are the amplitudes of the arms, and normalization requires that $\sum_{k=1}^K |a_k|^2 = 1$.

With the assistance of the arm index register, we define a *constrained quantum oracle* that outputs states entangling the arm index and reward registers. Assuming the oracle can access $m \in \{1, 2, \ldots, K\}$ arms simultaneously, for any subset of arms $\mathcal{S} \subseteq \mathcal{K}$ with $|\mathcal{S}| = m$ and $\sum_{k\in\mathcal{S}} |a_k|^2 = 1$, the oracle is defined as follows:

$$\begin{aligned} O_{\text{cons}}^{(\mathcal{S})} : &\sum_{k\in\mathcal{S}} a_k |k\rangle_I |0\rangle_R \\ &\mapsto \sum_{k\in\mathcal{S}} a_k |k\rangle_I \left(\sqrt{1-\mu_k}\,|0\rangle_R + \sqrt{\mu_k}\,|1\rangle_R\right). \end{aligned} \tag{4}$$

when $m = 1$, the oracle reduces to the weak quantum oracle in (3), and when $m = K$, it becomes the strong quantum oracles as follows,

$$\begin{aligned} O_{\text{stro}} : &\sum_{k=1}^{K} a_k |k\rangle_I |0\rangle_R \\ &\mapsto \sum_{k=1}^{K} a_k |k\rangle_I \left(\sqrt{1-\mu_k}\,|0\rangle_R + \sqrt{\mu_k}\,|1\rangle_R\right). \end{aligned} \tag{5}$$

The constrained quantum oracle in (4) is more powerful than the weak oracle in (3) because it can access multiple arms coherently in a single query, whereas the weak oracle only allows access to one arm at a time. In Section 4, we present a `BAI` algorithm using the m-constrained oracle, which outperforms the weak oracle when $m > 1$.

In practice, coherently querying a large number of channels may be technologically challenging, which motivates the general $m \leqslant K$ case. This limitation reflects a technology constraint where more options exist than can be accessed simultaneously. Such technological constraints may also affect, for example, access to quantum states stored in memory. In this case, a weak oracle would support individual calls to memory, while an m-constrained oracle functions like a dynamically loadable quantum random access memory (QRAM, see Appendix B), capable of querying multiple entries at once.

3 BAI with Weak Quantum Oracle

In this section, we address the BAI problem using a weak quantum oracle as described in (3). Querying this oracle for arm k yields the state $\sqrt{1-\mu_k}\,|0\rangle + \sqrt{\mu_k}\,|1\rangle$. To estimate μ_k efficiently, we use the following lemma:

Lemma 1 (Performance of QuEst, adapted from Montanaro (2015); Grinko et al. (2021))**.** *For a weak quantum oracle $O^{(k)}_{\text{weak}}$ in* (3)*, there exists a constant $C_1 > 1$ and a quantum estimation algorithm* QuEst$(O^{(k)}_{\text{weak}}, \epsilon, \delta)$ *that estimates μ_k with precision ϵ and confidence δ (i.e., $\mathbb{P}(|\hat{\mu}_k - \mu_k| \geqslant \epsilon) \leqslant \delta$), using at most $\frac{C_1}{\epsilon}\log\frac{1}{\delta}$ queries.*

This quantum estimator QuEst achieves a quadratic speedup over the classical estimators that require $O((1/\epsilon^2)\log(1/\delta))$ queries. Unfortunately, QuEst lacks flexibility: it does not generate any information before the entire procedure has completed, unlike classical estimators that improve estimates incrementally during the samples arriving and allows for sample reuse.

To address this issue, we first use QuEst to develop a batch-based elimination algorithm for BAI with the weak quantum oracle in Section 3.1. We then establish an upper bound on the query complexity of this algorithm in Section 3.2. Finally, in Section 3.3, we present a lower bound for any BAI algorithm using a weak quantum oracle, highlighting the fundamental limits of the task.

3.1 Algorithm Design

Algorithm 1 presents a quantum elimination algorithm (Q-Elim) for BAI. The core idea of the elimination process is to maintain a candidate arm set C (initially set to the full arm set $\mathcal{K}$), gradually identify and remove suboptimal arms from C as learning progresses, and terminate when C contains only one arm, which is then declared as the optimal arm.

Although several classical elimination algorithms, such as successive elimination (Even-Dar et al. 2006), have been proposed for BAI using classical oracles, these cannot be directly adapted by simply replacing classical estimators with the quantum estimator from Lemma 1 due to the rigidity of the quantum estimator (one cannot acquire any information from QuEst before the entire procedure completed).

A significant challenge in designing our quantum algorithm is determining when to perform quantum estimation QuEst and arm elimination. We address this by proposing a batch-based exploration and elimination scheme, where $j \in \{1, 2, \dots\}$ denotes the batch number. In each batch, we query all remaining arms in the candidate arm set C a number of times depending on the batch number j (Line 2), conduct QuEst $\left(O^{(k)}_{\text{weak}}, 2^{-j}, \frac{\delta}{2^j|C|}\right)$ to estimate the mean rewards of arms in C based on the queries from this batch (Line 3), and eliminate newly identified suboptimal arms (Line 5) at the end of the batch. As j increases, we progressively increase both the number of queries (Line 6) and the estimation accuracy of QuEst (Lines 2 and 3).

Algorithm 1 Q-Elim: Quantum elimination algorithm for BAI with weak quantum oracle

```
Input: fixed confidence parameter $\delta$ and number of arms $K$
Initialize: empirical mean $\hat{\mu}_k \leftarrow 0$, candidate arm set $C \leftarrow \mathcal{K}$, batch number $j \leftarrow 1$
1: while $|C| > 1$ do
2:     Query each arm $k \in C$ for $C_1 2^j \log(2^j|C|/\delta)$ times
3:     Run QuEst $\left(O^{(k)}_{\text{weak}}, 2^{-j}, \frac{\delta}{2^j|C|}\right)$ for each arm $k$ in $C$ and update these arms' estimates $\hat{\mu}_k$
4:     $\hat{\mu}_{\max} \leftarrow \max_{k \in C} \hat{\mu}_k$
5:     $C \leftarrow C \setminus \{k \in C : \hat{\mu}_k + 2 \cdot 2^{-j} \leqslant \hat{\mu}_{\max}\}$     ▷ Arm elimination
6:     $j \leftarrow j + 1$
Output: the single remaining arm in $C$.
```

3.2 Query Complexity Upper Bound for Elimination Algorithm

Theorem 1 (Query complexity upper bound of Algorithm 1)**.** *Given confidence parameter $\delta \in (0, 1)$, the query complexity of* Q-Elim *is upper bounded as follows,*

$$Q \leqslant \sum_{k \in \mathcal{K}} \log_2\left(\frac{4}{\Delta_k}\right) \frac{16C_1}{\Delta_k} \log\frac{K}{\delta},$$

where log *is the natural logarithm, and* $\log_2$ *is the logarithmic function base* 2*.*

Comparison with the query complexity lower bound in Theorem 2 shows that our upper bound in Theorem 1 is *tight up to logarithmic factors*. Compared to the classical oracle sample complexity upper bound of $O(\sum_{k \in \mathcal{K}}(1/\Delta_k)^2 \log(1/\delta))$ (Karnin, Koren, and Somekh 2013), the query complexity upper bound in Theorem 1 has a quadratic improvement in the dependence on $1/\Delta_k$ for each individual arm. In contrast, the strong quantum oracle sample complexity upper bound $\tilde{O}(\sqrt{\sum_k 1/\Delta_k^2}\log(1/\delta))$ (Wang et al. 2021) achieves an overall quadratic speedup. That is, as the first inequality of (1) shows, the coefficient of the query complexity lower bound of the weak quantum oracle is larger than that of the strong oracle, and is, in the worst case, $\sqrt{K}$ times larger.

3.3 Lower Bounds for BAI with weak quantum oracle

Lastly, we present a query complexity lower bound for BAI with a weak quantum oracle. This lower bound describes the fundamental limits of the BAI task with a weak quantum oracle and is independent of the specific algorithm used.

Theorem 2 (Query complexity lower bound for best arm identification)**.** *Given a quantum multi-armed bandits instance $\mathcal{I}_0 = \{\mu_1, \dots, \mu_K\}$ where $\mu_k \in (0, 1/2)$ for all k and $\mu_1 > \mu_2 \geqslant \mu_k$ for any $k \neq 1$, any algorithm that identifies the optimal arm with a given confidence $1 - \delta$, $\delta \in (0, 1)$ requires Q queries to the weak quantum oracle, where*

$$Q \geqslant \sum_{k \in \mathcal{K}} \frac{1}{4\Delta_k} \log\frac{1}{4\delta}.$$

Thus, to identify the best arm with confidence $1-\delta$, it is *necessary* to pull *each* arm k at least $1/(4\Delta_k)\log 1/(4\delta)$ times. The proof of this lower bound consists of two steps: (1) apply the quantum hypothesis testing techniques to prove a lower bound for the task of two arm identification, and (2) extend the lower bound of the two-arm case to multiple arms via adapting the lower bound proof of the classical best arm identification. The detailed proof is presented in Appendix F.

First, Theorem 2 demonstrates that the query complexity of `Q-Elim`, as established in Theorem 1, is near-optimal (up to some logarithm factors). Compared to the classical oracle's sample lower bound $\Omega\left(\sum_{k\in\mathcal{K}}\frac{1}{\Delta_k^2}\log\frac{1}{\delta}\right)$ (Mannor and Tsitsiklis 2004), our lower bound shows a linear dependence on $1/\Delta_k$ rather than quadratic. When compared to the strong quantum oracle's sample complexity lower bound $\Omega\left(\sqrt{\sum_k\frac{1}{\Delta_k^2}}(1-\sqrt{\delta(1-\delta)})\right)$ (Wang et al. 2021, Theorem 5), the weak oracle's query complexity lower bound has a larger coefficient, which can be up to $\sqrt{K}$ times greater in the worst case. However, our lower bound improves on the dependence on δ, as $\log(1/\delta)$ is significantly larger than $1-\sqrt{\delta(1-\delta)}$ when δ is small.

4 `BAI` with m-Constrained Quantum Oracle

In this section, we present a partition algorithm for `BAI` with the m-constrained quantum oracle. We first present some key subroutines in Section 4.1 on quantum computing, and then present our algorithm in Section 4.2, followed by the algorithm's query complexity upper bound in Section 4.3, as well as a lower bound for any partition algorithms in Section 4.4.

4.1 Key Quantum Subroutines

Variable-Time Algorithm Construction The variable-time algorithm of Ambainis (2010); Wang et al. (2021) can be used to transform an m-constrained quantum oracle with a reward register $|\cdot\rangle_R$ into an oracle (`VTA`) that outputs a state with a flag register $|\cdot\rangle_F$ which distinguishes arms with large mean rewards from other arms. For an m-constrained oracle $O_{\text{cons}}^{(\mathcal{S})}$ and a subset $\mathcal{S}$, `VTA` takes an interval $I=[a,b]$ with $0<a<b<1$ and a parameter $\alpha\in(0,1)$ as inputs. It divides $\mathcal{S}$ into three subsets: $\mathcal{S}_{\text{right}}:=\{k\in\mathcal{S}:\mu_k\geqslant b-\frac{b-a}{8}\}$ (high rewards); $\mathcal{S}_{\text{left}}:=\{k\in\mathcal{S}:\mu_k<b-\frac{b-a}{2}\}$ (low rewards); $\mathcal{S}_{\text{middle}}:=\mathcal{S}\setminus(\mathcal{S}_{\text{right}}\cup\mathcal{S}_{\text{left}})$ (intermediate rewards). The output state is:

$$\texttt{VTA}(O_{\text{cons}}^{(\mathcal{S})},\mathcal{S},I=[a,b],\alpha):\frac{1}{\sqrt{m}}\sum_{k\in\mathcal{S}}|k\rangle_I|1\rangle_F\to$$
$$\frac{1}{\sqrt{m}}\left(\sum_{k\in\mathcal{S}_{\text{right}}}|k\rangle_I|1\rangle_F+\sum_{k\in\mathcal{S}_{\text{left}}}|k\rangle_I|0\rangle_F+\sum_{k\in\mathcal{S}_{\text{middle}}}|k\rangle_I|\phi_k\rangle_F\right),\tag{6}$$

where $|\cdot\rangle_F$ indicates the subsets $\mathcal{S}_{\text{right}}$ and $\mathcal{S}_{\text{left}}$ with $|1\rangle_F$ and $|0\rangle_F$ respectively. Arms in $\mathcal{S}_{\text{middle}}$ are represented by $|\phi_k\rangle_F$, with specific states depending on α and the `MAB` instance. The probability of observing $|1\rangle_F$ is $p_{\text{good}}:=\frac{1}{m}\left(\left|\mathcal{S}_{\text{right}}\right|+\sum_{k\in\mathcal{S}_{\text{middle}}}|\beta_k|^2\right)$, where β_k depends on $|\phi_k\rangle_F$. The algorithm's pseudocode is in Appendix C.1.

Amplitude amplification (`Amplify`) and amplitude estimation (`Estimate`) The `Amplify` and `Estimate` are two fundamental quantum computing algorithms (Brassard et al. 2002). `Amplify` enhances the amplitude of a target basis state, while `Estimate` estimates the amplitude of that state. Since these algorithms are well-established, we omit their pseudocode and direct interested readers to Brassard et al. (2002) for details. In this work, we apply both algorithms with the `VTA` oracle in (6), using $|1\rangle_F$ as the target state. The performance of `Amplify` and `Estimate` in this context is discussed in Lemma 5 in Appendix C.1.

Good Ratio (`GoodRatio`) Subroutine The good ratio subroutine is based on the variable-time algorithm (`VTA`) and amplitude estimation (`Estimate`). It takes an m-constrained oracle $O_{\text{cons}}^{(\mathcal{S})}$ for a subset of arms $\mathcal{S}$, an interval $I=[a,b]$, and a confidence parameter δ as inputs, and outputs an estimate of the ratio of "good arms" in $\mathcal{S}$, where the "good arms" are the arms with mean reward greater than a in the interval I. The subroutine is detailed in Algorithm 4 in Appendix C.2. Lemma 2 provides the subroutine's performance guarantees.

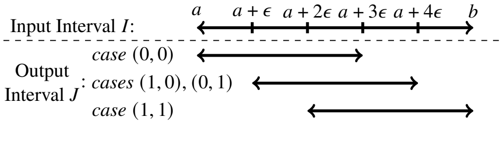
Lemma 2 (Performance of `GoodRatio`)**.** *Given an interval $I=[a,b]$ and a confidence parameter $0<\delta<1$, there exists a* `GoodRatio`$(O_{cons}^{(\mathcal{S})},\mathcal{S},I=[a,b],\delta)$ *subroutine which uses $O(G)$ queries to output an estimate $\hat{p}_{\text{good}}$ of the "good arm" ratio p_{good} such that*

$$0.9\left(p_{\text{good}}-\frac{0.1}{m}\right)<\hat{p}_{\text{good}}<1.1\left(p_{\text{good}}+\frac{0.1}{m}\right)$$

with probability at least $1-\delta$, where the parameter $G:=\sqrt{\frac{1}{(b-a)^2}+\frac{1}{|\mathcal{S}_{\text{right}}|}\sum_{k\in\mathcal{S}_{\text{left}}\cup\mathcal{S}_{\text{middle}}}\frac{1}{(b-\mu_k)^2}}\operatorname{polylog}\left(\frac{m}{\delta(b-a)}\right)$.

Lemma 2 guarantees that `GoodRatio` provides a good estimate of the ratio of good arms in the subset $\mathcal{S}$ with high probability and with in a reasonable number of queries.

Partition Shrink (`PartShrink`) Subroutine The partition shrink subroutine takes as input the m-constrained oracles $O_{\text{cons}}^{(\mathcal{S})}$ for each subset $\mathcal{S}$ in the partition set $\mathfrak{B}$, the partition set $\mathfrak{B}$ itself, an interval I, and parameters $h\in\{1,2\}$ and $\delta\in(0,1)$. The parameter $h=1$ (resp. $h=2$) directs the algorithm to shrink the input interval I so that the best arm μ_1 (resp. the second best arm μ_2) lies inside the output interval J. Utilizing a technique from quantum ground state preparation (Lin and Tong 2020), `PartShrink` divides the input interval $I=[a,b]$ into five sub-intervals of equal length and outputs a new interval J consisting of three consecutive sub-intervals, as illustrated below:



Which case the output interval J above corresponds to depends on the input parameters and the mean reward

Algorithm 2 Q-Part: Partition Algorithm for BAI with coherent query constrained m

```
Input: full arm set $\mathcal{K}$, confidence parameter $\delta$, constraint $m$
Initialize: $\delta \leftarrow \delta/2, I_1, I_2 \leftarrow [0,1], \delta' \leftarrow \delta$
 1: Partition the full arm sets to $\lceil K/m \rceil$ subsets, each with
    $m$ arms, together denoted as a set $\mathfrak{B}$
    ▷ Stage (i): identify best arm subset
 2: while $\min I_1 - \max I_2 < 2|I_1|$ or $|\mathfrak{B}| > 1$ do
 3:     $I_1 \leftarrow$ PartShrink $\left((O^{(\mathcal{S})}_{\text{cons}})_{\forall \mathcal{S} \in \mathfrak{B}}, \mathfrak{B}, I_1, 1, \delta'\right)$
 4:     $I_2 \leftarrow$ PartShrink $\left((O^{(\mathcal{S})}_{\text{cons}})_{\forall \mathcal{S} \in \mathfrak{B}}, \mathfrak{B}, I_2, 2, \delta'\right)$
 5:     for $\mathcal{S} \in \mathfrak{B}$ do
 6:         if GoodRatio $\left(O^{(\mathcal{S})}_{\text{cons}}, I_1, \delta'\right) = 0$ then
                ▷ If no good arm inside subset $\mathcal{S}$
 7:             $\mathfrak{B} \leftarrow \mathfrak{B} \setminus \mathcal{S}$   ▷ Subset elimination
 8:     $\delta' \leftarrow \delta'/2$ ▷ Halve confidence parameter
 9: $\ell_1 \leftarrow \min I_1, \ell_2 \leftarrow \max I_2$
10: $\mathcal{S} \leftarrow \mathfrak{B}$     ▷ Only remaining subset in $\mathfrak{B}$
    ▷ Stage (ii): identify best arm
11: Construct variable-time quantum algorithm $\mathcal{A} \leftarrow$
    VTA$(O^{(\mathcal{S})}_{\text{cons}}, \mathcal{S}, I = [\ell_2, \ell_1], 0.01\delta)$
12: $k \leftarrow$ Amplify$(\mathcal{A}, \delta')$
Output: arm $k$
```

of the arms in the interval I. We refer the detail to the PartShrink subroutine in Algorithm 5 in Appendix C.3. The performance guarantees are provided in Lemma 3.

Lemma 3 (Performance of PartShrink)**.** *Given $h \in \{1,2\}$, an interval $I = [a,b]$, and a confidence parameter $0 < \delta < 1$, supposing $\mu_h \in I$ and $|I| \geqslant \Delta_2/8$, there exists a* PartShrink $\left((O^{(\mathcal{S})}_{cons})_{\forall \mathcal{S} \in \mathfrak{B}}, \mathfrak{B}, I, h, \delta\right)$ *subroutine which*

1. *outputs an interval J with $|J| = 3|I|/5$ such that $\mu_h \in J$ with a probability of at least $1-\delta$, and*
2. *uses $O\left(\sum_{\mathcal{S}\in\mathfrak{B}} \sqrt{\sum_{k\in\mathcal{S}} \frac{1}{\Delta_k^2}} \operatorname{polylog}\left(\frac{K}{m\delta\Delta_2}\right)\right)$ queries.*

Lemma 3 guarantees that PartShrink outputs an interval J containing the mean reward μ_h with high probability and in a reasonable number of queries. The proofs of Lemmas 2 and 3 are presented in Appendix E.1.

Next, we present the partition algorithm for BAI with the m-constrained quantum oracle that builds on the GoodRatio and PartShrink subroutines.

4.2 Algorithm Design

This section presents the partition algorithm (Q-Part in Algorithm 2) using the m-constrained quantum oracle. The algorithm partitions K arms into K/m subsets[2] and queries arms within each subset to find the optimal one.

Initially, Q-Part partitions the K arms into K/m subsets $\mathcal{S}_1, \ldots, \mathcal{S}_{K/m}$ (Line 1), each containing m arms, and denote $\mathfrak{B} := \{\mathcal{S}_1, \ldots, \mathcal{S}_{K/m}\}$. The algorithm has two main stages: (i) identifying the subset containing the optimal arm (Lines 2-8) and (ii) finding the best arm within that subset (Lines 9-12).

To find the optimal arm's subset, Q-Part uses an elimination process. It starts with all subsets in $\mathfrak{B}$ and progressively removes those without the best arm until one remains. The algorithm maintains two intervals, I_1 and I_2, both initialized to $[0,1]$. Within the while loop (Line 2), PartShrink is applied to shrink the intervals I_1 and I_2 (Lines 3-4). Then, GoodRatio checks each remaining subset to see if it contains an arm with a mean reward in I_1. If not, the subset is eliminated (Line 7). The loop ends when only one subset remains, and I_1 and I_2 are separated by a gap of at least $2|I_1|$ (i.e., $\min I_1 - \max I_2 \geqslant 2|I_1|$).

Upon completion of Stage (i), Q-Part identifies the subset containing the best arm, with I_1 containing the mean reward μ_1 of the best arm and I_2 containing the mean reward μ_2 of the second-best arm. The endpoints $\ell_1 = \min I_1$ and $\ell_2 = \max I_2$ separate the best arm from the rest (Line 9). To find the optimal arm, Q-Part uses a variable-time algorithm (VTA) in (6) with the remaining subset $\mathcal{S}$ and interval $[\ell_2, \ell_1]$ as inputs (Line 11), which produces the expected output $\frac{1}{\sqrt{m}}(|k^*\rangle_I |1\rangle_F + \sum_{k\in\mathcal{S}\setminus\{k^*\}} |k\rangle_I |0\rangle_F)$. Amplify then determines the index of the optimal arm k^* (Line 12), which guarantees to output the best arm in the set $\mathcal{S}$ with a probability of at least $1-\delta'$.

4.3 Query Complexity Upper Bound for Partition Algorithm

We derive a query complexity upper bound for Q-Part (Algorithm 2), and its detail proof is deferred to Appendix E.2.

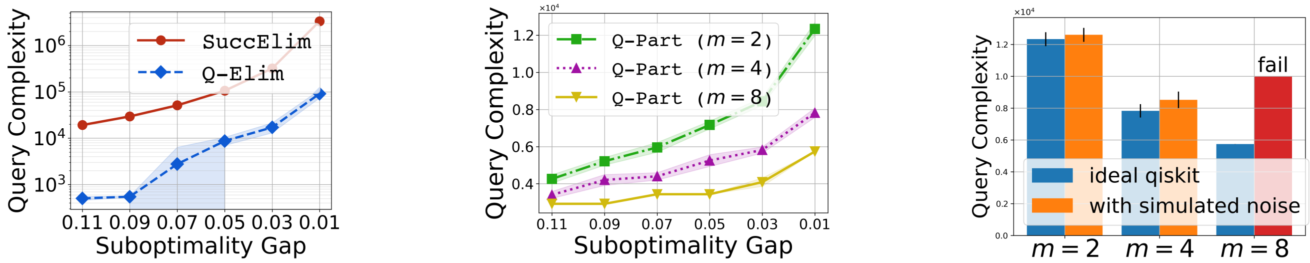
Theorem 3 (Query complexity upper bound for Q-Part of the m-constrained quantum oracle)**.** *With confidence parameter $\delta \in (0,1)$ and an arm partition $\mathfrak{B}$, the query complexity of Algorithm 2 is $O\left(\sum_{\mathcal{S}\in\mathfrak{B}} \sqrt{\sum_{k\in\mathcal{S}} \frac{1}{\Delta_k^2}} \operatorname{polylog}\left(\frac{K}{\delta\Delta_2}\right)\right)$, where $\Delta_k = \mu_1 - \mu_k$ is the reward gap of arm k, and Δ_2 is the minimal reward gap.*

As Δ_2 is the smallest reward gap, Theorem 3 simplifies the upper bound to $\tilde{O}((K/\sqrt{m})\Delta_2^{-1})$. Thus, a smaller m (better coherence) reduces the query complexity. When $m = 1$ (weak quantum oracle), Q-Part's complexity is $\tilde{O}(\sum_{k\in\mathcal{K}} \Delta_k^{-1} \log \frac{1}{\delta})$, which matches Q-Elim's bound for a weak oracle (Theorem 2). However, Q-Elim's bound $O(\log \frac{1}{\Delta} \log \frac{K}{\delta})$ is better than Q-Part's polylogarithmic factor $O(\operatorname{polylog} \frac{K}{\delta\Delta_2})$ (at most $\log^3 \frac{K}{\delta\Delta_2}$) because Q-Elim's parameters are optimized for weak oracles. When $m = K$ (strong quantum oracle), Q-Part reduces to the algorithm by Wang et al. (2021), as no further partitioning is needed ($\mathfrak{B} = \{\mathcal{K}\}$). For $1 < m < K$, Q-Part's complexity lies between that of Wang et al. (2021)'s strong oracle and Q-Elim's weak oracle (see (1)).

4.4 Query Complexity Lower Bounds for the m-Constrained Quantum Oracle

In Section 4.4, we establish lower bounds to demonstrate the tightness and optimality of the Q-Part algorithm. The key

[2] If K/m is not an integer, add n dummy arms (where $n < m$) to make $m \mid (K+n)$.



(a) SuccElim vs. Q-Elim (b) Q-Part with different m (c) Impact of noise on Q-Part

Figure 1: Performance evaluation of Q-Elim and Q-Part. The results for Q-Part are based on taking the constant multiplicative prefactor from Lemma 5 to be 1. In reality this constant may be larger than 1 and thus the results for Q-Part and Q-Elim are not directly comparable. Figure 1c is conducted for gap = 0.01 in a simulated 127-qubit quantum computer.

challenge is proving the lower bound with the outer summation over all subsets in $\mathfrak{B}$ (i.e., $\sum_{\mathcal{S}\in\mathfrak{B}}$). This summation indicates that queries on each subset are "orthogonal", meaning information gained from one subset does not overlap with others. To address this, we define a class of partition algorithms for m-constrained oracles, which ensures that queries on one subset of arms cannot be used to infer information about arms in any of the other subsets. We then derive a lower bound for any partition algorithm, as stated in Theorem 4, with a proof provided in Appendix G.2.

Definition 1 (Partition algorithm class)**.** *A partition algorithm for* BAI *with the m-constrained oracle is one that partitions the full arm set into several subsets at initialization, each with at most m arms, and always follows this fixed partition when querying arms during algorithm execution.*

Theorem 4 (Query complexity lower bound for m-constrained oracle)**.** *To identify the best arm with a probability of at least* $1 - \delta$ *with the m-constrained oracle with parameter m, any* partition algorithm *needs to spend at least the following number of queries,* $\Omega(\sum_{\mathcal{S}\in\mathfrak{B}} \sqrt{\sum_{k\in\mathcal{S}} 1/\Delta_k^2})$, *where* $\mathfrak{B}$ *is the partition of arms.*

Note that Q-Part (Algorithm 2) belongs to this partition algorithm class, and if the arm partition $\mathfrak{B}$ in the lower bound is the same as the one chosen in Q-Part, then this lower bound matches the upper bound for Q-Part in Theorem 3 up to some logarithmic factors. This implies that the bounds in both Theorems 3 and 4 are tight, and Q-Part is near-optimal within the partition algorithm class.

5 Qiskit-based Simulation

We compare the quantum algorithms Q-Elim (for the weak quantum oracle) and Q-Part (for the m-constrained quantum oracle) with the classical successive elimination SuccElim (Even-Dar et al. 2006).

We set $\delta = 0.1$ and $K = 8$ arms with mean rewards ranging from $0.99 - (k - 1) \times \Delta$ (where $k \in \{1, \dots, K\}$) and vary Δ from 0.11 to 0.01 in steps of 0.02 to analyze its effect on query complexity. Details of the Qiskit implementation are in Appendix H, and the code is provided in the supplementary material. We implement Q-Elim with $m = 1$ and Q-Part with $m = 2, 4$, and 8. For $m = 8$, the m-constrained oracle is equivalent to the strong quantum oracle. The default confidence parameter for SuccElim is $c = 4$ (Even-Dar et al. 2006). Results, averaged over 50 trials, are shown in Figure 1.

Figure 1a (with y-axis in log-scale) shows that Q-Elim outperforms SuccElim, demonstrating the benefits of quantum information with the weak oracle. As Δ decreases, SuccElim's query complexity increases faster than Q-Elim's, validating the quantum improvement of dependence on Δ from Δ^{-2} to Δ^{-1} (see Appendix H for curve-fitting). Figure 1b compares Q-Part's performance for $m = 2, 4, 8$ as Δ varies. Increasing m improves performance, confirming the advantage of quantum parallelism predicted by the $\tilde{O}((K/\sqrt{m})\Delta^{-1})$ bound from Theorem 3.

We also assess the impact of noise using Qiskit's *simulation* of IBM's 127-qubit device. Figure 1c shows that Q-Elim's performance decreases by 2.38% and 8.09% for $m = 2$ and $m = 4$, respectively. For m=8, Q-Part fails due to high noise, as the increased qubit and gate requirements exceed practical limits, impairing the algorithm's functionality. This highlights the importance to study the restrictive m-constrained quantum oracles under a noisy environment.

6 Conclusion

In this paper, we explore the best arm identification (BAI) problem using weak and m-constrained quantum oracles. We introduce the m-constrained oracle, which generalizes both the weak oracle ($m = 1$) and the strong oracle ($m = K$). Our quantum algorithms, Q-Elim for the weak oracle and Q-Part for the constrained oracle, offer significant improvements over classical methods. Specifically, Q-Elim achieves a quadratic speedup at the arm level due to quantum parallelism, while Q-Part provides a quadratic speedup at the subset level due to quantum entanglement. We establish query complexity lower bounds for both quantum BAI problems that align with our upper bounds, indicating near-optimal performance. Our experiments using Qiskit confirm these theoretical results.

Acknowledgments

The work of Mohammad Hajiesmaili is supported by NSF CNS-2325956, CAREER-2045641, CPS-2136199, CNS-2102963, and CNS-2106299. The work of John C.S. Lui is supported in part by SRFS2122-4S02. The work of Don Towsley is supported in part by the NSF grant CNS-1955744, NSF-ERC Center for Quantum Networks grant EEC-1941583, and DOE Grant AK000000001829.

References

Ambainis, A. 2000. Quantum lower bounds by quantum arguments. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, 636–643.

Ambainis, A. 2010. Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations. *arXiv preprint arXiv:1010.4458*.

Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J. C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F. G.; Buell, D. A.; et al. 2019. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779): 505–510.

Audibert, J.-Y.; Bubeck, S.; and Munos, R. 2010. Best arm identification in multi-armed bandits. In *COLT*, 41–53. Citeseer.

Auer, P.; and Ortner, R. 2010. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2): 55–65.

Azuma, K.; Economou, S. E.; Elkouss, D.; Hilaire, P.; Jiang, L.; Lo, H.-K.; and Tzitrin, I. 2022. Quantum repeaters: From quantum networks to the quantum internet. *arXiv preprint arXiv:2212.10820*.

Barrachina-Muñoz, S.; and Bellalta, B. 2017. Learning optimal routing for the uplink in LPWANs using similarity-enhanced e-greedy. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 1–5. IEEE.

Brahmachari, S.; Lumbreras, J.; and Tomamichel, M. 2023. Quantum contextual bandits and recommender systems for quantum data. *arXiv preprint arXiv:2301.13524*.

Brassard, G.; Hoyer, P.; Mosca, M.; and Tapp, A. 2002. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305: 53–74.

Bubeck, S.; Munos, R.; and Stoltz, G. 2011. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19): 1832–1852.

Buchholz, S.; Kübler, J. M.; and Schölkopf, B. 2023. Multi armed bandits and quantum channel oracles. *arXiv preprint arXiv:2301.08544*.

Cacciapuoti, A. S.; Caleffi, M.; Tafuri, F.; Cataliotti, F. S.; Gherardini, S.; and Bianchi, G. 2019. Quantum internet: Networking challenges in distributed quantum computing. *IEEE Network*, 34(1): 137–143.

Casalé, B.; Di Molfetta, G.; Kadri, H.; and Ralaivola, L. 2020. Quantum bandits. *Quantum Machine Intelligence*, 2(1): 1–7.

Childs, A. M. 2017. Lecture notes on quantum algorithms. *Lecture notes at University of Maryland*.

Childs, A. M.; Kothari, R.; and Somma, R. D. 2017. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46(6): 1920–1950.

Cho, B.; Xiao, Y.; Hui, P.; and Dong, D. 2022. Quantum bandit with amplitude amplification exploration in an adversarial environment. *arXiv preprint arXiv:2208.07144*.

Chow, J.; Dial, O.; and Gambetta, J. 2021. IBM Quantum breaks the 100-qubit processor barrier. *IBM Research Blog*.

Chuang, I. L.; and Yamamoto, Y. 1995. Simple quantum computer. *Physical Review A*, 52(5): 3489.

Dai, Z.; Lau, G. K. R.; Verma, A.; Shu, Y.; Low, B. K. H.; and Jaillet, P. 2023. Quantum Bayesian Optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Einstein, A.; Podolsky, B.; and Rosen, N. 1935. Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10): 777.

Even-Dar, E.; Mannor, S.; and Mansour, Y. 2002. PAC bounds for multi-armed bandit and Markov decision processes. In *COLT*, volume 2, 255–270. Springer.

Even-Dar, E.; Mannor, S.; Mansour, Y.; and Mahadevan, S. 2006. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *Journal of machine learning research*, 7(6).

Ganguly, B.; Wu, Y.; Wang, D.; and Aggarwal, V. 2023. Quantum Computing Provides Exponential Regret Improvement in Episodic Reinforcement Learning. *arXiv preprint arXiv:2302.08617*.

Giovannetti, V.; Lloyd, S.; and Maccone, L. 2008. Quantum random access memory. *Physical review letters*, 100(16): 160501.

Grinko, D.; Gacon, J.; Zoufal, C.; and Woerner, S. 2021. Iterative quantum amplitude estimation. *npj Quantum Information*, 7(1): 52.

Grover, L. K. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, 212–219.

Grover, L. K.; and Radhakrishnan, J. 2005. Is partial quantum search of a database any easier? In *Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures*, 186–194.

Hamoudi, Y. 2021. Quantum Sub-Gaussian Mean Estimator. In *29th Annual European Symposium on Algorithms (ESA 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Holevo, A. S. 2003. *Statistical structure of quantum theory*, volume 67. Berlin: Springer Science & Business Media.

Kargin, V. 2005. ON THE CHERNOFF BOUND FOR EFFICIENCY OF QUANTUM HYPOTHESIS TESTING. *The Annals of Statistics*, 33(2): 959–976.

Karnin, Z.; Koren, T.; and Somekh, O. 2013. Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning*, 1238–1246. PMLR.

Kaye, P.; Laflamme, R.; and Mosca, M. 2006. *An introduction to quantum computing*. Oxford: OUP Oxford.

Klauck, H.; Špalek, R.; and De Wolf, R. 2007. Quantum and classical strong direct product theorems and optimal time-space tradeoffs. *SIAM Journal on Computing*, 36(5): 1472–1493.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge: Cambridge University Press.

Li, T.; and Zhang, R. 2022. Quantum Speedups of Optimizing Approximately Convex Functions with Applications to Logarithmic Regret Stochastic Convex Bandits. *Advances in Neural Information Processing Systems*, 35: 3152–3164.

Li, X.-H.; Deng, F.-G.; and Zhou, H.-Y. 2008. Efficient quantum key distribution over a collective noise channel. *Physical Review A*, 78(2): 022321.

Lin, L.; and Tong, Y. 2020. Near-optimal ground state preparation. *Quantum*, 4: 372.

Liu, M.; Li, Z.; Wang, X.; and Lui, J. C. 2024. LinkSelFiE: Link Selection and Fidelity Estimation in Quantum Networks. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*, preprint. IEEE.

Lumbreras, J.; Haapasalo, E.; and Tomamichel, M. 2022. Multi-armed quantum bandits: Exploration versus exploitation when learning properties of quantum states. *Quantum*, 6: 749.

Mannor, S.; and Tsitsiklis, J. N. 2004. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun): 623–648.

Montanaro, A. 2015. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181): 20150301.

Naruse, M.; Chauvet, N.; Uchida, A.; Drezet, A.; Bachelier, G.; Huant, S.; and Hori, H. 2019. Decision making photonics: solving bandit problems using photons. *IEEE Journal of Selected Topics in Quantum Electronics*, 26(1): 1–10.

Nielsen, M. A.; and Chuang, I. 2002. Quantum computation and quantum information.

Ohno, H. 2023. Quantum greedy algorithms for multi-armed bandits. *Quantum Information Processing*, 22(2): 101.

Qiskit contributors. 2023. Qiskit: An Open-source Framework for Quantum Computing.

Robbins, H. 1952. Some aspects of the sequential design of experiments.

Wan, Z.; Zhang, Z.; Li, T.; Zhang, J.; and Sun, X. 2023. Quantum Multi-Armed Bandits and Stochastic Linear Bandits Enjoy Logarithmic Regrets. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Wang, D.; You, X.; Li, T.; and Childs, A. M. 2021. Quantum exploration algorithms for multi-armed bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 10102–10110.

Wehner, S.; Elkouss, D.; and Hanson, R. 2018. Quantum internet: A vision for the road ahead. *Science*, 362(6412): eaam9288.

Wu, Y.; Guan, C.; Aggarwal, V.; and Wang, D. 2023. Quantum Heavy-tailed Bandits. *arXiv preprint arXiv:2301.09680*.

Zhong, H.; Hu, J.; Xue, Y.; Li, T.; and Wang, L. 2023. Provably Efficient Exploration in Quantum Reinforcement Learning with Logarithmic Worst-Case Regret. *arXiv preprint arXiv:2302.10796*.

Zhou, Y.; Chen, X.; and Li, J. 2014. Optimal PAC multiple arm identification with applications to crowdsourcing. In *International Conference on Machine Learning*, 217–225. PMLR.

Appendix

The appendix is organized as follows:

- Appendix A provides an extended review of related works.
- Appendix B explains the dynamically loadable quantum random access memory (DL-QRAM) problem and how it maps to the m-constrained quantum oracle.
- Appendix C describes the construction detail of the variable-time quantum oracle VTA as well as the subroutine details of GoodRatio and PartShrink.
- Upper bound proofs:
  - Appendix D provides the proof of the query complexity upper bound for the Q-Elim algorithm for the weak quantum oracle.
  - Appendix E provides the proof of the query complexity upper bound for the Q-Part algorithm for the constrained quantum oracle.
- Lower bound proofs:
  - Appendix F proves the query complexity lower bound for the weak quantum oracle.
  - Appendix G proves the query complexity lower bound for the constrained quantum oracle.
- Appendix H provides the details of the Qiskit-based simulations and curve-fitting of the empirical performance evaluation.

A Related Works

Wan et al. (2023) studied regret minimization with a weak quantum oracle for both multi-armed bandits and linear bandits, and devised regret minimization algorithms for both cases with $O(\log T)$ and $O(\log^{5/2} T)$ upper bounds, respectively. Dai et al. (2023) extended the results of Wan et al. (2023) for linear bandits to (nonlinear) kernelized bandits with weak oracles, Wu et al. (2023) extended the results of Wan et al. (2023) to bandits with heavy-tailed rewards with weak oracles, and Ganguly et al. (2023); Zhong et al. (2023) studied regret minimization in reinforcement learning with weak quantum oracles. All of these works on weak quantum oracles studied the regret minimization objective. We believe our paper is the first to study the BAI problem with a weak quantum oracle. Furthermore, we are also the first to propose and study the m-constrained quantum oracle. Hamoudi (2021) provides a quantum mean estimator for sub-Gaussian random variables, which could potentially generalize the quantum Monte Carlo estimator (Lemma 1) used in our paper for more general reward distributions.

Our proposed m-constrained quantum oracle in (4) is novel. One related work is Grover and Radhakrishnan (2005), where they studied the partial quantum search problem: first partition all items into multiple blocks, each containing the same number of items, and then search which among these blocks contains the marked item. The "partial search" of Grover and Radhakrishnan (2005) aims to find a block (or, subset) containing the marked item, and their oracle is the same as the original one in Grover (1996). On the other hand, the BAI problem with the m-constrained oracle studied in this paper aims to find the best arm (the marked item) with a novel constrained oracle.

Other interdisciplinary works involving multi-armed bandits and quantum computation include (Lumbreras, Haapasalo, and Tomamichel 2022; Brahmachari, Lumbreras, and Tomamichel 2023; Ohno 2023; Buchholz, Kübler, and Schölkopf 2023; Naruse et al. 2019; Cho et al. 2022). For example, Lumbreras, Haapasalo, and Tomamichel (2022); Brahmachari, Lumbreras, and Tomamichel (2023) applied classical bandit algorithms to learn properties of quantum states and recommend quantum states. Ohno (2023) applied quantum maximization and amplitude encoding to speed up the classical ϵ-greedy algorithm in MAB. Cho et al. (2022) proposed quantum amplitude amplification exploration algorithm for adversarial MAB. Naruse et al. (2019) built a physical quantum system (based on photons) to implement classical MAB algorithms. Li and Zhang (2022) studies quantum speedup for optimizing approximately convex functions with bandit feedback. While both their work and ours aim to identify the best arm, their continuous convex setting is fundamentally different from our discrete action space, making their techniques not directly applicable to our problem.

B Motivating Example of the *m*-Constrained Quantum Oracle

While both the weak and strong quantum oracles were considered previously (Wang et al. 2021; Wan et al. 2023), the m-constrained quantum oracle is introduced in this work. Here we explain how, in addition to generalizing the weak and strong quantum bandit oracles, the m-constrained quantum oracle naturally captures an extension of quantum random access memory (QRAM) (Giovannetti, Lloyd, and Maccone 2008) that we call dynamically loadable quantum random access memory (DL-QRAM). A standard QRAM provides access to the contents of a classical array $(x_1, \ldots, x_N)$ in quantum superpositions. However, for large datasets, it is conceivable that only a subset of the classical data can be read by any fixed-sized QRAM. Thus, prior to querying the QRAM, a classical loading operation may be needed to select the subset of data to be read in superposition. We formalize this by defining a (K, m)-dynamically loadable QRAM as consisting of the following:

1. A classical memory Y which can store K r-bit numbers $y = (y_1, \ldots, y_K)$.
2. A classical memory X which can store m r-bit numbers $x = (x_1, \ldots, x_m)$, where $m \leqslant K$.
3. A QRAM which can access the contents of X in quantum superposition, i.e., a unitary U such that
$$U \left|j\right\rangle_I \left|b\right\rangle_A = \left|j\right\rangle_I \left|b \oplus x_j\right\rangle_A,$$
where I is a $\log m$-qubit index register, and A is an r-qubit ancillary register.
4. A set $\mathcal{L} \coloneqq \{L_0, L_1, \ldots, L_{\binom{K}{m}}\}$ of classical memory loading operations $L_i : \{0,1\}^{rm} \times \{0,1\}^{rK} \to \{0,1\}^{rm} \times \{0,1\}^{rK}$ such that
   (a) $L_0(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}, \boldsymbol{y})$ is the identity operation, and

(b) The indices $i \in \left\{1, 2, \ldots, \binom{K}{m}\right\}$ of L_i correspond to some indexing of the subsets of $\mathcal{S}_i \subseteq \{1, 2, \ldots, K\}$ of cardinality m, and the operation $L_i(\boldsymbol{x}, \boldsymbol{y})$ populates $\boldsymbol{x}$ with the set of elements $\boldsymbol{x}^{(\mathcal{S}_i)} := \{y_j : j \in \mathcal{S}_i\}$, where y_j is the j-th element among the K elements of $\boldsymbol{y}$, while preserving relative ordering of elements. That is, for $\mathcal{S}_i = \{j_1, j_2, \ldots, j_m\}$ where $j_1 < j_2 < \cdots < j_m$, the operation L_i works as follows,

$$L_i(\boldsymbol{x}, \boldsymbol{y}) = ((y_{j_1}, y_{j_2}, \ldots, y_{j_m}), \boldsymbol{y})$$

Then, a query to the DL-QRAM corresponds to the operation $U \circ L$ for the QRAM unitary U and some $L \in \mathcal{L}$, and the loading operation L can vary across queries.

The m-constrained quantum oracle defined in (4) corresponds to a (K, m)-dynamically loadable QRAM where the contents of $\boldsymbol{y}$ are the values $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$, followed by a conditional rotation (cRot) on the reward register R, controlled by an ancillary register A. Denoting classical memory registers with square brackets, we have

$$\begin{aligned}
&\sum_{k \in \mathcal{S}_i} a_k \left|k\right\rangle_I \left|0\right\rangle_R \left|0\right\rangle_A [\mathbf{0}]_X [\boldsymbol{\mu}]_Y \\
\xrightarrow{L_i}\ &\sum_{k \in \mathcal{S}_i} a_k \left|k\right\rangle_I \left|0\right\rangle_R \left|0\right\rangle_A [\mu_k, \forall k \in \mathcal{S}_i]_X [\boldsymbol{\mu}]_Y \\
\xrightarrow{U}\ &\sum_{k \in \mathcal{S}_i} a_k \left|k\right\rangle_I \left|0\right\rangle_R \left|\mu_k\right\rangle_A \text{ (omit classical registers hereon)} \\
\xrightarrow{\text{cRot}}\ &\sum_{k \in \mathcal{S}_i} a_k \left|k\right\rangle_I \left(\sqrt{1-\mu_k} \left|0\right\rangle_R + \sqrt{\mu_k} \left|1\right\rangle_R\right) \left|\mu_k\right\rangle_A \\
\xrightarrow{U^\dagger}\ &\sum_{k \in \mathcal{S}_i} a_k \left|k\right\rangle_I \left(\sqrt{1-\mu_k} \left|0\right\rangle_R + \sqrt{\mu_k} \left|1\right\rangle_R\right) \left|0\right\rangle_A .
\end{aligned}$$

Omitting the ancillary register A and classical registers X, Y, the above mapping is exactly a realization of the m-constrained quantum oracle in (4).

Remark 1. As defined above, a DL-QRAM involves $\binom{K}{m}$ classical memory loading operations L_i. However, in our algorithm design in Section 4, we show that it is not necessary to implement all $\binom{K}{m}$ possible corresponding queries, and only $\lceil K/m \rceil$ such L_i operations suffice for our needs.

C Extended Algorithm Details

C.1 Variable-time algorithm construction

Algorithm 3 presents the pseudocode of variable-time algorithm. With this algorithm, one can transfer the constrained oracle to a VTA oracle in (6). In Lemma 4, we present the property of the gapped amplitude estimation used in Line 4.

Lemma 4 (Gapped amplitude estimation (GAE) (Childs, Kothari, and Somma 2017, Lemma 22) (Wang et al. 2021, Corollary 2)). *Given a weak quantum oracle $O_{weak}^{(k)}$ with $O_{weak}^{(k)} \left|0\right\rangle = \sqrt{1-\mu_k} \left|0\right\rangle + \sqrt{\mu_k} \left|1\right\rangle$, there is a unitary procedure* GAE$(\epsilon, \delta; \ell)$*, making $O(\frac{1}{\epsilon} \log \frac{1}{\delta})$ queries to $O_{weak}^{(k)}$ and $(O_{weak}^{(k)})^\dagger$, that on input $(\sqrt{1-\mu_k} \left|0\right\rangle_R + \sqrt{\mu_k} \left|1\right\rangle_R) \left|0\right\rangle_C \left|0\right\rangle_P$, prepares a state of the form*

$$(\sqrt{1-\mu_k} \left|0\right\rangle_R + \sqrt{\mu_k} \left|1\right\rangle_R) (\beta_0 \left|0\right\rangle_C \left|\gamma_0\right\rangle_P + \beta_1 \left|1\right\rangle_C \left|\gamma_1\right\rangle_P),$$

where $\beta_0, \beta_1 \in [0, 1]$ satisfy $\beta_0^2 + \beta_1^2 = 1$ with $\beta_1 \leqslant \delta$ if $p \geqslant \ell - \epsilon$ and $\beta_0 \leqslant \delta$ if $p < \ell - 2\epsilon$.

Algorithm 3 $\texttt{VTA}(O_{\text{cons}}^{(S)}, \mathcal{S}, I, \alpha)$ (Adapted from Wang et al. (2021, Algorithm 1))

```
Input: Unified oracle $O_{\text{cons}}^{(S)}$ as in (4) with arm subset $\mathcal{S}$; interval $I = [a, b]$ where $0 < a < b < 1$; approximation parameter $0 < \alpha < 1$
Initialize: $\Delta \leftarrow b - a$; $h \leftarrow \lceil \log \frac{1}{\Delta} \rceil + 2$; $\gamma \leftarrow \frac{\alpha}{2hm^{3/2}}$
1: Initialize state to $\frac{1}{\sqrt{m}} \sum_{k \in \mathcal{S}} |k\rangle_I (\sqrt{1-\mu_k} |0\rangle + \sqrt{\mu_k} |1\rangle)_R |0\rangle_C |0\rangle_P |1\rangle_F$
2: for $j = 1, \ldots, h$ do
3:     if registers $C_1, \ldots, C_{j-1}$ are all in state $|0\rangle$ then
4:         Apply GAE$(2^{-j}, \gamma; b)$ on registers $R, C_j$ and $P_j$
5:     Apply controlled-not gate with control on register $C_j$ and target on register $F$
6: if registers $C_1, \ldots, C_h$ are in state $|0\rangle$ then
7:     Flip the bit stored in register $C_{h+1}$
```

C.2 Pseudocode of $\texttt{GoodRatio}$

We present the pseudocode of the $\texttt{GoodRatio}$ algorithm in Algorithm 4. $\texttt{GoodRatio}$ estimates the good arm ratio for the interval I for each subset $\mathcal{S}$ in the partition $\mathfrak{B}$. The ratio is denoted as $r^{(\mathcal{S})}$.

Algorithm 4 $\texttt{GoodRatio}(O_{\text{cons}}^{(S)}, \mathcal{S}, I, \delta)$

```
Input: Oracle $O_{\text{cons}}^{(S)}$ for the subset of arms $\mathcal{S}$ as in (4); interval $I$; parameter $\delta \in (0, 1)$;
    Construct variable-time quantum algorithms $\mathcal{A} \leftarrow \texttt{VTA}(O_{\text{cons}}^{(S)}, \mathcal{S}, I, 0.01\delta)$
Output: $r^{(\mathcal{S})} \leftarrow \texttt{Estimate}(\mathcal{A}, \epsilon = 0.1, \delta)$
```

C.3 Pseudocode of $\texttt{PartShrink}$

We present the pseudocode of the $\texttt{PartShrink}$ algorithm in Algorithm 5. $\texttt{GoodRatio}$ is used in Line 3 and Line 4to estimate the good arm ratios for the intervals $(a + \epsilon, a + 3\epsilon)$ and $(a + 2\epsilon, a + 4\epsilon)$, respectively, for each subset $\mathcal{S}$ in the partition $\mathfrak{B}$. These ratios are denoted as $r_i^{(\mathcal{S})}$ for $i = 1, 2$. Then, Line 5 computes the union of the events $r_h^{(\mathcal{S})} > \frac{h+0.5}{m+1}$ for all $\mathcal{S} \in \mathfrak{B}$. The indicator of the union event B_1 and B_2 signifies whether there are arms within or to the right of the intervals $(a + \epsilon, a + 3\epsilon)$ and $(a + 2\epsilon, a + 4\epsilon)$, respectively. Based on (B_1, B_2), $\texttt{PartShrink}$ then adjusts the interval length.

D Upper Bound Proofs for $\texttt{BAI}$ with Weak Oracle (Theorem 1)

Theorem 1 (Query complexity upper bound of Algorithm 1). *Given confidence parameter $\delta \in (0, 1)$, the query*

Algorithm 5 `PartShrink` $\left((O_{\text{cons}}^{(\mathcal{S})})_{\forall \mathcal{S}\in\mathfrak{B}}, \mathfrak{B}, I, h, \delta\right)$

```
Input: Oracle $O_{\text{cons}}^{(S)}$ as in (4) for each $S$ in $\mathfrak{B}$; partition set $\mathfrak{B}$; interval $I=[a,b]$; parameter $h\in\{1,2\}$ and $\delta\in(0,1)$
Initialize: $\epsilon \leftarrow (b-a)/5, \delta \leftarrow \delta/2$
Append arm 0 with mean 1 to $O_{\text{cons}}^{(\mathcal{S})}$ for all subsets of arms $\mathcal{S} \in \mathfrak{B}$; call the resulting oracles $O_{\text{cons}}^{(\mathcal{S}\cup\{0\})}$.
for $\mathcal{S} \in \mathfrak{B}$ do
    $r_1^{(\mathcal{S})}\leftarrow$GoodRatio($O_{\text{cons}}^{(\mathcal{S}\cup\{0\})}, \mathcal{S}\cup\{0\}, I=(a+\epsilon, a+3\epsilon), 0.01\delta$)
    $r_2^{(\mathcal{S})}\leftarrow$GoodRatio($O_{\text{cons}}^{(\mathcal{S}\cup\{0\})}, \mathcal{S}\cup\{0\}, I=(a+2\epsilon, a+4\epsilon), 0.01\delta$)
$B_1 \leftarrow \mathbb{1}\left\{\bigcup_{\mathcal{S}\in\mathfrak{B}}\left\{r_1^{(\mathcal{S})} > \frac{h+0.5}{m+1}\right\}\right\}, B_2 \leftarrow \mathbb{1}\left\{\bigcup_{\mathcal{S}\in\mathfrak{B}}\left\{r_2^{(\mathcal{S})} > \frac{h+0.5}{m+1}\right\}\right\}$
switch $(B_1, B_2)$
    case $(0,0) : J \leftarrow [a, a+3\epsilon]$
    case $(0,1)$ or $(1,0) : J \leftarrow [a+\epsilon, a+4\epsilon]$
    case $(1,1) : J \leftarrow [a+2\epsilon, a+5\epsilon]$
Output: interval $J$
```

complexity of `Q-Elim` *is upper bounded as follows,*

$$Q \leqslant \sum_{k\in\mathcal{K}} \log_2\left(\frac{4}{\Delta_k}\right)\frac{16C_1}{\Delta_k}\log\frac{K}{\delta},$$

where log *is the natural logarithm, and* $\log_2$ *is the logarithmic function base* 2*.*

Proof of Theorem 1. **Correctness:** If all estimates of `QuEst` are correct, i.e., $\mu_k \in (\hat{\mu}_k - 2^{-j}, \hat{\mu}_k + 2^{-p})$ for all arms in C, then the algorithm outputs arm 1. Hence, we only need to show the probability that any of these `QuEst` fail is upper bounded by δ.

In the j^{th} round, the probability that any of the $|C|$ quantum estimates fails is upper bounded by $|C| \times \frac{2^{-j}\delta}{|C|} = 2^{-j}\delta$. Therefore, the total failure probability over all rounds is upper bounded by $\sum_{j=1}^{\infty} 2^{-j}\delta = \delta$.

Query Complexity: Since failures of the `QuEst` procedures are accounted for by the above fixed confidence, we assume that $\mu_k \in (\hat{\mu}_k - 2^{-j}, \hat{\mu}_k + 2^{-j})$ holds for all arms in C, and derive an upper bound on the number of queries needed by `Q-Elim` to output the optimal arm.

Consider a complete execution of Algorithm 1. Fix a suboptimal arm k. Denote by s_k the batch during which arm k is eliminated. We show that this arm must have been eliminated when $4\cdot 2^{-j} < \Delta_k$. Otherwise, this arm is not removed, which implies that

$$\mu_k + 4\cdot 2^{-j} \overset{(a)}{\geqslant} \hat{\mu}_k + 3\cdot 2^{-j} \overset{(b)}{\geqslant} \hat{\mu}_{\max} + 2^{-j} \geqslant \hat{\mu}_1 + 2^{-j} \overset{(c)}{\geqslant} \mu_1,$$

where inequalities is (a) and (c) are due to the confidence interval $\mu_k \in (\hat{\mu}_k - 2^{-p}, \hat{\mu}_k + 2^{-j})$, and inequality (b) stems from the fact that the elimination condition of Line 5 does not hold. That is, if the arm is not eliminated, we have $4\cdot 2^{-j} \geqslant \mu_1 - \mu_k = \Delta_k$, which contradicts $4\cdot 2^{-j} < \Delta_k$. Therefore, assuming the last batch that the arm k is queried is s_k, we have $4\cdot 2^{-s_k} \geqslant \Delta_k$. After rearrangement, we have $2^{s_k} \leqslant 4/\Delta_k$. So, we can bound the number of queries to arm k as follows,

$$\sum_{j=1}^{s_k} C_1 2^{-j}\log\frac{K}{2^{-j}\delta} \leqslant C_1\log\frac{2K}{\delta}\sum_{j=1}^{s_k} 2^j(j+1)$$

$$\leqslant C_1\log\frac{2K}{\delta}\cdot(s_k+1)2^{s_k+1} \leqslant C_1\log\frac{2K}{\delta}\log_2\left(\frac{4}{\Delta_k}\right)\frac{16}{\Delta_k}.$$

Summing the number of queries over all arms concludes the proof. □

E Upper Bound Proofs for `BAI` with Constrained Oracle (Theorem 3)

Theorem 3 (Query complexity upper bound for `Q-Part` of the m-constrained quantum oracle)**.** *With confidence parameter* $\delta \in (0,1)$ *and an arm partition* $\mathfrak{B}$*, the query complexity of Algorithm 2 is* $O\left(\sum_{\mathcal{S}\in\mathfrak{B}}\sqrt{\sum_{k\in\mathcal{S}}\frac{1}{\Delta_k^2}}\,\text{polylog}\left(\frac{K}{\delta\Delta_2}\right)\right)$, *where* $\Delta_k = \mu_1 - \mu_k$ *is the reward gap of arm* k*, and* Δ_2 *is the minimal reward gap.*

The proof of Theorem 3 relies on two key properties of `PartShrink`, captured in the lemma below.

E.1 Proof of Key Lemmas

In Lemma 5, we show the query complexity of applying the variable-time algorithm to the `Amplify` and `Estimate` subroutines.

Lemma 5 (Adapted from Wang et al. (2021, Lemma 3))**.** *Let* $\mathcal{A} = \texttt{VTA}(O_{\text{cons}}^{(S)}, \mathcal{S}, I = [a,b], \alpha)$*. Then,* `Amplify`$(\mathcal{A}, \delta)$ *uses* $O(G)$ *queries to output an index* $k \in \mathcal{S}_{\text{right}}\cup\mathcal{S}_{\text{middle}}$ *with probability* $1-\delta$*, and* `Estimate`$(\mathcal{A}, \epsilon, \delta)$ *uses* $O(G/\epsilon)$ *queries to output an estimate* $\hat{p}_{\text{good}}$ *of* p_{good} *such that*

$$(1-\epsilon)\left(p_{\text{good}} - \frac{0.1}{m}\right) < \hat{p}_{\text{good}} < (1+\epsilon)\left(p_{\text{good}} + \frac{0.1}{m}\right)$$

with probability $\geqslant 1-\delta$*, where the parameter* $G :=$ $\sqrt{\frac{1}{(b-a)^2} + \frac{1}{|\mathcal{S}_{\text{right}}|}\sum_{k\in\mathcal{S}_{\text{left}}\cup\mathcal{S}_{\text{middle}}}\frac{1}{(b-\mu_k)^2}}\,\text{polylog}\left(\frac{m}{\delta(b-a)}\right)$.

Next, we prove the key lemmas for the `GoodRatio` and `PartShrink` subroutines.

Lemma 2 (Performance of `GoodRatio`)**.** *Given an interval* $I = [a,b]$ *and a confidence parameter* $0 < \delta < 1$*, there exists a* `GoodRatio`$(O_{cons}^{(S)}, \mathcal{S}, I = [a,b], \delta)$ *subroutine which uses* $O(G)$ *queries to output an estimate* $\hat{p}_{\text{good}}$ *of the "good arm" ratio* p_{good} *such that*

$$0.9\left(p_{\text{good}} - \frac{0.1}{m}\right) < \hat{p}_{\text{good}} < 1.1\left(p_{\text{good}} + \frac{0.1}{m}\right)$$

with probability at least $1-\delta$*, where the parameter* $G :=$ $\sqrt{\frac{1}{(b-a)^2} + \frac{1}{|\mathcal{S}_{\text{right}}|}\sum_{k\in\mathcal{S}_{\text{left}}\cup\mathcal{S}_{\text{middle}}}\frac{1}{(b-\mu_k)^2}}\,\text{polylog}\left(\frac{m}{\delta(b-a)}\right)$.

Proof of Lemma 2. This proof is a straightforward application of Lemma 5 to the `Estimate` in **Output** of Algorithm 4. □

Lemma 3 (Performance of `PartShrink`). *Given $h \in \{1,2\}$, an interval $I = [a,b]$, and a confidence parameter $0 < \delta < 1$, supposing $\mu_h \in I$ and $|I| \geqslant \Delta_2/8$, there exists a* `PartShrink` $\left((O_{cons}^{(\mathcal{S})})_{\forall \mathcal{S} \in \mathfrak{B}}, \mathfrak{B}, I, h, \delta\right)$ *subroutine which*

1. *outputs an interval J with $|J| = 3|I|/5$ such that $\mu_h \in J$ with a probability of at least $1-\delta$, and*
2. *uses $O\left(\sum_{\mathcal{S}\in\mathfrak{B}} \sqrt{\sum_{k\in\mathcal{S}} \frac{1}{\Delta_k^2}} \operatorname{polylog}\left(\frac{K}{m\delta\Delta_2}\right)\right)$ queries.*

Proof of Lemma 3. The first part of this lemma is similar to that of Lin and Tong (2020), and we refer the reader to its proof.

For the second part of Lemma 3, we fix a subset $\mathcal{S} \in \mathfrak{B}$. We first recall the query complexity of `Estimate` in Lines 3 and 4, which is

$$\sqrt{\frac{1}{(b-a)^2} + \frac{1}{|\mathcal{S}_{\text{right}}|} \sum_{k \in \mathcal{S}_{\text{left}} \cup \mathcal{S}_{\text{middle}}} \frac{1}{(b-\mu_k)^2}}$$

up to some polylog factors.

Next, we show the above cost is upper bounded by $\tilde{O}(\sqrt{\sum_{k\in\mathcal{S}} \frac{1}{\Delta_k^2}})$. With assumption $|I| \geqslant \Delta_2/8$, we have $b - a = 3\epsilon = 3|I|/5 \geqslant 3\Delta_2/40$, which implies $1/(b-a)^2 = O(1/\Delta_2^2)$. Also, note that $|\mathcal{S}_{\text{right}}| > 1$ because we appended a dummy arm with mean $\mu_0 = 1$ to arm set $\mathcal{S}$ in Line 1.

Next, we bound $1/(b-\mu_k)^2$ by $1/(\mu_1-\mu_k)^2$ for any arm $k \in \mathcal{S}_{\text{left}} \cup \mathcal{S}_{\text{middle}}$. By the definition of $\mathcal{S}_{\text{left}} \cup \mathcal{S}_{\text{middle}}$, we have $\mu_k < b - \Delta/8$, that is, $b - \mu_k > \Delta/8$. By the assumption that $\mu_h \in I$ and noticing $b \in I$, we have $|\mu_h - b| \leqslant |I| = 5\Delta/2$ for $h \in \{1,2\}$. If $h = 1$, that implies $|\mu_1 - b| \leqslant |I| = 5\Delta/2$. If $h = 2$, that implies $|\mu_2 - b| \leqslant |I| = 5\Delta/2$, which further suggests $|\mu_1 - b| \leqslant \Delta_2 + |\mu_2 - b| \leqslant 20\Delta + 5\Delta/2 < 25\Delta$. That is, regardless of whether $h = 1$ or $h = 2$, we always have $|\mu_1 - b| < 25\Delta$. Therefore,

$$\frac{\mu_1 - \mu_k}{b - \mu_k} = 1 + \frac{\mu_1 - b}{b - \mu_k} < 1 + \frac{25\Delta}{\Delta/8} = 201,$$

and so $1/(b-\mu_k)^2 = O(1/(\mu_1-\mu_k)^2)$, which proves the second statement. □

E.2 Proof of Theorem 3

Subset-Level Elimination in `Q-Part` Algorithm: Our `Q-Part` algorithm introduces a novel subset-level elimination mechanism for identifying the arm subset containing the best arm (as seen in Lines 6, 7 and the `GoodRatio` subroutine). Unlike the strong oracle algorithm in Wang et al. (2021), this design leverages the maintained reward mean interval $[I_2, I_1]$ for the optimal and second-best arms, using the upper bound I_1 to eliminate subsets that do not contain the optimal arm. By bridging information across different subsets, this approach reduces unnecessary queries for sub-optimal arms within subsets that lack the optimal arm.

Proof of Theorem 3. Let $E = \{\mu_1 \in I_1, \mu_2 \in I_2$ during the whole while loop$\}$ denote the event that interval I_1 always covers the mean reward of optimal arm, and interval I_2 for the second best arm during the execution of `Q-Part`.

Correctness Correctly outputting the optimal arm relies on (1) event E holding, i.e., `PartShrink` works properly, and (2) proper execution of `Amplify` in Line 12. Based on Lemmas 3 and 5, and the iterative halving of confidence parameter in Line 6, the failure probability of both conditions is at at most $\sum_{n\geqslant 1} \frac{\delta}{2^n} < \delta$.

Query Complexity In this part of the proof, we assume event E holds. With Lemma 3, we know that the intervals I_1 and I_2 shrink by a factor of $3/5$ in each iteration of the while loop. Therefore, at the end of the $(\lceil \log_{5/3} \Delta_2^{-1} \rceil + 3)$-th iteration, $|I_k| \leqslant \Delta_2/4$ for both $k = 1, 2$. Then,

$$\min I_1 - \max I_2 > \Delta_2 - 2 \times \frac{\Delta_2}{4} = \frac{\Delta_2}{2} > 2|I_1|,$$

which fulfills the first while-stop condition in Line 2. At the same time, $\min I_1 - \max I_2 > 2|I_1| > 0$ implies that the interval I_1 only contains one arm (the best one) among all arms in the partition set $\mathfrak{B}$. Therefore, except for the subset $\mathcal{S} \ni 1$, all other subsets $\mathcal{S}'(\neq \mathcal{S}) \in \mathfrak{B}$ do not have arms whose mean rewards lie in or on the right hand side of interval I_1. This guarantees that `GoodRatio`$(O_{\text{cons}}^{(\mathcal{S}')}, I_1, \delta) = 0$ for $\mathcal{S}' \neq \mathcal{S}$. So, the second while-stop condition in Line 2 would also be fulfilled on or before the $(\lceil \log_{5/3} \Delta_2^{-1} \rceil + 3)$-th iteration.

By Lemma 3, the number of queries in Lines 2-8 is upper bounded by $\tilde{O}\left(\sum_{\mathcal{S}\in\mathfrak{B}} \sqrt{\sum_{k\in\mathcal{S}} \Delta_k^{-2}}\right)$. The remaining queries of `Q-Part` are from the `Amplify` subroutine for the set of arms $\mathcal{S}$ in Line 12, which, by on Lemma 5, is $\tilde{o}\left(\sum_{\mathcal{S}\in\mathfrak{B}} \sqrt{\sum_{k\in\mathcal{S}} \Delta_k^{-2}}\right)$ and can be ignored. □

F Lower Bound Proofs for `BAI` with Weak Oracle

F.1 Preliminaries

In this section, we first start by reviewing a quantum hypothesis testing lower bound (Lemma 6), and then apply this lower bound to distinguish two quantum `MAB` models (Lemma 7) in Section F.2. Then, in Section 3.3, based on the previous two lower bounds, we derive a query complexity lower bound for `BAI` with a weak quantum oracle (Theorem 2). Last, we derive two lower bounds for `BAI` with the m-constrained quantum oracle (Theorems 6 and 4) via tailored quantum adversarial methods in Section 4.4.

F.2 Preliminary Lower Bounds

Quantum hypothesis testing (Holevo 2003, §2.2) aims to solve the following problem: Given multiple copies of one of two known quantum states, $|\psi_0\rangle, |\psi_1\rangle$, determine which of both states has been given.

Lemma 6 (Error probability lower bound for quantum pure state hypothesis testing (Kargin 2005)). *Given Q copies of one of two pure quantum states, $|\psi_0\rangle$ or $|\psi_1\rangle$ (equal prior), the error probability of deciding which state has been given is*

$$p_{error}^{(Q)} \geqslant \frac{1}{2}\left(1 - \sqrt{1 - |\langle\psi_0|\psi_1\rangle|^{2Q}}\right).$$

Next, we extend the hypothesis testing of two quantum pure states to distinguishing two quantum MAB instances $\mathcal{I}_0 = (\mu_1^{(0)}, \mu_2^{(0)}, \dots, \mu_K^{(0)})$ and $\mathcal{I}_1 = (\mu_1^{(1)}, \mu_2^{(1)}, \dots, \mu_K^{(1)})$. We consider the case where both instances have only one arm ℓ that differ in their oracles:

$$O_\ell^{(0)} : |0\rangle \to \sqrt{1-\mu_0}\,|0\rangle + \sqrt{\mu_0}\,|1\rangle\,,$$
$$O_\ell^{(1)} : |0\rangle \to \sqrt{1-\mu_1}\,|0\rangle + \sqrt{\mu_1}\,|1\rangle\,,$$

that is, $\mu_\ell^{(0)} = \mu_0 \neq \mu_1 = \mu_\ell^{(1)}$, and all other arm mean rewards are the same, i.e., $\mu_k^{(0)} = \mu_k^{(1)}$ for all arms $k \neq \ell$.

Lemma 7 (Query complexity lower bound for distinguishing two quantum MAB instances differing by exactly one arm's mean reward)**.** *Given $\mu_0, \mu_1 \in \left(0, \frac{1}{4}\right)$*[3]*, the number of queries Q required to distinguish the quantum MAB instances $\mathcal{I}_0$ and $\mathcal{I}_1$, with a probability of at least $1-\delta$, is lower bounded by $Q \geqslant \frac{1}{4|\mu_0-\mu_1|}\log\frac{1}{4\delta}$.*

F.3 Proof of Lemma 7

Proof of Lemma 7. **Step 1. Relax the task to quantum hypothesis testing.** We begin with an easier task than distinguishing two quantum MAB instances. We assume that the mean reward parameters of both instances are known a priori, that is, the arm index ℓ whose mean reward differs in instance $\mathcal{I}_0$ and $\mathcal{I}_1$ and the values of μ_0 and μ_1 are known. To address this relaxed task, one only needs to pull arm ℓ and test whether the mean reward of this arm is μ_0 or μ_1. We note that with the above additional information, the task becomes easier, and, hence, the query complexity lower bound of this relaxed task also serves as a lower bound for the original task.

Step 2. Calculate the query complexity lower bound from the quantum hypothesis testing result. Let $\sqrt{\mu_0} = \sin\theta_0$ and $\sqrt{\mu_1} = \sin\theta_1$ for $\theta_0, \theta_1 \in \left(0, \frac{\pi}{4}\right)$. We can rewrite the quantum states as

$$|\psi_0\rangle \coloneqq \sqrt{1-\mu_0}\,|0\rangle + \sqrt{\mu_0}\,|1\rangle = \cos\theta_0\,|0\rangle + \sin\theta_0\,|1\rangle\,,$$
$$|\psi_1\rangle \coloneqq \sqrt{1-\mu_1}\,|0\rangle + \sqrt{\mu_1}\,|1\rangle = \cos\theta_1\,|0\rangle + \sin\theta_1\,|1\rangle\,.$$

By Lemma 6, to differentiate both oracles with probability at least $1-\delta$, one needs

$$\delta \geqslant p_{\text{error}}^{(Q)} \geqslant \frac{1}{2}\left(1 - \sqrt{1 - |\langle\psi_0|\psi_1\rangle|^{2Q}}\right).$$

After rearranging the above inequality, we have

$$Q \geqslant \frac{\log\left(1-(1-2\delta)^2\right)}{2\log|\langle\psi_0|\psi_1\rangle|} = \frac{1}{-2\log|\langle\psi_0|\psi_1\rangle|}\log\frac{1}{4\delta(1-\delta)} \tag{7}$$

Algebraic calculations (see Appendix F.3) yield $\log|\langle\psi_0|\psi_1\rangle|^{-1} \leqslant (\theta_0-\theta_1)^2/2$ and $|\mu_0-\mu_1| \geqslant (\theta_0-\theta_1)^2/4$. Substituting both inequalities into (7) concludes the proof. □

Algebraic Details of Proof of Lemma 7 We first prove that $\log|\langle\psi_0|\psi_1\rangle|^{-1} \leqslant \frac{(\theta_0-\theta_1)^2}{2}$ as follows.

$$\begin{aligned}\log|\langle\psi_0|\psi_1\rangle| &= \log(\cos(\theta_0-\theta_1)) \\ &\overset{(a)}{\geqslant} \log\left(1 - \frac{(\theta_0-\theta_1)^2}{2}\right) \overset{(b)}{\geqslant} -\frac{(\theta_0-\theta_1)^2}{2}\end{aligned} \tag{8}$$

where inequality (a) is due to $\cos x \geqslant 1 - \frac{x^2}{2}$, and inequality (b) is due to to $\log(1-x) \geqslant -x$ for $x \in (0, 0.85)$.

Next, we upper bound $|\mu_0-\mu_1|$ with an expression of θ_0 and θ_1. With trigonometric identities, we have

$$\begin{aligned}&\mu_0 - \mu_1 \\ &= \sin^2\theta_0 - \sin^2\theta_1 \\ &= \sin^2((\theta_0-\theta_1)+\theta_1) - \sin^2\theta_1 \\ &= \sin^2(\theta_0-\theta_1)\cos^2\theta_1 + \cos^2(\theta_0-\theta_1)\sin^2\theta_1 \\ &\quad + 2\sin(\theta_0-\theta_1)\cos(\theta_0-\theta_1)\sin\theta_1\cos\theta_1 - \sin^2\theta_1 \\ &= \sin\theta_1\cos\theta_1\sin 2(\theta_0-\theta_1) + (1-2\sin^2\theta_1)\sin^2(\theta_0-\theta_1).\end{aligned}$$

Taking the absolute values of both sides, we obtain

$$\begin{aligned}&|\mu_0-\mu_1| \\ &\geqslant \left|(1-2\sin^2\theta_1)\sin^2(\theta_0-\theta_1)\right| - |\sin\theta_1\cos\theta_1\sin 2(\theta_0-\theta_1)| \\ &\geqslant \left|(1-2\sin^2\theta_1)\sin^2(\theta_0-\theta_1)\right| \\ &\overset{(a)}{\geqslant} |(1-2\mu_1)|\frac{(\theta_0-\theta_1)^2}{4} \\ &\overset{(b)}{\geqslant} \frac{(\theta_0-\theta_1)^2}{8}\end{aligned} \tag{9}$$

where inequality (a) is due to $\sin x \geqslant \frac{x}{2}$ for $x \in (0, 1.8)$, and inequality (b) is due to $\mu_1 \leqslant \frac{1}{4}$.

Lastly, we conclude the proof as follows,

$$\begin{aligned}Q &\overset{(a)}{\geqslant} \frac{1}{-2\log|\langle\psi_0|\psi_1\rangle|}\log\frac{1}{4\delta(1-\delta)} \\ &\overset{(b)}{\geqslant} \frac{1}{(\theta_0-\theta_1)^2}\log\frac{1}{4\delta(1-\delta)} \\ &\overset{(c)}{\geqslant} \frac{1}{8|\mu_0-\mu_1|}\log\frac{1}{4\delta(1-\delta)} \\ &\geqslant \frac{1}{8|\mu_0-\mu_1|}\log\frac{1}{4\delta},\end{aligned}$$

where inequalities (a), (b), and (c) are due to (7), (8), and (9) respectively.

F.4 Lower Bound Proof for BAI with Weak Oracle (Theorem 2)

Theorem 2 (Query complexity lower bound for best arm identification)**.** *Given a quantum multi-armed bandits instance $\mathcal{I}_0 = \{\mu_1, \dots, \mu_K\}$ where $\mu_k \in (0, 1/2)$ for all k and $\mu_1 > \mu_2 \geqslant \mu_k$ for any $k \neq 1$, any algorithm that identifies the optimal arm with a given confidence $1-\delta$, $\delta \in (0,1)$ requires Q queries to the weak quantum oracle, where*

$$Q \geqslant \sum_{k\in\mathcal{K}} \frac{1}{4\Delta_k}\log\frac{1}{4\delta}.$$

[3] This assumption is needed in the algebraic calculations in Appendix F.3, and such a constant constraint assumption is common for lower bound results because one often needs addition conditions to construct difficult instances, e.g., Mannor and Tsitsiklis (2004, Theorem 1).

Proof of Theorem 2. **For every suboptimal arm $k \neq 1$ in instance $\mathcal{I}_0$,** we consider another instance $\mathcal{I}_k = \{\mu_1^{(k)}, \ldots, \mu_K^{(k)}\}$ whose mean rewards are the same as in instance $\mathcal{I}_0$ except for the mean reward of arm k which assumes the form $\mu_k^{(k)} = \mu_1 + \epsilon$ for $0 < \epsilon < \frac{1}{2} - \mu_1$. Therefore, in instance $\mathcal{I}_k$, the best arm is $k \neq 1$. Because instances $\mathcal{I}_0$ and $\mathcal{I}_k$ have different best arms, any feasible policy must be able to distinguish these two instances with a confidence of at least $1 - \delta$. Given the additional information that all other arms have the same means, this task reduces to distinguishing two instances $\mathcal{I}_0$ and $\mathcal{I}_k$ as in Lemma 2. In particular, from the proof of Lemma 2, we know that, in order to distinguish two `MAB` instances that only differ in arm k's mean reward, spending $1/(4(\Delta_k + \epsilon)) \log 1/(4\delta)$ queries on arm k is necessary.

For the optimal arm $k = 1$ in instance $\mathcal{I}_0$, we consider another instance $\mathcal{I}_1 = \{\mu_1^{(1)}, \ldots, \mu_K^{(1)}\}$ whose oracles are the same as instance $\mathcal{I}_0$ except that the mean reward of arm 1 in $\mathcal{I}_1$ is $\mu_1^{(1)} = \mu_2 - \epsilon$ for $0 < \epsilon < \mu_2$ and recall that arm 2 is the second best arm in $\mathcal{I}_0$. Therefore, in instance $\mathcal{I}_1$, the best arm is 2. Similarly, Lemma 2 states that, to distinguish instances $\mathcal{I}_0$ and $\mathcal{I}_1$, it is necessary to pull arm 1 for $(1/(4(\Delta_2 + \epsilon))) \log 1/(4\delta) = (1/(4(\Delta_1 + \epsilon)) \log 1/(4\delta)$ times.

Last, summing the necessary query complexity spent on each arm and letting ϵ go to zero yields $Q \geqslant \sum_{k \in \mathcal{K}} 1/(4\Delta_k) \log 1/4\delta)$. □

F.5 Alternative Lower bound Proof via quantum adversary method for `BAI` with weak-oracle

In addition to Lemma 7 based on quantum hypothesis testing, we provide an alternative lower bound based on the quantum adversary method (Ambainis 2000).

Theorem 5. *Given $\mu_0, \mu_1 \in (\mu, 1 - \mu)$, the necessary number of queries to distinguish the quantum `MAB` instances $\mathcal{I}_0$ and $\mathcal{I}_1$, with a probability of at least $1 - \delta$, has the following lower bound,*

$$Q \geqslant \frac{1}{|\mu_0 - \mu_1|} \cdot \frac{1 - 2\sqrt{\delta(1-\delta)}}{1 + 2\sqrt{1/\mu(1-\mu)}}.$$

Replacing Lemma 7 by Theorem 5 in the proof of Theorems 2, one can obtain another two query complexity lower bounds for `BAI` as follows,

$$Q \geqslant \sum_{k \in \mathcal{K}} \frac{1}{\Delta_k} \cdot \frac{1 - 2\sqrt{\delta(1-\delta)}}{1 + 2\sqrt{1/\mu(1-\mu)}}.$$

Proof of Theorem 5. Without loss of generality we assume $\mu_1 > \mu_0$ and denote $\Delta = \mu_1 - \mu_0$. For $a = 0, 1$, denote $\left|\psi_a^{(t)}\right\rangle$ as the output after querying the oracle O_a t times . In the adversary method, we consider a weight function as follows,

$$s_t = \frac{1}{\Delta} \left\langle \psi_0^{(t)} \middle| \psi_1^{(t)} \right\rangle.$$

Note that $s_0 = \frac{1}{\Delta}$ and, to distinguish both oracles' output after T queries, we require that $s_T \leqslant \frac{1}{\Delta}\sqrt{2\delta(1-\delta)}$.

After the t^{th} query, we have

$$\left|\psi_0^{(t)}\right\rangle = \alpha_{0,0}\left|0\right\rangle + \alpha_{0,1}\left|1\right\rangle, \quad \left|\psi_1^{(t)}\right\rangle = \alpha_{1,0}\left|0\right\rangle + \alpha_{1,1}\left|1\right\rangle.$$

Denote the action of the quantum oracles by the following two unitary matrices,

$$\boldsymbol{A}_0 = \begin{bmatrix} \sqrt{1-\mu_0} & \sqrt{\mu_0} \\ \sqrt{\mu_0} & -\sqrt{1-\mu_0} \end{bmatrix}, \quad \boldsymbol{A}_1 = \begin{bmatrix} \sqrt{1-\mu_1} & \sqrt{\mu_1} \\ \sqrt{\mu_1} & -\sqrt{1-\mu_1} \end{bmatrix}.$$

Then, we have

$$\begin{aligned} &\left\langle \psi_0^{(t+1)} \middle| \psi_1^{(t+1)} \right\rangle - \left\langle \psi_0^{(t)} \middle| \psi_1^{(t)} \right\rangle \\ &= \left\langle \psi_0^{(t)} \middle| \boldsymbol{A}_0^\dagger \boldsymbol{A}_1 \middle| \psi_1^{(t)} \right\rangle - \left\langle \psi_0^{(t)} \middle| \psi_1^{(t)} \right\rangle \\ &= \left\langle \psi_0^{(t)} \middle| \boldsymbol{A}_0^\dagger \boldsymbol{A}_1 - \boldsymbol{I} \middle| \psi_1^{(t)} \right\rangle. \end{aligned}$$

Denote

$$\begin{aligned} &\begin{bmatrix} u & v \\ -v & u \end{bmatrix} := \boldsymbol{A}_0^\dagger \boldsymbol{A}_1 - \boldsymbol{I} \\ &= \begin{bmatrix} \sqrt{\mu_0\mu_1} + \sqrt{(1-\mu_0)(1-\mu_1)} - 1 & \sqrt{(1-\mu_0)\mu_1} - \sqrt{\mu_0(1-\mu_1)} \\ \sqrt{\mu_0(1-\mu_1)} - \sqrt{(1-\mu_0)\mu_1} & \sqrt{\mu_0\mu_1} + \sqrt{(1-\mu_0)(1-\mu_1) - 1} \end{bmatrix}. \end{aligned}$$

We have

$$\begin{aligned} |s_{t+1} - s_t| &\leqslant \frac{|u|}{\Delta^2}\left|\alpha_{0,0}\alpha_{1,0} + \alpha_{0,1}\alpha_{1,0}\right| + \frac{|v|}{\Delta^2}\left|\alpha_{0,0}\alpha_{1,1} - \alpha_{0,1}\alpha_{1,0}\right| \\ &\overset{(a)}{\leqslant} \frac{|u| + |v|}{\Delta^2} \\ &\overset{(b)}{\leqslant} \frac{1 + 2\sqrt{1/\mu(1-\mu)}}{\Delta}, \end{aligned}$$

where (a) is due to the Cauchy-Schwartz inequality, and (b) is due to the fact that

$$\begin{aligned} |u| &= \left|1 - \sqrt{\mu_0\mu_1} - \sqrt{(1-\mu_0)(1-\mu_1)}\right| \\ &\leqslant |1 - \mu_0 - (1 - \mu_1)| = \Delta, \\ |v| &= \frac{\mu_1 - \mu_0}{\sqrt{(1-\mu_0)\mu_1} + \sqrt{\mu_0(1-\mu_1)}} \leqslant \frac{\Delta}{2\sqrt{\mu(1-\mu)}}. \end{aligned}$$

At last, we have

$$\frac{1}{\Delta^2}\left(1 - 2\sqrt{\delta(1-\delta)}\right) \leqslant |s_T - s_0| \leqslant T \cdot \frac{1 + 2\sqrt{1/\mu(1-\mu)}}{\Delta}.$$

Rearranging the above equation yields

$$T \geqslant \frac{1}{\Delta} \cdot \frac{1 - 2\sqrt{\delta(1-\delta)}}{1 + 2\sqrt{1/\mu(1-\mu)}}.$$

□

G Lower Bound Proofs for `BAI` with *m*-Constrained Oracle (Theorems 6 and 4)

In this section, we first prove a query complexity lower bound for the m-constrained oracle in Theorem 6 which suits any quantum algorithm that uses the m-constrained oracle. Then, we turn the prove an improved lower bound for partition algorithms (Definition 1) in Theorem 4 that is presented in the main paper.

G.1 Proof of Theorem 6

Theorem 6 (Query complexity lower bound for m-constrained oracle (Version 1))**.** *Given $\mu_k \in (p, 1-p)$ for all arms $k \in \mathcal{K}$ with $p \in (0, 1/2)$, identifying the best arm with probability at least $1-\delta$ requires at least*

$$\Omega\left(\sum_{k \in \mathcal{K}} \frac{1}{\Delta_k^2} \Bigg/ \max_{\mathcal{S}:|\mathcal{S}|=m} \sqrt{\sum_{k' \in \mathcal{S}} \frac{1}{\Delta_{k'}^2}} \right)$$

queries to the m-constrained oracle.

Proof of Theorem 6. This proof is based on the quantum adversarial method. We first construct instances and define the weighted summations that are key components in configuring the adversarial method. Next, we bound the difference of any two consecutive weighted summations, which is the central step in applying the adversarial method, and this step needs tailoring to our m-constrained oracle setting as the standard adversarial method assumes a strong oracle (i.e., allows one to query all arms/items concurrently).

Step 1. Construct instances and define weighted summation. We define K instances as follows,

$$\begin{aligned}
&\mathcal{I}_1 = \{\mu_1, \mu_2, \dots, \mu_K\}, \\
&\mathcal{I}_2 = \{\mu_1, \mu_1', \dots, \mu_K\}, \\
&\dots \\
&\mathcal{I}_K = \{\mu_1, \mu_2, \dots, \mu_1'\},
\end{aligned}$$

where $\mu_1 > \mu_2 > \cdots > \mu_K$ and $\mu_1' = \mu_1 + \iota$ for some parameter $\iota > 0$. We also denote $\Delta_i' = \mu_1' - \mu_i = \Delta_i + \iota$ for all $i \in \{1, 2, \dots, K\}$. That is, in each instance $\mathcal{I}_i$, the arm i is optimal.

We define the weight summation for adversarial method as follows,

$$s_t \coloneqq \sum_{i=1}^{K} \frac{1}{(\Delta_i')^2} \left\langle \psi_{i,t} \middle| \psi_{1,t} \right\rangle,$$

where the summation is taken over all K instances, and $\left|\psi_{i,t}\right\rangle$ is the superposition after t times of m-constrained oracle queries and other unitary operations for instance $\mathcal{I}_i$. As the initial states $\left|\psi_{i,0}\right\rangle$ are the same, we have

$$s_0 = \sum_{i=1}^{K} \frac{1}{(\Delta_i')^2}.$$

As any of these two instances have different optimal arms, to correctly identify the best arm, we need to be able to distinguish any two states $\left|\psi_{i,T}\right\rangle$ of different instances after the last T^{th} query with a probability of at least $1-\delta$. Distinguishing two quantum states requires that $\left\langle \psi_{i,T} \middle| \psi_{0,T} \right\rangle \leqslant 2\sqrt{\delta(1-\delta)}$ (Kaye, Laflamme, and Mosca 2006, Appendix A.9). That is, the weight summation s_T at the last query should be bounded as follows,

$$s_T \leqslant \sum_{k>1} \frac{1}{\Delta_k^2} \cdot 2\sqrt{\delta(1-\delta)}.$$

Step 2. Bound the difference of two consecutive weighted summations. In this step, we upper bound the difference of two consecutive weighted summations, i.e.,

$$s_{t+1} - s_t = \sum_{i=1}^{K} \frac{1}{(\Delta_k')^2} \left(\left\langle \psi_{i,t+1} \middle| \psi_{0,t+1} \right\rangle - \left\langle \psi_{i,t} \middle| \psi_{0,t} \right\rangle \right).$$

Denote

$$\boldsymbol{A}_k = \begin{bmatrix} \sqrt{1-\mu_k} & \sqrt{\mu_k} \\ \sqrt{\mu_k} & -\sqrt{1-\mu_k} \end{bmatrix}, \forall k \in \mathcal{K},$$

$$\boldsymbol{A}_1' = \begin{bmatrix} \sqrt{1-\mu_1'} & \sqrt{\mu_1'} \\ \sqrt{\mu_1'} & -\sqrt{1-\mu_1'} \end{bmatrix}$$

as the unitary when querying arms $k \in \mathcal{K}$ and the arm with mean μ_1' respectively. Denote $\left|\psi_{i,t}\right\rangle = \sum_{k,r} \alpha_{i,k,r} \left|k, r\right\rangle$ and $\left|\psi_{1,t}\right\rangle = \sum_{k,r} \alpha_{1,k,r} \left|k, r\right\rangle$ where $k \in \mathcal{K}$ is an index for arm and $r \in \{0, 1\}$ is an index for reward. Then we have

$$\begin{aligned}
&\left|\psi_{i,t+1}\right\rangle = O_{\text{cons},i}^{(\mathcal{S}_{t+1})} \left|\psi_{i,t}\right\rangle \\
&= \sum_{k \in \mathcal{S}_{t+1} \setminus \{i\}, r} \alpha_{i,k,r} \left|k\right\rangle \boldsymbol{A}_k \left|r\right\rangle + \sum_{r} \alpha_{i,i,r} \left|i\right\rangle \boldsymbol{A}_1' \left|r\right\rangle \mathbb{1}\{i \in \mathcal{S}_{t+1}\}, \\
&\left|\psi_{1,t+1}\right\rangle = O_{\text{cons},1}^{(\mathcal{S}_{t+1})} \left|\psi_{1,t}\right\rangle = \sum_{k \in \mathcal{S}_{t+1}, r} \alpha_{1,k,r} \left|k\right\rangle \boldsymbol{A}_k \left|r\right\rangle.
\end{aligned}$$

Then, we have

$$\begin{aligned}
&\left\langle \psi_{i,t+1} \middle| \psi_{1,t+1} \right\rangle - \left\langle \psi_{i,t} \middle| \psi_{1,t} \right\rangle \\
&= \left\langle \psi_{i,t} \right| \left(O_{\text{cons},i}^{(\mathcal{S}_{t+1})} \right)^{\dagger} O_{\text{cons},1}^{(\mathcal{S}_{t+1})} \left|\psi_{1,t}\right\rangle - \left\langle \psi_{i,t} \middle| \psi_{1,t} \right\rangle \\
&= \sum_{r,r'} \alpha_{i,i,r}^{*} \alpha_{1,i,r} \left\langle r \right| \left((\boldsymbol{A}_1')^{\dagger} \boldsymbol{A}_i - \boldsymbol{I} \right) \left|r'\right\rangle \mathbb{1}\{i \in \mathcal{S}_{t+1}\}.
\end{aligned}$$

Hence, we can calculate the consecutive difference as follows,

$$\begin{aligned}
s_{t+1} - s_t &= \sum_{i=1}^{K} \sum_{r,r'} \alpha_{i,i,r}^{*} \alpha_{1,i,r} \left\langle r \right| \left((\boldsymbol{A}_1')^{\dagger} \boldsymbol{A}_i - \boldsymbol{I} \right) \left|r'\right\rangle \mathbb{1}\{i \in \mathcal{S}_{t+1}\} \\
&= \sum_{i \in \mathcal{S}_{t+1}} \sum_{r,r'} \alpha_{i,i,r}^{*} \alpha_{1,i,r} \left\langle r \right| \left((\boldsymbol{A}_1')^{\dagger} \boldsymbol{A}_i - \boldsymbol{I} \right) \left|r'\right\rangle.
\end{aligned}$$

The matrix in the middle is

$$\begin{aligned}
&(\boldsymbol{A}_1')^{\dagger} \boldsymbol{A}_i - \boldsymbol{I} \\
&= \begin{pmatrix} \sqrt{(1-\mu_1')(1-\mu_i)} + \sqrt{\mu_1' \mu_i} - 1 & \sqrt{(1-\mu_1')\mu_i} - \sqrt{\mu_1'(1-\mu_i)} \\ -\sqrt{(1-\mu_1')\mu_i} + \sqrt{\mu_1'(1-\mu_i)} & \sqrt{(1-\mu_1')(1-\mu_i)} + \sqrt{\mu_1' \mu_i} - 1. \end{pmatrix}
\end{aligned}$$

To simplify the remaining calculations, we denote $u_i = \sqrt{(1-\mu_1')(1-\mu_i)} + \sqrt{\mu_1' \mu_i} - 1$ and $v_i = \sqrt{(1-\mu_1')\mu_i} - \sqrt{\mu_1'(1-\mu_i)}$.

Then we can bound the absolute value of $s_{t+1} - s_t$ as follows,

$$\begin{aligned}
&\left|s_{t+1} - s_t\right| \leqslant \\
&\sum_{i \in \mathcal{S}_{t+1}} \left(\sum_{r=r'} \frac{|u_i|}{(\Delta_i')^2} \left|\alpha_{i,i,r}\right| \left|\alpha_{1,i,r}\right| + \sum_{r \neq r'} \frac{|v_i|}{(\Delta_i')^2} \left|\alpha_{i,i,r}\right| \left|\alpha_{1,i,r}\right| \right)
\end{aligned}$$

It is easily verified that $|u_i| \leqslant \Delta'_i$ and $|v_i| \leqslant \frac{\Delta'_i}{2\sqrt{(p-\epsilon)(1-(p-\epsilon))}}$ for $p \in (0, 1/2)$. Next, we bound the two terms in the above inequality as follows,

$$\begin{aligned}
&\sum_{i \in \mathcal{S}_{t+1}} \sum_{r=r'} \frac{|u_i|}{(\Delta'_i)^2} |\alpha_{i,i,r}| |\alpha_{1,i,r}| \\
&\overset{(a)}{\leqslant} \sqrt{\sum_{i \in \mathcal{S}_{t+1}, r=r'} \frac{|u_i|^4}{(\Delta'_i)^4} \left|\alpha^2_{i,i,r}\right|} \sqrt{\sum_{i \in \mathcal{S}_{t+1}, r \neq r'} |\alpha_{1,i,r}|^2} \\
&\leqslant \sqrt{\sum_{i \in \mathcal{S}_{t+1}} \frac{1}{(\Delta'_i)^2}},
\end{aligned}$$

where inequality (a) is due to Cauchy-Schwarz inequality.

$$\begin{aligned}
&\sum_{i \in \mathcal{S}_{t+1}} \sum_{r \neq r'} \frac{|v_i|}{(\Delta'_i)^2} |\alpha_{i,i,r}| |\alpha_{1,i,r}| \\
&= \sum_{r \neq r'} \sum_{i \in \mathcal{S}_{t+1}} \frac{|v_i|}{(\Delta'_i)^2} |\alpha_{i,i,r}| |\alpha_{1,i,r}| \\
&\overset{(a)}{\leqslant} \sum_{r \neq r'} \sqrt{\sum_{i \in \mathcal{S}_{t+1}} \frac{|v_i|^2}{(\Delta'_i)^4} |\alpha_{i,i,r}|^2} \sqrt{\sum_{i \in \mathcal{S}_{t+1}} |\alpha_{1,i,r}|} \\
&\leqslant \frac{1}{\sqrt{(p-\epsilon)(1-(p-\epsilon))}} \sqrt{\sum_{i \in \mathcal{S}_{t+1}} \frac{1}{(\Delta'_i)^2}},
\end{aligned}$$

where, again, we use Cauchy-Schwarz in (a).

Then, we have

$$|s_{t+1} - s_t| \leqslant \left(1 + \frac{1}{\sqrt{(p-\epsilon)(1-(p-\epsilon))}}\right) \sqrt{\sum_{i \in \mathcal{S}_{t+1}} \frac{1}{(\Delta'_i)^2}}.$$

Step 3. Combine Steps 1 and 2 to conclude query complexity lower bound. We combine the results of Steps 1 and 2 to give:

$$\begin{aligned}
&\sum_{i=1}^{K} \frac{1}{(\Delta'_i)^2} (1 - 2\sqrt{\delta(1-\delta)}) \\
&\overset{\text{(Step 1)}}{\leqslant} \quad s_0 - s_T \\
&\leqslant |s_0 - s_1| + \cdots + |s_{T-1} - s_T| \\
&\overset{\text{(Step2)}}{\leqslant} \sum_{t=1}^{T} \left(1 + \frac{1}{\sqrt{(p-\epsilon)(1-(p-\epsilon))}}\right) \sqrt{\sum_{i \in \mathcal{S}_t} \frac{1}{(\Delta'_i)^2}} \\
&\leqslant T \left(1 + \frac{1}{\sqrt{(p-\epsilon)(1-(p-\epsilon))}}\right) \cdot \max_{\mathcal{S}:|\mathcal{S}|=m} \sqrt{\sum_{i \in \mathcal{S}} \frac{1}{(\Delta'_i)^2}}.
\end{aligned}$$

Rearranging this inequality yields

$$T \geqslant \frac{1 - 2\sqrt{\delta(1-\delta)}}{1 + 1/\sqrt{(p-\epsilon)(1-(p-\epsilon))}} \cdot \frac{\sum_{i=1}^{K} \frac{1}{(\Delta'_i)^2}}{\max_{\mathcal{S}:|\mathcal{S}|=m} \sqrt{\sum_{i \in \mathcal{S}} \frac{1}{(\Delta'_i)^2}}}$$

Let $\epsilon = p(\mu_1 - \mu_2)/2$, we have

$$\Delta_i \leqslant \Delta'_i = \mu_1 + \epsilon - \mu_i \leqslant \left(1 + \frac{p}{2}\right)(\mu_1 - \mu_i) \leqslant \frac{5}{4}\Delta_i,$$

and

$$\sqrt{(p-\epsilon)(1-(p-\epsilon))} \geqslant \sqrt{\frac{p}{2}\left(1 - \frac{p}{2}\right)}.$$

Therefore, we have

$$T \geqslant \frac{16}{25} \frac{1 - 2\delta(1-\delta)}{1 + 1/\sqrt{(p/2)(1-(p/2))}} \frac{\sum_{i=1}^{K} \frac{1}{\Delta_i^2}}{\max_{\mathcal{S}:|\mathcal{S}|=m} \sqrt{\sum_{i \in \mathcal{S}} \frac{1}{\Delta_i^2}}},$$

which concludes the proof.

□

G.2 Proof of Theorem 4

Theorem 4 (Query complexity lower bound for m-constrained oracle)**.** *To identify the best arm with a probability of at least $1-\delta$ with the m-constrained oracle with parameter m, any* partition algorithm *needs to spend at least the following number of queries, $\Omega(\sum_{\mathcal{S} \in \mathfrak{B}} \sqrt{\sum_{k \in \mathcal{S}} 1/\Delta_k^2})$, where $\mathfrak{B}$ is the partition of arms.*

Proof of Theorem 4. The proof of Theorem 4 is also based on the quantum adversarial method, and it has three steps as that of Theorem 6. The first step in the instance construction is the same. Below, we present Steps 2 and 3. Denote $\mathfrak{B}$ as the partition of any algorithm and, for any $\mathcal{S} \in \mathfrak{B}$, define the following partial summation

$$s_t^{(\mathcal{S})} = \sum_{i \in \mathcal{S}} \frac{1}{(\Delta'_i)^2} \langle \psi_{i,t} | \psi_{1,t} \rangle .$$

Then, we have $s_t = \sum_{\mathcal{S} \in \mathfrak{B}} s_t^{(\mathcal{S})}$. In the proofs of Steps 2 and 3, we fix one subset of arms $\mathcal{S} \in \mathfrak{B}$.

Step 2. Bound the difference of two consecutive weighted summations. In this step, we upper bound the difference of two consecutive weighted summations for the subset of arms $\mathcal{S}$, i.e.,

$$s_{t+1}^{(\mathcal{S})} - s_t^{(\mathcal{S})} = \mathbb{1}\{\mathcal{S} = \mathcal{S}_{t+1}\} \sum_{i \in \mathcal{S}} \frac{1}{(\Delta'_k)^2} \left(\langle \psi_{i,t+1} | \psi_{0,t+1} \rangle - \langle \psi_{i,t} | \psi_{0,t} \rangle\right).$$

With similar derivation as the Step 2 proof of Theorem 6, we have

$$\begin{aligned}
&\left| s_{t+1}^{(\mathcal{S})} - s_t^{(\mathcal{S})} \right| \\
&\leqslant \mathbb{1}\{\mathcal{S}_{t+1} = \mathcal{S}\} \left(1 + \frac{1}{\sqrt{(p-\epsilon)(1-(p-\epsilon))}}\right) \sqrt{\sum_{i \in \mathcal{S}} \frac{1}{(\Delta'_i)^2}}
\end{aligned}$$

Step 3. Combine Steps 1 and 2 to conclude query complexity lower bound. For the fixed subset of arms $\mathcal{S}$, we have

$$
\begin{aligned}
&\sum_{i\in\mathcal{S}} \frac{1}{(\Delta_i')^2}(1-2\sqrt{\delta(1-\delta)}) \\
&\overset{\text{(Step 1)}}{\leqslant} s_0^{(\mathcal{S})} - s_T^{(\mathcal{S})} \\
&\leqslant \left|s_0^{(\mathcal{S})} - s_1^{(\mathcal{S})}\right| + \cdots + \left|s_{T-1}^{(\mathcal{S})} - s_T^{(\mathcal{S})}\right| \\
&\overset{\text{(Step2)}}{\leqslant} \sum_{t=1}^{T} \mathbb{1}\{\mathcal{S}_t = \mathcal{S}\}\left(1+\frac{1}{\sqrt{(p-\epsilon)(1-(p-\epsilon))}}\right)\sqrt{\sum_{i\in\mathcal{S}_t}\frac{1}{(\Delta_i')^2}} \\
&\leqslant T^{(\mathcal{S})}\left(1+\frac{1}{\sqrt{(p-\epsilon)(1-(p-\epsilon))}}\right)\cdot\sqrt{\sum_{i\in\mathcal{S}}\frac{1}{(\Delta_i')^2}},
\end{aligned}
$$

where we denote $T^{(\mathcal{S})} := \sum_{t=1}^{T} \mathbb{1}\{\mathcal{S}_t = \mathcal{S}\}$ as the number of times that the arm subset $\mathcal{S}$ is queried among T rounds.

Rearranging this inequality yields

$$T^{(\mathcal{S})} \geqslant \frac{1-2\sqrt{\delta(1-\delta)}}{1+1/\sqrt{(p-\epsilon)(1-(p-\epsilon))}}\cdot\sqrt{\sum_{i\in\mathcal{S}}\frac{1}{(\Delta_i')^2}}$$

Then, with similar derivations as the Step 3 proof of Theorem 6, we have

$$T^{(\mathcal{S})} \geqslant \frac{16}{25}\frac{1-2\delta(1-\delta)}{1+1/\sqrt{(p/2)(1-(p/2))}}\sqrt{\sum_{i\in\mathcal{S}}\frac{1}{\Delta_i^2}}.$$

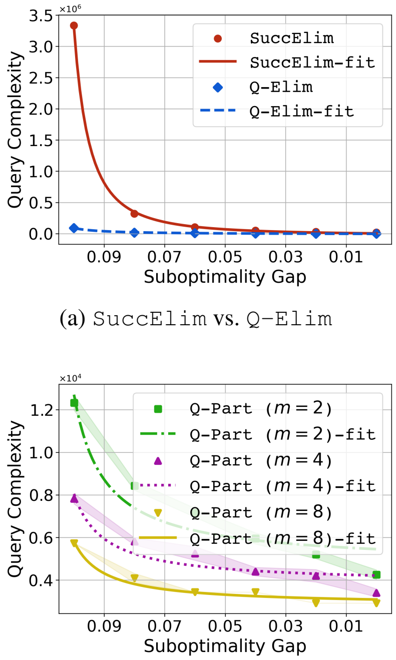
Notice that $T = \sum_{\mathcal{S}\in\mathfrak{B}} T^{(\mathcal{S})}$, we then prove the query complexity lower bound as follows,

$$T \geqslant \frac{16}{25}\frac{1-2\delta(1-\delta)}{1+1/\sqrt{(p/2)(1-(p/2))}}\sum_{\mathcal{S}\in\mathfrak{B}}\sqrt{\sum_{i\in\mathcal{S}}\frac{1}{\Delta_i^2}}.$$

□

H Detail of `Qiskit` Implementation of Quantum Subroutines

For our quantum algorithms, we use the `Qiskit` Python package (Qiskit contributors 2023) to implement all quantum subroutines (`QuEst`, `Amplify`, and `Estimate`), except for the `VTA` subroutine, which does not have a standard Qiskit implementation. Instead, we use a simplified quantum circuit that produces the same output as described in (6). Consequently, when counting the query complexity contribution from `VTA`, we use its theoretical upper bound from Lemma 5 (with the constant coefficient hidden by the big-O notation taken to be 1, noting that the constant may not be exactly 1; the true performance of `Q-Part` could differ from the numerical values presented by a multiplicative factor, making its numerical performance not directly comparable with `SuccElim` and `Q-Elim`) and the polylog factor in the expression for G taken to be $\log\left(\frac{m}{\delta(b-a)}\right)$ (the polynomial degree of the logarithmic factor could potentially impact the relative performance of `Q-Part` with different values of m, compared to the values depicted in Figure 1b; however, in the asymptotic limit of small gap Δ or large number of arms K, the relative order will agree with that illustrated in Figure 1b).



(a) `SuccElim` vs. `Q-Elim`

(b) `Q-Part` with different m

Figure 2: Curve-fitting of the empirical performance evaluation of `Q-Elim` and `Q-Part`. The fitting curves are $y = a/x^2 + b$ for `SuccElim` and $y = c/x + d$ for the quantum algorithms.

To further illustrate the relation between the empirical query times with our theoretical query complexity bounds, we provide fitting curves for the empirical data in Figure 2. For fitting, we used $y = a/x^2 + b$ for the `SuccElim` (classical) plot and $y = c/x + d$ for the quantum plots, where y represents empirical sample/query counts, x is the gap Δ, and a, b, c, and d are constants. Although we cannot include new figures in the rebuttal, we report the R^2 values (closer to 1 indicates a better fit), and we will add these figures to the final paper version.

In Figure 2a, the fitting curve for `SuccElim` achieved $R^2 = 0.9998$, and for `QElim`, $R^2 = 0.9941$. For Figure 2b, the R^2 values for `QElim` with $m = 2, 4, 8$ are 0.9159, 0.8854, and 0.9647, respectively. These results confirm that the query complexity of the quantum algorithms indeed depends on $1/\Delta$, aligning with our theoretical findings. The slightly lower R^2 values for Figure 2b are likely due to the actual sample complexity bound of `Q-Part`, which involves $\sum_{\mathcal{S}\in\mathfrak{B}}\sqrt{\sum_{k\in\mathcal{S}}\frac{1}{\Delta_k^2}}$ and depends on the arm partition $\mathfrak{B}$, approximating but not strictly matching $1/\Delta$ dependence.