Quantum exploration algorithms for multi-armed bandits

Daochen Wang*[1,2], Xuchen You*[1,3], Tongyang Li[1,3], and Andrew M. Childs[1,3]

[1]Joint Center for Quantum Information and Computer Science, University of Maryland
[2]Department of Mathematics, University of Maryland
[3]Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland

Abstract

Identifying the best arm of a multi-armed bandit is a central problem in bandit optimization. We study a quantum computational version of this problem with coherent oracle access to states encoding the reward probabilities of each arm as quantum amplitudes. Specifically, we show that we can find the best arm with fixed confidence using $\tilde{O}\big(\sqrt{\sum_{i=2}^{n}\Delta_i^{-2}}\big)$ quantum queries, where Δ_i represents the difference between the mean reward of the best arm and the i^{th}-best arm. This algorithm, based on variable-time amplitude amplification and estimation, gives a quadratic speedup compared to the best possible classical result. We also prove a matching quantum lower bound (up to poly-logarithmic factors).

1 Introduction

The multi-armed bandit (MAB) model is one of the most fundamental settings in reinforcement learning. This simple scenario captures crucial issues such as the tradeoff between exploration and exploitation. Furthermore, it has wide applications to areas including operations research, mechanism design, and statistics.

A basic challenge about multi-armed bandits is the problem of *best-arm identification*, where the goal is to efficiently identify the arm with the largest expected reward. This problem captures a common difficulty in practical scenarios, where at unit cost, only partial information about the system of interest can be obtained. A real-world example is a recommendation system, where the goal is to find appealing items for users. For each recommendation, only feedback on the recommended item is obtained. In the context of machine learning, best-arm identification can be viewed as a high-level abstraction and core component of active learning, where the goal is to minimize the uncertainty of an underlying concept, and each step only reveals the label of the data point being queried.

Quantum computing is a promising technology with potential applications to diverse areas including cryptanalysis, optimization, and simulation of quantum physics. Quantum computing devices have recently been demonstrated to experimentally outperform classical computers on a specific sampling task [8]. While noise limits the current practical usefulness of quantum computers, they can in principle be made fault tolerant and thus capable of executing a wide variety of algorithms. It is therefore of significant interest to understand quantum algorithms from a theoretical perspective to anticipate future applications. In particular, there has been increasing interest

*Equal contribution.

in *quantum machine learning* (see for example the surveys [6, 12, 19, 40]). In this paper, we study best-arm identification in multi-armed bandits, establishing quantum speedup.

Problem setup. We work in a standard multi-armed bandit setting [24] in which the MAB has n arms, where arm $i \in [n] \coloneqq \{1, \ldots, n\}$ is a Bernoulli random variable taking value 1 with probability p_i and value 0 with probability $1 - p_i$. Each arm can therefore be regarded as a coin with *bias* p_i. As our algorithms and lower bounds are symmetric with respect to the arms, we assume without loss of generality that $p_1 \geq \cdots \geq p_n$, and denote $\Delta_i \coloneqq p_1 - p_i$ for all $i \in \{2, \ldots, n\}$. We further assume that $p_1 > p_2$, i.e., the best arm is unique. Given a parameter $\delta \in (0, 1)$, our goal is to use as few queries as possible to determine the best arm with probability $\geq 1 - \delta$. This is known as the *fixed confidence setting*. We primarily characterize complexity in terms of the parameter

$$H \coloneqq \sum_{i=1}^{n} \frac{1}{\Delta_i^2} \tag{1.1}$$

which arises in the analysis of classical MAB algorithms (as discussed below).

We consider a quantum version of best-arm identification in which we can access the arms *coherently*. More precisely, we assume we have access to a quantum oracle $\mathcal{O}$ that acts as

$$\mathcal{O}\colon\ |i\rangle_I\,|0\rangle_B \mapsto |i\rangle_I\,|\mathrm{coin}\,p_i\rangle_B\ \ \forall\, i \in [n],\ \text{ where }\ |\mathrm{coin}\,p\rangle \coloneqq \sqrt{p}\,|1\rangle + \sqrt{1-p}\,|0\rangle \tag{1.2}$$

for any $p \in [0, 1]$. Register I is the "index" register with n states that correspond to the n arms. Register B is the single-qubit "bandit" register with two states, $|1\rangle$ corresponding to a reward and $|0\rangle$ corresponding to no reward. For convenience, we omit register labels when this causes no confusion. Compared to pulling an arm classically—which can be implemented using $\mathcal{O}$ by simply measuring the bandit register—the quantum oracle enables accessing different arms in superposition, an essential feature enabling the possibility of quantum speedup.

Previous work on quantum algorithms for clustering [32, 41] and reinforcement learning [19, 21] has discussed how to instantiate an oracle similar to[1] $\mathcal{O}$. In clustering, $\mathcal{O}$ is created using the SWAP test where for each i, p_i encodes the distance between some fixed vector and the i^{th} vector in some collection. Our algorithm can be used to speed up the algorithms of Refs. [32, 41]. In reinforcement learning, $\mathcal{O}$ naturally appears in stochastic agent environments; for instance, $\mathcal{O}$ can be viewed as a special case of the oracle in [21] for a Markov decision problem (MDP) of epoch length 1 and state set $\{0, 1\}$, where the goal of the agent is to reach the state 1.

Contributions. In this paper, we give a comprehensive study of best-arm identification using quantum algorithms. Specifically, we obtain the following main result:

Theorem 1. *Given a multi-armed bandit oracle $\mathcal{O}$ and confidence parameter $\delta \in (0, 1)$, there exists a quantum algorithm that, with probability $\geq 1 - \delta$, outputs the best arm using $\tilde{O}(\sqrt{H})$ queries to $\mathcal{O}$. Moreover, this query complexity is optimal up to poly-logarithmic factors in n, δ, and Δ_2.*

This represents a quadratic quantum speedup over what is possible classically. The speedup essentially derives from Grover's search algorithm [26] (see Appendix A for background), where a marker oracle is used to approximately "rotate" a uniform initial state to the marked state. One way to understand the quadratic speedup is to observe that each rotation step, making one query

[1] Technically, the oracle $\tilde{\mathcal{O}}$ in those references maps $|i\rangle\,|0\rangle \mapsto |i\rangle\,(\sqrt{p_i}\,|1\rangle\,|v_i\rangle + \sqrt{1-p_i}\,|0\rangle\,|u_i\rangle)$ for some states $|v_i\rangle\,, |u_i\rangle$. It is not difficult to see that the algorithm we propose works unchanged in this case. Trivially, our lower bound also applies to this model.

to the oracle, increases the amplitude of the marked state by $\Omega(1/\sqrt{n})$. This is possible due to the fact that quantum computation manipulate amplitudes, which are square roots of probabilities.

However, to establish Theorem 1 we use more sophisticated machinery that extends Grover's algorithm, namely variable-time amplitude amplification (VTAA) [4, 18] and estimation (VTAE) [15]. We apply VTAA and VTAE on a variable-time quantum algorithm $\mathcal{A}$ that we construct. $\mathcal{A}$ outputs a state with labeled "good" and "bad" parts. Using that label, VTAA removes the bad part so that only the good part remains, and VTAE estimates the proportion of the good part. In our application, the good part is eventually the best-arm state. If we instead used standard amplitude amplification and estimation, which are more immediate corollaries of Grover's algorithm, we would get a worse complexity of $\tilde{O}(\sqrt{n}/\Delta_2)$.

We emphasize that our quantum algorithm, like classical ones [24, 25, 29, 31, 37], does not require any prior knowledge about the p_i. This is achieved in two stages: first, we locate p_1 and p_2 through binary search with a carefully chosen stopping condition; second, we use the location of p_1 and p_2 to rotate to the best-arm state. Note that classical algorithms do not normally need to have two separate stages. This is because classical arm samples, stored in memory, contain information about both the location of the best arm as well as p_1 and p_2. However, in the quantum setting, we cannot simply obtain many such classical samples as this would collapse the quantum state, preventing quantum speedup.

Related work. Classically, a naive algorithm for best-arm identification is to simply sample each arm the same number of times and output the arm with the best empirical bias [24]. This algorithm has complexity $O(\frac{n}{\Delta_2^2}\log(\frac{n}{\delta}))$ but is sub-optimal for most multi-armed bandit instances. Therefore, classical research on best-arm identification [24, 25, 29, 31, 37] has primarily focused on proving bounds of the form $\tilde{O}(H)$ (recall that $H \coloneqq \sum_{i=2}^{n}\frac{1}{\Delta_i^2}$), which can be shown to be almost tight for every instance. The first work to provide an algorithm with such complexity is Ref. [24], giving $O(H\log(\frac{n}{\delta}) + \sum_{i=2}^{n}\Delta_i^{-2}\log(\Delta_i^{-1}))$. This was further improved to $O\big(H\log(\frac{1}{\delta}) + \sum_{i=2}^{n}\Delta_i^{-2}\log\log(\Delta_i^{-1})\big)$ by Refs. [25, 29, 31], which is almost optimal [37], except for the additive term of $\sum_{i=2}^{n}\Delta_i^{-2}\log\log(\Delta_i^{-1})$. More recent work [16, 17] has focused on bringing down even this additive term by tightening both the upper and lower bounds, leaving behind a gap only of the order $\sum_{i=2}^{n}\Delta_i^{-2}\log\log(\min\{n,\Delta_i^{-1}\})$.

Prior work on quantum machine learning has focused primarily on supervised [33, 35, 36, 39] and unsupervised learning [5, 32, 35, 41]. Refs. [20, 22, 30] give quantum algorithms for general reinforcement learning with provable guarantees, but do not consider the best-arm identification problem. The only directly comparable previous work on quantum algorithms for best-arm identification that we are aware of are [14] and [41].[2] By applying Grover's algorithm, Ref. [14] shows that quantum computers can find the best arm with confidence $p_1/\sum_{i=1}^{n}p_i$ quadratically faster than classical ones. However, Ref. [14] does not show how to find the best arm with arbitrarily high confidence. In fact, there is a relatively simple quantum algorithm, analogous to the naive classical algorithm, that can achieve arbitrarily high confidence with quadratic speedup in terms of n/Δ_2^2. This algorithm, which appears in Ref. [41, Fig. 3], works by using the quantum minimum finding of Dürr and Høyer [23] on top of quantum amplitude estimation [13]. As in the classical case, we show that this simple quantum algorithm is suboptimal for most multi-armed bandit instances. Specifically, we show that a quantum algorithm can achieve quadratic speedup in terms of the parameter H.

Open questions. This work leaves several natural open questions for future investigation:

[2] Ref. [41] is not framed as being about best-arm identification but is partly concerned with exactly this.

- Can we give fast quantum algorithms for the exploitation of multi-armed bandits? In particular, can we give online algorithms with favorable regret? The quantum hedging algorithm [27] and the quantum boosting algorithm [7] might be relevant to this challenge.

- Can we give fast quantum algorithms for other types of multi-armed bandits, such as contextual bandits or adversarial bandits (e.g. [1, 9, 11])?

- Can we give fast quantum algorithms for finding a near-optimal policy of a Markov decision process (MDP)? MDPs are a natural generalization of MABs, where the goal is to maximize the expected reward over sequences of decisions. Ref. [24] gives a reduction from this problem to best-arm identification by viewing the Q-function of each state as a multi-armed bandit.

2 Variable-time amplitude amplification and estimation

In this section we review variable-time amplitude amplification (VTAA) and estimation (VTAE), which are essential components of our algorithm. VTAA and VTAE are procedures applied on top of so-called "variable-time" quantum algorithms, which can be formally defined as follows:

Definition 1 (Variable-time quantum algorithm, cf. [4, Section 3.3] and [18, Section 5.1])**.** *Let $\mathcal{A}$ be a quantum algorithm in a space $\mathcal{H}$ that starts in the state $|0\rangle_{\mathcal{H}}$, the all-zeros state in $\mathcal{H}$. We say $\mathcal{A}$ is a* variable-time quantum algorithm *if the following conditions hold:*

1. *$\mathcal{A}$ is the product of m sub-algorithms, $\mathcal{A} = \mathcal{A}_m\mathcal{A}_{m-1}\cdots\mathcal{A}_1$.*
2. *$\mathcal{H}$ is a tensor product $\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_A$, where $\mathcal{H}_C$ is a tensor product of m single-qubit registers denoted $\mathcal{H}_{C_1}, \mathcal{H}_{C_2}, \ldots, \mathcal{H}_{C_m}$.*
3. *Each $\mathcal{A}_j$ is a controlled unitary that acts on the registers $\mathcal{H}_{C_j} \otimes \mathcal{H}_A$ controlled on the first $j-1$ qubits of $\mathcal{H}_C$ being set to $|0\rangle$.*
4. *The final state of the algorithm, $\mathcal{A}|0\rangle_{\mathcal{H}}$, is perpendicular to $|0\rangle_C \coloneqq |0\rangle_{C_1}|0\rangle_{C_2}\cdots|0\rangle_{C_m}$.*

In each iteration of the variable-time algorithm we shall construct, we use a subroutine that we call *gapped amplitude estimation* (GAE). Standard amplitude estimation [13] performs phase estimation on a particular unitary, and GAE is essentially the same as "gapped phase estimation" [18, Lemma 22] of that unitary. We recall the standard technique of amplitude estimation [13]:

Theorem 2 (Amplitude estimation)**.** *Suppose $\mathcal{O}_p$ is a unitary with $\mathcal{O}_p|0\rangle_B = |\mathrm{coin}\, p\rangle_B$. Then there is a unitary procedure $\mathsf{AE}(\epsilon,\delta)$, making $O(\frac{1}{\epsilon}\log\frac{1}{\delta})$ queries to $\mathcal{O}_p$ and $\mathcal{O}_p^\dagger$, that on input $|\mathrm{coin}\, p\rangle_B|0\rangle_P$ prepares a state of the form*

$$|\mathrm{coin}\, p\rangle_B\Big(\sum_{p'}\alpha_{p'}|p'\rangle_P + \alpha|p_\perp\rangle_P\Big), \tag{2.1}$$

where $|\alpha| \coloneqq \sqrt{1-\sum_{p'}|\alpha_{p'}|^2} \le \delta$, $\langle p'|p_\perp\rangle = 0$ for all p', and $|p'-p| \le \epsilon$ for all p'.

Strictly speaking, the parts of Theorem 2 involving δ come from measuring the output state of the original amplitude estimation procedure [13] $O(\log\frac{1}{\delta})$ times and taking the median. This can be made coherent by the principle of deferred measurement. Theorem 2 implies the following:

Corollary 1 (Gapped amplitude estimation)**.** *Suppose $\mathcal{O}_p$ is a unitary with $\mathcal{O}_p\,|0\rangle = |\mathrm{coin}\,p\rangle$. Then there is a unitary procedure $\mathsf{GAE}(\epsilon,\delta;l)$, making $O(\frac{1}{\epsilon}\log\frac{1}{\delta})$ queries to $\mathcal{O}_p$ and $\mathcal{O}_p^\dagger$, that on input $|\mathrm{coin}\,p\rangle_B\,|0\rangle_C\,|0\rangle_P$, prepares a state of the form*

$$|\mathrm{coin}\,p\rangle_B\,(\beta_0\,|0\rangle_C\,|\gamma_0\rangle_P + \beta_1\,|1\rangle_C\,|\gamma_1\rangle_P), \tag{2.2}$$

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

Proof. We first run $\mathsf{AE}(\epsilon/4,\delta)$ on registers B, P. Then, in register C, we output 1 if the value stored in register P is closer to $l-\epsilon$, and output 0 if it is closer to $l-2\epsilon$. This gives the desired unitary procedure. For convenience, we put any phase factors on the β_i into the $|\gamma_i\rangle$. $\square$

Theorem 3 (Variable-time amplitude amplification and estimation [4, 15, 18])**.** *Let $\mathcal{A} = \mathcal{A}_m\cdots\mathcal{A}_1$ be a variable-time quantum algorithm on the space $\mathcal{H} = \mathcal{H}_C\otimes\mathcal{H}_F\otimes\mathcal{H}_W$. Let $|0\rangle_{\mathcal{H}}$ be the all-zeros state in $\mathcal{H}$ and let t_j be the query complexity of the algorithm $\mathcal{A}_j\cdots\mathcal{A}_1$. We define*

$$w_j := \|\Pi_{C_j}\mathcal{A}_j\cdots\mathcal{A}_1\,|0\rangle_{\mathcal{H}}\|^2 \qquad \text{and} \qquad t_{\mathrm{avg}} := \sqrt{\textstyle\sum_{j=1}^m w_j t_j^2} \tag{2.3}$$

to be the probability of halting at step j and the root-mean-square average query complexity of the algorithm, respectively, where Π_{C_j} denotes the projector onto $|1\rangle$ in $\mathcal{H}_{C_j}$. We also define

$$p_{\mathrm{succ}} := \|\Pi_F\mathcal{A}_m\cdots\mathcal{A}_1\,|0\rangle_{\mathcal{H}}\|^2 \quad \text{and} \quad |\psi_{\mathrm{succ}}\rangle := \frac{\Pi_F\mathcal{A}_m\cdots\mathcal{A}_1\,|0\rangle_{\mathcal{H}}}{\|\Pi_F\mathcal{A}_m\cdots\mathcal{A}_1\,|0\rangle_{\mathcal{H}}\|} \tag{2.4}$$

to be the success probability of the algorithm and the corresponding output state, respectively, where Π_F projects onto $|1\rangle$ in $\mathcal{H}_F$. Then there exists a quantum algorithm that uses $O(Q)$ queries to output the state $|\psi_{\mathrm{succ}}\rangle$ with probability $\ge 1/2$ and a bit indicating whether it succeeds, where

$$Q := t_m\log(t_m) + \frac{t_{\mathrm{avg}}}{\sqrt{p_{\mathrm{succ}}}}\log(t_m). \tag{2.5}$$

There also exists a quantum algorithm that uses $O(\frac{Q}{\epsilon}\log^2(t_m)\log\log(\frac{t_m}{\delta}))$ queries to estimate p_{succ} with multiplicative error ϵ with probability $\ge 1-\delta$.

3 Fast quantum algorithm for best-arm identification

In this section, we construct a quantum algorithm for best-arm identification and analyze its performance. Specifically, we show the following:

Theorem 4. *Given a multi-armed bandit oracle $\mathcal{O}$ and confidence parameter $\delta \in (0,1)$, there exists a quantum algorithm that outputs the best arm with probability $\ge 1-\delta$ using $\tilde{O}(\sqrt{H})$ queries to $\mathcal{O}$.*

Throughout this section, we regard the oracle $\mathcal{O}$ as fixed, so we sometimes omit explicit reference to it. All logs have base 2.

Our construction proceeds in three steps, which we now describe at a high level.

First, in Section 3.1, we construct a variable-time quantum algorithm denoted $\mathcal{A}$ (Algorithm 1) that is initialized in a uniform superposition state $|u\rangle := \frac{1}{\sqrt{n}}\sum_{i\in[n]}|i\rangle$ (since initially we have no information about which arm is the best). Given an input interval $I = [l_2, l_1]$, $\mathcal{A}$ "flags" arm indices in $S'_{\mathrm{right}} := \{i \in [n] : p_i \ge l_1\}$ with a bit $f = 1$ and those in $S'_{\mathrm{left}} := \{i \in [n] : p_i \le l_2\}$ with a bit $f = 0$. The flag bit f is written to a separate flag register F, so that the state (approximately) becomes $\frac{1}{\sqrt{n}}\left(\sum_{i\in S'_{\text{right}}} |i\rangle |1\rangle_F + \sum_{i\in S'_{\text{left}}} |i\rangle |0\rangle_F + \sum_{i\in S'_{\text{middle}}} |i\rangle |\psi_i\rangle_F\right)$ for some qubit states $|\psi_i\rangle$, where $S'_{\text{middle}} \coloneqq [n] - (S'_{\text{left}} \cup S'_{\text{right}}) = \{i \in [n] : l_2 < p_i < l_1\}$. The flag bit f stored in the F register indicates whether VTAA (resp. VTAE), when applied on $\mathcal{A}$, should ($f = 1$) or should not ($f = 0$) amplify (resp. estimate) that part of the state.

Second, in Section 3.2, we apply VTAA and VTAE on $\mathcal{A}$ to construct two new algorithms called Amplify and Estimate, respectively. Amplify produces a uniform superposition of all those is with F register in $|1\rangle$, while Estimate counts the number of such is. More precisely, Estimate (approximately) counts the number of indices in S'_{right}, as their F register is in $|1\rangle$, plus some (unknown) fraction of indices in S'_{middle} as dictated by the fraction of $|1\rangle$ in the (unknown) states $|\psi_i\rangle$.

Third, in Section 3.3, we use Estimate as a subroutine in Locate (Algorithm 2) to find l_2, l_1 such that $p_2 < l_2 < l_2 + \Delta_2/4 \le l_1 < p_1$. Then, running Amplify with these l_2, l_1 in BestArm (Algorithm 4) gives the state $|1\rangle$ containing the best-arm index because only p_1 is to the right of l_2. Locate is a type of binary search that counts the number of indices in S'_{right} using Estimate. There is a technical difficulty here because Estimate actually counts the number of indices in S'_{right} plus some fraction of indices in S'_{middle}. Trying to fix this by simply setting $l_2 = l_1$, so that $S'_{\text{middle}} = \emptyset$, does not work as it would increase the cost of Estimate. We overcome this difficulty via the Shrink subroutine (Algorithm 3) of Locate, which employs a technique from recent work on quantum ground state preparation [34].

3.1 Variable-time quantum algorithm

We first construct a variable-time quantum algorithm (Algorithm 1) that we call $\mathcal{A}$ throughout. $\mathcal{A}$ uses the following registers: input register I; bandit register B; clock register $C = (C_1, \ldots, C_{m+1})$, where each C_i is a qubit; ancillary amplitude estimation register $P = (P_1, \ldots, P_m)$, where each P_i has $O(m)$ qubits; and flag register F. We set $m \coloneqq \lceil \log(1/(l_1 - l_2)) \rceil + 2$ as assigned in Algorithm 1.

$\mathcal{A}$ is indeed a variable-time quantum algorithm according to Definition 1. This is because we can write $\mathcal{A} = \mathcal{A}_{m+1}\mathcal{A}_m \cdots \mathcal{A}_1\mathcal{A}_0$ as a product of $m + 2$ sub-algorithms, where $\mathcal{A}_0$ is the initialization step (Line 4), $\mathcal{A}_j$ consists of the operations in iteration j of the for loop (Lines 6–9) for $j \in [m]$, and $\mathcal{A}_{m+1}$ is the termination step (Lines 10–11). The state spaces $\mathcal{H}_C$ and $\mathcal{H}_A$ in Definition 1 correspond to the state spaces of the C register and the remaining registers of $\mathcal{A}$, respectively. $\mathcal{A}_{m+1}$ ensures that Condition 4 of Definition 1 is satisfied.

We define the following three sets that partition $[n]$:

$$S_{\text{left}} \coloneqq \{i \in [n] : p_i < l_1 - \Delta/2\}, \tag{3.1}$$

$$S_{\text{middle}} \coloneqq \{i \in [n] : l_1 - \Delta/2 \le p_i < l_1 - \Delta/8\}, \tag{3.2}$$

$$S_{\text{right}} \coloneqq \{i \in [n] : p_i \ge l_1 - \Delta/8\}. \tag{3.3}$$

Note that these sets play the roles of S'_{left}, S'_{middle}, and S'_{right} mentioned at the start of Section 3. They can be regarded as functions of (the input to) $\mathcal{A}$. In the following, to quantify the proximity of quantum states, we say that $|\psi\rangle$ and $|\phi\rangle$ are ϵ-close if $\| |\psi\rangle - |\phi\rangle \| \le \epsilon$.

Algorithm 1: $\mathcal{A}(\mathcal{O}, l_2, l_1, \alpha)$

```
   Input: Oracle O as in (1.2); 0 < l_2 < l_1 < 1; approximation parameter 0 < α < 1.
1  Δ ← l_1 − l_2
2  m ← ⌈log (1/Δ)⌉ + 2
3  a ← α / (2 m n^{3/2})
4  Initialize state to (1/√n) Σ_{i=1}^{n} |i⟩_I |coin p_i⟩_B |0⟩_C |0⟩_P |1⟩_F
5  for j = 1, ..., m do
6      ε_j ← 2^{−j}
7      if register I is in state |i⟩ and registers C_1, ..., C_{j−1} are in state |0⟩ then
8          Apply GAE(ε_j, a; l_1) with O_{p_i} on registers B, C_j, and P_j
9      Apply controlled-NOT gate with control on register C_j and target on register F
10 if registers C_1, ..., C_m are in state |0⟩ then
11     Flip the bit stored in register C_{m+1}
```

Lemma 1 (Correctness of $\mathcal{A}$)**.** *The output state $|\phi(\mathcal{A})\rangle$ of $\mathcal{A}$ is (α/n)-close to*

$$\begin{aligned}|\psi(\mathcal{A})\rangle \coloneqq\ & \frac{1}{\sqrt{n}} \sum_{S_{\text{right}}} |i\rangle_I\, |\text{coin}\, p_i\rangle_B\, |\psi_i\rangle_{C,P}\, |1\rangle_F \\ &+ \frac{1}{\sqrt{n}} \sum_{S_{\text{left}}} |i\rangle_I\, |\text{coin}\, p_i\rangle_B\, |\psi_i\rangle_{C,P}\, |0\rangle_F \\ &+ \frac{1}{\sqrt{n}} \sum_{S_{\text{middle}}} |i\rangle_I\, |\text{coin}\, p_i\rangle_B \left(\beta_{i,1}\, |\psi_{i,1}\rangle_{C,P}\, |1\rangle_F + \beta_{i,0}\, |\psi_{i,0}\rangle_{C,P}\, |0\rangle_F\right)\end{aligned}$$

for some $\beta_{i,1}, \beta_{i,0} \in \mathbb{C}$ and states $|\psi_i\rangle, |\psi_{i,j}\rangle$. In particular, we have $|p_{\text{succ}} - p'_{\text{succ}}| \leq \frac{2\alpha}{n}$ where $p_{\text{succ}} \coloneqq \|\Pi_F\, |\phi(\mathcal{A})\rangle\, \|^2$ and $p'_{\text{succ}} \coloneqq \|\Pi_F\, |\psi(\mathcal{A})\rangle\, \|^2 = \frac{1}{n}\big(|S_{\text{right}}| + \sum_{i \in S_{\text{middle}}} |\beta_{i,1}|^2\big)$.

At a high level, at iteration j, Line 8 approximately identifies those $i \in S_{\text{left}}$ with $p_i \in [l_1 - 2\epsilon_j, l_1 - \epsilon_j)$ and stops computation on these is by setting their associated C registers to $|1\rangle$. Line 9 then flags these is by setting their associated F registers to $|0\rangle$, indicating failure. We give the detailed proof in Appendix B, which is mainly concerned with bounding the error in the aforementioned approximation. We also defer the proof of the following lemma about the complexity of $\mathcal{A}$ to Appendix B.

Lemma 2 (Complexity of $\mathcal{A}$)**.** *$\mathcal{A}(\mathcal{O}, l_2, l_1, \alpha)$ has the following complexities:*

1. *The j^{th} stopping time t_j of $\mathcal{A}_j \mathcal{A}_{j-1} \cdots \mathcal{A}_0$ is of order $\sum_{k=1}^{j} \frac{1}{\epsilon_k} \log \frac{1}{a} \leq 2^{j+1} \log \frac{1}{a}$. In particular, $t_{m+1} = O(\frac{1}{\Delta} \log \frac{1}{a})$.*
2. *The average stopping time squared, t_{avg}^2, is of order*

$$\frac{1}{n}\left(\frac{|S_{\text{right}}|}{\Delta^2} + \sum_{i \in S_{\text{left}} \cup S_{\text{middle}}} \frac{1}{(l_1 - p_i)^2}\right) \log^2\left(\frac{1}{a}\right). \tag{3.4}$$

3.2 Applying VTAA and VTAE to $\mathcal{A}$

In this subsection, we fix algorithm $\mathcal{A}$ and its input parameters. We always assume that $|S_{\text{right}}| > 0$, so $|S_{\text{right}}| \geq 1$, which we need for some of the following results to hold. This is without loss of generality as we can always add an artificial arm 0 to $\mathcal{O}$ with bias $p_0 = 1$, as we do in Line 3 of Algorithm 3.

We apply[3] VTAA and VTAE (Theorem 3) on our variable-time quantum algorithm $\mathcal{A}$ to prepare the state $|\psi_{\text{succ}}\rangle$ and estimate the probability p_{succ}, respectively. This gives two new algorithms $\mathsf{Amplify}$ and $\mathsf{Estimate}$ with the following performance guarantees.

Lemma 3 (Correctness and complexity of $\mathsf{Amplify}(\mathcal{A}, \delta)$, $\mathsf{Estimate}(\mathcal{A}, \epsilon, \delta)$)**.** *Let $\mathcal{A} = \mathcal{A}(\mathcal{O}, l_2, l_1, 0.01\delta)$. Then $\mathsf{Amplify}(\mathcal{A}, \delta)$ uses $O(Q)$ queries to output an index $i \in S_{\text{right}} \cup S_{\text{middle}}$ with probability $\geq 1-\delta$, and $\mathsf{Estimate}(\mathcal{A}, \epsilon, \delta)$ uses $O(Q/\epsilon)$ queries to output an estimate r of p'_{succ} (defined in Lemma 1) such that*

$$(1-\epsilon)\Big(p'_{\text{succ}} - \frac{0.1}{n}\Big) < r < (1+\epsilon)\Big(p'_{\text{succ}} + \frac{0.1}{n}\Big) \tag{3.5}$$

with probability $\geq 1 - \delta$, where

$$Q = \bigg(\frac{1}{\Delta^2} + \frac{1}{|S_{\text{right}}|} \sum_{S_{\text{left}} \cup S_{\text{middle}}} \frac{1}{(l_1 - p_i)^2}\bigg) \operatorname{poly}\Big(\log\Big(\frac{n}{\delta\Delta}\Big)\Big). \tag{3.6}$$

Proof sketch. Apply Lemma 1 and Lemma 2 to Theorem 3. See Appendix B for details. ☐

3.3 Quantum algorithm for best-arm identification

In this subsection, we use $\mathsf{Amplify}$ and $\mathsf{Estimate}$ to construct three algorithms (Algorithms 2–4) that work together to identify the best arm following the outline at the start of Section 3.

Algorithm 2: $\mathsf{Locate}(\mathcal{O}, \delta)$

Input: Oracle $\mathcal{O}$ as in (1.2); confidence parameter $0 < \delta < 1$.

1 $I_1, I_2 \leftarrow [0, 1]$
2 $\delta \leftarrow \delta/8$
3 **while** $\min I_1 - \max I_2 < 2\,|I_1|$ **do**
4 $\quad I_1 \leftarrow \mathsf{Shrink}(\mathcal{O}, 1, I_1, \delta)$
5 $\quad I_2 \leftarrow \mathsf{Shrink}(\mathcal{O}, 2, I_2, \delta)$
6 $\quad \delta \leftarrow \delta/2$
7 **return** I_1, I_2

Lemma 4 (Correctness and complexity of Algorithm 2)**.** *Fix a confidence parameter $0 < \delta < 1$. Then the event $E = \{p_1 \in I_1$ and $p_2 \in I_2$ in all iterations of the while loop$\}$ holds with probability $\geq 1 - \delta$. When E holds, Algorithm 2 also satisfies the following for both $k \in \{1, 2\}$:*

1. *its while loop (Line 3) breaks at or before the end of iteration $\lceil \log_{5/3}(\frac{1}{\Delta_2}) \rceil + 3$ and then returns I_k with $p_k \in I_k$ and $\min I_1 - \max I_2 \geq 2\,|I_1|$; during the while loop, we always have $|I_1| = |I_2| \geq \Delta_2/8$; and*
2. *it uses $O\big(\sqrt{H} \operatorname{poly}\big(\log(\frac{n}{\delta\,\Delta_2})\big)\big)$ queries.*

Lemma 5 (Correctness and complexity of Algorithm 3)**.** *Fix $k \in \{1, 2\}$, an interval $I = [a, b]$, and a confidence parameter $0 < \delta < 1$. Suppose that $p_k \in I$ and $|I| \geq \Delta_2/8$. Then Algorithm 3*

1. *outputs an interval J with $|J| = \frac{3}{5}\,|I|$ such that $p_k \in J$ with probability $\geq 1 - \delta$, and*

[3]The state spaces $\mathcal{H}_C$, $\mathcal{H}_F$, and $\mathcal{H}_W$ in Theorem 3 correspond to the state spaces of the C, F, and remaining registers of $\mathcal{A}$, respectively.

Algorithm 3: $\mathsf{Shrink}(\mathcal{O}, k, I, \delta)$

Input: Oracle $\mathcal{O}$ as in (1.2); $k \in \{1, 2\}$; interval $I = [a, b]$; confidence parameter $0 < \delta < 1$.

$\epsilon \leftarrow (b - a)/5$
$\delta \leftarrow \delta/2$
Append arm $i = 0$ with bias $p_0 = 1$ to $\mathcal{O}$; call the resulting oracle $\mathcal{O}'$
Construct variable-time quantum algorithms $\mathcal{A}_1, \mathcal{A}_2$:
    $\mathcal{A}_1 \leftarrow \mathcal{A}(\mathcal{O}', l_2 = a + \epsilon, l_1 = a + 3\epsilon, 0.01\delta)$
    $\mathcal{A}_2 \leftarrow \mathcal{A}(\mathcal{O}', l_2 = a + 2\epsilon, l_1 = a + 4\epsilon, 0.01\delta)$
$r_1 \leftarrow \mathsf{Estimate}(\mathcal{A}_1, \epsilon = 0.1, \delta)$
$r_2 \leftarrow \mathsf{Estimate}(\mathcal{A}_2, \epsilon = 0.1, \delta)$
$B_1 \leftarrow \mathbb{1}(r_1 > \frac{k+0.5}{n+1})$; $B_2 \leftarrow \mathbb{1}(r_2 > \frac{k+0.5}{n+1})$
switch (B_1, B_2) **do**
    case $(0, 0) : I \leftarrow [a, a + 3\epsilon]$
    case $(0, 1) : I \leftarrow [a + \epsilon, a + 4\epsilon]$
    case $(1, 0) : I \leftarrow [a + \epsilon, a + 4\epsilon]$
    case $(1, 1) : I \leftarrow [a + 2\epsilon, a + 5\epsilon = b]$
return I

Algorithm 4: $\mathsf{BestArm}(\mathcal{O}, \delta)$

Input: Oracle $\mathcal{O}$ as in (1.2); confidence parameter $0 < \delta < 1$.

1 $\delta \leftarrow \delta/2$
2 $I_1, I_2 \leftarrow \mathsf{Locate}(\mathcal{O}, \delta)$
3 $l_1 \leftarrow \min I_1$ (left endpoint of I_1)
4 $l_2 \leftarrow \max I_2$ (right endpoint of I_2)
5 Construct variable-time quantum algorithm $\mathcal{A}$:
6     $\mathcal{A} \leftarrow \mathcal{A}(\mathcal{O}, l_2, l_1, 0.01\delta)$
7 $i \leftarrow \mathsf{Amplify}(\mathcal{A}, \delta)$
8 **return** i

2. *uses* $O\big(\sqrt{H}\ \mathrm{poly}\big(\log\big(\frac{n}{\delta\,\Delta_2}\big)\big)\big)$ *queries.*

The proofs of Lemma 4 and Lemma 5 appear in Appendix B.

The following theorem is equivalent to Theorem 4.

Theorem 5 (Correctness and complexity of Algorithm 4)**.** *Fix a confidence parameter* $0 < \delta < 1$. *Then, with probability* $\geq 1 - \delta$, *Algorithm 4*

1. *outputs the best arm, and*
2. *uses* $O\big(\sqrt{H}\ \mathrm{poly}\big(\log\big(\frac{n}{\delta\,\Delta_2}\big)\big)\big)$ *queries.*

Proof. Note that δ is halved at the beginning, on Line 1. For the first claim, we know from the first claim of Lemma 4 that, with probability $\geq 1 - \delta/2$, the two intervals I_k assigned in Line 2 have $\min I_1 - \max I_2 \geq 2\,|I_1| \geq \Delta_2/4$ and $p_k \in I_k$. Assuming this holds, we have $p_2 < l_2 < l_2 + \Delta_2/4 \leq l_1 < p_1$ for the endpoints l_k assigned in Lines 3 and 4. This means that the variable-time quantum algorithm $\mathcal{A}$ defined in Line 6 has $S_{\text{right}} \cup S_{\text{middle}} = \{1\}$, so $\mathsf{Amplify}(\mathcal{A}, \delta/2)$ returns index 1 with probability $\geq 1 - \delta/2$. Therefore, the overall probability of Algorithm 4 returning the best arm is at least $1 - \delta$.

The second claim follows immediately from adding the complexity of $\mathsf{Locate}(\mathcal{O}, \delta/2)$ (Lemma 4) and $\mathsf{Amplify}(\mathcal{A}, \delta/2)$ (Lemma 3, using $l_1 - l_2 \geq \Delta_2/4$). □

By establishing Theorem 5, we have established Theorem 4, the main claim of Section 3. As discussed previously, the main complexity measure of interest in the classical case is H, and we see that we get a quadratic speedup in terms of this parameter.

We can see that the poly-logarithmic factor has degree about 6 from (B.16), (B.18), and (B.20). It would be interesting to reduce this degree. A more fundamental challenge is to remove the variable n that appears in our log factors. In the classical case, n was already removed from log factors in early work [24] by a procedure called "median elimination". However, quantizing the median elimination framework is nontrivial, as the query complexity for outputting the $n/2$ smallest items among n elements is $\Theta(n)$ [3, Theorem 1], exceeding our budget of $O(\sqrt{n})$.

As corollaries of our main results in the fixed confidence setting, we provide results on best-arm identification in the PAC (Probably Approximately Correct) and fixed-budget settings. In the (ϵ, δ)-PAC setting, the goal is to identify an arm i with $p_i \geq p_1 - \epsilon$ with probability $\geq 1 - \delta$. We provide an algorithm with query complexity $O\big(\sqrt{\min\{\frac{n}{\epsilon^2}, H\}}\ \mathrm{poly}(\log(\frac{n}{\delta\,\Delta_2}))\big)$. In the fixed-budget setting, the goal is to identify the best arm with high probability for a fixed number T of total queries (the budget). As a direct corollary of Theorem 4, when H is known in advance, there is an algorithm that returns the best arm with probability $\geq 1 - \exp\big(-\Omega(T/\sqrt{H})\big)$ by using a reduction similar to that from Monte Carlo to Las Vegas algorithms. See Appendix C for details.

4 Quantum lower bounds

We also establish a lower bound for the quantum best-arm identification problem. Our lower bound shows that the algorithm of Theorem 4 is optimal up to poly-logarithmic factors.

Theorem 6. *Let $p \in (0, 1/2)$. For arbitrary biases $p_i \in [p, 1-p]$, any quantum algorithm that identifies the best arm requires $\Omega(\sqrt{H})$ queries to the multi-armed bandit oracle $\mathcal{O}$.*

To prove this lower bound, we use the quantum adversary method to show quantum hardness of distinguishing oracles corresponding to the following n bandits. In the 1^{st} bandit, we assign bias p_i to arm i for all i. In the x^{th} bandit for $x \in \{2, \ldots, n\}$, we assign bias $p_1 + \eta$ to arm x and p_i to arm i for all $i \neq x$, where η is an appropriately chosen parameter. This hard set of bandits is inspired by the proof of a corresponding classical lower bound [37, Theorem 5]. We defer the full proof to Appendix D. For convenience, we also introduce the quantum adversary method in Appendix A.2.

Acknowledgements

DW thanks Robin Kothari, Jin-Peng Liu, and Yuan Su for helpful discussions. This work received support from the Army Research Office (W911NF-20-1-0015); the Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Algorithms Teams and Accelerated Research in Quantum Computing programs; an IBM Ph.D. Fellowship; the National Science Foundation (CCF-1813814); and a QISE-NET Triplet Award (NSF DMR-1747426).

References

[1] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire, *Taming the monster: A fast and simple algorithm for contextual bandits*, International Conference on Machine Learning, pp. 1638–1646, 2014.

[2] Andris Ambainis, *Quantum lower bounds by quantum arguments*, Journal of Computer and System Sciences **64** (2002), no. 4, 750–767, arXiv:quant-ph/0002066

[3] Andris Ambainis, *A new quantum lower bound method, with an application to a strong direct product theorem for quantum search*, Theory of Computing **6** (2010), no. 1, 1–25, arXiv:quant-ph/0508200

[4] Andris Ambainis, *Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations*, 2010, arXiv:1010.4458

[5] Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko, *Quantum Boltzmann machine*, Physical Review X **8** (2018), no. 2, 021050, arXiv:1601.02036

[6] Srinivasan Arunachalam and Ronald de Wolf, *Guest column: a survey of quantum learning theory*, ACM SIGACT News **48** (2017), no. 2, 41–67, arXiv:1701.06806

[7] Srinivasan Arunachalam and Reevu Maity, *Quantum boosting*, To appear in the Thirty-seventh International Conference on Machine Learning, 2020, arXiv:2002.05056

[8] Frank Arute et al., *Quantum supremacy using a programmable superconducting processor*, Nature **574** (2019), no. 7779, 505–510, arXiv:1910.11333

[9] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire, *The nonstochastic multiarmed bandit problem*, SIAM Journal on Computing **32** (2002), no. 1, 48–77.

[10] Aleksandrs Belovs, *Variations on quantum adversary*, 2015, arXiv:1504.06943

[11] Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert Schapire, *Contextual bandit algorithms with supervised learning guarantees*, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 19–26, 2011.

[12] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd, *Quantum machine learning*, Nature **549** (2017), no. 7671, 195, arXiv:1611.09347

[13] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp, *Quantum amplitude amplification and estimation*, Contemporary Mathematics **305** (2002), 53–74, arXiv:quant-ph/0005055

[14] Balthazar Casalé, Giuseppe Di Molfetta, Hachem Kadri, and Liva Ralaivola, *Quantum bandits*, 2020, arXiv:2002.06395

[15] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery, *The power of block-encoded matrix powers: Improved regression techniques via faster Hamiltonian simulation*, Proceedings of the 46th International Colloquium on Automata, Languages, and Programming, Leibniz International Proceedings in Informatics (LIPIcs), vol. 132, pp. 33:1–33:14, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, arXiv:1804.01973

[16] Lijie Chen and Jian Li, *On the optimal sample complexity for best arm identification*, 2015, arXiv:1511.03774

[17] Lijie Chen, Jian Li, and Mingda Qiao, *Towards instance optimal bounds for best arm identification*, Conference on Learning Theory, pp. 535–592, 2017, arXiv:1608.06031

[18] Andrew M. Childs, Robin Kothari, and Rolando D. Somma, *Quantum algorithm for systems of linear equations with exponentially improved dependence on precision*, SIAM Journal on Computing **46** (2017), no. 6, 1920–1950, arXiv:1511.02306

[19] Vedran Dunjko and Hans J. Briegel, *Machine learning & artificial intelligence in the quantum domain: a review of recent progress*, Reports on Progress in Physics **81** (2018), no. 7, 074001, arXiv:1709.02779

[20] Vedran Dunjko, Yi-Kai Liu, Xingyao Wu, and Jacob M. Taylor, *Exponential improvements for quantum-accessible reinforcement learning*, 2017, arXiv:1710.11160

[21] Vedran Dunjko, Jacob M. Taylor, and Hans J Briegel, *Quantum-enhanced machine learning*, Physical Review Letters **117** (2016), no. 13, 130501, arXiv:1610.08251

[22] Vedran Dunjko, Jacob M. Taylor, and Hans J. Briegel, *Advances in quantum reinforcement learning*, Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics, pp. 282–287, IEEE, 2017, arXiv:1811.08676

[23] Christoph Dürr and Peter Høyer, *A quantum algorithm for finding the minimum*, 1996, arXiv:quant-ph/9607014

[24] Eyal Even-Dar, Shie Mannor, and Yishay Mansour, *PAC bounds for multi-armed bandit and markov decision processes*, Computational Learning Theory (Berlin, Heidelberg) (Jyrki Kivinen and Robert H. Sloan, eds.), pp. 255–270, Springer Berlin Heidelberg, 2002.

[25] Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric, *Best arm identification: A unified approach to fixed budget and fixed confidence*, Advances in Neural Information Processing Systems, pp. 3212–3220, 2012.

[26] Lov K. Grover, *A fast quantum mechanical algorithm for database search*, Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, pp. 212–219, ACM, 1996, arXiv:quant-ph/9605043

[27] Yassine Hamoudi, Maharshi Ray, Patrick Rebentrost, Miklos Santha, Xin Wang, and Siyi Yang, *Quantum algorithms for hedging and the sparsitron*, 2020, arXiv:2002.06003

[28] Peter Høyer and Robert Spalek, *Lower bounds on quantum query complexity*, Bulletin of the EATCS **87** (2005), 78–103, arXiv:quant-ph/0509153

[29] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck, *lilucb: An optimal exploration algorithm for multi-armed bandits*, Conference on Learning Theory, pp. 423–439, 2014, arXiv:1312.7308

[30] Sofiene Jerbi, Hendrik Poulsen Nautrup, Lea M. Trenkwalder, Hans J. Briegel, and Vedran Dunjko, *A framework for deep energy-based reinforcement learning with quantum speed-up*, 2019, arXiv:1910.12760

[31] Zohar Karnin, Tomer Koren, and Oren Somekh, *Almost optimal exploration in multi-armed bandits*, International Conference on Machine Learning, pp. 1238–1246, 2013.

[32] Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, and Anupam Prakash, *q-means: A quantum algorithm for unsupervised machine learning*, Advances in Neural Information Processing Systems, pp. 4136–4146, 2019, arXiv:1812.03584

[33] Tongyang Li, Shouvanik Chakrabarti, and Xiaodi Wu, *Sublinear quantum algorithms for training linear and kernel-based classifiers*, International Conference on Machine Learning, pp. 3815–3824, 2019, arXiv:1904.02276

[34] Lin Lin and Yu Tong, *Near-optimal ground state preparation*, 2020, arXiv:2002.12508

[35] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost, *Quantum algorithms for supervised and unsupervised machine learning*, 2013, arXiv:1307.0411

[36] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost, *Quantum principal component analysis*, Nature Physics **10** (2014), no. 9, 631, arXiv:1307.0401

[37] Shie Mannor and John N. Tsitsiklis, *The sample complexity of exploration in the multi-armed bandit problem*, Journal of Machine Learning Research **5** (2004), no. Jun, 623–648.

[38] Michael A. Nielsen and Isaac L. Chuang, *Quantum computation and quantum information*, Cambridge University Press, 2000.

[39] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd, *Quantum support vector machine for big data classification*, Physical Review Letters **113** (2014), no. 13, 130503, arXiv:1307.0471

[40] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione, *An introduction to quantum machine learning*, Contemporary Physics **56** (2015), no. 2, 172–185, arXiv:1409.3097

[41] Nathan Wiebe, Ashish Kapoor, and Krysta M. Svore, *Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning*, Quantum Information & Computation **15** (2015), no. 3-4, 316–356, arXiv:1401.2142

A Quantum computing background

A.1 Definitions and notation

Quantum computing is naturally formulated in terms of linear algebra. In the space $\mathbb{C}^n$, we call $\{\vec{e}_1, \ldots, \vec{e}_n\}$ its *computational basis*, where $\vec{e}_i = (0, \ldots, 1, \ldots, 0)^\top$ and the 1 only appears in the i^{th} coordinate. These basis vectors can be written in *Dirac notation*: $|i\rangle \coloneqq \vec{e}_i$ (called a "ket"), and $\langle i| \coloneqq \vec{e}_i^{\top}$ (called a "bra"). An n-dimensional *quantum state* is a unit vector in $\mathbb{C}^n$, i.e., $|x\rangle = (x_1, \ldots, x_n)^\top$ such that $\sum_{i=1}^n |x_i|^2 = 1$. The *tensor product* of quantum states is their Kronecker product: if $|x\rangle \in \mathbb{C}^{n_1}$ and $|y\rangle \in \mathbb{C}^{n_2}$, then

$$|x\rangle\,|y\rangle \coloneqq |x\rangle \otimes |y\rangle \coloneqq (x_1y_1, x_1y_2, \ldots, x_{n_1}y_{n_2})^\top \in \mathbb{C}^{n_1} \otimes \mathbb{C}^{n_2}. \tag{A.1}$$

Every step of a quantum algorithm is unitary, i.e., a linear transformation U such that $U^\dagger = U^{-1}$. Recall that for multi-armed bandits, the quantum oracle we adopt in (1.2) is

$$\mathcal{O}\colon\ |i\rangle_I\,|0\rangle_B \mapsto |i\rangle_I\,|\mathrm{coin}\,p_i\rangle_B\ \ \forall\, i \in [n],\ \text{ where }\ |\mathrm{coin}\,p\rangle \coloneqq \sqrt{p}\,|1\rangle + \sqrt{1-p}\,|0\rangle\,. \tag{A.2}$$

Specifically, we can access the arms in *superposition* by querying the unitary oracle $\mathcal{O}$ with a state $|x\rangle$ in the register I, which gives the output quantum state

$$\mathcal{O}\,|x\rangle = \sum_{i=1}^{n} x_i\,|i\rangle_I \left(\sqrt{p_i}\,|1\rangle_B + \sqrt{1-p_i}\,|0\rangle_B\right). \tag{A.3}$$

Directly measuring this state corresponds to pulling an arm classically following the probability distribution $|x_i|^2$. But quantumly, as we show in Section 3, there are more advanced quantum algorithms that achieve quadratic speedup in terms of H.

Our quantum speedup originates from *Grover's search algorithm* [26]. Consider a function $f_w\colon [n] \to \{-1,1\}$ such that $f_w(i) = 1$ if and only if $i \neq w$, so that w can be viewed as a (unique) marked item. To search for w, classically we need $\Omega(n)$ queries to f_w. Quantumly, given a unitary U_w such that $U_w\,|i\rangle = |i\rangle$ for all $i \neq w$ and $U_w\,|w\rangle = -\,|w\rangle$, Grover considered the uniform superposition $|u\rangle \coloneqq \frac{1}{\sqrt{n}}\sum_{i\in[n]}|i\rangle$ as well as the state $|r\rangle \coloneqq \frac{1}{\sqrt{n-1}}\sum_{i\in[n]/\{w\}}|i\rangle$. The angle between $U_w\,|u\rangle$ and $|u\rangle$ is $\theta \coloneqq \arccos(1/n) = \Theta(1/\sqrt{n})$. Note that the unitary U_w reflects about $|r\rangle$, and the unitary $U_u = 2\,|u\rangle\langle u| - I$ reflects about $|u\rangle$. If we start with $|u\rangle$, the angle between $U_w\,|u\rangle$ and $U_uU_w\,|u\rangle$ is *amplified* to 2θ, and in general the angle between $U_w\,|u\rangle$ and $(U_uU_w)^k\,|u\rangle$ is $2k\theta$. It thus suffices to take $k = \Theta(\sqrt{n})$ to find w.

This method of alternatively applying two reflections to boost the amplitude for success can be generalized to a technique called *amplitude amplification*. For the case with some unknown number $k \in [n]$ of marked items, there is also a quadratic quantum speedup for estimating $\theta \coloneqq \arccos(k/n)$ via a technique called *amplitude estimation* [13]. Amplitude amplification and estimation are the main building blocks of VTAA and VTAE, which are presented in Section 2.

In the context of searching, consider a quantum procedure $\mathcal{A}$ that returns a state $|\psi\rangle$ with t oracle queries, such that the overlap between the target state $|w\rangle$ and output state $|\psi\rangle$ is $p_{\text{succ}} \coloneqq |\langle w|\psi\rangle|^2$. By amplitude amplification and estimation [13], $O(t/\sqrt{p_{\text{succ}}})$ oracle queries suffice to either amplify the overlap to constant order (AA) or to estimate p_{succ} (AE). The purpose of VTAA and VTAE is to reduce the number of oracle queries when the intermediate states of the algorithm have considerable overlap with the target state $|w\rangle$. See [4, 15, 18] for details.

In this paper, we mainly focus on *quantum query complexity*, which is defined as the total number of oracle queries. If we have an efficient quantum algorithm for an explicit computational problem in the query complexity setting, then if we are given an explicit circuit realizing the black-box transformation, we will have an efficient quantum algorithm for the problem.

We conclude with a few references for further background. The book by Nielsen and Chuang [38] is a standard textbook on quantum computing, with a very detailed introduction to basic definitions (Section 3), Grover's algorithm and amplitude amplification (Section 6), and other related topics. Amplitude estimation is also known as quantum counting and is a standard technique in quantum algorithms [13, Section 4].

A.2 Quantum lower bounds by the adversary method

Suppose we have n multi-armed bandit oracles $\mathcal{O}_x$, $x \in [n]$, corresponding to n multi-armed bandits where the best arm is located at a different index in each. Suppose that we also have a best-arm identification algorithm $\mathcal{A}$ that uses no more than T queries to identify the best arm with probability $\geq 1-\delta$.

The basic quantum adversary method [2, 28] considers a quantity of the form

$$s_k \coloneqq \sum_{x\neq y} w_{x,y}\,\langle\psi_x^{(k)}|\psi_y^{(k)}\rangle\,, \tag{A.4}$$

where $k \in [T]$, $x, y \in [n]$, $w_{x,y} \geq 0$, and $|\psi_x^{(k)}\rangle$ is the state of $\mathcal{A}$ after the k^{th} query to the oracle $\mathcal{O}_x$.

At step $k = 0$, $\mathcal{A}$ has made no queries to the oracle, so $|\psi_x^{(0)}\rangle$ must be the same for all x. Therefore $s_0 = \sum_{x \neq y} w_{x,y}$ as $\langle \psi_x^{(0)} | \psi_y^{(0)} \rangle = 1$.

At step $k = T$, $\mathcal{A}$ must output the index of the best arm with probability $\geq 1 - \delta$. Since the location of the best arm is different for each $\mathcal{O}_x$, the states $|\psi_x^{(T)}\rangle$ must be distinguishable by a quantum measurement with probability $\geq 1 - \delta$. This means that $|\langle \psi_x^{(T)} | \psi_y^{(T)} \rangle| \leq 2\sqrt{\delta(1-\delta)}$. Therefore $|s_T| \leq 2\sqrt{\delta(1-\delta)} \cdot \sum_{x \neq y} w_{x,y}$.

Combining the above observations, we have

$$|s_0 - s_T| \geq |s_0| - |s_T| \geq (1 - 2\sqrt{\delta(1-\delta)}) \cdot \sum_{x \neq y} w_{x,y}. \tag{A.5}$$

Hence, if we can upper bound $|s_{k+1} - s_k|$ by B for some constant B, we can deduce that

$$T \geq \frac{1 - 2\sqrt{\delta(1-\delta)}}{B} \cdot \sum_{x \neq y} w_{x,y}, \tag{A.6}$$

giving a lower bound on the query complexity.

Note that we apply the quantum adversary method to multi-armed bandit oracles of the form given in (1.2), whereas most results from the literature on quantum lower bounds assume a different form of oracle. We remark that Ref. [10] treats a more general class of oracles, so it should be possible to prove Theorem 6 using its results. However, we give a self-contained proof using the formulation described above as this approach is straightforward in our case.

B Proof details of the quantum upper bound

B.1 Proof of Lemma 1

As our proof is similar to that presented in Ref. [18, Section 5.3], we only sketch it in a way that highlights the differences. For comparison, it may be helpful to note that our states $|i\rangle_I |\text{coin}\, p_i\rangle$ are analogous to the matrix eigenstates $|\lambda\rangle$ in Ref. [18]. The controlled-NOT operation in Line 9 of our Algorithm 1 takes the place of the simulation subroutine called "W" in Ref. [18, Lemma 23], which is much more elaborate.

We proceed with the proof sketch. Let $\mathcal{A}_{\text{main}} := \mathcal{A}_{m+1} \cdots \mathcal{A}_1$ denote the part of $\mathcal{A}$ after initialization. We show that, for each fixed i, $\mathcal{A}_{\text{main}} |i\rangle_I |\text{coin}\, p_i\rangle_B |0\rangle_{C,P,F}$ is $(\frac{\alpha}{n^{3/2}})$-close to

Case $i \in S_{\text{middle}}$: $|i\rangle_I |\text{coin}\, p_i\rangle_B (\beta_{i,1} |\psi_i\rangle_{C,P} |1\rangle_F + \beta_{i,0} |\psi_{i,0}\rangle_{C,P} |0\rangle_F)$ for some $\beta_{i,1}, \beta_{i,0} \in \mathbb{C}$ and states $|\psi_i\rangle, |\psi_{i,j}\rangle$;

Case $i \in S_{\text{right}}$: $|i\rangle_I |\text{coin}\, p_i\rangle_B |\psi_i\rangle_{C,P} |1\rangle_F$;

Case $i \in S_{\text{left}}$: $|i\rangle_I |\text{coin}\, p_i\rangle_B |\psi_i\rangle_{C,P} |0\rangle_F$.

Then $|\phi(\mathcal{A})\rangle = \mathcal{A} |0\rangle_{I,B,C,P,F} = \mathcal{A}_{\text{main}} \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle_I |\text{coin}\, p_i\rangle_B |0\rangle_{C,P,F}$ is $(\frac{1}{\sqrt{n}} \cdot n \cdot \frac{\alpha}{n^{3/2}} = \frac{\alpha}{n})$-close to $|\psi(\mathcal{A})\rangle$ as claimed.

Case $i \in S_{\text{middle}}$. This is trivially true because $\beta_{i,1} |\psi_{i,1}\rangle_{C,P} |1\rangle_F + \beta_{i,0} |\psi_{i,0}\rangle_{C,P} |0\rangle_F$ can represent any state on registers C, P, F.

Case $i \in S_{\text{left}}$. Let $j \in [m-1]$ be such that $l_1 - 2\epsilon_j \le p_i < l_1 - \epsilon_j$. Note that this j uniquely exists by the definition of S_{left}, m, and ϵ_j. Then the state of the algorithm after the $(j-1)$st iteration of the for-loop in Line 5 is $(2(j-1)a)$-close to

$$|i\rangle_I \, |\text{coin}\, p_i\rangle_B \, |0\rangle_C \, |\gamma_0^1\rangle_{P_1} \cdots |\gamma_0^{j-1}\rangle_{P_{j-1}} \, |0\rangle_{P_j \cdots P_m} \, |1\rangle_F \,, \tag{B.1}$$

where, for each i, the state $|0\rangle_{C_i} |\gamma_0\rangle_{P_i}$ corresponds to the state $|0\rangle_C |\gamma_0\rangle$ in $\mathsf{GAE}(\epsilon_j, a; l_1)$. Note that we incur an error of at most $2a$ at each iteration which comes from running $\mathsf{GAE}(\epsilon_j, a; l_1)$ (cf. the case where $\beta_1 \le a$ in Corollary 1). This error accumulates additively.

The state after the j^{th} iteration is $(2ja)$-close to

$$\begin{aligned}
&\beta_0 \, |i\rangle_I \, |\text{coin}\, p_i\rangle_B \, |0\rangle_C \, |\gamma_0^1\rangle_{P_1} \cdots |\gamma_0^{j}\rangle_{P_j} \, |0\rangle_{P_{j+1} \cdots P_m} \, |1\rangle_F \\
&+ \beta_1 \, |i\rangle_I \, |\text{coin}\, p_i\rangle_B \, |\boldsymbol{j}\rangle_C \, |\gamma_0^1\rangle_{P_1} \cdots |\gamma_0^{j}\rangle_{P_j} \, |0\rangle_{P_{j+1} \cdots P_m} \, |1\rangle_F \,,
\end{aligned} \tag{B.2}$$

where $\boldsymbol{j} := 0^{j-1}10^{m-j}$ denotes a unary representation of the integer j.

At the $(j+1)$st iteration, the part of the state in the second line of Eq. (B.2) is unchanged because its register C indicates "stop", but the part in the first line of Eq. (B.2) changes to being $(2(j+1)a)$-close to

$$\beta_0 \, |i\rangle_I \, |\text{coin}\, p_i\rangle_B \, |\boldsymbol{j+1}\rangle_C \, |\gamma_0^1\rangle_{P_1} \cdots |\gamma_0^{j}\rangle_{P_j} \, |\gamma_0^{j+1}\rangle_{P_{j+1}} \, |0\rangle_{P_{j+2} \cdots P_m} \, |0\rangle_F \,. \tag{B.3}$$

Hence, the state after the $(j+1)$st iteration is $(2(j+1)a)$-close to

$$\begin{aligned}
&\beta_0 \, |i\rangle_I \, |\text{coin}\, p_i\rangle_B \, |\boldsymbol{j+1}\rangle_C \, |\gamma_0^1\rangle_{P_1} \cdots |\gamma_0^{j}\rangle_{P_j} \, |\gamma_0^{j+1}\rangle_{P_{j+1}} \, |0\rangle_{P_{j+2} \cdots P_m} \, |0\rangle_F \\
&+ \beta_1 \, |i\rangle_I \, |\text{coin}\, p_i\rangle_B \, |\boldsymbol{j}\rangle_C \, |\gamma_0^1\rangle_{P_1} \cdots |\gamma_0^{j}\rangle_{P_j} \, |0\rangle_{P_{j+1} \cdots P_m} \, |0\rangle_F \,.
\end{aligned} \tag{B.4}$$

Since the C register of all parts of the state in Eq. (B.4) indicates "stop", the remaining iterations $j+2, \ldots, m$ of $\mathcal{A}$ do not alter it. Hence the final state of $\mathcal{A}$ is $(2ma)$-close to the state in Eq. (B.4), which is of the form

$$|i\rangle_I \, |\text{coin}\, p_i\rangle_B \, |\psi_i\rangle_{C,P} \, |0\rangle_F \,. \tag{B.5}$$

Note that $2ma = \frac{\alpha}{n^{3/2}}$, so the closeness of approximation is as claimed.

Case $i \in S_{\text{right}}$. In this case, there does not exist a $j \in [m-1]$ such that $l_1 - 2\epsilon_j \le p_i < l_1 - \epsilon_j$. Thus a simplified version of the argument above, in which we do not have to consider different cases according to the iteration number, shows that the resulting state is $(2ma)$-close to a state of the same form as Eq. (B.5) but with the F register remaining in state 1.

Lastly, we show that p_{succ} is close to p'_{succ} as claimed:

$$\begin{aligned}
\left|p_{\text{succ}} - p'_{\text{succ}}\right| &= \left|\left(\sqrt{p_{\text{succ}}} + \sqrt{p'_{\text{succ}}}\right) \cdot \left(\sqrt{p_{\text{succ}}} - \sqrt{p'_{\text{succ}}}\right)\right| \\
&= \left(\sqrt{p_{\text{succ}}} + \sqrt{p'_{\text{succ}}}\right) \cdot \left|\left\|\Pi_F \, |\phi(\mathcal{A})\rangle\right\| - \left\|\Pi_F \, |\psi(\mathcal{A})\rangle\right\|\right| \\
&\le 2 \left\|\Pi_F (|\phi(\mathcal{A})\rangle - |\psi(\mathcal{A})\rangle)\right\| \\
&\le 2\, \frac{\alpha}{n}.
\end{aligned} \tag{B.6}$$

B.2 Proof of Lemma 2

The proof is similar to that presented in Ref. [18, Section 5.4]. For the first claim, note first that $\mathcal{A}_0$ and $\mathcal{A}_{m+1}$ use a constant number of queries (1 and 0, respectively), so we can ignore them. For $k \in [m]$, $\mathcal{A}_k$ only uses queries to perform $\mathsf{GAE}(\epsilon_k, d; l_1)$, which takes $O(\frac{1}{\epsilon_k} \log \frac{1}{a})$ queries. Therefore t_j, the number of queries in $\mathcal{A}_j \mathcal{A}_{j-1} \cdots \mathcal{A}_1$, is of order

$$\sum_{k=1}^{j} \frac{1}{\epsilon_k} \log\Big(\frac{1}{a}\Big) = \sum_{k=1}^{j} 2^k \log\Big(\frac{1}{a}\Big) \leq 2^{j+1} \log\Big(\frac{1}{a}\Big) \tag{B.7}$$

because $\epsilon_k = 2^{-k}$ by definition. In addition, we have $t_m = O(\frac{1}{\Delta} \log \frac{1}{a})$ because $m = \lceil \log \frac{1}{\Delta} \rceil + 2$ by definition. The first claim follows.

For the second claim, we have

$$t_{\text{avg}}^2 = \sum_{j=1}^{m} w_j t_j^2 = \sum_{j=1}^{m} \left\| \Pi_{C_j} \mathcal{A}_j \cdots \mathcal{A}_1 \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle_I \, |\text{coin}\, p_i\rangle_B \, |0\rangle_C \, |0\rangle_P \, |1\rangle_F \right\|^2 t_j^2 \tag{B.8}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{i,j} t_j^2 \tag{B.9}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \tau_i^2, \tag{B.10}$$

where $w_{i,j} \coloneqq \| \Pi_{C_j} \mathcal{A}_j \cdots \mathcal{A}_1 \, |i\rangle_I \, |\text{coin}\, p_i\rangle_B \, |0\rangle_C \, |0\rangle_P \, |1\rangle_F \, \|^2 \in [0,1]$ and $\tau_i \coloneqq \sum_{j=1}^{m} w_{i,j} t_j^2$.

Note that $w_{i,j}$ can be thought of as the probability that $\mathcal{A}$ stops at the end of iteration j if initialized with arm i; τ_i^2 can be thought of as the squared average stopping time of $\mathcal{A}$ if initialized with arm i.

For each fixed i, we consider τ_i^2 according to the following three cases.

Case $i \in S_{\text{right}}$. We have $\sum_{j=1}^{m} w_{i,j} = 1$, so $\tau_i^2 \leq t_m^2 = O(2^{2m} \log^2(\frac{1}{a})) = O(\frac{1}{\Delta^2} \log^2(\frac{1}{a}))$ because $m = \lceil \log \frac{1}{\Delta} \rceil + 2$ by definition.

Case $i \in S_{\text{middle}}$. We still have $\tau_i^2 = O(\frac{1}{\Delta^2} \log^2(\frac{1}{a}))$ as in the case $i \in S_{\text{right}}$, by exactly the same argument. But by the definition of S_{middle}, we have $l_1 - p_i \leq \Delta/2$, so we can also write $\tau_i^2 = O\big(\frac{1}{(l_1 - p_i)^2} \log^2(\frac{1}{a})\big)$.

Case $i \in S_{\text{left}}$. For $i \in S_{\text{left}}$, let $j \in [m-1]$ be such that $l_1 - 2\epsilon_j \leq p_i < l_1 - \epsilon_j$ as in the proof of Lemma 1.

We know that after the $(j+1)$st iteration, the state is $(ma = \alpha/n)$-close to the state in (B.4) on which the algorithm terminates. Therefore, the probability $w_{i,j+1}$ of terminating after the $(j+1)^{\text{st}}$ iteration is $1 - O((\alpha/n)^2)$. It can also be seen that the probability $w_{i,j+r}$ of terminating after the $(j+r)^{\text{th}}$ iteration is $(1 - O((\alpha/n)^2)) \cdot O((\alpha/n)^{2(r-1)})$. Hence

$$\tau_i^2 \leq t_{j+1}^2 + O\bigg(\sum_{r=2}^{m-j} \Big(\frac{\alpha}{n}\Big)^{2(r-1)} t_{j+r}^2\bigg) = O(t_{j+1}^2) = O\bigg(\frac{\log^2(\frac{1}{a})}{\epsilon_{j+1}}\bigg) = O\bigg(\frac{\log^2(\frac{1}{a})}{(l_1 - p_i)^2}\bigg), \tag{B.11}$$

where we used $\epsilon_{j+1} = \epsilon_j/2 \geq (l_1 - p_i)/4$ for the last inequality.

Substituting the above results into (B.10) tells us that t_{avg}^2 is of order

$$\frac{1}{n}\Big(\frac{|S_{\text{right}}|}{\Delta^2} + \sum_{i\in S_{\text{left}}\cup S_{\text{middle}}} \frac{1}{(l_1-p_i)^2}\Big)\cdot \log^2\Big(\frac{1}{a}\Big) \tag{B.12}$$

as desired.

B.3 Proof of Lemma 3

We set the approximation parameter in $\mathcal{A}$ to be $\alpha = c\delta$ for some constant $c < 0.05$ to be determined later. Then $\alpha < 0.05$.

We apply VTAA (Theorem 3) on $\mathcal{A}$. This gives an algorithm that outputs a state $|\psi_{\text{succ}}\rangle$ that is $(\frac{\alpha}{n} = \frac{c\delta}{n})$-close to the (normalized) state proportional to

$$\Pi_F\,|\psi(\mathcal{A})\rangle = \frac{1}{\sqrt{n}}\Big(\sum_{i\in S_{\text{right}}} |i\rangle_I\,|\text{coin}\,p_i\rangle_B\,|\psi_i\rangle_{C,P}\,|1\rangle_F + \sum_{i\in S_{\text{middle}}} \alpha_{i,1}\,|i\rangle_I\,|\text{coin}\,p_i\rangle_B\,|\psi_{i,1}\rangle_{C,P}\,|1\rangle_F\Big) \tag{B.13}$$

with success probability at least 1/2 and a bit indicating success or failure. Now, we repeat the entire procedure $O(\log\frac{1}{\delta})$ times to prepare $|\psi_{\text{succ}}\rangle$ at least once with probability $\geq 1-\delta/2$. Once $|\psi_{\text{succ}}\rangle$ has been successfully prepared, as indicated by the algorithm, we measure its index register I. This procedure outputs an arm index in $S_{\text{right}}\cup S_{\text{middle}}$ with probability $\geq (1-\delta/2)\cdot(1-2c\delta/n)$ which is $\geq 1-\delta$ for $c\leq 1/4$ sufficiently small. So, as we also need $c<0.05$, we choose $c = 0.01$. We call this procedure $\mathsf{Amplify}(\mathcal{A},\delta)$.

Let us consider the query complexity of $\mathsf{Amplify}(\mathcal{A},\delta)$. We have

$$t_{m+1} = O\Big(\frac{1}{\Delta}\log\Big(\frac{1}{a}\Big)\Big) = O\Big(\frac{1}{\Delta}\log\Big(n\log\Big(\frac{1}{\Delta}\Big)\Big)\Big) \tag{B.14}$$

because $a = \frac{\alpha}{2(\lceil\log(1/\Delta)\rceil+2)n^{3/2}}$ by definition. We also have

$$p_{\text{succ}} \geq p'_{\text{succ}} - \frac{2\alpha}{n} \geq \frac{|S_{\text{right}}|}{n} - \frac{0.1}{n} > \frac{|S_{\text{right}}|}{2n}, \tag{B.15}$$

where we used the assumption $|S_{\text{right}}| > 0$ for the last inequality. Lastly, t_{avg}^2 is of order given in (3.4) (reproduced in (B.12) above). Therefore, substituting all these bounds into (2.5) of Theorem 3, we see that $\mathsf{Amplify}(\mathcal{A},\delta)$ has query complexity of order

$$\Big(\frac{1}{\Delta^2} + \frac{1}{|S_{\text{right}}|}\sum_{S_{\text{left}}\cup S_{\text{middle}}}\frac{1}{(l_1-p_i)^2}\Big)\cdot\log\Big(\frac{n}{\delta}\log\frac{1}{\Delta}\Big)\cdot\log\Big(\frac{1}{\Delta}\log\Big(\frac{n}{\delta}\log\Big(\frac{1}{\Delta}\Big)\Big)\Big)\cdot\log\Big(\frac{1}{\delta}\Big). \tag{B.16}$$

We also apply VTAE (Theorem 3) with multiplicative accuracy ϵ and confidence δ on $\mathcal{A}$. This gives an algorithm, $\mathsf{Estimate}(\mathcal{A},\epsilon,\delta)$, that outputs an estimate r of p_{succ} with multiplicative accuracy ϵ (i.e., $|r-p_{\text{succ}}| < \epsilon p_{\text{succ}}$) with probability $\geq 1-\delta$. Combining $|r-p_{\text{succ}}| < \epsilon p_{\text{succ}}$ with $|p_{\text{succ}} - p'_{\text{succ}}| \leq \frac{2\alpha}{n} < \frac{0.1}{n}$ gives

$$(1-\epsilon)\Big(p'_{\text{succ}} - \frac{0.1}{n}\Big) < r < (1+\epsilon)\Big(p'_{\text{succ}} + \frac{0.1}{n}\Big) \tag{B.17}$$

as claimed.

The query complexity of $\mathsf{Estimate}(\mathcal{A},\epsilon,\delta)$ is

$$(B.16)\cdot\frac{1}{\epsilon}\log^2(t_{m+1})\log\Big(\log\Big(\frac{t_{m+1}}{\delta}\Big)\Big) = (B.16)\cdot O\Big(\frac{1}{\epsilon}\,\text{poly}\Big(\log\Big(\frac{n}{\delta\Delta}\Big)\Big)\Big) \tag{B.18}$$

according to Theorem 3 and (B.14).

B.4 Proof of Lemma 4

From the first claim of Lemma 5, we see that the probability of E^c is at most $\frac{\delta}{4}\sum_{i=0}^{\infty} 2^{-i} = \delta/2$, where the geometric series arises because of Line 6. Henceforth, we assume E.

Consider the first claim. For given intervals I_2, I_1, let us write

$$\operatorname{gap}(I_2, I_1) \coloneqq \min I_1 - \max I_2. \tag{B.19}$$

At the end of iteration $i \geq 1$ (i.e., after Line 6), we have $|I_k| = (3/5)^i$ by the first claim of Lemma 5. At the end of iteration $\lceil \log_{5/3}(\frac{1}{\Delta_2}) \rceil + 3$, we have $|I_k| < \Delta_2/4$, so $\operatorname{gap}(I_2, I_1) > \Delta_2 - 2\Delta_2/4 = \Delta_2/2 > 2\,|I_1|$ because $p_k \in I_k$. Therefore the while loop must break at this point if it has not done so earlier. For the returned I_k, we clearly have $p_k \in I_k$ because E holds, and $\operatorname{gap}(I_2, I_1) > 2\,|I_1|$ because the while loop has broken. During the while loop, because $|I_k|$ decreases from iteration to iteration, we always have $|I_k| \geq (3/5)^{\lceil \log_{5/3}(\Delta_2^{-1}) \rceil + 3} \geq \Delta_2/8$. Note that $|I_1| = |I_2|$ because, at each iteration of the while loop, the Shrink subroutine always shrinks intervals by the same factor of $3/5$ and $|I_1| = |I_2| = 1$ initially.

Now, consider the second claim. From the first claim, we know that the while loop breaks at or before the end of iteration $\lceil \log_{5/3}(\Delta_2^{-1})) \rceil + 3$, and we always have $1/\delta_i = O(2^{\log_{5/3}(\Delta_2^{-1})}/\delta) = O(\Delta_2^{-2}/\delta)$, where $\delta_i = \delta/2^{2+i}$ is the confidence parameter in Shrink at iteration i. Therefore, using the second claim of Lemma 5, the total number of queries used is at most

$$O(\log(\Delta_2^{-1})) \cdot O\Big(\sqrt{H} \cdot \operatorname{poly}\Big(\log\Big(\frac{n}{\Delta_2} \cdot \frac{\Delta_2^{-2}}{\delta}\Big)\Big)\Big), \tag{B.20}$$

which is $O\big(\sqrt{H} \cdot \operatorname{poly}(\log(\frac{n}{\delta\,\Delta_2}))\big)$ as desired.

B.5 Proof of Lemma 5

Throughout, we fix $k \in \{0, 1\}$.

For the first claim, it is clear that $|J| = 3\,|I|\,/5$ because all the intervals appearing in Lines 11–14 have length 3ϵ. Our proof that $p_k \in J$ with high probability is similar to that in Ref. [34, Section 4] so we only present a brief sketch below.

Let us write $x_j = a + j\epsilon$ for $j = 0, \ldots, 5$, so that $x_0 = a$ and $x_5 = b$. Let E be the event that both $\mathsf{Estimate}$s in Lines 7 and 8 return the correct result. The probability of E^c is at most δ so we restrict to the case of E in the following paragraph.

For $j \in \{1, 2\}$, we can use (3.5) in Lemma 3 to see that if $p_k \leq x_j$, then $B_j = 0$ because $r_j \leq (1 + 0.1)(\frac{k}{n+1} + \frac{0.1}{n+1}) < \frac{k+0.5}{n+1}$, whereas if $p_k \geq x_{j+2}$, then $B_j = 1$ because $r_j \geq (1 - 0.1)(\frac{k+1}{n+1} - \frac{0.1}{n+1}) > \frac{k+0.5}{n+1}$. Here we use the fact $k \in \{1, 2\}$. By considering the contrapositive of the previous two if-then statements, we establish the first claim.

For more details, we refer the reader to Ref. [34, Section 4], in particular its Table 2 and Algorithm 1. Note that in the case of $(B_1, B_2) = (0, 1)$, we could have shrunk the interval to $[a + 2\epsilon, a + 3\epsilon]$ and still maintained $p_k \in J$, as is done in Ref. [34]. However, it is important for us to keep the shrinkage factor $(3/5)$ the same in all cases because we use this to prove correctness in Lemma 4.

We now prove the second claim. Since we run $\mathsf{Estimate}$ with constant multiplicative error $\epsilon = 0.1$, its query complexity is of order (B.16), which is

$$\frac{1}{\Delta^2} + \frac{1}{|S_{\text{right}}|} \sum_{i \in S_{\text{left}} \cup S_{\text{middle}}} \frac{1}{(l_1 - p_i)^2} \tag{B.21}$$

up to polylog factors, where we recall that $\Delta = l_1 - l_2$. In addition, we recall

$$S_{\text{left}} \cup S_{\text{middle}} = \{i : p_i < l_1 - \Delta/8\} \tag{B.22}$$

from (3.2) and (3.3). Note that $|S_{\text{right}}| > 0$ because we appended an arm with bias $p_0 = 1$.

By assumption, $|I| \geq \Delta_2/8$. So, in view of Lines 5 and 6, we have $\Delta = 2\epsilon = 2|I|/5 \geq \Delta_2/20$. Therefore $1/\Delta^2 = O(1/\Delta_2^2)$.

We also need to compare $p_1 - p_i$ with $l_1 - p_i$ for $i \in S_{\text{left}} \cup S_{\text{middle}}$. By definition, we have $p_i < l_1 - \Delta/8$, so $l_1 - p_i > \Delta/8$. Note that we also have $|p_k - l_1| \leq |I| = 5\Delta/2$ because $p_k \in I$ by assumption and $l_1 \in I$ by definition. If $k = 1$, this says $|p_1 - l_1| \leq 5\Delta/2$. If $k = 2$, this says $|p_2 - l_1| \leq 5\Delta/2$, but we can still bound

$$|p_1 - l_1| \leq \Delta_2 + |p_2 - l_1| \leq 20\Delta + 5\Delta/2 < 25\Delta. \tag{B.23}$$

So regardless of whether $k = 1$ or $k = 2$, we have that $|p_1 - l_1| < 25\Delta$. Therefore

$$\frac{p_1 - p_i}{l_1 - p_i} = 1 + \frac{p_1 - l_1}{l_1 - p_i} < 1 + \frac{25\Delta}{\Delta/8} = 201, \tag{B.24}$$

and so $1/(l_1 - p_i)^2 = O(1/(p_1 - p_i)^2)$. Hence we have established the second claim.

C Corollaries for the PAC and fixed-budget settings

C.1 PAC setting

Another setting often considered in the classical literature is the (ϵ, δ)-PAC setting. The goal here is to identify an arm i with $p_i \geq p_1 - \epsilon$ with probability $\geq 1 - \delta$. Our best-arm identification algorithm can be modified to work in this setting as well. More precisely, we can modify Locate (Algorithm 2) by adding a breaking condition to the while loop when $|I_1|$ (or equivalently $|I_2|$) is smaller than ϵ. The resulting algorithm finds an ϵ-optimal arm with query complexity $O\big(\sqrt{\min\{\frac{n}{\epsilon^2}, H\}} \cdot \operatorname{poly}\big(\log\big(\frac{n}{\delta \Delta_2}\big)\big)\big)$. Note that our modification means that the Amplify step in Algorithm 4 takes an input interval I with $|I| = l_1 - l_2 \in [\epsilon/2, \epsilon]$. The correctness and complexity follow directly from Lemma 1 and Lemma 3. For comparison, Ref. [24] gave a classical PAC algorithm with complexity $O\big(\frac{n}{\epsilon^2}\log\big(\frac{n}{\delta}\big)\big)$, which was later improved to $O\big(\sum_{i=1}^{n} \min\{\epsilon^{-2}, \Delta_i^{-2}\} \cdot \log\big(\frac{n}{\delta \Delta_2}\big)\big)$ by Ref. [25].

C.2 Fixed-budget setting

As mentioned near the end of Section 3, by using a reduction similar to that from Monte Carlo to Las Vegas algorithms, we can construct a fixed-budget algorithm from our fixed-confidence one. For completeness, we state and prove the following result:

Lemma 6 (Reduction to fixed confidence)**.** *Let $\mathcal{O}$ be a multi-armed bandit oracle. Suppose that for any $\delta \in (0, 1)$, we have an algorithm $\mathcal{A}_c(\delta)$ that with probability $\geq 1 - \delta$, terminates before using $T_c(\delta)$ queries to $\mathcal{O}$ and returns the best-arm index $i^* = 1$. Suppose that we also know $T_c(\delta)$. Then, for any positive integer T, we can construct an algorithm $\mathcal{A}_b(T)$ that returns $i^* = 1$ with probability $\geq \min_{\delta \in (0,1)} \exp\big(-\lfloor T/T_c(\delta) \rfloor D(\frac{1}{2}\|\delta)\big)$ using at most T queries to $\mathcal{O}$, where $D(p\|q)$ is the relative entropy between Bernoulli random variables with bias p and q.*

Proof. Since $T_c(\delta)$ is known, consider the modified version of the fixed-confidence algorithm where the algorithm is forced to halt and return some blank symbol "$\perp$" if the running time exceeds $T_c(\delta)$. We refer to the modified algorithm as $A'_c(\delta)$. $A'_c(\delta)$ returns the best-arm index $i^* = 1$ with probability $\geq 1-\delta$ and returns some symbol in $\{2, \dots, n, \perp\}$ with probability $\leq \delta$.

For any T, we construct $\mathcal{A}_b(T)$ as follows. Pick some $\delta \in (0,1)$, run $A'_c(\delta)$ $m := \lfloor T/T_c(\delta) \rfloor$ times, and take a majority vote over the outcomes. The failure probability can be upper bounded by the probability that i^* is observed fewer than $m/2$ times. The Chernoff bound upper bounds the latter probability by $\exp(-mD(\frac{1}{2}\|\delta)) = \exp\big(-\lfloor T/T_c(\delta) \rfloor D(\frac{1}{2}\|\delta)\big)$. But δ was arbitrary, so we can take the δ that minimizes this upper bound. □

As a direct corollary of Theorem 4 and Lemma 6, we see that when H (therefore T_c) is known in advance, for sufficiently large T, there is a quantum algorithm using at most T queries that returns the best-arm with probability $\geq 1 - \exp(-\Omega(T/\sqrt{H}))$.

D Proof details of the quantum lower bound

D.1 Proof of Theorem 6

For convenience, we reproduce the statement of the result:

Theorem 6. *Let $p \in (0, 1/2)$. For arbitrary biases $p_i \in [p, 1-p]$, any quantum algorithm that identifies the best arm requires $\Omega(\sqrt{H})$ queries to the multi-armed bandit oracle $\mathcal{O}$.*

Proof. We use the adversary method (see Appendix A) and consider the following n different multi-armed bandit oracles.

In the 1st bandit, we assign bias p_i to arm i. Let $\eta > 0$ be a constant to be determined later. In the x^{th} bandit, $x \in \{2, \dots, n\}$, we assign bias $p'_1 := p_1 + \eta$ to arm x and p_i to arm i for all $i \neq x$. A best-arm identification algorithm must output arm x on assignment x for all $x \in [n]$ with probability $\geq 1-\delta$.

Following the adversary method, we consider the sum

$$s_k := \sum_{x>1} \frac{1}{\Delta'^2_x} \langle \psi_x^{(k)} | \psi_1^{(k)} \rangle \tag{D.1}$$

for $x \in [n]$, where $\Delta'_x := p'_1 - p_x$. Clearly

$$s_0 = \sum_{x>1} \frac{1}{\Delta'^2_x}. \tag{D.2}$$

We also have

$$s_T \leq \sum_{x>1} \frac{1}{\Delta'^2_x} \cdot 2\sqrt{\delta(1-\delta)}. \tag{D.3}$$

Next, we bound the difference $|s_{k+1} - s_k|$. For $i > 1$, we let

$$A_i := \begin{pmatrix} \sqrt{1-p_i} & \sqrt{p_i} \\ \sqrt{p_i} & -\sqrt{1-p_i} \end{pmatrix}, \tag{D.4}$$

while

$$A_1 := \begin{pmatrix} \sqrt{1-p'_1} & \sqrt{p'_1} \\ \sqrt{p'_1} & -\sqrt{1-p'_1} \end{pmatrix}, \tag{D.5}$$

where we recall $p_1' = p_1 + \eta$ by definition.

Now, let us write

$$|\psi_x^{(k)}\rangle = \sum_{z,i,b} \alpha_{x,z,i,b} |z,i,b\rangle \,, \quad |\psi_1^{(k)}\rangle = \sum_{z,i,b} \alpha_{1,z,i,b} |z,i,b\rangle \,. \tag{D.6}$$

Then

$$|\psi_x^{(k+1)}\rangle = \mathcal{O}_x |\psi_x^{(k)}\rangle = \sum_{z,b} \alpha_{x,z,x,b} |z,x\rangle A_1 |b\rangle + \sum_{i\neq x}\sum_{z,b} \alpha_{x,z,i,b} |z,i\rangle A_i |b\rangle \tag{D.7}$$

and similarly

$$|\psi_1^{(k+1)}\rangle = \mathcal{O}_1 |\psi_1^{(k)}\rangle = \sum_{z,b} \alpha_{1,z,x,b} |z,x\rangle A_x |b\rangle + \sum_{i\neq x}\sum_{z,b} \alpha_{1,z,i,b} |z,i\rangle A_i |b\rangle \,. \tag{D.8}$$

Then

$$|s_{k+1} - s_k| \leq \sum_{x>1} \frac{1}{\Delta_x'^2} \left| \langle\psi_x^{(k)}| \mathcal{O}_x^\dagger \mathcal{O}_1 |\psi_1^{(k)}\rangle - \langle\psi_x^{(k)}|\psi_1^{(k)}\rangle \right| . \tag{D.9}$$

Using (D.7) and (D.8), and after cancellations, we find that

$$\langle\psi_x^{(k)}| \mathcal{O}_x^\dagger \mathcal{O}_1 |\psi_1^{(k)}\rangle - \langle\psi_x^{(k)}|\psi_1^{(k)}\rangle = \sum_{z,b,b'} \alpha^*_{x,z,x,b}\alpha_{1,z,x,b'} \langle b| (A_1^\dagger A_x - \mathbb{I}) |b'\rangle \,. \tag{D.10}$$

With

$$\begin{aligned} \begin{pmatrix} u_x & v_x \\ -v_x & u_x \end{pmatrix} &:= A_1^\dagger A_x - \mathbb{I} \\ &= \begin{pmatrix} \sqrt{(1-p_1')(1-p_x)} + \sqrt{p_1' p_x} - 1 & \sqrt{(1-p_1')p_x} - \sqrt{p_1'(1-p_x)} \\ -\sqrt{(1-p_1')p_x} + \sqrt{p_1'(1-p_x)} & \sqrt{(1-p_1')(1-p_x)} + \sqrt{p_1' p_x} - 1 \end{pmatrix}, \end{aligned} \tag{D.11}$$

we have

$$|s_{k+1} - s_k| \leq \sum_{x>1}\sum_{z,b} \frac{|u_x|}{\Delta_x'^2} |\alpha_{x,z,x,b}| \, |\alpha_{1,z,x,b}| + \sum_{x>1}\sum_{z,b\neq b'} \frac{|v_x|}{\Delta_x'^2} |\alpha_{x,z,x,b}| \, |\alpha_{1,z,x,b'}| \,. \tag{D.12}$$

Clearly, $|u_x| = 1 - \sqrt{(1-p_1')(1-p_x)} - \sqrt{p_1' p_x} \leq 1 - (1-p_1') - p_x = p_1' - p_x = \Delta_x'$. It can also be seen that $|v_x| \leq \Delta_x'/c(p-\eta)$, where $c(x) := 2\sqrt{x(1-x)}$ is a monotone increasing function when $x \in [0,1/2]$. For completeness, we prove the latter inequality as an auxiliary Lemma 7 immediately after this proof.

We can establish the following bounds using Cauchy-Schwarz:

$$\begin{aligned} \sum_{x>1}\sum_{z,b} \frac{|u_x|}{\Delta_x'^2} |\alpha_{x,z,x,b}| \, |\alpha_{1,z,x,b}| &\leq \sqrt{\sum_{x>1,z,b} \frac{|u_x|^2}{\Delta_x'^4} |\alpha_{x,z,x,b}|^2} \cdot \sqrt{\sum_{x>1,z,b} |\alpha_{1,z,x,b}|^2} \\ &\leq \sqrt{\sum_{x>1} \frac{1}{\Delta_x'^2}} \end{aligned} \tag{D.13}$$

and

$$\sum_{x>1}\sum_{z,b\neq b'}\frac{|v_x|}{\Delta_x'^2}|\alpha_{x,z,x,b}|\,|\alpha_{1,z,x,b'}| = \sum_{b\neq b'}\sum_{x>1,z}\frac{|v_x|}{\Delta_x'^2}|\alpha_{x,z,x,b}|\,|\alpha_{1,z,x,b'}|$$

$$\leq \sum_{b\neq b'}\sqrt{\sum_{x>1,z}\frac{|v_x|^2}{\Delta_x'^4}|\alpha_{x,z,x,b}|^2}\cdot\sqrt{\sum_{x>1,z}|\alpha_{1,z,x,b'}|^2} \tag{D.14}$$

$$\leq \frac{2}{c(p-\eta)}\sqrt{\sum_{x>1}\frac{1}{\Delta_x'^2}}.$$

Therefore, we find that

$$|s_{k+1}-s_k| \leq \left(1+\frac{2}{c(p-\eta)}\right)\sqrt{\sum_{x>1}\frac{1}{\Delta_x'^2}}. \tag{D.15}$$

Hence, from Eqs. (D.2), (D.3), and (D.15), we find that

$$T \geq \frac{1-2\sqrt{\delta(1-\delta)}}{1+2/c(p-\eta)}\sqrt{\sum_{x>1}\frac{1}{\Delta_x'^2}}. \tag{D.16}$$

We then set $\eta = p(p_1-p_2)/2$. Now, it can be seen that

$$c(p-\eta) = c\Big(\Big(1-\frac{p_1-p_2}{2}\Big)p\Big) \geq c(p/2) \tag{D.17}$$

because $p \leq 1/2$ and $p_1 - p_2 \leq 1$. Moreover, for $x > 1$,

$$\Delta_x' = p_1+\eta-p_x = \frac{p}{2}(p_1-p_2)+(p_1-p_x) \leq \Big(1+\frac{p}{2}\Big)(p_1-p_x) \leq \frac{5}{4}\Delta_x \tag{D.18}$$

because $p_x \leq p_2$ and $p \leq 1/2$. Therefore, we find that

$$T \geq \frac{4}{5}\cdot\frac{1-2\sqrt{\delta(1-\delta)}}{1+2/c(p/2)}\sqrt{\sum_{x>1}\frac{1}{\Delta_x^2}}, \tag{D.19}$$

and hence $T = \Omega\big(\sqrt{\sum_{i=2}^{n}\frac{1}{\Delta_i^2}}\big)$. □

Lemma 7. *Suppose that $p_1, p_2 \in [p, 1-p]$ where $0 < p \leq 1/2$. Then*

$$|\sqrt{(1-p_1)p_2} - \sqrt{(1-p_2)p_1}| \leq \frac{|p_1-p_2|}{2\sqrt{p(1-p)}}, \tag{D.20}$$

and the term in the denominator is optimal.

Proof. Note that

$$\sqrt{(1-p_1)p_2} - \sqrt{(1-p_2)p_1} = \frac{(1-p_1)p_2-(1-p_2)p_1}{\sqrt{(1-p_1)p_2}+\sqrt{(1-p_2)p_1}} \tag{D.21}$$

$$= \frac{-(p_1-p_2)}{\sqrt{(1-p_1)p_2}+\sqrt{(1-p_2)p_1}}. \tag{D.22}$$

Therefore, it suffices to prove

$$\sqrt{(1-p_1)p_2} + \sqrt{(1-p_2)p_1} \geq 2\sqrt{p(1-p)}. \tag{D.23}$$

Since $p_1, p_2 \in [p, 1-p]$, we have

$$(p_1 - p)(p_1 - (1-p)) \leq 0 \tag{D.24}$$

$$(p_2 - p)(p_2 - (1-p)) \leq 0 \tag{D.25}$$

$$|2p_1 - 1| \leq 1 - 2p \tag{D.26}$$

$$|2p_2 - 1| \leq 1 - 2p. \tag{D.27}$$

Eqs. (D.24) and (D.25) are equivalent to

$$p_1 - p_1^2 \geq p(1-p), \quad p_2 - p_2^2 \geq p(1-p). \tag{D.28}$$

Eqs. (D.26) and (D.27) imply

$$4p_1p_2 - 2p_1 - 2p_2 + 1 = (2p_1 - 1)(2p_2 - 1) \leq (2p-1)^2 = 4p^2 - 4p + 1, \tag{D.29}$$

which gives

$$p_1 + p_2 - 2p_1p_2 \geq 2p - 2p^2. \tag{D.30}$$

Now, we have

$$\left(\sqrt{(1-p_1)p_2} + \sqrt{(1-p_2)p_1}\right)^2 = (1-p_1)p_2 + (1-p_2)p_1 + 2\sqrt{(1-p_1)p_2(1-p_2)p_1} \tag{D.31}$$

$$= p_1 + p_2 - 2p_1p_2 + 2\sqrt{p_1(1-p_1)}\sqrt{p_2(1-p_2)} \tag{D.32}$$

$$\geq 2p - 2p^2 + 2p(1-p) = (2\sqrt{p(1-p)})^2, \tag{D.33}$$

where the inequality comes from (D.28) and (D.30). Therefore, we have established (D.23). Note that this is optimal as taking $p_1 = p_2 = p$ makes the two sides in (D.23) equal. □