The Fortieth AAAI Conference on Artificial Intelligence (AAAI-26)

Quantum Lipschitz Bandits

Bongsoo Yi[1], Yue Kang[2], Yao Li[1]

[1] University of North Carolina at Chapel Hill, Chapel Hill, NC, USA
[2] Microsoft, Redmond, WA, USA
{bongsoo, yaoli}@unc.edu, yuekang@microsoft.com

Abstract

The Lipschitz bandit is a key variant of stochastic bandit problems where the expected reward function satisfies a Lipschitz condition with respect to an arm metric space. With its wide-ranging practical applications, various Lipschitz bandit algorithms have been developed, achieving the optimal regret performance in the classical setting. Motivated by recent advancements in quantum computing and the demonstrated success of quantum Monte Carlo in simpler bandit settings, we introduce the first quantum Lipschitz bandit algorithms to address the challenges of continuous action spaces and non-linear reward functions. Specifically, we first leverage the elimination-based framework to propose an efficient quantum Lipschitz bandit algorithm named Q-LAE. Next, we present novel modifications to the classical Zooming algorithm, which results in a simple quantum Lipschitz bandit method, Q-Zooming. Both algorithms exploit the computational power of quantum methods to obtain a provably improved regret bound over classical Lipschitz bandit algorithms. Comprehensive experiments further validate our improved theoretical findings, demonstrating superior empirical performance compared to existing Lipschitz bandit methods.

1 Introduction

The multi-armed bandit (Slivkins et al. 2019) is a fundamental and versatile framework for sequential decision-making problems, with applications in online recommendation (Li et al. 2010), prompt engineering (Lin et al. 2023), clinical trials (Villar, Bowden, and Wason 2015) and hyperparameter tuning (Ding et al. 2022). An important extension of this framework is the Lipschitz bandit (Kleinberg, Slivkins, and Upfal 2019), which addresses bandit problems with a continuous and infinite set of arms defined within a known metric space. The expected reward function in this setting satisfies a Lipschitz condition, ensuring that similar actions yield similar rewards. By leveraging the structure of the metric space, the Lipschitz bandit provides an effective and practical framework for solving complex problems with large or infinite arm sets, making it broadly applicable to real-world scenarios (Mao, Leme, and Schneider 2018; Feng et al. 2023; Taş, Hauser, and Lauer 2021). After extensive studies on this intriguing problem (Kleinberg, Slivkins, and Upfal 2019; Slivkins 2011), it is well-established that the cumulative regret lower bound for any Lipschitz bandit algorithm is $\tilde{O}\left(T^{(d_z+1)/(d_z+2)}\right)$, where T is the time horizon and d_z is the zooming dimension. While state-of-the-art algorithms achieve this optimal bound, the question remains whether more advanced techniques can further enhance performance. In this context, quantum computation (DiVincenzo 1995; Biamonte et al. 2017) presents an exciting opportunity to accelerate algorithmic performance and unlock new possibilities in solving Lipschitz bandit problems.

Quantum computation has been successfully integrated into various bandit frameworks, including multi-armed bandits (Wan et al. 2023), kernelized bandits (Dai et al. 2024; Hikima et al. 2024), and heavy-tailed bandits (Wu et al. 2023), where it has demonstrated significant improvements in regret performance. In these quantum bandit settings, rather than receiving immediate reward feedback, we interact with quantum oracles that encode the reward distribution for each chosen arm. The Quantum Monte Carlo (QMC) method (Montanaro 2015) is then utilized to efficiently estimate the expected rewards, requiring fewer queries than classical approaches. Despite these advancements, the application of quantum computing to Lipschitz bandits remains unexplored. Although Lipschitz bandits may appear as a natural extension of multi-armed bandits, they introduce fundamentally greater analytical challenges. Unlike the discrete and finite action space in classical bandits, Lipschitz bandits operate over a continuous and uncountably infinite arm space, with an unknown and non-linear reward function. This complexity makes direct extensions of existing quantum algorithms infeasible. Therefore, novel methodologies are required to effectively bridge quantum computing with the Lipschitz bandit framework.

In this work, we propose the first two effective quantum Lipschitz bandit algorithms that achieve improved regret bounds by employing quantum computing techniques. Our main contributions are summarized as follows:

- We propose an elimination-based algorithm named *Quantum Lipschitz Adaptive Elimination* (Q-LAE), which achieves an improved regret bound of order $\tilde{O}\left(T^{d_z/(d_z+1)}\right)$ under two standard noise assumptions. The algorithm leverages the concepts of covering and maximal packing, making it applicable to general met-

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ALGORITHM	REFERENCE	SETTING	NOISE	REGRET BOUND
		CLASSICAL	SUB-GAUSSIAN	$O\left(T^{\frac{d_z+1}{d_z+2}}(\log T)^{\frac{1}{d_z+2}}\right)$
Q-LAE	THEOREM 4.2	QUANTUM	BOUNDED VALUE	$O\left(T^{\frac{d_z}{d_z+1}}(\log T)^{\frac{2}{d_z+1}}\right)$
	THEOREM 4.3	QUANTUM	BOUNDED VARIANCE	$O\left(T^{\frac{d_z}{d_z+1}}(\log T)^{\frac{7}{2}\frac{1}{d_z+1}}(\log\log T)^{\frac{1}{d_z+1}}\right)$
Q-ZOOMING	THEOREM 5.1	QUANTUM	BOUNDED VALUE	$O\left(T^{\frac{d_z}{d_z+1}}(\log T)^{\frac{1}{d_z+1}}\right)$
	THEOREM 5.3	QUANTUM	BOUNDED VARIANCE	$O\left(T^{\frac{d_z}{d_z+1}}(\log T)^{\frac{5}{2}\frac{1}{d_z+1}}(\log\log T)^{\frac{1}{d_z+1}}\right)$

Table 1: Comparison of regret bounds on Lipschitz bandits.

ric spaces. Compared to the existing elimination-based method, our algorithm adopts a more rigorous definition of the zooming dimension d_z, detailed in Section 4.

- We introduce the *Quantum Zooming* (Q-Zooming) Algorithm, which extends the classical zooming algorithm to the quantum Lipschitz bandit setting, also achieving the regret bound of order $\tilde{O}\left(T^{d_z/(d_z+1)}\right)$. This extension is non-trivial, requiring substantial modifications such as a stage-based design that efficiently leverages quantum oracles and integrates the QMC method.
- We evaluate the performance of our proposed algorithms, Q-LAE and Q-Zooming, through numerical experiments. The results demonstrate that both algorithms consistently outperform the classical Zooming algorithm across a variety of Lipschitz bandit scenarios.

Table 1 outlines the regret bounds achieved by our methods along with state-of-the-art classical Lipschitz bandit algorithms under different noise assumptions.

2 Related Work

This section reviews prior work on quantum online learning, including quantum bandits and quantum reinforcement learning. Due to space constraints, related work on Lipschitz bandits is deferred to Appendix C.

Prior work on bandit algorithms has made significant strides in integrating quantum computing into various bandit problems. Wan et al. (2023) first incorporated quantum computing into the multi-armed and stochastic linear bandits with the linear reward model and showed that an improved regret bound over the classical lower bound could be attained. Wu et al. (2023) extended this line of research to quantum bandits with heavy-tailed rewards, while Li and Zhang (2022) explored the quantum stochastic convex optimization problem. The best-arm identification problem in quantum multi-armed bandits has also been studied in (Wang et al. 2021b; Casalé et al. 2020). Parallel to quantum bandits, significant progress has been made in quantum reinforcement learning, particularly under linear reward models, as summarized in the comprehensive survey by Meyer et al. (2022). For instance, Wang et al. (2021a) showed that their quantum reinforcement learning algorithm could achieve better sample complexity than classical methods under a generative model of the environment. Furthermore, Ganguly et al. (2023) achieved improved regret bounds in the quantum setting without relying on a generative model. To deal with non-linear reward models, Dai et al. (2024) proposed a simple quantum algorithm for kernelized bandits by improving the confidence ellipsoid, and then an improved quantum kernelized bandit method was developed in (Hikima et al. 2024). However, unlike kernelized bandits, which leverage well-defined kernel functions to structure the action space, Lipschitz bandits face additional challenges due to the inherent complexity of non-linear reward functions and the unstructured, diverse nature of arm metric spaces. As a result, the application of quantum computing to Lipschitz bandits remains an open and significantly more difficult problem, one that we aim to address in this work.

3 Problem Setting and Preliminaries

In this section, we present the problem setting of the Lipschitz bandit problem and provide essential background on quantum computation, forming the foundation for the quantum bandit framework and our methods.

3.1 Lipschitz Bandits

The Lipschitz bandit problem can be characterized by a triplet $(X, \mathcal{D}, \mu)$, where X denotes the action space of arms, $\mathcal{D}$ is a metric on X, and $\mu : X \to \mathbb{R}$ represents the unknown expected reward function. The reward function μ is assumed to be 1-Lipschitz with respect to the metric space $(X, \mathcal{D})$, which implies:

$$|\mu(x_1) - \mu(x_2)| \leq \mathcal{D}(x_1, x_2), \quad \forall x_1, x_2 \in X.$$

Without loss of generality, we assume $(X, \mathcal{D})$ is a compact doubling metric space with its diameter no more than 1.

At each time step $t \leq T$, the agent selects an arm $x_t \in X$. The stochastic reward associated with the chosen arm is drawn from an unknown distribution P_x and is observed as $y_t = \mu(x_t) + \eta_t$, where η_t is independent noise with zero mean and finite variance. As in standard bandit settings, the agent's objective is to minimize the cumulative regret over T rounds, defined as $R(T) = \sum_{t=1}^{T}(\mu^* - \mu(x_t))$, where $\mu^* = \max_{x \in X} \mu(x)$ denotes the maximum expected reward. Additionally, the *optimality gap* for an arm $x \in X$ is given by $\Delta_x = \mu^* - \mu(x)$.

Zooming Dimension. The zooming dimension (Kleinberg, Slivkins, and Upfal 2019; Bubeck et al. 2008) is a fundamental concept in Lipschitz bandit problems over metric spaces. It characterizes the complexity of the problem by accounting for both the geometric structure of the arm set and the behavior of the reward function $\mu(\cdot)$. A key component of this concept is the *zooming number* $N_z(r)$, which represents the minimal number of radius-$r/3$ balls required to cover the set of near-optimal arms:

$$X_r = \{x \in X : r \leq \Delta_x < 2r\}, \tag{1}$$

where each arm has an optimality gap between r and $2r$.

Building on this, the *zooming dimension* d_z is introduced to quantify the growth rate of the zooming number $N_z(r)$. Formally, it is defined as:

$$d_z := \min\{d \geq 0 : \exists \alpha > 0,\ N_z(r) \leq \alpha r^{-d},\ \forall r \in (0, 1]\}.$$

Unlike the covering dimension d_c (Kleinberg, Slivkins, and Upfal 2008), which depends solely on the metric structure of the arm set X, the zooming dimension d_z captures the interaction between the metric and the reward function $\mu(\cdot)$. While the covering dimension characterizes the entire metric space, the zooming dimension focuses specifically on the near-optimal subset of X, which often results in d_z being substantially smaller than d_c. Notably, the zooming dimension d_z is not directly available to the agent, as it depends on the unknown reward function $\mu(\cdot)$. This dependence makes designing algorithms based on d_z particularly challenging.

Covering and Packing. We discuss the concepts of covering, packing, and maximal packing, which are fundamental to the algorithm proposed in Section 4. Let $(X, \mathcal{D})$ be a metric space with a subset $S \subseteq X$.

Definition 3.1. *A set of points $\{x_1, ..., x_n\} \subseteq S$ is an ϵ-covering of S if $S \subseteq \bigcup_{i=1}^{n} \mathcal{B}(x_i, \epsilon)$, where $\mathcal{B}(x, \epsilon) = \{y \in X \mid \mathcal{D}(x, y) \leq \epsilon\}$ denotes the closed ball of radius ϵ centered at x.*

Definition 3.2. *A set of points $\{x_1, ..., x_n\} \subseteq S$ is an ϵ-packing of S if $\mathcal{D}(x_i, x_j) \geq \epsilon$ for all $i \neq j$.*

Definition 3.3. *An ϵ-packing $\{x_1, ..., x_n\}$ is called a maximal ϵ-packing if no additional point $x \in S$ can be added without violating the packing condition.*

3.2 Quantum Computation

Basics of Quantum States and Measurements. In quantum mechanics, superposition is a fundamental concept that describes how a quantum system can simultaneously exist in multiple states until it is measured or observed. This contrasts with classical systems, where a system can only exist in a single state at a time. A *quantum state* in a Hilbert space $\mathbb{C}^n$ is represented as an L^2-normalized column vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top$, where $|x_i|^2$ denotes the probability of being in the i-th basis state. This superposition of states is expressed using Dirac notation as $|\mathbf{x}\rangle$. The conjugate transpose $\mathbf{x}^\dagger$ is written as $\langle\mathbf{x}|$, forming the bra-ket notation.

Given two quantum states $|\mathbf{x}\rangle \in \mathbb{C}^n$ and $|\mathbf{y}\rangle \in \mathbb{C}^m$, their joint quantum state is expressed by the tensor product $|\mathbf{x}\rangle|\mathbf{y}\rangle$, which can be written explicitly as $[x_1y_1, \ldots, x_ny_m]^\top \in \mathbb{C}^{nm}$. Quantum mechanics does not offer full knowledge of a state vector, but only allows partial information to be accessed through *quantum measurements*. These measurements are typically represented by a set of positive semi-definite Hermitian matrices $\{E_i\}_{i\in\Lambda}$, where Λ is the set of possible outcomes, and $\sum_{i\in\Lambda} E_i = I$. The probability of observing the outcome i is given by $\langle\mathbf{x}|E_i|\mathbf{x}\rangle$, which represents the inner product between $\langle\mathbf{x}|$ and $E_i|\mathbf{x}\rangle$. The condition $\sum_{i\in\Lambda} E_i = I$ ensures that the total probability of all possible outcomes equals 1. Once a measurement is performed, the original quantum state collapses into the specific state associated with the observed outcome.

Quantum Reward Oracle and Bandit Settings. Quantum algorithms process quantum states using *unitary operators*, which are referred to as *quantum reward oracles*. These oracles encode information about the reward distribution for a given action, replacing the immediate sample reward used in classical bandit settings.

In the quantum bandit framework, at each round, the learner selects an action x and gains access to the quantum unitary oracle $\mathcal{O}_x$. This oracle encodes the reward distribution P_x associated with the chosen action x and is formally defined as: $\mathcal{O}_x : |0\rangle \to \sum_{\omega\in\Omega_x} \sqrt{P_x(\omega)}|\omega\rangle|y^x(\omega)\rangle$, where Ω_x is the sample space of P_x and $y^x : \Omega_x \to \mathbb{R}$ represents the random reward corresponding to arm x. This unitary operator enables quantum algorithms to interact with reward distributions in a fundamentally different way compared to classical approaches.

Quantum Monte Carlo (QMC) Method. Estimating the mean of an unknown reward distribution is a crucial task in bandit problems. In pursuit of quantum speedup, we employ the Quantum Monte Carlo method (Montanaro 2015), which is designed for efficient mean estimation of unknown distributions. This approach demonstrates enhanced sample efficiency compared to classical methods.

Lemma 3.4 (Quantum Monte Carlo method (Montanaro 2015))**.** *Let $y : \Omega \to \mathbb{R}$ be a random variable with bounded variance, where Ω is equipped with a probability measure P, and the quantum unitary oracle $\mathcal{O}$ encodes P and y. With certain assumption about the noise, the QMC method offers the following guarantees:*

- *Bounded Noise: If $y \in [0, 1]$, there exists a constant $C_1 > 1$ and a quantum algorithm $QMC_1(\mathcal{O}, \epsilon, \delta)$ that outputs an estimate $\hat{y}$ of $\mathbb{E}[y]$, such that $\Pr(|\hat{y} - \mathbb{E}[y]| \geq \epsilon) \leq \delta$, while requiring at most $\frac{C_1}{\epsilon} \log \frac{1}{\delta}$ queries to $\mathcal{O}$ and $\mathcal{O}^\dagger$.*
- *Noise with Bounded Variance: If $Var(y) \leq \sigma^2$, then for $\epsilon < 4\sigma$, there exists a constant $C_2 > 1$ and a quantum algorithm $QMC_2(\mathcal{O}, \epsilon, \delta)$ that outputs an estimate $\hat{y}$ of $\mathbb{E}[y]$, such that $\Pr(|\hat{y} - \mathbb{E}[y]| \geq \epsilon) \leq \delta$, while requiring at most $\frac{C_2\sigma}{\epsilon} \log_2^{3/2}\left(\frac{8\sigma}{\epsilon}\right) \log_2\left(\log_2 \frac{8\sigma}{\epsilon}\right) \log \frac{1}{\delta}$ queries to $\mathcal{O}$ and $\mathcal{O}^\dagger$.*

From Lemma 3.4, we observe a notable quadratic improvement in sample complexity with respect to ϵ when estimating $\mathbb{E}[y]$. While classical Monte Carlo methods require $\widetilde{O}\left(1/\epsilon^2\right)$ samples, QMC achieves the same range of confidence interval with only $\widetilde{O}\left(1/\epsilon\right)$ samples. This substantial reduction in sample complexity is a fundamental advantage of quantum computing, which we would utilize in our following proposed methods.

4 Quantum Lipschitz Adaptive Elimination Algorithm

The structure of the QMC algorithm requires repeatedly playing a specific arm multiple times to obtain an updated reward estimate. The algorithm continues to play the same arm until a sufficient number of samples is collected, without updating its estimate. Inspired by the nature of the QMC algorithm and recent advancements in elimination-based algorithms, we propose a novel method called Quantum Lipschitz Adaptive Elimination (Q-LAE), which adapts the batched elimination framework and leverages the power of quantum machine learning.

Unlike existing elimination-based Lipschitz bandit algorithms, our approach adopts the consistent definition of the zooming dimension introduced by (Kleinberg, Slivkins, and Upfal 2019; Slivkins et al. 2019), as presented in Section 3. By doing so, it achieves improved regret bounds through the integration of quantum machine learning. In contrast, existing algorithms (Feng, Huang, and Wang 2022; Kang, Hsieh, and Lee 2023) rely on an alternative definition of the zooming dimension, which may be larger than the classical one and can lead to looser bounds in some cases. A detailed discussion on this issue is provided in the following Remark 4.1. **To the best of our knowledge, this is the first elimination-based Lipschitz bandit algorithm that adheres to the consistent definition of the zooming dimension.**

Remark 4.1. *In the Lipschitz bandit literature, two distinct definitions of the zooming dimension are commonly used. Specifically, Kleinberg, Slivkins, and Upfal (2019) and Slivkins et al. (2019) define the zooming number as the minimal number of balls required to cover the set of near-optimal arms* $X_r = \{x \in X : r \le \Delta_x < 2r\}$*, whereas Feng, Huang, and Wang (2022) defines it based on a broader set* $Y_r = \{x \in X : \Delta_x \le 2r\}$*. These differing definitions can yield substantially different values.*

For example, if the expected reward is constant across the entire arm space, then under our classical definition, X_r *is empty for all* r*, resulting in a zooming dimension of 0. In contrast,* Y_r *equals the entire space* X *for all* r*, implying that the zooming dimension is equivalent to the covering dimension of the space. In this work, we propose the first elimination-based Lipschitz bandit algorithm that adopts the former (classical) definition of the zooming dimension.*

To achieve accurate reward estimation for each ball, our algorithm selects a single representative arm, which is the center arm of each cube, and plays it multiple times. By utilizing mean reward estimates from QMC, Q-LAE adaptively concentrates on high-reward regions in continuous spaces through a combination of *selective elimination* and *progressive refinement*. This approach significantly reduces exploration in less promising areas, enhancing both efficiency and overall performance.

Algorithm 1: Q-LAE Algorithm

Input: time horizon T, fail probability δ
Initialization: $\mathcal{A}_1 \leftarrow$ maximal-$\frac{1}{2}$ packing of X, $\mathcal{C}_1 \leftarrow X$, $\epsilon_m = 2^{-m}$ for all m

for stage $m = 1, 2, ...$ **do**
  $n_m \leftarrow \frac{C_1}{\epsilon_m} \log\left(\frac{T}{\delta}\right)$
  for each $x \in \mathcal{A}_m$ **do**
    Play x for the next n_m rounds and obtain $\hat{\mu}_m(x)$ by running the $\mathrm{QMC}_1(\mathcal{O}_x, \epsilon_m, \delta/T)$ algorithm. Terminate the algorithm if the number of rounds played exceeds T.
  end for
  `// selective elimination`
  $\hat{\mu}_{max} \leftarrow \max_{x \in \mathcal{A}_m} \hat{\mu}_m(x)$
  For each $x \in \mathcal{A}_m$, eliminate x if $\hat{\mu}_m(x) < \hat{\mu}_{max} - 3\epsilon_m$. Let $\mathcal{A}_m^+$ denote the set of points not eliminated.
  `// progressive refinement`
  $\mathcal{C}_{m+1} \leftarrow \bigcup_{x \in \mathcal{A}_m^+} \mathcal{B}(x, \epsilon_m)$
  Find a maximal ϵ_{m+1}-packing of $\mathcal{C}_{m+1}$ and define it as $\mathcal{A}_{m+1}$.
end for

To explain the algorithm in detail, consider the beginning of each stage m, where the algorithm receives $\mathcal{A}_m$, a maximal ϵ_m-packing of the current active arm region $\mathcal{C}_m$. For every point in $\mathcal{A}_m$, the algorithm performs n_m plays and uses the QMC algorithm to estimate the mean rewards. Based on these estimates, the algorithm applies *selective elimination* to discard low-performing regions. Specifically, any point x at stage m is eliminated if its estimated mean reward $\hat{\mu}_m(x)$ is more than $3\epsilon_m$ lower than the best estimated reward among all points in $\mathcal{A}_m$. The set of remaining points after this elimination step is denoted by $\mathcal{A}_m^+$.

Next, the algorithm performs *progressive refinement*, which incrementally narrows the search space. The remaining active region is refined by further subdividing areas with higher potential rewards. More precisely, the active region is updated as

$$\mathcal{C}_{m+1} \leftarrow \bigcup_{x \in \mathcal{A}_m^+} \mathcal{B}(x, \epsilon_m). \tag{2}$$

The updated region $\mathcal{C}_{m+1}$ is then discretized through its maximal ϵ_{m+1}-packing, forming the next set of active points $\mathcal{A}_{m+1}$. This iterative process allows the algorithm to progressively concentrate exploration on more promising regions of the arm space. Notably, since a maximal ϵ-packing is an ϵ-covering, we can utilize maximal packing as representative points for the active region. This mathematical fact is detailed in Appendix A. The complete learning procedure is outlined in Algorithm 1.

4.1 Regret Analysis

Here, we analyze the regret of the Q-LAE algorithm. In the bandit literature, a *clean event* refers to the scenario where the reward estimates for all active arms and stages are bounded within their confidence intervals with high probability. While the classical (non-quantum) setting typically uses the Chernoff bound to establish deviation inequality between the true reward and its estimate, our quantum setting relies on the QMC algorithm.

At the end of line 4 in Algorithm 1, Lemma 3.4 ensures that the QMC algorithm provides an estimate $\hat{\mu}_m(x)$ that satisfies $|\hat{\mu}_m(x) - \mu(x)| \le \epsilon_m$ with probability at least $1 - \delta/T$. Since every arm $x \in \mathcal{A}_m$ is played at least once in each stage m, applying the union bound over all stages and arms guarantees that $|\hat{\mu}_m(x) - \mu(x)| \le \epsilon_m$ holds for all stages m and $x \in \mathcal{A}_m$ with probability at least $1 - \delta$. We define this as the clean event for this section and assume it holds throughout the subsequent regret analysis.

We now present a theorem that provides an upper bound on the regret of the Q-LAE algorithm, with the detailed proof provided in Appendix A.1.

Theorem 4.2. *Under the bounded noise assumption, the cumulative regret $R(T)$ of the Q-LAE Algorithm is bounded with high probability, at least $1 - \delta$, as follows:*

$$R(T) = O\left(T^{\frac{d_z}{d_z+1}} (\log T)^{\frac{2}{d_z+1}}\right),$$

where d_z represents the zooming dimension of the problem instance.

Theorem 4.2 establishes that our Q-LAE algorithm achieves a regret bound of $\tilde{O}\left(T^{d_z/(d_z+1)}\right)$, significantly improving upon the optimal regret bound of $\tilde{O}\left(T^{(d_z+1)/(d_z+2)}\right)$ attained by classical algorithms. Since the zooming dimension d_z is a small nonnegative number, often substantially smaller than the covering dimension d_c, this represents a major improvement. A more detailed discussion of the zooming dimension can be found in Appendix D.

Additionally, we provide a theorem that analyzes the regret of the Q-LAE algorithm under the assumption of bounded noise variance. In this setting, with a different choice of n_m, the algorithm's regret bound of $\tilde{O}\left(T^{d_z/(d_z+1)}\right)$ matches the result for the bounded noise setting in Theorem 4.2 up to logarithmic factors. More details and the analysis of Theorem 4.3 are deferred to Appendix A.2.

Theorem 4.3. *With the choice of $n_m = \frac{C_2\sigma}{\epsilon_m} \log_2^{3/2}\left(\frac{8\sigma}{\epsilon_m}\right) \log_2\left(\log_2 \frac{8\sigma}{\epsilon_m}\right) \log\left(\frac{T}{\delta}\right)$ in line 2, under the bounded variance noise assumption, the cumulative regret $R(T)$ of the Q-LAE Algorithm is bounded with high probability, at least $1 - \delta$, as follows:*

$$R(T) = O\left(T^{\frac{d_z}{d_z+1}} (\log T)^{\frac{7}{2}\frac{1}{d_z+1}} (\log\log T)^{\frac{1}{d_z+1}}\right),$$

where d_z is the zooming dimension of the problem instance.

5 Quantum Zooming Algorithm

Next, we introduce our second methodology, the Quantum Zooming (Q-Zooming) algorithm. This algorithm is a novel adaptation of the classical Zooming algorithm (Kleinberg, Slivkins, and Upfal 2019), which efficiently focuses on promising regions of the arm space by adaptively discretization. Our Q-Zooming algorithm achieves an improved regret bound compared to our first algorithm, Q-LAE. While inspired by its classical counterpart, our algorithm is not a straightforward extension. Unlike the classical version, the Q-Zooming algorithm operates in stages, and follows two primary rules: the *activation rule* and the *selection rule*. In this section, we describe these two rules and highlight the novel modifications introduced in our Q-Zooming algorithm.

Algorithm 2: Q-Zooming Algorithm

Input: time horizon T, fail probability δ
Initialization: active arms set $S \leftarrow \emptyset$, confidence radius $\epsilon_0(\cdot) = 1$

1: **for** stage $s = 1, 2, ..., m$ **do**
2:   `// activation rule`
3:   **if** there exists an arm y that is not covered by the confidence balls of active arms **then**
4:     add y to the active set: $S \leftarrow S \cup \{y\}$
5:   **end if**
6:   `// selection rule`
7:   $x_s \leftarrow \arg\max_{x \in S} \hat{\mu}_{s-1}(x) + 2\epsilon_{s-1}(x)$
8:   $\epsilon_s(x) \leftarrow \begin{cases} \epsilon_{s-1}(x)/2 & \text{if } x = x_s, \\ \epsilon_{s-1}(x) & \text{if } x \ne x_s. \end{cases}$
9:   $N_s \leftarrow \frac{C_1}{\epsilon_s(x_s)} \log\left(\frac{m}{\delta}\right)$
10:   If $\sum_{k=1}^{s} N_s > T$, terminate the algorithm.
11:   Play x_s for the next N_s rounds and obtain $\hat{\mu}_s(x_s)$ by running the QMC$_1(\mathcal{O}_{x_s}, \epsilon_s(x_s), \delta/m)$ algorithm.
    Note: For all other arms $x \ne x_s$, retain the reward estimates $\hat{\mu}_{s-1}(x)$ from stage $s - 1$.
12: **end for**

The *activation rule* ensures that any arm not covered by the confidence balls of the active arms is added to the active set. Formally, an arm y is considered covered at stage s if there exists $x \in S$ such that $\mathcal{D}(x, y) \le \epsilon_{s-1}(x)$, where S is the current set of active arms, $\mathcal{D}(x, y)$ is the distance between arms x and y, and $\epsilon_{s-1}(x)$ is the confidence radius for arm x updated at stage $s - 1$. If such an uncovered arm y exists, it is added to the active set S.

The *selection rule* determines which arm to sample next from the active set. Specifically, it selects the arm $x \in S$ with the highest index, defined as $\hat{\mu}_{s-1}(x) + 2\epsilon_{s-1}(x)$, where $\hat{\mu}_{s-1}(x)$ denotes the estimated mean reward of arm x and $\epsilon_{s-1}(x)$ is its corresponding confidence radius.

Our Q-Zooming algorithm combines these activation and selection rules with several novel techniques to adapt to the quantum computing setting. Unlike the classical approach, where a single sample is played once per round, our algorithm leverages QMC for reward estimation. This fundamental difference requires multiple queries to the quantum oracle of a selected arm to accurately estimate its mean reward. To handle this, we partition the time horizon T into multiple stages. At each stage, the algorithm selects the arm with the highest index and allocates multiple queries to estimate its reward with high precision. Additionally, as described in line 8 of Algorithm 2, the confidence radius shrinks by half at each stage only when the corresponding arm is selected. This guarantees that the total number of plays for any given arm remains bounded, a property that is crucial for the analysis of our algorithm. We also show that the optimality gap of any active arm is bounded by its confidence radius. Since this confidence radius shrinks only when the arm is selected, it implies that, in later stages, only arms with rewards close to that of the optimal arm are repeatedly selected. This property is fundamental to the algorithm's effectiveness.

The procedure of the Q-Zooming algorithm is detailed in Algorithm 2. It begins with an empty set of active arms S and an initial confidence radius of 1 for all arms. At each stage s, the algorithm activates an arm that is not covered by the confidence balls and selects the best arm x_s based on the activation and selection rules described earlier. The confidence radius $\epsilon_s(x_s)$ for the selected arm x_s is then updated by halving its value. Using this updated $\epsilon_s(x_s)$, the algorithm determines the number of queries N_s required for the quantum oracle. The algorithm then plays the selected arm x_s for N_s rounds and updates the reward estimate $\hat{\mu}_s(x_s)$ using the QMC algorithm. This step provides a more accurate reward estimate, bounding the estimation error such that $|\hat{\mu}_s(x_s) - \mu_s(x_s)| \leq \epsilon_s(x_s)$ with probability at least $1 - \delta/m$, as established in Lemma 3.4. Once these steps are completed, the algorithm proceeds to the next stage unless the total number of rounds exceeds the time horizon T, in which case the algorithm terminates.

5.1 Regret Analysis

We provide a regret analysis for the proposed Q-Zooming algorithm. Similar to the approach described in Section 4.1, we utilize the clean event approach to structure the analysis.

Following each stage, Lemma 3.4 guarantees that QMC, with a sufficient number of queries N_s, provides an estimate $\hat{\mu}_s(x_s)$ satisfying $|\hat{\mu}_s(x_s) - \mu(x_s)| \leq \epsilon_s(x_s)$ with probability at least $1 - \frac{\delta}{m}$, where m is the total number of stages. By applying the union bound over all stages $1 \leq s \leq m$, it follows that $|\hat{\mu}_s(x_s) - \mu(x_s)| \leq \epsilon_s(x_s)$ holds for all $1 \leq s \leq m$ with probability at least $1 - \delta$.

Additionally, this result extends to all arms x. Specifically, $|\hat{\mu}_s(x) - \mu(x)| \leq \epsilon_s(x)$ holds for all $1 \leq s \leq m$ and every arm x, with probability at least $1 - \delta$. This is because $\epsilon_s(x)$ is initially set to 1 for arms that have never been played, and both $\epsilon_s(x)$ and $\hat{\mu}_s(x)$ remain unchanged for arms not selected during stage s. We define this condition as a clean event and assume it holds throughout the subsequent analysis. We now establish the following high-probability upper bound on the regret for the Q-Zooming algorithm. The proof is deferred to Appendix B.1.

Theorem 5.1. *Under the bounded noise assumption, the cumulative regret $R(T)$ of the Q-Zooming Algorithm is bounded with high probability, at least $1 - \delta$, as follows:*

$$R(T) = O\left(T^{\frac{d_z}{d_z+1}} (\log T)^{\frac{1}{d_z+1}}\right),$$

where d_z is the zooming dimension of the problem instance.

Theorem 5.1 shows that the Q-Zooming algorithm attains a regret bound of $\tilde{O}\left(T^{d_z/(d_z+1)}\right)$, surpassing the optimal regret bound of classical Lipschitz bandit algorithms, $\tilde{O}\left(T^{(d_z+1)/(d_z+2)}\right)$. This enhancement is similar to that achieved by the Q-LAE algorithm in Section 4. As noted in Section 4 and Appendix D, since d_z is often much smaller than d_c, this improvement is particularly significant.

Remark 5.2. *The regret bounds for the Q-Zooming and Q-LAE algorithms are identical, ignoring polylogarithmic factors. However, when considering the $\log T$ term, Q-Zooming exhibits a tighter bound. The key difference lies in their strategies: Q-LAE focuses on completely eliminating low-reward regions, while Q-Zooming, without eliminating arms, adopts a strategy of further discretizing high-reward regions to explore them more effectively. Despite their theoretical regret bounds, the practical performance of these two algorithms can vary depending on how quickly Q-LAE eliminates low-reward regions, as we illustrate in Section 6.*

Under the bounded variance assumption, using a different choice of N_s, the regret bound remains consistent with Theorem 5.1, differing only by logarithmic factors. A detailed analysis for the bounded variance case, including the proof of Theorem 5.3, is provided in Appendix B.2.

Theorem 5.3. *With the choice of $N_s = \frac{C_2\sigma}{\epsilon_s(x_s)} \log_2^{3/2}\left(\frac{8\sigma}{\epsilon_s(x_s)}\right) \log_2\left(\log_2 \frac{8\sigma}{\epsilon_s(x_s)}\right) \log\left(\frac{m}{\delta}\right)$ in line 9, the cumulative regret $R(T)$ of the Q-Zooming Algorithm under the bounded variance noise assumption is bounded with high probability, at least $1 - \delta$, as follows:*

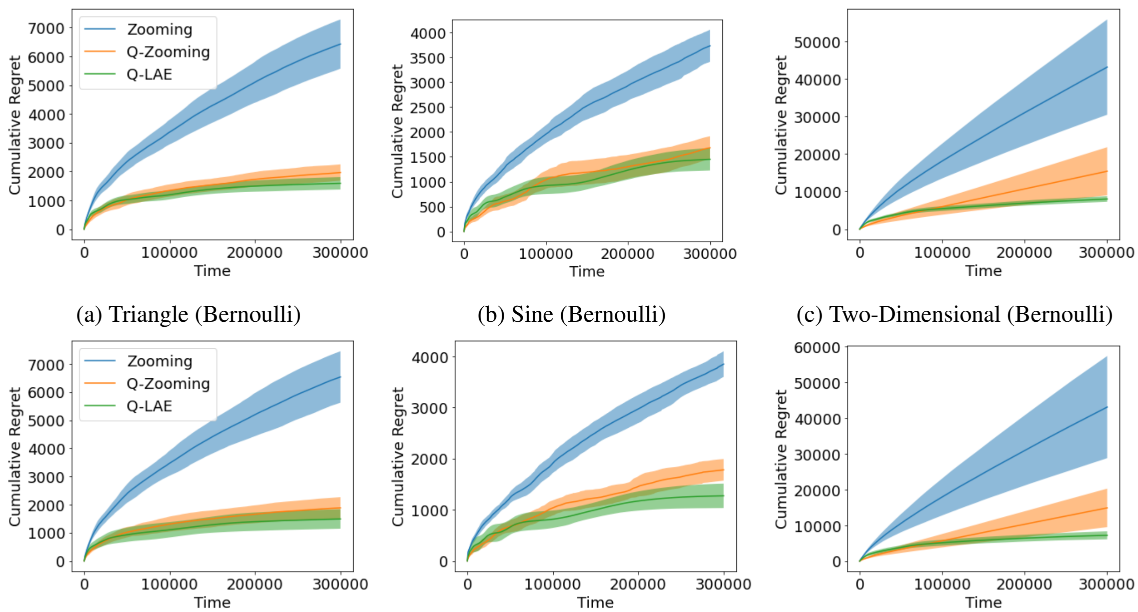
$$R(T) = O\left(T^{\frac{d_z}{d_z+1}} (\log T)^{\frac{5}{2}\frac{1}{d_z+1}} (\log\log T)^{\frac{1}{d_z+1}}\right),$$

where d_z is the zooming dimension of the problem instance.

6 Experiments

We evaluate the regret performance of our proposed quantum algorithms, Q-LAE and Q-Zooming, against the classical Zooming algorithm (Kleinberg, Slivkins, and Upfal 2019) on Lipschitz bandit problems. Specifically, we conduct experiments on three Lipschitz functions, following the common settings in studies on Lipschitz bandit problems (Kang, Hsieh, and Lee 2023; Feng, Huang, and Wang 2022): (1) $\mu(x) = 0.9 - 0.95|x - 1/3|$ with $(X, \mathcal{D}) = ([0, 1], |\cdot|)$ (triangle), (2) $\mu(x) = 0.35 \sin(3\pi x/2)$ with $(X, \mathcal{D}) = ([0, 1], |\cdot|)$ (sine), and (3) $\mu(x) = 1.2 - 0.95\|x - (0.8, 0.7)\|_2 - 0.3\|x - (0, 1)\|_2$ with $(X, \mathcal{D}) = ([0, 1]^2, \|\cdot\|_\infty)$ (two-dimensional). In all three cases, the reward function $\mu(x)$ is bounded within the interval $[0, 1]$. We also consider two types of noise, as described in Lemma 3.4 and aligned with our theoretical analysis: (a) bounded noise, where the output y is modeled as a Bernoulli random variable with $\mu(x)$ as the probability of success ($y = 1$), and (b) noise with bounded variance, where zero-mean Gaussian noise with variance σ^2 is added directly to $\mu(x)$ as the observed reward.

The QMC algorithm and our proposed quantum algorithms, Q-LAE and Q-Zooming, are implemented using the Python package Qiskit (Javadi-Abhari et al. 2024). In our experiments, we set the time horizon to $T = 300,000$ and the failure probability to $\delta = 0.05$. Gaussian noise is sampled from a normal distribution $N(0, \sigma^2 = 0.1)$. We evaluate performance by averaging the cumulative regret over 30



(a) Triangle (Bernoulli) (b) Sine (Bernoulli) (c) Two-Dimensional (Bernoulli)

(d) Triangle (Gaussian) (e) Sine (Gaussian) (f) Two-Dimensional (Gaussian)

Figure 1: Average Cumulative Regret. The regret performance for each function and noise model is evaluated as the average cumulative regret over 30 independent runs, with the corresponding standard deviations also reported.

independent trials. Both the mean and standard deviation of the cumulative regret are reported in Figure 1.

Figure 1 shows that both Q-LAE and Q-Zooming consistently outperform the classical Zooming algorithm across all scenarios. This provides strong empirical support for the effectiveness of our proposed quantum approaches. The advantage holds under both types of noise considered, highlighting the benefits of integrating quantum techniques into the Lipschitz bandit framework.

A noteworthy observation is that Q-LAE often achieves comparable or slightly better performance when compared to Q-Zooming. Although Q-Zooming possesses a theoretically superior regret bound due to its favorable polylogarithmic factor, Q-LAE's elimination-based structure offers distinct practical advantages. This practical benefit allows Q-LAE to marginally outperform Q-Zooming over time. Intuitively, Q-LAE achieves this by progressively eliminating low-reward regions, enabling it to become more focused and improve its performance in later stages. However, in the early stages, with most arms still active, Q-LAE explores broadly and may include suboptimal regions, potentially making it slightly less efficient than Q-Zooming initially. Nonetheless, as learning progresses, Q-LAE rapidly converges, ultimately delivering superior overall performance.

7 Discussion

In this work, we introduced the first two quantum Lipschitz bandit algorithms, named Q-LAE and Q-Zooming, under the non-linear reward functions and arbitrary continuous arm metric space. We provided a detailed theoretical analysis to illustrate that both of our efficient algorithms can achieve an improved regret bound of order $\tilde{O}(T^{d_z/(d_z+1)})$ when the noise is bounded or has finite variance. The superiority of our proposed methods over state-of-the-art approaches is validated under comprehensive experiments.

Comparison of Q-LAE and Q-Zooming: While Q-Zooming achieves a better regret bound when considering logarithmic terms, its empirical performance does not consistently outperform that of Q-LAE. As demonstrated in Section 6, Q-LAE can surpass Q-Zooming in certain scenarios due to its elimination-based strategy, which efficiently prunes low-reward regions and concentrates exploration on more promising areas. This targeted approach often leads to faster convergence, especially when the reward function contains large suboptimal regions. Moreover, Q-LAE is the first elimination-based Lipschitz bandit algorithm to adopt the consistent definition of the zooming dimension (see Remark 4.1), leading to a novel and distinctive regret analysis. We believe this offers a valuable contribution to the Lipschitz bandit literature. It is also worth noting that zooming-based algorithms, both classical and quantum, face scalability challenges in high-dimensional settings, whereas Q-LAE does not suffer from this limitation. Therefore, we view Q-LAE as a complementary alternative to Q-Zooming, offering unique strengths in both theoretical formulation and practical effectiveness.

Limitations: A limitation of our work is the absence of a theoretical lower bound, leaving the optimality of our achieved regret bound uncertain. However, as lower bounds are also unresolved for simpler cases like quantum multi-armed bandit, this would remain a challenging future work.

Acknowledgments

This work was supported in part by the National Science Foundation under grants DMS-2152289 and DMS-2134107. We would also like to express sincere gratitude to Dr. Zhongxiang Dai for his invaluable guidance throughout this work.

References

Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; and Lloyd, S. 2017. Quantum machine learning. *Nature*, 549(7671): 195–202.

Bubeck, S.; Stoltz, G.; Szepesvári, C.; and Munos, R. 2008. Online optimization in X-armed bandits. *Advances in Neural Information Processing Systems*, 21.

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

Dai, Z.; Lau, G. K. R.; Verma, A.; Shu, Y.; Low, B. K. H.; and Jaillet, P. 2024. Quantum bayesian optimization. *Advances in Neural Information Processing Systems*, 36.

Ding, Q.; Kang, Y.; Liu, Y.-W.; Lee, T. C. M.; Hsieh, C.-J.; and Sharpnack, J. 2022. Syndicated bandits: A framework for auto tuning hyper-parameters in contextual bandit algorithms. *Advances in Neural Information Processing Systems*, 35: 1170–1181.

DiVincenzo, D. P. 1995. Quantum computation. *Science*, 270(5234): 255–261.

Feng, Y.; Huang, z.; and Wang, T. 2022. Lipschitz Bandits with Batched Feedback. In *Advances in Neural Information Processing Systems*, volume 35, 19836–19848.

Feng, Y.; Luo, W.; Huang, Y.; and Wang, T. 2023. A Lipschitz bandits approach for continuous hyperparameter optimization. *arXiv preprint arXiv:2302.01539*.

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*.

Hikima, Y.; Kazunori.Murao; Takemori, S.; and Umeda, Y. 2024. Quantum Kernelized Bandits. In *The 40th Conference on Uncertainty in Artificial Intelligence*.

Javadi-Abhari, A.; Treinish, M.; Krsulich, K.; Wood, C. J.; Lishman, J.; Gacon, J.; Martiel, S.; Nation, P. D.; Bishop, L. S.; Cross, A. W.; Johnson, B. R.; and Gambetta, J. M. 2024. Quantum computing with Qiskit. arXiv:2405.08810.

Kang, Y.; Hsieh, C.-J.; and Lee, T. C. M. 2023. Robust lipschitz bandits to adversarial corruptions. *Advances in Neural Information Processing Systems*, 36: 10897–10908.

Kleinberg, R.; Slivkins, A.; and Upfal, E. 2008. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, 681–690.

Kleinberg, R.; Slivkins, A.; and Upfal, E. 2019. Bandits and experts in metric spaces. *Journal of the ACM (JACM)*, 66(4): 1–77.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670.

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.

Lin, X.; Wu, Z.; Dai, Z.; Hu, W.; Shu, Y.; Ng, S.-K.; Jaillet, P.; and Low, B. K. H. 2023. Use your instinct: Instruction optimization using neural bandits coupled with transformers. *arXiv preprint arXiv:2310.02905*.

Mao, J.; Leme, R.; and Schneider, J. 2018. Contextual pricing for lipschitz buyers. *Advances in Neural Information Processing Systems*, 31.

Meyer, N.; Ufrecht, C.; Periyasamy, M.; Scherer, D. D.; Plinge, A.; and Mutschler, C. 2022. A survey on quantum reinforcement learning. *arXiv preprint arXiv:2211.03464*.

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

Slivkins, A. 2011. Contextual bandits with similarity information. In *Proceedings of the 24th annual Conference On Learning Theory*, 679–702. JMLR Workshop and Conference Proceedings.

Slivkins, A.; et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2): 1–286.

Taş, Ö. Ş.; Hauser, F.; and Lauer, M. 2021. Efficient sampling in pomdps with lipschitz bandits for motion planning in continuous spaces. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, 1081–1088. IEEE.

Villar, S. S.; Bowden, J.; and Wason, J. 2015. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2): 199.

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*, volume 37, 10087–10094.

Wang, D.; Sundaram, A.; Kothari, R.; Kapoor, A.; and Roetteler, M. 2021a. Quantum algorithms for reinforcement learning with a generative model. In *International Conference on Machine Learning*, 10916–10926. PMLR.

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

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