# Quantum Non-Linear Bandit Optimization

**Zakaria Shams Siam**[1], **Chaowen Guan**[2], **Chong Liu**[1]

[1]University at Albany, State University of New York
[2]University of Cincinnati
{zsiam,cliu24}@albany.edu, guance@ucmail.uc.edu

## Abstract

We study non-linear bandit optimization where the learner maximizes a black-box function with zeroth order function oracle, which has been successfully applied in many critical applications such as drug discovery and materials design. Existing works have showed that with the aid of quantum computing, it is possible to break the classical $\Omega(\sqrt{T})$ regret lower bound and achieve the new $O(\mathrm{poly}\log T)$ upper bound. However, they usually assume that the objective function sits within the reproducing kernel Hilbert space and their algorithms suffer from the curse of dimensionality. In this paper, we propose the new Q-NLB-UCB algorithm which enjoys an *input dimension-free* $O(\mathrm{poly}\log T)$ upper bound, making it applicable for high-dimensional tasks. At the heart of our algorithm design are quantum Monte Carlo mean estimator, parametric function approximation technique, and a new quantum non-linear regression oracle, which can be of independent interests in more quantum machine learning problems. Our algorithm is also validated for its efficiency compared with other quantum algorithms on both high-dimensional synthetic and real-world tasks.

**Code** — https://github.com/ZakSiam/Quantum-Non-Linear-Bandit-Optimization

**Extended version** — https://arxiv.org/abs/2503.03023

## Introduction

Non-linear bandit optimization, a.k.a., Gaussian process bandits, kernelized bandits, or Bayesian optimization, is a sequential decision making-based machine learning task that aims at solving a black-box optimization problem. Due to its black-box nature, it has been successfully applied in many important real-world applications where objective functions are difficult to define explicitly, e.g., hyperparameter tuning (Wu et al. 2020), neural architecture search (Kandasamy et al. 2018), drug discovery (Korovina et al. 2020), and materials science (Frazier and Wang 2016).

In drug screening, the objective is to identify a drug candidate from a large pool of compounds that exhibits the highest binding affinity to a specific biological target. Here each candidate is usually described by a feature vector $\mathbf{x}$ and its binding affinity is a function of $\mathbf{x}$, denoted as $f_0(\mathbf{x})$. Due to the highly complex chemical and biological reactions, the function $f_0$ is usually considered "black-box", which implies that it may be non-linear, non-convex, and even non-differentiable w.r.t. $\mathbf{x}$. How to optimize $f_0$ then? The learner is allowed to sequentially query the function $f_0$. At each round $t$, the learner chooses to take a data point $\mathbf{x}_t$ and observes its performance $y_t$, which is the outcome of a wet lab test. Obviously, a good algorithm can help the learner select promising data points as the experiment progresses and find the best candidate within fewest tests.

In addition to its successful real-world applications, non-linear bandit optimization also enjoys solid theoretical guarantees. In literature, researchers usually define simple regret and cumulative regret to capture the convergence behavior of a certain algorithm, and many regret upper bounds have been established under different assumptions and with different kernels. All these positive theoretical results further contribute to more applications of non-linear bandit optimization. However, unfortunately an $\Omega(\sqrt{T})$ cumulative regret lower bound (Scarlett, Bogunovic, and Cevher 2017) cannot be further improved. What does that mean? It implies that given total $T$ rounds, no algorithm can incur cumulative regret less than $\Omega(\sqrt{T})$ asymptotically.

But can we do better? On the negative side, in classical (non-quantum) setting, the answer is "no". On the positive side, we have entered the quantum era where the power of quantum computing offers new hope for tackling this challenging optimization problem. (Wan et al. 2023) first studied the multi-armed bandits and linear bandits and proved that new $O(\mathrm{poly}\log T)$ regret bound can be achieved with the aid of quantum computing. Later Q-GP-UCB (Dai et al. 2023) and QMCKernelUCB (Hikima et al. 2024) studied the quantum Bayesian optimization, generalizing (Wan et al. 2023) to non-convex and non-linear settings, still achieving the $O(\mathrm{poly}\log T)$ regret bound. However, (Dai et al. 2023; Hikima et al. 2024) both heavily rely on the reproducing kernel Hilbert space assumption, which suffers from the curse of dimensionality. The problem is that, in practice, many input data sit in high-dimensional spaces. Again in drug discovery, for example, the dimension of protein sequences usually ranges from thousands (Clarke et al. 2008) to even millions (Rahnenführer et al. 2023). Then either $O(d_x^{\frac{3}{2}}(\log T)^{\frac{3}{2}})$ (linear kernel) or $O(T^{\frac{3d_x}{2v+d_x}})$ (Matérn kernel with $v$ being a

kernel parameter) regret bound (Dai et al. 2023) become vacuous when input dimension $d_x$ goes to millions, which further prevents their high-dimensional real-world applications. Therefore, can we design a new and efficient quantum non-linear bandit optimization algorithm that works well in high-dimensions?

In this paper, we answer this question affirmatively by proposing the Quantum Non-Linear Bandit with Upper Confidence Bound (Q-NLB-UCB) algorithm. The key design of Q-NLB-UCB relies on three techniques. First, Q-NLB-UCB runs in stages where in each stage the quantum oracle associated with the same action will be queried multiple times to achieve the quadratically improved sample complexity, guaranteed by quantum Monte Carlo mean estimator lemma (Montanaro 2015). Second, inspired by the success of parametric function approximation (Liu and Wang 2023), we use a parametric function class to approximate and optimize the black-box objective function. All information queried is handled in the parameter space, so we are able to prove the first input *dimension-free* regret bound for quantum non-linear bandit optimization. Finally, initialization of Q-NLB-UCB relies on a good estimated parameter $\hat{\mathbf{w}}_0$ of the quantum non-linear regression problem. Its convergence to the optimal parameter $\mathbf{w}^*$ enjoys a quadratic speed-up rate compared with classical non-linear regression, thanks to the quantum fast-forward technique (Apers and Sarlette 2019).

**Contributions.** Our contributions are summarized as:

(1) We solve quantum non-linear bandit optimization with quantum computing and parametric function approximation, and propose the new Q-NLB-UCB algorithm. The design of algorithmic framework is generic and the choice of parametric function can be a linear function, quadratic function, or even multi-layer deep neural network, depending on tasks.

(2) Different from existing works, Q-NLB-UCB does not suffer from the curse of dimensionality. We prove the first $O(d_w^2 \log^{\frac{3}{2}}(T) \log(d_w \log T))$ regret bound with $d_w$ being parameter complexity, which is also faster than the classical lower bound $\Omega(\sqrt{T})$ but *independent* to input dimension $d_x$.

(3) Experiments on high-dimensional synthetic functions and real-world tasks show that Q-NLB-UCB outperforms compared algorithms in regrets and runtime.

**Technical Novelties.** The design of Q-NLB-UCB is a highly non-trivial task, involving tackling multiple technical challenges. Key technical novelties are highlighted as follows.

(1) The classical $O(1/\sqrt{T_0})$ regression bound (Nowak 2009) with $T_0$ being number of samples converges too slow to work for Q-NLB-UCB. We introduce the quantum fast-forward technique (Apers and Sarlette 2019) to refine the analysis with Craig-Bernstein inequality (Craig 1933) to work with the quantum non-linear regression oracle, and achieve a *quadratic* improvement in query complexity. This enhancement is highly non-trivial. As far as we know, *no* existing classical or quantum methods can provide a comparable speed-up for the non-linear regression problem. Informally speaking, we prove that there exist such quantum regression solvers attaining this quadratic advantage, shedding light for a specific algorithm design in the future, which can be of independent interests in more quantum machine learning problems. While classical approaches (Diaconis and Miclo 2013) offer some improvement, they are limited to constant-factor gains, rather than quadratic ones.

(2) In confidence analysis, when constructing the covariance matrix, we take the gradient w.r.t. the fixed initial parameter $\hat{\mathbf{w}}_0$, rather than $\hat{\mathbf{w}}_t$, which still makes rank-1 updates in each stage but avoids the tedious inductive argument in (Liu and Wang 2023). After the first-order approximation using gradients, multiple parameters sitting in the same convex confidence region are used as bridges to apply convexity properties, ensuring the high-order terms are still well bounded.

## Related Work

**Classical Optimization.** Bayesian optimization (Frazier 2018) is one of the most popular methods to solve global optimization. Based on the Gaussian process (Williams and Rasmussen 2006) or reproducing kernel Hilbert space (Chowdhury and Gopalan 2017) assumption, Bayesian optimization runs in multiple rounds where in each round the learner takes an action suggested by an acquisition function. Common choices of acquisition function include expected improvement (Jones, Schonlau, and Welch 1998), knowledge gradient (Scott, Frazier, and Powell 2011), upper confidence bound (Srinivas et al. 2010), and Thompson sampling (Russo et al. 2018). Without the classical Gaussian process assumption, (Snoek et al. 2015; Springenberg et al. 2016) used neural networks as the backbone surrogate models.

Besides Bayesian optimization, recent bandit works studied global optimization with neural network approximation (Zhou, Li, and Gu 2020; Zhang et al. 2021; Dai et al. 2022) or generic parametric function approximation (Liu and Wang 2023). In addition to bandit optimization, (Wang, Balakrishnan, and Singh 2019) studied the global optimization of an unknown non-convex smooth function. When gradient information is available, (Allen-Zhu 2018; Fang et al. 2018) also proposed algorithms to solve global optimization problem. However, our work is different from all of them since we focus on the quantum bandit optimization.

**Quantum Optimization.** In recent years, there has been increasing interest in exploring quantum speed-up for optimization problems. This research direction began with quantum algorithms for linear and semi-definite programs (Brandão and Svore 2017; Apeldoorn and Gilyén 2018; Casares and Martin-Delgado 2020; Kerenidis and Prakash 2020), later extending to more general convex optimization (van Apeldoorn et al. 2020; Chakrabarti et al. 2020; He et al. 2022, 2024). Recent advancements include quantum algorithms for slightly convex problems (Li and Zhang 2022; Chen et al. 2025; Zhang et al. 2024), escaping saddle points in non-convex landscapes (Zhang, Leng, and Li 2021; Childs et al. 2022), and identifying global minima in specific non-convex cases (Liu, Su, and Li 2023; Leng et al. 2023). Alongside these algorithmic developments, quantum lower bounds have been established for both convex (Garg et al. 2021b,a) and non-convex optimization (Gong, Zhang, and Li 2025; Zhang and Li 2023).

In a parallel line of research, stochastic quantum methods were proposed (Sidford and Zhang 2023), demonstrating the advantages of quantum stochastic first-order oracles for

smooth objectives in low-dimensional settings. Most recently, there were efforts (Liu et al. 2024) on investigating quantum speed-up for minimizing non-smooth, non-convex objectives, which represent the most general and fundamental function class. At the same time, (Zhang et al. 2024) focused on studying quantum algorithms and lower bounds for finite-sum optimization, addressing both convex and non-convex cases.

# Preliminaries

## Problem Statement

In this paper, we consider the non-linear bandit optimization problem: $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} f_0(\mathbf{x})$, where $f_0 : \mathcal{X} \to \mathcal{Y}$ is the unknown black-box objective function, which can be non-linear, non-convex, and not necessarily differentiable in $\mathbf{x}$. $\mathcal{X} \subseteq \mathbf{R}^{d_x}$ is the function domain and $\mathcal{Y} \subseteq \mathbf{R}$ is the function range. To solve this problem, the learner has zeroth-order access to $f_0$ and the whole process runs in rounds. At each round $t = 1, ..., T$, after querying action $\mathbf{x}_t$ the oracle will return a noisy function observation $y_t$. In classical (non-quantum) setting, after taking $\mathbf{x}_t$, the function returns $y_t = f_0(\mathbf{x}_t) + \eta_t$, where $\eta_t$ is the zero-mean, independent, $\sigma$-sub-Gaussian noise. However, in (bounded value) quantum bandit setting, after taking the same action $\mathbf{x}_t$ multiple times, the function oracle returns $y_t$ that satisfies $|y_t - f_0(\mathbf{x}_t)| \leq \epsilon_t$, where $\epsilon_t$ is an error term. Either in classical or quantum setting, throughout $T$ rounds, we can always utilize the cumulative regret to evaluate the optimization process, $R_T = \sum_{t=1}^{T} f_0(\mathbf{x}^*) - f_0(\mathbf{x}_t)$, where $r_t = f_0(\mathbf{x}^*) - f_0(\mathbf{x}_t)$ is the instantaneous regret at round $t$. An algorithm $\mathcal{A}$ is said to be a no-regret algorithm if $\lim_{t\to\infty} R_T(\mathcal{A})/T \to 0$.

Since we are using a parametric function class to approximate the objective function, we use $\mathcal{W} \subseteq \mathbf{R}^{d_w}$ to denote the parameter class and its corresponding parametric function class is $\mathcal{F} = \{f_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y} | \mathbf{w} \in \mathcal{W}\}$. Here we abuse the notation since $f_{\mathbf{w}}(\mathbf{x}) = f_{\mathbf{x}}(\mathbf{w})$ and $f_{\mathbf{x}}(\mathbf{w})$ is used when $\mathbf{w}$ is the variable of interest and $\mathbf{x}$ is the parameter, and vice versa. Also, we use $\nabla f_{\mathbf{x}}(\mathbf{w})$ and $\nabla^2 f_{\mathbf{x}}(\mathbf{w})$ denote the gradient vector and Hessian matrix of function w.r.t. $\mathbf{w}$. For a vector $\mathbf{x}$, its $\ell_p$ norm is defined as $\|\mathbf{x}\|_p = (\sum_{i=1}^{d} |\mathbf{x}_i|^p)^{1/p}$. For a matrix $\mathbf{A}$, its operator norm is denoted as $\|\mathbf{A}\|_{\text{op}}$. For a vector $\mathbf{x}$ and a matrix $\mathbf{A}$, let $\|\mathbf{x}\|_{\mathbf{A}}^2 = \mathbf{x}^\top \mathbf{A}\mathbf{x}$. Let $\ell(\cdot, \cdot) : \mathbf{R}^2 \to \mathbf{R}$ denote a loss function, then the expected risk of a parameter $\mathbf{w}$ is defined as $L(\mathbf{w}) = \mathbf{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell(f_{\mathbf{w}}(\mathbf{x}), y)]$ with respect to distribution $\mathcal{D}$ and the empirical risk of $\mathbf{w}$ is defined as $\hat{L}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(f_{\mathbf{w}}(\mathbf{x}_i), y_i)$ with respect to $n$ data points. For readers' convenience, we use standard big $O$ notation to hide universal constants and use $\tilde{O}$ notation to further hide logarithmic factors.

## Quantum Computation

**Quantum Basics.** A quantum state can be seen as a vector $\mathbf{x} = (x_1, x_2, \ldots, x_m)^\top$ in Hilbert space $\mathbb{C}^m$ such that $\sum_i |x_i|^2 = 1$. We follow the Dirac bra/ket notation on quantum states, i.e., we denote the quantum state for $\mathbf{x}$ by $|\mathbf{x}\rangle$ and denote $\mathbf{x}^\dagger$ by $\langle\mathbf{x}|$ , where $\dagger$ means the Hermitian conjugation. Given a state $|\mathbf{x}\rangle = \sum_{i=1}^{m} x_i|i\rangle$, we call $x_i$ the amplitude of the state $|i\rangle$. Given two quantum states $|\mathbf{x}\rangle \in \mathbb{C}^m$ and $|\mathbf{y}\rangle \in \mathbb{C}^m$, we denote their inner product by $\langle\mathbf{x}|\mathbf{y}\rangle := \sum_i x_i^\dagger y_i$. Given $|\mathbf{x}\rangle \in \mathbb{C}^n$ and $|\mathbf{y}\rangle \in \mathbb{C}^m$, their tensor product is defined as $|\mathbf{x}\rangle \otimes |\mathbf{y}\rangle := (x_1y_1, \cdots, x_1y_m, \cdots, x_ny_1, \cdots, x_ny_m)^\top$. A quantum algorithm works by applying a sequence of unitary operators to an input quantum state. In many cases, the quantum algorithms may have access to input data via unitary operators called *quantum oracles*. This operator can be accessed multiple times by a quantum algorithm. Hence, the *quantum query complexity* of a quantum algorithm is defined as the number of a quantum oracle being used. See (Nielsen and Chuang 2010) for detailed introductions to quantum computing.

**Quantum Noisy Function Oracle.** Our quantum non-linear bandit optimization setting follows that of the quantum multi-armed bandits in (Wan et al. 2023). In the quantum realm, each input $\mathbf{x}$ is associated with a quantum sampling oracle. This oracle follows quantum sampling oracle (see appendix) and encodes the distribution of the corresponding noisy function value. More formally, let $Y_{\mathbf{x}}$ be the random variable of the noisy function value with input $\mathbf{x}$, and let $\Omega_{\mathbf{x}}$ be the finite sample space of this distribution. Then the sampling oracle for the noisy function value with input $\mathbf{x}$ is defined as:

$$\mathcal{O}_{\mathbf{x}} : |\mathbf{0}\rangle \to \sum_{y\in\Omega_{\mathbf{x}}} \sqrt{\Pr[Y_{\mathbf{x}} = y]}\, |y\rangle \otimes |\psi_y\rangle, \qquad (1)$$

where $|\psi_y\rangle$ is an arbitrary quantum state for each $y$. More detailed discussions on the justification of quantum oracles' feasibility and the relationships between them and their classical counterparts can be found in appendix.

**Quantum Mean Estimation.** For estimating the mean of an unknown distribution, we will use the following quantum Monte Carlo mean estimator as in (Wan et al. 2023; Wu et al. 2023; Dai et al. 2023):

**Lemma 1** (Quantum Monte Carlo mean estimator (Montanaro 2015))**.** *Given the access to a quantum sampling oracle $\mathcal{O}_Y$ (and its inverse $\mathcal{O}_Y^\dagger$) that encodes the distribution of a random variable $Y$, as defined in Eq.* (1)*.*

*(1)* ***Bounded value****: If the value of $Y$ is taken from the interval $[0, 1]$, then there exists a constant $C_1 > 1$ a quantum algorithm $\mathsf{QME}_1(\mathcal{O}_Y, \epsilon, \delta)$ which returns an estimate $\hat{y}$ such that with probability at least $1 - \delta$, $|\hat{y} - \mathbf{E}[Y]| \leq \epsilon$, using at most $\frac{C_1}{\epsilon}\log(1/\delta)$ queries to $\mathcal{O}_Y$ and its inverse.*

*(2)* ***Bounded variance****: If $\mathrm{Var}[Y] \leq \sigma^2$, then for $\epsilon < 4\sigma$, there is a constant $C_2 > 1$ and a quantum algorithm $\mathsf{QME}_2(\mathcal{O}_Y, \epsilon, \delta)$ which returns an estimate $\hat{y}$ such that with probability at least $1 - \delta$, $|\hat{y} - \mathbf{E}[Y]| \leq \epsilon$, using at most $\frac{C_2\sigma}{\epsilon}\log^{3/2}(8\sigma/\epsilon)\log(\log(8\sigma/\epsilon))\log(1/\delta)$ queries to $\mathcal{O}_Y$ and its inverse.*

As briefly discussed in (Wan et al. 2023), when aiming for a mean estimation error of $\epsilon$, the QME algorithm achieves a quadratic reduction in query complexity compared to the classical one, which is crucial for the quantum speed-up in (Wan et al. 2023; Wu et al. 2023; Dai et al. 2023).

## Assumptions

**Assumption 2** (Realizable parametric function class)**.** There exists an optimal $\mathbf{w}^* \in \mathcal{W}$ such that $f_0 = f_{\mathbf{w}^*}$. Also, w.l.o.g.,

$\mathcal{W} \subseteq [0,1]^{d_w}$.

This assumption is commonly used in bandits (Foster and Rakhlin 2020; Simchi-Levi and Xu 2022) and reinforcement learning (Zhan et al. 2022; Zanette 2023). In Bayesian optimization (Srinivas et al. 2010; Chowdhury and Gopalan 2017), the RKHS assumption essentially assumes that the objective function is realizable within a certain RKHS function class. The realizable assumption allows one not to handle the function in misspecified settings (Bogunovic and Krause 2021; Liu, Yin, and Wang 2023), which is beyond the scope of this paper. The second assumption is on the structure of parametric function $f_{\mathbf{w}}$.

**Assumption 3** (Bounded, differentiable, and smooth function)**.** There exist constants $C_f, C_g, C_h > 0$ such that $\forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{w} \in \mathcal{W}$, it holds that $|f_{\mathbf{x}}(\mathbf{w})| \leq C_f, \|\nabla f_{\mathbf{x}}(\mathbf{w})\|_2 \leq C_g, \|\nabla^2 f_{\mathbf{x}}(\mathbf{w})\|_{\text{op}} \leq C_h$.

This is a common assumption used in many non-convex optimization works (Kohler and Lucchi 2017; Li et al. 2023). Note it only puts mild structure conditions on the smoothness of parametric function $f_{\mathbf{w}}$ w.r.t. its parameter $\mathbf{w}$, rather than input $\mathbf{x}$, and the objective function $f_0$ can still be a black-box function of $\mathbf{x}$. The last assumption is on the expected loss function over uniform distribution $\mathcal{U}$.

**Assumption 4** (Geometric conditions of loss function (Liu and Wang 2023))**.** $L(\mathbf{w}) = \mathbf{E}_{\mathbf{x}\sim\mathcal{U}}(f_{\mathbf{x}}(\mathbf{w}) - f_{\mathbf{x}}(\mathbf{w}^*))^2$ satisfies $(\tau,\gamma)$-growth condition or $\mu$-local strong convexity at $\mathbf{w}^*$, i.e., $\forall \mathbf{w} \in \mathcal{W}$,

$$\min\left\{\frac{\mu}{2}\|\mathbf{w}-\mathbf{w}^*\|_2^2, \frac{\tau}{2}\|\mathbf{w}-\mathbf{w}^*\|_2^\gamma\right\} \leq L(\mathbf{w}) - L(\mathbf{w}^*),$$

for constants $\mu, \tau > 0, \mu < d_w$ and $0 < \gamma < 2$. Also, $L(\mathbf{w})$ satisfies a $c$-local self-concordance assumption at $\mathbf{w}^*$.

This assumption is needed for technical reasons in analyzing quantum regression oracle. Note the loss function $L(\mathbf{w})$ can be a highly non-convex function since it only assumes *local* strong convexity in the neighboring region of $\mathbf{w}^*$, strictly weaker than the global strong convexity, and growth condition when $\mathbf{w}$ is away from $\mathbf{w}^*$. Careful readers are referred to Figure 1 in (Liu and Wang 2023) for a non-convex function example satisfying this assumption.

## Q-NLB-UCB Algorithm

In this section, we show full details of the Q-NLB-UCB algorithm (Algorithm 1). First, in Step 1, we take a subroutine Algorithm 2 to query the quantum regression oracle QNLRO for $T_0$ times, which aims at solving the following non-linear regression problem to get an estimated parameter $\hat{\mathbf{w}}_0 \leftarrow \arg\min_{\mathbf{w}\in\mathcal{W}} \sum_{j=1}^{T_0}(f_0(\mathbf{x}_j) - y_j)^2$. Our goal is to make sure that $\hat{\mathbf{w}}_0$ satisfies

$$\|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2 \leq \frac{C_0}{T_0}, \tag{2}$$

where $C_0$ denotes a constant. Careful readers may have noticed that in the classical (non-quantum) regime, the best upper bound is only $\|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2 \leq O(1/\sqrt{T_0})$, which can be obtained using the small variance property of squared losses near optimal solution and applying the Craig-Bernstein (CB) inequality (Craig 1933). How can we achieve the quadratic improvement in quantum case? In short, we work with the quantum fast-forward technique (Apers and Sarlette 2019) and refine the analysis with the CB inequality. This gives the desirable convergence rate, but the parameter vector $\hat{\mathbf{w}}_0$ is returned in the form of a quantum state. To use it in classical state later, necessary techniques for retrieving the classical information of all the entries in $\hat{\mathbf{w}}_0$ will be employed. Specifically, we use *non-destructive amplitude estimation* (NDAE) (Rall and Fuller 2023). More details on techniques and analyses are given in the next section and full proofs are shown in appendix.

**Algorithm 1** Q-NLB-UCB

**Input:** Objective function $f_0$, initial covariance matrix $\Sigma_0 = \lambda\mathbf{I}$, quantum non-linear regression oracle QNLRO, regularization weight $\lambda$, confidence sequence $\beta_s$, constant $C_1$.

1: $\hat{\mathbf{w}}_0 \leftarrow \mathsf{QNLRO}(f_0, T_0, \delta/4)$
2: **for** each stage $s = 1, 2, \cdots$ **do**
3: Update $\Sigma_s$ by Eq. (3).
4: Update $\hat{\mathbf{w}}_s$ by Eq. (4).
5: Update $\text{Ball}_s$ by Eq. (5).
6: Select $\mathbf{x}_s = \arg\max_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{w}\in\text{Ball}_s} f_{\mathbf{x}}(\mathbf{w})$.
7: Update $\epsilon_s = \|\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\|_{\Sigma_s^{-1}}$.
8: **for** the next $\frac{C_1}{\epsilon_s}\log\frac{m}{\delta}$ rounds **do**
9: Take actions $\mathbf{x}_s$ and run $\mathsf{QME}_1(O_{\mathbf{x}_s}, \epsilon_s, \delta/m)$.
10: Obtain $y_s$ as an estimation of $f_0(\mathbf{x}_s)$.
11: **end for**
12: **end for**

**Output:** $\hat{\mathbf{x}} \sim \mathcal{U}(\{\mathbf{x}_1, \cdots, \mathbf{x}_T\})$.

**Algorithm 2** Quantum Non-Linear Regression Oracle (QNLRO)

**Input:** Objective function $f_0$, time $T_0$, failure parameter $\delta \in [0, 1/2]$, quantum regression oracle Oracle.

1: $|\hat{\mathbf{w}}_0\rangle \leftarrow \mathsf{Oracle}(f_0, T_0, \delta/4)$
2: **for** $i = 1, \cdots, d_w$ **do**
3: set projector $P_i = |i\rangle\langle i|$
4: obtain $\tilde{a}_i \leftarrow \mathsf{NDAE}(|\hat{\mathbf{w}}_0\rangle, P_i, \frac{1}{d_w \cdot T_0^2}, \frac{\delta}{4d_{\mathbf{w}}})$
5: **end for**

**Output:** $\hat{\mathbf{w}}_0 = (\tilde{a}_1, \cdots, \tilde{a}_{d_w})$.

From Step 2 to Step 12, the algorithm runs in $m$ stages where multiple rounds are conducted in each stage. Why doesn't it run in simply $T$ rounds like classical bandit optimization? This design is due to the quantum Monte Carlo mean estimation (Lemma 1) where the same action needs to be taken multiple times. And we set $\epsilon_s = \|\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\|_{\Sigma_s^{-1}}$ and $m = d_w \log\left(\frac{C_g^2 T^2}{d_w \lambda} + 1\right)$ (Lemma 13) to ensure the total number of rounds is $T$. In addition to using $\mathsf{QME}_1$ in Algorithm 1, our algorithm also works with $\mathsf{QME}_2$ which is the bounded variance case in Lemma 1. The analysis will be similar to that of $\mathsf{QME}_1$.

Specifically in each stage $s = 1, ..., m$, in Step 3, we construct the covariance matrix $\Sigma_s$ by

$$\Sigma_s = \Sigma_{s-1} + \frac{1}{\epsilon_{s-1}^2} \nabla f_{\mathbf{x}_{s-1}}(\hat{\mathbf{w}}_0) \nabla f_{\mathbf{x}_{s-1}}(\hat{\mathbf{w}}_0)^\top, \quad (3)$$

where the $1/\epsilon_{s-1}^2$ is the weight assigned to query in each stage. Note here $\nabla f_{\mathbf{x}_{s-1}}(\hat{\mathbf{w}}_0)$ is the gradient of the parametric function $f$ taken w.r.t. $\hat{\mathbf{w}}_0$, which can be easily obtained, and the objective function $f_0$ can still be a black-box function without any derivative information. Different from (Liu and Wang 2023), $\nabla f_{\mathbf{x}_{s-1}}(\hat{\mathbf{w}}_0)$ is not taken w.r.t. the fixed $\hat{\mathbf{w}}_{s-1}$ to save the tedious inductive argument in it while still doing the rank-1 updates since the action $\mathbf{x}_{s-1}$ changes over stages. And the rank-1 updates are needed because the algorithm can save all historical information and add only one new matrix at each stage, according to Eq. (3). Then we define the following regression problem to estimate $\hat{\mathbf{w}}_s$:

$$\hat{\mathbf{w}}_s = \arg\min_{\mathbf{w}} \frac{\lambda}{2} \|\mathbf{w} - \hat{\mathbf{w}}_0\|_2^2 + \frac{1}{2} \sum_{i=0}^{s-1} \frac{1}{\epsilon_i^2} \left( (\mathbf{w} - \hat{\mathbf{w}}_0)^\top \nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0) + f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0) - y_i \right)^2. \quad (4)$$

Note in the first term $\|\mathbf{w} - \hat{\mathbf{w}}_0\|_2^2$, the regression center is set to be $\hat{\mathbf{w}}_0$ so that we can take advantage of Eq. (2) to reach a much faster convergence rate than constant as in QLinUCB (Wan et al. 2023). The design of the second term is using the first order Taylor expansion of parametric function $f_{\mathbf{x}_i}$ to approximate the noisy observation $y_i$. Solution to optimization problem in Eq. (4), $\hat{\mathbf{w}}_s$, further serves as the center of the parameter uncertainty region $\mathrm{Ball}_s$, defined as

$$\mathrm{Ball}_s = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w} - \hat{\mathbf{w}}_s\|_{\Sigma_s}^2 \leq \beta_s\}. \quad (5)$$

The key design of $\mathrm{Ball}_s$ is to contain the optimal parameter $\mathbf{w}^*$ in each stage $s$ w.h.p., so that Q-NLB-UCB can keep track of $\mathbf{w}^*$ at all times. The radius parameter $\beta_s$ plays an important role in this design and later in Lemma 14 our confidence analysis shows that it suffices to choose $\beta_s$ as

$$\beta_s = 3d_w s + \frac{3\lambda C_0^2}{T_0^2} + \frac{3C_h^2 C_0^2 s T^2}{4T_0^4}. \quad (6)$$

In Step 6, the choice of $\mathbf{x}_s$ is generated by solving a cross optimization problem defined in both $\mathcal{X}$ and $\mathrm{Ball}_s$. While the exact solution of this step is hard to find, in practice one can use gradient ascent as a surrogate solution, and it works well in our experiments. The final output $\hat{\mathbf{x}}$ is uniformly sampled from all historical actions $\mathbf{x}_1, ..., \mathbf{x}_T$ since $f^* - \mathbf{E}f(\hat{\mathbf{x}}) \leq R_T/T$, i.e., one can easily obtain the theoretical guarantee of $\hat{\mathbf{x}}$ using cumulative regret bound with uniform sampling. But in practice, one can also select $\mathbf{x}_T$ as the output.

## Theoretical Analysis

In this section, we provide theoretical analysis for Q-NLB-UCB. First, we analyze the quantum regression oracle that outputs $\hat{\mathbf{w}}_0$ to start the algorithm, then we provide the regret analysis to prove its input dimension-free $O(\mathrm{poly}\log T)$ regret bound, supported by the confidence analysis. Full proofs and time complexity analysis are deferred to appendix.

### Quantum Non-Linear Regression Oracle

Here we show the existence of a quantum non-linear regression oracle which outputs the estimated parameter $\hat{\mathbf{w}}_0$ that satisfies Eq. (2). This algorithm involves two primary procedures: (1) obtaining a quantum state that encodes $\hat{\mathbf{w}}_0$; (2) retrieving the classical information of all entries in $\hat{\mathbf{w}}_0$. Roughly speaking, this result can be achieved by "quantizing" the proof of Theorem 5.2 in (Liu and Wang 2023) with quantum fast-forward technique (Apers and Sarlette 2019), shown below.

**Lemma 5** (Adapted from Theorem 5.2 in (Liu and Wang 2023))**.** *Suppose Assumptions 2, 3, and 4 hold. There is an absolute value $C$ such that after $T_0$ iterations in step 1 of Algorithm 1 where $T_0$ satisfies $T_0 \geq C d_{\mathbf{w}} \iota \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{2C_g^2}{\mu c^2}\right\}$, with probability $1 - \delta/2$, the quantum regression oracle returns an estimate $\hat{\mathbf{w}}_0$ that satisfies $\|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2 \leq \sqrt{\frac{C d_{\mathbf{w}} \iota}{T_0}}$, where $\iota$ is the logarithmic term depending on $T_0, C_h, 1/\delta$.*

Note that this lemma still gives $O(1/\sqrt{T_0})$ asymptotical rate after $T_0$ iterations, the same as in classical setting. To attain the desired faster $O(1/T_0)$ rate, we apply the quantum fast-forward technique introduced in (Apers and Sarlette 2019). An informal description below follows that in (Ambainis et al. 2020).

**Lemma 6** (Informal statement of quantum fast-forward (Ambainis et al. 2020))**.** *Let $\epsilon \in (0, 1), s \in [0, 1]$ and $t \in \mathbf{N}$. Given a reversible Markov chain determined by a matrix $\mathbf{D}$, there is a quantum algorithm with $O(\sqrt{t \log(1/\epsilon)})$ quantum walk steps that takes input $|\overline{0}\rangle|\psi\rangle \in \mathrm{span}\{|\overline{0}\rangle|x\rangle : x \in X\}$, and outputs a state that is $\epsilon$-close to a state of the form $|0\rangle^{\otimes a}|\overline{0}\rangle D^t|\psi\rangle + |\Gamma\rangle$ where $a = O(\log(t \log(1/\epsilon)))$, $|\overline{0}\rangle$ is some fixed reference state, and $|\Gamma\rangle$ is some garbage state that has no support on states containing $|0\rangle^{\otimes a}|\overline{0}\rangle$ in the first two registers.*

Consequently, we can summarize the goal of Step 1 in Algorithm 1 with the following theorem. Full proofs are deferred to appendix.

**Theorem 7.** *Suppose Assumptions 2, 3, and 4 hold. There is an absolute value $C$ such that after $\tilde{O}(T_0)$ iterations in step 1 of Algorithm 2 where $T_0$ satisfies $T_0^2 \geq C d_{\mathbf{w}} \iota \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{2C_g^2}{\mu c^2}\right\}$ with probability $1 - \delta/2$, the quantum regression oracle returns a quantum state that encodes an estimate $\hat{\mathbf{w}}_0$ that satisfies $\|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2 \leq \frac{\sqrt{C d_{\mathbf{w}} \iota}}{T_0}$, where $\iota$ is the logarithmic term depending on $T_0, C_h, 1/\delta$.*

So far we have proved the convergence of information provided by $|\hat{\mathbf{w}}_0\rangle$. This means that after applying the quantum regression oracle, we will be returned with a quantum state $|\hat{\mathbf{w}}_0\rangle$ which encodes the parameter vector $\hat{\mathbf{w}}_0$ that meets the convergence guarantee. Next for the classical usage of parameter vector $\hat{\mathbf{w}}_0$, we need to retrieve them from the quantum state $|\hat{\mathbf{w}}_0\rangle$. A standard approach will be employing quantum state tomography. But this will be an overkill because in general quantum state tomography considers the cases where the $\log N$-qubit quantum state input to be in $\mathbf{C}^N$. Recall that with loss of generality, our work focuses on $|\hat{\mathbf{w}}_0\rangle \in [0, 1]^{d_w}$.

Hence, we can employ a more efficient tool, named *quantum amplitude estimation*, which outputs an estimate of $\langle\psi|P|\psi\rangle$ upon input a quantum state $|\psi\rangle$ and a projector $P$. However, the execution of straight quantum amplitude estimation algorithm will cause the input state $|\psi\rangle$ to collapse, which means $O(\mathrm{poly}(d_w, \epsilon, \delta))$ many copies of the input quantum states are needed for gaining the classical information of all the entries in $|\psi\rangle$. In our problem, this preparation of multiple copies of the state $|\hat{\mathbf{w}}_0\rangle$ will require queries to the quantum sampling oracles $\mathcal{O}_{\mathbf{x}}$, thus dramatically increasing the cumulative regret, which is undesirable. That means that our problem lies in scenarios where $|\psi\rangle$ is extremely expensive to prepare. To avoid this, we turn to *non-destructive amplitude estimation* which can return an estimation of $\langle\psi|P|\psi\rangle$ and also give the copy of $|\psi\rangle$ back. There are multiple existing works in this field. (Rall and Fuller 2023) listed multiple of them and we pick one among them. An informal description is stated below.

**Theorem 8** ((Informal) Non-destructive amplitude estimation in (Rall and Fuller 2023))**.** *Given one copy of a quantum state $|\psi\rangle$, a projector $P$, and $\epsilon, \delta \in [0, 1/2]$. Let $a = \langle\psi|P|\psi\rangle$. Then there is an algorithm $\mathsf{NDAE}(|\psi\rangle, P, \epsilon, \delta)$ that with probability at least $1-\delta$, outputs an estimate $\tilde{a}$ and a copy of $|\psi\rangle$ such that $|\tilde{a} - a| \leq \epsilon$.*

Let Oracle denote a quantum algorithm that solves the non-linear regression problem and outputs a quantum state encoding the solution that satisfies the convergence bound provided by Theorem 7. Combining Theorem 7 and Theorem 8, we obtain the following result for the quantum non-linear regression oracle (QNLRO, Algorithm 2).

**Theorem 9.** *Suppose Assumptions 2, 3, and 4 hold. Then Algorithm 2 returns with probability $1 - \delta/2$, a classical vector of an estimate $\hat{\mathbf{w}}_0$ that satisfies $\|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2 \leq \frac{\sqrt{C d_{\mathbf{w}} \iota}}{T_0}$, where $\iota$ is the logarithmic term depending on $T_0, C_h, 1/\delta$ and $T_0$ satisfies $T_0^2 \geq C d_{\mathbf{w}} \iota \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{2C_g^2}{\mu c^2}\right\}$.*

The above theorem implies that it is sufficient to use only one copy of $|\hat{\mathbf{w}}_0\rangle$ to extract the classical information of all of it's entries, given the fact that all the entries belong to $[0, 1]$. More concretely, Algorithm 2 first calls a quantum algorithm Oracle to obtain a quantum state that has the solution parameter vector. Then it uses NDAE with projectors that project onto each computational basis, i.e. $\langle\hat{\mathbf{w}}_0|i\rangle\langle i|\hat{\mathbf{w}}_0\rangle$, to retrieve each entry, respectively. Note that each time NDAE returns not only an estimate of an entry but also a copy of the original $|\hat{\mathbf{w}}_0\rangle$. Additionally, since NDAE doesn't query the sampling oracle $\mathcal{O}_{\mathbf{x}}$, its execution incurs no extra cumulative regret.

## Regret Analysis

Now we present the cumulative regret bound of our proposed Q-NLB-UCB algorithm.

**Theorem 10** (Cumulative regret bound of Q-NLB-UCB)**.** *Suppose Assumptions 2, 3, and 4 hold. There is an absolute value $C$ such that after $\tilde{O}(T_0)$ iterations in Step 1 of Algorithm 1 where $T_0$ satisfies $T_0^2 \geq C d_{\mathbf{w}} \iota \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{2C_g^2}{\mu c^2}\right\}$ with $\iota$ denoting a logarithmic term depending on $T_0, C_h, 1/\delta$. Then Algorithm 1 with parameters $T_0 = \sqrt{T}, \lambda = T$ satisfies that with probability at least $1-\delta$, $R_T = O\big(d_w^2 \log^{\frac{3}{2}}(T) \log(d_w \log(T))\big)$.*

*Remark* 11. Note $d_w$ is the parameter complexity of $f_{\mathbf{w}}$, which is *not* necessarily related to the input dimension $d_x$ of $f_0$, therefore, our regret bound is input dimension-free. In practice, $d_w$ can be smaller or larger than $d_x$ since $d_w$ solely depends on the users' choice of parametric functions, which can be linear or quadratic functions, or even deep neural networks. When $f_{\mathbf{w}}$ is chosen to be a linear function, our algorithm reduces back to the linear bandits as in QLinUCB (Wan et al. 2023). Compared with Q-GP-UCB (Dai et al. 2023) and QMCKernelUCB (Hikima et al. 2024), our algorithm takes a different technical route to successfully avoid the curse of dimensionality limitation. Moreover, our bound is at the $\log^{\frac{3}{2}}(T) \log\log T$ rate, which is also faster than classical lower bound $\Omega(\sqrt{T})$, showing the power of quantum computing.

*Remark* 12. The choices of $T_0 = \sqrt{T}, \lambda = T$ require careful analyses among quantum regression oracle, regret analysis, and confidence analysis. $T_0$ cannot be chosen too large to enforce a large $T$ and it cannot be chosen too small to break the property of $\hat{\mathbf{w}}_0$ in Eq. (2). The choice of $\lambda$ is obtained by balancing between different terms in regret analysis to ensure a $\mathrm{poly}\log T$-style bound.

**Proof Sketch.** The proof starts from the instantaneous regret in a single round in one stage. Since we are dealing with non-linear bandit optimization where the objective function is not necessarily linear, we use Taylor's theorem to expand the objective function into first order terms and high-order terms. The first order terms are handled like linear bandits (Wan et al. 2023), but the remaining high-order terms are bounded creatively using the convex property of parameter uncertainty region $\mathrm{Ball}_s$ multiple times. After we obtain the upper bound for a single round in one stage, we multiply it by number of rounds in one stage and obtain the bound for one stage. Later, we prove the following lemma to show the total number of stages in Q-NLB-UCB to make sure the total number of rounds is $T$.

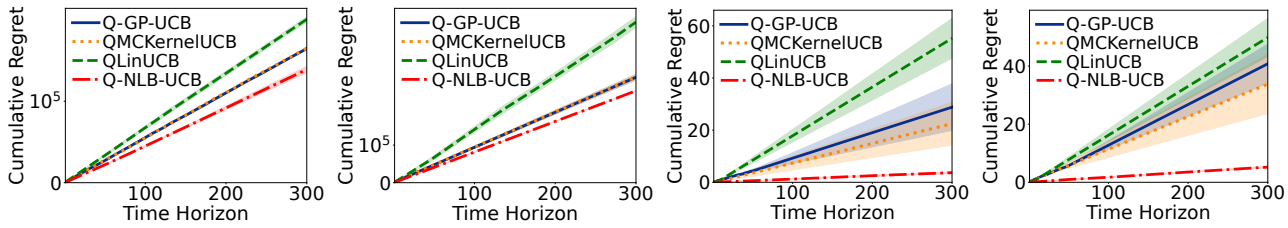
**Lemma 13** (Number of stages)**.** *The Algorithm 1 runs at most $m = d_w \log\left(\frac{C_g^2 T^2}{d_w \lambda} + 1\right)$ stages.*

Note again in this lemma, $d_w$ is the dimension of parameters, rather than $d_x$. This is because we are using $\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)$ as the feature vector and its dimension is $d_w$. After proving Lemma 13, we take the summation of upper bounds and reach the upper bound of cumulative regret.

## Confidence Analysis

The previous regret builds upon the successful construction of the confidence ball $\mathrm{Ball}_s$ for each stage $s$, which is summarized in the following confidence analysis lemma.

**Lemma 14** (Confidence bound of Q-NLB-UCB)**.** *Suppose Assumptions 2, 3, and 4 hold and $\beta_s$ is chosen as Eq.* (6)*. Then with parameters $T_0 = \sqrt{T}, \lambda = T$ in each stage $s$ in Algorithm 1, the optimal parameter $\mathbf{w}^*$ is trapped in*




(a) 30D Rastrigin (b) 30D Styblinski-Tang (c) AutoML (Cancer) (d) AutoML (Diabetes)

Figure 1: Cumulative regrets (the lower the better) of all compared quantum bandit algorithms.

*confidence ball* $\mathrm{Ball}_s$ *with probability at least* $1-\delta$*, i.e.,* $\|\hat{\mathbf{w}}_s - \mathbf{w}^*\|^2_{\Sigma_s} \leq \beta_s$.

*Remark* 15. The design of $\mathrm{Ball}_s$ is similar to that in LinUCB (Abbasi-Yadkori, Pál, and Szepesvári 2011) and QLinUCB (Wan et al. 2023), but our choice of $\beta_s$ is different. $\beta_s$ cannot be too small to lose the track of optimal parameter $\mathbf{w}^*$ and it cannot be too large as it appears in final regret bound. Overall the confidence analysis ensures that $\beta_s = \tilde{O}(1)$ and slightly grows as the stage index $s$.

## Experiments

**Experimental Setup.** We compare the performance of our Q-NLB-UCB algorithm with QLinUCB (Wan et al. 2023), Q-GP-UCB (Dai et al. 2023) and QMCKernelUCB (Hikima et al. 2024). According to (Hikima et al. 2024), the only difference between Q-GP-UCB and QMCKernelUCB is a trade-off parameter $\eta$ balancing the precision of quantum amplitude estimation and the noise in observations, and when $\eta = 1$, they become the same. In our experiments, we set $\eta = 0.1$ for QMCKernelUCB. To run Q-NLB-UCB, we set our parametric function model $f_{\mathbf{w}}$ to be a two linear layer neural network with the sigmoid activation function. Our implementation is based upon the sklearn, BoTorch (Balandat et al. 2020), and Qiskit (Javadi-Abhari et al. 2024). To stay consistent with theoretical analysis, cumulative regret is also used to evaluate algorithms. In order to reduce the impact of randomness in algorithms, we repeat each experiment 5 times and report the mean and adjusted standard errors of cumulative regrets, i.e., $\mathrm{mean} \pm \mathrm{std}/\sqrt{5}$.

**High-Dimensional Synthetic Functions.** We test all four algorithms on two functions, Rastrigin function and Styblinski-Tang function, defined in 30-dimensional space $[-5, 5]^{30}$. Q-NLB-UCB performs well as evidenced in Figure 1(a)(b), where it consistently achieves the lowest cumulative regret across multiple runs, outperforming all other algorithms. It is not a surprise as Q-GP-UCB and QMCKernelUCB suffer from the curse of dimensionality and Q-GP-UCB was only tested in 1-d and 2-d settings in (Dai et al. 2023). QLinUCB performs the worst since it is designed to work on linear functions only while both two test functions here are highly non-linear.

In addition, we report the runtime of three quantum bandit algorithms in seconds shown in Table 1. Among the three algorithms, our Q-NLB-UCB algorithm achieves a significantly low runtime, which again shows the efficiency of Q-NLB-UCB and validates our theoretical time complexity analysis in appendix. We don't list the runtime of QLinUCB because it is a linear bandit algorithm running faster but not comparable to algorithms designed for non-linear optimization.

| Algorithms | Rastrigin | Styblinski-Tang |
|---|---|---|
| Q-GP-UCB | 4629.6453 | 4139.2478 |
| QMCKernelUCB | 3744.1179 | 2565.0690 |
| Q-NLB-UCB (ours) | 861.2402 | 919.7602 |

Table 1: Runtime (in seconds) on two synthetic functions

**Real-World AutoML Tasks.** We test all four quantum bandit algorithms on three different hyperparameter tuning tasks for Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and Gradient Boosting (GB). We are tuning 4 hyperparameters in SVM, 8 in MLP, and 11 for GB. Each classifier is trained by different hyperparameter configurations and the goal is to maximize the validation accuracy on a hold-out set. Due to page limit, we only show results of MLP in Figure 1(c)(d) on both breast cancer and diabetes datasets, and readers are referred to appendix for similar results of SVM and GB. Again our Q-NLB-UCB algorithm outperforms all other algorithms by achieving significantly smaller regrets, demonstrating its strong potential for practical applications.

## Conclusion

With the aid of quantum computing, recent works (Dai et al. 2023; Hikima et al. 2024) showed that new $O(\mathrm{poly}\log T)$ regret bound can be achieved in quantum non-linear bandit optimization, but their works heavily rely on the RKHS assumption which suffers from the curse of dimensionality. Real-world data usually sit in high-dimensional spaces, making their regret bounds vacuous. In this paper, we develop the new Q-NLB-UCB algorithm which efficiently solves the problem in high-dimensional cases. The key design of Q-NLB-UCB involves quantum Monte Carlo mean estimation, parametric function approximation, and quantum fast-forward techniques, which all contribute to the new *input dimension-free* regret bound of Q-NLB-UCB. Moreover, the choice of parametric functions can be generic, such as linear or quadratic functions, or even deep neural networks. Technically, our analysis of the new quantum non-linear regression oracle can be of independent interests in more quantum machine learning problems in the future.

## Acknowledgments

The authors would like to thank the anonymous reviewers for helpful comments that improved the final version of this paper.

## References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. *NIPS*, 24.

Allen-Zhu, Z. 2018. Katyusha: The first direct acceleration of stochastic gradient methods. *JMLR*, 18(221): 1–51.

Ambainis, A.; Gilyén, A.; Jeffery, S.; and Kokainis, M. 2020. Quadratic speedup for finding marked vertices by quantum walks. In *STOC*.

Apeldoorn, J. v.; and Gilyén, A. 2018. Improvements in Quantum SDP-Solving with Applications. In *ICALP*.

Apers, S.; and Sarlette, A. 2019. Quantum fast-forwarding: Markov chains and graph property testing. *Quantum Information and Computation*, 19(3–4): 181–213.

Balandat, M.; Karrer, B.; Jiang, D.; Daulton, S.; Letham, B.; Wilson, A. G.; and Bakshy, E. 2020. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *NeurIPS*, 33.

Bogunovic, I.; and Krause, A. 2021. Misspecified gaussian process bandit optimization. *NeurIPS*, 34.

Brandão, F. G. S. L.; and Svore, K. M. 2017. Quantum Speed-Ups for Solving Semidefinite Programs. In *FOCS*.

Casares, P. A. M.; and Martin-Delgado, M. A. 2020. A quantum interior-point predictor–corrector algorithm for linear programming. *Journal of Physics A: Mathematical and Theoretical*, 53(44): 445305.

Chakrabarti, S.; Childs, A. M.; Li, T.; and Wu, X. 2020. Quantum algorithms and lower bounds for convex optimization. *Quantum*, 4: 221.

Chen, Z.; Lu, Y.; Wang, H.; Liu, Y.; and Li, T. 2025. Quantum Langevin Dynamics for Optimization. *Communications in Mathematical Physics*, 406(3): 52.

Childs, A. M.; Leng, J.; Li, T.; Liu, J.-P.; and Zhang, C. 2022. Quantum simulation of real-space dynamics. *Quantum*, 6: 860.

Chowdhury, S. R.; and Gopalan, A. 2017. On kernelized multi-armed bandits. In *ICML*.

Clarke, R.; Ressom, H. W.; Wang, A.; Xuan, J.; Liu, M. C.; Gehan, E. A.; and Wang, Y. 2008. The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nature Reviews Cancer*, 8(1): 37–49.

Craig, C. C. 1933. On the Tchebychef inequality of Bernstein. *The Annals of Mathematical Statistics*, 4(2): 94–102.

Dai, Z.; Lau, G. K. R.; Verma, A.; Shu, Y.; Low, B. K. H.; and Jaillet, P. 2023. Quantum bayesian optimization. *NeurIPS*, 36.

Dai, Z.; Shu, Y.; Low, B. K. H.; and Jaillet, P. 2022. Sample-then-optimize batch neural Thompson sampling. *NeurIPS*, 35.

Diaconis, P.; and Miclo, L. 2013. On the spectral analysis of second-order Markov chains. *Annales de la Faculté des sciences de Toulouse: Mathématiques*, 22(3): 573–621.

Fang, C.; Li, C. J.; Lin, Z.; and Zhang, T. 2018. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *NIPS*, 31.

Foster, D.; and Rakhlin, A. 2020. Beyond ucb: Optimal and efficient contextual bandits with regression oracles. In *ICML*.

Frazier, P. I. 2018. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Frazier, P. I.; and Wang, J. 2016. Bayesian optimization for materials design. *Information science for materials discovery and design*, 45–75.

Garg, A.; Kothari, R.; Netrapalli, P.; and Sherif, S. 2021a. Near-optimal lower bounds for convex optimization for all orders of smoothness. *NeurIPS*, 34.

Garg, A.; Kothari, R.; Netrapalli, P.; and Sherif, S. 2021b. No Quantum Speedup over Gradient Descent for Non-Smooth Convex Optimization. In *Innovations in Theoretical Computer Science Conference*.

Gong, W.; Zhang, C.; and Li, T. 2025. Robustness of Quantum Algorithms for Nonconvex Optimization. In *ICLR*.

He, J.; Liu, C.; Liu, X.; Li, L.; and Lui, J. C. 2024. Quantum algorithm for online exp-concave optimization. In *Proceedings of the 41st International Conference on Machine Learning*, 17946–17971.

He, J.; Yang, F.; Zhang, J.; and Li, L. 2022. Quantum algorithm for online convex optimization. *Quantum Science and Technology*, 7(2): 025022.

Hikima, Y.; Murao, K.; Takemori, S.; and Umeda, Y. 2024. Quantum Kernelized Bandits. In *UAI*.

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.; et al. 2024. Quantum computing with Qiskit. *arXiv preprint arXiv:2405.08810*.

Jones, D. R.; Schonlau, M.; and Welch, W. J. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13: 455–492.

Kandasamy, K.; Neiswanger, W.; Schneider, J.; Poczos, B.; and Xing, E. P. 2018. Neural architecture search with bayesian optimisation and optimal transport. *NIPS*, 31.

Kerenidis, I.; and Prakash, A. 2020. A quantum interior point method for LPs and SDPs. *ACM Transactions on Quantum Computing*, 1(1): 1–32.

Kohler, J. M.; and Lucchi, A. 2017. Sub-sampled cubic regularization for non-convex optimization. In *ICML*.

Korovina, K.; Xu, S.; Kandasamy, K.; Neiswanger, W.; Poczos, B.; Schneider, J.; and Xing, E. 2020. Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations. In *AISTATS*.

Leng, J.; Hickman, E.; Li, J.; and Wu, X. 2023. Quantum hamiltonian descent. *arXiv preprint arXiv:2303.01471*.

Li, H.; Qian, J.; Tian, Y.; Rakhlin, A.; and Jadbabaie, A. 2023. Convex and non-convex optimization under generalized smoothness. *NeurIPS*, 36.

Li, T.; and Zhang, R. 2022. Quantum speedups of optimizing approximately convex functions with applications to logarithmic regret stochastic convex bandits. *NeurIPS*, 35.

Liu, C.; Guan, C.; He, J.; and Lui, J. C. 2024. Quantum Algorithms for Non-smooth Non-convex Optimization. *NeurIPS*, 37.

Liu, C.; and Wang, Y.-X. 2023. Global optimization with parametric function approximation. In *ICML*.

Liu, C.; Yin, M.; and Wang, Y.-X. 2023. No-regret linear bandits beyond realizability. In *UAI*.

Liu, Y.; Su, W. J.; and Li, T. 2023. On quantum speedups for nonconvex optimization via quantum tunneling walks. *Quantum*, 7: 1030.

Montanaro, A. 2015. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181): 1–20.

Nielsen, M. A.; and Chuang, I. L. 2010. *Quantum computation and quantum information*. Cambridge university press.

Nowak, R. 2009. Lecture notes: Complexity regularization for squared error loss.

Rahnenführer, J.; De Bin, R.; Benner, A.; Ambrogi, F.; Lusa, L.; Boulesteix, A.-L.; Migliavacca, E.; Binder, H.; Michiels, S.; Sauerbrei, W.; et al. 2023. Statistical analysis of high-dimensional biomedical data: a gentle introduction to analytical goals, common approaches and challenges. *BMC Medicine*, 21(1): 182.

Rall, P.; and Fuller, B. 2023. Amplitude estimation from quantum signal processing. *Quantum*, 7: 937.

Russo, D. J.; Van Roy, B.; Kazerouni, A.; Osband, I.; Wen, Z.; et al. 2018. *A tutorial on thompson sampling*. now publishers.

Scarlett, J.; Bogunovic, I.; and Cevher, V. 2017. Lower bounds on regret for noisy gaussian process bandit optimization. In *COLT*.

Scott, W.; Frazier, P.; and Powell, W. 2011. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization*, 21(3): 996–1026.

Sidford, A.; and Zhang, C. 2023. Quantum speedups for stochastic optimization. *NeurIPS*, 36.

Simchi-Levi, D.; and Xu, Y. 2022. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Mathematics of Operations Research*, 47(3): 1904–1931.

Snoek, J.; Rippel, O.; Swersky, K.; Kiros, R.; Satish, N.; Sundaram, N.; Patwary, M.; Prabhat, M.; and Adams, R. 2015. Scalable bayesian optimization using deep neural networks. In *ICML*.

Springenberg, J. T.; Klein, A.; Falkner, S.; and Hutter, F. 2016. Bayesian optimization with robust Bayesian neural networks. *NIPS*, 29.

Srinivas, N.; Krause, A.; Kakade, S.; and Seeger, M. 2010. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *ICML*.

van Apeldoorn, J.; Gilyén, A.; Gribling, S.; and de Wolf, R. 2020. Convex optimization using quantum oracles. *Quantum*, 4: 220.

Wan, Z.; Zhang, Z.; Li, T.; Zhang, J.; and Sun, X. 2023. Quantum multi-armed bandits and stochastic linear bandits enjoy logarithmic regrets. In *AAAI*.

Wang, Y.; Balakrishnan, S.; and Singh, A. 2019. Optimization of smooth functions with noisy observations: Local minimax rates. *IEEE Transactions on Information Theory*, 65(11): 7350–7366.

Williams, C. K.; and Rasmussen, C. E. 2006. *Gaussian processes for machine learning*. MIT press Cambridge, MA.

Wu, J.; Toscano-Palmerin, S.; Frazier, P. I.; and Wilson, A. G. 2020. Practical multi-fidelity Bayesian optimization for hyperparameter tuning. In *UAI*.

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

Zanette, A. 2023. When is realizability sufficient for off-policy reinforcement learning? In *ICML*.

Zhan, W.; Huang, B.; Huang, A.; Jiang, N.; and Lee, J. 2022. Offline reinforcement learning with realizability and single-policy concentrability. In *COLT*.

Zhang, C.; Leng, J.; and Li, T. 2021. Quantum algorithms for escaping from saddle points. *Quantum*, 5: 529.

Zhang, C.; and Li, T. 2023. Quantum lower bounds for finding stationary points of nonconvex functions. In *ICML*.

Zhang, W.; Zhou, D.; Li, L.; and Gu, Q. 2021. Neural Thompson Sampling. In *ICLR*.

Zhang, Y.; Zhang, C.; Fang, C.; Wang, L.; and Li, T. 2024. Quantum Algorithms and Lower Bounds for Finite-Sum Optimization. In *ICML*.

Zhou, D.; Li, L.; and Gu, Q. 2020. Neural contextual bandits with ucb-based exploration. In *ICML*.