Quantum Non-Linear Bandit Optimization

Zakaria Shams Siam ZSIAM@ALBANY.EDU
University at Albany, State University of New York

Chaowen Guan GUANCE@UCMAIL.UC.EDU
University of Cincinnati

Chong Liu CLIU24@ALBANY.EDU
University at Albany, State University of New York

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 $O(\text{poly}\log T)$ 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* $O(\text{poly}\log T)$ upper bound, making it applicable for high-dimensional tasks. Furthermore, its time complexity is rigorously demonstrated to be lower than that of existing quantum bandit optimization algorithms. 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.

1. 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 [37, 51], neural architecture search [26, 49], drug discovery [28, 41], and materials science [19, 57]. 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 [40] cannot be further improved, implying 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. [48] first studied the multi-armed bandits and linear bandits and proved that new $O(\text{poly}\log T)$ regret bound can be achieved with the aid

© Z.S. Siam, C. Guan & C. Liu.

of quantum computing. Later Q-GP-UCB [16] and QMCKernelUCB [24] studied the quantum Bayesian optimization, generalizing [48] to non-convex and non-linear settings, still achieving the $O(\text{poly}\log T)$ regret bound. However, [16, 24] 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 [12] to even millions [38]. 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 [16] become vacuous when input dimension d_x goes to millions, which further prevents their high-dimensional 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 [34]. Second, inspired by the success of parametric function approximation [14, 25, 32], 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 [4].

Contributions. 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. Different from existing work, 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. The time complexity of Q-NLB-UCB is demonstrated to be lower than that of other quantum non-linear bandit algorithms in terms of T. Experiments on high-dimensional synthetic functions and real-world tasks show that Q-NLB-UCB outperforms compared algorithms in regrets and runtime.

Related Works. Bayesian optimization [18] is one of the most popular methods to solve global optimization. Based on the Gaussian process [50] or reproducing kernel Hilbert space [11] assumption, Bayesian optimization runs in multiple rounds where in each round the learner takes an action suggested by an acquisition function. Without the classical Gaussian process assumption, [43, 44] used neural networks as the backbone surrogate models. Recent bandit work also studied global optimization with neural network approximation [15, 17, 55, 58] or generic parametric function approximation [32]. 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 [3, 5–7, 27, 45], later extending to general convex optimization [8, 46]. Recent advancements include quantum algorithms for slightly convex problems [9, 30, 56], escaping saddle points in non-convex landscapes [10, 54], and identifying global minima in specific non-convex cases [29, 33]. Alongside these algorithmic developments, quantum lower bounds have

been established for both convex [20, 21] and non-convex optimization [23, 53]. Also, stochastic quantum methods have been proposed [42], demonstrating the advantages of quantum stochastic first-order oracles for smooth objectives in low-dimensional settings. Most recently, there are efforts [31] on investigating quantum speed-up for minimizing non-smooth, non-convex objectives, which represent the most general function class.

2. Preliminaries

Problem Statement. 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. 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 setting, after taking $\mathbf{x}_t$, the function returns $y_t = f_0(\mathbf{x}_t) + \eta_t$, where η_t is the zero-mean, independent, σ-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 ϵ_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 $\mathcal{F} = \{f_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y} | \mathbf{w} \in \mathcal{W}\}$.

Quantum Computation. Our quantum non-linear bandit optimization setting follows that of the quantum multi-armed bandits in [48]. In the quantum realm, each input $\mathbf{x}$ is associated with a quantum sampling oracle. This oracle follows quantum sampling oracle (see Definition 9) 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:

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

where $|\psi_y\rangle$ is an arbitrary quantum state for each y. For estimating the mean of an unknown distribution, we will use the following quantum Monte Carlo mean estimator as in [16, 48, 52]:

Lemma 1 (Quantum Monte Carlo mean estimator [34]) *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 definition 9. If the value of Y is taken from the interval $[0, 1]$, then there exists a constant $C_1 > 1$ a quantum algorithm $\mathrm{QME}_1(\mathcal{O}_Y, \epsilon, \delta)$ 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.*

As briefly discussed in [48], when aiming for a mean estimation error of ϵ, the QME algorithm achieves a quadratic reduction in query complexity compared to the classical one, which is crucial for the quantum speed-up in [16, 48, 52].

Assumptions. Here we list all assumptions.

Assumption 2 (Realizable parametric function class) *There exists $\mathbf{w}^* \in \mathcal{W}$ such that $f_0 = f_{\mathbf{w}^*}$. Also, w.l.o.g., $\mathcal{W} \subseteq [0, 1]^{d_w}$.*

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* $|f_{\mathbf{x}}(\mathbf{w})| \leq C_f, \|\nabla f_{\mathbf{x}}(\mathbf{w})\|_2 \leq C_g, \|\nabla^2 f_{\mathbf{x}}(\mathbf{w})\|_{\text{op}} \leq C_h$.

Assumption 4 (Geometric conditions of loss function [32]) $L(\mathbf{w}) = \mathbf{E}_{\mathbf{x}\sim\mathcal{U}}(f_{\mathbf{x}}(\mathbf{w}) - f_{\mathbf{x}}(\mathbf{w}^*))^2$ *satisfies* (τ, γ)*-growth condition or* μ*-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* $\mathbf{w}^*$*.*

3. 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 Oracle 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. 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 [13]. To achieve the quadratic improvement in quantum case, we work with the quantum fast-forward technique [4] 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) [39]. From Step 2 to Step 12, the algorithm runs in m stages where multiple rounds are conducted in each stage. 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 27) to ensure the total number of rounds is T. In addition to using QME_1 in Algorithm 1, our algorithm also works with QME_2 which is the bounded variance case in Lemma 1. The analysis will be similar to that of QME_1. Specifically in each stage $s = 1, ..., m$, in Step 3, we construct the covariance matrix Σ_s by

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

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

Algorithm 1: Q-NLB-UCB

Input: Objective function f_0, initial covariance matrix $\Sigma_0 = \lambda\mathbf{I}$, quantum regression oracle Oracle, regularization weight λ, confidence sequence β_s, constant C_1.

1: $\hat{\mathbf{w}}_0 \leftarrow \text{QNLRO}(f_0, T_0, \delta/4)$
2: **for** each stage $s = 1, 2, \cdots$ **do**
3: Update Σ_s by Eq. (3).
4: Update $\hat{\mathbf{w}}_s$ by Eq. (4).
5: Update 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 $\text{QME}_1(O_{\mathbf{x}_s}, \epsilon_s, \delta/m)$.
10: Obtain y_s as an estimation of $f_0(\mathbf{x}_s)$.
11: **end for**
12: **end for**

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

function f_0 can still be a black-box function without any derivative information. Different from [32], $\nabla f_{\mathbf{x}_{s-1}}(\hat{\mathbf{w}}_0)$ 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 $\hat{\mathbf{w}}_s$:

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

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 [48]. 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 Ball_s, defined as

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

The key design of 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 β_s plays an important role in this design and later in Lemma 29 our confidence analysis shows that it suffices to choose β_s as

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

In Step 6, the choice of $\mathbf{x}_s$ is generated by solving a cross optimization problem defined in both $\mathcal{X}$ and Ball_s. While the theoretical 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 in Section 5.

4. Theoretical Analysis

First we prove the following result for the quantum regression oracle (Algorithm 2 and other details in Appendix C).

Theorem 5 *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 $\|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2 \leq \frac{\sqrt{Cd_{\mathbf{w}}\iota}}{T_0}$, where ι is the logarithmic term depending on $T_0, C_h, 1/\delta$ and T_0 satisfies $T_0^2 \geq Cd_{\mathbf{w}}\iota \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{2C_g^2}{\mu c^2}\right\}$.*

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.

Then we have our regret bound:

Theorem 6 (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 ι 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 $1-\delta$, $R_T = O\big(d_w^2 \log^{\frac{3}{2}}(T) \log(d_w \log(T))\big)$.*

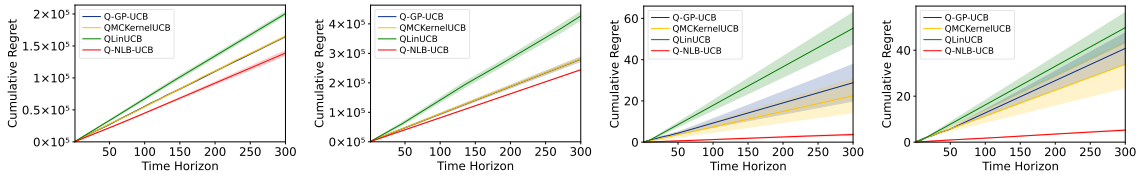
Note d_w is the parameter complexity of $f_{\mathbf{w}}$, which is *not* necessarily related to the input dimension d_x of f_0, therefore, our regret bound is input dimension-free. In practice, d_w can be smaller or larger than d_x since d_w solely depends on the users' choice of parametric functions, 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 [48]. Compared with Q-GP-UCB [16] and QMCKernelUCB [24], 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.

5. Experiments

Experimental Setup. We compare the performance of Q-NLB-UCB with QLinUCB [48], Q-GP-UCB [16] and QMCKernelUCB [24]. 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. 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., $\text{mean} \pm \text{std}/\sqrt{5}$.

High-Dimensional Synthetic Functions. We test all four algorithms on Rastrigin and Styblinski-Tang functions, defined in $[-5, 5]^{30}$. Q-NLB-UCB performs well in high-dimensional problems, as evidenced in Figure 1(a)(b), where it consistently achieves the lowest cumulative regret across multiple runs. 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 [16].

Real-World AutoML Tasks. We test all four quantum bandit algorithms on three different hyper-parameter 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 D 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 application.



(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.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

[2] Andris Ambainis, András Gilyén, Stacey Jeffery, and Martins Kokainis. Quadratic speedup for finding marked vertices by quantum walks. In *Annual ACM SIGACT Symposium on Theory of Computing*, pages 412–424, 2020.

[3] Joran van Apeldoorn and András Gilyén. Improvements in quantum sdp-solving with applications. In *International Colloquium on Automata, Languages and Programming*, 2018.

[4] Simon Apers and Alain Sarlette. Quantum fast-forwarding: Markov chains and graph property testing. *Quantum Information and Computation*, 19(3–4):181–213, 2019.

[5] Fernando G. S. L. Brandão and Krysta Marie Svore. Quantum speed-ups for solving semidefinite programs. *Annual Symposium on Foundations of Computer Science*, 2017.

[6] Fernando G. S. L. Brando, Amir Kalev, Tongyang Li, Yen Yu Lin, Krysta M. Svore, and Xiaodi Wu. Quantum sdp solvers: Large speed-ups, optimality, and applications to quantum learning. In *International Colloquium on Automata, Languages and Programming*, 2019.

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

[8] Shouvanik Chakrabarti, Andrew M Childs, Tongyang Li, and Xiaodi Wu. Quantum algorithms and lower bounds for convex optimization. *Quantum*, 4:221, 2020.

[9] Zherui Chen, Yuchen Lu, Hao Wang, Yizhou Liu, and Tongyang Li. Quantum langevin dynamics for optimization. *arXiv preprint arXiv:2311.15587*, 2023.

[10] Andrew M Childs, Jiaqi Leng, Tongyang Li, Jin-Peng Liu, and Chenyi Zhang. Quantum simulation of real-space dynamics. *Quantum*, 6:860, 2022.

[11] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, 2017.

[12] Robert Clarke, Habtom W Ressom, Antai Wang, Jianhua Xuan, Minetta C Liu, Edmund A Gehan, and Yue Wang. The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nature Reviews Cancer*, 8(1):37–49, 2008.

[13] Cecil C Craig. On the tchebychef inequality of bernstein. *The Annals of Mathematical Statistics*, 4(2):94–102, 1933.

[14] Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. Sbeed: Convergent reinforcement learning with nonlinear function approximation. In *International conference on machine learning*, 2018.

[15] Zhongxiang Dai, Yao Shu, Bryan Kian Hsiang Low, and Patrick Jaillet. Sample-then-optimize batch neural thompson sampling. *Advances in Neural Information Processing Systems*, 35, 2022.

[16] Zhongxiang Dai, Gregory Kang Ruey Lau, Arun Verma, Yao Shu, Bryan Kian Hsiang Low, and Patrick Jaillet. Quantum bayesian optimization. *Advances in Neural Information Processing Systems*, 36, 2023.

[17] Kefan Dong, Jiaqi Yang, and Tengyu Ma. Provable model-based nonlinear bandit and reinforcement learning: Shelve optimism, embrace virtual curvature. *Advances in Neural Information Processing Systems*, 34, 2021.

[18] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[19] Peter I Frazier and Jialei Wang. Bayesian optimization for materials design. *Information science for materials discovery and design*, pages 45–75, 2016.

[20] Ankit Garg, Robin Kothari, Praneeth Netrapalli, and Suhail Sherif. Near-optimal lower bounds for convex optimization for all orders of smoothness. *Advances in Neural Information Processing Systems*, 34, 2021.

[21] Ankit Garg, Robin Kothari, Praneeth Netrapalli, and Suhail Sherif. No quantum speedup over gradient descent for non-smooth convex optimization. In *Innovations in Theoretical Computer Science Conference*, 2021.

[22] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, 2019.

[23] Weiyuan Gong, Chenyi Zhang, and Tongyang Li. Robustness of quantum algorithms for nonconvex optimization. *arXiv preprint arXiv:2212.02548*, 2022.

[24] Yasunari Hikima, Kazunori Murao, Sho Takemori, and Yuhei Umeda. Quantum kernelized bandits. In *Conference on Uncertainty in Artificial Intelligence*, 2024.

[25] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on learning theory*, 2020.

[26] Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Neural architecture search with bayesian optimisation and optimal transport. *Advances in Neural Information Processing Systems*, 31, 2018.

[27] Iordanis Kerenidis and Anupam Prakash. A quantum interior point method for lps and sdps. *ACM Transactions on Quantum Computing*, 1(1):1–32, 2020.

[28] Ksenia Korovina, Sailun Xu, Kirthevasan Kandasamy, Willie Neiswanger, Barnabas Poczos, Jeff Schneider, and Eric Xing. Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations. In *International Conference on Artificial Intelligence and Statistics*, 2020.

[29] Jiaqi Leng, Ethan Hickman, Joseph Li, and Xiaodi Wu. Quantum hamiltonian descent. *arXiv preprint arXiv:2303.01471*, 2023.

[30] Tongyang Li and Ruizhe Zhang. Quantum speedups of optimizing approximately convex functions with applications to logarithmic regret stochastic convex bandits. *Advances in Neural Information Processing Systems*, 35, 2022.

[31] Chengchang Liu, Chaowen Guan, Jianhao He, and John CS Lui. Quantum algorithms for non-smooth non-convex optimization. *Advances in Neural Information Processing Systems*, 37, 2024.

[32] Chong Liu and Yu-Xiang Wang. Global optimization with parametric function approximation. In *International Conference on Machine Learning*, 2023.

[33] Yizhou Liu, Weijie J Su, and Tongyang Li. On quantum speedups for nonconvex optimization via quantum tunneling walks. *Quantum*, 7:1030, 2023.

[34] Ashley Montanaro. Quantum speedup of monte carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):1–20, 2015.

[35] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[36] Robert Nowak. Lecture notes: Complexity regularization for squared error loss, 2009.

[37] Valerio Perrone, Huibin Shen, Matthias W Seeger, Cedric Archambeau, and Rodolphe Jenatton. Learning search spaces for bayesian optimization: Another view of hyperparameter transfer learning. *Advances in neural information processing systems*, 32, 2019.

[38] Jörg Rahnenführer, Riccardo De Bin, Axel Benner, Federico Ambrogi, Lara Lusa, Anne-Laure Boulesteix, Eugenia Migliavacca, Harald Binder, Stefan Michiels, Willi Sauerbrei, et al. Statistical analysis of high-dimensional biomedical data: a gentle introduction to analytical goals, common approaches and challenges. *BMC Medicine*, 21(1):182, 2023.

[39] Patrick Rall and Bryce Fuller. Amplitude estimation from quantum signal processing. *Quantum*, 7:937, 2023.

[40] Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. Lower bounds on regret for noisy gaussian process bandit optimization. In *Conference on Learning Theory*, 2017.

[41] Benjamin J Shields, Jason Stevens, Jun Li, Marvin Parasram, Farhan Damani, Jesus I Martinez Alvarado, Jacob M Janey, Ryan P Adams, and Abigail G Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, 2021.

[42] Aaron Sidford and Chenyi Zhang. Quantum speedups for stochastic optimization. *Advances in Neural Information Processing Systems*, 36, 2023.

[43] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable bayesian optimization using deep neural networks. In *International conference on machine learning*, 2015.

[44] Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust bayesian neural networks. *Advances in neural information processing systems*, 29, 2016.

[45] Joran Van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Quantum sdp-solvers: Better upper and lower bounds. In *Annual Symposium on Foundations of Computer Science*, pages 403–414, 2017.

[46] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Convex optimization using quantum oracles. *Quantum*, 4:220, 2020.

[47] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

[48] Zongqi Wan, Zhijie Zhang, Tongyang Li, Jialin Zhang, and Xiaoming Sun. Quantum multi-armed bandits and stochastic linear bandits enjoy logarithmic regrets. In *AAAI Conference on Artificial Intelligence*, 2023.

[49] Colin White, Willie Neiswanger, and Yash Savani. Bananas: Bayesian optimization with neural architectures for neural architecture search. In *AAAI conference on Artificial Intelligence*, 2021.

[50] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

[51] Jian Wu, Saul Toscano-Palmerin, Peter I Frazier, and Andrew Gordon Wilson. Practical multi-fidelity bayesian optimization for hyperparameter tuning. In *Conference on Uncertainty in Artificial Intelligence*, 2020.

[52] Yulian Wu, Chaowen Guan, Vaneet Aggarwal, and Di Wang. Quantum heavy-tailed bandits. *arXiv preprint arXiv:2301.09680*, 2023.

[53] Chenyi Zhang and Tongyang Li. Quantum lower bounds for finding stationary points of nonconvex functions. In *International Conference on Machine Learning*, 2023.

[54] Chenyi Zhang, Jiaqi Leng, and Tongyang Li. Quantum algorithms for escaping from saddle points. *Quantum*, 5:529, 2021.

[55] Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural thompson sampling. In *International Conference on Learning Representations*, 2021.

[56] Yexin Zhang, Chenyi Zhang, Cong Fang, Liwei Wang, and Tongyang Li. Quantum algorithms and lower bounds for finite-sum optimization. In *International Conference on Machine Learning*, 2024.

[57] Yichi Zhang, Daniel W Apley, and Wei Chen. Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Scientific reports*, 10(1):4924, 2020.

[58] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, 2020.

Appendix A. Auxiliary Lemmas

In this section, we show auxiliary lemmas and definitions that will be used later in proofs.

Lemma 7 (Upper bound of summation of weights (Adapted from Lemma 2 in [16])) *Set* $\epsilon_s = \|\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\|_{\Sigma_s^{-1}}$*, then it satisfies that*

$$\sum_{s=1}^{m} \frac{1}{\epsilon_s} \leq T \quad \text{and} \quad \sum_{s=1}^{m} \frac{1}{\epsilon_s^2} \leq T^2.$$

Proof First we can lower bound the total number of rounds in m stages. Following Eq. (9), we have

$$\sum_{s=1}^{m} \frac{C_1}{\epsilon_s} \log \frac{m}{\delta} \geq \sum_{s=1}^{m} \frac{1}{\epsilon_s} \geq \sqrt{\sum_{s=1}^{m} \frac{1}{\epsilon_s^2}}.$$

Suppose that $\sum_{s=1}^{m} \frac{1}{\epsilon_s^2} \geq T^2$, then we have

$$\sum_{s=1}^{m} \frac{C_1}{\epsilon_s} \log \frac{m}{\delta} > T,$$

which is a contradiction. Similarly we can prove both statements. ■

Lemma 8 ([47]) *Let* X *be a random variable with mean* μ *and variance* σ^2*. If* X *satisfies the following Bernstein moment condition,*

$$\mathbf{E}[|X-\mu|^k] \leq \frac{\sigma^2}{2} k!\, \beta^{k-2}$$

for some $\beta > 0$ *and all* $k \geq 2$*, then*

$$\mathbf{E}[e^{\lambda(X-\mu)}] \leq \exp\left(\frac{\lambda^2\sigma^2}{2(1-\beta\lambda)}\right), \quad \forall \lambda \in \left[0, \frac{1}{\beta}\right).$$

Appendix B. Discussion on Quantum Oracle

Unlike classical access to samples of random variables, in quantum realm, the corresponding distribution is being accessed by making query to quantum sampling oracle.

Definition 9 (Quantum sampling oracle) *For a random variable* Y *with (finite) sample space* Ω*, its quantum sampling oracle* $\mathcal{O}_Y$ *is defined as*

$$\mathcal{O}_Y : |0\rangle \longmapsto \sum_{\mathbf{y}\in\Omega} \sqrt{\Pr[Y=\mathbf{y}]}|\mathbf{y}\rangle \otimes |\psi_{\mathbf{y}}\rangle, \tag{7}$$

where $|\psi_{\mathbf{y}}\rangle$ *is an arbitrary quantum state for every* $\mathbf{y}$*.*

The content in second quantum register can also be viewed as possible quantum garbage appeared during the implementation of the oracle. Observe that if we directly measure the output of $\mathcal{O}_Y$, it will collapse to a classical sampling access to Y that returns a random sample $\mathbf{y}$ with respect to probability $\Pr[Y=\mathbf{y}]$. In particular, the quantum noisy function Eq. (1) is an instance of the quantum sampling oracle.

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

Feasibility and Practicality. This same type of quantum oracles were also used in previous work [16, 24, 42, 48]. As discussed in [16], a quantum oracle is available when the learning environment is implemented by a quantum algorithm, which makes the setting of their quantum Bayesian optimization fairly general. Therefore, for example, such quantum oracles will arise naturally when quantum bandit algorithms are used to optimize hyperparameters of quantum neural networks and any other quantum machine learning models. Furthermore, there are standard techniques [35] in theory for implementing the quantum analogs of classical algorithms. Specifically, if there is a classical circuit for the given classical oracle, its quantum version of the same asymptotic computational complexity can be built using standard technique. Thus, in some scenarios, quantum oracles can be considered as a quantum generalization of their classical counterparts, and then this framework can be applied to optimize the parameters of models implemented on a quantum computer or in quantum systems where the data itself is inherently quantum, as pointed out in [16]. In particular, [16] successfully implemented their algorithm in the IBM real quantum computer, which shows that the quantum oracle can be fully realized in the real world.

In practice, however, the current development of quantum hardware is still at a relatively early stage. The actual implementation feasibility depends on factors such as circuit size and the complexity of the required quantum gates. Therefore, we believe that our quantum oracle can be practically feasible in small-size problem settings where the oracle is given as an explicit circuit.

Appendix C. Full Proofs

In this section, we show full proofs of all technical results in the main paper.

C.1. Quantum Non-Linear Regression Oracle

In this section, we first review the details and properties for step 1 of Algorithm 2, and the rest of the algorithm. At the end we provide the necessary theoretical analysis for the complete Algorithm 2.

C.1.1. PROOF OVERVIEW ON Oracle

In this section, we describe the quantum regression oracle, denoted as Oracle in step 1 in Algorithm 2, that we need for solving a non-linear *non-linear least square* problem.

Let $D^{T_0}\mathbf{w}$ denote a classical algorithm for solving a *non-linear least square* problem where $\mathbf{w}$ is an initial value of the solution and D is a matrix representing one iteration of the algorithm. It is easy to see that Markov chain-based algorithms (e.g SGD) can be represented using this notation.

Before presenting the actual proof of Theorem 16, we provide a high-level overview of the key technical aspects. The core idea in proving Lemma 5.1 is formulating the difference between expected risk and empirical risk as bounded variables satisfying a certain condition and then applying the Craig-Bernstein (CB) inequality [13]. In the formulation, the variable is linked to the unknown parametric function within the optimization problem (i.e. $f_{\mathbf{w}}(\mathbf{x})$), and the inequality is then utilized in a summation across a set of data samples. This approach is standard for classical algorithms. However, to fully harness the quantum advantage, we can employ a quantum sampling oracle on the dataset, as defined in Eq. (7). With such oracle, the random variables being used in the proof will be associated with the expected risk function $L(\mathbf{w})$, and then the summation in the application of CB inequality will be computed over the the number of iterations. Note that this approach still only yields a rate of $\tilde{O}(1/T_0)$. To obtain an improved rate of $\tilde{O}(1/T_0^2)$, we leverage the quadratic speed-up provided by quantum computing when implementing algorithms that satisfy the framework specified as in $D^{T_0}\mathbf{w}$. While such quantum enhancements can be realized using quantum singular value transformation [22], we instead adopt the quantum fast-forward technique introduced in [4], following the description in [2].

Theorem 10 (Formal rephrase of Lemma 15 in [2]) *Let $\epsilon \in (0,1), s \in [0,1]$ and $t \in \mathbf{N}$. Let P be any transition matrix which defines a reversible Markov chain on state space X and D be its discriminant matrix. Also let Q be the cost of implementing a quantum walk step. There is a quantum algorithm with complexity $O(\mathsf{Q}\sqrt{t\log(1/\epsilon)}$ that takes input $|\bar{0}\rangle|\psi\rangle \in \mathrm{span}\{|\bar{0}\rangle|x\rangle : x \in X\}$, and outputs a state that is ϵ-close to a state of the form*

$$|0\rangle^{\otimes a}|\bar{0}\rangle D^t|\psi\rangle + |\Gamma\rangle \tag{8}$$

where $a = O(\log(t\log(1/\epsilon)))$, $|\bar{0}\rangle$ is some fixed reference state, and $|\Gamma\rangle$ is some garbage state that has no support on states containing $|0\rangle^{\otimes a}|\bar{0}\rangle$ in the first two registers.

Informally, this theorem shows that a t-step classical random walk/Markov chain can be closely approximated by quantum walk using only $\sqrt{t}$ steps. This implies that the same performance guarantees achieved by a t-step classical random walk can also be attained by a quantum algorithm with time complexity $\propto \sqrt{t}$.

A more visual intuition on how the step 1 of Algorithm 1 works is pictured in Figure 2. It describes a high-level picture of a quantum algorithm implementing $D^{T_0}\mathbf{w}$ which denotes an execution of a classical algorithm solving a non-linear least square problem. More concretely, the input qubits to the quantum algorithm can be divided into three different parts: parameter register, input data register, and auxiliary qubits. Then the complete quantum process before the measurement step gives $(UL)^{T_0}|w\rangle|0\rangle|0\rangle$ (here 0's in the last two registers can represent multiple qubits), and the resulting state will have information of $D^{T_0}\mathbf{w}$. The composite operator UL makes one step of the quantum walk, like one update in SGD. During each step, operator L transforms the input from the last step into a quantum state whose expected value yields the gradient, and U performs the update process.

In quantum algorithms leveraging Markov chains for optimization problems, their inputs can be decomposed into three parts. The parameter register consists of qubits that store the parameter information. The input data register is dedicated to generating the distribution for data samples. The auxiliary qubits provide the workspace for temporary information produced during the quantum algorithms that are not part of the final output. Given an input state across those three registers, the quantum oracle $\mathcal{L}$–whose detailed description will be discussed in the subsequent paragraph–transforms the input into a quantum state whose expectation value yields the gradient $\nabla\mathcal{L}$. While

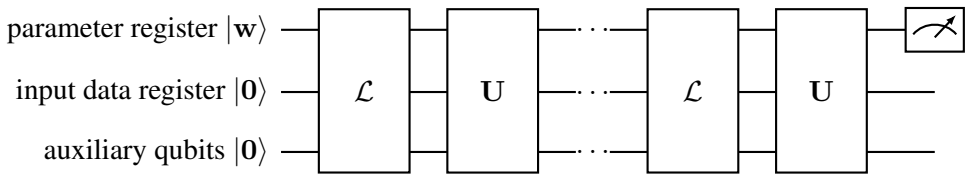


Figure 2: High-level description of a quantum algorithm computing $D^{T_0}\mathbf{w}$

techniques from [42] enable unbiased mean estimation for gradient extraction, we omit further details here as they lie outside our scope. The unitary $\mathbf{U}$ then processes the output of $\mathcal{L}$, implementing a single iteration of a Markov-chain-based update rule, say determined by $\mathbf{D}$, for solving non-linear least square problems. Roughly speaking, $\mathbf{U}$ combines two components: (1) an unbiased non-destructive multivariate mean estimation subroutine; and (2) a quantum walk step over the parameter space. The composite operation $\mathbf{U}\mathcal{L}$ constitutes one full iteration of the algorithm, as formalized in Theorem 10. Starting from the initial state, after T_0 iterations, i.e. $(\mathbf{U}\mathcal{L})^{T_0}|\mathbf{w}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle$, the final quantum state will encodes $\mathbf{D}^{T_0}\mathbf{w}$. To gain a quadratic speed-up, we apply the quantum fast-forward technique to $(\mathbf{U}\mathcal{L})^{T_0}$. Consequently, the step 1 of Algorithm 1 approximates Eq. (8) with $\widetilde{O}(\sqrt{T_0})$ iterations.

Since the quantum oracle $\mathcal{L}$ plays a pivotal role in the process, we now give a detailed discussion of its implementation requirements and constraints. To formalize its purpose, without loss of generality, consider the gradient of loss function $\mathbf{E}_{\mathbf{x}\sim D}[(f_{\mathbf{w}}(\mathbf{x}) - f_{\mathbf{w}^*}(\mathbf{x}))^2]$, where the parametric function f can be viewed as a function with input $(\mathbf{w}, \mathbf{x})$. By standard differentiation, its gradient is $\mathbf{E}_{\mathbf{x}\sim D}[2(f_{\mathbf{w}}(\mathbf{x}) - f_{\mathbf{w}^*}(\mathbf{x}))\nabla f_{\mathbf{w}}(\mathbf{x})]$. The oracle $\mathcal{L}$ must therefore prepare a quantum state whose expectation value encodes this gradient. Note that $\mathcal{L}$ viewed as an operator maps $|\mathbf{w}\rangle$ to a quantum state $\sum_Y \sqrt{\Pr[Y]}|\mathcal{A}(Y)\rangle$ for some operations $\mathcal{A}$, and its expectation gives $\mathbf{E}_Y[\mathcal{A}(Y)]$. Hence, we need to design the necessary $\mathcal{A}$ so that the multivariate mean value $\mathbf{E}_Y[\mathcal{A}(Y)]$ produces $\mathbf{E}_{\mathbf{x}\sim D}[2(f_{\mathbf{w}}(\mathbf{x}) - f_{\mathbf{w}^*}(\mathbf{x}))\nabla f_{\mathbf{w}}(\mathbf{x})]$. One caveat is that $f_{\mathbf{w}^*}(\mathbf{x})$ is not accessible directly. Instead, we are given access to the quantum noisy function oracle Eq. (1). However, this limitation can be addressed by leveraging the fact that $\mathbf{E}[y] = f_{\mathbf{w}^*}(\mathbf{x})$. Overall, we can construct the quantum oracle $\mathcal{L}$ as follows.

Given the quantum noisy function oracle $\mathcal{O}_{\mathbf{x}}$ (as defined in Eq. (1)) for all $\mathbf{x}$ in $\mathcal{X}$, we can obtain the following quantum oracle $\mathcal{L}$:

$$\mathcal{L} : |\mathbf{w}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle \longmapsto \sum_{\mathbf{x}}\sum_{\mathbf{e}} \sqrt{\Pr[\mathbf{x}]} \cdot \sqrt{\Pr[\mathbf{e}]}|(f_{\mathbf{w}}(\mathbf{x}) - (f_{\mathbf{w}^*}(\mathbf{x}) + \mathbf{e})) \cdot \frac{\partial f_{\mathbf{w}}(\mathbf{x})}{\partial w_1}\rangle$$
$$\otimes \cdots |(f_{\mathbf{w}}(\mathbf{x}) - (f_{\mathbf{w}^*}(\mathbf{x}) + \mathbf{e})) \cdot \frac{\partial f_{\mathbf{w}}(\mathbf{x})}{\partial w_d}\rangle \otimes |\text{garbage}\rangle.$$

The proof for this conversion is straightforward. By standard quantum computation principles, any classical circuit g can be embedded into a quantum circuit G such that: $|\mathbf{x}\rangle|\mathbf{0}\rangle \xmapsto{G} |\mathbf{x}\rangle|g(\mathbf{x})\rangle$. Building on this, without loss of generality, we construct a quantum circuit that prepares a quantum state whose expectation value encodes the gradient of expected loss function $L = \mathbf{E}_{\mathbf{x}}[(f_{\mathbf{x}}(\mathbf{w}) - f_{\mathbf{x}}(\mathbf{w}^*))^2]$. Three components enable this: (1) the parametric function f is classically specified; (2) the data input distribution can be created from scratch efficiently; and (3) the value $f_{\mathbf{x}}(\mathbf{w}^*)$ can be

obtained as the mean of the operator in Eq. (1). Following the standard embedding process mentioned above, these three components are combined into a unified quantum circuit acting on registers.

C.1.2. OVERVIEW ON ALGORITHM 2

Let $|\hat{\mathbf{w}}_0\rangle$ be the quantum state returned by Oracle in step 1 and write $|\hat{\mathbf{w}}_0\rangle = \sum_i^{d_\mathbf{w}} a_i|i\rangle$. The rest of Algorithm 2 is about retrieving a_i's for $i = 1, \cdots, d_\mathbf{w}$. One standard approach to achieve that is using quantum state tomography. However, in our case, we can use amplitude estimation instead because each $a_i \in [0, 1]$. At the same we want to avoid repeatedly calling Oracle for obtaining more copies of $|\hat{\mathbf{w}}_0\rangle$ since that will dramatically increase the number of queries to the quantum sampling oracle $\mathcal{O}_\mathbf{x}$, which will in turn cause extra cumulative regret. Hence, we will a technique called *non-destructive amplitude estimation* which can return both an estimate of a_i and a copy of the input state $|\hat{\mathbf{w}}_0\rangle$. There are multiple existing works in this field. [39] listed multiple of them and we pick one among them. Below is a more formal statement of Theorem 17.

Theorem 11 (Non-destructive amplitude estimation in [39]) *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 a quantum algorithm* NDAE$(|\psi\rangle, P, \epsilon, \delta)$ *that outputs with probability at least $1 - \delta$, an estimate $\tilde{a}$ and a copy of $|\psi\rangle$ such that $|\tilde{a} - a| \leq \epsilon$ and it uses the reflections and rotations on $|\psi\rangle$ and P $\tilde{\mathcal{O}}(\frac{1}{\epsilon\sqrt{\delta}})$ times.*

Remark 12 *Given a projector P, its reflection can be defined as $2P - I$ or $I - 2P$, and its rotation is defined as $e^{2\theta(2P-I)}$ or $e^{2\theta(I-2P)}$ for arbitrary phases θ. Note that the projector for a quantum state $|\psi\rangle$ can be represented as $|\psi\rangle\langle\psi|$. Hence, the reflections and rotations for state $|\psi\rangle$ can be defined similarly. Since neither reflections or rotations require queries to the quantum sampling oracles $\mathcal{O}_\mathbf{s}$, the usage of them will not contribute to cumulative regret. Hence, their implementations are outside the scope of our work. Interested readers can refer to [39] for details on how to implement them using the given state and projector.*

With NDAE, we can retrieve an estimate for each a_i for $i = 1, \cdots, d_\mathbf{w}$ with only one copy of $|\hat{\mathbf{w}}_0\rangle$, thus avoiding the expenses caused by repeatedly solving a non-linear least square problem. Overall, the correctness of Algorithm 2 is a result of Theorem 16 and Theorem 11, which can be summarized as follows.

Theorem 13 (Restatement of Theorem 5) *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*

$$\|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2 \leq \frac{\sqrt{Cd_\mathbf{w}\iota}}{T_0},$$

where ι is the logarithmic term depending on $T_0, C_h, 1/\delta$ and T_0 satisfies $T_0^2 \geq Cd_\mathbf{w}\iota \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{2C_g^2}{\mu c^2}\right\}$.

Its proof is presented in the next section.

C.1.3. THEORETICAL ANALYSIS

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 [32] with quantum fast-forward technique [4], shown below.

Lemma 14 (Adapted from Theorem 5.2 in [32]) *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 ι 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 [4]. An informal description below follows that in [2].

Lemma 15 (Informal statement of quantum fast-forward [2]) *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 $|\bar{0}\rangle|\psi\rangle \in \mathrm{span}\{|\bar{0}\rangle|x\rangle : x \in X\}$, and outputs a state that is ϵ-close to a state of the form $|0\rangle^{\otimes a}|\bar{0}\rangle D^t|\psi\rangle + |\Gamma\rangle$ where $a = O(\log(t\log(1/\epsilon)))$, $|\bar{0}\rangle$ is some fixed reference state, and $|\Gamma\rangle$ is some garbage state that has no support on states containing $|0\rangle^{\otimes a}|\bar{0}\rangle$ 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 16 *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 ι 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. [39] listed multiple of them and we pick one among them. An informal description is stated below.

Theorem 17 ((Informal) Non-destructive amplitude estimation in [39]) *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 $|\tilde{a} - a| \leq \epsilon$.*

Recall that a quantum sampling oracle enables the access to a superposition that entirely encodes a distribution of finite size. The following notations will be found convenient in constructing our proofs. Let $Y(Z) = f_{\mathbf{w}^*}(Z)$ and define $F_{\mathbf{w}}(X) := \sum_{Z\in D(X)}(f_{\mathbf{w}}(Z) - Y(Z))^2 = \sum_{i=1}^{M}(f_{\mathbf{w}}(Z_i) - Y(Z))^2$ where $D(X)$ is some probability distribution with finite sample space $\Omega \subseteq \mathbf{R}^d$ centered at X of size $|\Omega| = M$, $L(f_{\mathbf{w}}) = \mathbf{E}_X F_{\mathbf{w}}(X)$, and $\hat{L}(f_{\mathbf{w}}) = 1/T_0 \sum_{i=1}^{T_0} F_{\mathbf{w}}(X_i)$. It is easy to check that $\mathbf{E}_X[\hat{L}(f_{\mathbf{w}})] = L(f_{\mathbf{w}})$. We also assume that the output value of f is contained in a interval of length $b \geq 1$. Note that if $M = 1$, then the quantum sampling oracle becomes a classical oracle. Hence, without loss of generality, we can have $M \gg 1$.

Remark 18 *In classical setting, $Y(Z) = f_{\mathbf{w}^*}(Z) + \eta$, and a $Y(Z)$ is returned when querying with input Z, and one can estimate $f_{\mathbf{w}^*}(Z)$ by Chernoff bound. However, in quantum realm, one can extract $f_{\mathbf{w}^*}(Z)$ while still maintaining it as a quantum superposition instead of a classical sample by mean estimation algorithms [34, 42] without performing the measurement. Therefore, it is not a strong relaxation to have $Y(Z) = f_{\mathbf{w}^*}(Z)$ in the analysis.*

Lemma 19 *Given an oracle that accesses a dataset $\{Z, f_{\mathbf{w}^*}(Z)\}_{Z\in D(X)}$ where D is some distribution with finite sample space $\Omega \subseteq \mathbf{R}^d$ centered at X when queried with input X. Let $\mathcal{F}$ be a finite function class satisfying $\mathcal{F} \subset \{f : [0,1]^d \to [-b, b]\}$ for some $b \geq 1$ and define empirical risk minimizer (ERM) $\hat{f}_{T_0} := \arg\min_{f\in\mathcal{F}} \left\{\hat{L}(f)\right\}$. Then with probability at least $1-\delta$, the following holds*

$$L(\hat{f}_{T_0}) \leq \frac{1+\alpha}{1-\alpha} L(\tilde{f}) + 2\frac{\log|\mathcal{F}| + \log 2/\delta}{(1-\alpha)T_0\epsilon}$$

where $\epsilon < \frac{c_1^{k/(k-2)}}{4Lb^2}$, $\alpha = \epsilon b^2$, and $\tilde{f} \in \mathcal{F}$ is arbitrary.

Proof The proof follows a structure similar to that in [36], but some extra work is needed for our application. Define variables $U_i = -F_{\mathbf{w}}(X_i)$. Then $L(f_{\mathbf{w}}) - \hat{L}(f_{\mathbf{w}}) = \frac{1}{T_0}\sum_{i=1}^{T_0}(U_i - \mathbf{E}_X[U_i])$. In order to apply the CB inequality [13] as in [36], we need to (1) verify that the variables U_i satisfy the Berstein's moment condition

$$\mathbf{E}_{X_i}[|U_i - \mathbf{E}_{X_i}[U_i]|^k] \leq \frac{\mathrm{Var}\,[U_i]}{2} k!\, h^{k-2}$$

for some $h > 0$ and all $k \geq 2$; (2) find an upper bound for $\mathrm{Var}\,[U_i]$. They can be achieved by the following two claims.

Claim 20 *The Berstein's moment condition holds with $h = \frac{2\sqrt{M}b}{c_1}$ with some constant c_1 where $1 \leq c_1 < \sqrt{\mathrm{Var}\,[U]}/\beta$, $\sqrt{\mathrm{Var}\,[U]} \neq c_1\beta$, and $\beta = \frac{2b^2}{3}$.*

Claim 21

$$\mathrm{Var}\left[U_i\right] \leq b^2 L(F_{\mathbf{w}}).$$

Suppose Claim 20 and Claim 21 hold. By applying the CB inequality with certain values ϵ and c, which will be determined later, we obtain that, with probability at least $1-\delta$,

$$L(f_{\mathbf{w}}) - \hat{L}(f_{\mathbf{w}}) \leq \frac{\log 1/\delta}{T_0\epsilon} + \frac{\epsilon b^2 L(F_{\mathbf{w}})}{2(1-c)}$$

for $0 < \epsilon h \leq c < 1$. Following the Kraft inequality and union bound trick on the finite set $\mathcal{F}$ in [36], we have that, for any $\delta > 0$,

$$L(f_{\mathbf{w}}) - \hat{L}(f_{\mathbf{w}}) \leq \frac{\log|\mathcal{F}| + \log 1/\delta}{T_0\epsilon} + \frac{\epsilon b^2 L(f_{\mathbf{w}})}{2(1-c)}, \quad \forall f_{\mathbf{w}} \in \mathcal{F}$$

with probability at least $1-\delta$.

Claim 22 *Let* $0 < \epsilon < \frac{c_1}{4Mb^2}$, $c = \epsilon h$, *and* $\alpha = \epsilon b^2$. *Then* $0 < c < \frac{1}{2}$ *and* $\frac{\epsilon b^2}{2(1-c)} < \alpha < 1$.

By Claim 22, we rearrange to get

$$(1-\alpha)L(f_{\mathbf{w}}) \leq \hat{L}(f_{\mathbf{w}}) + \frac{\log|\mathcal{F}| + \log 1/\delta}{T_0\epsilon}, \quad \forall f_{\mathbf{w}} \in \mathcal{F}$$

with probability at least $1-\delta$. Define

$$\hat{f}_{T_0} := \arg\min_{f_{\mathbf{w}} \in \mathcal{F}} \left\{ \hat{L}(f_{\mathbf{w}}) \right\}.$$

Then with probability at least $1-\delta$,

$$\begin{aligned}(1-\alpha)L(\hat{f}_{T_0}) &\leq \hat{L}(\hat{f}_{T_0}) + \frac{\log|\mathcal{F}| + \log 1/\delta}{T_0\epsilon}, \\ &\leq \hat{L}(\tilde{f}) + \frac{\log|\mathcal{F}| + \log 1/\delta}{T_0\epsilon},\end{aligned}$$

where $\tilde{f} \in \mathcal{F}$ is arbitrary. Similarly, we apply CB inequality to $\hat{L}(\tilde{f}) - L(\tilde{f}) = \frac{1}{T_0}\sum_{i=1}^{T_0} -(U_i - \mathbf{E}_X[U_i])$ and attain

$$\hat{L}(\tilde{f}) - L(\tilde{f}) \leq \alpha L(\tilde{f}) + \frac{\log|\mathcal{F}| + \log 1/\delta}{T_0\epsilon}$$

with probability at least $1-\delta$. By union bound, we get

$$L(\hat{f}_{T_0}) \leq \frac{1+\alpha}{1-\alpha} L(\tilde{f}) + 2\frac{\log|\mathcal{F}| + \log 1/\delta}{(1-\alpha)T_0\epsilon}$$

with probability at least $1-2\delta$, for any $\delta > 0$. ■

Lemma 23 *Suppose Assumption 2 and Assumption 3 hold. There is an absolute constant C', such that after T_0 iterations in Step 1 of Algorithm 1, with probability at least $1-\delta/2$, the quantum regression oracle returns an estimate $\hat{\mathbf{w}}_0$ that satisfies*

$$L(f_{\hat{\mathbf{w}}_0}) \leq \frac{C' d_{\mathbf{w}} \iota}{T_0},$$

where ι is a logarithmic term depending on $T_0, C_h, 1/\delta$.

Proof Its proof follows that of Lemma 5.1 in [32] with minor changes. We provide it here for completeness.

Let $\tilde{\mathbf{w}}, \widetilde{\mathcal{W}}$ denote the ERM parameter and finite parameter class after applying covering number argument on $\mathcal{W}$. By Lemma 19, we obtain that with probability at least $1-\delta/2$,

$$\begin{aligned} L(f_{\tilde{\mathbf{w}}}) &\leq \frac{1+\alpha}{1-\alpha} L(f_{\mathbf{w}^*}) + 2\frac{\log|\mathcal{F}| + \log 4/\delta}{(1-\alpha)T_0\epsilon}, \\ &\leq 2\frac{\log|\mathcal{F}| + \log 4/\delta}{(1-\alpha)T_0\epsilon}, \end{aligned}$$

where the first inequality follows from $\mathbf{w}^* \in \mathcal{F} = \widetilde{\mathcal{W}} \cup \{\mathbf{w}^*\}$. Our parameter class $\mathcal{W} \subseteq [0,1]^{d_{\mathbf{w}}}$, so by ϵ-covering number argument, $\log(|\widetilde{\mathcal{W}}|) = \log(1/\epsilon'^{d_{\mathbf{w}}}) = d_{\mathbf{w}} \log(1/\epsilon')$, and then we have with probability at least $1-\delta/2$,

$$L(f_{\tilde{\mathbf{w}}}) \leq C'' \frac{d_{\mathbf{w}} \log(1/\epsilon') + \log 4/\delta}{T_0},$$

where C'' is a universal constant determined by α and ϵ picked in Lemma 19. By $(a+b)^2 \leq 2a^2 + 2b^2$,

$$\begin{aligned} L(f_{\hat{\mathbf{w}}_0}) &\leq 2\mathbf{E}_X[\sum_{Z \in D(X)} (f_{\hat{\mathbf{w}}_0}(Z) - f_{\tilde{\mathbf{w}}}(Z))^2] + 2L(f_{\tilde{\mathbf{w}}}), \\ &\leq 2\epsilon'^2 C_h^2 + 2C'' \frac{d_{\mathbf{w}} \log(1/\epsilon') + \log 4/\delta}{T_0}, \end{aligned}$$

where the second inequality applies discretization error ϵ' and Assumption 3. By choosing $\epsilon' = \frac{1}{C_h\sqrt{T_0}}$, the bound above becomes

$$\frac{2}{T_0} + \frac{C'' d_{\mathbf{w}} \log(T_0 C_h^2)}{T_0} + \frac{2C'' \log 4/\delta}{T_0} \leq C' \frac{d_{\mathbf{w}} \log(T_0 C_h^2) + \log(4/\delta)}{T_0},$$

where we can take $C' = 2C''$ and assume $2 < C'' d_{\mathbf{w}} \log(T_0 C_h^2)$. The proof completes by defining ι as the logarithmic term depending on $T_0, C_h, 1/\delta$. ■

Lemma 24 (Restatement of Lemma 14) *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*

$$\|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2 \leq \sqrt{\frac{Cd_{\mathbf{w}}\iota}{T_0}},$$

where ι *is the logarithmic term depending on* $T_0, C_h, 1/\delta$*.*

Its proof follows that of Theorem 5.2 in [32] with no changes required. Note that the rate of $\|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2$ is still $\tilde{O}(1/\sqrt{T_0})$ after T_0 iterations. To achieve the same rate of $\tilde{O}(1/\sqrt{T_0})$ after $\sqrt{T_0}$ iterations, we turn to Theorem 10 which shows that the T_0 iterations can be approximately with only $\sqrt{T_0}$ iterations.

Theorem 25 (Restatement of Theorem 16) *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*

$$T_0^2 \geq Cd_{\mathbf{w}}\iota \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{2C_g^2}{\mu c^2}\right\},$$

with probability $1-\delta/2$*, the quantum regression oracle returns an estimate* $\hat{\mathbf{w}}_0$ *that satisfies*

$$\|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2 \leq \frac{\sqrt{Cd_{\mathbf{w}}\iota}}{T_0},$$

where ι *is the logarithmic term depending on* $T_0, C_h, 1/\delta$*.*

Proof Suppose we select an algorithm used in this step following $D^{T_0}\mathbf{w}$. Let T_0 be the number of iterations in this algorithm. Then it can be formulated $D^{T_0}\mathbf{w}$ with a random initial parameter vector $\mathbf{w}$, and the estimate output $\hat{\mathbf{w}}_0 = D^{T_0}\mathbf{w}$. By Lemma 14, we obtain that with probability at least $1-\delta/2$,

$$\|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2^2 \leq \frac{Cd_{\mathbf{w}}\iota}{T_0}$$

after T_0 iterations of exact calculation.

Next by Theorem 10, we have that there is a quantum algorithm $\mathcal{A}$ can output a vector $\mathbf{w}'$ that is $\sqrt{\epsilon_1}$-close to $D^{T_0}\mathbf{w}$ with time complexity $O(\mathsf{Q}\sqrt{T_0\log(1/\sqrt{\epsilon_1})}$ where Q is the cost of performing one quantum walk step. That implies that with $O(\sqrt{T_0\log(1/\sqrt{\epsilon_1})}$ iterations, $\mathcal{A}$ will return an estimate $\sqrt{\epsilon_1}$-close to $\hat{\mathbf{w}}_0$. Overall, there is a quantum algorithm that takes input a random initial parameter $\mathbf{w}$ and outputs an estimate $\hat{\mathbf{w}_0}'$ such that with probability at least $1-\delta/2$,

$$\|\hat{\mathbf{w}_0}' - \mathbf{w}^*\|_2^2 \leq \frac{Cd_{\mathbf{w}}\iota}{T_0} + \epsilon_1$$

with $O(\sqrt{T_0\log(1/\sqrt{\epsilon_1})}$ steps. Finally, the desired statement can be derived by setting $\epsilon_1 = 1/T_0$ and change of variable. ■

Proof [Proof for Theorem 13] Denote $|\hat{\mathbf{w}}_0\rangle = \sum_i^{d_{\mathbf{w}}} b_i|i\rangle$. It is easy to check that $\langle\hat{\mathbf{w}}_0|P_i|\hat{\mathbf{w}}_0\rangle$ gives $|b_i|^2$. Note that as a vector $\hat{\mathbf{w}}_0 \in [0,1]^{d_{\mathbf{w}}}$, which means $|b_i| = b_i$.

Let $\delta' = \frac{\delta}{4d_{\mathbf{w}}}$. Then by Theorem 11, $\mathsf{NDAE}(|\hat{\mathbf{w}}_0\rangle, P_i, \frac{1}{d_{\mathbf{w}}T_0^2}, \delta')$ outputs with probability at least $1-\delta'$, an estimate $\tilde{a}_i$ of b_i^2 such that $|\tilde{a}_i - b_i^2| \leq \frac{1}{d_{\mathbf{w}}T_0^2}$. Since both $\sqrt{\tilde{a}_i}$ and b_i are nonnegative, we have

$$|\sqrt{\tilde{a}_i} - b_i| \leq \sqrt{|\tilde{a}_i - b_i^2|} \leq \frac{1}{\sqrt{d_{\mathbf{w}} \cdot T_0^2}}.$$

Let $\tilde{\mathbf{w}}_0 = (\sqrt{\tilde{a}_1}, \cdots, \sqrt{\tilde{a}_{d_{\mathbf{w}}}})$. Then

$$\|\tilde{\mathbf{w}}_0 - \hat{\mathbf{w}}_0\|_2^2 = \sum_i^{d_{\mathbf{w}}} |\sqrt{\tilde{a}_i} - b_i|^2 \leq \frac{1}{T_0^2},$$

and by triangle inequality,

$$\|\tilde{\mathbf{w}}_0 - \mathbf{w}^*\|_2 \leq \|\tilde{\mathbf{w}}_0 - \hat{\mathbf{w}}_0\|_2 + \|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2 \leq \frac{1+\sqrt{Cd_{\mathbf{w}}\iota}}{T_0}$$

where the last equality follows from the combination of the upper bound on $\|\tilde{\mathbf{w}}_0 - \hat{\mathbf{w}}_0\|_2$ and that on $\|\hat{\mathbf{w}}_0 - \mathbf{w}^*\|_2$ provided by Theorem 25.

Next, it is left to prove the success probability. First, step 1 in Algorithm 2 calls Oracle which has $\mathcal{O}(T_0)$ iterations and its success probability is at least $1-\delta/4$. Second, each $\mathsf{NDAE}(|\hat{\mathbf{w}}_0\rangle, P_i, \frac{1}{d_{\mathbf{w}}T_0^2}, \delta')$ succeeds with probability at $1-\delta'$. Hence, all the $d_{\mathbf{w}}$ calls to NDAE outputs the correct estimates with probability at least

$$\left(1-\delta'\right)^{d_{\mathbf{w}}} \geq 1 - d_{\mathbf{w}} \cdot \delta' = 1 - \frac{\delta}{4}$$

where the first equality follows from Bernoulli's inequality and the last step comes from the definition of δ'.

Overall, Algorithm 2 successfully outputs $\hat{\mathbf{w}}_0 = (\tilde{a}_1, \cdots, \tilde{a}_{d_w})$ with probability at least $(1 - \frac{\delta}{4})^2 \geq 1 - \frac{\delta}{2}$ as desired.

■

Proof [Proof of Claim 20] For each independent random variable $Z_i \sim D(X)$, define random variable $J_i = -(f(Z_i|\mathbf{w}) - f(Z_i|\mathbf{w}^*))^2$ and $V_i = J_i - \mathbf{E}[J_i]$. Then we can see that $U = \sum_{Z_i \sim D(X_i)} J_i$ with $\text{Var}[J_i] \leq b^2$, and $\mathbf{E}[V_i] = 0$. Since $D(X)$ has finite sample space Ω of size $|\Omega| = M$, we write $U = \sum_{\ell=1}^M J_\ell$, and hence $\text{Var}[U] \leq Mb^2$.

Next let $W = U - \mathbf{E}[U] = \sum_{\ell=1}^M V_i$ and observe that $\mathbf{E}[W] = 0$. As in [36], note that each independent J_i satisfies the following Bernstein condition

$$\mathbf{E}[|J_i - \mathbf{E}[J_i]|^k] = \mathbf{E}[|V_i|^k] \leq \frac{\text{Var}[J_i]}{2} k!\, \beta^{k-2}$$

for all $k \geq 2$ and with $\beta = \frac{2b^2}{3}$. Then by Lemma 8, we have that for each J_i,

$$\mathbf{E}[e^{\lambda V_i}] = \mathbf{E}[e^{\lambda(J_i - \mathbf{E}[J_i])}] \leq \exp\left(\frac{\lambda^2 \text{Var}[J_i]}{2(1-\lambda\beta)}\right), \quad \forall \lambda \in \left[0, \frac{1}{\beta}\right).$$

Next we put all M independent V_i together to get, with the same λ,

$$\mathbf{E}[e^{\lambda W}] = \mathbf{E}[\prod_{i=1}^M e^{\lambda V_i}] = \prod_{i=1}^M \mathbf{E}[e^{\lambda V_i}] \leq \exp\left(\frac{\lambda^2 \text{Var}[U]}{2(1-\lambda\beta)}\right), \quad \forall \lambda \in \left[0, \frac{1}{\beta}\right).$$

Note that this also holds when replacing W with $-W$. Hence, for $m = 1, 2, \cdots$

$$\frac{\lambda^{2m}}{(2m)!}\mathbf{E}[W^{2m}] \leq \mathbf{E}[\frac{e^{\lambda W} + e^{-\lambda W}}{2}] \leq \exp\left(\frac{\lambda^2 \text{Var}\,[U]}{2(1-\lambda\beta)}\right),$$

where the first inequality follows from the Taylor series of exponential functions.

Next we let $\lambda = \frac{c_1}{\sqrt{\text{Var}[U]}}$ with some constant $c_1 \geq 1$ such that $\sqrt{\text{Var}\,[U]} \neq c_1\beta$ and $\lambda < 1/\beta$. Then we can verify that $\exp\left(\frac{\lambda^2 \text{Var}[U]}{2(1-\lambda\beta)}\right)$ as a function of $\text{Var}\,[U]$ is decreasing for all $\text{Var}\,[U] \geq 0$ and $\sqrt{\text{Var}\,[U]} \neq c_1\beta$.

With substitution and rearrangement, we can derive

$$\mathbf{E}[W^{2m}] \leq (2m)!\frac{1}{\lambda^{2m}} \leq (2m)!\,(\frac{\sqrt{\text{Var}\,[U]}}{c_1})^{2m}.$$

Next we apply Cauchy-Schwarz inequality to attain, for $m = 1, 2, \cdots$,

$$\begin{aligned}
\mathbf{E}[|W|^{2m+1}] &\leq \sqrt{\mathbf{E}[W^{2m}]}\sqrt{\mathbf{E}[W^{2m+2}]}, \\
&\leq \sqrt{(2m)!\,(\frac{\sqrt{\text{Var}\,[U]}}{c_1})^{2m}}\sqrt{(2m+2)!\,(\frac{\sqrt{\text{Var}\,[U]}}{c_1})^{2m+2}}, \\
&\leq (2m+1)!\,(\frac{\sqrt{\text{Var}\,[U]}}{c_1})^{2m+1}.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\mathbf{E}[|U - \mathbf{E}[U]|^k] &= \mathbf{E}[|W|^k], \\
&\leq k!\,(\frac{\sqrt{\text{Var}\,[U]}}{c_1})^k, \\
&\leq k!\,\frac{1}{c_1^k}\text{Var}\,[U]\,(\sqrt{\text{Var}\,[U]})^{k-2}, \\
&\leq \frac{\text{Var}\,[U]}{2}k!\,\frac{(2\sqrt{M}b)^{k-2}}{c_1^k}, \\
&\leq \frac{\text{Var}\,[U]}{2}k!\,\left(\frac{2\sqrt{M}b}{c_1}\right)^{k-2},
\end{aligned}$$

where the last step follows from $c_1 \geq 1$. ■

Proof [Proof of Claim 21] By the definition of U_i, we have

$$\begin{aligned}
U_i^2 &= F_{\mathbf{w}}(X_i)^2, \\
&= \sum_{Z,Z' \in D(X)} (f_{\mathbf{w}}(\mathbf{Z}) - f_{\mathbf{w}^*}(\mathbf{Z}))^2 (f_{\mathbf{w}}(\mathbf{Z}') - f_{\mathbf{w}^*}(\mathbf{Z}'))^2, \\
&\leq b^2 \sum_{Z \in D(X)} (f_{\mathbf{w}}(\mathbf{Z}) - f_{\mathbf{w}^*}(\mathbf{Z}))^2,
\end{aligned}$$

and then

$$\text{Var}\,[U_i] \leq \mathbf{E}[U_i^2] = L(F_{\mathbf{w}}).$$

■

Proof [Proof of Claim 22] When $\epsilon < \frac{c_1}{4Mb^2}$,

$$c = \epsilon h = \epsilon \frac{2\sqrt{M}b}{c_1} < \frac{1}{2\sqrt{M}b} < \frac{1}{2}$$

where the second step is by the value of h from Claim 20 and the last step follows from the fact that $M, b \geq 1$.

Next,

$$\alpha = \epsilon b^2 < \frac{c_1}{4M}.$$

It is straightforward that $\alpha > \frac{\epsilon b^2}{2(1-c)}$ since $c > 0$. Note that by Claim 20,

$$1 \leq c_1 < \frac{\sqrt{\text{Var}\,[U]}}{\beta} \leq \frac{3\sqrt{M}}{2b},$$

since $\sqrt{\text{Var}\,[U]} \leq \sqrt{M}b$ and $\beta = 2b^2/3$. Then

$$\alpha < \frac{c_1}{4M} < \frac{3\sqrt{M}}{2b} \cdot \frac{1}{4M} = \frac{3}{8b\sqrt{M}} < 1,$$

where the last step stems from the fact that $M, b \geq 1$. ■

C.2. Regret Analysis

Theorem 26 (Restatement of Theorem 6) *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 ι 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 $1-\delta$,

$$R_T = O\left(d_w^2 \log^{\frac{3}{2}}(T) \log(d_w \log(T))\right).$$

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 [48], but the remaining high-order terms are bounded creatively using the convex property of parameter uncertainty region 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 27 (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 27, we take the summation of upper bounds and reach the upper bound of cumulative regret.

Proof Since Algorithm 1 runs in multiple stages where the same action is played for multiple rounds, so we first focus on the instantaneous regret of one round in each stage and then calculate the cumulative regret in all stages. At stage s, the instantaneous regret of one round r_s is defined as

$$r_s = f_0(\mathbf{x}^*) - f_0(\mathbf{x}_s) = f_{\mathbf{x}^*}(\mathbf{w}^*) - f_{\mathbf{x}_s}(\mathbf{w}^*),$$

where the second equation is due to Assumption 2. Recall that the selection process of $\mathbf{x}_s$ in Algorithm 1 is

$$\mathbf{x}_s = \arg\max_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{w}\in \mathrm{Ball}_s} f_{\mathbf{x}}(\mathbf{w}),$$

and we define $\tilde{\mathbf{w}}$ to be the parameter that maximizes the function value at $\mathbf{x}_s$, i.e, $\tilde{\mathbf{w}} = \arg\max_{\mathbf{w}\in\mathrm{Ball}_s} f_{\mathbf{x}_s}(\mathbf{w})$, then we have

$$r_s \le f_{\mathbf{x}_s}(\tilde{\mathbf{w}}) - f_{\mathbf{x}_s}(\mathbf{w}^*) = (\tilde{\mathbf{w}} - \mathbf{w}^*)^\top \nabla f_{\mathbf{x}_s}(\dot{\mathbf{w}}),$$

where the equation is by first order Taylor's theorem and $\dot{\mathbf{w}}$ is a parameter lying between $\tilde{\mathbf{w}}$ and $\mathbf{w}^*$. Due to the convex structure of Ball_s for each stage s, it implies that $\dot{\mathbf{w}} \in \mathrm{Ball}_s$. By adding and removing $\hat{\mathbf{w}}_s$ and $\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)$, we have

$$\begin{aligned}
r_s &\le (\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s + \hat{\mathbf{w}}_s - \mathbf{w}^*)^\top (\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0) - \nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0) + \nabla f_{\mathbf{x}_s}(\dot{\mathbf{w}})),\\
&= (\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s)^\top \nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0) + (\hat{\mathbf{w}}_s - \mathbf{w}^*)^\top \nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\\
&\quad + (\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s + \hat{\mathbf{w}}_s - \mathbf{w}^*)^\top (\nabla f_{\mathbf{x}_s}(\dot{\mathbf{w}}_s) - \nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)),\\
&\le \|\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s\|_{\Sigma_s} \|\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\|_{\Sigma_s^{-1}} + \|\hat{\mathbf{w}}_s - \mathbf{w}^*\|_{\Sigma_s} \|\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\|_{\Sigma_s^{-1}}\\
&\quad + (\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s + \hat{\mathbf{w}}_s - \mathbf{w}^*)^\top (\nabla f_{\mathbf{x}_s}(\dot{\mathbf{w}}_s) - \nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)),
\end{aligned}$$

where the last inequality is due to Holder's inequality.

Since both $\tilde{\mathbf{w}}$ and $\mathbf{w}^*$ are in Ball_s and $\epsilon_s = \|\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\|_{\Sigma_s^{-1}}$, we have

$$r_s \le 2\sqrt{\beta_s}\epsilon_s + (\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s + \hat{\mathbf{w}}_s - \mathbf{w}^*)^\top (\nabla f_{\mathbf{x}_s}(\dot{\mathbf{w}}) - \nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)).$$

Again by first order Taylor's theorem where $\ddot{\mathbf{w}}$ lies between $\dot{\mathbf{w}}$ and $\hat{\mathbf{w}}_0$, we have

$$\begin{aligned}
r_s &\le 2\sqrt{\beta_s}\epsilon_s + (\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s + \hat{\mathbf{w}}_s - \mathbf{w}^*)^\top \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}})(\dot{\mathbf{w}} - \hat{\mathbf{w}}_0),\\
&= 2\sqrt{\beta_s}\epsilon_s + (\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s + \hat{\mathbf{w}}_s - \mathbf{w}^*)^\top \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}})(\dot{\mathbf{w}} - \hat{\mathbf{w}}_s + \hat{\mathbf{w}}_s - \mathbf{w}^* + \mathbf{w}^* - \hat{\mathbf{w}}_0),\\
&= 2\sqrt{\beta_s}\epsilon_s + (\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s + \hat{\mathbf{w}}_s - \mathbf{w}^*)^\top \Sigma_s^{\frac{1}{2}} \Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{\frac{1}{2}} \Sigma_s^{-\frac{1}{2}} (\dot{\mathbf{w}} - \hat{\mathbf{w}}_s + \hat{\mathbf{w}}_s - \mathbf{w}^* + \mathbf{w}^* - \hat{\mathbf{w}}_0),\\
&= 2\sqrt{\beta_s}\epsilon_s + (\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s)^\top \Sigma_s^{\frac{1}{2}} \Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}} \Sigma_s^{\frac{1}{2}} (\dot{\mathbf{w}} - \hat{\mathbf{w}}_s)\\
&\quad + (\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s)^\top \Sigma_s^{\frac{1}{2}} \Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}} \Sigma_s^{\frac{1}{2}} (\hat{\mathbf{w}}_s - \mathbf{w}^*)
\end{aligned}$$

$$
\begin{aligned}
&+ (\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s)^\top \Sigma_s^{\frac{1}{2}} \Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}} \Sigma_s^{\frac{1}{2}} (\mathbf{w}^* - \hat{\mathbf{w}}_0) \\
&+ (\hat{\mathbf{w}}_s - \mathbf{w}^*)^\top \Sigma_s^{\frac{1}{2}} \Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}} \Sigma_s^{\frac{1}{2}} (\dot{\mathbf{w}} - \hat{\mathbf{w}}_s) \\
&+ (\hat{\mathbf{w}}_s - \mathbf{w}^*)^\top \Sigma_s^{\frac{1}{2}} \Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}} \Sigma_s^{\frac{1}{2}} (\hat{\mathbf{w}}_s - \mathbf{w}^*) \\
&+ (\hat{\mathbf{w}}_s - \mathbf{w}^*)^\top \Sigma_s^{\frac{1}{2}} \Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}} \Sigma_s^{\frac{1}{2}} (\mathbf{w}^* - \hat{\mathbf{w}}_0), \\
\leq\ & 2\sqrt{\beta_s}\epsilon_s + \|(\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s)^\top \Sigma_s^{\frac{1}{2}}\|_2 \|\Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}}\|_{\mathrm{op}} \|\Sigma_s^{\frac{1}{2}} (\dot{\mathbf{w}} - \hat{\mathbf{w}}_s)\|_2 \\
&+ \|(\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s)^\top \Sigma_s^{\frac{1}{2}}\|_2 \|\Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}}\|_{\mathrm{op}} \|\Sigma_s^{\frac{1}{2}} (\hat{\mathbf{w}}_s - \mathbf{w}^*)\|_2 \\
&+ \|(\tilde{\mathbf{w}} - \hat{\mathbf{w}}_s)^\top \Sigma_s^{\frac{1}{2}}\|_2 \|\Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}}\|_{\mathrm{op}} \|\Sigma_s^{\frac{1}{2}} (\mathbf{w}^* - \hat{\mathbf{w}}_0)\|_2 \\
&+ \|(\hat{\mathbf{w}}_s - \mathbf{w}^*)^\top \Sigma_s^{\frac{1}{2}}\|_2 \|\Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}}\|_{\mathrm{op}} \|\Sigma_s^{\frac{1}{2}} (\dot{\mathbf{w}} - \hat{\mathbf{w}}_s)\|_2 \\
&+ \|(\hat{\mathbf{w}}_s - \mathbf{w}^*)^\top \Sigma_s^{\frac{1}{2}}\|_2 \|\Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}}\|_{\mathrm{op}} \|\Sigma_s^{\frac{1}{2}} (\hat{\mathbf{w}}_s - \mathbf{w}^*)\|_2 \\
&+ \|(\hat{\mathbf{w}}_s - \mathbf{w}^*)^\top \Sigma_s^{\frac{1}{2}}\|_2 \|\Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}}\|_{\mathrm{op}} \|\Sigma_s^{\frac{1}{2}} (\mathbf{w}^* - \hat{\mathbf{w}}_0)\|_2,
\end{aligned}
$$

where the second line is by adding and removing $\hat{\mathbf{w}}_s$ and $\mathbf{w}^*$ and the last inequality is due to Holder's inequality. Since the center of Ball_s is $\hat{\mathbf{w}}_s$ and $\mathbf{w}^*, \tilde{\mathbf{w}}, \dot{\mathbf{w}} \in \mathrm{Ball}_s$, then we have

$$
\begin{aligned}
r_s &\leq 2\sqrt{\beta_s}\epsilon_s + 4\beta_s \|\Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}}\|_{\mathrm{op}} + 2\sqrt{\beta_s} \|\Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_t}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}}\|_{\mathrm{op}} \|\Sigma_s^{\frac{1}{2}} (\mathbf{w}^* - \hat{\mathbf{w}}_0)\|_2, \\
&\leq 2\sqrt{\beta_s}\epsilon_s + 4\beta_s \|\Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_s}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}}\|_{\mathrm{op}} + 2\sqrt{\beta_s} \|\Sigma_s^{-\frac{1}{2}} \nabla^2 f_{\mathbf{x}_t}(\ddot{\mathbf{w}}) \Sigma_s^{-\frac{1}{2}}\|_{\mathrm{op}} \|\Sigma_s^{\frac{1}{2}}\|_{\mathrm{op}} \|(\mathbf{w}^* - \hat{\mathbf{w}}_0)\|_2, \\
&\leq 2\sqrt{\beta_s}\epsilon_s + \frac{4\beta_s C_h}{\lambda} + \frac{2\sqrt{\beta_s} C_h C_0}{T_0 \sqrt{\lambda}},
\end{aligned}
$$

where the second inequality is again due to Holder's inequality and the last inequality is by Assumption 3, the choice of Σ_s, and convergence guarantee of $\hat{\mathbf{w}}_0$.

Recall that in stage s, the algorithm plays actions x_s for $\frac{C_1}{\epsilon_s} \log \frac{m}{\delta}$ rounds, therefore, the cumulative regret in stage s is bounded as

$$
\frac{C_1}{\epsilon_s} \left(2\sqrt{\beta_s}\epsilon_s + \frac{4\beta_s C_h}{\lambda} + \frac{2\sqrt{\beta_s} C_h C_0}{T_0 \sqrt{\lambda}} \right) \log \left(\frac{m}{\delta} \right).
$$

In total, there are m stages, so the cumulative regret is bounded as

$$
\begin{aligned}
R_T &\leq \sum_{s=1}^{m} \frac{C_1}{\epsilon_s} \left(2\sqrt{\beta_s}\epsilon_s + \frac{4\beta_s C_h}{\lambda} + \frac{2\sqrt{\beta_s} C_h C_0}{T_0 \sqrt{\lambda}} \right) \log \left(\frac{m}{\delta} \right), \\
&= 2C_1 \log \left(\frac{m}{\delta} \right) \sum_{s=1}^{m} \left(\sqrt{\beta_s} + \frac{2\beta_s C_h}{\epsilon_s \lambda} + \frac{\sqrt{\beta_s} C_h C_0}{\epsilon_s T_0 \sqrt{\lambda}} \right).
\end{aligned}
$$

Recall that our choice of β_s is

$$
\beta_s = 3d_w s + \frac{3\lambda C_0^2}{T_0^2} + \frac{3C_h^2 C_0^2 s T^2}{4T_0^4},
$$

which is increasing in s, so $\beta_s \leq \beta_m$ where m is the number of stages. Therefore, we have

$$\begin{aligned} R_T &\leq 2C_1 \log\left(\frac{m}{\delta}\right)\left(m\sqrt{\beta_m} + \left(\frac{2\beta_m C_h}{\lambda} + \frac{\sqrt{\beta_m}C_h C_0}{T_0\sqrt{\lambda}}\right)\sum_{s=1}^{m}\frac{1}{\epsilon_s}\right), \\ &\leq 2C_1 \log\left(\frac{m}{\delta}\right)\left(m\sqrt{\beta_m} + \frac{2\beta_m C_h T}{\lambda} + \frac{\sqrt{\beta_m}C_h C_0 T}{T_0\sqrt{\lambda}}\right), \end{aligned}$$

where the second inequality is due to Lemma 7.

Now plug in our choices $T_0 = \sqrt{T}, \lambda = T$ and maximum stage number $m = d_w \log(\frac{C_g^2 T}{d_w} + 1)$ (Lemma 27), and we have

$$\begin{aligned} R_T &\leq 2C_1 \log\left(\frac{m}{\delta}\right)\left(m\sqrt{\beta_m} + 2\beta_m C_h + \sqrt{\beta_m}C_h C_0\right), \\ \beta_m &= 3d_w m + 3C_0^2 + \frac{3C_h^2 C_0^2 m}{4} = O(d_w m). \end{aligned}$$

Reorganize them and we have

$$\begin{aligned} R_T &= O\left((m\sqrt{d_w m} + d_w m)\log\left(\frac{m}{\delta}\right)\right), \\ &= O\left((d_w \log(T))^{\frac{3}{2}}\sqrt{d_w} + d_w^2 \log(T))\log(d_w \log(T))\right), \\ &= O\left(d_w^2 \log^{\frac{3}{2}}(T)\log(d_w \log(T))\right), \end{aligned}$$

which completes the proof. ■

Lemma 28 (Restatement of Lemma 27) *The Algorithm 1 runs at most* $m = d_w \log\left(\frac{C_g^2 T^2}{d_w \lambda} + 1\right)$ *stages.*

Proof The proof follows the outline of that for Lemma 2 in [48]. First, we show that $\det(\Sigma_s) = 2\det(\Sigma_{s-1}), \forall s = 1, 2, \cdots, m$.

$$\begin{aligned} \det(\Sigma_s) &= \det\left(\Sigma_{s-1} + \frac{1}{\epsilon_s^2}\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)^\top\right), \\ &= \det\left(\Sigma_{s-1}^{1/2}\left(\mathbf{I}_{d_w} + \frac{1}{\epsilon_s^2}\Sigma_{s-1}^{-\frac{1}{2}}\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)^\top\Sigma_{s-1}^{-\frac{1}{2}}\right)\Sigma_{s-1}^{\frac{1}{2}}\right), \\ &= \det(\Sigma_{s-1})\det\left(\mathbf{I} + \frac{1}{\epsilon_s^2}\Sigma_{s-1}^{-\frac{1}{2}}\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)^\top\Sigma_{s-1}^{-\frac{1}{2}}\right), \\ &= \det(\Sigma_{s-1})\left(1 + \frac{1}{\epsilon_s^2}\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)^\top\Sigma_{s-1}^{-\frac{1}{2}}\Sigma_{s-1}^{-\frac{1}{2}}\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\right), \\ &= \det(\Sigma_{s-1})\left(1 + \frac{1}{\epsilon_s^2}\|\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\|_{\Sigma_{s-1}^{-1}}^2\right), \\ &= 2\det(\Sigma_{s-1}), \end{aligned}$$

where the fourth line follows from the matrix determinant lemma: $\det(\mathbf{A}+\mathbf{u}\mathbf{v}^\top) = (1+\mathbf{v}^\top\mathbf{A}^{-1}\mathbf{u})\det(\mathbf{A})$ where $\mathbf{A}$ is an invertible square matrix and $\mathbf{u}, \mathbf{v}$ are column vectors. Thus, $\det(\Sigma_m) = 2^m \det(\Sigma_0) = 2^m \lambda^{d_w}$.

On the other hand, note that

$$\Sigma_m = \lambda \mathbf{I}_{d_w} + \sum_{s=1}^{m} \frac{1}{\epsilon_s^2} \nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0) \nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)^\top,$$

and hence by cyclic property of matrix trace,

$$\mathrm{tr}(\Sigma_m) = d_w \lambda + \sum_{s=1}^{m} \frac{\|\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\|_2^2}{\epsilon_s^2}.$$

Then by Assumption 3 and the trace-determinant inequality: $d_w \cdot \det(\mathbf{A})^{\frac{1}{d_w}} \leq \mathrm{tr}(\mathbf{A})$, we have

$$d_w \cdot \lambda 2^{\frac{m}{d_w}} \leq d_w \lambda + \sum_{s=1}^{m} \frac{\|\nabla f_{\mathbf{x}_s}(\hat{\mathbf{w}}_0)\|_2^2}{\epsilon_s^2} \leq d_w \lambda + \sum_{s=1}^{m} \frac{C_g^2}{\epsilon_s^2},$$

and hence

$$\sum_{s=1}^{m} \frac{1}{\epsilon_s^2} \geq \frac{d_w \lambda}{C_g^2} (2^{\frac{m}{d_w}} - 1).$$

Note that during each stage s, we query the quantum oracle for $\frac{C_1}{\epsilon_s} \log \frac{m}{\delta}$ times. Then with $C_1 > 1$ and $\delta \in (0, 1/2]$, we have

$$\sum_{s=1}^{m} \frac{C_1}{\epsilon_s} \log \frac{m}{\delta} \geq \sum_{s=1}^{m} \frac{1}{\epsilon_s} \geq \sqrt{\sum_{s=1}^{m} \frac{1}{\epsilon_s^2}} \geq \frac{1}{C_g} \sqrt{d_w \lambda (2^{\frac{m}{d_w}} - 1)}. \tag{9}$$

Now we derive an upper bound on m by contradiction. Suppose $m > d_w \log \left(\frac{C_g^2 T^2}{d_w \lambda} + 1 \right)$. This implies that

$$\sum_{s=1}^{m} \frac{C_1}{\epsilon_s} \log \frac{m}{\delta} \geq \frac{1}{C_g} \sqrt{d_w \lambda (2^{\frac{m}{d_w}} - 1)} > T,$$

which is a contradiction. Therefore, Q-NLB-UCB algorithm has at most $d_w \log \left(\frac{C_g^2 T^2}{d_w \lambda} + 1 \right)$ stages. ■

C.3. Confidence Analysis

The previous regret builds upon the successful construction of the confidence ball Ball_s for each stage s, which is summarized in the following confidence analysis lemma.

Lemma 29 (Confidence bound of Q-NLB-UCB) *Suppose Assumptions 2, 3, and 4 hold and β_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 confidence ball Ball_s with probability at least $1 - \delta$, i.e., $\|\hat{\mathbf{w}}_s - \mathbf{w}^*\|_{\Sigma_s}^2 \leq \beta_s$.*

Remark 30 *The design of* Ball_s *is similar to that in LinUCB [1] and QLinUCB [48], but our choice of* β_s *is different.* β_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*.*

Proof Sketch. The proof has two steps. In the first step, we solve the regression problem defined in Eq. (4) to get a closed form solution of $\hat{\mathbf{w}}_s$. Then in the second step, we upper bound multiple terms in $\|\hat{\mathbf{w}}_s - \mathbf{w}^*\|_{\Sigma_s}^2$ and the final upper bound is chosen as the valid β_s.

Proof By setting the derivative of objective function in Eq. (4) w.r.t. $\mathbf{w}$ as 0, we find the optimal criterion is

$$0 = \lambda(\hat{\mathbf{w}}_s - \hat{\mathbf{w}}_0) + \sum_{i=0}^{s-1} \frac{1}{\epsilon_i^2}((\hat{\mathbf{w}}_s - \hat{\mathbf{w}}_0)^\top \nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0) + f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0) - y_i)\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0).$$

Rearrange the equation and we have

$$\begin{aligned}
\lambda(\hat{\mathbf{w}}_s - \hat{\mathbf{w}}_0) + \sum_{i=0}^{s-1} \frac{1}{\epsilon_i^2}(\hat{\mathbf{w}}_s - \hat{\mathbf{w}}_0)^\top \nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0) &= \sum_{i=0}^{s-1} \frac{1}{\epsilon_i^2}(y_i - f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0))\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0), \\
\lambda(\hat{\mathbf{w}}_s - \hat{\mathbf{w}}_0) + \sum_{i=0}^{s-1} \frac{1}{\epsilon_i^2}\hat{\mathbf{w}}_s^\top \nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0) &= \sum_{i=0}^{s-1} \frac{1}{\epsilon_i^2}\left(\hat{\mathbf{w}}_0^\top \nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0) + y_i - f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\right)\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0), \\
\left(\lambda \mathbf{I} + \sum_{i=0}^{s-1} \frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)^\top\right)\hat{\mathbf{w}}_s - \lambda\hat{\mathbf{w}}_0 &= \sum_{i=0}^{s-1} \frac{1}{\epsilon_i^2}\left(\hat{\mathbf{w}}_0^\top \nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0) + y_i - f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\right)\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0),
\end{aligned}$$

where the third line is due to definition of observation noise ϵ_i. By our choice of Σ_s (Eq. (3)) which is inevitable, we have

$$\Sigma_s \hat{\mathbf{w}}_s = \lambda\hat{\mathbf{w}}_0 + \sum_{i=0}^{s-1} \frac{1}{\epsilon_i^2}\left(\hat{\mathbf{w}}_0^\top \nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0) + y_i - f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\right)\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0),$$

And the closed form solution of $\hat{\mathbf{w}}_s$ is shown as:

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

Further, $\hat{\mathbf{w}}_s - \mathbf{w}^*$ can be written as

$$\begin{aligned}
\hat{\mathbf{w}}_s - \mathbf{w}^* &= \Sigma_s^{-1}\left(\sum_{i=0}^{s-1} \frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\left(\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)^\top \hat{\mathbf{w}}_0 + y_i - f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\right)\right) + \lambda\Sigma_s^{-1}\hat{\mathbf{w}}_0 - \Sigma_s^{-1}\Sigma_s\mathbf{w}^*, \\
&= \Sigma_s^{-1}\left(\sum_{i=0}^{s-1} \frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\left(\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)^\top \hat{\mathbf{w}}_0 + y_i - f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\right)\right) + \lambda\Sigma_s^{-1}(\hat{\mathbf{w}}_0 - \mathbf{w}^*) \\
&\quad - \Sigma_s^{-1}\left(\sum_{i=0}^{s-1} \frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)^\top\right)\mathbf{w}^*,
\end{aligned}$$

$$
\begin{aligned}
&= \Sigma_s^{-1}\left(\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_0}(\hat{\mathbf{w}}_0)\left(\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)^\top(\hat{\mathbf{w}}_0-\mathbf{w}^*)+f_{\mathbf{x}_i}(\mathbf{w}^*)-f_{\mathbf{x}_i}(\mathbf{w}^*)+y_i-f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\right)\right)\\
&\quad+\lambda\Sigma_s^{-1}(\hat{\mathbf{w}}_0-\mathbf{w}^*),\\
&= \Sigma_s^{-1}\left(\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)(\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)^\top(\hat{\mathbf{w}}_0-\mathbf{w}^*)+f_{\mathbf{x}_i}(\mathbf{w}^*)-f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0))\right)\\
&\quad+\Sigma_s^{-1}\left(\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)(y_i-f_{\mathbf{x}_i}(\mathbf{w}^*))\right)+\lambda\Sigma_s^{-1}(\hat{\mathbf{w}}_0-\mathbf{w}^*),\\
&= \Sigma_s^{-1}\left(\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\frac{1}{2}\|\mathbf{w}^*-\hat{\mathbf{w}}_0\|^2_{\nabla^2 f_{\mathbf{x}_i}(\tilde{\mathbf{w}})}\right)\\
&\quad+\Sigma_s^{-1}\left(\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)(y_i-f_{\mathbf{x}_i}(\mathbf{w}^*))\right)+\lambda\Sigma_s^{-1}(\hat{\mathbf{w}}_0-\mathbf{w}^*), \qquad (10)
\end{aligned}
$$

where the second line is again by our choice of Σ_s and the last equation is by the second order Taylor's theorem of $f_{\mathbf{x}_i}(\mathbf{w}^*)$ at $\hat{\mathbf{w}}_0$ where $\tilde{\mathbf{w}}$ lies between $\mathbf{w}^*$ and $\hat{\mathbf{w}}_0$.

Multiply both sides of Eq. (10) by $\Sigma_s^{\frac{1}{2}}$ and we have

$$
\begin{aligned}
\Sigma_s^{\frac{1}{2}}(\hat{\mathbf{w}}_s-\mathbf{w}^*) = \frac{1}{2}\Sigma_s^{-\frac{1}{2}}&\left(\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\|\mathbf{w}^*-\hat{\mathbf{w}}_0\|^2_{\nabla^2 f_{\mathbf{x}_i}(\tilde{\mathbf{w}})}\right)\\
&+\Sigma_s^{-\frac{1}{2}}\left(\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)(y_i-f_{\mathbf{x}_i}(\mathbf{w}^*))\right)+\lambda\Sigma_s^{-\frac{1}{2}}(\hat{\mathbf{w}}_0-\mathbf{w}^*).
\end{aligned}
$$

Take square of both sides and by inequality $(a+b+c)^2\le 3(a^2+b^2+c^2)$ and we obtain

$$
\begin{aligned}
\|\hat{\mathbf{w}}_s-\mathbf{w}^*\|^2_{\Sigma_s} \le\ & \underbrace{3\left\|\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)(y_i-f_{\mathbf{x}_i}(\mathbf{w}^*))\right\|^2_{\Sigma_s^{-1}}}_{(a)}+\underbrace{3\lambda^2\|\hat{\mathbf{w}}_0-\mathbf{w}^*\|^2_{\Sigma_s^{-1}}}_{(b)}\\
&+\underbrace{\frac{3}{4}\left\|\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\|\mathbf{w}^*-\hat{\mathbf{w}}_0\|^2_{\nabla^2 f_{\mathbf{x}_i}(\tilde{\mathbf{w}})}\right\|^2_{\Sigma_s^{-1}}}_{(c)}. \qquad (11)
\end{aligned}
$$

Now we bound terms (a), (b), (c) separately.

Bounding (a). Let

$$
\begin{aligned}
\mathbf{E}_s &= \mathrm{diag}(1/\epsilon_0^2,\cdots,1/\epsilon_{s-1}^2),\\
\mathbf{G}_s &= [\nabla f_{\mathbf{x}_0}(\hat{\mathbf{w}}_0),\cdots,\nabla f_{\mathbf{x}_{s-1}}(\hat{\mathbf{w}}_0)]^\top,\\
\mathbf{f}_s &= [f_{\mathbf{x}_0}(\mathbf{w}^*),\cdots,f_{\mathbf{x}_{s-1}}(\mathbf{w}^*)],\\
\mathbf{y}_s &= [y_0,\cdots,y_{s-1}],
\end{aligned}
$$

then (a) can be rewritten as:

$$(a) = 3\left\|\mathbf{G}_s^\top \mathbf{E}_s(\mathbf{f}_s - \mathbf{y}_s)\right\|_{\Sigma_s^{-1}}^2.$$

Define $\Gamma_s = \mathbf{E}_s^{\frac{1}{2}}(\mathbf{y}_s - \mathbf{f}_s) \in \mathbb{R}^s$. Then

$$\begin{aligned}
(a) &= 3\|\mathbf{G}_s^\top \mathbf{E}_s^{\frac{1}{2}}\Gamma_s\|_{\Sigma_s^{-1}}^2, \\
&= 3\Gamma_s^\top \mathbf{E}_s^{\frac{1}{2}}\mathbf{G}_s\Sigma_s^{-1}\mathbf{G}_s^\top \mathbf{E}_s^{\frac{1}{2}}\Gamma_s, \\
&\leq 3\|\Gamma_s\|_2^2 \cdot \|\mathbf{E}_s^{\frac{1}{2}}\mathbf{G}_x\Sigma_s^{-1}\mathbf{G}_s^\top \mathbf{E}_s^{\frac{1}{2}}\|_2, \\
&\leq 3s \cdot \|\Gamma_s\|_\infty^2 \cdot \mathrm{tr}(\mathbf{E}_s^{\frac{1}{2}}\mathbf{G}_x\Sigma_s^{-1}\mathbf{G}_s^\top \mathbf{E}_s^{\frac{1}{2}}), \\
&\leq 3s \cdot \mathrm{tr}(\Sigma_s^{-1}\mathbf{G}_s^\top \mathbf{E}_s\mathbf{G}_s), \\
&= 3s \cdot \mathrm{tr}(\mathbf{I}_{d_w} - \lambda\Sigma_s^{-1}), \\
&\leq 3d_w s,
\end{aligned}$$

where the first inequality is due to Cauchy-Schwarz inequality, the second inequality is by property of ℓ_2 norm, and the third inequality is due to $|f_{\mathbf{x}_i}(\mathbf{w}^*) - y_i| \leq \epsilon_i, \forall i \in [s]$ (Lemma 1), and definition of $\Sigma_s = \lambda\mathbf{I} + \mathbf{G}_s\mathbf{E}_s\mathbf{G}_s^\top$.

Bounding (b). Then (b) can be bounded as:

$$(b) \leq 3\lambda^2\|\Sigma_s^{-1}\|_{\mathrm{op}}\|\hat{\mathbf{w}} - \mathbf{w}^*\|_2^2 \leq 3\lambda\|\hat{\mathbf{w}} - \mathbf{w}^*\|_2^2 \leq \frac{3\lambda C_0^2}{T_0^2},$$

where the first inequality is by Holder's inequality, the second inequality is due to property of Σ_s and the last inequality follows from Eq. (2).

Bounding (c). Next, (c) can be rewritten and bounded as

$$\begin{aligned}
(c) &\leq \frac{3}{4}\left\|\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\|\nabla^2 f_{\mathbf{x}_i}(\tilde{\mathbf{w}})\|_{\mathrm{op}}\|\mathbf{w}^* - \hat{\mathbf{w}}_0\|_2^2\right\|_{\Sigma_s^{-1}}^2, \\
&\leq \frac{3}{4}\left\|\sum_{i=0}^{s-1}\frac{C_h C_0^2}{\epsilon_i^2 T_0^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\right\|_{\Sigma_s^{-1}}^2, \\
&= \frac{3C_h^2C_0^2}{4T_0^4}\left\|\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\right\|_{\Sigma_s^{-1}}^2, \qquad (12)
\end{aligned}$$

where the first inequality is due to Holder's inequality, the second inequality is due to Assumption 3 and Eq. (2). Further we can rewrite Eq. (12) as

$$\begin{aligned}
\frac{3C_h^2C_0^2}{4T_0^4}\left\|\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\right\|_{\Sigma_s^{-1}}^2 &= \frac{3C_h^2C_0^2}{4T_0^4}\left(\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\right)^\top \Sigma_s^{-1}\left(\sum_{j=0}^{s-1}\frac{1}{\epsilon_j^2}\nabla f_{\mathbf{x}_j}(\hat{\mathbf{w}}_0)\right), \\
&= \frac{3C_h^2C_0^2}{4T_0^4}\sum_{i=0}^{s-1}\sum_{j=0}^{s-1}\frac{1}{\epsilon_i^2\epsilon_j^2}\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)^\top\Sigma_s^{-1}\nabla f_{\mathbf{x}_j}(\hat{\mathbf{w}}_0),
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{3C_h^2C_0^2}{4T_0^4}\sum_{i=0}^{s-1}\sum_{j=0}^{s-1}\frac{1}{\epsilon_i^2\epsilon_j^2}\|\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\|_{\Sigma_s^{-1}}\|\nabla f_{\mathbf{x}_j}(\hat{\mathbf{w}}_0)\|_{\Sigma_s^{-1}},\\
&= \frac{3C_h^2C_0^2}{4T_0^4}\left(\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\|\nabla f_{\mathbf{x}_i}(\hat{\mathbf{w}}_0)\|_{\Sigma_s^{-1}}\right)\left(\sum_{j=0}^{s-1}\frac{1}{\epsilon_j^2}\|\nabla f_{\mathbf{x}_j}(\hat{\mathbf{w}}_0)\|_{\Sigma_s^{-1}}\right),\\
&= \frac{3C_h^2C_0^2}{4T_0^4}\left(\sum_{i=0}^{s-1}\frac{1}{\epsilon_i}\right)\left(\sum_{j=0}^{s-1}\frac{1}{\epsilon_j}\right),\\
&\leq \frac{3C_h^2C_0^2}{4T_0^4}\left(\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\right)\left(\sum_{i=0}^{s-1}1\right),\\
&\leq \frac{3C_h^2C_0^2s}{4T_0^4}\left(\sum_{i=0}^{s-1}\frac{1}{\epsilon_i^2}\right),\\
&\leq \frac{3C_h^2C_0^2sT^2}{4T_0^4},
\end{aligned}
$$

where the first and second inequalities are due to Cauchy-Schwarz inequality and the last inequality is by Lemma 7.

Combine (a), (b), and (c), and we have:

$$
\|\hat{\mathbf{w}}_s - \mathbf{w}^*\|_{\Sigma_s}^2 \leq 3d_w s + \frac{3\lambda C_0^2}{T_0^2} + \frac{3C_h^2C_0^2sT^2}{4T_0^4}, \tag{13}
$$

which reflects our choice of β_s. ■

C.4. Time Complexity Analysis

We show the time complexity of Q-NLB-UCB in Table 1 in comparison with Q-GP-UCB [16] and QMCKernelUCB [24]. The time complexity is analyzed in their quantum parts and classical parts, respectively. Let U_0 denote the cost for one query of quantum oracle, then all algorithms have quantum time complexity $O(TU_0)$ since they all run in T rounds.

However, in the classical part, we can see that our Q-NLB-UCB algorithm enjoys a better time complexity in terms of T. Note d_w denotes parameter dimension and d_x denotes input dimension. In Algorithm 1, line 3,4,5,6 and 7 are all classical steps, and their time complexity is dominated by inversion of the covariance matrix $\Sigma_s \in \mathbf{R}^{d_w \times d_w}$, which requires time complexity $O(d_w^3)$. Therefore,

Table 1: Time complexity comparison

Algorithm	Quantum part	Classical part
Q-GP-UCB [16]	$O(TU_0)$	$O(d_x^4(\log T)^4)$
QMCKernelUCB [24]	$O(TU_0)$	$O(d_x^4(\log T)^4)$
Q-NLB-UCB (ours)	$O(TU_0)$	$O(d_w^4 \log T)$

total classical time complexity of Q-NLB-UCB becomes $O(md_w^3) = O(d_w^4 \log T)$. While in Q-GP-UCB and QMCKernelUCB, the classical steps are dominated by the kernel matrix inversion, which takes $O(s^3)$, so the time complexity becomes $O(\sum_{s=1}^{m} s^3) = O(m^4) = O(d_x^4 (\log T)^4)$ in total.

Appendix D. Additional Experimental Results

D.1. Runtime Comparison

Table 2: Runtime on two synthetic functions (in seconds).

Algorithms	Rastrigin	Styblinski-Tang
Q-GP-UCB [16]	4629.6453	4139.2478
QMCKernelUCB [24]	3744.1179	2565.0690
Q-NLB-UCB (ours)	861.2402	919.7602

We report the runtime of three quantum bandit algorithms in seconds shown in Table 2. 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. 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.

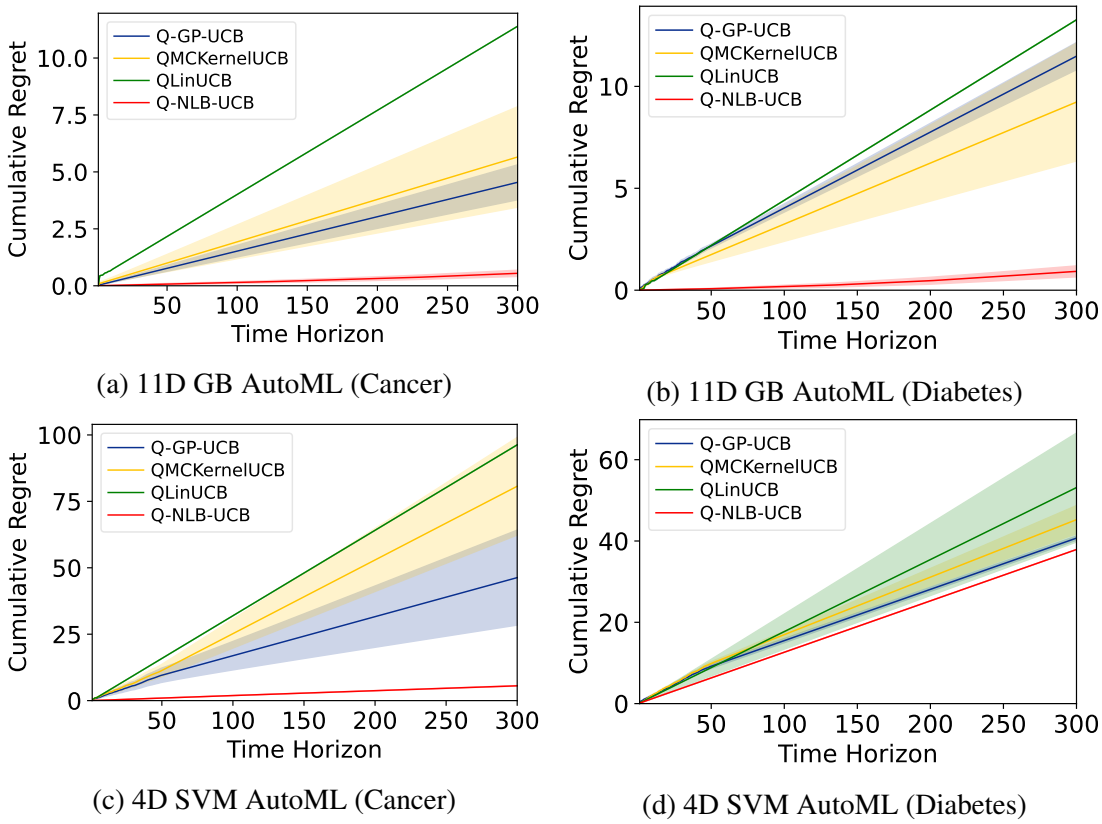
D.2. Real-World AutoML Results

In the main paper, we only show results on the MLP hyperparameter tuning tasks. Here in Figure 3, we show results on GB and SVM tasks. It again shows that our Q-NLB-UCB algorithm outperforms all the other algorithms by achieving significantly smaller cumulative regret on both datasets.

Our experiments reveal that Q-NLB-UCB can consistently find out the near-optimal Gradient Boosting and SVM hyperparameters while utilizing relatively few queries. The quantum amplitude estimation step in Q-NLB-UCB allows it to refine its hyperparameter selection in a way that balances exploration and exploitation very effectively. The final classification accuracies at the identified hyperparameters were very close to the true maximum of each dataset's domain, delineating the viability of Q-NLB-UCB as a quantum-inspired powerful tool for hyperparameter tuning on different real-world datasets.

D.3. Ablation Experiments on Quality of Quantum Mean Estimator

For Step 1 of Q-NLB-UCB in Algorithm 1, while many algorithms have been designed to solve quantum linear regression problems, quantum non-linear regression is still an open problem where no specific algorithm has been developed to achieve the quantum speed-up. Our work in Step 1 in Algorithm 1 proves the existence of a quantum regression oracle that solves this problem, shading the light to develop such a realizable algorithm in the future. Therefore, in experiments, we used classical gradient descent as the surrogate model for Step 1 in Algorithm 1. Even with that, our Q-NLB-UCB algorithm outperforms all other compared quantum bandit algorithms.



(a) 11D GB AutoML (Cancer)

(b) 11D GB AutoML (Diabetes)

(c) 4D SVM AutoML (Cancer)

(d) 4D SVM AutoML (Diabetes)

Figure 3: Cumulative regrets (the lower the better) of all compared quantum bandit algorithms.

Table 3: Quantum Implementation Results

Stage	**Selected action** x_s	**True reward** $f(x_s)$	**QME estimated** y_s	$\lvert y_s - f(x_s)\rvert$
1	$[-0.6, 0.5, -0.1]$	-40.6200	-40.6109	0.0091
2	$[-0.5, 0.4, 0]$	-38.5002	-38.5085	0.0083
3	$[-0.5, 0.3, 0]$	-33.4302	-33.4329	0.0027
4	$[-0.5, 0.3, 0.1]$	-35.3500	-35.3491	0.0009
5	$[-0.5, 0.3, 0.1]$	-35.3500	-35.3486	0.0014

Table 4: Classical Implementation Results

Stage	**Selected action** x_s	**True reward** $f(x_s)$	**Classical mean** y_s	$\lvert y_s - f(x_s)\rvert$
1	$[-0.6, 0.5, -0.1]$	-40.6200	-26.1172	14.5028
2	$[-0.7, 0.5, -0.1]$	-35.7500	-33.2826	2.4674
3	$[-0.6, 0.5, -0.1]$	-40.6200	-40.7809	0.1609
4	$[-0.5, 0.4, 0]$	-38.5002	-36.8721	1.6281
5	$[-0.5, 0.4, 0]$	-38.5002	-40.9617	2.4615

For Step 9 in Algorithm 1, we have conducted the ablation experiments, shown in Table 3 and Table 4. We can see that observations from quantum oracles are much closer to the true function value than observations from the classical query, showing great quality of QME.