Quantum Algorithm for Online Exp-concave Optimization

Jianhao He [1] **Chengchang Liu** [1] **Xutong Liu** [1] **Lvzhou Li** [2] **John C.S. Lui** [1]

Abstract

We explore whether quantum advantages can be found for the zeroth-order feedback online exp-concave optimization problem, which is also known as bandit exp-concave optimization with multi-point feedback. We present quantum online quasi-Newton methods to tackle the problem and show that there exists quantum advantages for such problems. Our method approximates the Hessian by quantum estimated inexact gradient and can achieve $O(n \log T)$ regret with $O(1)$ queries at each round, where n is the dimension of the decision set and T is the total decision rounds. Such regret improves the optimal classical algorithm by a factor of $T^{2/3}$.

1. Introduction

In this paper, we consider the problem of online convex optimization for exp-concave loss functions, which is also known as online exp-concave optimization. Online convex optimization is an important framework in online learning, and particularly useful in sequential decision making problems, such as online routing (Awerbuch & Kleinberg, 2008), portfolio selection (Hazan et al., 2007), and recommendation systems (Hazan & Kale, 2012). Online convex optimization can be represented as a T round iterative game between a player and an adversary. At every round $t \in [T]$, the player generates a decision x_t from a fixed and known convex set $\mathcal{K} \subseteq \mathbb{R}^n$, where n is the dimension of the decision set. The adversary observes x_t and chooses a convex loss function $f_t : \mathcal{K} \to \mathbb{R}$, with which the player suffers a loss of $f_t(x_t)$. Through this sequential process, some information about the loss function f_t is revealed to the player as feedback. The goal of the player is to minimize his *regret*, which is defined as

$$R(T) \triangleq \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x).$$

A good algorithm/strategy of the player should have a sublinear regret (that is, its regret is sublinear as a function of T) since this implies that as T grows, the accumulated loss of the algorithm converges to that with the best fixed strategy in hindsight (Hazan, 2016; Lattimore & Szepesvári, 2020).

We study the setting of exp-concave loss functions. This setting is also know as online exp-concave optimization (OXO). Note that OXO is useful in many machine learning problems, such as online supervised learning problems (Rakhlin et al., 2015; Bartlett et al., 2015; Gaillard et al., 2019), the portfolio selection problem (Luo et al., 2018; Mhammedi & Rakhlin, 2022; Jézéquel et al., 2022). The loss functions are assumed to be exp-concave, which is defined as:

Definition 1 (Exp-concave functions)**.** *A function $f : \mathcal{K} \to \mathbb{R}$ is α-exp-concave over a convex set $\mathcal{K}$ if $g : \mathcal{K} \to \mathbb{R}, g = e^{-\alpha f(x)}$ is a concave function, where $\alpha \geq 0$.*

Typical algorithms for this setting, such as the online Newton step proposed by Hazan (2007), can guarantee a regret bound of $O(n \log T)$ after T rounds. But the online Newton step method requires full information feedback, which means that the loss function is revealed to the player as feedback in each round, so that the player can obtain an exact gradient to approximate the Newton's direction. However, in many applications, the loss function is not always easy to obtain. When only partial information is revealed as feedback, the player has to make a prediction with limited information, which will often lead to worse regret. Settings with such feedback are called zeroth-order feedback settings, and the feedback is modeled as a zeroth-order oracle of the loss function which is only accessible to the player after making his decision in each round. The zeroth-order feedback setting is well studied for the online convex optimization problem, as shown in the extended related works in Appendix A.1. But for online exp-concave optimization problems, it remains open. Liu et. al. (2018) proposed an online Newton step algorithm with estimated gradient,

[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China [2]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. Correspondence to: Lvzhou Li <lilvzh@mail.sysu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

where the gradient is estimated by an unbiased estimator, and achieved an expected regret of $O\left(nT^{2/3}\log T\right)$. However, there are still much room for improvement as the existing OXO work of zeroth-order feedback setting only achieves sub-optimal regret.

In the past decade, quantum computing techniques have been applied to accelerate many optimization problems, readers are referred to the extended related works in Appendix A.2 for a briefly review. Online convex optimization problems can also be addressed using quantum computing. In 2022, He et. al. (2022) studied the quantum algorithm for online convex optimization of zeroth-order feedback setting, which achieves a regret bound of $O(\sqrt{T})$ that is independent of the dimension of the decision set. Under the same setting, this outperforms the optimal classical algorithms which have a square root dependence of the dimension. Motivated by this, in this paper, we present quantum algorithms for the online exp-concave optimization problem with the multi-query bandit setting and we explore whether quantum advantages can be achieved.

Problem Setting. Here, an algorithm is allowed to query the zeroth-order oracle multiple times after committing a prediction so to receive feedback in each round. A classical zeroth-order oracle O_f to a loss function f, queried with a vector $x \in \mathcal{K}$, outputs $O_f(x) = f(x)$. A quantum zeroth-order oracle Q_f is a unitary transformation that maps a quantum state $|x\rangle|q\rangle$ to the state $|x\rangle|q+f(x)\rangle$, where $|x\rangle$, $|q\rangle$ and $|q+f(x)\rangle$ are basis states corresponding to the floating-point representations of x, q and $q+f(x)$ respectively. Moreover, given the superposition input $\sum_{x,q}\alpha_{x,q}|x\rangle|q\rangle$, by linearity, the quantum oracle will output the state $\sum_{x,q}\alpha_{x,q}|x\rangle|q+f(x)\rangle$. Note that we do not need to limit the power of the adversary, namely, the adversary in our setting can be *completely adaptive*, and the adversary can choose f_t after observing the player's choice x_t. We only assume that both the player and the adversary are quantum enabled, which means that the adversary returns a quantum oracle as feedback and the player can use a quantum computer and query the oracle with a superposition input.

In addition, we make the following assumptions which are common in online convex optimization. We assume the loss functions are α-exp-concave and G-Lipschitz continuous, (i.e. $|f_t(x) - f_t(y)| \leqslant G\|y-x\|, \quad \forall x, y \in \mathcal{K}$). We also assume that the feasible set $\mathcal{K}$ is bounded and its diameter has an upper bound D, that is, $\forall x, y \in \mathcal{K}, \|x-y\|_2 \leq D$. $\mathcal{K}, D, G$ are known to the player. In addition, to avoid the mathematical proof in this paper being too lengthy and tedious, the loss functions are assumed to be β-smooth directly. The case where the smoothness assumption is not satisfied can be solved by using the mollification technique as shown in (He et al., 2022; Chakrabarti et al., 2020). We will show how to extend our work to non-smooth case in Section 5.3.

Contribution. The contribution of this paper is threefold.

- We propose a quantum randomized algorithm that can achieve the regret bound $O\left(\left(DG+\frac{1}{\alpha}\right)n\log(T+\log T)\right)$, by querying the oracle $O(1)$ times in each round (Theorem 1), which means that our quantum algorithm outperforms the known optimal classical algorithm. Specifically, the regret of our quantum algorithm improves the optimal classical algorithm (Liu et al., 2018) by a factor of $T^{2/3}$, as shown in Table 1.

- We extend the analysis of online Newton step method for the situation that the gradient is inexact but the error is controllable and can be by ℓ_1 norm. This is the theoretical basis for selecting the parameters of quantum circuits and the learning rate appropriately. This technique also implies that there exists a classical algorithm with finite difference method can achieve a logarithm regret by querying the oracle $O(n)$ times in each round, as finite difference method can give an error controllable estimated gradient bounded by ℓ_1 norm as well.

- We generalize our methods and propose quantum adaptive gradient methods and quantum online Newton methods with Hessian update respectively. The analysis of quantum adaptive gradient methods is closely related to online Newton methods. For the later extension, we show how to extend the quantum gradient estimation algorithm to the quantum Hessian estimation algorithm for the case that the quantum first-order oracle is accessible. Noticing that there is no 'real' online Newton method yet, namely, there is no classical online algorithm which use the Hessian to design the update rule. Methods which approximate the Newton direction by the gradient information are generally called the quasi-Newton methods in offline optimization problems, but the word 'quasi' is usually omitted in online optimization problems. To fill this void, we propose a framework of online Newton method with Hessian update, and show that it can achieve logarithm regret with appropriate parameters (Theorem 2). We also generalize our methods to the non-smooth case (Theorem 3).

The remainder of this paper is organized as follows. Section 2 introduces the basic concept of quantum computing and basic framework of online Newton step method of (Hazan et al., 2007). Section 3 presents the quantum online quasi-Newton method, where the parameter choosing

Table 1. Regret bound for online exp-concave optimization.

Paper	Feedback	Regret
(Hazan et al., 2007)	Full information (first-order)	$O\left(DGn\log T\right)$
(Liu et al., 2018)	Zeroth-order	$O\left(DGnT^{2/3}\log T\right)$
This work	Zeroth-order	$O\left(DGn\log(T+\log T)\right)$

- D is the diameter upper bound of the feasible set, G is the Lipschitz constant, n is the dimension of the feasible set, T is the total decision rounds.

and the regret analysis of the algorithm are given in Section 4. Section 5.1 gives the idea about extending our work to quantum adaptive gradient method. Section 5.2 gives the online Newton method with Hessian update and its regret analysis, accompanying with the quantum Hessian estimation algorithm. Section 5.3 shows how to extend our work to non-smooth loss functions. Finally, we conclude with a discussion in Section 6. The extended related works of online convex optimization and quantum optimization are placed in Appendix A. The proof details are given in Appendix B.

2. Preliminaries

2.1. Notations

Quantum computing can be described with Dirac notations. We denote the computational basis of $\mathbb{C}^d$ as $\{|i\rangle\}_{i=0}^{d-1}$, where $|i\rangle$ is a d-dimensional vector with 1 in the ith entry and 0 in other entries. A d-dimensional quantum state can be described with an unit vector $|v\rangle = (v_1, v_2, \ldots, v_d)^T = \sum_i v_i |i\rangle \in \mathbb{C}^d$ where v_i is called the amplitude of $|i\rangle$ and $\sum_i |v_i|^2 = 1$. The composite system of quantum systems can be described by tensor product of quantum states, that is, for $|v\rangle \in \mathbb{C}^{d_1}, |u\rangle \in \mathbb{C}^{d_2}$, the composite state of this two states is $|v\rangle \otimes |u\rangle = (v_1u_1, v_1u_2, \ldots . v_2u_1, \ldots, v_{d_1}u_{d_2}) \in \mathbb{C}^{d_1 \times d_2}$. The notation $\otimes$ is often omitted when there is no ambiguity. The evolution of a closed quantum system can be described by unitary transformations. When we say 'measure a quantum state', we mean measuring the state in computational basis unless otherwise stated. The measurement will give one of the state in the computational basis with the probability of the square of the magnitude of its amplitude. For example, if we measure $|v\rangle = \sum_i v_i |i\rangle$, we will get i with probability $|v_i|^2$, and the state will collapse in $|i\rangle$ after measurement, for all i.

We also introduce the basic notations for the optimization methods as follows. The horizon, namely, the total decision rounds is denoted by T, and in order to distinguish it from the notation of matrix transpose, the notation of matrix transpose is denoted by Room letter T. The decision set is denoted by $\mathcal{K} \subseteq \mathbb{R}^n$, where n is the dimension of the decision set and its diameter has an upper bound D. The decision made by the player in round t is denoted by x_t. The loss function chose by the adversary in round t is denoted by f_t, the corresponding gradient is denoted by ∇f_t and the Hessian is denoted by $H_t(f_t)$. The Lipschitz parameters of the loss functions and its gradient functions are denoted by G and L, respectively. We use $\|\cdot\|$ to present the Euclidean norm of vector, and given a positive definite matrix A, we use $\|x\|_A \triangleq \sqrt{x^{\mathrm{T}}Ax}$ to present the weighted norm of vector x. The projection according to the weighted norm defined by the matrix A is denoted as $\hat{P}_{\mathcal{K}}^{(A_t)}$:

$$\hat{P}_{\mathcal{K}}^{(A_t)}(y) \triangleq \arg\min_{x\in\mathcal{K}} \|x-y\|_{A_t}. \tag{1}$$

2.2. Basic Concepts and Frameworks

In this subsection, we introduce basic concept of quantum circuit and the framework of classical online Newton methods.

Analogous to the way a classical computer is built from an electrical circuit containing wires and logic gates, a quantum computer is built from a quantum circuit containing wires and elementary quantum gates to carry around and manipulate the quantum information (Nielsen & Chuang, 2002). Thus, the quantum circuit is an efficient way to describe quantum algorithms. The basic idea of a quantum circuit is as follows: horizontal lines are used to represent quantum bits (or quantum registers). In general, single lines for quantum bits (registers) and double lines for classical information. Symbols within a box are used to represent quantum gates and quantum measurements, with the position of the box representing the quantum bits to be executed. Time flows from left to right, and the quantum circuit is executed from left to right. Note that any classical circuit can be replaced by an equivalent circuit containing only reversible elements, and thus can be simulated using a quantum circuit.

We present the online Newton methods (Hazan et al., 2007) in Algorithm 1, which is a basic framework for solving the online exp-concave optimization problem. It starts with a random decision, generates the decision of the next round by moving the current decision in the quasi-Newton direction of the current loss function and then projecting to the

decision set, where the matrix A_t in the update rule is updated by a rank-1 matrix generated by the exact gradient information. With the exact gradient information, the online Newton step method can achieve logarithm regret.

Algorithm 1 Online Newton Step Method (Hazan et al., 2007)

Input: Step size γ, parameter ϵ.
Output: $x_1, x_2, x_3, \ldots x_T$

1: Choose the initial point $x_1 \in \mathcal{K}$ randomly, let $A_0 = \epsilon I_n$.
2: **for** $t = 1$ to T **do**
3: Play x_t, observe the loss function f_t from the adversary.
4: Update $x_{t+1} = \hat{P}_{\mathcal{K}}^{(A_t)}\left(x_t - \frac{1}{\gamma}A_t^{-1}\nabla f_t(x_t)\right)$, where $A_t = A_{t-1} + \nabla f_t(x_t)(\nabla f_t(x_t))^{\mathrm{T}}$.
5: **end for**

3. Quantum Online Quasi-Newton Method

In this section, for the OXO problem stated in Section 1, given the total horizon T and δ, we present Algorithm 2 to produce a decision sequence $x_1, x_2, x_3, \ldots, x_T$ for the player, such that with probability greater than $1 - \delta$, it achieves a regret being logarithm of T. Specifically, δ is divided into a series of parameters ρ_t which are intermediate parameters used to adjust the success probability of the quantum gradient estimation (Lemma 1).

Initially, the algorithm chooses x_1 randomly from $\mathcal{K}$, and then sequentially produces $x_2, x_3, \ldots, x_T$ by online Newton step method. Steps 4-11 are the process of quantum gradient estimation, where definitions of basic quantum gates/circuit is given in Appendix B.1, and the quantum circuit is illustrated in Figure 2. The quantum circuit of Q_{F_t} in Step 6 is constructed by using Q_{f_t} twice; Since the evolution of a quantum operation is reversible, the Q_F^{-1} can be operationally realized by reversing the inputs and outputs of Q_F. $\mathbb{1}$ in Step 6 is the n-dimensional all 1's vector; the last register and the operation of addition modulo 2^c in Step 7 are used for implementing the common technique in quantum algorithm known as phase kickback which adds a phase shift related to the oracle; Step 8 is known as uncompute trick which recovers the ancillary registers to the initial states so that they can be reused in the next iterative.

For the update rule, if one needs to compute the invert of A_t for each round t, it will incur a complexity of $O(n^3)$. But with the help of Sherman-Morrison-Woodbury Formula (1950; 1950),

$$\begin{aligned} A_t^{-1} &= \left(A_{t-1} + \widetilde{\nabla} f_t(x_t)(\widetilde{\nabla} f_t(x_t))^T\right)^{-1} \\ &= A_{t-1}^{-1} - A_{t-1}^{-1}\widetilde{\nabla} f_t(x_t) \\ &\quad \times \left(1 + (\widetilde{\nabla} f_t(x_t))^T A_{t-1}^{-1}\widetilde{\nabla} f_t(x_t))\right)^{-1} \\ &\quad \times (\widetilde{\nabla} f_t(x_t))^T A_{t-1}^{-1}, \end{aligned} \tag{2}$$

the computation complexity can be reduced to $O(n^2)$. The initial A_0 is a identity matrix multiplied by a constant, thus A_0^{-1} can be computed in $O(n)$.

Algorithm 2 Quantum Online Quasi-Newton Method (QONS)

Input: Step sizes $\{\eta_t\}$, parameters $\{r_t\}, \rho, \epsilon$.
Output: $x_1, x_2, x_3, \ldots x_T$

1: Choose the initial point $x_1 \in \mathcal{K}$ randomly, let $A_0 = \epsilon I_n$.
2: **for** $t = 1$ to T **do**
3: Play x_t, get the oracle of loss function Q_{f_t} from the adversary.
4: Prepare the initial state: n b-qubit registers $\left|0^{\otimes b}, 0^{\otimes b}, \ldots, 0^{\otimes b}\right\rangle$ where $b = \log_2 \frac{G\rho}{4\pi n^2 \beta r_t}$. Prepare 1 c-qubit register $|0^{\otimes c}\rangle$ where $c = \log_2 \frac{16\pi n}{\rho} - 1$. And prepare $|y_0\rangle = \frac{1}{\sqrt{2^n}}\sum_{a\in\{0,1,\ldots,2^n-1\}} e^{\frac{2\pi i a}{2^n}}|a\rangle$.
5: Apply Hadamard transform to the first n registers.
6: Perform the quantum query oracle Q_{F_t} to the first $n+1$ registers, where $F_t(u) = \frac{2^b}{2Gr_t}\left[f_t\left(x_t + \frac{r_t}{2^b}\left(u - \frac{2^b}{2}\mathbb{1}\right)\right) - f_t(x_t)\right]$, and the result is stored in the $(n+1)$th register.
7: Perform the addition modulo 2^c operation to the last two registers.
8: Apply the inverse evaluating oracle $Q_{F_t}^{-1}$ to the first $n+1$ registers.
9: Perform quantum inverse Fourier transformations to the first n registers separately.
10: Measure the first n registers in computation bases respectively to get $m_1, m_2, \ldots, m_n$.
11: $\widetilde{\nabla} f_t(x_t) = \frac{2G}{2^b}\left(m_1 - \frac{2^b}{2}, m_2 - \frac{2^b}{2}, \ldots, m_n - \frac{2^b}{2}\right)^{\mathrm{T}}$.
12: Update $x_{t+1} = \hat{P}_{\mathcal{K}}^{(A_t)}\left(x_t - \frac{1}{\eta_t}A_t^{-1}\widetilde{\nabla} f_t(x_t)\right)$, where $A_t = A_{t-1} + \widetilde{\nabla} f_t(x_t)(\widetilde{\nabla} f_t(x_t))^{\mathrm{T}}$.
13: Bitwise erase the first n registers with control-not gates controlled by the corresponding classical information of the measurement results $m_1, m_2, \ldots, m_n$.
14: **end for**

We now analyze the query complexity of Algorithm 2. In each round, it needs to call the oracle twice to construct

Q_F, and twice to perform the uncompute step Q_F^{-1}, so totally 4 times for computing the gradient. Thus, $O(1)$ times for each round. Then, the error bound of the estimated gradient is shown in Lemma 1, and the proof is given in Appendix B.1.

Lemma 1. *In Algorithm 2, for all timestep t, if f_t is β-smooth function, then for any $r_t > 0$ and $1 \geq \rho_t > 0$, the estimated gradient $\widetilde{\nabla} f_t(x_t)$ satisfies*

$$\Pr\left[\|\nabla f_t(x_t) - \widetilde{\nabla} f_t(x_t)\|_1 > 8\pi n^3 (n/\rho_t + 1)\beta r_t/\rho_t\right] < \rho_t. \tag{3}$$

The framework of quantum gradient estimator originates from Jordan's quantum gradient estimation method (Jordan, 2005), but Jordan's algorithm did not give any error bound because the analysis of it was given by omitting the high-order terms of Taylor expansion of the function directly. In 2019, the quantum gradient estimation method with error analysis was given in (Gilyén et al., 2019), and was applied to the general convex optimization problem (van Apeldoorn et al., 2020; Chakrabarti et al., 2020). In those case, however, $O(\log n)$ repetitions were needed to estimate the gradient within a acceptable error, which means that the query complexity to the zeroth-order oracle is $O(\log n)$. We improve the analysis of the quantum gradient estimation method in our previous work (He et al., 2022), and show that $O(1)$ queries is sufficient in our problem, instead of $O(\log n)$ repetitions. The key technique is, for each coordinate, the failure probability of a single repetition can be made small at the expense of a weaker quality of approximation, and then the worse approximation guarantee can be fixed by choosing a finer grid.

4. Parameters Selection and Regret Analysis

In this section, we show how to choose appropriate parameters such that Algorithm 2 guarantees $O\left(\left(DG + \frac{1}{\alpha}\right) n \log(T + \log T)\right)$ regret, which gives Theorem 1. Firstly, in each round, we need to bound the difference between the loss suffered by the player and the function value of best fixed strategy x^*, but the latter one is unknown to us when analyzing the regret bound, so we set to prove a stronger property that holds for any points in the decision set as shown in Lemma 2. Then, let this 'any point' be x^* will give what we need. Secondly, summing up these inequalities for all T round will give the regret bound, and we need to choose appropriate parameters to bound every term in $O(\log T)$ so to make the regret as small as possible. Since the gradient estimator is not accurate, it leads to more challenging analysis.

Lemma 2. *In Algorithm 2, for all timestep t, let $f_t : \mathcal{K} \to \mathbb{R}$ be α-exp-concave functions with Lipschitz parameter G, where $\mathcal{K}$ is a convex set with diameter D, then for any $y \in \mathcal{K}$ and any $\eta_t \leq \min\left\{\frac{1}{8GD}, \frac{\alpha}{2}\right\}$, the estimated gradient $\widetilde{\nabla} f_t(x_t)$ satisfies*

$$\begin{aligned} f_t(y) \geq & f_t(x_t) + \widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y - x_t) \\ & + \frac{\eta_t}{2}(y - x_t)^{\mathrm{T}} \widetilde{\nabla} f_t(x_t) \widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y - x_t) \\ & - \frac{9D}{8}\left\|\nabla f_t(x_t) - \widetilde{\nabla} f_t(x_t)\right\|_1 \\ & - \frac{D}{16G}\left\|\nabla f_t(x_t) - \widetilde{\nabla} f_t(x_t)\right\|_1^2. \end{aligned} \tag{4}$$

The proof is given in Appendix B.2. In the following, we show that with appropriate parameters, Algorithm 2 can guarantee logarithm regret.

Theorem 1. *Algorithm 2 with parameters $\eta = \min\left\{\frac{1}{8GD}, \frac{\alpha}{2}\right\}$, $\epsilon = \frac{1}{\eta^2 D^2}$, $\left\{r_t = \frac{\rho G}{\pi n^3 (n/\rho + 1)\beta t}\right\}_{t=1}^{T}$, can achieve the regret bound $O\left(\left(DG + \frac{1}{\alpha}\right) n \log(T + \log T)\right)$ with probability greater than $1 - T\rho$, and its query complexity is $O(1)$ in each round.*

The proof is given in Appendix B.3. Replacing $\{r_t\}$ into b, we have $b = \log_2 \frac{n(n+\rho)t}{4\rho} = O(\log(Tn/\delta))$, and $c = \log_2 \frac{16\pi n}{\rho} - 1 = O(\log(Tn/\delta))$, where $\delta = T\rho$ is the total failure probability we set for the algorithm. Thus, $O(n \log(Tn/\delta))$ qubits are needed totally.

5. Extensions

We present how to extend the quantum online quasi-Newton methods to quantum adaptive gradient method and quantum online Newton method in this section.

5.1. Quantum Adaptive Gradient Method

The adaptive gradient method (AdaGrad) proposed by Duchi et. al. (2011) is widely applied for deep neural networks. It has similar structure as online Newton methods, hence we adapt it as an extension of this work. The adaptive gradient method has nearly the same update rule as online Newton methods, except that the matrix in quasi-Newton direction is replaced by $G_t = (A_t)^{1/2}$. Note that it is computationally impractical to compute the root of the the outer product matrix G_t, Duchi et. al. (2011) specialized G_t to a diagonal matrix $\operatorname{diag}(A_t)^{1/2}$. Then, both the inverse and root of the diagonal matrix can be computed in linear time so that the AdaGrad method has the same computational complexity as the first order of gradient method, and thus can be regarded as a modification of gradient descent. It can also be generalized to $G_t = \operatorname{diag}(A_t)^{1-p}$ for $p \in [0.5, 1]$, which includes both the AdaGrad method and gradient descent method.

For those cases where only the zeroth-order oracle is accessible, we can use the $O(1)$-query quantum gradient es-

timation method to accelerate the calculation and keep the regret bound inline with the full information setting, which gives the quantum adaptive gradient method as shown in Algorithm 3.

Algorithm 3 Quantum Adaptive Gradient Method

Input: Step sizes γ, parameters $p \in [0.5, 1], \epsilon, \{r_t\}$.
Output: $x_1, x_2, x_3, \dots x_T$

1: Choose the initial point $x_1 \in \mathcal{K}$ randomly, let $A_0 = \epsilon I_n$.
2: **for** $t = 1$ to T **do**
3: Play x_t, get the oracle of loss function Q_{f_t} from the adversary.
4: Use the quantum gradient estimation circuit with the parameter r_t to get the estimated gradient $\widetilde{\nabla} f_t(x_t)$.
5: Let $A_t = A_{t-1} + \widetilde{\nabla} f_t(x_t)(\widetilde{\nabla} f_t(x_t))^{\mathrm{T}}$.
6: Update $x_{t+1} = \hat{P}_{\mathcal{K}}^{(G_t)}\left(x_t - \frac{1}{\gamma} G_t^{-1} \widetilde{\nabla} f_t(x_t)\right)$, where $G_t = \mathrm{diag}\,(A_t)^{1-p}$.
7: **end for**

Although the method is called the gradient method, its analysis is closely related to online Newton methods. With similar analysis, appropriate parameters can be chosen to guarantee the following regret:

$$\begin{aligned}&\sum_{t=1}^{T}\left(f_t(x_t) - f_t(x^*)\right)\\ \leq &\frac{1}{2p\gamma}\,\mathrm{tr}\,(G_T^p) + \frac{D_q^2\gamma}{2}\,\mathrm{tr}\,(G_T^p)^{(1-p)/p}\\ \leq &O\left(D_q\sqrt{T}\right),\end{aligned} \tag{5}$$

where D_q is the upper bound of the ℓ_q-diameter of the decision set, and $q = 2p/(2p-1)$. Note that this algorithm does not require the assumption of exponential concavity.

5.2. Quantum Online Newton Method with Hessian Update

Since the estimation of each row of a Hessian matrix can be seen as a gradient estimation process of one dimension of the gradient function, the quantum Hessian estimation algorithm with the first-order oracle can be designed using the same framework of the quantum gradient estimation algorithm with the zeroth-order oracle. It outputs one row of the Hessian matrix per execution, thus $O(n)$ repeats of it can output a error controllable estimated Hessian matrix. Here, with the quantum first-order oracle, given the superposition input, one can receive a superposition state of corresponding gradients at different points. Figure 1 gives the quantum circuit of estimating the i-th row of the Hessian matrix. Running such circuit for all $i \in [n]$, we will have a estimation of the Hessian.

Compared with the quantum gradient estimation circuit, the sub-circuits Q_{F_t} and $Q_{F_t}^{-1}$ are replaced by the first-order version $Q_{\nabla F_t}$ and $Q_{\nabla F_t}^{-1}$, respectively, where $\nabla F_t(u) = \frac{2^b}{2Lr_t}\left[\nabla f_t\left(x_t + \frac{r_t}{2^b}\left(u - \frac{2^b}{2}\mathbb{1}\right)\right) - \nabla f_t(x_t)\right]$, which needs n c-qubit registers to store the output. Here, L is the Lipschitz constant of the gradient function, and the other symbols have the same meaning as those in the quantum gradient estimation method. Before and after the modular addition, swap gates between the i-th c qubits register and the last c qubits register are added. Then, the estimation of the element in i-th row and j-th column of the Hessian matrix is given by $\widetilde{H}_t^{(i,j)} = \frac{2L}{2^b}\left(m_j - \frac{2^b}{2}\right)$, where m_j is the measurement result of the j-th b-qubit register, for all $j \in [n]$.

Noticing that there is no online algorithm which uses the Hessian information to design the update rule. Thus, we also give a classical framework online Newton method with Hessian update, and show that it can achieve logarithm regret. We still study the online exp-concave optimization problem, but assume here the player can get the gradient and the Hessian of the loss function in each round efficiently after suffering the loss. Consequentially, the loss functions should assume to be twice differentiable. We assume that the gradient of the loss functions is L-Lipschitz continuous, that is, $|\nabla f_t(x) - \nabla f_t(y)| \leqslant L\|y - x\|, \quad \forall x, y \in \mathcal{K}$. Similar to the exp-concave assumption, we further assume that $\forall x, y \in \mathcal{K}, f_t(y) \geq f_t(x) + \nabla f_t(x)^{\mathrm{T}}(y-x) + \frac{\eta}{2}(y-x)^{\mathrm{T}} H(f_t)(x)(y-x)$, for all $t \in [T]$, where $H(f_t)(x)$ is the Hessian of the loss function f_t at point x. The right-hand side of this assumption is the Taylor expansion of the loss function omitting high-order terms and multiplying a factor to the second-order term, and thus will be easy to satisfy when η is small since the Hessian of a convex function is positive semi-definite.

Algorithm 4 Online Newton Method with Hessian Update

Input: Step sizes η, parameters ϵ.
Output: $x_1, x_2, x_3, \dots x_T$

1: Choose the initial point $x_1 \in \mathcal{K}$ randomly, let $A_0 = \epsilon I_n$.
2: **for** $t = 1$ to T **do**
3: Play x_t, observe the loss function f_t from the adversary.
4: Update $x_{t+1} = \hat{P}_{\mathcal{K}}^{(A_t)}\left(x_t - \frac{1}{\eta} A_t^{-1} \nabla f_t(x_t)\right)$, where $A_t = A_{t-1} + H_t$.
5: **end for**

We set $x_{t+1} = \hat{P}_{\mathcal{K}}^{(A_t)}\left(x_t - \frac{1}{\eta} A_t^{-1} \nabla f_t(x_t)\right)$ as the update rule, where $A_t = A_{t-1} + H(f_t)(x_t)$. Below we will write

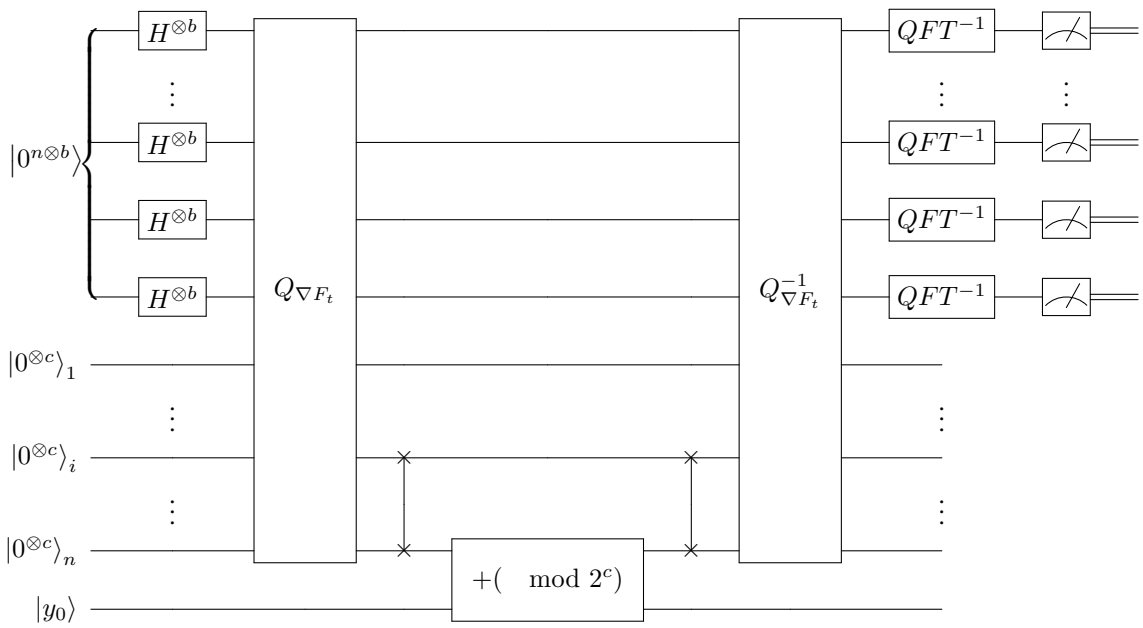

Figure 1. Quantum circuit of estimating the i-th row of the Hessian matrix.

the Hessian $H(f_t)(x_t)$ as H_t for short. The algorithm is presented in Algorithm 4, and then we show how to choose appropriate parameters such that Algorithm 4 guarantees logarithm regret, which gives Theorem 2.

Theorem 2. *Algorithm 4 with parameters $\eta = \frac{1}{2DL}$ and $\epsilon = L$, can achieve the regret bound $O\left(\frac{DLn}{\alpha}\log(T+1)\right)$.*

The proof is given Appendix in B.4.

5.3. Extend to non-smooth loss functions

In this section, we present how to extend Algorithm 2 to non-smooth loss functions. It has been shown that Lipschitz continue functions are smooth in a small region with high probability (Chakrabarti et al., 2020). However, adopting this technique necessitates a few additional steps. In their proof, they shows that non-smooth loss functions are still smooth in a small region with high probability by bounding the trace of the Hessian matrices. As they do not assume that the loss functions are second-order differentiable (nor do we), they need to use the mollifier functions to approximate the loss functions. Yet, for certain loss functions, the approximation error becomes uncontrollable. To address this challenge, they introduce classical randomness into the gradient evaluation, that is, sampling randomly from the neighborhood of any given point, and then use the gradient of the sample point as the gradient of the given point. We denote the sample point as $z_t \in B_\infty(x_t, r'_t)$, where $B_\infty(x_t, r'_t)$ is the ball in L_∞ norm with radius $r'_t \in \mathbb{R}$ and center x_t. Lemma 3 shows that for any sample point z_t, the trace of the Hessian matrices is small in a region with high probability, and therefore we can let $\beta = nG/p_t r'_t$.

Lemma 3. *((Chakrabarti et al., 2020)) In Algorithm 2, for all timestep t, let $f_t : \mathcal{K} \to \mathbb{R}$ be a convex function with Lipschitz parameter G. Then for any $r_t, r'_t > 0$ and $1 \geq p_t > 0$, we have*

$$\Pr_{z_t \in B_\infty(x_t, r'_t)}\left[\exists y \in B_\infty(z_t, r_t), \mathrm{Tr}\left\{\nabla^2 f_t(y)\right\} \geq \frac{nG}{p_t r'_t}\right] \leq p_t. \tag{6}$$

By combining the observation that in the proof of the quantum gradient estimation, the smooth condition is only used in a small region, we can give a process (for each round) to extend our algorithm to the non-smooth case, as follows:

1 Sample $z_t \in B_\infty(x_t, r'_t)$.

2 Apply the quantum gradient estimation circuit on point z_t to get the estimated gradient $\widetilde{\nabla} f_t(z_t)$, let $\widetilde{\nabla} f_t(x_t) = \widetilde{\nabla} f_t(z_t)$

3 Update $x_{t+1} = \hat{P}_{\mathcal{K}}^{(A_t)}\left(x_t - \frac{1}{\eta_t} A_t^{-1} \widetilde{\nabla} f_t(x_t)\right)$, where $A_t = A_{t-1} + \widetilde{\nabla} f_t(x_t)(\widetilde{\nabla} f_t(x_t))^{\mathrm{T}}$.

Then, by Lemma 1, we will establish the error bound of the form $\left\|\nabla f_t(z_t) - \widetilde{\nabla} f_t(z_t)\right\|_1$. However, Lemma 2 is

no longer applicable here as it requires the error bound of $\left\|\nabla f_t(x_t) - \widetilde{\nabla} f_t(x_t)\right\|_1$. To address this, we give Lemma 4. Initially, we utilize the property of exp-concavity at the point z_t, which yields a relationship between y and z_t. Subsequently, we leverage this relationship to establish a connection between y and x_t.

Lemma 4. *In the non-smooth version of Algorithm 2, for all timestep t, let $f_t : \mathcal{K} \to \mathbb{R}$ be α-exp-concave functions with Lipschitz parameter G, where $\mathcal{K}$ is a convex set with diameter D, then for any $y \in \mathcal{K}$ and any $\eta_t \leq \min\left\{\frac{1}{8GD}, \frac{\alpha}{2}\right\}$, the estimated gradient $\widetilde{\nabla} f_t(x_t)$ satisfies*

$$\begin{aligned} f_t(y) \geq & f_t(x_t) + \widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y - x_t) \\ & + \frac{\eta_t}{2}(y - x_t)^{\mathrm{T}}\widetilde{\nabla} f_t(x_t)\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y - x_t) \\ & - \frac{9D}{8}\left\|\nabla f_t(z_t) - \widetilde{\nabla} f_t(z_t)\right\|_1 \\ & - \frac{D}{16G}\left\|\nabla f_t(z_t) - \widetilde{\nabla} f_t(z_t)\right\|_1^2 \\ & - 3G\sqrt{n}r_t' - \frac{nr_t'^2 G}{2}. \end{aligned} \tag{7}$$

The proof is given in Appendix B.5. The remaining part involves parameters selections, which is similar to the proof of Theorem 1.

Theorem 3. *The non-smooth version of Algorithm 2 with parameters $\eta = \min\left\{\frac{1}{8GD}, \frac{\alpha}{2}\right\}$, $\epsilon = \frac{1}{\eta^2 D^2}$, $\left\{r_t = \frac{\rho p}{\pi n^{9/2}(n/\rho+1)t^2}\right\}_{t=1}^T$, $\left\{r_t' = \frac{1}{t\sqrt{n}}\right\}_{t=1}^T$ can achieve the regret bound $O\left(\left(DG + \frac{1}{\alpha}\right) n \log(T + \log T)\right)$ with probability greater than $1 - T(\rho + p)$, and its query complexity is $O(1)$ in each round.*

The proof is given in Appendix B.6.

6. Conclusion and Discussion

In this paper, we considered the multi-points bandit feedback setting for online exp-concave optimization against the completely adaptive adversary. We provided a quantum algorithm and proved that it can achieve $O(n \log T)$ regret where only $O(1)$ queries were needed in each round. These results showed that potential quantum advantages are possible for problems of online exp-concave optimization under such setting since the classical online Newton method with zero-order feedback only achieves $O(nT^{2/3} \log T)$ regret. Furthermore, we presented how to extend our work to quantum adaptive gradient method and quantum online Newton method with Hessian update. For the Hessian update case, we also showed how to extend the quantum gradient estimation algorithm to the quantum Hessian estimation algorithm for the case that the quantum first-order oracle is accessible. Table 2 summarizes the update rules of these algorithms.

We also proposed a classical framework of online Newton method with Hessian update, and showed that it can achieve logarithm regret with appropriate parameters. There are still much room for research between the Hessian update (also known as full-rank update) and the rank-1 update as shown in Algorithm 1 and 2, namely, online Newton methods with rank-k update for $1 < k < n$ have not been explored yet. In the offline optimization setting, the convergence rate is better with higher rank update (Liu et al., 2023a;b), but for the online optimization setting, the problem whether high rank update can improve the regret bound still remains open.

Furthermore, online algorithms with the quantum first-order oracle have not been fully explored. With the quantum first-order oracle, given the superposition input, one can receive a superposition state of corresponding gradients at different points, but due to the properties of quantum computing, we need to explore algorithmic design to utilize some kind of overall information if we want to keep the advantage of quantum computing. Thus, on one hand, one needs to answer whether it can guarantee a better regret if we can make use of some kind of overall information of gradients of the loss function at different points. On the other hand, one also needs to consider how to design a quantum algorithm to estimate the Hessian matrix. But if we design the quantum Hessian estimation algorithm with the first-order oracle using the same framework of the quantum gradient estimation algorithm with the zeroth-order oracle, it can only output one row of the Hessian matrix per execution, which will lead to $O(n)$ queries for estimating the total Hessian matrix. Thus, it's also interesting to consider whether there exists a quantum algorithm that can estimate the Hessian with $O(1)$-query to the quantum first-order oracle.

Acknowledgements

The work of John C.S. Lui was supported in part by the RGC SRFS2122-4S02. The work of Lvzhou Li was supported in part by the National Science Foundation of China under Grant 62272492 and the Guangdong Provincial Quantum Science Strategic Initiative under Grant GDZX2303007.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Table 2. Summary of the update rules.

Algorithm	Update rules
Quantum online quasi-Newton method (Section 3)	$x_{t+1} = \hat{P}_{\mathcal{K}}^{(A_t)}\left(x_t - \frac{1}{\eta}A_t^{-1}\widetilde{\nabla} f_t(x_t)\right)$, where $A_t = A_{t-1} + \widetilde{\nabla} f_t(x_t)(\widetilde{\nabla} f_t(x_t))^{\mathrm{T}}$
Quantum adaptive gradient method (Subsection 5.1)	$x_{t+1} = \hat{P}_{\mathcal{K}}^{(G_t)}\left(x_t - \frac{1}{\gamma}G_t^{-1}\widetilde{\nabla} f_t(x_t)\right)$, where $G_t = \mathrm{diag}\,(A_t)^{1-p}$ and $A_t = A_{t-1} + \widetilde{\nabla} f_t(x_t)(\widetilde{\nabla} f_t(x_t))^{\mathrm{T}}$
Quantum online Newton method (Subsection 5.2)	$x_{t+1} = \hat{P}_{\mathcal{K}}^{(A_t)}\left(x_t - \frac{1}{\eta}A_t^{-1}\nabla f_t(x_t)\right)$, where $A_t = A_{t-1} + \widetilde{H}_t$

References

Abernethy, J. D., Hazan, E., and Rakhlin, A. Interior-point methods for full-information and bandit online learning. *IEEE Transactions on Information Theory*, 58(7):4164–4175, 2012.

Agarwal, A., Dekel, O., and Xiao, L. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *Conference on Learning Theory*, pp. 28–40, New York, NY, 2010. ACM.

Ambainis, A. and Špalek, R. Quantum algorithms for matching and network flows. In *Annual Symposium on Theoretical Aspects of Computer Science*, pp. 172–183, Berlin, 2006. Springer.

Awerbuch, B. and Kleinberg, R. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008.

Bartlett, P. L., Koolen, W. M., Malek, A., Takimoto, E., and Warmuth, M. K. Minimax fixed-design linear regression. In *Conference on Learning Theory*, pp. 226–239. PMLR, 2015.

Blekos, K., Brand, D., Ceschini, A., Chou, C.-H., Li, R.-H., Pandya, K., and Summer, A. A review on quantum approximate optimization algorithm and its variants. *arXiv preprint arXiv:2306.09198*, 2023.

Brandao, F. G. and Svore, K. M. Quantum speed-ups for solving semidefinite programs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 415–426, Piscataway, NJ, 2017. IEEE.

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

Bubeck, S. and Eldan, R. Multi-scale exploration of convex functions and bandit convex optimization. In *Conference on Learning Theory*, pp. 583–589, New York, NY, 2016. ACM.

Bubeck, S., Lee, Y. T., and Eldan, R. Kernel-based methods for bandit convex optimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 72–85, New York, NY, 2017. ACM.

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

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

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Dürr, C. and Høyer, P. A quantum algorithm for finding the minimum. *arXiv preprint quant-ph/9607014*, 1996.

Dürr, C., Heiligman, M., Høyer, P., and Mhalla, M. Quantum query complexity of some graph problems. *SIAM Journal on Computing*, 35(6):1310–1328, 2006.

Flaxman, A. D., Kalai, A. T., Kalai, A. T., and McMahan, H. B. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 385–394, Philadelphia, PA, 2005. SIAM.

Gaillard, P., Gerchinovitz, S., Huard, M., and Stoltz, G. Uniform regret bounds over r^d for the sequential linear regression problem with the square loss. In *Algorithmic Learning Theory*, pp. 404–432. PMLR, 2019.

Gilyén, A., Arunachalam, S., and Wiebe, N. Optimizing quantum optimization algorithms via faster quantum gradient computation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1425–1444, Philadelphia, PA, 2019. SIAM.

Grover, L. K. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212–219, New York, NY, 1996. ACM.

Harrow, A. W. and Montanaro, A. Quantum computational supremacy. *Nature*, 549(7671):203, 2017.

Harrow, A. W., Hassidim, A., and Lloyd, S. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

Hazan, E. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Hazan, E. and Kale, S. Projection-free online learning. In *29th International Conference on Machine Learning, ICML 2012*, pp. 521–528, 2012.

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

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

He, X., Zhang, J., and Sun, X. Quantum search with prior knowledge. *arXiv preprint arXiv:2009.08721*, 2020.

Jézéquel, R., Ostrovskii, D. M., and Gaillard, P. Efficient and near-optimal online portfolio selection. *arXiv preprint arXiv:2209.13932*, 2022.

Jordan, S. P. Fast quantum algorithm for numerical gradient estimation. *Physical review letters*, 95(5):050501, 2005.

Kalev, A., Li, T., Lin, C. Y.-Y., Svore, K. M., and Wu, X. Quantum sdp solvers: Large speed-ups, optimality, and applications to quantum learning. *Leibniz international proceedings in informatics*, 2019.

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

Kerenidis, I. and Prakash, A. Quantum gradient descent for linear systems and least squares. *Physical Review A*, 101 (2):022316, 2020b.

Kerenidis, I., Prakash, A., and Szilágyi, D. Quantum algorithms for portfolio optimization. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pp. 147–155, 2019a.

Kerenidis, I., Prakash, A., and Szilágyi, D. Quantum algorithms for second-order cone programming and support vector machines. *arXiv preprint arXiv:1908.06720*, 2019b.

Kerenidis, I., Prakash, A., and Szilágyi, D. A quantum interior-point method for second-order cone programming. *[Research Report] IRIF. 2019. ffhal-02138307*, 2019c.

Lattimore, T. Improved regret for zeroth-order adversarial bandit convex optimisation. *Mathematical Statistics and Learning*, 2(3):311–334, 2020.

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

Li, T. and Zhang, R. Quantum speedups of optimizing approximately convex functions with applications to logarithmic regret stochastic convex bandits. *Advances in Neural Information Processing Systems*, 35:3152–3164, 2022.

Li, T., Chakrabarti, S., and Wu, X. Sublinear quantum algorithms for training linear and kernel-based classifiers. In *International Conference on Machine Learning*, pp. 3815–3824, Palo Alto, CA, 2019. AAAI.

Lim, D. and Rebentrost, P. A quantum online portfolio optimization algorithm. *arXiv preprint arXiv:2208.14749*, 2022.

Liu, B., Li, J., Song, Y., Liang, X., Jian, L., and Liu, H. Online Newton step algorithm with estimated gradient, 2018. URL https://arxiv.org/pdf/1811.09955v3.pdf.

Liu, C., Chen, C., and Luo, L. Symmetric rank-k methods. *arXiv preprint arXiv:2303.16188*, 2023a.

Liu, C., Chen, C., Luo, L., and Lui, J. C. Block Broyden's methods for solving nonlinear equations. In *Advances in Neural Information Processing Systems*, 2023b.

Luo, H., Wei, C.-Y., and Zheng, K. Efficient online portfolio with logarithmic regret. *Advances in neural information processing systems*, 31, 2018.

Mhammedi, Z. and Gatmiry, K. Quasi-Newton steps for efficient online exp-concave optimization. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 4473–4503. PMLR, 2023.

Mhammedi, Z. and Rakhlin, A. Damped online Newton step for portfolio selection. In *Conference on Learning Theory*, pp. 5561–5595. PMLR, 2022.

Mizel, A. Critically damped quantum search. *Physical review letters*, 102(15):150501, 2009.

Nielsen, M. A. and Chuang, I. *Quantum computation and quantum information*. Cambridge University Press, Cambridge, 2002.

Rakhlin, A., Sridharan, K., and Tewari, A. Online learning via sequential complexities. *J. Mach. Learn. Res.*, 16(1): 155–186, 2015.

Rebentrost, P., Schuld, M., Wossnig, L., Petruccione, F., and Lloyd, S. Quantum gradient descent and Newton's method for constrained polynomial optimization. *New Journal of Physics*, 21(7):073023, 2019.

Sadowski, P. Quantum search with prior knowledge. *arXiv preprint arXiv:1506.04030*, 2015.

Shamir, O. On the complexity of bandit and derivative-free stochastic convex optimization. In *Conference on Learning Theory*, pp. 3–24, New York, NY, 2013. ACM.

Shamir, O. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *The Journal of Machine Learning Research*, 18(1):1703–1713, 2017.

Sherman, J. and Morrison, W. J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21 (1):124–127, 1950.

Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

van Apeldoorn, J. and Gilyén, A. Improvements in quantum sdp-solving with applications. In *46th International Colloquium on Automata, Languages, and Programming*, pp. 99, Wadern, 2019a. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

van Apeldoorn, J. and Gilyén, A. Quantum algorithms for zero-sum games. *arXiv preprint arXiv:1904.03180*, 2019b.

Van Apeldoorn, J., Gilyén, A., Gribling, S., and de Wolf, R. Quantum sdp-solvers: Better upper and lower bounds. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 403–414, Piscataway, NJ, 2017. IEEE.

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

Wan, Z., Zhang, Z., Li, T., Zhang, J., and Sun, X. Quantum multi-armed bandits and stochastic linear bandits enjoy logarithmic regrets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10087–10094, 2023.

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

Woodbury, M. A. *Inverting modified matrices*. Department of Statistics, Princeton University, 1950.

Yoder, T. J., Low, G. H., and Chuang, I. L. Fixed-point quantum search with an optimal number of queries. *Physical review letters*, 113(21):210501, 2014.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 928–936, Palo Alto, CA, 2003. AAAI.

A. Extended Related Works

A.1. Extended Related Works of Online Convex Optimizaiton

In general, less feedback information received by the player in each round will lead to a worse regret. In early works (Zinkevich, 2003; Hazan et al., 2007), it was assumed that the player can get full information of loss functions as feedback, or has access to gradient oracles of loss functions. In 2003, Zinkevich (2003) defined the online convex optimization model and showed that with gradient oracles feedback, the online gradient descent could achieve $O(\sqrt{T})$ regret. Additionally, it has been shown that the lower bound of this setting is $\Omega(\sqrt{T})$ (Theorem 3.2 of (Hazan, 2016)).

Contrarily, online convex optimization in a single-query bandit setting was proposed by Flaxman et al. (2005), where the only feedback was the value of the loss. Note that in the bandit setting, a regret bound for any strategy against the completely adaptive adversary is $\Omega(T)$. Thus, it needs to assume that the adversary is either adaptive or oblivious, i.e., the adversary must choose the loss function before observing the player's choice or before the start of the game, respectively. In any case, the expected regret of Flaxman's algorithm is $O(\sqrt{n}T^{3/4})$. The dependence on T was improved to $O(\text{poly}(n)\sqrt{T}\log T)$ by Bubeck and Eldan (2016; 2017) and Lattimore (2020) with the price of increasing the dimension-dependence.

Better regret can be achieved if the player can query the value of the loss function at more than one point in each round. In 2010, Agarwal et al. (2010) considered the multi-query bandit setting, and proposed an algorithm with an expected regret bound of $O(n^2\sqrt{T})$, where the player queries $O(1)$ points in each round. In 2017, the upper bound was improved to $O(\sqrt{nT/k})$ by Shamir (2017), where the player queries k points in each round. Although much effort has been made to minimize the impact of the dimension, but there is still a polynomial dependence on it, until the introducing of quantum techniques. In 2022, He et al. (2022) gave a quantum subgradient descent method that achieves $O(\sqrt{T})$ regret without dependence of the dimension n.

For works of the general convex loss functions setting mentioned above, it seems difficult to achieve regret logarithmic in T even with the help of quantum computers. But for specific loss functions, achieving logarithmic regret is possible. For strongly convex loss functions setting, $O(\log T)$ regret could be achieved by using online gradient descent with full information feedback (Hazan et al., 2007) and by using quantum online subgradient descent with zeroth-order feedback (He et al., 2022). For exp-concave loss functions setting, Hazan et al. (2007) showed that $O(n\log T)$ regret could be achieved by using online Newton step method with full information feedback. The online Newton step method requires a generalized projection. For the full information feedback or first-order feedback setting, such projection can be avoided by using barrier regularizer for some cases (Abernethy et al., 2012; Mhammedi & Rakhlin, 2022; Mhammedi & Gatmiry, 2023). But the setting of exp-concave loss functions with zeroth-order feedback is still an open problem. To the best of our best knowledge, the classical online Newton method with zeroth-order feedback only achieves $O(nT^{2/3}\log T)$ regret(Liu et al., 2018).

Table 1 provides a comprehensive summary for various works in online convex optimization.

A.2. Extended Related Works of Quantum Optimization

Optimization problem is one of the most important problems in science and engineering. Since quantum computing has shown its advantages over classical computing, e.g., the famous Shor algorithm (Shor, 1999), Grover algorithm (Grover, 1996) and HHL algorithm (Harrow et al., 2009; Harrow & Montanaro, 2017), people have been exploring how to accelerate the optimization process using quantum computing techniques. In recent years, some significant advantages have been made on quantum algorithms for optimization problems. We classify these quantum optimization works in four classes: whether the problems are continuous or discrete, and whether the problems are online or offline.

Offline discrete optimization. This kind of optimization problems are also called combinatorial optimization, which was shown to be acceleratable by using quantum techniques such as Grover's algorithm or quantum walks (Grover, 1996; Ambainis & Špalek, 2006; Dürr et al., 2006; Dürr & Høyer, 1996; Mizel, 2009; Yoder et al., 2014; Sadowski, 2015; He et al., 2020). In the NISQ area, series of variational quantum algorithms were developed to solve combinatorial optimization, readers are referred to (Blekos et al., 2023) for more information.

Offline continuous optimization. When the optimization problems are continuous, usually one can device efficient algorithms because continuous problems tend to have more properties to exploit such as the gradient information. In recent years, some quantum improvements were achieved for offline continuous optimization in linear programming (Kerenidis

Table 3. Regret bound for online convex optimization with different settings.

	Paper	Feedback	Adversary	Regret
		general convex loss functions		
1	(Zinkevich, 2003)	Full information (first-order oracles)	Completely	$O\left(DG\sqrt{T}\right)$
	(Hazan, 2016)	Full information (first-order oracles)	Completely	$\Omega\left(DG\sqrt{T}\right)$
2	(Flaxman et al., 2005)	Single query (zeroth-order oracles)	Oblivious	$O\left(\sqrt{DGCn}T^{3/4}\right)$
	(Bubeck & Eldan, 2016)	Single query (zeroth-order oracles)	Oblivious	$O\left(DGn^{11}\sqrt{T}\log T\right)$
	(Bubeck et al., 2017)	Single query (zeroth-order oracles)	Oblivious	$O\left(DGn^{9.5}\sqrt{T}\log T\right)$
	(Lattimore, 2020)	Single query (zeroth-order oracles)	Oblivious	$O\left(DGn^{2.5}\sqrt{T}\log T\right)$
3	(Agarwal et al., 2010)	$O(1)$ queries (zeroth-order oracles)	Adaptive	$O\left((D^2+n^2G^2)\sqrt{T}\right)$
	(Shamir, 2017)	k queries (zeroth-order oracles)	Adaptive	$O\left(DG\sqrt{nT/k}\right)$
	(He et al., 2022)	$O(n)$ queries (zeroth-order oracles)	Completely	$O\left(DG\sqrt{T}\right)$
	(He et al., 2022)	$O(1)$ queries (quantum zeroth-order oracles)	Completely	$O\left(DG\sqrt{T}\right)$
		α-strongly convex function		
4	(Hazan et al., 2007)	Full information (first-order oracles)	Completely	$O\left(G^2\log T\right)$
	(Shamir, 2013)	Single query (zeroth-order oracles)	Completely	$\Omega\left(n\sqrt{T}\right)$
	(He et al., 2022)	$O(1)$ queries (quantum zeroth-order oracles)	Completely	$O\left(G^2\log T\right)$
		α-exp-concave functions		
5	(Hazan et al., 2007)	Full information (first-order oracles)	Completely	$O\left(DGn\log T\right)$
	(Liu et al., 2018)	Single queries (zeroth-order oracles)	Oblivious	$O\left(DGnT^{2/3}\log T\right)$
	This work	$O(1)$ queries (quantum zeroth-order oracles)	Completely	$O\left(DGn\log(T+\log T)\right)$

- Regret bound for online convex optimization with different settings. For a strategy, the less feedback information that the player uses, the better; the stronger adversary the player faces, the better. The full information feedback model reveals the most information about the function, while the single-query feedback model reveals the least. The completely adaptive adversary is strongest, thus the corresponding models are the least restrictive ones in using, while the oblivious adversary is weakest, thus the corresponding models are the most stringent ones in using.
- D is the diameter upper bound of the feasible set, G Lipschitz constant, C is the diameter upper bound of the range of loss functions, n is the dimension of the feasible set, T is the horizon.

& Prakash, 2020a; Li et al., 2019; van Apeldoorn & Gilyén, 2019b), second-order cone programming (Kerenidis et al., 2019c;b;a), quadratic programming (Kerenidis & Prakash, 2020b), polynomial optimization (Rebentrost et al., 2019), semi-definite optimization (Kerenidis & Prakash, 2020a; van Apeldoorn & Gilyén, 2019a; Brandao & Svore, 2017; Kalev et al., 2019; Van Apeldoorn et al., 2017), and general convex optimization (van Apeldoorn et al., 2020; Chakrabarti et al., 2020). For the stochastic convex bandits problem, (Li & Zhang, 2022) gave a quantum algorithm that achieved exponential speedup in T with the price of increasing the dimension-dependence. Quantum techniques such as quantum Fourier transform, HHL-like algorithms and QRAM play an important role in offline continuous optimization.

Online discrete optimization. Recently, people are considering how to apply quantum computing methods to online optimization problems. The online discrete optimization problem is best known as the multi-arm bandit problem. In 2020, two quantum algorithms for the best arm identification problem, a central problem in multi-armed bandit, were given (Casalé et al., 2020; Wang et al., 2021). They showed that with the help of quantum coherent bandit oracle, quantum speedup can be achieved in the best arm identification problem. Later, for multi-armed bandits and stochastic linear bandits problems (the later problem is continuous), exponentially improving for the dependence in T was shown using a weaker quantum bandit oracle (Wan et al., 2023). Quantum amplitude amplification and quantum Monte Carlo method play a central role in the study of quantum multi-arm bandit.

Online continuous optimization. In online continuous optimization, it is usually assumed that there is an adversary who is responsible for choosing the loss functions in each round. Thus, it is often called the adversarial optimization problems. In 2022, (He et al., 2022) studied the quantum algorithm of online convex optimization, which achieves a regret bound that is independent of the dimension so that outperforms the optimal classical algorithms in the same setting which still have polynomial dependence of the dimension. For the online portfolio optimization problem, (Lim & Rebentrost, 2022) gave a quantum algorithm that provides a quadratic speedup in the time complexity by using techniques such as quantum state preparation, inner product estimation and multi-sampling.

B. Proof detail

B.1. Proof of Lemma 1

We first provide some basic definitions of quantum gates/circuits (Nielsen & Chuang, 2002), as follows:

- Hadamard gate: $H\left|0\right\rangle := \frac{\left|0\right\rangle+\left|1\right\rangle}{\sqrt{2}}$, $H\left|1\right\rangle := \frac{\left|0\right\rangle-\left|1\right\rangle}{\sqrt{2}}$,
- modulo add: $+\left|u\right\rangle\left|v\right\rangle := \left|u\right\rangle\left|(u+v) \mod 2^c\right\rangle$,
- Inverse quantum Fourier transform: $QFT^{-1}\left|v\right\rangle := \frac{1}{2^b}\sum_{v\in\{-2^{b-1},\ldots,2^{b-1}\}} e^{-\frac{2\pi i v w}{2^b}}\left|w\right\rangle, \forall w \in \{-2^{b-1},\ldots,2^{b-1}\}$.

The oracle Q_{F_t}, defined as $Q_{F_t}\left|u\right\rangle\left|0\right\rangle := \left|u\right\rangle\left|0+F_t(u)\right\rangle$, where $F_t(u) = \dfrac{2^b}{2Gr_t}\left[f_t\left(x_t+\dfrac{r_t}{2^b}\left(u-\dfrac{2^b}{2}\mathbb{1}\right)\right)-f_t(x_t)\right]$, can be construct by two Q_{f_t}, as follows:

$$
\begin{aligned}
&\left|u\right\rangle\left|x_t\right\rangle\left|0\right\rangle\left|0\right\rangle\left|0\right\rangle\left|0\right\rangle\left|0\right\rangle \\
\longrightarrow\ &\left|u\right\rangle\left|x_t\right\rangle\left|x_t+\frac{r_t}{2^b}\left(u-\frac{2^b}{2}\mathbb{1}\right)\right\rangle\left|0\right\rangle\left|x_t\right\rangle\left|0\right\rangle\left|0\right\rangle \\
\xrightarrow{\text{two } Q_{f_t}}\ &\left|u\right\rangle\left|x_t\right\rangle\left|x_t+\frac{r_t}{2^b}\left(u-\frac{2^b}{2}\mathbb{1}\right)\right\rangle\left|f_t\left(x_t+\frac{r_t}{2^b}\left(u-\frac{2^b}{2}\mathbb{1}\right)\right)\right\rangle\left|x_t\right\rangle\left|f_t(x_t)\right\rangle\left|0\right\rangle \\
\longrightarrow\ &\left|u\right\rangle\left|x_t\right\rangle\left|x_t+\frac{r_t}{2^b}\left(u-\frac{2^b}{2}\mathbb{1}\right)\right\rangle\left|f_t\left(x_t+\frac{r_t}{2^b}\left(u-\frac{2^b}{2}\mathbb{1}\right)\right)\right\rangle\left|x_t\right\rangle\left|f_t(x_t)\right\rangle\left|F_t(u)\right\rangle,
\end{aligned}
\tag{8}
$$

and then uncompute all of the auxiliary registers.

Now we are ready to prove Lemma 1.

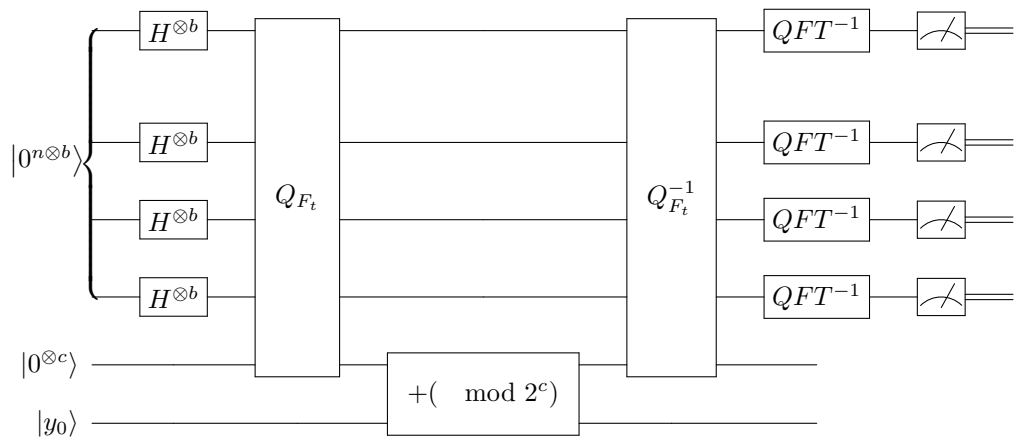

Figure 2. Quantum gradient estimation.

Lemma 1. *In Algorithm 2, for all timestep t, if f_t is β-smooth function, then for any $r_t > 0$ and $1 \geq \rho_t > 0$, the estimated gradient $\widetilde{\nabla} f_t(x_t)$ satisfies*

$$\Pr\Big[\|\nabla f_t(x_t) - \widetilde{\nabla} f_t(x_t)\|_1 > 8\pi n^3(n/\rho_t + 1)\beta r_t/\rho_t\Big] < \rho_t. \tag{3}$$

Proof. We omit the subscript t in the proof as the lemma holds for each timestep t. The states after Step 4 will be:

$$\frac{1}{\sqrt{2^n}} \sum_{a\in\{0,1,\ldots,2^n-1\}} e^{\frac{2\pi i a}{2^n}} \left|0^{\otimes b}, 0^{\otimes b}, \ldots, 0^{\otimes b}\right\rangle \left|0^{\otimes c}\right\rangle |a\rangle . \tag{9}$$

After Step 5:

$$\frac{1}{\sqrt{2^{bn+c}}} \sum_{u_1,u_2,\ldots,u_n\in\{0,1,\ldots,2^b-1\}} \sum_{a\in\{0,1,\ldots,2^c-1\}} e^{\frac{2\pi i a}{2^n}} |u_1, u_2, \ldots, u_n\rangle \left|0^{\otimes c}\right\rangle |a\rangle . \tag{10}$$

After Step 6:

$$\frac{1}{\sqrt{2^{bn+c}}} \sum_{u_1,u_2,\ldots,u_n\in\{0,1,\ldots,2^b-1\}} \sum_{a\in\{0,1,\ldots,2^c-1\}} e^{\frac{2\pi i a}{2^n}} |u_1, u_2, \ldots, u_n\rangle |F(u)\rangle |a\rangle . \tag{11}$$

After Step 7:

$$\frac{1}{\sqrt{2^{bn+c}}} \sum_{u_1,u_2,\ldots,u_n\in\{0,1,\ldots,2^b-1\}} \sum_{a\in\{0,1,\ldots,2^c-1\}} e^{2\pi i F(u)} e^{\frac{2\pi i a}{2^n}} |u_1, u_2, \ldots, u_n\rangle |F(u)\rangle |a\rangle . \tag{12}$$

After Step 8:

$$\frac{1}{\sqrt{2^{bn+c}}} \sum_{u_1,u_2,\ldots,u_n\in\{0,1,\ldots,2^b-1\}} \sum_{a\in\{0,1,\ldots,2^c-1\}} e^{2\pi i F(u)} e^{\frac{2\pi i a}{2^n}} |u_1, u_2, \ldots, u_n\rangle \left|0^{\otimes c}\right\rangle |a\rangle . \tag{13}$$

In the following, the last two registers will be omitted:

$$\frac{1}{\sqrt{2^{bn}}} \sum_{u_1,u_2,\ldots,u_n\in\{0,1,\ldots,2^b-1\}} e^{2\pi i F(u)} |u_1, u_2, \ldots, u_n\rangle . \tag{14}$$

And then we simply relabel the state by changing $u \to v = u - \frac{2^b}{2}$:

$$\frac{1}{\sqrt{2^{bn}}} \sum_{v_1,v_2,\ldots,v_n \in \{-2^{b-1},-2^{b-1}+1,\ldots,2^{b-1}\}} e^{2\pi i F(v)} |v\rangle . \tag{15}$$

We denote Formula (15) as $|\phi\rangle$. Let $g = \nabla f(x)$, and consider the idealized state

$$|\psi\rangle = \frac{1}{\sqrt{2^{bn}}} \sum_{v_1,v_2,\ldots,v_n \in \{-2^{b-1},-2^{b-1}+1,\ldots,2^{b-1}\}} e^{\frac{2\pi i g\cdot v}{2G}} |v\rangle . \tag{16}$$

After Step 10, from the analysis of phase estimation (Brassard et al., 2002):

$$\Pr\left[\left|\frac{Ng_i}{2G} - m_i\right| > e\right] < \frac{1}{2(e-1)}, \forall i \in [n]. \tag{17}$$

Let $e = n/\rho + 1$, where $1 \geq \rho > 0$. We have

$$\Pr\left[\left|\frac{Ng_i}{2G} - m_i\right| > n/\rho + 1\right] < \frac{\rho}{2n}, \forall i \in [n]. \tag{18}$$

Note that the difference in the probabilities of measurement on $|\phi\rangle$ and $|\psi\rangle$ can be bounded by the trace distance between the two density matrices:

$$\| \, |\phi\rangle\langle\phi| - |\psi\rangle\langle\psi| \, \|_1 = 2\sqrt{1 - |\langle\phi|\psi\rangle|^2} \leq 2\| \, |\phi\rangle - |\psi\rangle \, \|. \tag{19}$$

Since f is β-smooth, we have

$$\begin{aligned} F(v) &\leq \frac{2^b}{2Gr}[f(x + \frac{rv}{N}) - f(x)] + \frac{1}{2^{c+1}} \\ &\leq \frac{2^b}{2Gr}[\frac{r}{2^b} g \cdot v + \frac{\beta (rv)^2}{2^{2b}}] + \frac{1}{2^{c+1}} \\ &\leq \frac{g \cdot v}{2G} + \frac{2^b \beta r n}{4G} + \frac{1}{2^{c+1}}. \end{aligned} \tag{20}$$

Then,

$$\begin{aligned} \| \, |\phi\rangle - |\psi\rangle \, \|^2 &= \frac{1}{2^{bn}} \sum_v |e^{2\pi i F(v)} - e^{\frac{2\pi i g\cdot v}{2G}}|^2 \\ &\leq \frac{1}{2^{bn}} \sum_v |2\pi i F(v) - \frac{2\pi i g \cdot v}{2G}|^2 \\ &\leq \frac{1}{2^{bn}} \sum_v 4\pi^2 (\frac{2^b \beta r n}{4G} + \frac{1}{2^{c+1}})^2. \end{aligned} \tag{21}$$

Set $b = \log_2 \frac{G\rho}{4\pi n^2 \beta r}$, $c = \log_2 \frac{4G}{2^b n \beta r} - 1$. We have

$$\| \, |\phi\rangle - |\psi\rangle \, \|^2 \leq \frac{\rho^2}{16n^2}, \tag{22}$$

which implies $\| \, |\phi\rangle\langle\phi| - |\psi\rangle\langle\psi| \, \|_1 \leq \frac{\rho}{2n}$. Therefore, by the union bound,

$$\Pr\left[\left|\frac{2^b g_i}{2G} - m_i\right| > n/\rho + 1\right] < \frac{\rho}{n}, \forall i \in [n]. \tag{23}$$

Furthermore, there is

$$\Pr\left[\left|g_i - \widetilde{\nabla}_i f(x)\right| > \frac{2G(n/\rho + 1)}{2^b}\right] < \frac{\rho}{n}, \forall i \in [n], \tag{24}$$

as $b = \log_2 \frac{G\rho}{4\pi n^2 \beta r}$, we have

$$\Pr\Big[\Big|g_i - \widetilde{\nabla}_i f(x)\Big| > 8\pi n^2 (n/\rho + 1)\beta r/\rho\Big] < \frac{\rho}{n}, \forall i \in [n]. \tag{25}$$

By the union bound, we have

$$\Pr\Big[\|g - \widetilde{\nabla} f(x)\|_1 > 8\pi n^3 (n/\rho + 1)\beta r/\rho\Big] < \rho, \tag{26}$$

which gives the lemma. □

B.2. Proof of Lemma 2

Lemma 2. *In Algorithm 2, for all timestep t, let $f_t : \mathcal{K} \to \mathbb{R}$ be α-exp-concave functions with Lipschitz parameter G, where $\mathcal{K}$ is a convex set with diameter D, then for any $y \in \mathcal{K}$ and any $\eta_t \leq \min\left\{\frac{1}{8GD}, \frac{\alpha}{2}\right\}$, the estimated gradient $\widetilde{\nabla} f_t(x_t)$ satisfies*

$$\begin{aligned} f_t(y) \geq & f_t(x_t) + \widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y - x_t) \\ & + \frac{\eta_t}{2}(y - x_t)^{\mathrm{T}} \widetilde{\nabla} f_t(x_t) \widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y - x_t) \\ & - \frac{9D}{8}\Big\|\nabla f_t(x_t) - \widetilde{\nabla} f_t(x_t)\Big\|_1 \\ & - \frac{D}{16G}\Big\|\nabla f_t(x_t) - \widetilde{\nabla} f_t(x_t)\Big\|_1^2. \end{aligned} \tag{4}$$

Proof. Note that we omit the subscript t in the proofs of the lemmas as they hold for each timestep t. By the definition of α-exp-concave, $e^{-\alpha f(x)}$ is a concave function. For $\eta \leq \frac{\alpha}{2}$, $e^{-2\eta f(x)}$ is a concave function as well. By the definition of concave functions, for any $y \in \mathcal{K}$, we have

$$\begin{aligned} e^{-2\eta f(y)} & \leq e^{-2\eta f(x)} + \nabla\left(e^{-2\eta f(x)}\right)^{\mathrm{T}}(y - x) \\ & = e^{-2\eta f(x)} + -2\eta e^{-2\eta f(x)} \nabla f(x)^{\mathrm{T}}(y - x), \end{aligned} \tag{27}$$

Simplifying gives

$$\begin{aligned} f(y) & \geq f(x) - \frac{1}{2\eta}\log\left(1 - 2\eta \nabla f(x)^{\mathrm{T}}(y - x)\right) \\ & \geq f(x) + \nabla f(x)^{\mathrm{T}}(y - x) + \frac{\eta}{2}(y - x)^{\mathrm{T}} \nabla f(x) \nabla f(x)^{\mathrm{T}}(y - x), \end{aligned} \tag{28}$$

where the last inequality comes from $-\log(1 - z) \geq z + \frac{z^2}{4}$ for any $|z| \leq 1$. Further, for the second term on the right hand side of Inequality (28),

$$\begin{aligned} \nabla f(x)^{\mathrm{T}}(y - x) & = \nabla f(x)^{\mathrm{T}}(y - x) + \widetilde{\nabla} f(x)^{\mathrm{T}}(y - x) - \widetilde{\nabla} f(x)^{\mathrm{T}}(y - x) \\ & = \widetilde{\nabla} f(x)^{\mathrm{T}}(y - x) + \left(\nabla f(x) - \widetilde{\nabla} f(x)\right)^{\mathrm{T}}(y - x) \\ & \geq \widetilde{\nabla} f(x)^{\mathrm{T}}(y - x) - \Big\|\nabla f(x) - \widetilde{\nabla} f(x)\Big\|_1 \|y - x\|_\infty \\ & \geq \widetilde{\nabla} f(x)^{\mathrm{T}}(y - x) - \Big\|\nabla f(x) - \widetilde{\nabla} f(x)\Big\|_1 \|y - x\|_2 \end{aligned} \tag{29}$$

Since $\|y - x\|_2 \leq D$, we have

$$\nabla f(x)^{\mathrm{T}}(y - x) \geq \widetilde{\nabla} f(x)^{\mathrm{T}}(y - x) - D\Big\|\nabla f(x) - \widetilde{\nabla} f(x)\Big\|_1. \tag{30}$$

For the third term on the right hand side of Inequality (28),

$$\frac{\eta}{2}(y - x)^{\mathrm{T}} \nabla f(x) \nabla f(x)^{\mathrm{T}}(y - x)$$

$$
\begin{aligned}
=&\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(x)\nabla f(x)^{\mathrm{T}}(y-x)+\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)\\
&-\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)\\
=&\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)+\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(x)\nabla f(x)^{\mathrm{T}}(y-x)\\
&-\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)+\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)\\
&-\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)\\
=&\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)+\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(x)\left(\nabla f(x)-\widetilde{\nabla} f(x)\right)^{\mathrm{T}}(y-x)\\
&+\frac{\eta}{2}(y-x)^{\mathrm{T}}\left(\nabla f(x)-\widetilde{\nabla} f(x)\right)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)\\
\geq&\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)-\frac{\eta}{2}\|y-x\|_2\|\nabla f(x)\|_2\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1\|y-x\|_\infty\\
&-\frac{\eta}{2}\|y-x\|_\infty\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1\left\|\widetilde{\nabla} f(x)\right\|_2\|y-x\|_2\\
=&\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)-\frac{\eta}{2}\|y-x\|_2\|\nabla f(x)\|_2\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1\|y-x\|_\infty\\
&-\frac{\eta}{2}\|y-x\|_\infty\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1\left\|\widetilde{\nabla} f(x)-\nabla f(x)+\nabla f(x)\right\|_2\|y-x\|_2\\
\geq&\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)-\frac{\eta}{2}\|y-x\|_2\|\nabla f(x)\|_2\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1\|y-x\|_\infty\\
&-\frac{\eta}{2}\|y-x\|_\infty\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1\left(\left\|\widetilde{\nabla} f(x)-\nabla f(x)\right\|_2+\|\nabla f(x)\|_2\right)\|y-x\|_2\\
\geq&\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)-\eta\|y-x\|_2\|\nabla f(x)\|_2\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1\|y-x\|_\infty\\
&-\frac{\eta}{2}\|y-x\|_\infty\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1^2\|y-x\|_2\\
\geq&\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)-\eta\|y-x\|_2\|\nabla f(x)\|_2\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1\|y-x\|_2\\
&-\frac{\eta}{2}\|y-x\|_2\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1^2\|y-x\|_2\\
=&\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)-\eta\|\nabla f(x)\|_2\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1\|y-x\|_2^2\\
&-\frac{\eta}{2}\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1^2\|y-x\|_2^2.
\end{aligned}
\tag{31}
$$

Since $\eta \leq \frac{1}{2DG}$, $\|y-x\|_2 \leq D$, $\|\nabla f(x)\|_2 \leq G$, we have

$$
\begin{aligned}
\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(x)\nabla f(x)^{\mathrm{T}}(y-x) \geq&\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)\\
&-\frac{D}{8}\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1-\frac{D}{16G}\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1^2.
\end{aligned}
\tag{32}
$$

In summary, we have

$$
\begin{aligned}
f_t(y) \geq& f_t(x_t)+\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y-x_t)+\frac{\eta_t}{2}(y-x_t)^{\mathrm{T}}\widetilde{\nabla} f_t(x_t)\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y-x_t)\\
&-\frac{9D}{8}\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1-\frac{D}{16G}\left\|\nabla f(x)-\widetilde{\nabla} f(x)\right\|_1^2.
\end{aligned}
\tag{33}
$$

which gives the lemma. □

B.3. Proof of Theorem 1

Theorem 1. *Algorithm 2 with parameters* $\eta = \min\left\{\frac{1}{8GD}, \frac{\alpha}{2}\right\}$, $\epsilon = \frac{1}{\eta^2 D^2}$, $\left\{r_t = \frac{\rho G}{\pi n^3 (n/\rho+1)\beta t}\right\}_{t=1}^{T}$, *can achieve the regret bound* $O\left(\left(DG+\frac{1}{\alpha}\right) n \log(T+\log T)\right)$ *with probability greater than* $1-T\rho$, *and its query complexity is* $O(1)$ *in each round.*

Proof. By Lemma 1 and Lemma 2, for any $y \in \mathcal{K}$ and any $\eta_t \leq \frac{1}{2}\min\left\{\frac{1}{GD}, \alpha\right\}$, in any time step t, with probability greater than $1-\rho_t$, the gradient $\widetilde{\nabla} f_t(x_t)$ estimated by Algorithm 2 satisfied

$$\begin{aligned} f_t(y) \geq & f_t(x_t) + \widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y-x_t) + \frac{\eta_t}{2}(y-x_t)^{\mathrm{T}}\widetilde{\nabla} f_t(x_t)\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y-x_t) \\ & - 9D\pi n^3(n/\rho_t+1)\beta r_t/\rho_t - \frac{4D(\pi n^3(n/\rho_t+1)\beta r_t/\rho_t)^2}{G}. \end{aligned} \tag{34}$$

Inequality (34) is required to hold for all T rounds. Let B_t be the event that Algorithm 2 fails to satisfy Inequality (34) in the t-th round. First, set the failure rate of each round to be the same, specifically, equal to ρ. Then by Lemma 2, we have $\Pr(B_1) = \Pr(B_2) = \cdots = \Pr(B_T) \leq \rho$. By the union bound (that is, for any finite or countable event set, the probability that at least one of the events happens is no greater than the sum of the probabilities of the events in the set), we have $\Pr\left(\cup_{t=1}^{T} B_t\right) \leq \sum_{t=1}^{T}\Pr(B_t) \leq T\rho$. Namely, the probability that Algorithm 2 fails to satisfy Inequality (34) at least one round is less than $T\rho$, which means the probability that Algorithm 2 succeeds for all T round is greater than $1-T\rho$. Let $x^* \in \arg\min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x)$. for the fixed $y = x^*$, for all $t \in [T]$ with probability $1-T\rho$, we have

$$\begin{aligned} f_t(x_t) - f_t(x^*) \leq & \widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(x_t - x^*) - \frac{\eta_t}{2}(x_t-x^*)^{\mathrm{T}}\widetilde{\nabla} f_t(x_t)\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(x_t-x^*) \\ & + 9D\pi n^3(n/\rho+1)\beta r_t/\rho + \frac{4D(\pi n^3(n/\rho+1)\beta r_t/\rho)^2}{G}. \end{aligned} \tag{35}$$

By the update rule for x_{t+1} and the Pythagorean theorem, there is

$$\begin{aligned} \|x_{t+1}-x^*\|_{A_t}^2 &= \left\| \hat{P}_{\mathcal{K}}^{(A_t)}\left(x_t - \frac{1}{\eta_t}A_t^{-1}\widetilde{\nabla} f_t(x_t)\right) - x^* \right\|_{A_t}^2 \\ &\leq \left\| x_t - \frac{1}{\eta_t}A_t^{-1}\widetilde{\nabla} f_t(x_t) - x^* \right\|_{A_t}^2 \\ &\leq (x_t-x^*)^{\mathrm{T}}A_t(x_t-x^*) - \frac{2}{\eta_t}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(x_t-x^*) + \frac{1}{\eta_t^2}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}A_t^{-1}\widetilde{\nabla} f_t(x_t). \end{aligned} \tag{36}$$

Since $\|x_{t+1}-x^*\|_{A_t}^2 = (x_{t+1}-x^*)^{\mathrm{T}}A_t(x_{t+1}-x^*)$, combine with Inequality (36), we have

$$\begin{aligned} \widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(x_t-x^*) \leq & \frac{1}{2\eta_t}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}A_t^{-1}\widetilde{\nabla} f_t(x_t) + \frac{\eta_t}{2}(x_t-x^*)^{\mathrm{T}}A_t(x_t-x^*) \\ & - \frac{\eta_t}{2}(x_{t+1}-x^*)^{\mathrm{T}}A_t(x_{t+1}-x^*). \end{aligned} \tag{37}$$

summing Inequality (37) from $t=1$ to T, we have

$$\begin{aligned} \sum_{t=1}^{T}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(x_t-x^*) \leq & \sum_{t=1}^{T}\frac{1}{2\eta_t}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}A_t^{-1}\widetilde{\nabla} f_t(x_t) + \sum_{t=1}^{T}\frac{\eta_t}{2}(x_t-x^*)^{\mathrm{T}}A_t(x_t-x^*) \\ & - \sum_{t=1}^{T}\frac{\eta_t}{2}(x_{t+1}-x^*)^{\mathrm{T}}A_t(x_{t+1}-x^*) \\ = & \sum_{t=1}^{T}\frac{1}{2\eta_t}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}A_t^{-1}\widetilde{\nabla} f_t(x_t) + \frac{\eta_1}{2}(x_1-x^*)^{\mathrm{T}}A_1(x_1-x^*) \\ & + \frac{1}{2}\sum_{t=2}^{T}(x_t-x^*)^{\mathrm{T}}(\eta_t A_t - \eta_{t-1}A_{t-1})(x_t-x^*) \\ & - \frac{\eta_T}{2}(x_{T+1}-x^*)^{\mathrm{T}}A_T(x_{T+1}-x^*). \end{aligned} \tag{38}$$

Set $\eta_1 = \eta_2 = \cdots = \eta_T = \eta$, we have

$$\sum_{t=1}^{T}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(x_t-x^*) \leq \sum_{t=1}^{T}\frac{1}{2\eta}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}A_t^{-1}\widetilde{\nabla} f_t(x_t) + \frac{\eta}{2}(x_1-x^*)^{\mathrm{T}}A_1(x_1-x^*)$$

$$
\begin{aligned}
&+ \frac{\eta}{2}\sum_{t=2}^{T}(x_t - x^*)^{\mathrm{T}}(A_t - A_{t-1})(x_t - x^*) \\
&- \frac{\eta}{2}(x_{T+1} - x^*)^{\mathrm{T}} A_T (x_{T+1} - x^*) \\
=& \sum_{t=1}^{T}\frac{1}{2\eta}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}} A_t^{-1}\widetilde{\nabla} f_t(x_t) + \frac{\eta}{2}(x_1 - x^*)^{\mathrm{T}} A_1 (x_1 - x^*) \\
&+ \frac{\eta}{2}\sum_{t=2}^{T}(x_t - x^*)^{\mathrm{T}}\widetilde{\nabla} f_t(x_t)(\widetilde{\nabla} f_t(x_t))^{\mathrm{T}}(x_t - x^*) \\
&- \frac{\eta}{2}(x_{T+1} - x^*)^{\mathrm{T}} A_T (x_{T+1} - x^*) \\
=& \sum_{t=1}^{T}\frac{1}{2\eta}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}} A_t^{-1}\widetilde{\nabla} f_t(x_t) \\
&+ \frac{\eta}{2}(x_1 - x^*)^{\mathrm{T}}\left(A_1 - \widetilde{\nabla} f_1(x_1)(\widetilde{\nabla} f_1(x_1))^T\right)(x_1 - x^*) \\
&+ \frac{\eta}{2}\sum_{t=1}^{T}(x_t - x^*)^{\mathrm{T}}\widetilde{\nabla} f_t(x_t)(\widetilde{\nabla} f_t(x_t))^{\mathrm{T}}(x_t - x^*) \\
&- \frac{\eta}{2}(x_{T+1} - x^*)^{\mathrm{T}} A_T (x_{T+1} - x^*) \\
=& \sum_{t=1}^{T}\frac{1}{2\eta}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}} A_t^{-1}\widetilde{\nabla} f_t(x_t) + \frac{\epsilon\eta}{2}(x_1 - x^*)^{\mathrm{T}} I (x_1 - x^*) \\
&+ \frac{\eta}{2}\sum_{t=1}^{T}(x_t - x^*)^{\mathrm{T}}\widetilde{\nabla} f_t(x_t)(\widetilde{\nabla} f_t(x_t))^{\mathrm{T}}(x_t - x^*) \\
&- \frac{\eta}{2}(x_{T+1} - x^*)^{\mathrm{T}} A_T (x_{T+1} - x^*) \\
\leq& \sum_{t=1}^{T}\frac{1}{2\eta}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}} A_t^{-1}\widetilde{\nabla} f_t(x_t) + \frac{\epsilon\eta}{2}\|x_1 - x^*\|_2\|x_1 - x^*\|_2 \\
&+ \frac{\eta}{2}\sum_{t=1}^{T}(x_t - x^*)^{\mathrm{T}}\widetilde{\nabla} f_t(x_t)(\widetilde{\nabla} f_t(x_t))^{\mathrm{T}}(x_t - x^*) \\
&- \frac{\eta}{2}(x_{T+1} - x^*)^{\mathrm{T}} A_T (x_{T+1} - x^*) \\
\leq& \sum_{t=1}^{T}\frac{1}{2\eta}\widetilde{\nabla} f_t(x_t)^{\mathrm{T}} A_t^{-1}\widetilde{\nabla} f_t(x_t) + \frac{\epsilon\eta D^2}{2} \\
&+ \frac{\eta}{2}\sum_{t=1}^{T}(x_t - x^*)^{\mathrm{T}}\widetilde{\nabla} f_t(x_t)(\widetilde{\nabla} f_t(x_t))^{\mathrm{T}}(x_t - x^*) \\
&- \frac{\eta}{2}(x_{T+1} - x^*)^{\mathrm{T}} A_T (x_{T+1} - x^*).
\end{aligned} \tag{39}
$$

For any $t \in [T]$, by the update rule, $A_t = \epsilon I + \sum_{s=1}^{t}\widetilde{\nabla} f_s(x_s)(\widetilde{\nabla} f_s(x_s))^{\mathrm{T}}$, since

$$
A_t^{\mathrm{T}} = \epsilon I^{\mathrm{T}} + \sum_{s=1}^{t}\left(\widetilde{\nabla} f_s(x_s)(\widetilde{\nabla} f_s(x_s))^{\mathrm{T}}\right)^{\mathrm{T}} = \epsilon I + \sum_{s=1}^{t}\widetilde{\nabla} f_s(x_s)(\widetilde{\nabla} f_s(x_s))^{\mathrm{T}} = A_t, \tag{40}
$$

Therefore, A_t are symmetric matrices, and for any $z \in \mathbb{R}^n$,

$$
z^{\mathrm{T}} A_t z = \epsilon z^{\mathrm{T}} z + \sum_{s=1}^{t} z^{\mathrm{T}}\widetilde{\nabla} f_s(x_s)(\widetilde{\nabla} f_s(x_s))^{\mathrm{T}} z
$$

$$= \epsilon z^{\mathrm{T}} z + \sum_{s=1}^{t} \left(\widetilde{\nabla} f_s(x_s))^{\mathrm{T}} z \right)^{\mathrm{T}} (\widetilde{\nabla} f_s(x_s))^{\mathrm{T}} z, \tag{41}$$

when $\epsilon \geq 0$, $z^{\mathrm{T}} A_t z$ is non-negative, namely, A_t are positive semi-definite matrices. Set $\eta \geq 0, \epsilon \geq 0$, the last term in the right hand side of Inequality (39) is negative, can be discarded. For the first term in the right hand side of Inequality (39), we have

$$\begin{aligned}
\sum_{t=1}^{T} \frac{1}{2\eta} \widetilde{\nabla} f_t(x_t)^{\mathrm{T}} A_t^{-1} \widetilde{\nabla} f_t(x_t) &= \frac{1}{2\eta} \sum_{t=1}^{T} \mathrm{Tr} \left(A_t^{-1} \widetilde{\nabla} f_t(x_t) \widetilde{\nabla} f_t(x_t)^{\mathrm{T}} \right) \\
&= \frac{1}{2\eta} \sum_{t=1}^{T} \mathrm{Tr} \left(A_t^{-1} (A_t - A_{t-1}) \right) \\
&= \frac{1}{2\eta} \sum_{t=1}^{T} \mathrm{Tr} \left(I - A_t^{-0.5} A_{t-1} A_t^{-0.5} \right) \\
&= \frac{1}{2\eta} \sum_{t=1}^{T} \sum_{s=1}^{n} \left(1 - \lambda_s \left(A_t^{-0.5} A_{t-1} A_t^{-0.5} \right) \right) \\
&\leq -\frac{1}{2\eta} \sum_{t=1}^{T} \sum_{s=1}^{n} \log \left(\lambda_s \left(A_t^{-0.5} A_{t-1} A_t^{-0.5} \right) \right) \\
&= -\frac{1}{2\eta} \sum_{t=1}^{T} \log \left(\Pi_{s=1}^{n} \lambda_s \left(A_t^{-0.5} A_{t-1} A_t^{-0.5} \right) \right) \\
&= -\frac{1}{2\eta} \sum_{t=1}^{T} \log \left(\det \left(A_t^{-0.5} A_{t-1} A_t^{-0.5} \right) \right) \\
&= -\frac{1}{2\eta} \sum_{t=1}^{T} \log \left(\det \left(A_t^{-0.5} \right) \det \left(A_{t-1} \right) \det \left(A_t^{-0.5} \right) \right) \\
&= -\frac{1}{2\eta} \sum_{t=1}^{T} \log \left(\det \left(A_t^{-1} \right) \det \left(A_{t-1} \right) \right) \\
&= -\frac{1}{2\eta} \log \left(\Pi_{t=1}^{T} \det \left(A_t^{-1} \right) \det \left(A_{t-1} \right) \right) \\
&= -\frac{1}{2\eta} \log \left(\det \left(A_T^{-1} \right) \det \left(A_0 \right) \right) \\
&= \frac{1}{2\eta} \log \left(\det \left(A_T \right) \det \left(A_0^{-1} \right) \right) \\
&= \frac{1}{2\eta} \log \left(\det \left(\left(\epsilon I + \sum_{t=1}^{T} \widetilde{\nabla} f_t(x_t) (\widetilde{\nabla} f_t(x_t))^{\mathrm{T}} \right) (\epsilon I)^{-1} \right) \right) \\
&\leq \frac{1}{2\eta} \log \left(\left(\epsilon + \sum_{t=1}^{T} \left\| \widetilde{\nabla} f_t(x_t) \right\|_2^2 \right) \epsilon^{-1} \right)^n \\
&= \frac{n}{2\eta} \log \left(\left(\epsilon + \sum_{t=1}^{T} \left\| \widetilde{\nabla} f_t(x_t) - \nabla f_t(x_t) + \nabla f_t(x_t) \right\|_2^2 \right) \epsilon^{-1} \right) \\
&\leq \frac{n}{2\eta} \log \left(\left(\epsilon + \sum_{t=1}^{T} \left(\left\| \widetilde{\nabla} f_t(x_t) - \nabla f_t(x_t) \right\|_2 + \| \nabla f_t(x_t) \|_2 \right)^2 \right) \epsilon^{-1} \right) \\
&\leq \frac{n}{2\eta} \log \left(\left(\epsilon + \sum_{t=1}^{T} \left(\left\| \widetilde{\nabla} f_t(x_t) - \nabla f_t(x_t) \right\|_1 + G \right)^2 \right) \epsilon^{-1} \right)
\end{aligned}$$

$$\leq \frac{n}{2\eta}\log\left(\left(\epsilon+\sum_{t=1}^{T}\left(8\pi n^3(n/\rho+1)\beta r_t/\rho+G\right)^2\right)\epsilon^{-1}\right), \tag{42}$$

where $\lambda_s(A)$ is the s-th eigenvalue of matrix A, $\det(A)$ is the determinant of matrix A. Summing Inequality (35) from $t=1$ to T, and combine Inequality (39) (42), we have

$$\begin{aligned}\sum_{t=1}^{T} f_t(x_t)-f_t(x^*) \leq & \frac{n}{2\eta}\log\left(\left(\epsilon+\sum_{t=1}^{T}\left(8\pi n^3(n/\rho+1)\beta r_t/\rho+G\right)^2\right)\epsilon^{-1}\right)+\frac{\epsilon\eta D^2}{2} \\ & +\sum_{t=1}^{T} 9D\pi n^3(n/\rho+1)\beta r_t/\rho+\sum_{t=1}^{T}\frac{4D(\pi n^3(n/\rho+1)\beta r_t/\rho)^2}{G}.\end{aligned} \tag{43}$$

Set $\eta=\min\left\{\frac{1}{2GD},\frac{\alpha}{2}\right\}, \epsilon=\frac{1}{\eta^2D^2}, \left\{r_t=\frac{\rho G}{\pi n^3(n/\rho+1)\beta t}\right\}_{t=1}^{T}$, we have

$$\begin{aligned}\sum_{t=1}^{T} f_t(x_t)-f_t(x^*) &\leq \frac{n}{2\eta}\log\left(1+\sum_{t=1}^{T}\left(\frac{8G}{t}+G\right)^2\eta^2D^2\right)+\frac{1}{2\eta}+\sum_{t=1}^{T}\frac{9DG}{t}+\sum_{t=1}^{T}\frac{4DG^2}{t^2G} \\ &\leq \frac{n}{2\eta}\log\left(1+\frac{1}{64}\sum_{t=1}^{T}\left(\frac{8}{t}+1\right)^2\right)+\frac{1}{2\eta}+\sum_{t=1}^{T}\frac{9DG}{t}+\sum_{t=1}^{T}\frac{4DG}{t} \\ &\leq \frac{n}{2\eta}\log\left(1+\sum_{t=1}^{T}\left(\frac{1}{t^2}+\frac{1}{4t}+1\right)\right)+\frac{1}{2\eta}+\sum_{t=1}^{T}\frac{13DG}{t} \\ &\leq \frac{n}{2\eta}\log\left(1+\frac{5}{4}\log T+T\right)+\frac{1}{2\eta}+13DG\log T \\ &\leq \left(DG+\frac{1}{\alpha}\right)n\log\left(1+\frac{5}{4}\log T+T\right)+\left(DG+\frac{1}{\alpha}\right)+13DG\log T \\ &= O\left(\left(DG+\frac{1}{\alpha}\right)n\log(T+\log T)\right).\end{aligned} \tag{44}$$

Thus, the theorem follows. □

B.4. Proof of Theorem 2

Theorem 2. *Algorithm 4 with parameters $\eta=\frac{1}{2DL}$ and $\epsilon=L$, can achieve the regret bound $O\left(\frac{DLn}{\alpha}\log(T+1)\right)$.*

Proof. By the assumption, let $x^*\in\arg\min_{x\in\mathcal{K}}\sum_{t=1}^{T} f_t(x)$, for the fixed $y=x^*$, for all $t\in[T]$, we have

$$f_t(x_t)-f_t(x^*)\leq \nabla f_t(x_t)^{\mathrm{T}}(x_t-x^*)-\frac{\eta}{2}(x_t-x^*)^{\mathrm{T}}H(f_t)(x_t)(x_t-x^*). \tag{45}$$

By the update rule for x_{t+1} and the Pythagorean theorem, there is

$$\begin{aligned}\|x_{t+1}-x^*\|_{A_t}^2 &= \left\|\hat{P}_{\mathcal{K}}^{(A_t)}\left(x_t-\frac{1}{\eta_t}A_t^{-1}\nabla f_t(x_t)\right)-x^*\right\|_{A_t}^2 \\ &\leq \left\|x_t-\frac{1}{\eta_t}A_t^{-1}\nabla f_t(x_t)-x^*\right\|_{A_t}^2 \\ &\leq (x_t-x^*)^{\mathrm{T}}A_t(x_t-x^*)-\frac{2}{\eta_t}\nabla f_t(x_t)^{\mathrm{T}}(x_t-x^*)+\frac{1}{\eta_t^2}\nabla f_t(x_t)^{\mathrm{T}}A_t^{-1}\nabla f_t(x_t).\end{aligned} \tag{46}$$

Since $\|x_{t+1}-x^*\|_{A_t}^2=(x_{t+1}-x^*)^{\mathrm{T}}A_t(x_{t+1}-x^*)$, combine with Inequality (46), we have

$$\nabla f_t(x_t)^{\mathrm{T}}(x_t-x^*)\leq\frac{1}{2\eta_t}\nabla f_t(x_t)^{\mathrm{T}}A_t^{-1}\nabla f_t(x_t)+\frac{\eta_t}{2}(x_t-x^*)^{\mathrm{T}}A_t(x_t-x^*)$$

$$- \frac{\eta_t}{2}(x_{t+1} - x^*)^{\mathrm{T}} A_t (x_{t+1} - x^*). \tag{47}$$

summing Inequality (47) from $t = 1$ to T, we have

$$\begin{aligned}
\sum_{t=1}^{T} \nabla f_t(x_t)^{\mathrm{T}}(x_t - x^*) &\leq \sum_{t=1}^{T} \frac{1}{2\eta_t} \nabla f_t(x_t)^{\mathrm{T}} A_t^{-1} \nabla f_t(x_t) + \sum_{t=1}^{T} \frac{\eta_t}{2}(x_t - x^*)^{\mathrm{T}} A_t (x_t - x^*) \\
&\quad - \sum_{t=1}^{T} \frac{\eta_t}{2}(x_{t+1} - x^*)^{\mathrm{T}} A_t (x_{t+1} - x^*) \\
&= \sum_{t=1}^{T} \frac{1}{2\eta_t} \nabla f_t(x_t)^{\mathrm{T}} A_t^{-1} \nabla f_t(x_t) + \frac{\eta_1}{2}(x_1 - x^*)^{\mathrm{T}} A_1 (x_1 - x^*) \\
&\quad + \frac{1}{2} \sum_{t=2}^{T} (x_t - x^*)^{\mathrm{T}} (\eta_t A_t - \eta_{t-1} A_{t-1})(x_t - x^*) \\
&\quad - \frac{\eta_T}{2}(x_{T+1} - x^*)^{\mathrm{T}} A_T (x_{T+1} - x^*).
\end{aligned} \tag{48}$$

Set $\eta_1 = \eta_2 = \cdots = \eta_T = \eta$, we have

$$\begin{aligned}
\sum_{t=1}^{T} \nabla f_t(x_t)^{\mathrm{T}}(x_t - x^*) &\leq \sum_{t=1}^{T} \frac{1}{2\eta} \nabla f_t(x_t)^{\mathrm{T}} A_t^{-1} \nabla f_t(x_t) + \frac{\eta}{2}(x_1 - x^*)^{\mathrm{T}} A_1 (x_1 - x^*) \\
&\quad + \frac{\eta}{2} \sum_{t=2}^{T} (x_t - x^*)^{\mathrm{T}} (A_t - A_{t-1})(x_t - x^*) \\
&\quad - \frac{\eta}{2}(x_{T+1} - x^*)^{\mathrm{T}} A_T (x_{T+1} - x^*) \\
&= \sum_{t=1}^{T} \frac{1}{2\eta} \nabla f_t(x_t)^{\mathrm{T}} A_t^{-1} \nabla f_t(x_t) + \frac{\eta}{2}(x_1 - x^*)^{\mathrm{T}} (A_1 - H_1)(x_1 - x^*) \\
&\quad + \frac{\eta}{2} \sum_{t=1}^{T} (x_t - x^*)^{\mathrm{T}} H_t (x_t - x^*) - \frac{\eta}{2}(x_{T+1} - x^*)^{\mathrm{T}} A_T (x_{T+1} - x^*) \\
&\leq \sum_{t=1}^{T} \frac{1}{2\eta} \nabla f_t(x_t)^{\mathrm{T}} A_t^{-1} \nabla f_t(x_t) + \frac{\epsilon \eta D^2}{2} + \frac{\eta}{2} \sum_{t=1}^{T} (x_t - x^*)^{\mathrm{T}} H_t (x_t - x^*).
\end{aligned} \tag{49}$$

For the first term in the right hand side of Inequality (49), we have

$$\begin{aligned}
\sum_{t=1}^{T} \frac{1}{2\eta} \nabla f_t(x_t)^{\mathrm{T}} A_t^{-1} \nabla f_t(x_t) &= \frac{1}{2\eta} \sum_{t=1}^{T} \mathrm{Tr}\left(A_t^{-1} \nabla f_t(x_t) \nabla f_t(x_t)^{\mathrm{T}}\right) \\
&\leq \frac{1}{2\alpha\eta} \sum_{t=1}^{T} \mathrm{Tr}\left(A_t^{-1} H_t\right) \\
&= \frac{1}{2\alpha\eta} \sum_{t=1}^{T} \mathrm{Tr}\left(A_t^{-1}(A_t - A_{t-1})\right) \\
&\leq \frac{1}{2\eta} \log\left(\det\left(A_T\right) \det\left(A_0^{-1}\right)\right) \\
&= \frac{1}{2\alpha\eta} \log\left(\det\left(\left(\epsilon I + \sum_{t=1}^{T} H_t\right)(\epsilon I)^{-1}\right)\right) \\
&\leq \frac{n}{2\alpha\eta} \log\left((\epsilon + LT)\,\epsilon^{-1}\right).
\end{aligned} \tag{50}$$

set $\epsilon = L, \eta = \frac{1}{2DL}$. Summing Inequality (45) from $t = 1$ to T, and combine Inequality (49) (50), we have

$$\begin{aligned}\sum_{t=1}^{T}(f_t(x_t) - f_t(x^*)) &\leq \frac{n}{2\alpha\eta}\log\left((\epsilon + LT)\,\epsilon^{-1}\right) + \frac{\epsilon\eta D^2}{2} \\ &\leq O\left(\frac{DLn}{\alpha}\log(T+1)\right). \end{aligned} \tag{51}$$

It therefore gives the theorem. □

B.5. Proof of Lemma 4

Lemma 4. *In the non-smooth version of Algorithm 2, for all timestep t, let $f_t : \mathcal{K} \to \mathbb{R}$ be α-exp-concave functions with Lipschitz parameter G, where $\mathcal{K}$ is a convex set with diameter D, then for any $y \in \mathcal{K}$ and any $\eta_t \leq \min\left\{\frac{1}{8GD}, \frac{\alpha}{2}\right\}$, the estimated gradient $\widetilde{\nabla} f_t(x_t)$ satisfies*

$$\begin{aligned} f_t(y) \geq & f_t(x_t) + \widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y - x_t) \\ & + \frac{\eta_t}{2}(y - x_t)^{\mathrm{T}}\widetilde{\nabla} f_t(x_t)\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y - x_t) \\ & - \frac{9D}{8}\left\|\nabla f_t(z_t) - \widetilde{\nabla} f_t(z_t)\right\|_1 \\ & - \frac{D}{16G}\left\|\nabla f_t(z_t) - \widetilde{\nabla} f_t(z_t)\right\|_1^2 \\ & - 3G\sqrt{n}r_t' - \frac{nr_t'^2 G}{2}. \end{aligned} \tag{7}$$

Proof. Note that we omit the subscript t in the proofs of the lemmas as they hold for each timestep t. By the property of α-exp-concave

$$f(y) \geq f(z) + \nabla f(z)^{\mathrm{T}}(y - z) + \frac{\eta}{2}(y - z)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y - z), \tag{52}$$

For the first and the second term on the right hand side of Inequality (52),

$$\begin{aligned} f(z) + \nabla f(z)^{\mathrm{T}}(y - z) &= f(z) + \nabla f(z)^{\mathrm{T}}(y - z) + (\widetilde{\nabla} f(x)^{\mathrm{T}}(y - x) - \widetilde{\nabla} f(x)^{\mathrm{T}}(y - x)) + (f(x) - f(x)) \\ &= f(x) + \widetilde{\nabla} f(x)^{\mathrm{T}}(y - x) + (\nabla f(z) - \widetilde{\nabla} f(x))^{\mathrm{T}}(y - x) + (f(z) - f(x)) + \nabla f(z)^{\mathrm{T}}(x - z) \\ &\geq f(x) + \widetilde{\nabla} f(x)^{\mathrm{T}}(y - x) - \|\nabla f(z) - \widetilde{\nabla} f(x)\|_1\|y - x\|_\infty - G\|z - x\|_2 - \|\nabla f(z)\|_2\|x - z\|_2 \\ &\geq f(x) + \widetilde{\nabla} f(x)^{\mathrm{T}}(y - x) - \|\nabla f(z) - \widetilde{\nabla} f(x)\|_1\|y - x\|_2 - 2G\sqrt{n}r'. \\ &\geq f(x) + \widetilde{\nabla} f(x)^{\mathrm{T}}(y - x) - D\|\nabla f(z) - \widetilde{\nabla} f(x)\|_1 - 2G\sqrt{n}r'. \end{aligned} \tag{53}$$

Further, for the third term on the right hand side of Inequality (52),

$$\begin{aligned} & \frac{\eta}{2}(y - z)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y - z) \\ =& \frac{\eta}{2}(y - x)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y - x) + \frac{\eta}{2}(y - z)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y - z) \\ & - \frac{\eta}{2}(y - x)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y - x) \\ =& \frac{\eta}{2}(y - x)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y - x) + \frac{\eta}{2}(y - z)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y - z) \\ & - \frac{\eta}{2}(y - z)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y - x) + \frac{\eta}{2}(y - z)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y - x) \\ & - \frac{\eta}{2}(y - x)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y - x) \\ =& \frac{\eta}{2}(y - x)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y - x) + \frac{\eta}{2}(y - z)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(x - z) \end{aligned}$$

$$
\begin{aligned}
&+\frac{\eta}{2}(x-z)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y-x)\\
=&\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y-x)+\frac{\eta}{2}(x-z)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(x-z)\\
&+\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(x-z)+\frac{\eta}{2}(x-z)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y-x)\\
\geq&\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y-x)-\frac{\eta}{2}\|x-z\|^2\|\nabla f(z)\|^2-\eta\|x-z\|\|\nabla f(z)\|^2\|y-x\|\\
\geq&\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y-x)-\frac{\eta n r'^2G^2}{2}-\eta\sqrt{n}r'DG^2\\
\geq&\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y-x)-\frac{nr'^2G}{2}-\sqrt{n}r'G.
\end{aligned}
\tag{54}
$$

We then handle the term $\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y-x)$ with the same technique as shown in Inequality (31), we have

$$
\begin{aligned}
\frac{\eta}{2}(y-x)^{\mathrm{T}}\nabla f(z)\nabla f(z)^{\mathrm{T}}(y-x)\geq&\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(z)\widetilde{\nabla} f(z)^{\mathrm{T}}(y-x)\\
&-\frac{D}{8}\left\|\nabla f(z)-\widetilde{\nabla} f(z)\right\|_1-\frac{D}{16G}\left\|\nabla f(z)-\widetilde{\nabla} f(z)\right\|_1^2\\
=&\frac{\eta}{2}(y-x)^{\mathrm{T}}\widetilde{\nabla} f(x)\widetilde{\nabla} f(x)^{\mathrm{T}}(y-x)\\
&-\frac{D}{8}\left\|\nabla f(z)-\widetilde{\nabla} f(z)\right\|_1-\frac{D}{16G}\left\|\nabla f(z)-\widetilde{\nabla} f(z)\right\|_1^2.
\end{aligned}
\tag{55}
$$

In summary, we have

$$
\begin{aligned}
f_t(y)\geq& f_t(x_t)+\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y-x_t)+\frac{\eta_t}{2}(y-x_t)^{\mathrm{T}}\widetilde{\nabla} f_t(x_t)\widetilde{\nabla} f_t(x_t)^{\mathrm{T}}(y-x_t)\\
&-\frac{9D}{8}\left\|\nabla f(z_t)-\widetilde{\nabla} f(z_t)\right\|_1-\frac{D}{16G}\left\|\nabla f(z_t)-\widetilde{\nabla} f(z_t)\right\|_1^2-3G\sqrt{n}r'-\frac{nr'^2G}{2}.
\end{aligned}
\tag{56}
$$

which gives the lemma. □

B.6. Proof of Theorem 3

Theorem 3. *The non-smooth version of Algorithm 2 with parameters* $\eta=\min\left\{\frac{1}{8GD},\frac{\alpha}{2}\right\}$, $\epsilon=\frac{1}{\eta^2D^2}$, $\left\{r_t=\frac{\rho p}{\pi n^{9/2}(n/\rho+1)t^2}\right\}_{t=1}^T$, $\left\{r'_t=\frac{1}{t\sqrt{n}}\right\}_{t=1}^T$ *can achieve the regret bound* $O\left(\left(DG+\frac{1}{\alpha}\right)n\log(T+\log T)\right)$ *with probability greater than* $1-T(\rho+p)$*, and its query complexity is* $O(1)$ *in each round.*

Proof. Similar to the proof of Theorem 1, we have

$$
\begin{aligned}
\sum_{t=1}^T f_t(x_t)-f_t(x^*)\leq&\frac{n}{2\eta}\log\left(\left(\epsilon+\sum_{t=1}^T\left(8\pi n^3(n/\rho+1)\beta r_t/\rho+G\right)^2\right)\epsilon^{-1}\right)+\frac{\epsilon\eta D^2}{2}\\
&+\sum_{t=1}^T 9D\pi n^3(n/\rho+1)\beta r_t/\rho+\sum_{t=1}^T\frac{4D(\pi n^3(n/\rho+1)\beta r_t/\rho)^2}{G}\\
&+\sum_{t=1}^T 3G\sqrt{n}r'_t+\sum_{t=1}^T\frac{nr_t'^2G}{2}.
\end{aligned}
\tag{57}
$$

By Lemma 3, we let $\beta=nG/p_tr'_t$. Set $p_1=p_2=\cdots=p_T=p$. Then, set $\eta=\min\left\{\frac{1}{8GD},\frac{\alpha}{2}\right\},\epsilon=\frac{1}{\eta^2D^2},\left\{r_t=\frac{\rho p}{\pi n^{9/2}(n/\rho+1)t^2}\right\}_{t=1}^T,\left\{r'_t=\frac{1}{t\sqrt{n}}\right\}_{t=1}^T$ we have

$$
\sum_{t=1}^T f_t(x_t)-f_t(x^*)\leq\frac{n}{2\eta}\log\left(1+\sum_{t=1}^T\left(\frac{8G}{t}+G\right)^2\eta^2D^2\right)+\frac{1}{2\eta}+\sum_{t=1}^T\frac{9DG}{t}+\sum_{t=1}^T\frac{4DG^2}{t^2G}+\sum_{t=1}^T\frac{3G}{t}+\sum_{t=1}^T\frac{G}{2t^2}
$$

$$\leq \left(4DG + \frac{1}{\alpha}\right) n \log\left(1 + \frac{5}{4}\log T + T\right) + \left(4DG + \frac{1}{\alpha}\right) + 13DG\log T + \frac{7}{2}G\log T$$
$$= O\left(\left(DG + \frac{1}{\alpha}\right) n \log(T + \log T)\right). \tag{58}$$

Thus, the theorem follows. □