Quantum Algorithms and Lower Bounds for Finite-Sum Optimization

Yexin Zhang [*] [1] **Chenyi Zhang** [*] [2] **Cong Fang** [3] [4] **Liwei Wang** [3] [4] **Tongyang Li** [5] [6]

Abstract

Finite-sum optimization has wide applications in machine learning, covering important problems such as support vector machines, regression, etc. In this paper, we initiate the study of solving finite-sum optimization problems by quantum computing. Specifically, let $f_1, \dots, f_n \colon \mathbb{R}^d \to \mathbb{R}$ be ℓ-smooth convex functions and $\psi \colon \mathbb{R}^d \to \mathbb{R}$ be a μ-strongly convex proximal function. The goal is to find an ϵ-optimal point for $F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x}) + \psi(\mathbf{x})$. We give a quantum algorithm with complexity $\tilde{O}\big(n + \sqrt{d} + \sqrt{\ell/\mu}\big(n^{1/3}d^{1/3} + n^{-2/3}d^{5/6}\big)\big)$,[1] improving the classical tight bound $\tilde{\Theta}\big(n + \sqrt{n\ell/\mu}\big)$. We also prove a quantum lower bound $\tilde{\Omega}(n + n^{3/4}(\ell/\mu)^{1/4})$ when d is large enough. Both our quantum upper and lower bounds can extend to the cases where ψ is not necessarily strongly convex, or each f_i is Lipschitz but not necessarily smooth. In addition, when F is nonconvex, our quantum algorithm can find an ϵ-critial point using $\tilde{O}(n + \ell(d^{1/3}n^{1/3} + \sqrt{d})/\epsilon^2)$ queries.

1. Introduction

In machine learning, especially supervised learning, it is common that the overall loss function can be written as a sum of loss functions at each data point. In particular, let $f_1, \dots, f_n \colon \mathbb{R}^d \to \mathbb{R}$ be a sequence of functions, our goal is to find an approximate minimum of the function

$$F(\mathbf{x}) \coloneqq f(\mathbf{x}) + \psi(\mathbf{x}), \tag{1}$$

where $f(\mathbf{x})$ satisfies

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \tag{2}$$

and $\psi(\mathbf{x})$ is a known convex function, sometimes referred to as the *proximal function*. For instance, given n training data $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$ where $\mathbf{x}_i \in \mathbb{R}^{d-1}, \mathbf{y}_i \in \mathbb{R}$ for each $i \in [n]$, if f_i is the square loss $f_i(\mathbf{w}, b) = (\mathbf{w}^\top x_i - y_i - b)^2$ for $\mathbf{w} \in \mathbb{R}^{d-1}$ and $b \in \mathbb{R}$, then Eq. (2) gives linear least squares regression of these data. In general, such finite-sum optimization problems arises in many places in machine learning, statistics, and operations research, such as support vector machines, logistic regression, Lasso, etc.

Finite-sum optimization has been widely studied in previous literature given their wide applicability. Zhang (2004) solved finite-sum optimization by randomly selecting an index $i \in [n]$ and applying stochastic gradient descent (SGD). More efficient algorithms apply variance reduction, including SAG by Roux et al. (2012), SDCA by Shalev-Shwartz & Zhang (2013), SVRG by Johnson & Zhang (2013), and also many other works (Zhang et al., 2013; Defazio et al., 2014; Xiao & Zhang, 2014; Allen-Zhu & Yuan, 2016; Allen-Zhu & Hazan, 2016b; Reddi et al., 2016; Shalev-Shwartz, 2016; Allen-Zhu, 2017; 2018; Lin et al., 2018).

More recently, quantum computing is rapidly advancing and there is also significant interest in quantum algorithms for continuous optimization faster than classical counterparts. This began with quantum algorithms for solving linear and semidefinite programs (Brandão & Svore, 2017; van Apeldoorn & Gilyén, 2019; Brandão et al., 2019; van Apeldoorn et al., 2017; Casares & Martin-Delgado, 2020; Kerenidis & Prakash, 2020), then for general convex optimization (Apeldoorn et al., 2020; Chakrabarti et al., 2020), and now there are also quantum algorithms for slightly nonconvex problems (Li & Zhang, 2022; Chen et al., 2023), escaping saddle points in nonconvex landscapes (Zhang et al., 2021; Childs et al., 2022), and also finding global minima in some special classes of nonconvex problems (Liu et al., 2023; Leng et al., 2023). On the other hand, quantum lower bounds for convex optimization (Garg et al., 2021; Garg et al., 2021)

[*] Equal contribution [1] School of Electronics Engineering and Computer Science, Peking University, China [2] Computer Science Department, Stanford University, USA [3] National Key Lab of General Artificial Intelligence, School of Intelligence Science and Technology, Peking University [4] Institute for Artificial Intelligence, Peking University [5] Center on Frontiers of Computing Studies, Peking University, China [6] School of Computer Science, Peking University, China. Correspondence to: Tongyang Li <tongyangli@pku.edu.cn>, Liwei Wang <wanglw@pku.edu.cn>, Cong Fang <fangcong@pku.edu.cn>.

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

[1] The $\tilde{O}$ and $\tilde{\Omega}$ notations omit poly-logarithmic terms, i.e., $\tilde{O}(f) = O(f\,\mathrm{poly}(\log f))$ and $\tilde{\Omega}(f) = \Omega(f\,\mathrm{poly}(\log f))$.

and nonconvex optimization (Gong et al., 2022; Zhang & Li, 2023) are also established. However, these optimization results mainly investigate the block-box setting with a single function $f(x)$, and the perspective of quantum algorithms for finite-sum stochastic optimization is widely open.

Contributions. In this work, we initiate the study of the quantum analogue of the standard finite-sum optimization problem. We assume quantum access to a finite-sum oracle, or access to a *quantum finite-sum oracle* for brevity, that allows us to query the gradients of different f_i's at the same time in *quantum superpositions*.

Definition 1 (Quantum finite-sum oracle)**.** *For an $F\colon \mathbb{R}^d \to \mathbb{R}$ and sub-functions $f_1, \dots, f_n\colon \mathbb{R}^d \to \mathbb{R}$ satisfying the finite-sum structure in Eq. (2), its quantum finite-sum oracle O_F is defined as*[2]

$$O_F \left|\mathbf{x}\right\rangle \otimes \left|i\right\rangle \otimes \left|0\right\rangle \to \left|\mathbf{x}\right\rangle \otimes \left|i\right\rangle \otimes \left|\nabla f_i(\mathbf{x})\right\rangle \quad \forall i \in [n]. \quad (3)$$

Here, the Dirac notation $|\cdot\rangle$ denotes input or output registers made of qubits that may present as *quantum superpositions*. Specifically, for an $\mathbf{x} \in \mathbb{R}^d$ and a coefficient vector $\mathbf{c} \in \mathbb{C}^n$ with $\sum_{i\in[n]} |c_i|^2 = 1$, the quantum register could be in the quantum state $\left|\mathbf{x}\right\rangle \sum_{i\in[n]} c_i \left|i\right\rangle \otimes \left|\nabla f_i(\mathbf{x})\right\rangle$, which is a quantum superposition over all these n sub-functions simultaneously.[3] If we measure this quantum state, we will get $\nabla f_i(\mathbf{x})$ with probability $|c_i|^2$. If we query O_F with no superposition over the first two registers, it collapses to a classical finite-sum oracle that returns the gradient of a specific f_i at $\mathbf{x}$.

Next, we characterize the objective functions using the following properties:

- L-Lipschitzness: For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,
$$\|f(\mathbf{x}) - f(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|.$$
- μ-strong convexity: For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,
$$f(\mathbf{x}) - f(\mathbf{y}) \le \nabla f(\mathbf{x})^\top(\mathbf{x} - \mathbf{y}) - \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2.$$
- ℓ-smoothness: For any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,
$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le \ell\|\mathbf{x} - \mathbf{y}\|.$$

Finite-sum convex optimization. In this work, we systematically investigate the quantum analogue of the finite-sum convex optimization problem.

Problem 1 (Quantum finite-sum convex optimization (QFCO))**.** *In the quantum finite-sum convex optimization (QFCO) problem we are given query access to a quantum finite-sum oracle O_F for a convex function $F\colon \mathbb{R}^d \to \mathbb{R}$ satisfying (1). The goal is to output an expected ϵ-optimal point $\mathbf{x}^* \in \mathbb{R}^d$ satisfying $\mathbb{E}[F(\mathbf{x}^*)] \le \inf_{\mathbf{x}} F(\mathbf{x}) + \epsilon$.*

We consider the following four cases of Problem 1:

1. ψ is μ-strongly convex, each f_i is convex and ℓ-smooth, and $F(\mathbf{0}) - F^* \le \Delta$. Example: ridge regression, elastic net regularization;
2. ψ is not necessarily strongly convex, each f_i is convex and ℓ-smooth, and F achieves its minimum at $\mathbf{x}^*$ with $\|\mathbf{x}^*\| \le R$. Example: logistic regression, Lasso;
3. ψ is μ-strongly convex, each f_i is convex and L-Lipschitz (not necessarily smooth), and $F(\mathbf{0}) - F^* \le \Delta$. Example: ℓ_2-norm support vector machine;
4. ψ is not necessarily strongly convex, each f_i is convex and L-Lipschitz (not necessarily smooth), and F achieves its minimum at $\mathbf{x}^*$ with $\|\mathbf{x}^*\| \le R$. Example: ℓ_1-norm support vector machine.

We develop quantum algorithms for all the four cases of Problem 1 respectively in Section 3. The query complexities of these algorithms are summarized in the following.

Theorem 1 (Informal version of Theorem 4 and Corollary 2)**.** *There exist four quantum algorithms that solve all the cases of Problem 1, respectively, with the following query complexities:*

- *Case 1:* $\tilde{O}\big(n + \sqrt{d} + \sqrt{\frac{\ell}{\mu}}\big(n^{1/3}d^{1/3} + n^{-2/3}d^{5/6}\big)\big)$*;*
- *Case 2:* $\tilde{O}\big(n + \sqrt{d} + R\sqrt{\frac{\ell}{\epsilon}}\big(n^{1/3}d^{1/3} + n^{-2/3}d^{5/6}\big)\big)$*;*
- *Case 3:* $\tilde{O}\big(n + \sqrt{d} + \frac{L}{\sqrt{\lambda\mu}}\big(n^{1/3}d^{1/3} + n^{-2/3}d^{5/6}\big)\big)$*;*
- *Case 4:* $\tilde{O}\big(n + \sqrt{d} + \frac{LR}{\epsilon}\big(n^{1/3}d^{1/3} + n^{-2/3}d^{5/6}\big)\big)$*.*

Compared to Allen-Zhu (2017), our quantum algorithms achieve a better query complexity when the dimension d is relatively small. Prior to our work, Ozgul et al. (2023) studied non-logconcave sampling from a distribution $\pi(\mathbf{x}) \propto \exp(-\beta f(\mathbf{x}))$ where $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x})$ is a

[2] In this paper, whenever we access a quantum oracle U, it is a unitary operation and we also have access to its corresponding inverse operation denoted as $U^\dagger$, which satisfies $U^\dagger U = UU^\dagger = I$. This is a standard assumption, explicitly or implicitly employed, in previous research on quantum algorithms, see e.g., (Brassard et al., 2002; Cornelissen et al., 2022; Sidford & Zhang, 2023).

[3] Note that we store real numbers in these quantum registers, which solicit encoding from binary numbers to real numbers. Similar to classical encoding, we represent a number as a binary string and store each bit in a quantum bit. For example, for the real number 3.25, we encode by its binary representation 11.01, and then represent this number using four qubits. This method is commonly used when real numbers need to be manipulated mathematically in quantum algorithms. Of course, this encoding method also incurs errors when encoding real numbers, just like classical encoding, thus we may consider precision beforehand when using this encoding method. Basic operations can be implemented with constant time overhead (Nielsen & Chuang, 2000).

finite sum of functions. As $\pi(\mathbf{x})$ is larger when $f(\mathbf{x})$ is smaller, measuring such a distribution can in principle solve the finite-sum optimization problem. However, compared to our result, there algorithms have dimension factor being at least $\Omega(d)$, and it is also not clear how large β should be to reach the same criteria in optimization. As far as we know, we give the first quantum algorithm with speedup for finite-sum convex optimization. A more detailed comparison can be found in Table 1.

We also establish quantum complexity lower bounds for the four cases of Problem 1 respectively in Section 5, with the specific forms as follows. These quantum lower bounds confirm that the speedup for finite-sum optimization by quantum computing is at most polynomial in all parameters.

Theorem 2 (Informal version of Corollary 3, Corollary 4, Corollary 5, and Corollary 6)**.** *There exist four families of functions corresponding to the four cases such that when d is large enough, any quantum algorithm that finds an ϵ-optimal point requires the following query complexities:*

- *Case 1:* $\tilde{\Omega}\left(n + n^{\frac{3}{4}}\left(\frac{\ell}{\mu}\right)^{\frac{1}{4}}\right)$;
- *Case 2:* $\tilde{\Omega}\left(n + n^{\frac{3}{4}}\left(\frac{\ell}{\epsilon}\right)^{\frac{1}{4}} R^{\frac{1}{2}}\right)$;
- *Case 3:* $\tilde{\Omega}\left(n + n^{\frac{3}{4}}\left(\frac{1}{\epsilon\mu}\right)^{\frac{1}{4}} L^{\frac{1}{2}}\right)$;
- *Case 4:* $\tilde{\Omega}\left(n + n^{\frac{3}{4}}\left(\frac{LR}{\epsilon}\right)^{\frac{1}{2}}\right)$.

Finite-sum critical point computation. Additionally, we develop a quantum algorithm for finding critical points, i.e., points with small gradients, of (possibly) non-convex functions in the finite-sum setting.

Problem 2 (Quantum finite-sum critical point computation (QFCP))**.** *In the quantum finite-sum critical point computation (QFCP) problem we are given query access to a quantum finite-sum oracle O_F for a (possibly) nonconvex function $F\colon \mathbb{R}^d \to \mathbb{R}$ satisfying (1) where $\psi(\mathbf{x}) \equiv 0$ and each f_i is ℓ-smooth. Moreover, $F(\mathbf{0}) - F^* \leq \Delta$. The goal is to output an expected ϵ-critical point $\mathbf{x} \in \mathbb{R}^d$ satisfying $\mathbb{E}\|\nabla f(\mathbf{x})\| \leq \epsilon$.*

Leveraging Fang et al. (2018) and Sidford & Zhang (2023), we develop a quantum algorithm that solves Problem 2:

Theorem 3 (Informal version of Theorem 5)**.** *There exist a quantum algorithm that solves Problem 2 using an expected $\tilde{O}\big(n + \big(d^{1/3}n^{1/3} + \sqrt{d}\big)/\epsilon^2\big)$ queries.*

Compared to Fang et al. (2018) using $O(n + n^{1/2}/\epsilon^2)$ queries, our quantum algorithm achieves a better query complexity when $d < \sqrt{n}$.

Techniques. In our quantum algorithms, our main contribution is leveraging quantum variance reduction with speedup on the variance reduction step in the state-of-the-art finite-sum optimization algorithms, in particular Katyusha (Allen-Zhu, 2017) for the convex setting and SPIDER (Fang et al., 2018) for the nonconvex setting. Technically, our quantum speedup is originated from quantum mean estimation (Montanaro, 2015). Classical i.i.d. random variables with variance σ^2 need $\Omega(\sigma^2/\epsilon^2)$ samples to approximate their mean within ϵ with high success probability (Dagum et al., 2000), but quantum mean estimation by Montanaro (2015) achieves sample complexity $\tilde{O}(\sigma/\epsilon)$. However, most existing quantum mean estimation algorithms (Hamoudi & Magniez, 2019; Hamoudi, 2021; Cornelissen et al., 2022; Kothari & O'Donnell, 2023) have bias, which hinders their combination with Katyusha (Allen-Zhu, 2017) and SPIDER (Fang et al., 2018) assuming unbiased inputs. Since Katyusha and SPIDER are iterative algorithms, bias would accumulate during the algorithm, jeopardizing their convergence guarantee. Therefore, to achieve quantum speedup for finite-sum optimization, an unbiased quantum mean estimator is solicited. We apply the version by Sidford & Zhang (2023), which employs a classical multi-level Monte-Carlo (MLMC) scheme (Blanchet & Glynn, 2015; Asi et al., 2021) to the multivariate mean estimation algorithm by Cornelissen et al. (2022) to obtain an unbiased mean estimation. Note that Cornelissen & Hamoudi (2023) also developed an almost unbiased quantum mean estimation algorithm. However, it is applicable only in the one-dimensional case and retains a slight degree of bias, making it complicated when integrating with high-dimensional optimization algorithms.

In our quantum lower bounds, our primary contribution is the combination of the classical randomized lower bounds on finite-sum optimization (Woodworth & Srebro, 2016) with quantum adversary methods. Existing quantum lower bounds for optimization problems, such as Garg et al. (2021); Zhang & Li (2022), were proved by the "zero-chain" approach. They represented quantum algorithms as sequences of unitaries and proved their lower bounds by a hybrid argument where the information accessible at each step of the algorithm is restricted. However, in the finite-sum problem, since quantum algorithms have the capability to access different sub-functions at the same time, such a hybrid argument is not valid, and we cannot assume that the algorithm accesses these sub-functions in a specific predetermined order. To address this issue, we apply the quantum adversary method first introduced in Ambainis (2000), which extends the hybrid method and takes an average over many pairs of inputs. However, the original quantum adversary method in Ambainis (2000; 2006) were designed for lower bounding the query complexity of boolean functions, but finite-sum optimization outputs a vector rather than a single bit. Therefore, we employ a more powerful version

Table 1: Comparisons between algorithms and lower bounds for finite-sum convex optimization, both in classical and quantum settings. The columns from left to right cover the four cases respectively. n is the number of functions f_i, ϵ is the error to the optimal value, Δ is an upper bound on the difference between $F(\mathbf{0})$ and the optimal value, and R is an upper bound on the norm of the optimum $\mathbf{x}^*$.

	ℓ-Smooth		L-Lipschitz	
	μ-Strongly Convex	Convex	μ-Strongly Convex	Convex
Classical Upper Bound (Allen-Zhu, 2017)	$\tilde{O}\left(n+\sqrt{\frac{n\ell}{\mu}}\right)$	$\tilde{O}\left(n+R\sqrt{\frac{n\ell}{\epsilon}}\right)$	$\tilde{O}\left(n+L\sqrt{\frac{n}{\mu\epsilon}}\right)$	$\tilde{O}\left(n+LR\frac{\sqrt{n}}{\epsilon}\right)$
Classical Lower Bound (Woodworth & Srebro, 2016)	$\Omega\left(n+\sqrt{\frac{n\ell}{\mu}}\log\frac{\Delta}{\epsilon}\right)$	$\Omega\left(n+R\sqrt{\frac{n\ell}{\epsilon}}\right)$	$\Omega\left(n+\frac{\sqrt{n}L}{\sqrt{\mu\epsilon}}\right)$	$\Omega\left(n+\frac{\sqrt{n}LR}{\sqrt{\epsilon}}\right)$
Quantum Upper Bound **(this work)**	$\tilde{O}(n+\sqrt{d}+\sqrt{\frac{\ell}{\mu}}(n^{\frac{1}{3}}d^{\frac{1}{3}}+n^{-\frac{2}{3}}d^{\frac{5}{6}}))$	$\tilde{O}(n+\sqrt{d}+R\sqrt{\frac{\ell}{\epsilon}}(n^{\frac{1}{3}}d^{\frac{1}{3}}+n^{-\frac{2}{3}}d^{\frac{5}{6}}))$	$\tilde{O}(n+\sqrt{d}+\frac{L}{\sqrt{\lambda\mu}}(n^{\frac{1}{3}}d^{\frac{1}{3}}+n^{-\frac{2}{3}}d^{\frac{5}{6}}))$	$\tilde{O}(n+\sqrt{d}+\frac{LR}{\epsilon}(n^{\frac{1}{3}}d^{\frac{1}{3}}+n^{-\frac{2}{3}}d^{\frac{5}{6}}))$
Quantum Lower Bound **(this work)**	$\tilde{\Omega}\left(n+n^{\frac{3}{4}}\left(\frac{\ell}{\mu}\right)^{\frac{1}{4}}\right)$	$\tilde{\Omega}\left(n+n^{\frac{3}{4}}\left(\frac{\ell}{\epsilon}\right)^{\frac{1}{4}}R^{\frac{1}{2}}\right)$	$\tilde{\Omega}\left(n+n^{\frac{3}{4}}\left(\frac{1}{\epsilon\mu}\right)^{\frac{1}{4}}L^{\frac{1}{2}}\right)$	$\tilde{\Omega}\left(n+n^{\frac{3}{4}}\left(\frac{LR}{\epsilon}\right)^{\frac{1}{2}}\right)$

of the adversary method in Zhang (2005) that extends to general non-boolean functions. The proofs for our quantum query lower bounds consist of the following steps:

1. Adapt the hard instance (Definition 3) corresponding to the random algorithms introduced by Woodworth & Srebro (2016). Prove that we need to obtain enough information from over half of the sub-functions f_i to find the ϵ-optimal point. Notice that in the classical case, it is proved that we must obtain information about the vectors later in the sequence of a sub-function to find the ϵ-optimal point. In our proof, to match with the adversary method, we establish a stronger conclusion: we need to acquire information about every vector to find the ϵ-optimal point of a sub-function (Lemma 5).

2. Construct the hard instance such that each sub-function f_i can only be accessed sequentially to obtain the function's construction. Reduce the problem to a quantum computing problem of determining the values of all elements in a binary matrix with a sequential verification oracle (Problem 4 in appendices).

3. For the reduced problem, employ quantum adversary methods (Lemma 9 in appendices) to prove its corresponding quantum complexity lower bound.

Open questions. Our work leaves several natural directions for future investigation:

- Can we give quantum algorithms for finite-sum optimization with better complexities? On the other hand, can we prove tighter quantum lower bounds on finite-sum optimization? Specifically, we prove our quantum lower bound by reducing to the matrix detection problem (Problem 4 in appendices), which for an $n \times k$ matrix has quantum query lower bound of $\Omega(n\sqrt{k})$. It is worth investigating whether we can improve the lower bound for this problem or other hard instances.

- Can quantum algorithms provide speedup for finding second-order stationary points in finite-sum nonconvex optimization? This had been systematically investigated in the classical setting by NEON (Xu et al., 2018; Allen-Zhu & Li, 2018) and SPIDER (Fang et al., 2018). Since our quantum algorithm for finding critical points (Algorithm 3) was built upon SPIDER, it is worth investigating whether quantum speedup for finding second-order stationary points can be given.

- Can we apply our quantum algorithm for solving machine learning problems with speedup? Several quantum algorithms for relevant machine learning problems have been proposed, for instance support vector machine (Rebentrost et al., 2014) and regression (Wang, 2017; Liu & Zhang, 2017; Chen & de Wolf, 2023; Shao, 2023), but the quantum speedup from variance reduction in the finite sum is widely open.

2. Preliminaries

Notation. We use bold letters, i.e., $\mathbf{x}, \mathbf{y}$ to denote vectors and use $\|\cdot\|$ to denote the Euclidean norm. For a d-dimensional random variable X, we refer to the trace of the covariance matrix of X as its variance, denoted by $\mathrm{Var}[X]$. We define $[n] := \{1, \ldots, n\}$. By default, the logarithms are in base 2

To model a classical probability distribution p over $\mathbb{R}^d$ in the quantum setting, we can use the quantum state $\sum_{\mathbf{x}\in\mathbb{R}^d}\sqrt{p(\mathbf{x})}\,|\mathbf{x}\rangle$. If we measure this state, the measurement outcome is described by the probability density function p. When applicable, we use $|\mathrm{garbage}(\cdot)\rangle$ to represent possible garbage states[4] that emerge during the implementation of a quantum oracle.

[4] The garbage state serves as a quantum counterpart to classical garbage information that emerges during the preparation of the classical stochastic gradient oracle that cannot be erased or uncomputed. In this work, we adopt a broad model without making specific assumptions about the garbage state. For a comparable discussion on this conventional usage of garbage quantum states, refer to Gilyén & Li (2020); Sidford & Zhang (2023) for similar discussions of this standard use of garbage quantum states.

Quantum variance reduction. Mean estimation a well-studied problem in quantum computing (Hamoudi, 2021; Cornelissen et al., 2022; Kothari & O'Donnell, 2023), which collectively demonstrate a quadratic quantum speedup for mean estimation. However, the output of these quantum mean estimation algorithms may exhibit bias, posing a limitation when integrating them with optimization algorithms that assume unbiased inputs. This concern was addressed in Sidford & Zhang (2023) where they developed the *quantum variance reduction* algorithm that eliminates the bias leveraging the multilevel Monte Carlo (MLMC) technique. In particular, they proved the following result.

Lemma 1 (Theorem 4 of Sidford & Zhang (2023))**.** *For a d-dimensional random variable X with $\mathrm{Var}[X] \leq \sigma^2$ and some $\hat{\sigma} \geq 0$, suppose we are given access to a quantum sampling oracle*

$$O_X |0\rangle \to \sum_{\mathbf{x} \in \mathbb{R}^d} \sqrt{\Pr(X = \mathbf{x})} \otimes |\mathrm{garbage}(\mathbf{x})\rangle\,,$$

there exists a quantum algorithm $\texttt{QuantumVarianceReduction}(O_X, \hat{\sigma})$ that outputs an unbiased estimate $\hat{\mu}$ of $\mathbb{E}[X]$ satisfying $\mathbb{E}\|\hat{\mu} - \mathbb{E}[X]\|^2 \leq \hat{\sigma}^2$ using an expected $\tilde{O}(d^{1/2}\sigma/\hat{\sigma})$ queries to O_X.

3. Quantum Algorithms in Convex Settings

In this section, we present our quantum algorithms for solving all the four cases of Problem 1.

3.1. Strongly convex setting

In this subsection, we present our quantum algorithm for Case 1 of Problem 1. Our approach is based on the framework of `Katyusha` developed in Allen-Zhu (2017), which is an accelerated stochastic variance reduction algorithm. At the beginning of the algorithm, `Katyusha` first computes the exact gradient of $\mathbf{x}_0$ by querying $\nabla f_i(\mathbf{x}_0)$ for every $i \in [n]$. Then for further iterations $\mathbf{x}_t$ that are not very far away from $\mathbf{x}_{\mathrm{ref}} := \mathbf{x}_0$, `Katyusha` uses $\nabla f(\mathbf{x}_{\mathrm{ref}})$ as a reference to approximate $\nabla f(\mathbf{x}_t)$ by approximating $\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{\mathrm{ref}})$, which has a small norm when $\mathbf{x}_t$ is close to $\mathbf{x}_{\mathrm{ref}}$ given that each f_i is ℓ-smooth. Whenever the current iteration is too far away from the reference, `Katyusha` computes the exact gradient of this iteration and makes it the new reference.

Compared to their algorithm, our algorithm (Algorithm 1) replaces the variance reduction step of computing $\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{\mathrm{ref}})$ by the quantum variance reduction technique in Sidford & Zhang (2023), as shown in Algorithm 2.

Theorem 4. *Algorithm 1 solves Case 1 of Problem 1 using the following number of queries in expectation:*

$$\tilde{O}\big(n + \sqrt{d} + \sqrt{\ell/\mu}\big(n^{1/3}d^{1/3} + n^{-2/3}d^{5/6}\big)\big).$$

Algorithm 1: Q-Katyusha

```
Input: Function $F\colon \mathbb{R}^d \to \mathbb{R}$, precision $\epsilon$, smoothness
       $\ell$, strong convexity $\mu$
Parameters: $S = 5(1 + \ell^{1/2}(bm\mu)^{-1/2})\log(\Delta/\epsilon)$,
            $b = \lceil n^{2/3}d^{-1/3} \rceil$, $m = \lceil n^{2/3}d^{-1/3} \rceil$
Output: An $\epsilon$-optimal point of $f$
1  $\tau_2 \leftarrow \frac{1}{2b}$, $\tau_1 \leftarrow \tau_2 \cdot \min\left\{\sqrt{\frac{8bm\mu}{3L}}, 1\right\}$
2  $\mathbf{y}_0 = \mathbf{z}_0 = \tilde{\mathbf{x}}^0 \leftarrow \mathbf{0}$
3  for $s = 0, 1, \ldots, S-1$ do
4      $\gamma^s \leftarrow \nabla f(\tilde{\mathbf{x}}^s)$
5      for $j = 0, 1, \ldots, m-1$ do
6          $k \leftarrow (sm) + j$
7          $\mathbf{x}_{k+1} \leftarrow \tau_1 \mathbf{z}_k + \tau_2 \tilde{\mathbf{x}}^s + (1 - \tau_1 - \tau_2)\mathbf{y}_k$
8          $\hat{\mathbf{g}}_{k+1} \leftarrow \texttt{QVRG}(\mathbf{x}_{k+1}, \tilde{\mathbf{x}}_s, \ell\|\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_s\|/\sqrt{b})$
9          $\tilde{\nabla}_{k+1} \leftarrow \gamma^s + \hat{\mathbf{g}}_{k+1}$
10         $\mathbf{z}_{k+1} \leftarrow$
             $\arg\min_{\mathbf{z}} \left\{\frac{3\tau_1\ell}{2}\|\mathbf{z} - \mathbf{z}_k\|^2 + \langle \tilde{\nabla}_{k+1}, \mathbf{z}\rangle + \psi(\mathbf{z})\right\}$
11         $\mathbf{y}_{k+1} \leftarrow$
12         $\arg\min_{\mathbf{y}} \left\{\frac{3\ell}{2}\|\mathbf{y} - \mathbf{x}_{k+1}\|^2 + \langle \tilde{\nabla}_{k+1}, \mathbf{y}\rangle + \psi(\mathbf{y})\right\}$
13     $\tilde{\mathbf{x}}^{s+1} \leftarrow \frac{\sum_{j=0}^{m-1} \mathbf{y}_{sm+j+1}}{m}$
14 return $\mathbf{x}^{\mathrm{out}} \leftarrow \frac{\tau_2 m \tilde{\mathbf{x}}^S + (1 - \tau_1 - \tau_2)\mathbf{y}_{Sm}}{\tau_2 m + (1 - \tau_1 - \tau_2)}$
```

Algorithm 2: Quantum variance-reduced gradient (QVRG)

```
Input: Function $F\colon \mathbb{R}^d \to \mathbb{R}$, $\mathbf{x}, \mathbf{x}_{\mathrm{ref}} \in \mathbb{R}^d$, accuracy
       $\hat{\sigma}$
Denote $\mathbf{g}_i := \nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{x}_{\mathrm{ref}})$ for all $i \in [n]$.
  Implement the oracle $O_{\mathbf{g}}|0\rangle \to$
  $\frac{1}{\sqrt{n}} \sum_i |\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{x}_{\mathrm{ref}})\rangle \otimes |\mathrm{garbage}(i)\rangle$.
$\hat{\mathbf{g}} \leftarrow \texttt{QuantumVarianceReduction}(O_{\mathbf{g}}, \hat{\sigma})$
return $\hat{\mathbf{g}}$
```

Prior to proving Theorem 4, we first establish an upper bound on the query complexity of running the subroutine `QVRG` (Algorithm 2) in Line 8.

Lemma 2. *If every f_i is ℓ-smooth, Algorithm 2 outputs an unbiased estimate $\hat{\mathbf{g}}$ of $\bar{\mathbf{g}} := \frac{1}{n}\sum_{i=1}^{n}(\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{x}_{\mathrm{ref}}))$ satisfying $\mathbb{E}\|\hat{\mathbf{g}} - \bar{\mathbf{g}}\| \leq \hat{\sigma}^2$ using an expected $\tilde{O}(d^{1/2}\ell\|\mathbf{x} - \mathbf{x}_{\mathrm{ref}}\|/\hat{\sigma})$ queries to O_F.*

The proof of Lemma 2 is deferred to Appendix A.

The following result from Allen-Zhu (2017) bounds the rate at which Algorithm 1 decreases the function error of F. Note that correctness of this result relies solely on the fact that, at Line 8, the variance of the unbiased estimate $\hat{\mathbf{g}}_{k+1}$ of $\frac{1}{n}\sum_i(\nabla f_i(\mathbf{x}_{k+1}) - \nabla f_i(\tilde{\mathbf{x}}_s))$ is upper bounded by $\ell\|\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_s\|/\sqrt{b}$, regardless of its implementation.

Lemma 3 (Theorem 5.2, Allen-Zhu (2017))**.** *In Case 1 of*

Problem 1, for any $b, m \in [n]$, the output $\mathbf{x}^{\text{out}}$ of Algorithm 1 satisfies

$$\mathbb{E}[F(\mathbf{x}^{\text{out}})] - F^* \leq \begin{cases} O\left(\left(1+\sqrt{b\mu/(6\ell m)}\right)^{-Sm} \cdot \Delta\right), & \\ \qquad \text{if } \frac{m\mu b}{\ell} \leq \frac{3}{8} \text{ and } b \leq m, & \\ O\left(\left(1+\sqrt{\mu/(6\ell)}\right)^{-Sm} \cdot \Delta\right), & \\ \qquad \text{if } \frac{m^2\mu}{\ell} \leq \frac{3}{8} \text{ and } b > m, & \\ O\left(1.25^{-S} \cdot \Delta\right), & \text{otherwise.} \end{cases} \tag{4}$$

Equipped with these results, we can prove Theorem 4 now.

Proof of Theorem 4. Given the parameter choices of Algorithm 1, its output $\mathbf{x}^{\text{out}}$ satisfies either the first case or the third case of (4) in Lemma 3. Hence, the output $\mathbf{x}^{\text{out}}$ of Algorithm 1 satisfies

$$\mathbb{E}[F(\mathbf{x}^{\text{out}}) - F^*] \leq \max\{O\Big(\left(1+\sqrt{b\mu/(6\ell m)}\right)^{-Sm} \cdot \Delta\Big),$$
$$O\left(1.25^{-S} \cdot \Delta\right)\} = O(\epsilon)$$

The query complexity of Algorithm 1 is a combination of two components: the complete gradient computation step in Line 4 and the QVRG step in Line 8. Throughout Algorithm 1, there are in total S full gradient computation steps and each step takes $O(n)$ queries by using O_F just as a classical finite sum oracle, i.e., we query O_F without employing quantum superposition. As for the second part, as per Lemma 2, each call to QVRG takes an expected

$$\tilde{O}\left(d^{1/2}\ell\|\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_s\|/\hat{\sigma}\right) = \tilde{O}(\sqrt{bd})$$

queries to the quantum finite-sum oracle. Consequently, to find an expected ϵ-optimal point of F, the overall query complexity equals

$$\begin{aligned} &S \cdot O(n) + Sm \cdot \tilde{O}(\sqrt{bd}) \\ &= \tilde{O}\big(n + \sqrt{d} + \sqrt{\ell/\mu}\big(n^{1/3}d^{1/3} + n^{-2/3}d^{5/6}\big)\big). \end{aligned}$$

□

3.2. Corollaries for non-smooth or non-strongly convex settings

The algorithm for Problem 1 in the non-strongly convex setting can be obtained via applying a black-box reduction introduced in Allen-Zhu & Hazan (2016a).

Definition 2 (HOOD property, Allen-Zhu & Hazan (2016a))**.** *We say an algorithm solving Case 1 of Problem 1 satisfies the homogenous objective decrease (HOOD) property with query complexity $\mathcal{Q}(\ell, \mu)$ if for every starting point $\mathbf{x}_0$, it produces output $\mathbf{x}^{\text{out}}$ such that*

$$\mathbb{E}[F(\mathbf{x}^{\text{out}})] - F^* \leq (F(\mathbf{x}_0) - F^*)/4$$

using an expected $\mathcal{Q}(\ell, \mu)$ queries.

Setting $\epsilon = (F(\mathbf{x}_0) - F^*)/4$ in Theorem 4 gives:

Corollary 1. *Algorithm 1 satisfies the HOOD property with $\mathcal{Q}(\ell, \mu) = \tilde{O}\big(n + \sqrt{d} + \sqrt{\ell/\mu}\big(n^{1/3}d^{1/3} + n^{-2/3}d^{5/6}\big)$.*

Lemma 4 (Theorem 3.4, Allen-Zhu (2017))**.** *Given a quantum algorithm $\mathcal{A}$ that satisfies the HOOD property with query complexity $\mathcal{Q}(\ell, \mu)$, there exist three quantum algorithms that separately solves*

- *Case 2 of Problem 1 using $\sum_{s=0}^{S-1} \mathcal{Q}\big(\ell, \frac{\tilde{\mu}}{2^s}\big)$ queries, where $\tilde{\mu} = \frac{F(\mathbf{0})-F^*}{\|\mathbf{x}^*\|^2}$ and $S = \log\frac{F(\mathbf{0})-F^*}{\epsilon}$,*
- *Case 3 of Problem 1 using $\sum_{s=0}^{S-1} \mathcal{Q}\big(\frac{2^s}{\lambda}, \mu\big)$ queries, where $\lambda = \frac{F(\mathbf{0})-F^*}{L^2}$ and $S = \log\frac{F(\mathbf{0})-F^*}{\epsilon}$, and*
- *Case 4 of Problem 1 using $\sum_{s=0}^{S-1} \mathcal{Q}\big(\frac{2^s}{\lambda}, \frac{\tilde{\mu}}{2^s}\big)$ queries, where $\lambda = \frac{F(\mathbf{0})-F^*}{L^2}$, $\tilde{\mu} = \frac{F(\mathbf{0})-F^*}{\|\mathbf{x}^*\|^2}$, and $S = \log\frac{F(\mathbf{0})-F^*}{\epsilon}$.*

Combining Corollary 1 and Lemma 4, we can have the following corollary.

Corollary 2. *There exists 3 quantum algorithm that solves Case 2, 3, 4 of Problem 1, respectively, with the following query complexities:*

Case 2: $\tilde{O}\big(n + \sqrt{d} + R\sqrt{\ell/\epsilon}\big(n^{1/3}d^{1/3} + n^{-2/3}d^{5/6}\big)\big)$;

Case 3: $\tilde{O}\big(n + \sqrt{d} + L\big(n^{1/3}d^{1/3} + n^{-2/3}d^{5/6}\big)/\sqrt{\lambda\mu}\big)$;

Case 4: $\tilde{O}\big(n + \sqrt{d} + LR\big(n^{1/3}d^{1/3} + n^{-2/3}d^{5/6}\big)/\epsilon\big)$.

The proof of Corollary 2 is deferred to Appendix A.

4. Quantum Algorithm in the Nonconvex Setting

In this section, we present our quantum algorithm that solves Problem 2. Our approach builds upon the SPIDER algorithm (Fang et al., 2018), which is a variance reduction technique that can estimate the gradient of an iterate with lower cost by utilizing the smoothness of each f_i and reuse the gradient estimations of previous iterations. Our algorithm is a specialization of the SPIDER algorithm, where we replace the classical variance reduction step by QVRG introduced in Section 3.

Compared to Algorithm 7 of Sidford & Zhang (2023), which is another quantum algorithm based on SPIDER (Fang et al., 2018), our Algorithm 3 works in the finite-sum setting which enables us to compute the full gradient in Line 4. Moreover, we carefully choose the parameters of the algorithm, taking into account of the difference in query complexity between QVRG and the classical variance reduction step, with a corresponding convergence analysis showing that the output of Algorithm 3 still converges to an ϵ-critical point despite the changes in the parameters. In particular, we prove:

Algorithm 3: Finite-Sum-Q-SPIDER

```
Input: Function $f\colon \mathbb{R}^d \to \mathbb{R}$, precision $\epsilon$, smoothness $\ell$
Parameters: $q = \lceil n^{2/3} d^{-1/3} \rceil$, $\hat{\epsilon} = \frac{\epsilon}{5}$, total iteration
            budget $\mathcal{T} = \lceil \frac{4\ell\Delta}{\epsilon^2} \rceil$
Output: An $\epsilon$-critical point of $f$
Set $\mathbf{x}_0 \leftarrow \mathbf{0}$
for $t = 0, 1, 2, \ldots, \mathcal{T}$ do
    if $\mod (t, q) = 0$ then
        $\mathbf{v}_t \leftarrow \nabla f(\mathbf{x}_t)$
    else
        $\mathbf{g}_t \leftarrow \texttt{QVRG}(\mathbf{x}_t, \mathbf{x}_{t-1}, \hat{\epsilon}/\sqrt{2q})$
        $\mathbf{v}_t \leftarrow \mathbf{v}_{t-1} + \mathbf{g}_t$
    if $\|\mathbf{v}_t\| \leq \hat{\epsilon}$ then return $\mathbf{x}_t$;
    else $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \frac{\hat{\epsilon}}{2\ell} \cdot \frac{\mathbf{v}_t}{\|\mathbf{v}_t\|}$
Uniformly randomly choose $\mathbf{x}^{\text{out}}$ from $\mathbf{x}_0, \ldots, \mathbf{x}_{\mathcal{T}-1}$
return $\mathbf{x}^{\text{out}}$
```

Theorem 5. *Algorithm 3 solves Problem 2 using the following number of queries in expectation:*

$$\tilde{O}\left(n + \ell\Delta\left(d^{1/3}n^{1/3} + \sqrt{d}\right)/\epsilon^2\right).$$

The proof of Theorem 5 is deferred to Appendix B.

5. Quantum Lower Bounds

In this section, we establish our quantum lower bounds for Problem 1. Our approach leverages the framework of Woodworth & Srebro (2016), which gives a "hard instance" based on randomly selected orthogonal spaces. We show that for the hard function presented in this paper, a quantum algorithm cannot find an ϵ-*optimal* solution until it has made enough queries to the quantum oracles.

Regarding the transition from the classical hard function to the quantum complexity lower bound, our approach is based on the adversary lower bound introduced by Zhang (2005) that applies to non-boolean functions. Initially, we reduce the task of finding the ϵ-*optimal* point for the hard function to "finding all elements on several chains". Subsequently, we apply the adversary method to obtain the corresponding quantum complexity lower bound.

5.1. Smooth and strongly convex setting

We first consider Case 1 of Problem 1. Without loss of generality, we assume that $\ell = 1$ and n is even. If n is odd, we can simply take one of the sub-functions to 0, and the query complexity is reduced by a factor proportional to $\frac{n-1}{n}$. Then we define the following hard instance.

Definition 3 (Hard instance for Case 1 of Problem 1). *For constants k, C, and ζ to be decided upon later. Let $\tilde{\mu} := n\cdot\mu$. For $i = 1, 2, \ldots, \lfloor n/2 \rfloor$, we define*

$$\begin{aligned} f_{i,1}(\mathbf{x}) &:= \frac{1-\tilde{\mu}}{16}\left(\langle \mathbf{x}, \mathbf{v}_{i,0}\rangle^2 - 2C\langle \mathbf{x}, \mathbf{v}_{i,0}\rangle \right. \\ &\quad \left. + \sum_{r\ \text{even}}^{k} \phi_c\left(\langle \mathbf{x}, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x}, \mathbf{v}_{i,r}\rangle\right)\right); \\ f_{i,2}(\mathbf{x}) &:= \frac{1-\tilde{\mu}}{16}\left(\zeta\phi_c(\langle \mathbf{x}, \mathbf{v}_{i,k}\rangle) \right. \\ &\quad \left. + \sum_{r\ \text{odd}}^{k} \phi_c\left(\langle \mathbf{x}, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x}, \mathbf{v}_{i,r}\rangle\right)\right), \end{aligned}$$

where $\{\mathbf{v}_{i,r}\}$ are orthonormal vectors randomly selected from $\mathbb{R}^d$, and ϕ_c is a 4-smooth helper function defined as follows:

$$\phi_c(z) := \begin{cases} 0 & |z| \leq c; \\ 2(|z| - c)^2 & c < |z| \leq 2c; \\ z^2 - 2c^2 & |z| > 2c \end{cases}. \tag{5}$$

Then the hard instance is defined as follows.

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{\frac{n}{2}}\sum_{j=1}^{2} f_{i,j}(\mathbf{x}), \quad \psi(\mathbf{x}) = \frac{\mu}{2}\|\mathbf{x}\|^2.$$

This hard function has the following property: for $f_{i,1}$ and $f_{i,2}$, when we only know the values of $\mathbf{v}_{i,j}$ for $j \leq t$, then after a new query, we can only find a vector $\mathbf{x}$ such that $|\langle \mathbf{x}, \mathbf{v}_{i,t+1}\rangle| > \frac{c}{2}$. However, with large probability, $\mathbf{x}$ satisfies $|\langle \mathbf{x}, \mathbf{v}_{i,j}\rangle| \leq \frac{c}{2}$ for all $j \geq t+1$. This property forms the basis for proving the lower bound, since the ϵ-optimal point is related to every vector $\mathbf{v}_{i,j}$, as shown in the following lemma.

Lemma 5. *Denote $Q = \frac{1}{2}\left(\frac{1}{n\mu} - 1\right) + 1$, $q = \frac{\sqrt{Q}-1}{\sqrt{Q}+1}$. For $\zeta = 1 - q$, $k = \lfloor \frac{\sqrt{Q}-1}{4} \log \frac{\Delta}{n\epsilon(\sqrt{Q}-1)^2)} \rfloor - 1$, $C = \sqrt{\frac{\Delta}{\mu}} \frac{4}{\sqrt{Q}-1}$, $c = \min\left\{\frac{1}{\sqrt{N}}, \sqrt{\frac{8n\epsilon}{(1-\tilde{\mu})(k+1)}}, Cq^{k+1}\right\}$. For any $\epsilon \leq \frac{4}{3}\mu\Delta$ and any $\mathbf{x} \in \mathbb{R}^d$, if there exists a vector $\mathbf{v}_{ij}$ ($i \in \{1, 2, \ldots, \lfloor n/2 \rfloor\}$, $j \in \{1, 2, \ldots, k\}$) with $|\langle \mathbf{x}, \mathbf{v}_{ij}\rangle| < \frac{c}{2}$, then $\mathbf{x}$ cannot be an ϵ-optimal point of the hard function $F(\mathbf{x})$ defined in Definition 3.*

Lemma 5 shows that the ϵ-optimal point must have a relatively large inner product with **all vectors** $\mathbf{v}_{ij}$. Consequently, any algorithm seeking an ϵ-optimal point must find a vector with a significant inner product with all $\mathbf{v}_{ij}$. Then, we can reduce the problem of finding the ϵ-optimal point to the problem of finding all elements of several sub-functions, where each oracle query can only reveal the next element of one sub-function with high probability. We can then use the adversary method to establish a lower bound on the query complexity for the quantum query problem described above. Note that Lemma 5 is stronger than that in Woodworth & Srebro (2016), where it is only proved that for a specific t, at least half of the sub-functions f_i satisfy that the ϵ-optimal point must have a significant inner product with $\mathbf{v}_{ij_i}$ for some $j_i > t$.

Remark 1. *If we directly use the property proved in Woodworth & Srebro (2016), to prove that the output x of an algorithm is far away from being ϵ-optimal, we must ensure that x satisfies the following property: for more than half of the sub-functions with index i, $|\langle \mathbf{x},\, \mathbf{v}_{i,j_i}\rangle| < \frac{c}{2}$ holds true for all $j_i > t$. To violate this property, it is sufficient to find information about only one vector that has a large inner product with x for each pair of sub-functions. This case can at most correspond to an $n \times 1$ quantum query problem and thus the quantum adversary method would only yield a trivial lower bound of n. However, under the property given in Lemma 5, we impose a higher requirement for an efficient algorithm that finds an ϵ-optimal point: it must find information about all vectors in each pair of sub-functions. This naturally reduces to an $n \times k$ quantum query problem, which is evidently more difficult than the former one.*

Now we have transformed the lower bound of an hard instance in the optimization problem into the quantum query lower bound of the following query problem:

Problem 3 (Multi Chain Problem)**.** *For input size n and k, we are given oracle access to n k-bit strings $x_1, x_2, \cdots x_n$. Specifically, for a set of strings $x = [x_1, x_2, \cdots x_n]$, a query is specified by a set (i, j, s), where s is a j-bit string, and returns 1 if and only if s is exactly a prefix of length j for x_i. The corresponding quantum oracle which allows us to query different strings at the same time is defined as follows:*

$$O_x \left|i\right\rangle \otimes \left|t\right\rangle \otimes \left|x'_{i1}x'_{i2}\cdots x'_{it}\right\rangle \otimes \left|0\right\rangle \to \begin{cases} \left|i\right\rangle \otimes \left|j\right\rangle \otimes \left|x'_{i1}x'_{i2}\cdots x'_{it}\right\rangle \otimes \left|0\right\rangle, \\ \quad \textit{if } x_{ij} = x'_{ij} \textit{ for each } j \in \{1,\cdots t\}; \\ \left|i\right\rangle \otimes \left|j\right\rangle \otimes \left|x'_{i1}x'_{i2}\cdots x'_{it}\right\rangle \otimes \left|1\right\rangle, \\ \quad \textit{if there exists } j \in \{1,\cdots t\} \textit{ such that } x_{ij} \neq x'_{ij}. \end{cases}$$

The goal is to output the matrix x.

The query lower bound for this problem is proven in Appendix C.1. The above analysis can be summarized into the following conclusion:

Corollary 3. *For any $\ell, \mu > 0$ such that $\frac{\ell}{\mu} \geq 100n$, for any $\epsilon < \frac{4}{3}\frac{\mu\Delta}{\ell}$, $d = \tilde{\Omega}(\frac{\Delta\ell^2}{\mu^2 n\epsilon})$, any quantum algorithm that solves Case 1 of Problem 1 with success probability at least $2/3$ must make at least the following number of queries in the worst case:*

$$\Omega\left(n + n^{\frac{3}{4}}\left(\frac{\ell}{\mu}\right)^{\frac{1}{4}}\log^{\frac{1}{2}}\left(\frac{\Delta\mu}{\epsilon\ell}\right)\right).$$

The detailed proof is deferred to Appendix C.3.

5.2. Smooth and non-strongly convex setting

In the non-strongly convex setting, our hard instance is similar to the strongly convex case, except for a regularization parameter. Without loss of generality, we assume that $\ell = R = 1$ and n is even, and our hard instance is constructed as follows.

Definition 4 (Hard instance for Case 2 of Problem 1)**.** *For $i = 1, \ldots, \lfloor n/2 \rfloor$, take values C and k to be fixed later, define*

$$\begin{aligned} f_{i,1}(\mathbf{x}) &:= \frac{1}{16}(\langle \mathbf{x},\, \mathbf{v}_{i,0}\rangle^2 - 2C\,\langle \mathbf{x},\, \mathbf{v}_{i,0}\rangle \\ &\quad + \sum_{r \text{ even}}^{k} \phi_c\left(\langle \mathbf{x},\, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x},\, \mathbf{v}_{i,r}\rangle\right)); \\ f_{i,2}(\mathbf{x}) &:= \frac{1}{16}(\phi_c(\langle \mathbf{x},\, \mathbf{v}_{i,k}\rangle) \\ &\quad + \sum_{r \text{ odd}}^{k} \phi_c\left(\langle \mathbf{x},\, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x},\, \mathbf{v}_{i,r}\rangle\right)) \end{aligned}$$

with orthonormal vectors $\mathbf{v}_{ij}$ chosen randomly on the unit sphere in $\mathbb{R}_d$, and ϕ_c is the same helper function as in Definition 3. Then the two component of the hard instance is defined as follows:

$$f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{\frac{n}{2}}\sum_{j=1}^{2} f_{i,j}(\mathbf{x}), \quad \psi(\mathbf{x}) = 0.$$

Lemma 6. *Let $k = \lfloor \frac{1}{16\sqrt{\epsilon n}} \rfloor - 1$, $C = \sqrt{\frac{6}{nk}}$, and $c = \min\{\frac{1}{\sqrt{N}}, (2-\sqrt{3})C, 8\sqrt{\frac{\epsilon}{k}}\}$. For any $\epsilon < \frac{1}{4096n}$ and any $\mathbf{x} \in \mathbb{R}^d$, if for at least $\frac{n}{4}$ i's that there exists j_i which holds that $j_i \leq t := \lfloor k/2 \rfloor$ and $|\langle \mathbf{x}, \mathbf{v}_{i,j_i}\rangle| < \frac{c}{2}$, then $\mathbf{x}$ cannot be an ϵ-optimal point of the hard function F defined in Definition 4.*

Equipped with Lemma 6, we prove the following result.

Corollary 4. *For any $\epsilon < \frac{\ell R^2}{4096n}$, $d = \tilde{\Omega}\left(\sqrt{\frac{n}{\epsilon}} + \frac{1}{\epsilon^2}\right)$, any quantum algorithm that solves Case 2 of Problem 1 with success probability at least $2/3$ must make at least the following number of queries in the worst case:*

$$\Omega\left(n + \frac{n^{\frac{3}{4}}}{\log n}\left(\frac{\ell}{\epsilon}\right)^{\frac{1}{4}} R^{\frac{1}{2}}\right).$$

The detailed proof is deferred to Appendix C.4.

5.3. Lipschitz and non-strongly convex setting

In the Lipschitz and non-strongly convex setting, we also assume for simplicity n is even. For the required Lipschitz property, we use a new helper function $\chi_c(z)$ based on the absolute value, which is defined as follows:

$$\chi_c(z) = \max\{0, |z| - c\}.$$

Notice that this helper function is 1-Lipschitz and it hides the information about $\mathbf{x}$ if the norm of $\mathbf{x}$ is relatively small. Then, we can define $\frac{n}{2}$ pairs of sub-functions as our hard instance:

Definition 5 (Hard instance for Case 4 of Problem 1)**.** *For $i = 1, \dots, \lfloor n/2 \rfloor$, take values b, c and k to be fixed later, let*

$$\begin{aligned} f_{i,1}(\mathbf{x}) &:= \frac{1}{\sqrt{2}} |b - \langle \mathbf{x}, \mathbf{v}_{i,0} \rangle| \\ &\quad + \frac{1}{2\sqrt{k}} \sum_{r \text{ even}}^{k} \chi_c \left(\langle \mathbf{x},\, \mathbf{v}_{i,r-1} \rangle - \langle \mathbf{x},\, \mathbf{v}_{i,r} \rangle \right); \\ f_{i,2}(\mathbf{x}) &:= \frac{1}{2\sqrt{k}} \sum_{r \text{ odd}}^{k} \chi_c \left(\langle \mathbf{x},\, \mathbf{v}_{i,r-1} \rangle - \langle \mathbf{x},\, \mathbf{v}_{i,r} \rangle \right). \end{aligned}$$

with orthonormal vectors $\mathbf{v}_{ij}$ chosen randomly on the unit sphere in $\mathbb{R}_d$. The two component of the hard instance is defined as follows

$$f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{\frac{n}{2}} \sum_{j=1}^{2} f_{i,j}(\mathbf{x}), \quad \psi(\mathbf{x}) = 0.$$

Similar to the smooth case, the ϵ-optimal point of the hard instance can only be found after querying a sufficient number of different vectors $\mathbf{v}_{i,j}$.

Lemma 7. *Let $k = \lfloor \frac{1}{10\epsilon\sqrt{n}} \rfloor$, $c = \min\{\frac{1}{\sqrt{N}}, \frac{\epsilon}{\sqrt{k}}\}$ and $b = \sqrt{\frac{2}{n(k+1)}}$. For any $\epsilon < \frac{3}{10\sqrt{n}}$ and any $\mathbf{x} \in \mathbb{R}^d$, if for at least $\frac{n}{4}$ i's that there exists j_i such that $|\langle \mathbf{x}, \mathbf{v}_{i,j_i} \rangle| < \frac{c}{2}$, then $\mathbf{x}$ cannot be an ϵ-optimal point of the hard function F defined in Definition 5.*

Corollary 5. *For any $\epsilon < \frac{3LR}{10\sqrt{n}}$, $d = \tilde{\Omega}\left(\frac{1}{\epsilon^3\sqrt{n}}\right)$, any quantum algorithm that solves Case 4 of Problem 1 with success probability at least $2/3$ must make at least the following number of queries in the worst case:*

$$\Omega\left(n + n^{\frac{3}{4}} \left(\frac{LR}{\epsilon}\right)^{\frac{1}{2}} \frac{1}{\log n}\right).$$

The detailed proof is deferred to Appendix C.5.

5.4. Lipschitz and strongly convex setting

In the Lipschitz and strongly convex setting, we can construct a hard instance with corresponding quantum lower bound by adding a regularizer to Definition 5. Thus, we can directly use the result of Corollary 5.Technical details can be found in the appendix.

Corollary 6. *For any $\epsilon < \frac{9L^2}{200n\mu}$ and $d = \tilde{\Omega}\left(\frac{1}{\sqrt{\epsilon^3 n}}\right)$, any quantum algorithm that solves Case 3 of Problem 1 with success probability at least $2/3$ must make at least the following number of queries in the worst case:*

$$\Omega\left(n + n^{\frac{3}{4}} \left(\frac{1}{\epsilon\mu}\right)^{\frac{1}{4}} L^{\frac{1}{2}} \frac{1}{\log n}\right).$$

The detailed proof is deferred to Appendix C.6.

Impact Statement

This paper presents work whose goal is to advance theories of machine learning. Since this work is purely theoretical, we do not have specific societal consequences to highlight here.

Acknowledgements

Y. Zhang and T. Li were supported by the National Natural Science Foundation of China (No. 62372006 and No. 92365117), and The Fundamental Research Funds for the Central Universities, Peking University. C. Fang was supported by the NSF National Natural Science Foundation of China (No. 62376008).

References

Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017. arXiv:1603.05953

Allen-Zhu, Z. Katyusha X: Practical momentum method for stochastic sum-of-nonconvex optimization, 2018. arXiv:1802.03866

Allen-Zhu, Z. and Hazan, E. Optimal black-box reductions between optimization objectives. *Advances in Neural Information Processing Systems*, 29, 2016a. arXiv:1603.05642

Allen-Zhu, Z. and Hazan, E. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pp. 699–707. PMLR, 2016b. arXiv:1603.05643

Allen-Zhu, Z. and Li, Y. Neon2: Finding local minima via first-order oracles. In *Advances in Neural Information Processing Systems*, pp. 3716–3726, 2018. arXiv:1711.06673

Allen-Zhu, Z. and Yuan, Y. Improved SVRG for non-strongly-convex or sum-of-non-convex objectives. In *International Conference on Machine Learning*, pp. 1080–1089. PMLR, 2016. arXiv:1506.01972

Ambainis, A. Quantum lower bounds by quantum arguments. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pp. 636–643, 2000. arXiv:quant-ph/0002066

Ambainis, A. Polynomial degree vs. quantum query complexity. *Journal of Computer and System Sciences*, 72(2): 220–238, 2006. arXiv:quant-ph/0305028

Ambainis, A. and Montanaro, A. Quantum algorithms for search with wildcards and combinatorial group testing. *Quantum Information & Computation*, 14(5&6):439–453, 2014. arXiv:1210.1148

Apeldoorn, J. v., Gilyén, A., Gribling, S., and de Wolf, R. Convex optimization using quantum oracles. *Quantum*, 4:220, 2020. arXiv:1809.00643

Asi, H., Carmon, Y., Jambulapati, A., Jin, Y., and Sidford, A. Stochastic bias-reduced gradient methods. *Advances in Neural Information Processing Systems*, 34:10810–10822, 2021. arXiv:2106.09481

Blanchet, J. H. and Glynn, P. W. Unbiased Monte Carlo for optimization and functions of expectations via multi-level randomization. In *2015 Winter Simulation Conference (WSC)*, pp. 3656–3667. IEEE, 2015.

Brandão, F. G. and Svore, K. Quantum speed-ups for semidefinite programming. In *Proceedings of the 58th Annual Symposium on Foundations of Computer Science*, pp. 415–426, 2017. arXiv:1609.05537

Brandão, F. G., 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. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 27:1–27:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. arXiv:1710.02581

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

Casares, P. A. M. and Martin-Delgado, M. A. A quantum interior-point predictor–corrector algorithm for linear programming. *Journal of physics A: Mathematical and Theoretical*, 53(44):445305, 2020. arXiv:1902.06749

Chakrabarti, S., Childs, A. M., Li, T., and Wu, X. Quantum algorithms and lower bounds for convex optimization. *Quantum*, 4:221, 2020. arXiv:1809.01731

Chen, Y. and de Wolf, R. Quantum algorithms and lower bounds for linear regression with norm constraints. In *50th International Colloquium on Automata, Languages, and Programming*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 38:1–38:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. arXiv:2110.13086

Chen, Z., Lu, Y., Wang, H., Liu, Y., and Li, T. Quantum Langevin dynamics for optimization, 2023. arXiv:2311.15587

Childs, A. M., Leng, J., Li, T., Liu, J.-P., and Zhang, C. Quantum simulation of real-space dynamics. *Quantum*, 6:680, 2022. arXiv:2203.17006

Cleve, R., Iwama, K., Le Gall, F., Nishimura, H., Tani, S., Teruyama, J., and Yamashita, S. Reconstructing strings from substrings with quantum queries. In *Scandinavian Workshop on Algorithm Theory*, pp. 388–397. Springer, 2012.

Cornelissen, A. and Hamoudi, Y. A sublinear-time quantum algorithm for approximating partition functions. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1245–1264. SIAM, 2023. arXiv:2207.08643

Cornelissen, A., Hamoudi, Y., and Jerbi, S. Near-optimal quantum algorithms for multivariate mean estimation. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 33–43, 2022. arXiv:2111.09787

Dagum, P., Karp, R., Luby, M., and Ross, S. An optimal algorithm for Monte Carlo estimation. *SIAM Journal on Computing*, 29(5):1484–1496, 2000.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in Neural Information Processing Systems*, 27, 2014. arXiv:1407.0202

Fang, C., Li, C. J., Lin, Z., and Zhang, T. SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018. arXiv:1807.01695

Garg, A., Kothari, R., Netrapalli, P., and Sherif, S. No quantum speedup over gradient descent for non-smooth convex optimization. In *Proceedings of the 12th Innovations in Theoretical Computer Science*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021. arXiv:2010.01801

Garg, A., Kothari, R., Netrapalli, P., and Sherif, S. Near-optimal lower bounds for convex optimization for all orders of smoothness. *Advances in Neural Information Processing Systems*, 34:29874–29884, 2021. arXiv:2112.01118

Gilyén, A. and Li, T. Distributional property testing in a quantum world. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. arXiv:1902.00814

Gong, W., Zhang, C., and Li, T. Robustness of quantum algorithms for nonconvex optimization, 2022. arXiv:2212.02548

Hamoudi, Y. Quantum sub-Gaussian mean estimator. In *29th Annual European Symposium on Algorithms (ESA 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021. arXiv:2108.12172

Hamoudi, Y. and Magniez, F. Quantum Chebyshev's inequality and applications. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, 2019. arXiv:1807.06456

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems*, 26, 2013.

Kerenidis, I. and Prakash, A. A quantum interior point method for LPs and SDPs. *ACM Transactions on Quantum Computing*, 1(1):1–32, 2020. arXiv:1808.09266

Kothari, R. and O'Donnell, R. Mean estimation when you have the source code; or, quantum Monte Carlo methods. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1186–1215. SIAM, 2023. arXiv:2208.07544

Leng, J., Hickman, E., Li, J., and Wu, X. Quantum Hamiltonian descent, 2023. arXiv:2303.01471

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. arXiv:2209.12897

Lin, H., Mairal, J., and Harchaoui, Z. Catalyst acceleration for first-order convex optimization: from theory to practice. *Journal of Machine Learning Research*, 18(1): 7854–7907, 2018. arXiv:1506.02186

Liu, Y. and Zhang, S. Fast quantum algorithms for least squares regression and statistic leverage scores. *Theoretical Computer Science*, 657:38–47, 2017.

Liu, Y., Su, W. J., and Li, T. On quantum speedups for nonconvex optimization via quantum tunneling walks. *Quantum*, 7:1030, 2023. arXiv:2209.14501

Montanaro, A. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A*, 471(2181):20150301, 2015. arXiv:1504.06987

Nielsen, M. A. and Chuang, I. L. *Quantum computation and quantum information*. Cambridge University Press, 2000.

Ozgul, G., Li, X., Mahdavi, M., and Wang, C. Stochastic quantum sampling for non-logconcave distributions and estimating partition functions, 2023. arXiv:2310.11445

Rebentrost, P., Mohseni, M., and Lloyd, S. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(13):130503, 2014. arXiv:1307.0471

Reddi, S. J., Hefny, A., Sra, S., Poczos, B., and Smola, A. Stochastic variance reduction for nonconvex optimization. In *International Conference on Machine Learning*, pp. 314–323. PMLR, 2016. arXiv:1603.06160

Roux, N., Schmidt, M., and Bach, F. A stochastic gradient method with an exponential convergence rate for finite training sets. *Advances in Neural Information Processing Systems*, 25, 2012. arXiv:1202.6258

Shalev-Shwartz, S. SDCA without duality, regularization, and individual convexity. In *International Conference on Machine Learning*, pp. 747–754. PMLR, 2016. arXiv:1602.01582

Shalev-Shwartz, S. and Zhang, T. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(1), 2013. arXiv:1209.1873

Shao, C. An improved quantum algorithm for low-rank rigid linear regressions with vector solution outputs, 2023. arXiv:2301.06107

Sidford, A. and Zhang, C. Quantum speedups for stochastic optimization. *Advances in Neural Information Processing Systems*, 37, 2023. arXiv:2308.01582

van Apeldoorn, J. and Gilyén, A. Improvements in quantum SDP-solving with applications. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 99:1–99:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. arXiv:1804.05058

van Apeldoorn, J., Gilyén, A., Gribling, S., and de Wolf, R. Quantum SDP-solvers: Better upper and lower bounds. In *58th Annual Symposium on Foundations of Computer Science*. IEEE, 2017. arXiv:1705.01843

Wang, G. Quantum algorithm for linear regression. *Physical review A*, 96(1):012335, 2017. arXiv:1402.0660

Woodworth, B. E. and Srebro, N. Tight complexity bounds for optimizing composite objectives. *Advances in neural information processing systems*, 29, 2016.

Xiao, L. and Zhang, T. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014. arXiv:1403.4699

Xu, Y., Jin, R., and Yang, T. First-order stochastic algorithms for escaping from saddle points in almost linear time. In *Advances in Neural Information Processing Systems*, pp. 5530–5540, 2018. arXiv:1711.01944

Zhang, C. and Li, T. Quantum lower bounds for finding stationary points of nonconvex functions, 2022. arXiv:2212.03906

Zhang, C. and Li, T. Quantum lower bounds for finding stationary points of nonconvex functions. In *International Conference on Machine Learning*, pp. 41268–41299. PMLR, 2023. arXiv:2212.03906

Zhang, C., Leng, J., and Li, T. Quantum algorithms for escaping from saddle points. *Quantum*, 5:529, 2021. arXiv:2007.10253

Zhang, L., Mahdavi, M., and Jin, R. Linear convergence with condition number independent access of full gradients. *Advances in Neural Information Processing Systems*, 26, 2013.

Zhang, S. On the power of ambainis lower bounds. *Theoretical Computer Science*, 339(2-3):241–256, 2005.

Zhang, T. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the 21st International Conference on Machine Learning*, pp. 116, 2004.

A. Proofs Details for Quantum Algorithms in Convex Settings

We give proof details for claims in Section 3 here.

Proof of Lemma 2. First observe that one query to $O_{\mathbf{g}}$ defined in Line 1 in Algorithm 2 can be implemented by applying O_F twice to the state

$$\frac{1}{\sqrt{n}}\sum_i |i\rangle\otimes|\mathbf{x}\rangle\otimes|\mathbf{x}_{\mathrm{ref}}\rangle\otimes|0\rangle\otimes|0\rangle$$

to obtain the state

$$\frac{1}{\sqrt{n}}\sum_i |i\rangle\otimes|\mathbf{x}\rangle\otimes|\mathbf{x}_{\mathrm{ref}}\rangle\otimes|\nabla f_i(\mathbf{x})\rangle\otimes|\nabla f_i(\mathbf{x}_{\mathrm{ref}})\rangle\,,$$

and making the difference between the last two registers while regarding other registers as the garbage state.

Since each f_i is ℓ-smooth, we have

$$\|\mathbf{g}_i\| = \|\nabla f_i(\mathbf{x})-\nabla f_i(\mathbf{x}_{\mathrm{ref}})\| \leq \ell\|\mathbf{x}-\mathbf{x}_{\mathrm{ref}}\|,\quad \forall i\in[n].$$

which leads to

$$\mathbb{E}_i\|\mathbf{g}_i-\bar{\mathbf{g}}\|^2 \leq \mathbb{E}_i\|\mathbf{g}_i\|^2 \leq \ell^2\|\mathbf{x}-\mathbf{x}_{\mathrm{ref}}\|^2.$$

Then by Lemma 1, the subroutine `QuantumVarianceReduction` outputs an unbiased estimate $\hat{\mathbf{g}}$ of $\bar{\mathbf{g}}$ with $\mathbb{E}\|\hat{\mathbf{g}}-\bar{\mathbf{g}}\|\leq\hat{\sigma}^2$ using an expected $\tilde{O}(d^{1/2}\ell\|\mathbf{x}-\mathbf{x}_{\mathrm{ref}}\|/\hat{\sigma})$ queries to $O_{\mathbf{g}}$, and thus asymptotically the same number of queries to O_F. □

Proof of Corollary 2. To solve Case 2 of Problem 1, the query complexity equals

$$\begin{aligned}\sum_{s=0}^{S-1}\mathcal{Q}\Big(\ell,\frac{\tilde{\mu}}{2^s}\Big) &= \sum_{s=0}^{S-1}\tilde{O}\big(n+\sqrt{d}+\sqrt{2^s\ell/\tilde{\mu}}\big(n^{1/3}d^{1/3}+n^{-2/3}d^{5/6}\big)\big)\\ &= \tilde{O}(S(n+\sqrt{d}))+\tilde{O}\big(\sqrt{\ell/\tilde{\mu}}\big(n^{1/3}d^{1/3}+n^{-2/3}d^{5/6}\big)\big)\sum_{s=0}^{S-1}2^{s/2}\\ &= \tilde{O}\big(n+\sqrt{d}+R\sqrt{\ell/\epsilon}\big(n^{1/3}d^{1/3}+n^{-2/3}d^{5/6}\big)\big).\end{aligned}$$

To solve Case 3 of Problem 1, the query complexity equals

$$\begin{aligned}\sum_{s=0}^{S-1}\mathcal{Q}\Big(\ell,\frac{\tilde{\mu}}{2^s}\Big) &= \sum_{s=0}^{S-1}\tilde{O}\big(n+\sqrt{2^s/(\lambda\mu)}\big(n^{1/4}d^{1/4}+\sqrt{d}\big)\big)\\ &= \tilde{O}(S(n+\sqrt{d}))+\tilde{O}\big(\big(n^{1/3}d^{1/3}+n^{-2/3}d^{5/6}\big)/\sqrt{\lambda\mu}\big)\sum_{s=0}^{S-1}2^{s/2}\\ &= \tilde{O}\big(n+\sqrt{d}+L\big(n^{1/3}d^{1/3}+n^{-2/3}d^{5/6}\big)/\sqrt{\lambda\mu}\big).\end{aligned}$$

To solve Case 4 of Problem 1, the query complexity equals

$$\begin{aligned}\sum_{s=0}^{S-1}\mathcal{Q}\Big(\ell,\frac{\tilde{\mu}}{2^s}\Big) &= \sum_{s=0}^{S-1}\tilde{O}\big(n+\sqrt{d}+\sqrt{2^{2s}/(\lambda\tilde{\mu})}\big(n^{1/3}d^{1/3}+n^{-2/3}d^{5/6}\big)\big)\\ &= \tilde{O}(S(n+\sqrt{d}))+\tilde{O}\big(\big(n^{1/3}d^{1/3}+n^{-2/3}d^{5/6}\big)/\sqrt{\lambda\mu}\big)\sum_{s=0}^{S-1}2^{s}\\ &= \tilde{O}\big(n+\sqrt{d}+LR\big(n^{1/3}d^{1/3}+n^{-2/3}d^{5/6}\big)/\epsilon\big).\end{aligned}$$

□

B. Proof Details for Quantum Algorithm in the Nonconvex Setting

In this section, we prove Theorem 5. We first present a useful lemma from Fang et al. (2018).

Lemma 8 (Lemma 2 & Lemma 4, Fang et al. (2018))**.** *In the setting of Problem 2, if we have*

$$\mathbb{E}[\mathbf{v}_t]=\nabla f(\mathbf{x}_t)-\nabla f(\mathbf{x}_{t-1}),\quad \mathrm{Var}[\mathbf{v}_t]\leq\frac{\hat{\epsilon}^2}{2q}$$

for any iteration $t\in[\mathcal{T}]$ of Algorithm 3 with $\mathrm{mod}(t,q)\neq 0$, *then the following inequality holds for all $t\in[\mathcal{T}]$:*

$$\mathbb{E}[f(\mathbf{x}_{t+1})-f(\mathbf{x}_t)]\leq-\frac{\hat{\epsilon}}{4\ell}\mathbb{E}\|\mathbf{v}_t\|+\frac{3\hat{\epsilon}^2}{4\ell}.$$

Equipped with Lemma 8, we present the proof of Theorem 5 below.

Proof of Theorem 5. By Lemma 2, for any iteration $t\in[\mathcal{T}]$ of Algorithm 3 with $\mathrm{mod}(t,q)\neq 0$ we have

$$\mathbb{E}[\mathbf{g}_t]=\nabla f(\mathbf{x}_t)-\nabla f(\mathbf{x}_{t-1}),\quad \mathrm{Var}[\mathbf{g}_t]\leq\frac{\hat{\epsilon}^2}{2q}.$$

Hence, by telescoping the result from Lemma 8, we have

$$\frac{\hat{\epsilon}}{4\ell}\sum_{t=0}^{\mathcal{T}-1}\mathbb{E}\|\mathbf{v}_t\|\leq f(\mathbf{0})-\mathbb{E}f(\mathbf{x}_{\mathcal{T}})+\frac{3\mathcal{T}\hat{\epsilon}^2}{4\ell}\leq\Delta+\frac{3\mathcal{T}\hat{\epsilon}^2}{4\ell}$$

and

$$\frac{1}{\mathcal{T}}\sum_{t=0}^{\mathcal{T}-1}\mathbb{E}\|\mathbf{v}_t\|\leq\frac{4\ell\Delta}{\hat{\epsilon}\mathcal{T}}+3\hat{\epsilon}\leq 4\hat{\epsilon},$$

where for each $t\in[\mathcal{T}]$ we have

$$\mathbb{E}\|\mathbf{v}_t\|=\mathbb{E}\|(\mathbf{v}_t-\nabla f(\mathbf{x}_t))+\nabla f(\mathbf{x}_t)\|\geq\mathbb{E}\|\nabla f(\mathbf{x}_t)\|-\mathbb{E}\|\mathbf{v}_t-\nabla f(\mathbf{x}_t)\|\geq\mathbb{E}\|\nabla f(\mathbf{x}_t)\|-\hat{\epsilon}$$

by Lemma 8, which leads to

$$\mathbb{E}\|\mathbf{x}^{\mathrm{out}}\|=\frac{1}{\mathcal{T}}\sum_{t=0}^{\mathcal{T}-1}\mathbb{E}\|\nabla f(\mathbf{x}_t)\|+\hat{\epsilon}\leq 5\hat{\epsilon}=\epsilon,$$

indicating that the output of Algorithm 3 is an expected ϵ-critical point.

The query complexity of Algorithm 3 is a combination of two components: the complete gradient computation step in Line 4 and the QVRG step in Line 6, where each full gradient computation steps and each step takes $O(n)$ queries by using O_F just as a classical finite sum oracle, i.e., we query O_F without employing quantum superposition. As for the second part, as per Lemma 2, each call to QVRG takes an expected

$$\tilde{O}\left(\frac{d^{1/2}\ell\|\mathbf{x}_t-\mathbf{x}_{t-1}\|}{\hat{\epsilon}/\sqrt{2q}}\right)=\tilde{O}(\sqrt{dq/2})$$

queries to the quantum finite-sum oracle. Hence, for every q iterations, the query complexity equals

$$n+q\tilde{O}(\sqrt{dq/2}),$$

and the overall query complexity of Algorithm 3 equals

$$\left(1+\frac{\mathcal{T}}{q}\right)\cdot\left(n+q\tilde{O}(\sqrt{dq/2})\right)=\tilde{O}\left(n+\frac{\ell\Delta}{\epsilon^2}\left(d^{1/3}n^{1/3}+\sqrt{d}\right)\right).$$

□

C. Proof Details for Quantum Lower Bounds

In this section, we list several lemmas to prove our quantum lower bounds for convex settings of Problem 1.

C.1. The strong weighted adversary method

The last step of our proof is using the non-negative quantum adversary method.

Lemma 9 (Lemma 6, Cleve et al. (2012))**.** *Let f be a function from a finite set S to another finite set T, and let Q be a finite set of possible query strings. Given an unknown input $x \in S$, the oracle O_x corresponding to x is the unitary transformation $O_x |q\rangle |a\rangle |z\rangle = |q\rangle |a \oplus \xi(x;q)\rangle |z\rangle$, where q is a query string from Q, $a \in \{0,1\}$ is the register of binary answer, z is the auxiliary register, and $\xi : S \times Q \to \{0,1\}$ is a function that defines the response to oracle queries. Also, let w, w' denote a weight scheme as follows:*

- *Every pair $(x,y) \in S \times S$ is assigned a non-negative weight $w(x,y) = w(y,x)$ that satisfies $w(x,y) = 0$ whenever $f(x) = f(y)$;*
- *Every triple $(x,y,q) \in S \times S \times Q$ is assigned a non-negative weight $w'(x,y,q)$ that satisfies $w'(x,y,q) = 0$ for all x, y, q such that $\xi(x;q) = \xi(y;q)$ or $f(x) = f(y)$, and $w'(x,y,q) \cdot w'(y,x,q) \geq w(x,y)^2$ for all x, y, q such that $\xi(x;q) \neq \xi(y;q)$ and $f(x) \neq f(y)$.*

For all $x \in S$ and $q \in Q$, let $\mu(x) = \sum_y w(x,y)$ and $\nu(x,q) = \sum_y w'(x,y,q)$. Then any quantum algorithm that computes $f(x)$ with success probability at least $\frac{2}{3}$ on an arbitrary input x must make

$$\Omega\left(\min_{\substack{x,y,q;\, w(x,y)>0,\\ \xi(x,q)\neq\xi(y,q)}} \sqrt{\frac{\mu(x)\cdot\mu(y)}{\nu(x,q)\cdot\nu(y,q)}}\right)$$

queries to the oracle O_x.

Using the strong weighted adversary method, we prove the following quantum lower bound for the multi chain problem:

Lemma 10. *Any quantum algorithms that solves Problem 3 on $n \times k$ (n strings, k bits for each string) with success probability at least $\frac{2}{3}$ must take $\Omega(n\sqrt{k})$ queries.*

Proof. Our proof is similar to Ambainis & Montanaro (2014). In our multi-chain problem, the input is a string $x \in \{0,1\}^{n\times k}$, and $f(x)$ is defined as $f(x) = x$. Queries can be formalized as follows: $q = (i, t, x'_{i1}, x'_{i2}, \dots x'_{it})$, where $i \in \{1, 2, \dots n\}$, $t \in \{1, 2, \dots k\}$ and $x'_{ij} \in \{0,1\}$. Here $\xi(x,q) = 1$ if and only if $x'_{ij} = x_{ij}$ for every $j \in 1, 2, \dots t$. Let $d(x,y)$ denote the Hamming distance of two inputs x and y (i.e., the number of differing bits between the two inputs.). Then we define the following weight schemes:

- $w(x,y) = 1$ if $d(x,y) = 1$, and $w(x,y) = 0$ otherwise;
- $w'(x,y,q) = w'(y,x,q) = 1$ if $d(x,y) = 1$ and $\xi(x,q) \neq \xi(y,q)$, and $w'(x,y,q) = w'(y,x,q) = 0$ otherwise.

Then for any $x \in \{0,1\}^{n\times k}$, we have $\mu(x) = nk$ and

$$\nu(x,q) = \sum_y \mathbf{1}_{\{y:\, d(x,y)=1,\ \xi(x,q)\neq\xi(y,q)\}}(y) = \begin{cases} t, & \text{if } \xi(x,q) = 1; \\ 1, & \text{if } \xi(x,q) = 0 \text{ and } |j : j \leq t\ x_{ij} \neq x'_{ij}| = 1; \\ 0, & \text{otherwise.} \end{cases}$$

Therefore,

$$\min_{\substack{x,y,q;\, w(x,y)>0,\\ \xi(x,q)\neq\xi(y,q)}} \sqrt{\frac{\mu(x)\cdot\mu(y)}{\nu(x,q)\cdot\nu(y,q)}} = \min\sqrt{\frac{nk\cdot nk}{1\cdot t}} = \sqrt{\frac{nk\cdot nk}{1\cdot k}} = n\sqrt{k}.$$

By Lemma 9, any quantum algorithms must take $\Omega(n\sqrt{k})$ queries to solve the multi chain problem with success probability more than $\frac{2}{3}$. □

C.2. Reduction from optimization to Problem 3

Problem 4 (Matrix Detection Problem)**.** *Given a matrix where each element is a vector, and each vector can take one of two known possible values. Specifically, given a $n \times k$ matrix A, each element $\mathbf{a}_{ij}$ can be one of the vectors $\mathbf{v}_{ij0}$ or $\mathbf{v}_{ij1}$. We have the following quantum oracle O_A:*

$$O_A\left|i\right\rangle \otimes \left|j\right\rangle \otimes \left|m_1 m_2 \cdots m_j\right\rangle \otimes \left|0\right\rangle\left|0\right\rangle \to \begin{cases} \left|i\right\rangle \otimes \left|j\right\rangle \otimes \left|m_1 m_2 \cdots m_j\right\rangle \otimes \left|1\right\rangle\left|0\right\rangle \\ \qquad \textit{if } \mathbf{a}_{i,p} = \mathbf{v}_{i,p,m_p} \textit{ for each } p \in \{1, 2, , \cdots j\} \textit{ and } \mathbf{a}_{i,j+1} = \mathbf{v}_{i,j+1,0}; \\ \left|i\right\rangle \otimes \left|j\right\rangle \otimes \left|m_1 m_2 \cdots m_j\right\rangle \otimes \left|1\right\rangle\left|1\right\rangle \\ \qquad \textit{if } \mathbf{a}_{i,p} = \mathbf{v}_{i,p,m_p} \textit{ for each } p \in \{1, 2, \cdots j\} \textit{ and } \mathbf{a}_{i,j+1} = \mathbf{v}_{i,j+1,1}; \\ \left|i\right\rangle \otimes \left|j\right\rangle \otimes \left|m_1 m_2 \cdots m_j\right\rangle \otimes \left|0\right\rangle\left|0\right\rangle, \\ \qquad \textit{if there exists } p \in \{1, 2, \cdots j\},\ \mathbf{a}_{i,p} \neq \mathbf{v}_{i,p,m_p}\ . \end{cases}$$

The goal is to output the matrix A.

This problem is quite similar to finding all vectors in our hard function. Intuitively, for each row, we have to start from the beginning and detect the specific values of each vector in order.

Lemma 11. *For dimension $d = \Omega\left(n(k+1) + \frac{8R^2}{c^2}\log\left(2nkN^3\right)\right)$, given a quantum algorithm that finds all $\mathbf{v}_{ij}$ of the hard instance in Definition 3 with success probability more than $\frac{2}{3}$ using $s \leq N$ queries, we can construct a quantum algorithm solving Problem 4 with success probability more than $\frac{2}{3}$ using $O(s)$ queries.*

Proof. First, for the hard instance F, we construct the following quantum oracle:

$$O_{\mathbf{v}}\left|\mathbf{x}_1\right\rangle\left|\mathbf{x}_2\right\rangle \cdots \left|\mathbf{x}_k\right\rangle \otimes \left|i\right\rangle \otimes \left|0\right\rangle\left|0\right\rangle \cdots \left|0\right\rangle \to \left|\mathbf{x}_1\right\rangle\left|\mathbf{x}_2\right\rangle \cdots \left|\mathbf{x}_k\right\rangle \otimes \left|i\right\rangle \otimes \left|\mathbf{x}'_1\right\rangle\left|\mathbf{x}'_2\right\rangle \cdots \left|\mathbf{x}'_k\right\rangle \tag{6}$$

where

$$\mathbf{x}'_j = \begin{cases} \mathbf{v}_{i,j}, & \text{if } \left|\langle \mathbf{x}'_j, \mathbf{v}_{i,j}\rangle\right| > \frac{c}{2} \text{ or } \left|\langle \mathbf{x}'_j, \mathbf{v}_{i,j-1}\rangle\right| > \frac{c}{2} \text{ or } \left|\langle \mathbf{x}'_j, \mathbf{v}_{i,j+1}\rangle\right| > \frac{c}{2}; \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

We demonstrate that when the task is to find a vector $\mathbf{x}$ with a large inner product with each vector $\mathbf{v}_{i,j}$, this oracle is stronger than the quantum finite-sum oracle (3). Specifically, we can simulate a query to (3) using a query to (6). For example, consider the hard instance in Definition 3. For a query to oracle (3) with input $\left|\mathbf{x}\right\rangle \otimes \left|i\right\rangle$, consider a query to $O_{\mathbf{v}}$ with input $\left|\mathbf{x}\right\rangle\left|\mathbf{x}\right\rangle \cdots \left|\mathbf{x}\right\rangle \otimes \left|i\right\rangle$. Observe that for j with $\max\{\langle \mathbf{x}, \mathbf{v}_{i,j-1}\rangle, \langle \mathbf{x}, \mathbf{v}_{i,j}\rangle, \langle \mathbf{x}, \mathbf{v}_{i,j+1}\rangle\} \leq \frac{c}{2}$, since the helper function takes value 0, $\nabla f_i(\mathbf{x})$ does not contain any information about $\mathbf{v}_{i,j}$. Therefore, $\nabla f_i(\mathbf{x})$ can be computed by the output of the oracle $O_{\mathbf{v}}$.

For a quantum algorithm A with access to $O_{\mathbf{v}}$ that can find the values of all events, suppose A makes at most N queries. We define "bad event" to be

$$B_{i,j,t} := \left[\text{the input of } t\text{-th query } \mathbf{x}_j^{(t)} \text{ satisfies } \left|\langle \mathbf{x}_j^{(t)}, \mathbf{v}_{i,j}\rangle\right| > \frac{c}{2}, \text{ but } \mathbf{v}_{i,j} \text{ is not the output of any previous query}\right].$$

Intuitively, when $B_{i,j,t}$ happens, before A queries $\mathbf{x}_j^{(t)}$, A knows at most $t-1$ vectors that have relatively small inner products with $\mathbf{v}_{i,j}$.

Since $\mathbf{v}_{i,j}$ is randomly chosen as a unit vector orthogonal to other $\mathbf{v}$, we can bound the probability that the inner product of a fixed unit vector and a uniformly random unit vector in $d - n(k+1) + 1$ dimensions is larger than $\frac{c}{2}$. This probability can be explained as the ratio of the combined areas of the upper and lower caps with a radius of $r := \sqrt{1 - \left(\frac{c}{2}\right)^2}$ to the surface area of the sphere. Thus,

$$\Pr\left[\left|\langle \mathbf{x}, \mathbf{v}_{i,j}\rangle\right| > \frac{c}{2}\right] \leq \frac{r^{d-n(k+1)+1}}{1^{d-n(k+1)+1}} = \left(1 - \left(\frac{c}{2}\right)^2\right)^{\frac{d-n(k+1)+1}{2}} \leq e^{-\frac{c^2(d-n(k+1)+1)}{8}}.$$

Therefore, the probability that the event $B_{i,j,t}$ occurs is less than the probability of finding $\mathbf{x}$ such that $|\langle \mathbf{x}, \mathbf{v}_{i,j}\rangle| > \frac{c}{2}$ using less than t queries, for some constant h:

$$\Pr[B_{i,j,t}] \leq t^2 h \cdot e^{-\frac{c^2(d-n(k+1)+1)}{8}}.$$

Define "good event" $G := \cup_{i=1}^{n/2} \cup_{j=1}^{k} \cup_{t=1}^{N} \overline{B_{i,j,t}}$. Thus, the probability that all "bad" event do not happen is

$$\Pr[G] \geq 1 - \sum_{i=1}^{n/2} \sum_{j=1}^{k} \sum_{t=1}^{N} \Pr[B_{i,j,t}] \leq 1 - \frac{1}{2} nkhN^3 e^{-\frac{c^2(d-n(k+1)+1)}{8}}.$$

Take $d \geq n(k+1) + \frac{8}{c^2} \log(2nkhN^3)$, we have

$$\Pr[G] \geq 1 - \frac{1}{16} = \frac{15}{16}.$$

The above proof assumes that all queries are within a sphere of radius 1. For a general radius R, we can replace c with $\frac{c}{R}$ to adjust the parameters accordingly.

The occurrence of the "good" event implies that with high probability, the algorithm cannot guess the value of the vector but can only obtain information about the vector through the provided oracle $O_{\mathbf{v}}$. In such circumstance, we can simulate $O_{\mathbf{v}}$ using an another oracle $O'_{\mathbf{v}}$ that demands more precise inputs:

$$O_v |\mathbf{x}_1\rangle |\mathbf{x}_2\rangle \cdots |\mathbf{x}_k\rangle \otimes |i\rangle \otimes |0\rangle |0\rangle \cdots |0\rangle \to |\mathbf{x}_1\rangle |\mathbf{x}_2\rangle \cdots |\mathbf{x}_k\rangle \otimes |i\rangle \otimes |\mathbf{x}'_1\rangle |\mathbf{x}'_2\rangle \cdots |\mathbf{x}'_k\rangle \tag{7}$$

where

$$\mathbf{x}'_j = \begin{cases} \mathbf{v}_{i,j}, & \text{if } \mathbf{x}_{j'} = \mathbf{v}_{i,j'} \text{ for all } j' < j; \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

Note that when G occurs, when the algorithm queries $\mathbf{x}_j^{(t)}$ such that $\left|\langle \mathbf{x}_j^{(t)}, \mathbf{v}_{i,j}\rangle\right| > \frac{c}{2}$, $\mathbf{v}_{i,j}$ has been the output of a previous query, so we can directly substitute the input with $\mathbf{v}_{i,j}$, and the algorithm still works using the oracle $O'_{\mathbf{v}}$. If event $B_{i,j,t}$ occurs, we simply mark it as a failure of the algorithm.

Now, we can consider that a quantum algorithm using oracle O_F to find all $\mathbf{v}_{i,j}$'s can be implemented with a quantum algorithm using $O'_{\mathbf{v}}$ (with a small failure probability less than $\frac{1}{16}$) with the same number of queries. Adding a constraint on $\mathbf{v}_{i,j}$, specifically that $\mathbf{v}_{i,j}$ is one of the two known vectors $\mathbf{v}_{i,j,0}$ and $\mathbf{v}_{i,j,1}$, implies that the task of using $O'_{\mathbf{v}}$ to find all vectors is exactly equivalent to Problem 4. Therefore, from a quantum algorithm using oracle O_F that finds all $\mathbf{v}_{i,j}$ of the hard instance with success probability more than $\frac{2}{3}$ using s queries, we can construct a quantum algorithm solving Problem 4 using $O(s)$ quantum queries through the aforementioned reduction process. The failure probability could be controlled by repeating the algorithm a constant number of times. □

Next, we present the following lemma, demonstrating that the complexity of a quantum algorithm to find vectors for all sub-functions is only different from finding vectors for half of the sub-functions by a logarithmic factor. This lemma is essential in the proof of cases 2-4 of Problem 1, as in these situations, finding an ϵ-optimal point for the total function only requires acquiring information about half of the sub-functions.

Lemma 12. *For dimension $d = \Omega\left(n(k+1) + \frac{8}{c^2}\log(2nkN^3)\right)$, given a quantum algorithm which finds vectors $\mathbf{v}_{i,j}$ for all j of $\frac{n}{4}$ different (and uncertain) i's of a hard instance with success probability more than $\frac{2}{3}$ using s queries, we can construct a quantum algorithm solving Problem 4 with success probability more than $\frac{2}{3}$ using $O\big((s + \frac{n}{2})\log n\big)$ quantum queries, where N is the maximum number of query times.*

Proof. The proof is similar to the proof for Lemma 11. For the task of finding a vector $\mathbf{x}$ such that $|\langle \mathbf{x}, \mathbf{v}_{i,j}\rangle| > \frac{c}{2}$ for all j of $\frac{n}{4}$ different (and uncertain) i's, $O_{\mathbf{v}}$ (6) is stronger than the quantum finite-sum oracle O_F (3). Specifically, we can simulate a query to (3) using a query to (6). Therefore, given a quantum algorithm using oracle O_F, we can construct another algorithm A using oracle $O_{\mathbf{v}}$ with the same number of queries. Now we directly use A to construct an algorithm B that finds $\mathbf{x}'$ such that $\langle \mathbf{x}, \mathbf{v}_{i,j}\rangle < \frac{c}{2}$ for **all** i, j**'s** of the same hard instance.

The algorithm B works as follows: Repeat the following steps T times, where T will be specified later. In the t-th round, B randomly permute the $\frac{n}{2}$ items $(f_{i,1}, f_{i,2})$ for $i = 1, 2, \cdots \frac{n}{2}$. Then it simulates A and outputs $\mathbf{x}_t$. After receiving $\mathbf{x}_t$, it queries $O_{\mathbf{v}}$ with inputs $|\mathbf{x}_t\rangle\,|\mathbf{x}_t\rangle \cdots |\mathbf{x}_t\rangle \otimes |i\rangle$ for all i. Since $\mathbf{x}_t$ satisfies that $|\langle \mathbf{x}_t, \mathbf{v}_{i,j}\rangle| > \frac{c}{2}$ for all j of $\frac{n}{4}$ different i's, the above $\frac{n}{2}$ queries will receive values of $\mathbf{v}_{i,j}$ for all j corresponding to at least $\frac{n}{4}$ different i's. After the T rounds, B output $\sum_{i=1}^{n/2}\sum_{j=1}^{k} \mathbf{v}_{i,j}$ if it receives all $\mathbf{v}_{i,j}$, and fails otherwise.

Now we bound the failure probability. In each round, the probability that vectors of i-th sub-function are found is larger than $\frac{1}{2}$, and due to the independence between each round, $\Pr[\ B \text{ fails to find the information of } \mathbf{v}_{i,j} \text{ for all } j \leq k\ ] \leq \left(\frac{1}{2}\right)^T$. Hence

$$\begin{aligned}
&\Pr\Big[\ B \text{ finds the value of } \mathbf{v}_{i,j} \text{ for all } i \leq \frac{n}{2} \text{ and } j \leq k\ \Big] \\
&\geq 1 - \sum_{i=1}^{\frac{n}{2}} \Pr[\ B \text{ fails to find the value of } \mathbf{v}_{i,j} \text{ for all } j \leq k\ ] \\
&\geq 1 - \frac{n}{2} \cdot \left(\frac{1}{2}\right)^T .
\end{aligned}$$

Take $T = 4 + \lceil \log\left(\frac{n}{2}\right) \rceil$, the success probability of B is larger than $\frac{15}{16}$. The number of queries to the oracle $O_{\mathbf{v}}$ is $\Theta\big(\big((s + \frac{n}{2}\big) \log n\big)\big)$.

The remaining steps are consistent with Lemma 11. We can construct a quantum algorithm to solve Problem 4 with complexity $O\big(\big((s + \frac{n}{2}\big) \log n\big)\big)$. Here we repeat the algorithm a constant number of times to control the failure probability. □

Next, we prove that Problem 4 can be reduced to Problem 3, for which problem we can construct a lower bound using the adversary method.

Lemma 13. *Given an algorithm that solves Problem 4 for input size $n \times k$ with success probability larger than $\frac{2}{3}$ using s queries, we can construct a quantum algorithm solving Problem 3 for the same input size with success probability larger than $\frac{2}{3}$ using $O(s)$ queries.*

Proof. The proof is based on a reduction between the two quantum oracles. While using existing results on adversary bounds, it is worth noting that a quantum oracle in Problem 4 can be implemented using two oracles from Problem 3: We can simulate the results of $O_A\,|i\rangle \otimes |j\rangle \otimes |m_1 m_2 \cdots m_j\rangle \otimes |0\rangle\,|0\rangle$ by querying oracle O_C with inputs $|i\rangle \otimes |j+1\rangle \otimes |m_1 m_2 \cdots m_j 0\rangle \otimes |0\rangle$ and $|i\rangle\,|j+1\rangle \otimes |m_1 m_2 \cdots m_j 1\rangle \otimes |0\rangle$, adding several quantum unitaries. Specifically, the first qubit of the oracle O_A is $|1\rangle$ if and only if the output of two queries to the oracle O_C contains at least one $|1\rangle$, which means that $(m_1, \cdots m_j) = (a_{i,1}, \cdots a_{i,j})$. Under the above conditions the second qubit of the oracle O_A is $|0\rangle$ when the first query of O_C is $|1\rangle$ and $|1\rangle$ otherwise.

Since the output of Problem 4 can be represented as a binary matrix, which is the same as Problem 3, we can build an instance of Problem 4 out of an instance of Problem 3 by adding several random vectors. Therefore, We can construct an algorithm with query complexity $O(s)$ to solve Problem 3. □

C.3. Proofs for the smooth and strongly convex setting

The proof of Lemma 5 is inspired by Theorem 8 of Woodworth & Srebro (2016). Note that compared to their proof, we show that to find an ϵ-optimal point we need the information of all vectors rather than just one vector located later in each sub-function.

Proof of Lemma 5. Intuitively, our proof strategy is roughly as follows: We initially decompose $F(\mathbf{x})$ into the average of several sub-functions defined on orthogonal subspaces. If the projection of $\mathbf{x}$ onto a certain subspace has a small enough inner product with one of the randomly chosen vectors, we directly prove that the function value corresponding to $\mathbf{x}$ on that subspace has a sufficiently large gap from the minimum value of that sub-function.

Firstly, in order to bound the influence of $\langle \mathbf{x}, \mathbf{v}_{i,j} \rangle$ on the total function $F(\mathbf{x})$, it is convenient to bundle together all terms affecting the value of $\langle \mathbf{x}, \mathbf{v}_{i,j} \rangle$ from the components of all sub-functions. These terms are contained in $f_{i,1}(\mathbf{x})$, $f_{i,2}(\mathbf{x})$ and $\psi(\mathbf{x})$. Consider the projection operator P_i which projects the vector $\mathbf{x}$ onto the subspace spanned by the vector set $\{\mathbf{v}_{i,j}\}_{j=0}^{k}$, and define $P_\perp$ as the operator projecting $\mathbf{x}$ onto the subspace orthogonal to $\mathbf{v}_{i,j}$ for all i, j. Then we have

$$\psi(\mathbf{x}) = \frac{\mu}{2}\|x\|^2 = \frac{\mu}{2}\sum_{i=1}^{n/2}\left(\|P_i\mathbf{x}\|^2\right) + \frac{\mu}{2}\|P_\perp \mathbf{x}\|^2.$$

Split $\psi(\mathbf{x})$ amongst $f_{i,1}$ and $f_{i,2}$, we obtain the modified sub-functions:

$$\tilde{f}_{i,1}(\mathbf{x}) = f_{i,1} + \frac{\tilde{\mu}}{4}\|P_i\mathbf{x}\|^2, \quad \tilde{f}_{i,2}(\mathbf{x}) = f_{i,2} + \frac{\tilde{\mu}}{4}\|P_i\mathbf{x}\|^2.$$

Then, we can represent the total function $F(\mathbf{x})$ as follows:

$$F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n/2}\left(\tilde{f}_{i,1}(\mathbf{x}) + \tilde{f}_{i,2}(\mathbf{x})\right) + \frac{\mu}{2}\|P_\perp \mathbf{x}\|^2.$$

Notice that there is still a remaining term $\frac{\mu}{2}\|P_\perp \mathbf{x}\|^2$ here, but this part is not crucial for our analysis, as when minimizing the total function $F(\mathbf{x})$, we can always set $P_\perp \mathbf{x} = \mathbf{0}$. Next, we consider the summation

$$\begin{aligned}&\frac{1}{2}\left(\tilde{f}_{i,1}(\mathbf{x}) + \tilde{f}_{i,2}(\mathbf{x})\right)\\ =\ &\frac{1-\tilde{\mu}}{32}\big(\left\langle \mathbf{x},\, \mathbf{v}_{i,0}\right\rangle^2 - 2C\left\langle \mathbf{x},\, \mathbf{v}_{i,0}\right\rangle + \zeta\phi_c(\langle \mathbf{x},\, \mathbf{v}_{i,k}\rangle) + \sum_{j=1}^{k}\phi_c\left(\langle \mathbf{x},\, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x},\, \mathbf{v}_{i,r}\rangle\right)\big) + \frac{\tilde{\mu}}{4}\|P_i\mathbf{x}\|^2.\end{aligned}$$

We define

$$F_i(\mathbf{x}) = \frac{1-\tilde{\mu}}{32}\left(\left\langle \mathbf{x},\, \mathbf{v}_{i,0}\right\rangle^2 - 2C\left\langle \mathbf{x},\, \mathbf{v}_{i,0}\right\rangle + \zeta\left\langle \mathbf{x},\, \mathbf{v}_{i,k}\right\rangle^2 + \sum_{j=1}^{k}\left(\langle \mathbf{x},\, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x},\, \mathbf{v}_{i,r}\rangle\right)^2\right) + \frac{\tilde{\mu}}{4}\|P_i\mathbf{x}\|^2$$

and

$$F_i^t(\mathbf{x}) = \frac{1-\tilde{\mu}}{32}\left(\left\langle \mathbf{x},\, \mathbf{v}_{i,0}\right\rangle^2 - 2C\left\langle \mathbf{x},\, \mathbf{v}_{i,0}\right\rangle + \left\langle \mathbf{x},\, \mathbf{v}_{i,t}\right\rangle^2 + \sum_{j=1}^{t}\left(\langle \mathbf{x},\, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x},\, \mathbf{v}_{i,r}\rangle\right)^2\right) + \frac{\tilde{\mu}}{4}\|P_i\mathbf{x}\|^2.$$

Intuitively, $F_i(\mathbf{x})$ is an approximation to $\frac{1}{2}\left(\tilde{f}_{i,1}(\mathbf{x}) + \tilde{f}_{i,2}(\mathbf{x})\right)$, and $F_i^t(\mathbf{x})$ is a truncation of $F_i(\mathbf{x})$ to $\mathbf{v}_{i,t}$, neglecting the vectors beyond $\mathbf{v}_{i,t+1}$. Consider the helper function (5), we know that when $|z| \le c$, the function is constant at 0, and for any z,

$$z^2 - 2c^2 \le \phi_c(z) \le z^2. \tag{8}$$

According to the properties of the helper function, we have

$$\begin{aligned}F_i(\mathbf{x}) &\le \frac{1}{2}\left(\tilde{f}_{i,1}(\mathbf{x}) + \tilde{f}_{i,2}(\mathbf{x})\right) + \frac{(1-\tilde{\mu})(k+\zeta)}{16}c^2\\ \frac{1}{2}\left(\tilde{f}_{i,1}(\mathbf{x}) + \tilde{f}_{i,2}(\mathbf{x})\right) &\le F_i(\mathbf{x}).\end{aligned}$$

for any $\mathbf{x}$. For convenience, let $Q \coloneqq \frac{1}{2}(\frac{1}{\tilde{\mu}} - 1) + 1$, then

$$\begin{aligned}F_i(\mathbf{x}) &= \frac{1}{2}\left(\frac{\tilde{\mu}(Q-1)}{8}\left(\left\langle \mathbf{x},\, \mathbf{v}_{i,0}\right\rangle^2 - 2C\left\langle \mathbf{x},\, \mathbf{v}_{i,0}\right\rangle + \zeta\left\langle \mathbf{x},\, \mathbf{v}_{i,k}\right\rangle^2 + \sum_{j=1}^{k}\left(\langle \mathbf{x},\, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x},\, \mathbf{v}_{i,r}\rangle\right)^2\right) + \frac{\tilde{\mu}}{2}\|P_i\mathbf{x}\|^2\right),\\ F_i^t(\mathbf{x}) &= \frac{1}{2}\left(\frac{\tilde{\mu}(Q-1)}{8}\left(\left\langle \mathbf{x},\, \mathbf{v}_{i,0}\right\rangle^2 - 2C\left\langle \mathbf{x},\, \mathbf{v}_{i,0}\right\rangle + \left\langle \mathbf{x},\, \mathbf{v}_{i,t}\right\rangle^2 + \sum_{j=1}^{t}\left(\langle \mathbf{x},\, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x},\, \mathbf{v}_{i,r}\rangle\right)^2\right) + \frac{\tilde{\mu}}{2}\|P_i\mathbf{x}\|^2\right).\end{aligned}$$

Let $\hat{\mathbf{x}} \coloneqq \arg\min_{\mathbf{x}} F_i(\mathbf{x})$, by the first-order condition for the minimum value of $F_i(\mathbf{x})$, $\hat{\mathbf{x}}$ satisfies

$$\begin{gathered}
2\frac{Q+1}{Q-1}\langle \hat{\mathbf{x}}, \mathbf{v}_{i,0}\rangle - \langle \hat{\mathbf{x}}, \mathbf{v}_{i,1}\rangle = C, \\
\langle \hat{\mathbf{x}}, \mathbf{v}_{i,j-1}\rangle - 2\frac{Q+1}{Q-1}\langle \hat{\mathbf{x}}, \mathbf{v}_{i,j}\rangle + \langle \hat{\mathbf{x}}, \mathbf{v}_{i,j+1}\rangle = 0, \\
\big(1+\zeta+\frac{4}{Q-1}\big)\langle \hat{\mathbf{x}}, \mathbf{v}_{i,k}\rangle - \langle \hat{\mathbf{x}}, \mathbf{v}_{i,k-1}\rangle = 0
\end{gathered}$$

for $j = 1, 2, \cdots k-1$.

Define $q \coloneqq \frac{\sqrt{Q}-1}{\sqrt{Q}+1}$ $(q < 1)$ and set $\zeta = 1 - q$. Then $\hat{\mathbf{x}}$ can be expressed as follows:

$$\hat{\mathbf{x}} = C\sum_{j=0}^{k} q^{j+1}\mathbf{v}_j$$

and

$$F_i(\hat{\mathbf{x}}) = -\frac{\tilde{\mu}C^2}{16}(\sqrt{Q}-1)^2.$$

Therefore, the suboptimality of point $\mathbf{0}$ is $\epsilon_i \coloneqq F_i(\mathbf{0}) - F_i(\hat{\mathbf{x}}) = \frac{\mu C^2}{16}(\sqrt{Q}-1)^2$.

Now consider an arbitrary vector $\mathbf{x}$. If there exists an index t that $|\langle \mathbf{x}, \mathbf{v}_{it}\rangle| < \frac{c}{2}$, take $c \leq Cq^{k+1}$ since F_i is a $\frac{\tilde{\mu}}{2}$-strongly convex function, $F_i(\mathbf{x}) - F(\hat{\mathbf{x}}) \geq \frac{\tilde{\mu}}{4}\|\mathbf{x} - \hat{\mathbf{x}}\|^2$. Thus

$$\begin{aligned}
&\frac{F_i(\mathbf{x}) - F(\hat{\mathbf{x}})}{F_i(\mathbf{0}) - F(\hat{\mathbf{x}})} \\
\geq\; &\frac{\frac{\tilde{\mu}}{4}\|\mathbf{x} - \hat{\mathbf{x}}\|^2}{\frac{\mu C^2}{16}(\sqrt{Q}-1)^2} \\
\geq\; &\frac{4}{C^2}\cdot\frac{(\left|Cq^{k+1}\right| - |c|)^2}{(\sqrt{Q}-1)^2} \\
\geq\; &\frac{1}{C^2}\cdot\frac{C^2q^{2k+2}}{(\sqrt{Q}-1)^2} \\
=\; &\frac{1}{(\sqrt{Q}-1)^2}\cdot\exp\bigg\{-2(k+1)\log\frac{1}{q}\bigg\} \\
=\; &\frac{1}{(\sqrt{Q}-1)^2}\cdot\exp\bigg\{-2(k+1)\log{(1+\frac{2}{\sqrt{Q}-1})}\bigg\} \\
\geq\; &\frac{1}{(\sqrt{Q}-1)^2}\cdot\exp\bigg\{\frac{-4(k+1)}{\sqrt{Q}-1}\bigg\}.
\end{aligned}$$

Take $k = \lfloor\frac{\sqrt{Q}-1}{4}\log\frac{\epsilon_i}{n\epsilon(\sqrt{Q}-1)^2)}\rfloor - 1$, since $t \leq k$, we have

$$\frac{F_i(\mathbf{x}) - F(\hat{\mathbf{x}})}{F_i(\mathbf{0}) - F(\hat{\mathbf{x}})} \geq \frac{n\epsilon}{\epsilon_i},$$

which leads to

$$\begin{aligned}
n\epsilon &\leq F_i^t(\mathbf{x}) - F_i(\hat{\mathbf{x}}) \\
&\leq \frac{1}{2}\left(\tilde{f}_{i,1}(\mathbf{x}) + \tilde{f}_{i,2}(\mathbf{x})\right) + \frac{(1-\tilde{\mu})(k+\zeta)}{16}c^2 - \frac{1}{2}\left(\tilde{f}_{i,1}(\hat{\mathbf{x}}) + \tilde{f}_{i,2}(\hat{\mathbf{x}})\right) \\
&\leq \frac{1}{2}\left(\tilde{f}_{i,1}(\mathbf{x}) + \tilde{f}_{i,2}(\mathbf{x})\right) + \frac{(1-\tilde{\mu})(k+1)}{16}c^2 - \frac{1}{2}\left(\tilde{f}_{i,1}(\mathbf{x}^*) + \tilde{f}_{i,2}(\mathbf{x}^*)\right).
\end{aligned}$$

Thus,

$$\left(\tilde{f}_{i,1}(\mathbf{x}) + \tilde{f}_{i,2}(\mathbf{x})\right) - \left(\tilde{f}_{i,1}(\mathbf{x}^*) + \tilde{f}_{i,2}(\mathbf{x}^*)\right) \geq 2n\epsilon - \frac{(1-\tilde{\mu})(k+1)}{8}c^2.$$

Take

$$c = \min\left\{\frac{1}{\sqrt{N}}, \sqrt{\frac{8n\epsilon}{(1-\tilde{\mu})(k+1)}}, Cq^{k+1}\right\},$$

then we have

$$\left(\tilde{f}_{i,1}(\mathbf{x}) + \tilde{f}_{i,2}(\mathbf{x})\right) - \left(\tilde{f}_{i,1}(\mathbf{x}^*) + \tilde{f}_{i,2}(\mathbf{x}^*)\right) \geq n\epsilon.$$

Therefore, if there exists a vector $\mathbf{v}_{ij}$ which holds that $|\langle \mathbf{x}, \mathbf{v}_{ij}\rangle| < \frac{c}{2}$, then $\mathbf{x}$ cannot be an ϵ-optimal point of the hard function $F(\mathbf{x})$. □

Proof of Corollary 3. Our proof relies on some results related to adversary bounds, which are detailed in Appendix C.1.

By Lemma 5, of for a hard function $F(\mathbf{x})$ defined in Definition 3, with parameters set by Lemma 5, we must output a vector $\mathbf{x}$ such that $|\langle \mathbf{x},\, \mathbf{v}_{i,j}\rangle| > \frac{c}{2}$ for any i and j to find an ϵ-optimal point of $F(\mathbf{x})$.

By Lemma 11 and Lemma 13, when d is sufficiently large, for a quantum algorithm that finds $\mathbf{x}$ with the above requirements using s queries, we can easily construct an algorithm solving Problem 3 with input size $n \times k$ using $O(s)$ queries. Then, we can use the adversary method to provide a lower bound on the complexity of Problem 3. Consider Lemma 10, any quantum algorithms solving Problem 3 must take at least $\Omega(n\sqrt{k})$ queries. This gives a lower bound for case 1 of Problem 1:

$$s = n\sqrt{k} = \Omega\left(n^{\frac{3}{4}}\left(\frac{1}{\mu}\right)^{\frac{1}{4}}\log^{\frac{1}{2}}\left(\frac{\Delta\mu}{\epsilon}\right)\right).$$

Considering the smoothness parameter ℓ, we scale the hard function as follows:

$$F'(\mathbf{x}) = \frac{1}{\ell}F(\mathbf{x}),$$

then $f_i'(\mathbf{x})$ is 1-lipschitz and $\psi'(\mathbf{x})$ is $\frac{\mu}{\ell}$-strongly convex, and the ϵ-optimal point of $F(\mathbf{x})$ is equivalent to the $\frac{\epsilon}{\ell}$-optimal point of $F(\mathbf{x})$. We can complete the proof using a lower bound with the parameter ℓ:

$$\Omega\left(n^{\frac{3}{4}}\left(\frac{\ell}{\mu}\right)^{\frac{1}{4}}\log^{\frac{1}{2}}\left(\frac{\Delta\mu}{\epsilon\ell}\right)\right).$$

Note that this proof also provides a trivial lower bound $\Omega(n)$ since k is greater than 1, and hence our lower bound is

$$\Omega\left(n + n^{\frac{3}{4}}\left(\frac{\ell}{\mu}\right)^{\frac{1}{4}}\log^{\frac{1}{2}}\left(\frac{\Delta\mu}{\epsilon\ell}\right)\right).$$

Finally, we calculate the required constraints on the dimension d. By strong-convexity $F(\mathbf{0}) \leq F(\mathbf{x}^*) + \frac{\mu}{2}\|\mathbf{x}^*\|^2$, so $\|\mathbf{x}^*\| \leq \sqrt{\frac{2\Delta}{\mu}} := R$. Since the optimal point lies in the R-ball, we restrict the algorithm to query only at points $\mathbf{x}$ such that $\|\mathbf{x}\| \leq R$. We argue that by a slight modification of the hard instance, Querying points beyond the R-ball will not yield additional information. The statement is based on the construction in Appendix C.4 of Woodworth & Srebro (2016). Define $f'_{i,j}$ through its gradient as:

$$\nabla f'_{i,j}(\mathbf{x}) = \begin{cases} \nabla f_{i,j}(\mathbf{x}) & \|\mathbf{x}\| \leq R; \\ \nabla f_{i,j}\left(R\frac{\mathbf{x}}{\|\mathbf{x}\|}\right) & \|\mathbf{x}\| \geq R. \end{cases}$$

$$\nabla \psi'(\mathbf{x}) = \begin{cases} \nabla \psi(\mathbf{x}) & \|\mathbf{x}\| \leq R; \\ \nabla \psi\left(R\frac{\mathbf{x}}{\|\mathbf{x}\|}\right) - \mu R\frac{\mathbf{x}}{\|\mathbf{x}\|} + \frac{\mu}{2}\|\mathbf{x}\|^2 & \|\mathbf{x}\| \geq R. \end{cases}$$

Note that for the new construcion, $f_i'(\mathbf{x})$ is continuous and ℓ-smooth, and $\psi'(\mathbf{x})$ is still μ-strongly convex. Furthermore, it also has the property that querying the function at a point $\mathbf{x}$ beyond the R-ball cannot find more information than querying at $R\frac{\mathbf{x}}{\|\mathbf{x}\|}$. So we can restrict the algorithm not to query points outside the R ball while still maintaining the same capability. Then, we can use the Lemma 11 to calculate the requirements for dimension d: $n(k+1) + \frac{8R^2}{c^2}\log\left(2nkN^3\right) = O\left(\frac{\Delta\ell^2}{\mu^2 n\epsilon}\log^2\frac{\Delta\mu}{\epsilon\ell}\log\frac{n\ell}{\mu}\right)$, so by Lemma 11 we can take $d = \left(\frac{\Delta\ell^2}{\mu^2 n\epsilon}\log^2\frac{\Delta\mu}{\epsilon\ell}\log\frac{n\ell}{\mu}\right)$. □

C.4. Proofs for the smooth and non-strongly convex setting

Proof of Lemma 6. The proof is inspired by the results from Appendix C.3 in Woodworth & Srebro (2016).

Without loss of gengerality, we can assume that $\ell = R = 1$. As in Definition 4, for the parameters C, k and c given later, we define $\frac{n}{2}$ pairs of functions:

$$f_{i,1}(\mathbf{x}) = \frac{1}{16}\Big(\langle \mathbf{x}, \mathbf{v}_{i,0}\rangle^2 - 2C\langle \mathbf{x}, \mathbf{v}_{i,0}\rangle + \sum_{r \text{ even}}^{k}\phi_c\left(\langle \mathbf{x}, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x}, \mathbf{v}_{i,r}\rangle\right)\Big),$$

$$f_{i,2}(\mathbf{x}) = \frac{1}{16}\Big(\phi_c(\langle \mathbf{x}, \mathbf{v}_{i,k}\rangle) + \sum_{r \text{ odd}}^{k}\phi_c\left(\langle \mathbf{x}, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x}, \mathbf{v}_{i,r}\rangle\right)\Big)$$

where $\mathbf{v}_{ij}$ are orthonormal vectors chosen randomly on the unit sphere in $\mathbb{R}^d$. For the non-strongly convex case, we directly set $\psi(\mathbf{x})$ to 0.

For $i \in \{1, 2, \cdots \frac{n}{2}\}$, define

$$\begin{aligned} F_i(\mathbf{x}) &= \frac{1}{2}\big(f_{i,1}(\mathbf{x}) + f_{i,2}(\mathbf{x})\big) \\ &= \frac{1}{32}\Big(\langle \mathbf{x}, \mathbf{v}_{i,0}\rangle^2 - 2C\langle \mathbf{x}, \mathbf{v}_{i,0}\rangle + \sum_{r=1}^{k}\phi_c\left(\langle \mathbf{x}, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x}, \mathbf{v}_{i,r}\rangle\right) + \phi_c(\langle \mathbf{x}, \mathbf{v}_{i,k}\rangle)\Big) \end{aligned}$$

then the total function is

$$F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{\frac{n}{2}}\sum_{j=1}^{2} f_{i,j}(\mathbf{x}) = \frac{2}{n}\sum_{i=1}^{\frac{n}{2}}(F_i(\mathbf{x})).$$

To estimate the value of the function $F_i(\mathbf{x})$, we define

$$F_i'(\mathbf{x}) = \frac{1}{32}\Big(\langle \mathbf{x}, \mathbf{v}_{i,0}\rangle^2 - 2C\langle \mathbf{x}, \mathbf{v}_{i,0}\rangle + \sum_{r=1}^{k}\left(\langle \mathbf{x}, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x}, \mathbf{v}_{i,r}\rangle\right)^2 + (\langle x, \mathbf{v}_{i,k}\rangle)^2\Big).$$

Considering the property (8) of the helper function, we have

$$F_i'(\mathbf{x}) - \frac{k+1}{16}c^2 \le F_i(\mathbf{x}) \le F_i'(\mathbf{x}). \tag{9}$$

By first order optimality conditions for F_i', the optimum point $\tilde{\mathbf{x}}_i$ of $F_i'(\mathbf{x})$ must satisfy that

$$\begin{gathered} 2\langle \tilde{\mathbf{x}}_i, \mathbf{v}_{i,0}\rangle - \langle \tilde{\mathbf{x}}_i, \mathbf{v}_{i,1}\rangle = C; \\ \langle \tilde{\mathbf{x}}_i, \mathbf{v}_{i,j-1}\rangle - 2\langle \tilde{\mathbf{x}}_i, \mathbf{v}_{i,j}\rangle + \langle \tilde{\mathbf{x}}_i, \mathbf{v}_{i,j+1}\rangle = 0 \quad \text{for } 1 \le j \le k-1; \\ \langle \tilde{\mathbf{x}}_i, \mathbf{v}_{i,k-1}\rangle - 2\langle \tilde{\mathbf{x}}_i, \mathbf{v}_{i,k}\rangle = 0. \end{gathered}$$

It can be easily verified that the solution to the above system is

$$\tilde{\mathbf{x}}_i = C\sum_{j=0}^{k}\left(1 - \frac{j+1}{k+2}\right)\mathbf{v}_{i,j}$$

and the corresponding minimum function value is

$$F_i'(\tilde{\mathbf{x}}_i) = -\frac{C^2}{32}\frac{k+1}{k+2}.$$

Now, we can calculate the norm of the optimal point $\tilde{\mathbf{x}}_i$:

$$\begin{aligned}
\|\tilde{\mathbf{x}}_i\|^2 &= C^2 \cdot \sum_{j=0}^{k}\left(1 - \frac{j+1}{k+2}\right)^2 \\
&= C^2 \cdot \frac{(k+1)(k+2)(2k+3)}{6(k+2)^2} \\
&\leq \frac{C^2 k}{3}, \quad \text{when } k \geq 1.
\end{aligned}$$

Due to the orthogonality of $\mathbf{v}_{i,j}$, the minimization of the total function $F'(\mathbf{x}) = \frac{2}{n}\sum_{i=1}^{\frac{n}{2}} F_i'(\mathbf{x})$ can be equivalently represented as the minimization over different orthogonal subspaces $V_i = \operatorname{span}\{\mathbf{v}_{i,0}, \mathbf{v}_{i,1}, \cdots \mathbf{v}_{i,k}\}$, i.e., the minimization over $\frac{n}{2}$ sub-functions. Therefore, $\tilde{\mathbf{x}}' \coloneqq \sum_{i=1}^{\frac{n}{2}} \tilde{\mathbf{x}}_i$ is the optimal point of $F'(\mathbf{x})$.

Take $C = \sqrt{\frac{6}{nk}}$, we can ensure that $\left\|\sum_{i=1}^{\frac{n}{2}} \tilde{\mathbf{x}}_i\right\| \leq 1 = R$.

Now, we will bound $F_i'(\mathbf{x})$ at a point $\mathbf{x}$ such that there exists $q \in \{1, 2, \cdots \lfloor k/2 \rfloor\}$ satisfying $|\langle \mathbf{x}, \mathbf{v}_{i,q}\rangle| \leq \frac{c}{2}$. Define

$$F_i^t(\mathbf{x}) \coloneqq \frac{1}{32}\big(\left\langle x,\, \mathbf{v}_{i,0}\right\rangle^2 - 2C\left\langle x,\, \mathbf{v}_{i,0}\right\rangle + \sum_{r=1}^{t}(\left\langle x,\, \mathbf{v}_{i,r-1}\right\rangle - \left\langle x,\, \mathbf{v}_{i,r}\right\rangle)^2\big).$$

Observe that $F_i^t(\mathbf{x}) \leq F_i'(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^d$. Hence, we can find the minimum value of F_i^t under the condition that $\langle \mathbf{x}, \mathbf{v}_{i,t}\rangle$ is fixed as b_t. Similarly, By first order optimality conditions for F_i^t, its optimum point $\tilde{\mathbf{x}}_i^t$ must satisfy that

$$\begin{gathered}
2\langle \tilde{\mathbf{x}}_i^t, \mathbf{v}_{i,0}\rangle - \langle \tilde{\mathbf{x}}_i^t, \mathbf{v}_{i,1}\rangle = C; \\
\langle \tilde{\mathbf{x}}_i^t, \mathbf{v}_{i,j-1}\rangle - 2\langle \tilde{\mathbf{x}}_i^t, \mathbf{v}_{i,j}\rangle + \langle \tilde{\mathbf{x}}_i^t, \mathbf{v}_{i,j+1}\rangle = 0 \quad \text{for } 1 \leq j \leq q-1.
\end{gathered}$$

The solution to the above system is

$$\tilde{\mathbf{x}}_i^t = \sum_{j=0}^{t}\left(\frac{t-j}{t+1}C + \frac{j+1}{t+1}b_t\right)\mathbf{v}_{i,j}, \quad \text{where } b_t = \langle \mathbf{x}, \mathbf{v}_{i,t}\rangle,$$

and the corresponding minimal value is

$$F_i^t(\tilde{\mathbf{x}}_i^t) = \frac{1}{32}\left(-C^2 + \frac{(C-b_t)^2}{t+1}\right).$$

Consider the case that $|b_q| = |\langle \mathbf{x}, \mathbf{v}_{i,q}\rangle| \leq \frac{c}{2}$, take $c \leq (2-\sqrt{3})C$, we have

$$\begin{aligned}
F_i'(\mathbf{x}) &\geq F_i^t(\mathbf{x}) \geq F_i^t(\tilde{\mathbf{x}}_i^t) \\
&\geq \frac{1}{32}\left(-C^2 + \frac{(C-b_q)^2}{q+1}\right) \\
&\geq \frac{1}{32}\left(-C^2 + \frac{(C-\frac{c}{2})^2}{\frac{k}{2}+1}\right) \\
&\geq -\frac{C^2}{32}\left(1 - \frac{1}{\frac{2}{3}k + \frac{4}{3}}\right) \\
&\geq -\frac{C^2}{32}\left(1 - \frac{1}{\frac{2}{3}k + 1}\right).
\end{aligned} \tag{10}$$

Combine (9) and (10), when $k \geq 3$ and $|\langle \mathbf{x}, \mathbf{v}_{i,q} \rangle| \leq \frac{c}{2}$,

$$\begin{aligned} F_i(\mathbf{x}) - F_i(\mathbf{x}^*) &\geq F_i'(\mathbf{x}) - \frac{(k+1)c^2}{16} - F_i(\tilde{\mathbf{x}}_i) \\ &\geq F_i'(\mathbf{x}) - \frac{(k+1)c^2}{16} - F_i'(\tilde{\mathbf{x}}_i) \\ &\geq -\frac{C^2}{32}\left(1 - \frac{1}{\frac{2}{3}k+1}\right) + \frac{C^2}{32}\frac{k+1}{k+2} - \frac{(k+1)c^2}{16} \\ &\geq \frac{C^2}{32}\left(\frac{1}{\frac{2}{3}k+1} - \frac{1}{k+2}\right)\frac{(k+1)c^2}{16} \\ &\geq \frac{1}{32(k+1)^2 n} - \frac{(k+1)c^2}{16}. \end{aligned}$$

When $\epsilon < \frac{1}{4096n}$, setting $k = \lfloor \frac{1}{16\sqrt{\epsilon n}} \rfloor - 1 \geq 3$ and $c = \min\{\frac{1}{\sqrt{N}}, (2-\sqrt{3})C, 8\sqrt{\frac{\epsilon}{k}}\}$, we can ensure that

$$F_i(\mathbf{x}) - F_i(x^*) \geq 8\epsilon - 4\epsilon = 4\epsilon.$$

Therefore, if for at least $\frac{n}{4}$ of the i's it holds that $|\langle \mathbf{x},\, \mathbf{v}_{i,j_i} \rangle| < \frac{c}{2}$ for some $j_i \leq q = \lfloor \frac{k}{2} \rfloor$, then $\mathbf{x}$ cannot be an ϵ-optimal point of the total function $F(\mathbf{x})$. □

Proof of Corollary 4. Lemma 6 tells us that to find an ϵ-optimal point of the hard function defined in Definition 4, with parameters set by Lemma 6, we must output a vector $\mathbf{x}$ such that for at least $\frac{n}{4}$ of the i's it holds that $|\langle \mathbf{x},\, \mathbf{v}_{i,j} \rangle| > \frac{c}{2}$ for any $j \leq q = \lfloor \frac{k}{2} \rfloor$.

By Lemma 12 and Lemma 13, when d is sufficiently large, for a quantum algorithm finding $\mathbf{x}$ that satisfies the above requirements using s queries, we can construct an algorithm solving Problem 3 with input size $n \times q$ using $O\big((s + \frac{n}{2})\log n\big)$ queries. From Lemma 10, any quantum algorithms solving Problem 3 must take at least $\Omega(n\sqrt{q}) = \Omega(n\sqrt{k})$ queries. Therefore, we obtain a lower bound for case 2 of Problem 1:

$$s = \frac{1}{\log n} \cdot n\sqrt{k} = \Omega\left(n^{\frac{3}{4}} \frac{1}{\log n}\left(\frac{1}{\epsilon}\right)^{\frac{1}{4}}\right).$$

Take ℓ and R into account, we can simply scale the hard function as follows:

$$F'(\mathbf{x}) = \frac{1}{lR^2} F(\frac{\mathbf{x}}{R}).$$

Take $\epsilon' = \frac{\epsilon}{\ell R^2}$, we can finish the proof with a lower bound

$$\Omega\left(n^{\frac{3}{4}} \frac{1}{\log n}\left(\frac{\ell}{\epsilon}\right)^{\frac{1}{4}} R^{\frac{1}{2}}\right).$$

Next, we point out that $\Omega(n)$ is a trivial lower bound of Problem 1, as the quantum algorithm must take at least $\Omega(n)$ queries to get the values of vector of $\frac{n}{4}$ sub-function in orthogonal subspace. Taking into account the aforementioned discussion, our lower bound for the smooth and non-strongly convex is

$$\Omega\left(n + n^{\frac{3}{4}} \frac{1}{\log n}\left(\frac{\ell}{\epsilon}\right)^{\frac{1}{4}} R^{\frac{1}{2}}\right).$$

Finally, we verify the requirements to dimension d: $n(k+1) + \frac{8R^2}{c^2}\log(2nkN^3) = O\left(\left(\sqrt{\frac{n}{\epsilon}} + \frac{1}{\epsilon^2}\right)\log\left(\frac{n}{\epsilon}\right)\right)$, so by Lemma 12 we can take $d = \Omega\left(\left(\sqrt{\frac{n}{\epsilon}} + \frac{1}{\epsilon^2}\right)\log\left(\frac{n}{\epsilon}\right)\right)$.

□

C.5. Proofs for the Lipschitz and non-strongly convex setting

Proof of Lemma 7. The proof is inspired by the results from Appendix C.1 in Woodworth & Srebro (2016).

Without loss of generality, we can suppose that m is even. Otherwise we can simply set the last sub-function to 0, and the query complexity is reduced by a factor $\frac{m-1}{m}$. As in Definition 5, for values b, c and k to be fixed later, we define $\frac{n}{2}$ pairs of sub-functions:

$$f_{i,1}(\mathbf{x}) = \frac{1}{\sqrt{2}}|b - \langle \mathbf{x}, \mathbf{v}_{i,0}\rangle| + \frac{1}{2\sqrt{k}}\sum_{r \text{ even}}^{k} \chi_c\left(\langle \mathbf{x},\, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x},\, \mathbf{v}_{i,r}\rangle\right),$$

$$f_{i,2}(\mathbf{x}) = \frac{1}{2\sqrt{k}}\sum_{r \text{ odd}}^{k} \chi_c\left(\langle \mathbf{x},\, \mathbf{v}_{i,r-1}\rangle - \langle \mathbf{x},\, \mathbf{v}_{i,r}\rangle\right).$$

where $\mathbf{v}_{ij}$ are random orthonormal vectors on the unit sphere in $\mathbb{R}_d$.

For convenience, for $i \in \{1, 2, \cdots \frac{n}{2}\}$, we define

$$\begin{aligned} F_i(\mathbf{x}) &= \frac{1}{2}\left(f_{i,1}(\mathbf{x}) + f_{i,2}(\mathbf{x})\right) \\ &= \frac{1}{2\sqrt{2}}|b - \langle \mathbf{x}, \mathbf{v}_{i,0}\rangle| + \frac{1}{4\sqrt{k}}\sum_{j=1}^{k} \chi_c(\langle \mathbf{x}, \mathbf{v}_{i,j-1}\rangle - \langle \mathbf{x}, \mathbf{v}_{i,j}\rangle). \end{aligned}$$

It is straightforward to verify that $F_i(\mathbf{x})$ takes its minimum value 0 when $\langle \mathbf{x}, \mathbf{v}_{i,j}\rangle = b$ for all j. Since each sub-function is defined on orthogonal subspaces, the total function $F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{\frac{n}{2}}\left(f_{i,1}(\mathbf{x}) + f_{i,2}(\mathbf{x})\right)$ is minimized at point

$$\mathbf{x}^* = b \cdot \sum_{i=1}^{\frac{n}{2}}\sum_{j=0}^{k} \mathbf{v}_{i,j}$$

and $\mathbf{x}^*$ is also an optimal point of $F_i(\mathbf{x})$ for all i. We set $b = \sqrt{\frac{2}{n(k+1)}}$, such that $\|\mathbf{x}^*\| = 1 = R$.

Now we bound $F_i(\mathbf{x})$ at a point $\mathbf{x}$ when there exists $t \le k$ such that $\langle \mathbf{x}, \mathbf{v}_{i,t}\rangle \le \frac{c}{2}$.

$$\begin{aligned} F_i(\mathbf{x}) - F_i(\mathbf{x}^*) &\ge F_i(\mathbf{x}) - 0 \\ &\ge \frac{1}{2\sqrt{2}}|b - \langle \mathbf{x}, \mathbf{v}_{i,0}\rangle| + \frac{1}{4\sqrt{k}}\sum_{j=1}^{k}\left(|\langle \mathbf{x}, \mathbf{v}_{i,j-1}\rangle - \langle \mathbf{x}, \mathbf{v}_{i,j}\rangle| - c\right) \\ &= \frac{1}{2\sqrt{2}}|b - \langle \mathbf{x}, \mathbf{v}_{i,0}\rangle| + \frac{1}{4\sqrt{k}}\sum_{j=1}^{k}\left(|\langle \mathbf{x}, \mathbf{v}_{i,j-1}\rangle - \langle \mathbf{x}, \mathbf{v}_{i,j}\rangle|\right) - \frac{k}{4\sqrt{k}}c \\ &\ge \frac{1}{2\sqrt{2}}|b - \langle \mathbf{x}, \mathbf{v}_{i,0}\rangle| + \frac{1}{4\sqrt{k}}\sum_{j=1}^{t}\left(|\langle \mathbf{x}, \mathbf{v}_{i,j-1}\rangle - \langle \mathbf{x}, v_{i,j}\rangle|\right) - \frac{k}{4\sqrt{k}}c \\ &\ge \frac{1}{2\sqrt{2}}|b - \langle \mathbf{x}, \mathbf{v}_{i,0}\rangle| + \frac{1}{4\sqrt{k}}|\langle \mathbf{x}, \mathbf{v}_{i,0}\rangle - \langle \mathbf{x}, \mathbf{v}_{i,t}\rangle| - \frac{k}{4\sqrt{k}}c \\ &\ge \frac{1}{2\sqrt{2}}|b - \langle \mathbf{x}, \mathbf{v}_{i,0}\rangle| + \frac{1}{4\sqrt{k}}|\langle \mathbf{x}, \mathbf{v}_{i,0}\rangle| - \frac{1}{4\sqrt{k}}\frac{c}{2} - \frac{k}{4\sqrt{k}}c \\ &\ge -\frac{2k+1}{8\sqrt{k}}c + \min_{z\in\mathbb{R}}\left(\frac{1}{2\sqrt{2}}|b - z| + \frac{1}{4\sqrt{k}}|z|\right) \\ &= -\frac{2k+1}{8\sqrt{k}}c + \frac{b}{4\sqrt{k}} \\ &\ge -\frac{2k+1}{8\sqrt{k}}c + \frac{1}{4k\sqrt{n}}. \end{aligned}$$

Take $c = \min\{\frac{1}{\sqrt{N}}, \frac{\epsilon}{\sqrt{k}}\}$ and set $k = \lfloor \frac{1}{10\epsilon\sqrt{n}} \rfloor$, we have

$$F_i(\mathbf{x}) - F_i(\mathbf{x}^*) \geq -\frac{\epsilon}{2} + \frac{5}{2}\epsilon = 2\epsilon.$$

Therefore, if for at least $\frac{n}{4}$ i's that there exists i_j such that $\langle \mathbf{x}, \mathbf{v}_{i,i_j} \rangle < \frac{c}{2}$, then

$$F(\mathbf{x}) - F(\mathbf{x}^*) \geq \frac{2}{n} \cdot \frac{n}{4} \cdot 2\epsilon = \epsilon,$$

i.e., $\mathbf{x}$ cannot be an ϵ-optimal point of the hard instance F. □

Proof of Corollary 5. Lemma 7 informs us that to find an ϵ-optimal point of the hard function defined in Definition 5, with parameters set by Lemma 7, we must output a vector $\mathbf{x}$ such that for at least $\frac{n}{4}$ of the i's it holds that $|\langle \mathbf{x}, \mathbf{v}_{i,j} \rangle| > \frac{c}{2}$ for any j.

From Lemma 12 and Lemma 13, when d is sufficiently large, for a quantum algorithm finding $\mathbf{x}$ that satisfies the above requirements using s queries, we can construct an algorithm solving Problem 3 with input size $n \times k$ using $O\big((s + \frac{n}{2}) \log n\big)$ queries. Then we consider Lemma 10, any quantum algorithms solving Problem 3 must take at least $\Omega(n\sqrt{k})$ queries. This gives a lower bound for case 4 of Problem 1:

$$s = \frac{1}{\log n} \cdot n\sqrt{k} = \Omega\left(n^{\frac{3}{4}} \frac{1}{\log n} \left(\frac{1}{\epsilon}\right)^{\frac{1}{2}}\right).$$

Consider L and R, similarly we can scale the hard function as follows:

$$F'(\mathbf{x}) = \frac{1}{LR} F\left(\frac{\mathbf{x}}{R}\right).$$

Take $\epsilon' = \frac{\epsilon}{LR}$, we can finish the proof with a lower bound

$$\Omega\left(n + n^{\frac{3}{4}} \frac{1}{\log n} \left(\frac{LR}{\epsilon}\right)^{\frac{1}{2}}\right).$$

The part $\Omega(n)$ here is similar to Corollary 4.

Finally, we verify the requirements to dimension d: $n(k+1) + \frac{8R^2}{c^2} \log\left(2nkN^3\right) = O\left(\frac{1}{\epsilon^3\sqrt{n}} \log\left(\frac{n}{\epsilon}\right)\right)$, so by Lemma 12 we take $d = \Omega\left(\frac{1}{\epsilon^3\sqrt{n}} \log\left(\frac{n}{\epsilon}\right)\right)$. □

C.6. Proofs for the Lipschitz and strongly convex setting

Inspired by Appendix C.2 in Woodworth & Srebro (2016), we now use a reduction from case 4 of Problem 1, i.e., the Lipschitz and non-strongly convex setting to prove Corollary 6.

Proof of Corollary 6. We use proof by contradiction, assuming that there exists an algorithm A that can find a ϵ-optimal point of the total function in case 3 with $o\left(n + n^{\frac{3}{4}} \left(\frac{1}{\epsilon\mu}\right)^{\frac{1}{4}} L^{\frac{1}{2}} \frac{1}{\log n}\right)$ quantum queries.

For a function $F(\mathbf{x})$ that satisfies case 4, suppose $F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) + \psi(\mathbf{x})$, where $f_i(\mathbf{x})$ is convex and L-lipschitz, $\psi(\mathbf{x})$ is convex and the optimal point satisfies that $\|\mathbf{x}^*\| \leq R$. We construct an another function which can be minimized by the algorithm A:

$$\tilde{F}(\mathbf{x}) \coloneqq \frac{1}{n}\sum_{i=1}^{n} \tilde{f}_i(\mathbf{x}) + \tilde{\psi}(\mathbf{x}), \quad \text{where } \tilde{f}_i(\mathbf{x}) = f_i(\mathbf{x}) \text{ and } \tilde{\psi}(\mathbf{x}) = \psi(\mathbf{x}) + \frac{\mu}{2}\|\mathbf{x}\|^2.$$

Note that $\tilde{f}_i(\mathbf{x})$ is still convex and L-lipschitz and $\tilde{\psi}(\mathbf{x})$ is μ-strongly convex. Therefore, by assumption, A can find a $\frac{\epsilon}{2}$-optimal point $\hat{\mathbf{x}}$ of $\tilde{F}(\mathbf{x})$ with $o\left(n + n^{\frac{3}{4}}\left(\frac{1}{\epsilon\mu}\right)^{\frac{1}{4}} L^{\frac{1}{2}} \frac{1}{\log n}\right)$ quantum queries. Furthermore, set $\mu = \frac{\epsilon}{R^2}$, we have

$$F(\mathbf{x}) \leq \tilde{F}(\mathbf{x}) \leq F(\mathbf{x}) + \frac{\epsilon}{2R^2}\|\mathbf{x}\|^2 \leq F(\mathbf{x}) + \frac{\epsilon}{2}.$$

So

$$F(\hat{\mathbf{x}}) - F(\mathbf{x}^*) \leq \tilde{F}(\hat{\mathbf{x}}) - \tilde{F}(\mathbf{x}^*) + \frac{\epsilon}{2} \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon$$

which means that $\hat{\mathbf{x}}$ is an ϵ-optimal point of $F(\mathbf{x})$. The total number of queries is

$$o\left(n + n^{\frac{3}{4}}\left(\frac{1}{\epsilon\mu}\right)^{\frac{1}{4}} L^{\frac{1}{2}} \frac{1}{\log n}\right) = o\left(n + n^{\frac{3}{4}}\left(\frac{LR}{\epsilon}\right)^{\frac{1}{2}} \frac{1}{\log n}\right)$$

which contradicts the conclusion of Corollary 5. Here we need $\epsilon < \frac{9L^2}{200n\mu}$ and $d = \tilde{\Omega}\left(\frac{1}{\sqrt{\epsilon^3 n}}\right)$ to achieve the contradiction. □