Quantum Speedups of Optimizing Approximately Convex Functions with Applications to Logarithmic Regret Stochastic Convex Bandits

Tongyang Li
Advanced Institute of Information Technology,
Center on Frontiers of Computing Studies,
and School of Computer Science,
Peking University
tongyangli@pku.edu.cn

Ruizhe Zhang
Department of Computer Science
The University of Texas at Austin
ruizhe@utexas.edu

Abstract

We initiate the study of quantum algorithms for optimizing approximately convex functions. Given a convex set $\mathcal{K} \subseteq \mathbb{R}^n$ and a function $F\colon \mathbb{R}^n \to \mathbb{R}$ such that there exists a convex function $f\colon \mathcal{K} \to \mathbb{R}$ satisfying $\sup_{x\in\mathcal{K}} |F(x) - f(x)| \leq \epsilon/n$, our quantum algorithm finds an $x^* \in \mathcal{K}$ such that $F(x^*) - \min_{x\in\mathcal{K}} F(x) \leq \epsilon$ using $\tilde{O}(n^3)$ quantum evaluation queries to F. This achieves a polynomial quantum speedup compared to the best-known classical algorithms. As an application, we give a quantum algorithm for zeroth-order stochastic convex bandits with $\tilde{O}(n^5 \log^2 T)$ regret, an exponential speedup in T compared to the classical $\Omega(\sqrt{T})$ lower bound. Technically, we achieve quantum speedup in n by exploiting a quantum framework of simulated annealing and adopting a quantum version of the hit-and-run walk. Our speedup in T for zeroth-order stochastic convex bandits is due to a quadratic quantum speedup in multiplicative error of mean estimation.

1 Introduction

Optimization theory is a central research topic in computer science, mathematics, operations research, etc. Currently, many efficient algorithms for optimizing convex functions have been proposed (see for instance [10]), but much less is known for nonconvex optimization. In this paper, we investigate polynomial-time algorithms for optimizing approximately convex functions. On the one hand, such algorithms enjoy robustness and cover many natural scenarios including stochastic convex optimization, empirical risk minimization, etc. On the other hand, approximately convex optimization paves the way of understanding nonconvex optimization in the general case.

Specifically, let $\mathcal{K} \subseteq \mathbb{R}^n$ be a convex set. We call $F\colon \mathbb{R}^n \to \mathbb{R}$ as an *approximately convex function* over $\mathcal{K}$ if there is a convex function $f\colon \mathcal{K} \to \mathbb{R}$ such that

$$\sup_{x\in\mathcal{K}} |F(x) - f(x)| \leq \epsilon/n. \tag{1}$$

Throughout the paper, we assume that f is L-Lipschitz with respect to ℓ_∞ norm, i.e., $|f(x)-f(y)| \leq L\|x-y\|_\infty$ for any $x, y \in \mathcal{K}$. We assume that $\mathcal{B}_2(0,1) \subseteq \mathcal{K} \subseteq \mathcal{B}_2(0,R)$, i.e., the convex body $\mathcal{K}$ contains the unit ball centered at 0 and is contained by a ball of radius R centered at 0. Unless otherwise mentioned, we aim at algorithms with $\mathrm{poly}(\log R)$ dependence in R.

It is standard to assume the *zeroth-order oracle* of F, which returns the function value $F(x)$ given an input x. As far as we know, the state-of-the-art algorithm for finding an $x^* \in \mathcal{K}$ such that $F(x^*) -$

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

$\min_{x\in\mathcal{K}} F(x) \le \epsilon$ with high probability was proposed by Belloni et al. [8], which takes $\tilde{O}(n^{4.5})$[1] queries to the zeroth-order oracle. On the other hand, Ref. [34] proved that if the approximation error in (1) is at least $\tilde{\Omega}(\max\{\epsilon/n, \epsilon^2/\sqrt{n}\})$, there exists a function F which no algorithm can find a point $x^* \in \mathcal{K}$ such that $F(x^*) - \min_{x\in\mathcal{K}} F(x) \le \epsilon$ using $\mathrm{poly}(n, 1/\epsilon)$ queries to F. In other words, the ϵ/n term in (1) is fundamental, and it has been the standard assumption of studying approximately convex optimization in [8, 34].

Based on approximate convex functions, a closely related scenario is *stochastic convex functions*, where we have a function $F\colon \mathbb{R}^n \to \mathbb{R}$ such that

$$F(x) = f(x) + \epsilon_x.$$

Here $f\colon \mathbb{R}^n \to \mathbb{R}$ is a convex function and ϵ_x is a sub-Gaussian random variable with parameter σ, i.e., $\mathbb{E}[\exp(\lambda\epsilon_x)] \le \exp\big(\sigma^2\lambda^2/2\big)$ for any λ. This implies that

$$\Pr[|\epsilon_x| \ge \sigma t] \le 2\exp\big(-t^2/2\big) \quad \forall t \ge 0. \tag{2}$$

Using this fact, Belloni et al. [8] gave an algorithm for stochastic convex optimization with $\tilde{O}(n^{7.5}/\epsilon^2)$ queries to a zeroth-order oracle of F.

Contributions. In this paper, we conduct a systemic study of quantum algorithms for optimization of approximately convex functions, with applications to zeroth-order stochastic convex bandits. Quantum computing is a rapidly advancing technology, the capability of quantum computers is dramatically increasing and recently reached "quantum supremacy" by Google [6] and USTC [42]. In optimization theory, quantum advantages have been proven for semidefinite programs [3, 4, 11, 12], general convex optimization [5, 15], the escaping from saddle point problem in optimization [41], etc. Nevertheless, as far as we know, quantum algorithms for approximately convex optimization and stochastic convex optimization are widely open. In this paper, we consider these problems using the quantum zeroth-order evaluation oracle O_F, a standard model used in previous quantum computing literature [5, 14, 15, 41]:

$$O_F|x, y\rangle = |x, F(x) + y\rangle \quad \forall x \in \mathbb{R}^n, y \in \mathbb{R}. \tag{3}$$

Here $|\cdot\rangle$ is the Dirac notation, and preliminaries of quantum computing will be covered in Section 2. Intuitively, Eq. (3) can take inputs with form $\sum_{i=1}^m \mathbf{c}_i\,|\mathbf{x}_i\rangle \otimes |0\rangle$ where $\mathbf{x}_i \in \mathbb{R}^n\ \forall i \in [m]$ and $\sum_{i=1}^m |\mathbf{c}_i|^2 = 1$, and if we measure the outcome quantum state, we get $F(\mathbf{x}_i)$ with probability $|\mathbf{c}_i|^2$. In particular, the quantum zeroth-order oracle allows the ability to query different locations in *superposition*, which is stronger than the classical counterpart (i.e., $m = 1$). Nevertheless, if the classical zeroth-order oracle can be implemented by explicit arithmetic circuits, the quantum oracle in (3) can be implemented by quantum circuits of the same size up to logarithmic factors. As for stochastic convex functions, similar to (3), we assume the following oracle:

$$O_f|x, y\rangle = |x\rangle \int_{\xi\in\mathbb{R}} \sqrt{g_x(\xi)}|f(x) + y + \xi\rangle\,\mathrm{d}\xi \quad \forall x \in \mathbb{R}^n, y \in \mathbb{R}, \tag{4}$$

where for any $x \in \mathbb{R}^n$, $g_x(\xi)$ follows a sub-Gaussian distribution as in (2).

Our first result is a quantum algorithm for optimizing approximately convex functions:

Theorem 1. *With probability at least* 0.9*, we can find an* $x^* \in \mathcal{K}$ *such that*

$$F(x^*) - \min_{x\in\mathcal{K}} F(x) \le \epsilon \tag{5}$$

using $\tilde{O}(n^3)$ *queries to the quantum evaluation oracle (3).*

We remark that the succss probability can be easily boosted up to $1-\delta$ for any $\delta \in (0, 1)$, by paying an extra $\log(1/\delta)$ factor in the quantum query complexity. Compared to the best-known classical result by Belloni et al. [8] with query complexity $\tilde{O}(n^{4.5})$, we achieve a polynomial quantum speedup in terms of n. Technically, Belloni et al.'s algorithm is based on the simulated annealing process and using the Hit-and-Run walk to generate samples in each stage. In this work, we give a *user-friendly*

[1]The $\tilde{O}$ and $\tilde{\Omega}$ notation omits poly-logarithmic terms, i.e., $\tilde{O}(g) = O(g\,\mathrm{poly}(\log g))$ and $\tilde{\Omega}(g) = \Omega(g\,\mathrm{poly}(\log g))$. Unless otherwise mentioned, both notations also omit $\mathrm{poly}(\log 1/\epsilon)$ terms.

version of the quantum walk framework (Theorem 3) that can be applied very easily to obtain quantum speedup for the mixing of classical Markov chains. We then analyze the warmness and the overlap between adjacent Hit-and-Run walks in the simulated annealing process, showing that our quantum walk framework is applicable to each classical Hit-and-Run sampler. By implementing the random walk quantumly, we improve the query complexity of the sampling procedure from $\widetilde{O}(n^3)$ to $\widetilde{O}(n^{1.5})$ (Theorem 4). We also design a non-destructive rounding procedure (Lemma 4) in each stage of the simulated annealing that simulates the classical rounding procedure in [8] but will not destroy the quantum states of the quantum walks. Combining them together gives the $\widetilde{O}(n^3)$-query quantum algorithm for minimizing an approximately convex function. Our result can also be applied to give an $\widetilde{O}(n^3)$-query quantum algorithm for optimizing approximately convex functions with decreasing fluctuations.

See Section 3 for more details and the proof of Theorem 1.

Furthermore, to estimate the mean of a random variable up to multiplicative error ϵ, classical algorithms need to take $1/\epsilon^2$ samples due to concentration inequalities such as Chernoff's bound, whereas quantum algorithms can take roughly $1/\epsilon$ queries (see Proposition 1). As a result, we obtain polynomial quantum speedup in both n and $1/\epsilon$ for stochastic convex optimization:

Corollary 1. *With probability at least* 0.8*, we can find an* $x^* \in \mathcal{K}$ *such that*

$$f(x^*) - \min_{x\in\mathcal{K}} f(x) \le \epsilon \tag{6}$$

using $\tilde{O}(n^5/\epsilon)$ *queries to the quantum stochastic evaluation oracle (4).*

We apply Corollary 1 to solve the zeroth-order stochastic convex bandit problem, which is a widely studied bandit model (see e.g., [22, 24, 25]). The problem is defined as follows. Let $\mathcal{K} \subseteq \mathbb{R}^n$ be a convex body and $f\colon \mathcal{K} \to [0,1]$ be a convex function. Here $\mathcal{B}_2(0,1) \subseteq \mathcal{K} \subseteq \mathcal{B}_2(0,R)$. An online learner and environment interact alternatively over T rounds. In each round $t \in [T]$, the learner makes a query to the quantum stochastic evaluation oracle (4), and returns a value $x_t \in \mathcal{K}$ as the current guess. The learner aims to minimize the regret

$$\mathcal{R}_T := \mathbb{E}\left[\sum_{t=1}^{T}(f(x_t) - f^*)\right], \quad \text{where } f^* = \min_{x\in\mathcal{K}} f(x), \tag{7}$$

and the expectation is taken over all randomness. The classical state-of-the-art algorithm using a classical stochastic evaluation in each round achieves a regret bound of $\tilde{O}(n^{4.5}\sqrt{T})$ [25], and there is a classical lower bound $\Omega(n\sqrt{T})$ on the regret [19]. Here we prove:

Theorem 2. *There is a quantum algorithm for which* $\mathcal{R}_T = \tilde{O}(n^5 \log(T)\log(TR))$.

This achieves $\mathrm{poly}(\log T)$ regret for zeroth-order stochastic convex bandits, **an exponential quantum advantage** in terms of T compared to classical zeroth-order stochastic convex bandits. As far as we know, we give the first quantum algorithms with poly-logarithmic regret bound on online learning problems. An independent work [37] gave quantum algorithms with poly-logarithmic regret for multi-armed bandits and stochastic linear bandits, but these two types of bandits concern reward of discrete objects and linear functions, respectively, which are fundamentally different from the stochastic convex bandits we study.

To achieve this $\mathrm{poly}(\log T)$ regret for zeroth-order stochastic convex bandits, we divide the T iterations into $\lfloor \log_2 T \rfloor$ intervals with doubling length $1, 2, 4, \ldots$. We use the quantum queries from a previous interval to run our quantum stochastic convex optimization algorithm and use the output as the guess for all iterations in the next interval. Since we can achieve linear dependence in $1/\epsilon$ in Corollary 1, it can be calculated that the total regret in each iteration is at most $\tilde{O}(n^5 \log(TR))$, leading to our claim. The proof details are given in Section 4.

Open questions. Our work raises several natural questions for future investigation:

- Can we further improve the dimension dependence of our approximate convex optimization algorithm? The current quantum speedup mainly leverages the quantum hit-and-run walk; it is of general interest to understand whether we can gain quantum advantage at other steps, or whether the convergence analysis of the hit-and-run walk per se can be improved.

- Can we improve the dimension dependence of the regret of our zeroth-order stochastic convex bandits? It is natural to check whether our n^5 term can be improved by quantizing the classical state-of-the-art [25] and other recent works.
- Can we give fast quantum algorithms for more general nonconvex optimization problems, or with poly-logarithmic regret for more general bandit problems?

Limitations and societal impacts. This work is purely theoretical. Researchers working on theoretical aspects of quantum computing and optimization theory may benefit from our results at the moment. In the long term, once we have fault-tolerant quantum computers, our results can be applied to approximate convex optimization and stochastic bandit scenarios arising in the real world. As far as we are aware, our work does not have negative societal impacts.

2 Preliminaries

2.1 Basics of quantum computing

We briefly introduce basic notations and concepts of quantum computing in this section. More details are covered in standard textbooks, for instance [30].

Quantum computing can be formulated in terms of linear algebra. Specifically, the computational basis of space $\mathbb{C}^d$ can be denoted by $\{\vec{e}_1, \ldots, \vec{e}_d\}$, where $\vec{e}_i = (0, \ldots, 1, \ldots, 0)^\top$ with the ith entry being 1 and other entries being 0. These basis vectors are typically written by the *Dirac notation*, where we denote $\vec{e}_i$ by $|i\rangle$ (called a "ket"), and denote $\vec{e}_i^\top$ by $\langle i|$ (called a "bra").

A d-dimensional *quantum state* is an ℓ_2-norm unit vector in $\mathbb{C}^d$, e.g., $|v\rangle = (v_1, \ldots, v_d)^\top$ such that $\sum_i |v_i|^2 = 1$. For each i, v_i is called the *amplitude* in $|i\rangle$. The *tensor product* of quantum states is their Kronecker product: if $|u\rangle \in \mathbb{C}^{d_1}$ and $|v\rangle \in \mathbb{C}^{d_2}$, then

$$|u\rangle \otimes |v\rangle := (u_1 v_1, u_1 v_2, \ldots, u_{d_1} v_{d_2})^\top \in \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}.$$

We often omit the operator $\otimes$ and simply write $|u\rangle\,|v\rangle$ or $|u, v\rangle$ for being concise.

In general, the definition of quantum states can be extended to a continuous domain. For instance,

$$|v\rangle = \int_{\mathbb{R}^n} v_x |x\rangle \mathrm{d}x$$

represents a quantum state as long as $\int_{\mathbb{R}^n} |v_x|^2 \mathrm{d}x = 1$. To keep quantum states normalized in ℓ_2 norm, operations in quantum computing are *unitary transformations*.

2.2 Classical and quantum walks

A classical Markov over the space Ω is a sequence of random variables $\{X_i\}_{i\in\mathbb{N}}$ such that for any $i > 0$ and $x_0, \ldots, x_i \in \Omega$,

$$\Pr[X_i = x_i \mid X_0 = x_0, \cdots, X_{i-1} = x_{i-1}] = \Pr[X_i = x_i \mid X_{i-1} = x_{i-1}].$$

The Markov chain can be represented by its stochastic transition matrix P such that $\sum_{y\in\Omega} P(x, y) = 1$ for any $x \in \Omega$. A distribution π is *stationary* if it satisfies $\sum_{x\in\Omega} \pi(x)P(x, y) = \pi(y)$ for any $y \in \Omega$. A Markov chain is *reversible* if its stationary distribution satisfies the following detailed balance condition:

$$\pi(x)P(x, y) = \pi(y)P(y, x) \quad \forall x, y \in \Omega.$$

The *mixing time* of a Markov chain with initial distribution π_0 is the number of steps $t = t(\epsilon) \in \mathbb{N}$ such that the total variation distance between the time-t distribution and the stationary distribution is at most ϵ for any $\epsilon \in (0, 1)$, i.e.,

$$d_{\mathrm{TV}}(P^t \pi_0, \pi) \le \epsilon.$$

In quantum, we can also define the discrete-time quantum walk [36], which is a quantum-analogue of classical Markov chain. More precisely, we use a quantum state to represent a classical probability distribution:

$$\{\pi(x)\}_{x\in\Omega} \longleftrightarrow |\pi\rangle = \sum_{x\in\Omega} \sqrt{\pi(x)}\,|x\rangle\,.$$

In quantum walk, there are two operations:

- **Reflect:** An operator R that reflects the quantum states with respect to the subspace $\mathrm{span}\{|x\rangle |\psi_x\rangle\}_{x\in\Omega}$, where $|\psi_x\rangle = \sum_{y\in\Omega} \sqrt{P(x,y)}\, |y\rangle$.
- **Swap:** An operator S that swap the two quantum registers: $S\,|x\rangle\,|y\rangle = |y\rangle\,|x\rangle$.

Then, the quantum walk operator W is defined as: $W := S \circ R$. Intuitively, the first quantum register contains the current position of the random walk, and the second register contains the previous position. In each step of quantum walk, the R operator makes a superposition in the second register of the next step positions with amplitudes proportional to their transition probabilities. Then, the S operator swaps the two quantum registers, using the first register to store the new positions and the second register to store the old position. In this way, we complete one-step of the random walk coherently (in superposition).

The quantum advantage of the quantum walk comes from the spectrum of W. Note that the state of stationary distribution $\sum_{x\in\Omega} \sqrt{\pi(x)}\, |x\rangle\, |\psi_x\rangle$ is invariant under W. In other words, it is an eigenvector with eigenvalue 1 (eigenphase 0). On the other hand, the other eigenvectors of W has eigenphase at least $\sqrt{2\delta}$, where δ is the spectral gap[2] of the transition matrix P. Therefore, applying the *quantum phase estimation* algorithm using $O(1/\sqrt{\delta})$ calls to W can distinguish the state corresponding to the stationary distribution and other eigenstates. See [17, 36, 38] for more details.

2.3 Hit-and-Run walk

Hit-and-Run walk was introduced by R.L. Smith [35], and has a long line of reserach [1, 7, 9, 16, 26–29, 40] for its mixing time and applications in sampling, optimization, and volume estimation. Intuitively, the Hit-and-Run walk is defined as follows:

1. Pick a uniformly distributed random line ℓ through the current point x.
2. Move to a random point y along the line chosen from the restricted distribution π_ℓ.

More specifically, let $f : \mathbb{R}^n \to \mathbb{R}$ be a logconcave distribution density. Then, the distribution induced by restricting f to the line ℓ is defined as follows:

$$\pi_\ell(S) := \frac{\int_S f(x)\mathrm{d}x}{\int_\ell f(x)\mathrm{d}x} \quad \forall S \subset \ell.$$

The following lemma show the transition probability of the Hit-and-Run walk.

Lemma 1 ([26])**.** *Let f be the density of a logconcave distribution. If the current point of Hit-and-Run is u, then the density function of the distribution of the next point x is*

$$f_u(x) = \frac{2}{n\pi_n} \frac{f(x)}{\mu_f(u,x)\|x-u\|^{n-1}},$$

where $\pi_n = \frac{\pi^{n/2}}{\Gamma(1+n/2)}$ and $\mu_f(u,x)$ is the f-measure of the chord through u and x.

3 Quantum Algorithm for Optimizing Approximately Convex Functions

We give a polynomial quantum speedup for minimizing an approximately convex functions using the quantum walk algorithm [36, 38]. More specifically, we consider the quantum walk in continuous space.[3] Let P denote the (column) stochastic transition density of a reversible Markov chain, i.e.,

$$\int_{\mathbb{R}^n} P(x,y)\mathrm{d}y = 1 \quad \forall x \in \mathbb{R}^n,$$

and let $\pi(x)$ denote the density of the stationary distribution. Then, we can implement a quantum walk unitary $W(P)$ such that its unique eigenvector with eigenvalue 1 (or equivalently eigenphase 0) is:

$$\int_{\mathbb{R}^n} \sqrt{\pi(x)}\, |x\rangle \otimes |\psi_x\rangle\, \mathrm{d}x,$$

[2] The difference between the first and the second largest eigenvalues.

[3] It can be naturally discretized as we do in simulating a Markov chain on classical digital computers.

where $|\psi_x\rangle := \int_{\mathbb{R}^n} \sqrt{P(x,y)}\,|y\rangle\,\mathrm{d}y$ mixes all the points that can be moved from x, with amplitudes proportional to the transition probabilities. This quantum state can be considered as a *coherent encoding* of the classical distribution π. The advantage of quantum walk comes from the fact that $W(P)$ has phase gap $\sqrt{\delta}$, where δ is the spectral gap of P. Therefore, by the quantum phase estimation algorithm [23], quantum walk can achieve quadratic speedup in δ. In general, quantum walk algorithm can quadratically speedup the hitting time of classical Markov chain. For the mixing time, it requires some additional complicated constraints on the Markov chain and distributions (see e.g., [2, 13, 31, 39]). Based on previous studies on quantum walk mixing, we propose the following user-friendly quantum walk framework:

Theorem 3 (User-friendly quantum walk framework)**.** *Let M_0 be the initial Markov chain with stationary distribution π_0, M_1 be the target Markov chain with stationary distribution π_1. Suppose M_0, M_1 satisfy the following properties:*

- **Mixing time:** $d_{\mathrm{TV}}(P_1^{t_0}\cdot\pi_0,\pi_1)\le\epsilon$ *and* $d_{\mathrm{TV}}(P_0^{t_1}\cdot\pi_1,\pi_0)\le\epsilon$.
- **Warmness:** $\|\pi_0/\pi_1\|=O(1)$ *and* $\|\pi_1/\pi_0\|=O(1)$, *where* $\|\pi/\sigma\|:=\int_{\mathbb{R}^n}\frac{\pi(x)}{\sigma(x)}\pi(x)\mathrm{d}x$.
- **Overlap:** $|\langle\pi_0|\pi_1\rangle|=\int_{\mathbb{R}^n}\sqrt{\pi_0(x)\pi_1(x)}\mathrm{d}x=\Omega(1)$.

Furthermore, suppose we have access to a unitary U that prepares the initial state $|\pi_0\rangle = \int_{\mathbb{R}^n}\sqrt{\pi_0(x)}\,|x\rangle\,\mathrm{d}x$. Then, we can obtain a state $|\widetilde{\pi}_1\rangle$ with $\|\,|\widetilde{\pi}_1\rangle-|\pi_1\rangle\,\|_2\le\epsilon$ using

$$O\left(\sqrt{t_0+t_1}\log^2(1/\epsilon)\right)$$

calls to the quantum walk operators.

More details and proofs are deferred to Appendix A.

Next, we can use Theorem 3 to speed-up the best-known classical algorithm for optimizing approximately convex functions [8], which has the following three levels:

- **High level:** Perform a simulated annealing with K stages. At the i-th stage, the target distribution π_{g_i} has density $\propto g_i(x)=e^{-F(x)/T_i}$, where $T_i:=(1-1/\sqrt{n})^i$.
- **Middle level:** Use N samples from π_{g_i} to construct a linear transformation Σ_i, rounding the distribution to near-isotropic position.
- **Low level:** Run the hit-and-run walk to evolve the distribution from $\pi_{g_{i-1}}$ to π_{g_i}.

Here, each step of the hit-and-run walk picks a uniformly random direction at the current point, and then walks on the 1-dimensional chord intersected by the direction and $\mathcal{K}$ with probability density proportional to the logconcave density. We formally state the hit-and-run walk in Algorithm 3.

We focus on speeding-up the Low level using Theorem 3. Hence, we need to show that π_{g_i} and $\pi_{g_{i+1}}$ satisfy the properties therein. First of all, it has been proved in [8] that $\|\pi_{g_i}/\pi_{g_{i+1}}\|=O(1)$ (Lemma 11). We prove the following lemma with proof deferred to Appendix C.1.

Lemma 2 (Informal version of Lemma 12)**.** *Let π_{g_i} be a distribution with density proportional to $g_i(x)=\exp(-F(x)/T_i)$, where $F(x)$ is β-approximately convex. Then, for any $0\le i\le K-1$,*

$$\|\pi_{g_{i+1}}/\pi_{g_i}\|\le 8\exp(2\beta/T_{i+1})\le O(1).$$

Therefore, the warmness property is satisfied.

We also prove that the Markov chains in this annealing schedule are slowly evolving:

Lemma 3 (Informal version of Lemma 13)**.** *Let π_{g_i} be a distribution with density proportional to $g_i(x)=\exp(-F(x)/T_i)$, where $F(x)$ is β-approximately convex. Then, for any $0\le i\le K-1$,*

$$|\langle\pi_i|\pi_{i+1}\rangle|\ge\exp(-(\beta/T_{i+1}+1)/2)=\Omega(1).$$

Hence, the overlap property is also satisfied.

With the warmness and classical analysis of the hit-and-run walk (Theorem 7), we get that the classical mixing time from π_{g_i} to $\pi_{g_{i+1}}$ and vice versa can be bounded by $\widetilde{O}(n^3)$.

Moreover, we also show in Appendix C.1 and Lemma 14 that each call to the quantum walk operator can be implemented by querying the evaluation oracle $O(1)$ times.

Thus, by Theorem 3, we get the following theorem:

Theorem 4 (Low level quantum speedup, informal version of Theorem 11)**.** *Let $\gamma \in (0, 1/e)$. Let $g_i(x) = \exp(-F(x)/T_i)$ be the density of π_{g_i} with $F(x)$ being $\beta/2$-approximately convex. Let $T_i = (1 - 1/\sqrt{n})^i$ for $0 \le i \le K$. Then, for each $0 \le i \le K-1$, given a state $|\pi_{g_i}\rangle$, we can produce a state $|\hat{\sigma}_i^{(m)}\rangle$ such that*

$$\||\pi_{g_{i+1}}\rangle - |\hat{\sigma}_i^{(m)}\rangle\|_2 \le O(\gamma),$$

using $m = \widetilde{O}(n^{1.5})$ calls for the evaluation oracle of F.

In the Middle level, we need to use N independent samples from π_{g_i} to construct a linear transformation Σ_i for rounding. However, we cannot directly measure the state $|\pi_{g_i}\rangle$, since it will destroy the quantum coherence. Instead, we propose a non-destructive approach to construct the linear transformation (the proof is deferred to Appendix C.2):

Lemma 4 (Non-destructive rounding, informal version of Lemma 15)**.** *For each $i \in [K]$, the linear transformation Σ_i can be obtained using $\widetilde{O}(N)$ copies of the states $\left|\pi_{g_{i-1}}\right\rangle$, with query complexity $\widetilde{O}(N \cdot n^{1.5})$. Moreover, the states $\left|\pi_{g_{i-1}}\right\rangle$ will be recovered with high probability.*

Now, we can put all the components together and obtain a quantum algorithm for optimizing approximately convex function with $\widetilde{O}(n^{1.5})$ quantum query complexity (Algorithm 1). We sketch the proof in below and the formal proof is given in Appendix C.3:

Proof sketch of Theorem 1. Observe that at each annealing stage, the sample distribution is the same as the classical algorithm. Thus, by the classical analysis (Theorem 9) in [8], the same optimization guarantee still holds for the quantum algorithm. Thus, if we take $K = \sqrt{n}\log(n/\epsilon)$ and $N = \widetilde{O}(n)$, the output x_* of QSIMANNEALING procedure satisfies:

$$F(x_*) - \min_x F(x) \le O(\epsilon)$$

with high probability.

Then, consider the query complexity. There are K stages in the annealing process, where each stage maintains $\widetilde{O}(N)$ samples (quantum states). By Theorem 4, evolving each state takes $\widetilde{O}(n^{1.5})$ queries. Therefore, the total query complexity is

$$K \cdot N \cdot \widetilde{O}(n^{1.5}) = \widetilde{O}(n^3).$$

Here, we assume that the convex body $\mathcal{K}$ is known, e.g., $\mathcal{K} = \mathbb{R}^n$ or $\mathbb{S}^n$. However, even if $\mathcal{K}$ is unknown, we can call its membership oralce to run our algorithm. More specifically, in constructing the initial state $|\pi_0\rangle$, we need to query the membership oracle for $\widetilde{O}(1)$ times. And since we prepare N copies, this step takes $\widetilde{O}(N)$ queries in total. Then, in each step of the quantum walk, we need to query the membership oracle for $\widetilde{O}(1)$ times to determine the intersection point for the hit-and-run process. Thus, the number of queries to the membership oracle of $\mathcal{K}$ is the same as the the number of evaluation oracle queries. Hence, our algorithm will query the membership oracle for $\widetilde{O}(n^3)$ times in all.

As for the number of qubits to implement our algorithm, each state $|\pi_0\rangle$ uses $\widetilde{O}(n)$ qubits. Thus, $\widetilde{O}(n^2)$ qubits are used to store all states. And we need $O(n)$ ancilla qubits for the quantum walk unitaries. Therefore, our algorithm uses $\widetilde{O}(n^2)$ qubits in total. □

Optimization of approximately convex functions with decreasing fluctuations. Beyond the ℓ_∞-norm assumption for all $x \in \mathcal{K}$ in Eq. (1), it is also possible to give efficient algorithms for optimizing other types of approximately convex functions. Specifically, Belloni et al. [8, Section 7] studied approximately convex functions with decreasing fluctuations.

Algorithm 1 Quantum speedup for approximately convex optimization (Informal version)

procedure QSIMANNEALING ▷ Theorem 1
  $N \leftarrow \widetilde{O}(n)$, $K \leftarrow \sqrt{n}\log(n/\epsilon)$
  Prepare N (approximately) copies of $|\pi_0\rangle$, denoted as $|\widetilde{\pi}_0^{(1)}\rangle, \ldots, |\widetilde{\pi}_0^{(N)}\rangle$
  for $i \leftarrow 1, \ldots, K$ **do**
    Use $\{|\widetilde{\pi}_0^{(j)}\rangle\}_{j\in[N]}$ to nondestructively construct Σ_i ▷ Lemma 4
    Apply quantum walk to evolve the states $|\widetilde{\pi}_{i-1}^{(j)}\rangle$ to $|\widetilde{\pi}_i^{(j)}\rangle$ ▷ Theorem 4
  end for
  $x_K^j \leftarrow$ measure the final state $|\widetilde{\pi}_K^{(j)}\rangle$ for $j \in [N]$
  return $\arg\min_{j\in[N]} F(x_K^j)$
end procedure

Suppose that the function f in (1) is 1-Lipschtiz and α-strongly convex with minimum at $x_{\min} \in \mathcal{K}$:

$$f(x) - f(x_{\min}) \geq \langle \nabla f(x_{\min}), x - x_{\min}\rangle + \tfrac{\alpha}{2}\|x - x_{\min}\|^2 \geq \tfrac{\alpha}{2}\|x - x_{\min}\|^2.$$

Define a measure of "non-convexity" of F w.r.t to f in an n-dimensional ball of radius r near $x_{\min}$:

$$\Delta(r) := \sup_{x\in\mathcal{B}_2(x_{\min},r)} |F(x) - f(x)|.$$

We can call Theorem 1 iteratively. Suppose that at the start of the tth iteration we have a ball $\mathcal{B}_2(x_{t-1}, 2r_{t-1})$ satisfying

$$\mathcal{B}_2(x_{\min}, r_{t-1}) \subset \mathcal{B}_2(x_{t-1}, 2r_{t-1}) \subset \mathcal{B}_2(x_{\min}, 3r_{t-1}).$$

After executing Theorem 1 in this iteration with $\widetilde{O}(n^3)$ quantum queries, we reach a point x_t such that with high probability

$$f(x_t) - f(x_{\min}) \leq Cn\Delta(3r_{t-1}))$$

for some global constant $C > 0$. Due to strong convexity, this gives a new radius r_t recursively:

$$\frac{\alpha}{2Cn} r_t^2 := \frac{\alpha}{2Cn}\|x_t - x_{\min}\|^2 \leq \Delta(3r_{t-1}).$$

When $\Delta(r) = cr^p$ for some $c > 0, p \in (0, 2)$, the iteration stops when

$$r^* = (2 \cdot 3^p cCn/\alpha)^{1/(2-p)}$$

, and when

$$\Delta(r) = c\log(1 + dr)$$

for some $c, d > 0$, the iteration stops at r^* satisfying

$$(2cCn/\alpha)\log(1 + 3dr^*) = (r^*)^2.$$

The total number of quantum queries is still $\widetilde{O}(n^3)$.

4 Quantum Algorithm for Zeroth-Order Stochastic Convex Bandits

We first prove Corollary 1 using quantum mean estimation with a Gaussian tail:

Proposition 1 (Adapted from [20, Theorem 4.2])**.** *Suppose that X is a random variable on a probability space (Ω, p) with mean μ and variance σ^2. Suppose we have a unitary oracle U satisfying $U|0\rangle = \int_{x\in\Omega}\sqrt{p(x)}|x\rangle \mathrm{d}x$. Then, for any $\Delta \in (0, 1)$ and $\tau \in \mathbb{N}$ such that $\tau \geq \log(1/\Delta)$, there is a quantum algorithm that outputs a mean estimate $\tilde{\mu}$ such that*

$$\Pr\left[|\tilde{\mu} - \mu| > \frac{\sigma\log(1/\Delta)}{\tau}\right] \leq \Delta,$$

using $O(\tau\log^{3/2}(\tau)\log\log(\tau))$ queries to U.

Proof of Corollary 1. We follow Section 6 of [8] while use Theorem 1 and Proposition 1. Specifically, for a parameter $0 < \alpha < 1$, we let $\mathcal{N}_\alpha$ be a box grid of $\mathcal{K}$ with side length α. In other words, $\mathcal{N}_\alpha$ is α-net of $\mathcal{K}$ in ℓ_∞ norm. Since $\mathcal{K} \subseteq \mathcal{B}_2(0, R)$, $|\mathcal{N}_\alpha| \le (R/\alpha)^n$.

Note that for the sub-Gaussian random variable ϵ_x in (2), it has variance at most $4\sigma^2$ because

$$\mathbb{E}[|\epsilon_x|^2] = \int_0^\infty \Pr\big[|\epsilon_x| > \sqrt{s}\big]\,\mathrm{d}s \le 2\int_0^\infty e^{-\frac{s}{2\sigma^2}}\,\mathrm{d}s = 4\sigma^2.$$

Upon a query $x' \in \mathcal{K}$, we define an oracle $O_f^{\tau,\alpha}$ which returns $f(x) + \tilde{\epsilon}_x$ for $x \in \mathcal{N}_\alpha$ which is closest to x', and the $\tilde{\epsilon}_x$ here is obtained by applying Proposition 1 with the unitary oracle O_f in Eq. (4) to estimate the function value $f(x)$. Specifically, with $\Delta = \exp\big(-t^2\big)$ where t is a parameter determined later, we have

$$\Pr\left[|\tilde{\epsilon}_x| > \frac{\sigma t^2}{\tau}\right] \le \exp\big(-t^2\big). \tag{8}$$

We note that in our algorithm based on the hit-and-run walk, with probability 1 we do not revisit the same point. As a result, $O_f^{\tau,\alpha}$ is no more powerful than O_f since the learner only obtains information on $\mathcal{N}_\alpha$, and in the rest of the proof we assume $O_f^{\tau,\alpha}$ as the oracle we use. We take

$$\alpha = \epsilon/2nL, \quad t = \sqrt{n\ln(R/\alpha) + \ln 10}.$$

Note that the value of t promises that $\exp\big(-t^2\big)(R/\alpha)^n \le 0.1$. In other words, with probability at least 0.9, we promise that

$$\max_{x\in\mathcal{N}_\alpha} |\tilde{\epsilon}_x| \le \frac{\sigma t^2}{\tau} = \frac{\sigma(n\ln(R/\alpha) + \ln 10)}{\tau}. \tag{9}$$

Finally, we take τ such that the RHS of (9) equals to $\epsilon/2n$, which is equivalent to

$$\tau = \frac{2n\sigma(n\ln(R/\alpha) + \ln 10)}{\epsilon} = \tilde{O}(n^2/\epsilon).$$

This will finally promise that

$$\sup_{x\in\mathcal{K}} |F(x) - f(x)| \le \max_{x\in\mathcal{N}_\alpha} |\tilde{\epsilon}_x| + \alpha L \le \frac{\epsilon}{2n} + \frac{\epsilon}{2n} = \frac{\epsilon}{n}, \tag{10}$$

meeting the condition of Theorem 1. Consequently, with probability at least $0.9 \cdot 0.9 > 0.8$, we can find an $x^* \in \mathcal{K}$ such that $f(x^*) - \min_{x\in\mathcal{K}} f(x) \le \epsilon$ using $\tilde{O}(n^3) \cdot \tau = \tilde{O}(n^5/\epsilon)$ queries to the quantum stochastic evaluation oracle (4). □

Proof of Theorem 2. We prove that Algorithm 2 satisfies Theorem 2.

Intuitively, we divide the T rounds into $m+1$ intervals where $m \leftarrow \lfloor \log_2 T \rfloor$, such that $[T] = \bigcup_{i=1}^{m+1} \mathcal{T}_i$ and $\mathcal{T}_i := \{2^{i-1}, 2^{i-1}+1, \ldots, 2^i - 1\}$ for each $i \in [m]$. When executing in the interval $\mathcal{T}_i$, the output required by the online learner is always $x_t = x_{2^{i-1}}$, the x at the end of the last interval. On the other hand, the queries in the current interval are applied to running the quantum stochastic convex optimization algorithm in Corollary 1 and output a nearly-optimal solution with probability at least $1 - O(1/T)$. With $|\mathcal{T}_i| = 2^{i-1}$ queries at hand, we divide them into $\log(TR)$ repeats of Corollary 1, each using $2^{i-1}/\log(TR)$ queries in the quantum algorithm. As a result, each repeat $j \in [\log(TR)]$ outputs a value $\tilde{x}_{2^i,j}$ such that

$$f(\tilde{x}_{2^i,j}) - \min_{x\in\mathcal{K}} f(x) \le \tilde{O}(n^5 \log(TR)/2^{i-1})$$

with probability at least 0.8. We take $x_{2^i} := \arg\min_{j\in[\log T]} f(\tilde{x}_{2^i,j})$. With probability at least $1 - 0.8^{\log(TR)} = 1 - O(1/TR)$, we have

$$f(x_{2^i}) - \min_{x\in\mathcal{K}} f(x) \le \tilde{O}(n^5 \log(TR)/2^i). \tag{11}$$

Going through all $i \in [m+1]$ intervals, by the union bound, with probability at least

$$1 - (m+1)\cdot O\Big(\frac{1}{TR}\Big) = 1 - O\Big(\frac{\log T}{TR}\Big),$$

Algorithm 2 Quantum zeroth-order stochastic convex bandits

procedure QBANDITS(T)
  $m \leftarrow \lfloor \log_2 T \rfloor$, $K \leftarrow \lfloor \log_2(TR) \rfloor$
  $x_1 \leftarrow 0$
  for $i \leftarrow 1, 2, \ldots, m+1$ **do**
    $t \leftarrow 2^{i-1}$
    if $t \leq m$ **then**
      $L \leftarrow 2^{i-1}$ ▷ $T_i := \{2^{i-1}, 2^{i-1}+1, \ldots, 2^i - 1\}$
    else
      $L \leftarrow T - 2^m + 1$ ▷ $T_{m+1} := \{2^m, 2^m+1, \ldots, T\}$
    end if
    for $j \leftarrow 1, 2, \ldots, K$ **do**
      $y_j \leftarrow$ QMINSTOCCONV($\mathcal{O}_f^{\tau,\alpha}, L/K$) ▷ Corollary 1 with $|T_i|/K$ queries
      for $l \leftarrow 0, 1, \ldots, L/K$ **do**
        $x_{t+l} \leftarrow x_{2^{i-1}}$ ▷ Onliner learner's output at time $t+l$
      end for
      if $f(y_j) < f(x_{2^i})$ **then**
        $x_{2^i} \leftarrow y_j$
      end if
    end for
  end for
end procedure

we have

$$f(x_{2^i}) - \min_{x \in \mathcal{K}} f(x) \leq \tilde{O}(n^5 \log(TR)/2^{i-1}) \quad \forall i \in [m+1]. \tag{12}$$

In all, we get that the regret bound as desired:

$$\begin{aligned}
\mathcal{R}_T &= \mathbb{E}\left[\sum_{t=1}^{T}(f(x_t) - f^*)\right] \\
&\leq \left(1 - O\left(\frac{\log T}{TR}\right)\right) \cdot \sum_{i=1}^{m+1} 2^{i-1} \cdot \tilde{O}\left(\frac{n^5 \log TR}{2^{i-1}}\right) + O\left(\frac{\log T}{TR}\right) \cdot T \cdot LR \\
&= \tilde{O}(n^5 \log(T) \log(TR)) + O(L \log T) \\
&= \tilde{O}(n^5 \log(T) \log(TR)).
\end{aligned}$$

□

Acknowledgement

We thank anonymous referees for valuable comments. TL was supported by a startup fund from Peking University. RZ was supported by the University Graduate Continuing Fellowship from UT Austin.

References

[1] Yasin Abbasi-Yadkori, Peter Bartlett, Victor Gabillon, and Alan Malek, *Hit-and-run for sampling and planning in non-convex spaces*, Artificial Intelligence and Statistics, pp. 888–895, PMLR, 2017, arXiv:1610.08865.

[2] Dorit Aharonov, Andris Ambainis, Julia Kempe, and Umesh Vazirani, *Quantum walks on graphs*, Proceedings of the thirty-third annual ACM symposium on Theory of computing, pp. 50–59, 2001, arXiv:quant-ph/0012090.

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

[4] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf, *Quantum SDP-solvers: Better upper and lower bounds*, Proceedings of the 58th Annual Symposium on Foundations of Computer Science, IEEE, 2017, arXiv:1705.01843.

[5] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf, *Convex optimization using quantum oracles*, Quantum **4** (2020), 220, arXiv:1809.00643.

[6] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, and Joseph C. Bardin et al., *Quantum supremacy using a programmable superconducting processor*, Nature **574** (2019), no. 7779, 505–510, arXiv:1910.11333.

[7] Claude J.P. Bélisle, H. Edwin Romeijn, and Robert L. Smith, *Hit-and-run algorithms for generating multivariate distributions*, Mathematics of Operations Research **18** (1993), no. 2, 255–266.

[8] Alexandre Belloni, Tengyuan Liang, Hariharan Narayanan, and Alexander Rakhlin, *Escaping the local minima via simulated annealing: Optimization of approximately convex functions*, Conference on Learning Theory, pp. 240–265, PMLR, 2015, arXiv:1501.07242.

[9] H.C.P. Berbee, C.G.E. Boender, A.H.G. Rinnooy Ran, C.L. Scheffer, Robert L. Smith, and Jan Telgen, *Hit-and-run algorithms for the identification of nonredundant linear inequalities*, Mathematical Programming **37** (1987), no. 2, 184–207.

[10] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge University Press, New York, NY, USA, 2004.

[11] Fernando G.S.L. Brandão, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu, *Quantum SDP solvers: Large speed-ups, optimality, and applications to quantum learning*, Proceedings of the 46th International Colloquium on Automata, Languages, and Programming, Leibniz International Proceedings in Informatics (LIPIcs), vol. 132, pp. 27:1–27:14, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, arXiv:1710.02581.

[12] Fernando G.S.L. Brandão and Krysta Svore, *Quantum speed-ups for semidefinite programming*, Proceedings of the 58th Annual Symposium on Foundations of Computer Science, pp. 415–426, 2017, arXiv:1609.05537.

[13] Yu Cao, Jianfeng Lu, and Lihan Wang, *On explicit L^2-convergence rate estimate for underdamped Langevin dynamics*, 2019, arXiv:1908.04746.

[14] Shouvanik Chakrabarti, Andrew M. Childs, Shih-Han Hung, Tongyang Li, Chunhao Wang, and Xiaodi Wu, *Quantum algorithm for estimating volumes of convex bodies*, 2019, arXiv:1908.03903.

[15] Shouvanik Chakrabarti, Andrew M. Childs, Tongyang Li, and Xiaodi Wu, *Quantum algorithms and lower bounds for convex optimization*, Quantum **4** (2020), 221, arXiv:1809.01731.

[16] Ming-Hui Chen and Bruce Schmeiser, *Performance of the Gibbs, hit-and-run, and Metropolis samplers*, Journal of Computational and Graphical Statistics **2** (1993), no. 3, 251–272.

[17] Andrew M. Childs, *Lecture notes on quantum algorithms*, Lecture notes at the University of Maryland, https://www.cs.umd.edu/ amchilds/qa/qa.pdf, 2021.

[18] Andrew M. Childs, Tongyang Li, Jin-Peng Liu, Chunhao Wang, and Ruizhe Zhang, *Quantum algorithms for sampling log-concave distributions and estimating normalizing constants*, 2022, manuscript.

[19] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade, *Structured logconcave sampling with a restricted gaussian oracle*, Conference on Learning Theory, pp. 355–366, 2009.

[20] Yassine Hamoudi, *Quantum Sub-Gaussian Mean Estimator*, 29th Annual European Symposium on Algorithms, Leibniz International Proceedings in Informatics (LIPIcs), vol. 204, pp. 50:1–50:17, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, arXiv:2108.12172.

[21] Aram W. Harrow and Annie Y. Wei, *Adaptive quantum simulated annealing for Bayesian inference and estimating partition functions*, Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 193–212, 2020, arXiv:1907.09965.

[22] Elad Hazan and Yuanzhi Li, *An optimal algorithm for bandit convex optimization*, 2016, arXiv:1603.04350.

[23] Alexei Yu Kitaev, *Quantum measurements and the Abelian stabilizer problem*, 1995, arXiv:quant-ph/9511026.

[24] Tor Lattimore, *Improved regret for zeroth-order adversarial bandit convex optimisation*, Mathematical Statistics and Learning **2** (2020), no. 3, 311–334, arXiv:2006.00475.

[25] Tor Lattimore and Andras Gyorgy, *Improved regret for zeroth-order stochastic convex bandits*, Conference on Learning Theory, pp. 2938–2964, PMLR, 2021.

[26] László Lovász, *Hit-and-run mixes fast*, Mathematical Programming **86** (1999), no. 3, 443–461.

[27] László Lovász and Santosh Vempala, *Hit-and-run is fast and fun*, 2003, Preprint, Microsoft Research, https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2003-05.pdf.

[28] László Lovász and Santosh Vempala, *Hit-and-run from a corner*, SIAM Journal on Computing **35** (2006), no. 4, 985–1005.

[29] László Lovász and Santosh Vempala, *The geometry of logconcave functions and sampling algorithms*, Random Structures and Algorithms **30** (2007), no. 3, 307–358.

[30] Michael A. Nielsen and Isaac L. Chuang, *Quantum computation and quantum information*, Cambridge University Press, 2000.

[31] Davide Orsucci, Hans J. Briegel, and Vedran Dunjko, *Faster quantum mixing for slowly evolving sequences of markov chains*, Quantum **2** (2018), 105, arXiv:1503.01334.

[32] András Prékopa, *Logarithmic concave measures with applications to stochastic programming*, Acta Scientiarum Mathematicarum **32** (1971), 301–316.

[33] András Prékopa, *On logarithmic concave measures and functions*, Acta Scientiarum Mathematicarum **34** (1973), 335–343.

[34] Andrej Risteski and Yuanzhi Li, *Algorithms and matching lower bounds for approximately-convex optimization*, Advances in Neural Information Processing Systems, vol. 29, 2016.

[35] Robert L. Smith, *Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions*, Operations Research **32** (1984), no. 6, 1296–1308.

[36] Mario Szegedy, *Quantum speed-up of Markov chain based algorithms*, 45th Annual IEEE Symposium on Foundations of Computer Science, pp. 32–41, IEEE, 2004.

[37] Zongqi Wan, Zhijie Zhang, Tongyang Li, Jialin Zhang, and Xiaoming Sun, *Quantum multi-armed bandits and stochastic linear bandits enjoy logarithmic regrets*, 2022, arXiv:2205.14988.

[38] Pawel Wocjan and Anura Abeyesinghe, *Speedup via quantum sampling*, Physical Review A **78** (2008), no. 4, 042336, arXiv:0804.4259.

[39] Pawel Wocjan, Chen-Fu Chiang, Daniel Nagaj, and Anura Abeyesinghe, *Quantum algorithm for approximating partition functions*, Physical Review A **80** (2009), no. 2, 022340, arXiv:0811.0596.

[40] Zelda B. Zabinsky, Robert L. Smith, J. Fred McDonald, H. Edwin Romeijn, and David E. Kaufman, *Improving hit-and-run for global optimization*, Journal of Global Optimization **3** (1993), no. 2, 171–192.

[41] Chenyi Zhang, Jiaqi Leng, and Tongyang Li, *Quantum algorithms for escaping from saddle points*, Quantum **5** (2021), 529, arXiv:2007.10253.

[42] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, and Li-Chao Peng et al., *Quantum computational advantage using photons*, Science **370** (2020), no. 6523, 1460–1463, arXiv:2012.01625.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Roadmap. In Appendix A, we provide more details of quantum walk and give our user-friendly framework. In Appendix B, we introduce the classical method for optimizing approximately convex functions in a self-contained way. In Appendix C, we prove our main result of quantum approximately convex optimization.

A Basic Facts about Quantum Walk

In this section, we first define the quantum walk operators and introduce some spectral properties. Then, we show how to efficiently implement a quantum walk.

A.1 Definitions and spectral properties of quantum walk

Let P be the transition operator of the classical Markov chain over the space K such that

$$\int_K P(x,y)\mathrm{d}y = 1 \quad \forall x \in K.$$

We define the following states, which capture key properties of the quantum walk:

$$|\psi_x\rangle := \int_K \sqrt{P(x,y)}\,|y\rangle\,\mathrm{d}y \quad \forall x \in K.$$

Definition 1 (Quantum walk operators)**.** *The quantum walk uses the following three operators:*

- $U := \int_K |x\rangle |\psi_x\rangle \langle x| \langle 0| \,\mathrm{d}x$ *for any* $x \in K$.
- $\Pi := \int_K |x\rangle |\psi_x\rangle \langle x| \langle \psi_x| \,\mathrm{d}x$ *is the projection to the subspace* $\mathrm{span}\{|x\rangle |\psi_x\rangle\}_{x\in K}$.
- $S := \int_K \int_K |y\rangle |x\rangle \langle x| \langle y| \,\mathrm{d}x\mathrm{d}y$ *is to swap the two quantum registers.*

Then, the quantum walk operator W is defined by:

$$W := S(2\Pi - I).$$

Definition 2 (Alternative definition of quantum walk operator, [38])**.** *Define the quantum walk operator*

$$W' := U^\dagger S U R_A U^\dagger S U R_A,$$

where $R_{\mathcal{A}}$ denotes the reflection about the subspace $\mathcal{A} := \{|x\rangle |0\rangle \mid x \in K\}$ for random walk space K, S is the swap operator, and U is the following operator:

$$U|x\rangle|0\rangle = \int_{y\in K} \sqrt{P(x,y)}\,|x\rangle|y\rangle\,\mathrm{d}y,$$

for P being the transition operator of the Markov chain.

Fact A.1 (Equivalence of the definitions, [14])**.** *Let W be defined as in Definition 1 and let W' be defined as in Definition 2. Then, W' and W have the same set of eigenvalues.*

Fact A.2 ([14])**.** *Let D be the discriminant operator of P defined as $D(x,y) := \sqrt{P(x,y)P(y,x)}$. Then, P and D have the same set of eigenvalues.*

Fact A.3 ([14])**.** *Let $\{\lambda_j\}$ be the eigenvalues of D. Then, the eigenvalues of W are*

$$\left\{\pm 1, \lambda_j + \sqrt{1-\lambda_j^2}\,i\right\}.$$

The following lemma shows that when the initial stationary is a warm start, then the eigenvalues whose eigenspaces have big overlap with the initial state are bounded away from 1.

Lemma 5 (Effective spectral gap for warm start, [18, Lemma C.7])**.** *Let $M = (\Omega, p)$ be an ergodic reversible Markov chain with a transition operator P and unique stationary state with a corresponding density ρ. Let $\{(\lambda_i, f_i)\}$ be the set of eigenvalues and eigenfunctions of P, and $|\psi_i\rangle$ be the eigenvectors of the corresponding quantum walk operator W. Let ρ_0 be a probability density that is*

a warm start for ρ and mixes up to TV-distance ϵ in t steps of M. Furthermore, assume that ρ_0 is a β-warm start of ρ.

Let $|\phi_{\rho_0}\rangle$ be the resulting state of applying the quantum walk update operator U to the state $|\rho_0\rangle$:

$$|\phi_{\rho_0}\rangle = \int_\Omega \sqrt{\rho_0(x)} \int_\Omega \sqrt{P(x,y)}\, |x\rangle\, |y\rangle \, \mathrm{d}x\mathrm{d}y.$$

Then, we have $|\langle\phi_{\rho_0}|\psi_i\rangle| = O(\beta\sqrt{\epsilon})$ for all i with $1 > \lambda_i \geq 1 - O(1/t)$.

Lemma 5 also applies to the initial distribution with a bounded ℓ_2-warmness:

Lemma 6 (Effective spectral gap for ℓ_2-warm start, [13])**.** *Let $M = (\Omega, p)$ be an ergodic reversible Markov chain with a transition operator P and unique stationary state with a corresponding density ρ. Let $\{(\lambda_i, f_i)\}$ be the set of eigenvalues and eigenfunctions of P, and $|\psi_i\rangle$ be the eigenvectors of the corresponding quantum walk operator W. Let ρ_0 be a probability density that is a warm start for ρ and mixes up to TV-distance ϵ in t steps of M. Furthermore, assume that $\|\rho/\rho_0\| = \int_\Omega \frac{\rho(x)}{\rho_0(x)}\rho(x)\mathrm{d}x \leq \gamma$.*

Let $|\phi_{\rho_0}\rangle$ be the resulting state of applying the quantum walk update operator U to the state $|\rho_0\rangle$:

$$|\phi_{\rho_0}\rangle = \int_\Omega \sqrt{\rho_0(x)} \int_\Omega \sqrt{P(x,y)}\, |x\rangle\, |y\rangle \, \mathrm{d}x\mathrm{d}y.$$

Then, we have $|\langle\phi_{\rho_0}|\psi_i\rangle| = O(\gamma^{1/4}\epsilon^{3/4} + \sqrt{\epsilon})$ for all i with $1 > \lambda_i \geq 1 - O(1/t)$.

A.2 Efficient implementation of quantum walk

The goal of this section is to give a user-friendly quantum walk implementation cost-analysis (Theorem 5).

Lemma 7 (Approximate reflector, [14, Corollary 4.1])**.** *Let W be a unitary operator with a unique leading eigenvector $|\psi_0\rangle$ with eigenvalue 1. Denote the remaining eigenvectors by $|\psi_j\rangle$ with corresponding eigenvaluese $e^{2\pi i\xi_j}$ for $j \geq 1$. For any $\Delta \in (0,1]$ and $\epsilon < 1/2$, define $a := \log(1/\Delta)$ and $c := \log(1/\sqrt{\epsilon})$. Let R be the reflector such that $R = \alpha\, |\psi_0\rangle\langle\psi_0| + (I - |\psi_0\rangle\langle\psi_0|)$.*

For any constant $\alpha \in \mathbb{C}$, there exists a quantum circuit $\widetilde{R}$ that uses $a \cdot c$ ancilla qubits and invokes the controlled-W gate $2^{a+1}c$ times such that

- $\widetilde{R}\, |\psi_0\rangle\, |0\rangle^{\otimes ac} = R\, |\psi_0\rangle\, |0\rangle^{\otimes ac}$.
- $\left\| \widetilde{R}\, |\psi_j\rangle\, |0\rangle^{\otimes ac} - R\, |\psi_j\rangle\, |0\rangle^{\otimes ac} \right\|_2 \leq \sqrt{\epsilon}$ *for $j \geq 1$ with $\xi_j \geq \Delta$.*

Lemma 8 ($\pi/3$-amplitude amplification, [38, Lemma 1])**.** *Let $|\psi\rangle, |\phi\rangle$ be two quantum states with $|\langle\psi|\phi\rangle| \geq p$ for some $p \in (0,1]$. Let $\omega = e^{i\pi/3}$. Define $R_\psi := \omega\, |\psi\rangle\langle\psi| + (I - |\psi\rangle\langle\psi|)$, and $R_\phi := \omega\, |\phi\rangle\langle\phi| + (I - |\phi\rangle\langle\phi|)$. Then, for $m \geq 1$, there exists a sequence of unitaries:*

$$V_0 = I, \;\; V_{j+1} = V_j R_\psi V_j^\dagger R_\phi V_j \quad \forall j \in [m],$$

such that

$$|\langle\psi|V_m|\phi\rangle|^2 \geq 1 - (1-p)^{3^m}.$$

Furthermore, the unitaries R_ϕ, R_ψ and their inverses are used at most 3^m times in V_m.

Theorem 5 (Quantum walk implementation cost)**.** *Let M_0, M_1 be two ergodic reversible Markov chains with stationary distributions π_0, π_1, respectively. Suppose π_0 is β_0-warm with respect to M_1 and mixes up to total variation distance ϵ in $t_0(\epsilon)$ steps. Similarly, suppose π_1 is β_1-warm with respect to M_0 and mixes in $t_1(\epsilon)$ steps. Let $\beta := \max\{\beta_0, \beta_1\}$. Moreover, we assume that $|\langle\pi_0|\pi_1\rangle| \geq p$.*

Given $|\pi_0\rangle$, we can obtain a state $|\widetilde{\pi}_1\rangle$ such that $\| \, |\widetilde{\pi}_1\rangle - |\pi_1\rangle \, \|_2 \leq \epsilon$ using

$$O\left(\sqrt{t_0(\epsilon) + t_1(\epsilon)} \cdot p^{-1} \log(\beta/p) \log^2(1/(p\epsilon))\right)$$

calls to the controlled walk operators controlled-W_0', controlled-W_1'.

Proof. By assumption, we know that π_0 mixes in $t_0' = t_0(\epsilon_1/\beta_0^2)$ steps in M_1 to achieve total variation distance ϵ_1/β_0^2, where ϵ_1 is a parameter to be chosen later. Similarly, π_1 mixes in $t_1' = t_1(\epsilon_1/\beta_0^2)$ steps in M_0 to achieve total variation distance ϵ_1/β_1^2.

We start from $|\pi_0\rangle$. By Lemma 5, we have $|\pi_0\rangle = |\pi_{0,\mathsf{good}}\rangle + |e_0\rangle$, where $|\pi_{0,\mathsf{good}}\rangle$ lies in the subspace spanned by the eigenvectors $|\psi_j\rangle$ of W_1' with corresponding eigenvalue λ_j of P_1 such that $\lambda_j = 0$ or $\lambda_j \leq 1 - \Omega(1/t_0')$. Let $e^{2\pi i \xi_j}$ be the eigenvalue of $|\psi_j\rangle$ of W_1'. By Fact A.1 and Fact A.3, we get that $\xi_j = 0$ or $\xi_j \geq \Omega({t_0'}^{-1/2})$. By Lemma 5, we also have $\| \, |e_0\rangle \, \| \leq \epsilon_1$.

Then, by Lemma 7 with $\Delta = \Omega({t_0'}^{-1/2})$ and $\epsilon = \epsilon_1^2$, we can implement $\widetilde{R}_1$ such that $\|R_1 \, |\phi\rangle - \widetilde{R}_1 \, |\phi\rangle \, \|_2 \leq 2\epsilon_1$ using $O(\sqrt{t_0'} \log(1/\epsilon_1))$ calls to controlled-W_1', where $|\phi\rangle$ is any state that occurs during $\pi/3$-amplitude amplification (Lemma 8) for $|\pi_0\rangle$ towards $|\pi_1\rangle$.

In the same way, we can start from $|\pi_1\rangle$ and show that $\widetilde{R}_0$ can be implemented using $O(\sqrt{t_1'} \log(1/\epsilon_1))$ calls to controlled-W_0' such that $\|R_0 \, |\phi'\rangle - \widetilde{R}_0 \, |\phi'\rangle \, \| \leq 2\epsilon_1$, where $|\phi'\rangle$ is any state that occurs during $\pi/3$-amplitude amplification for $|\pi_1\rangle$ towards $|\pi_0\rangle$.

Suppose we can implement R_0 and R_1 perfectly. Then, we can prepare a state $|\widetilde{\pi}_1\rangle$ such that $|\langle \widetilde{\pi}_1 | \pi_1 \rangle| \geq 1 - (1-p)^{3^m}$ using 3^m calls to R_0, R_1 and their inverses, by applying $\pi/3$-amplitude amplification (Lemma 8) to $|\pi_i\rangle$. Thus, by taking $m = O(p^{-1} \log(1/\epsilon_2))$ where ϵ_2 is a parameter to be chosen later, we have $\| \, |\pi_1\rangle - |\widetilde{\pi}_1\rangle \, \|_2 \leq \epsilon_2$. However, since each call to $\widetilde{R}_0$ or $\widetilde{R}_1$ causes an error of ϵ_1, the total error will be

$$O(\epsilon_2 + \epsilon_1 \cdot p^{-1} \log(1/\epsilon_2)) = \epsilon,$$

where we take $\epsilon_1 := O(p\epsilon \log^{-1}(1/\epsilon))$ and $\epsilon_2 := \epsilon_1^2$.

Therefore, the total number of calls to controlled-W_0', controlled-W_1' is

$$O\left((\sqrt{t_0'} + \sqrt{t_1'}) \cdot p^{-1} \log^2(1/p\epsilon) \right),$$

where $t_i' = t_i(\epsilon_1/\beta_i^2) = O(t_i(\epsilon) \cdot \log(\beta_i/p))$.

The theorem is then proved. □

The following corollary is an immediate consequence of Theorem 5, and it also gives Theorem 3.

Corollary 2 (Quantum walk implementation cost (ℓ_2-warm starts))**.** *Let M_0, M_1 be two ergodic reversible Markov chains with stationary distributions π_0, π_1, respectively. Suppose π_0 mixes towards π_1 in M_1 up to total variation distance ϵ in $t_0(\epsilon)$ steps. Similarly, suppose π_1 mixes towards π_0 in M_0 in $t_1(\epsilon)$ steps. Suppose $\|\pi_0/\pi_1\| = O(1)$ and $\|\pi_1/\pi_0\| = O(1)$. Moreover, we assume that $|\langle \pi_0 | \pi_1 \rangle| = \Omega(1)$.*

Given $|\pi_0\rangle$, we can obtain a state $|\widetilde{\pi}_1\rangle$ such that $\| \, |\widetilde{\pi}_1\rangle - |\pi_1\rangle \, \|_2 \leq \epsilon$ using

$$O\left(\sqrt{t_0(\epsilon) + t_1(\epsilon)} \log^2(1/\epsilon) \right)$$

calls to the controlled walk operators controlled-W_0', controlled-W_1'.

B Classical Approach for Optimizing Approximately Convex Functions

In this section, we introduce the classical approach [8] for the optimization of approximately convex functions as in Eq. (1).

B.1 Low level: Hit-and-Run for approximate log-concave distributions

The Hit-and-Run walk uses a unidimensional rejection sampler to sample a point from the distribution π_g restricted to a line ℓ. The following lemma shows the performance guarantee of the unidimensional sampler:

Lemma 9 (Unidimensional rejection sampler, [8, Lemma 5])**.** *Given $\beta = O(1)$. Let g be a β-log-concave function and ℓ be a bounded line segment on $\mathcal{K}$. For $\epsilon \in (0, e^{-2\beta}/2)$, Algorithm 4 outputs*

Algorithm 3 Hit-and-Run walk

procedure HITANDRUN(π_0, π_g, Σ, m) ▷ π_g is the target distribution on $\mathcal{K}$ induced by a nonnegative function g, Σ is a linear transformation
  $\mathbf{x}_0 \leftarrow$ sample from π_0
  Choose accuracy parameter ϵ_ℓ
  for $i \leftarrow 1, \ldots, m$ **do**
    $\mathbf{u} \leftarrow$ uniformly sample from the surface of ellipse given by Σ acting on sphere
    $\ell(t) := \mathbf{x}_{i-1} + t\mathbf{u}$, compute $[\mathbf{s}, \mathbf{t}] \leftarrow \ell \cap \mathcal{K}$
    $\mathbf{x}_i \leftarrow$ UNISAMPLER(g, β, $[\mathbf{s}, \mathbf{t}]$, ϵ_ℓ)
  end for
  return $\mathbf{x}_m$
end procedure

a point $\mathbf{x} \in \ell$ with a distribution $\widetilde{\pi}_\ell$ such that

$$d_{\mathrm{TV}}(\widetilde{\pi}_\ell, \pi_g|_\ell) \leq 3e^{2\beta}\epsilon.$$

Moreover, the algorithm requires $\widetilde{O}(1)$ evaluations of the function g.

The following theorem gives the mixing time of the standard Hit-and-Run walk for an approximate log-concave distribution, where we assume that in each step we directly sample from the restricted distribution $\pi_g|_\ell$.

Theorem 6 (Mixing time of Hit-and-Run for approximate log-concave distribution, [8, Theorem 4])**.** *Let π_g be the stationary measure associated with the Hit-and-Run walk based on a $\beta/2$-approximately log-concave function g, and let $\sigma^{(0)}$ be an initial distribution with ℓ_2-warmness $M := \|\sigma^{(0)}/\pi_g\|$. There is a universal constant C such that for any $\gamma \in (0, 1/2)$, if*

$$m \geq Cn^2 \frac{e^{6\beta}R^2}{r^2} \log^4\left(\frac{e^\beta MnR}{r\gamma^2}\right) \log\left(\frac{M}{\gamma}\right),$$

then m steps of the Hit-and-Run random walk based on g yield

$$d_{\mathrm{TV}}(\sigma^{(m)}, \pi_g) \leq \gamma.$$

The next theorem shows the closeness between the output distribution of Algorithm 3 and the target distribution π_g. Due to the unidimensional rejection sampler (Algorithm 4), the stationary distribution of Algorithm 3 may not be exactly π_g. Nevertheless, we can still show that it will not deviate a lot.

Theorem 7 (The effect of the rejection sampler, [8, Theorem 5])**.** *Let π_g, $\sigma^{(0)}$ be defined as in Theorem 6. Let $\hat{\sigma}^{(m)}$ denote the output distribution of Algorithm 3 with initial distribution $\hat{\sigma}^{(0)}$ in m steps. Let ϵ_ℓ be the accuracy parameter for the unidimensional rejection sampler (Algorithm 4). Then, we have*

$$d_{\mathrm{TV}}(\hat{\sigma}^{(m)}, \sigma^{(m)}) \leq m\epsilon_\ell + 2d_{\mathrm{TV}}(\hat{\sigma}^{(0)}, \sigma^{(0)}).$$

In particular, for $\gamma \in (0, 1/e)$, suppose $d_{\mathrm{TV}}(\hat{\sigma}^{(0)}, \sigma^{(0)}) \leq \gamma/8$. Let $s \in (0, 1)$ be such that $H_s \leq \gamma/4$, where H_s is defined to be:

$$H_s := \sup_{A \subset \mathcal{K}: \pi_g(A) \leq s} |\pi_g(A) - \sigma^{(0)}(A)|.$$

Then, there is a constant C' such that, if we take $\epsilon_\ell := \gamma e^{-2\beta}/(12m)$ and

$$m \geq C'n^2 \frac{e^{6\beta}R^2}{r^2} \log^4\left(\frac{e^\beta nR}{rs}\right) \log(1/s),$$

we have

$$d_{\mathrm{TV}}(\hat{\sigma}^{(m)}, \pi_g) \leq \gamma.$$

Algorithm 4 Unidimensional rejection sampler

procedure INITP(g, β, $\ell = [\mathbf{s}, \mathbf{t}]$)
  while true **do**
    $\mathbf{x}_1 \leftarrow \frac{3}{4}\mathbf{s} + \frac{1}{4}\mathbf{t}$, $\mathbf{x}_2 \leftarrow \frac{1}{2}\mathbf{s} + \frac{1}{2}\mathbf{t}$, $\mathbf{x}_3 \leftarrow \frac{1}{4}\mathbf{s} + \frac{3}{4}\mathbf{t}$
    if $|\log(g(\mathbf{x}_1)) - \log(g(\mathbf{x}_3))| > \beta$ **then**
      $\mathbf{t} \leftarrow \mathbf{x}_3$ if $g(\mathbf{x}_1) > g(\mathbf{x}_3)$; $\mathbf{s} \leftarrow \mathbf{x}_1$ otherwise
    else if $|\log(g(\mathbf{x}_1)) - \log(g(\mathbf{x}_2))| > \beta$ **then**
      $\mathbf{t} \leftarrow \mathbf{x}_2$ if $g(\mathbf{x}_1) > g(\mathbf{x}_2)$; $\mathbf{s} \leftarrow \mathbf{x}_1$ otherwise
    else if $|\log(g(\mathbf{x}_2)) - \log(g(\mathbf{x}_3))| > \beta$ **then**
      $\mathbf{t} \leftarrow \mathbf{x}_3$ if $g(\mathbf{x}_2) > g(\mathbf{x}_3)$; $\mathbf{s} \leftarrow \mathbf{x}_2$ otherwise
    else
      return $\mathbf{p} \leftarrow \arg\max_{\mathbf{x} \in \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}} g(\mathbf{x})$
    end if
  end while
end procedure
procedure BINSEARCH(g,$\mathbf{x}_l$, $\mathbf{x}_r$, V_l, V_r)
  while true **do**
    $\mathbf{x}_m \leftarrow (\mathbf{x}_l + \mathbf{x}_r)/2$
    if $g(\mathbf{x}_m) > V_r$ **then**
      $\mathbf{x}_r \leftarrow \mathbf{x}_m$
    else if $g(\mathbf{x}_m) < V_l$ **then**
      $\mathbf{x}_l \leftarrow \mathbf{x}_m$
    else
      return $\mathbf{x}_m$
    end if
  end while
end procedure
procedure INITE(g, β, $\ell = [\mathbf{s}, \mathbf{t}]$, $\mathbf{p}$, ϵ_ℓ)
  if $g(\mathbf{s}) \geq \frac{1}{2}e^{-\beta}\epsilon_\ell g(\mathbf{p})$ **then**
    $\mathbf{e}_0 \leftarrow \mathbf{s}$
  else
    $\mathbf{e}_0 \leftarrow$ BINSEARCH$(g, \mathbf{s}, \mathbf{p}, \frac{1}{2}e^{-\beta}\epsilon_\ell g(\mathbf{p}), \epsilon_\ell g(\mathbf{p}))$
  end if
  if $g(\mathbf{t}) \geq \frac{1}{2}e^{-\beta}\epsilon_\ell g(\mathbf{p})$ **then**
    $\mathbf{e}_1 \leftarrow \mathbf{t}$
  else
    $\mathbf{e}_1 \leftarrow$ BINSEARCH$(g, \mathbf{p}, \mathbf{t}, \frac{1}{2}e^{-\beta}\epsilon_\ell g(\mathbf{p}), \epsilon_\ell g(\mathbf{p}))$
  end if
  return $\mathbf{e}_0, \mathbf{e}_1$
end procedure
procedure UNISAMPLER(g, β, $\ell = [\mathbf{s}, \mathbf{t}]$, ϵ_ℓ)
  $\mathbf{p} \leftarrow$ INITP(g, β, ℓ)
  $\mathbf{e}_0, \mathbf{e}_1 \leftarrow$ INITE(g, β, ℓ, $\mathbf{p}$, ϵ_ℓ)
  while true **do**
    $\mathbf{x} \leftarrow \mathrm{Uniform}([\mathbf{e}_0, \mathbf{e}_1])$, $r \leftarrow \mathrm{Uniform}([0, 1])$
    if $r \leq g(\mathbf{x})/(e^{3\beta}g(\mathbf{p}))$ **then**
      return $\mathbf{x}$
    end if
  end while
end procedure

B.2 Mid level: rounding into isotropic position

The following lemma rounds a β-log-concave distribution to near-isotropic position.

Lemma 10 (Rounding β-log-concave distribution, [8, Lemma 9])**.** *Let π be a β-log-concave distribution in $\mathbb{R}^n$. By taking $N = \Theta(n \log n)$ i.i.d. samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from π, we have*

$$\frac{1}{2} \leq \sigma_{\min}\Big(\frac{1}{N}\sum_{i\in[N]} \mathbf{x}_i\mathbf{x}_i^\top\Big) \leq \sigma_{\max}\Big(\frac{1}{N}\sum_{i\in[N]} \mathbf{x}_i\mathbf{x}_i^\top\Big) \leq \frac{3}{2}$$

holds with probability at least $1 - n^{-O(1)}$.

B.3 High level: simulated annealing

At high level, we run a simulated annealing for a series of functions:

$$h_i(x) := \exp(-f(x)/T_i), \text{ and } g_i(x) := \exp(-F(x)/T_i),$$

where f, F satisfy Eq. (1) and $\{T_i\}_{i\in[K]}$ are parameters to be chosen later.

Algorithm 5 Simulated annealing

procedure SIMANNEALING(K, $\{T_i\}_{i\in[K]}$)
  $N \leftarrow \Theta(n \log n)$ ▷ The number of strands
  $X_0^j \sim \mathrm{Uniform}(\mathcal{K})$ for $j = 1, \ldots, N$
  $\mathcal{K}_0 \leftarrow \mathcal{K}$, $\Sigma_0 \leftarrow I$
  $m \leftarrow \widetilde{O}(n^3)$ ▷ Theorem 7
  for $i \leftarrow 1, \ldots, K$ **do**
    $\Sigma_i' \leftarrow$ the rounding linear transformation for $\{X_{i-1}^j\}_{j\in[N]}$
    $\Sigma_i \leftarrow \Sigma_i' \circ \Sigma_{i-1}$
    for $j \leftarrow 1, \ldots, N$ **do**
      $X_i^j \leftarrow$ HITANDRUN($X_{i-1}^j, \pi_{g_i}, \Sigma_i, m$) ▷ Algorithm 3
    end for
  end for
  return $\arg\min_{i\in[K], j\in[N]} F(X_i^j)$
end procedure

Lemma 11 (The warmness of annealing distributions, [8, Lemma 8])**.** *Let $g(x) = exp(-F(x))$ be a β-log-concave function. Let π_{g_i} be a distribution with density proportional to $g_i(x) = \exp(-F(x)/T_i)$, supported on $\mathcal{K}$. Let $T_i := T_{i-1}\left(1 - \frac{1}{\sqrt{n}}\right)$. Then,*

$$\|\pi_{g_i}/\pi_{g_{i+1}}\| \leq C_\gamma = 5\exp(2\beta/T_i).$$

Theorem 8 (Sample Guarantee for the simulated annealing, [8, Theorem 6])**.** *Fix a target accuracy $\gamma \in (0, 1/e)$ and let g be an $\beta/2$-approximately log-concave function in $\mathbb{R}^n$. Suppose the simulated annealing algorithm (Algorithm 5) is run for $K = \sqrt{n}\log(1/\rho)$ epochs with temperature parameters $T_i = (1 - 1/\sqrt{n})^i$ for $0 \leq i \leq K$. If the Hit-and-Run with the unidimensional sampling scheme (Algorithm 3) is run for $m = \widetilde{O}(n^3)$ number of steps prescribed in Theorem 7, the algorithm maintains that*

$$d_{\mathrm{TV}}(\hat{\sigma}_i^{(m)}, \pi_{g_i}) \leq e\gamma$$

for each $i \in [K]$, where $\hat{\sigma}_i^{(m)}$ is the distribution of the m-th step of Hit-and-Run. Here, m depends polylogarithmically on $1/\rho$.

Then, we have the following optimization guarantee for the simulated annealing procedure:

Theorem 9 (Optimization guarantee for the simulated annealing, [8, Corollary 1])**.** *Suppose F is approximately convex and $|F - f| \leq \epsilon/n$ as in Eq. (1). The simulated annealing method with $K = \sqrt{n}\log(n/\epsilon)$ epochs produces a random point X such that*

$$\mathbb{E}[f(X)] - \min_{\mathbf{x}\in\mathcal{K}} f(\mathbf{x}) \leq \epsilon,$$

and thus,

$$\mathbb{E}[F(X)] - \min_{\mathbf{x}\in\mathcal{K}} F(\mathbf{x}) \leq 2\epsilon.$$

Furthermore, the number of oracle queries required by the method is $\widetilde{O}(n^{4.5})$.

C Quantum Speedup for Optimizing Approximately Convex Functions

As we discussed in previous section, there are three levels for the optimization algorithm. The goal of this section is to prove Theorem 1, where we improve the classical query complexity $\widetilde{O}(n^{4.5})$ (Theorem 9) to quantum query complexity $\widetilde{O}(n^3)$. The main idea is to use quantum walk algorithm (introduced in Appendix A) to speed-up the low level such that each sample can be generated with less queries.

C.1 Quantum speedup for low-level

In this section, we show how to use the quantum walk algorithm to speedup the sampling procedure in the simulated annealing process. According to the framework (Corollary 2), we first show that the each Markov chain's stationary distribution in the annealing process is a warm-start for its adjacent chains, and the Markov chains are slowly-varying. Then, we show how to implement the quantum walk operator for the Hit-and-Run walk. Finally, we prove the quantum speedup from $\widetilde{O}(n^3)$ classical query complexity to $\widetilde{O}(n^{1.5})$ quantum query complexity.

Warmness and overlap for the stationary distributions. We first show that π_{g_i} is a warm-start for $\pi_{g_{i+1}}$, and vice versa.

By lemma 11, we know that $\|\pi_{g_i}/\pi_{g_{i+1}}\| \leq 5\exp(2\beta/T_i)$. Similarly, we can also bound $\|\pi_{g_{i+1}}/\pi_{g_i}\|$:

Lemma 12. *Let $g(x) = exp(-F(x))$ be a β-log-concave function. Let π_{g_i} be a distribution with density proportional to $g_i(x) = \exp(-F(x)/T_i)$, supported on $\mathcal{K}$. Let $T_i := T_{i-1}\left(1 - \frac{1}{\sqrt{n}}\right)$. Then,*

$$\|\pi_{g_{i+1}}/\pi_{g_i}\| \leq 8\exp(2\beta/T_{i+1}).$$

Proof. Define $Y(a) := \int_{\mathcal{K}} \exp(-F(x)a)\mathrm{d}x$. Then, we have

$$\begin{aligned}
\|\pi_{g_{i+1}}/\pi_{g_i}\| &= \frac{\int_{\mathcal{K}} \exp(-F(x)(2/T_{i+1} - 1/T_i))\mathrm{d}x \cdot \int_{\mathcal{K}} \exp(-F(x)/T_i)\mathrm{d}x}{\left(\int_{\mathcal{K}} \exp(-F(x)/T_{i+1})\mathrm{d}x\right)^2} \\
&= \frac{Y(2/T_{i+1} - 1/T_i)Y(1/T_i)}{Y(1/T_{i+1})^2}.
\end{aligned}$$

Define $G(x,t) := g(x/t)^t$. Then, we have

$$\begin{aligned}
G(\lambda x + (1-\lambda)x', \lambda t + (1-\lambda)t') &= g\Big(\frac{\lambda x + (1-\lambda)x'}{\lambda t + (1-\lambda)t'}\Big)^{\lambda t + (1-\lambda)t'} \\
&= g\Big(\frac{\lambda t}{\lambda t + (1-\lambda)t'}\frac{x}{t} + \frac{(1-\lambda)t'}{\lambda t + (1-\lambda)t'}\frac{x'}{t'}\Big)^{\lambda t + (1-\lambda)t'} \\
&\geq \exp(-\beta(\lambda t + (1-\lambda)t')) \cdot g\Big(\frac{x}{t}\Big)^{\lambda t} \cdot g\Big(\frac{x'}{t'}\Big)^{(1-\lambda)t'} \\
&= \exp(-\beta(\lambda t + (1-\lambda)t')) \cdot G(x,t)^{\lambda} \cdot G(x',t')^{1-\lambda} \\
&= \left(\exp(-\beta t)G(x,t)\right)^{\lambda} \cdot \left(\exp(-\beta t')G(x',t')\right)^{1-\lambda},
\end{aligned}$$

where the inequality follows from g is β-log-concave.

By Prékopa–Leindler inequality (Theorem 10), it implies that

$$\int_{\mathcal{K}} G(x, \lambda t + (1-\lambda)t')\mathrm{d}x \geq \Big(\int_{\mathcal{K}} \exp(-\beta t) G(x,t)\mathrm{d}x\Big)^{\lambda} \cdot \Big(\int_{\mathcal{K}} \exp(-\beta t') G(x,t')\mathrm{d}x\Big)^{1-\lambda}.$$

Note that

$$\int_{\mathcal{K}} G(x,t)\mathrm{d}x = \int_{\mathcal{K}} g\Big(\frac{x}{t}\Big)^t \mathrm{d}x = t^n \int_{\mathcal{K}} g(x)^t \mathrm{d}x = t^n \int_{\mathcal{K}} \exp(-F(x)t)\mathrm{d}x = t^n Y(t).$$

Hence, for $\lambda = \frac{1}{2}$, we have

$$\Big(\frac{t+t'}{2}\Big)^{2n} Y\Big(\frac{t+t'}{2}\Big)^2 \geq \exp(-\beta(t+t')/2) \cdot t^n Y(t) \cdot t'^n Y(t'),$$

which implies that

$$\frac{Y(t)Y(t')}{Y(\frac{t+t'}{2})^2} \leq \exp\Big(\frac{\beta(t+t')}{2}\Big) \cdot \Big(\frac{(t+t')^2/4}{tt'}\Big)^n. \tag{13}$$

By taking $t = 2/T_{i+1} - 1/T_i$ and $t' = 1/T_i$, we have

$$\begin{aligned}
\|\pi_{g_{i+1}}/\pi_{g_i}\| &\leq \frac{Y(2/T_{i+1} - 1/T_i)Y(1/T_i)}{Y(1/T_{i+1})^2} \\
&\leq \exp(2\beta/T_{i+1}) \cdot \Big(\frac{(1/T_{i+1})^2}{(2/T_{i+1} - 1/T_i)(1/T_i)}\Big)^n \\
&= \exp(2\beta/T_{i+1}) \cdot \Big(\frac{1}{(2-(1-1/\sqrt{n}))(1-1/\sqrt{n})}\Big)^n \\
&= \exp(2\beta/T_{i+1}) \cdot \Big(1 + \frac{1}{n-1}\Big)^n \\
&\leq \exp(2\beta/T_{i+1}) \cdot \exp(n/(n-1)) \\
&\leq 8\exp(2\beta/T_{i+1}),
\end{aligned}$$

where the third step follows from $T_{i+1} = T_i(1 - \frac{1}{\sqrt{n}})$.

The lemma is then proved. □

Remark 1. *Since we assume that $|F(x) - f(x)| \leq \epsilon/n$ in Eq. (1), i.e., $\beta = \epsilon/n$, by Lemmas 11 and 12, we know that the warmness $M := \max\{\|\pi_{g+i}/\pi_{g_{i+1}}\|, \|\pi_{g_{i+1}}/\pi_{g_i}\|\}$ can be bounded by $O(\exp(2\epsilon/(nT_{i+1})))$. Since we choose the final temperature $T_k = \epsilon/n$, we get that $M = O(1)$. Therefore, it satisfies the warmness condition in Corollary 2.*

Theorem 10 (Prékopa–Leindler inequality, [32, 33])**.** *Let $0 < \lambda < 1$ and let $f, g, h : \mathbb{R}^n \to [0, \infty)$ be measurable functions. Suppose that these functions satisfy*

$$h(\lambda x + (1-\lambda)y) \geq f(x)^{\lambda} \cdot g(y)^{1-\lambda} \quad \forall x, y \in \mathbb{R}^n.$$

Then, we have

$$\int_{\mathbb{R}^n} h(x)\mathrm{d}x \geq \Big(\int_{\mathbb{R}^n} f(x)\mathrm{d}x\Big)^{\lambda} \cdot \Big(\int_{\mathbb{R}^n} g(x)\mathrm{d}x\Big)^{1-\lambda}.$$

Lemma 13 (Bound distribution overlap)**.** *Let $g(x) = \exp(-F(x))$ be a β-log-concave function. Let π_{g_i} be a distribution with density proportional to $g_i(x) = \exp(-F(x)/T_i)$, supported on $\mathcal{K}$. Let $T_i := T_{i-1}\Big(1 - \frac{1}{\sqrt{n}}\Big)$. Then,*

$$\langle \pi_i | \pi_{i+1} \rangle \geq \exp(-(\beta/T_{i+1} + 1)/2).$$

Proof. We can write the overlap as follows:

$$\begin{aligned}
\langle \pi_{g_i} | \pi_{g_{i+1}} \rangle &= \frac{\int_{\mathcal{K}} \sqrt{g_i(x) g_{i+1}(x)}\mathrm{d}x}{(\int_{\mathcal{K}} g_i(x)\mathrm{d}x)^{1/2} \cdot (\int_{\mathcal{K}} g_{i+1}(x)\mathrm{d}x)^{1/2}} \\
&= \frac{\int_{\mathcal{K}} \exp(-F(x)(1/T_i + 1/T_{i+1})/2)\mathrm{d}x}{(\int_{\mathcal{K}} \exp(-F(x)/T_i)\mathrm{d}x)^{1/2} \cdot (\int_{\mathcal{K}} \exp(-F(x)/T_{i+1})\mathrm{d}x)^{1/2}} \\
&= \frac{Y((1/T_i + 1/T_{i+1})/2)}{Y(1/T_i)^{1/2} Y(1/T_{i+1})^{1/2}},
\end{aligned}$$

where $Y(t) := \int_{\mathcal{K}} \exp(-F(x)t)\mathrm{d}x$.

By Eq. (13), we have

$$\begin{aligned}
\frac{Y(1/T_i)Y(1/T_{T_{i+1}})}{Y((1/T_i+1/T_{i+1})/2)^2} &\leq \exp(\beta(1/T_i+1/T_{i+1})/2)\cdot\Big(\frac{(1/T_i+1/T_{i+1})^2/4}{1/(T_iT_{i+1})}\Big)^n \\
&= \exp\big(\beta(2-1/\sqrt{n})/(2T_{i+1})\big)\cdot\Big(1+\frac{1}{4(n-\sqrt{n})}\Big)^n \\
&\leq \exp\left(\frac{1}{4}\frac{\sqrt{n}}{\sqrt{n}-1}\right)\cdot\exp(\beta/T_{i+1}) \\
&\leq \exp(\beta/T_{i+1}+1),
\end{aligned}$$

where the second step follows from $T_{i+1} = T_i(1-1/\sqrt{n})$.

Therefore,

$$\langle \pi_{g_i} | \pi_{g_{i+1}} \rangle \geq \exp(-(\beta/T_{i+1}+1)/2).$$

□

Remark 2. *By taking $\beta = \epsilon/n$ and $T_i \geq \epsilon/n$ in Lemma 13, we have for any $i \in [K-1]$, the overlap can be upper-bounded by:*

$$\langle \pi_{g_i} | \pi_{g_{i+1}} \rangle \geq \exp(-(\beta/T_K+1)/2) = e^{-1}.$$

Implementing the quantum walk operator. We introduce how to implement the quantum walk update operator U such that:

$$U\,|x\rangle\,|0\rangle = \int_{\mathcal{K}} \sqrt{P_{x,y}}\,|x\rangle\,|y\rangle\,\mathrm{d}y,$$

where P is the stochastic transition matrix for the Hit-and-Run walk.

Given an input state $|x\rangle$. We first prepare an n-dimensional Gaussian state in an ancilla register:

$$|x\rangle\,|0\rangle \longrightarrow |x\rangle \int_{\mathbb{R}^n} (2\pi)^{-n/4}\,|z\rangle\,\mathrm{d}z.$$

Then, by normalizing z and applying the linear transformation Σ in another quantum register, we get that

$$|x\rangle \int_{\mathbb{R}^n} (2\pi)^{-n/4}\,|z\rangle \left|\frac{\Sigma z}{\|z\|}\right\rangle \mathrm{d}z.$$

If we un-compute the $|z\rangle$ register, we get that (ignoring the normalization factor):

$$|x\rangle \int_{\Sigma\mathbb{S}^n} |u\rangle\,\mathrm{d}u.$$

Next, we coherently compute the two end-points of $\ell \cap \mathcal{K}$ for $\ell(t) := x + ut$ in the ancilla registers:

$$\int_{\Sigma\mathbb{S}^n} \mathrm{d}u\,|x\rangle\,|u\rangle\,|0\rangle \longrightarrow \int_{\Sigma\mathbb{S}^n} \mathrm{d}u\,|x\rangle\,|u\rangle\,|s,t\rangle$$

We coherently simulate the unidimensional sampler (Algorithm 4). More specifically, we can compute the points p, e_0, e_1 in ancilla registers:

$$\int_{\Sigma\mathbb{S}^n} \mathrm{d}u\,|x\rangle\,|u\rangle\,|s,t,p,e_0,e_1\rangle$$

Then, we prepare two unifrom distribution states in the next two ancilla qubits:

$$\int_{\Sigma\mathbb{S}^n} \mathrm{d}u\,|x\rangle\,|u\rangle\,|s,t,p,e_0,e_1\rangle \int_{[0,1]^2} |r',r\rangle\,\mathrm{d}r'\mathrm{d}r$$

And the next proposed point y can be computed via $y := e_0 + r'(e_1 - e_0)$:

$$\int_{\Sigma\mathbb{S}^n} \mathrm{d}u \, |x\rangle \, |u\rangle \, |s, t, p, e_0, e_1\rangle \int_{[0,1]^2} |r', r\rangle \, \mathrm{d}r' \mathrm{d}r \, |y\rangle$$

Then, we check the condition $r \leq g(y)/(3^\beta g(p))$ by querying the evaluation oracle twice and use an ancilla qubit to indicate whether it is satisfied:

$$\int_{\Sigma\mathbb{S}^n} \int_{[0,1]^2} \mathrm{d}u \mathrm{d}r' \mathrm{d}r \, |x\rangle \, |u\rangle \, |s, t, p, e_0, e_1\rangle \, |r', r\rangle \, |y\rangle \, |b\rangle \, ,$$

where $b \in \{0, 1\}$. Then, we post-select[4] the last qubit for $b = 1$. By un-computing the registers for $u, s, t, p, e_0, e_1, r, r'$, we get the desired state:

$$|x\rangle \int_{\mathcal{K}} \sqrt{P_{x,y}} \, |y\rangle \, .$$

By Lemma 9, we get that this procedure (including the post-selection cost) takes $O(1)$ oracle queries with high probability. Therefore, we get the following lemma:

Lemma 14 (Implementation cost of the quantum walk update operator)**.** *For the Hit-and-Run walk (Algorithm 3) with the unidimensional sampler (Algorithm 4), the quantum walk update operator U can be implemented by querying the evaluation oracle $O(1)$ times.*

$\widetilde{O}(n^{1.5})$-query quantum algorithm. We have the following theorem:

Theorem 11 (Quantum speedup for the Hit-and-Run sampler)**.** *Let $\gamma \in (0, 1/e)$. Let π_g be the stationary measure associated with the Hit-and-Run walk based on a $\beta/2$-approximately log-concave function g. Let $T_i = (1 - 1/\sqrt{n})^i$ for $0 \leq i \leq K$ be the annealing schedule. Suppose we use the quantum walk to implement the Hit-and-Run walk (Algorithm 3). Then, for each $0 \leq i \leq K - 1$, given a state $|\pi_{g_i}\rangle$, we can produce a state $|\hat{\sigma}_i^{(m)}\rangle$ such that*

$$\| |\pi_{g_{i+1}}\rangle - |\hat{\sigma}_i^{(m)}\rangle \|_2 \leq O(\gamma),$$

using $m = \widetilde{O}(n^{1.5})$ calls for the evaluation oracle.

Proof. We use the quantum walk framework in Corollary 2.

For the warmness, by Remark 1, we know that in this annealing schedule, $\|\pi_{g_i}/\pi_{g_{i+1}}\|$ and $\|\pi_{g_{i+1}}/\pi_{g_i}\|$ are upper-bounded by some constants.

Then, by Theorem 6, we get that the number of steps for evolving from π_{g_i} to $\pi_{g_{i+1}}$ and from $\pi_{g_{i+1}}$ to π_{g_i} is $\widetilde{O}(n^3)$ classically. The proof of Theorem 7 implies that the stationary distribution of the Hit-and-Run walk with unidimensional sampler is very close to the original Markov chain, only causing a constant blowup to the total variation distance. Thus, for $\gamma' = O(\gamma)$, we have $t_1(\gamma'), t_2(\gamma') = \widetilde{O}(n^3)$ in Corollary 2.

By Remark 2, we know that in this annealing schedule, the adjacent distributions have a big overlap. In particular, we have $|\langle \pi_{g_i} | \pi_{g_{i+1}} \rangle| \geq \Omega(1)$, satisfying the condition in Corollary 2.

Therefore, by Corollary 2, we get that the state $|\hat{\sigma}_i^{(m)}\rangle$ satisfying $\| |\pi_{g_{i+1}}\rangle - |\hat{\sigma}_i^{(m)}\rangle \|_2 \leq O(\gamma)$ can be prepared using $\widetilde{O}(n^{1.5})$ calls to the controlled walk operators.

By Lemma 14, each call to the quantum walk operator can be implemented with $O(1)$ query to the evaluation oracle. Hence, the total query complexity is $\widetilde{O}(n^{1.5})$.

The theorem is then proved. □

[4]We can measure the last qubit. If the measurement outcome is 0, we reinitialize the r, r' registers and re-preapre $|y\rangle$ and $|b\rangle$. We repeat this process until we measure $b = 1$.

C.2 Non-destructive rounding in the mid-level

In the middle level, we need to compute the linear transformation Σ_i that rounds the β-logconcave distribution to near-isotropic position. Moreover, we are given access to N copies of the quantum states $|\pi_{g_i}\rangle$ and we will compute Σ_i in a non-destructive way.

Classically, by Lemma 10, we can take

$$\Sigma_i'(x^1, \ldots, x^N) := \frac{1}{N} \sum_{i=1}^{N} x^j i^{j\top},$$

where x_i^j is the j-th independent sample from π_{g_i}. Then, the linear transformation in the i-th iteration is Σ_i composite with the linear transformation in the $(i-1)$-th iteration, i.e.,

$$\Sigma_i(x_1, \ldots, x_n) := \Sigma_i'(x_1, \ldots, x_N) \cdot \Sigma_{i-1}.$$

In quantum, we can use a quantum circuit to simulate the classically computation for Σ_i' and Σ_i coherently, which computes the following superposition state:

$$\int_{\mathcal{K}} \mathrm{d}x^1 \cdots \int_{\mathcal{K}} \mathrm{d}x^N \prod_{j=1}^{N} \sqrt{\pi_{g_i}(x^j)} \cdot \left|x^1\right\rangle \cdots \left|x^N\right\rangle \left|\Sigma_i(x_1, \ldots, x^n)\right\rangle. \tag{14}$$

That is, the first N quantum registers contain N copies of the state $|\pi_{g_i}\rangle$, and the last quantum register contains the linear transformation Σ_i. If we directly measure the last register, we can get the desired matrix, but the coherence of the quantum states $|\pi_{g_i}\rangle$ are also destroyed.

To resolve this issue, we use the following theorem of Harrow and Wei:

Theorem 12 (Non-destructive amplitude estimation, [21])**.** *Let P be an observable. Given state $|\psi\rangle$ and reflections $R_\psi = 2|\psi\rangle\langle\psi| - I$ and $R = 2P - I$, and any $\eta > 0$, there exists a quantum algorithm that outputs $\widetilde{a}$, an approximation to $a := \langle\psi|P|\psi\rangle$, so that*

$$|a - \widetilde{a}| \leq 2\pi \frac{a(1-a)}{M} + \frac{\pi^2}{M^2}.$$

with probability at least $1-\eta$ and $O(\log(1/\eta)M)$ uses of R_ψ and R. Morover the algorithm restores the state $|\psi\rangle$ with probability at least $1-\eta$.

Then, we can create $O(\log N)$ copies of the state in Eq. (14), and non-destructively estimate the mean of the last quantum register via the procedure in [14, 21]. More specifically, we start from $\widetilde{O}(N)$ copies of the states $\left|\pi_{g_{i-1}}\right\rangle$, and evolve them to $|\pi_{g_i}\rangle$. In the same time, the reflection operator R can be approximately implemented by Lemma 7. Then, the mean value can be estimated by Theorem 12. Note that we can estimate all the coordinates of Σ_i in the same time using the non-destructive mean estimation quantum circuit. And we get that the success probability of this procedure is at least $1 - 1/\operatorname{poly}(N)$. After that, the states in the first N registers will be restored. Therefore, we get that:

Lemma 15 (Non-destructive rounding)**.** *For $i \in [K]$, the linear transformation Σ_i at the i-iteration of the annealing process (Algorithm 5) can be obtained using $\widetilde{O}(N)$ copies of the states $\left|\pi_{g_{i-1}}\right\rangle$, with query complexity $\widetilde{O}(N \cdot \mathcal{C})$ where $\mathcal{C}$ is the cost of evolving $\left|\pi_{g_{i-1}}\right\rangle$ to $|\pi_{g_i}\rangle$. Moreover, the states $\left|\pi_{g_{i-1}}\right\rangle$ will be recovered with high probability.*

C.3 Proof of Theorem 1

Proof of Theorem 1. The quantum algorithm for optimizing an approximately convex function is given in Algorithm 6. By Theorem 11 and Lemma 15, we know that it has the same optimization guarantee as the classical procedure (Algorithm 5). Thus, we take $K = \sqrt{n}\log(n/\epsilon)$. And the output x_* of QSIMANNEALING procedure satisfies:

$$F(x_*) - \min_{x \in \mathcal{K}} F(x) \leq O(\epsilon)$$

with high probability.

Algorithm 6 Quantum speedup for approximately convex optimization.

1: **procedure** QSIMANNEALING(K, $\{T_i\}_{i\in[K]}$)
2: $N \leftarrow \widetilde{O}(n)$ ▷ The number of strands
3: Prepare N (approximately) copies of $|\pi_0\rangle$, denoted as $|\widetilde{\pi}_0^{(1)}\rangle, \ldots, |\widetilde{\pi}_0^{(N)}\rangle$, where $\pi_0 = \mathrm{Uniform}(\mathcal{K})$
4: **for** $i \leftarrow 1, \ldots, K$ **do**
5: Use the N copies of the state $|\pi_{i-1}\rangle$ to nondestructively obtain the linear transformation Σ_i. Let $|\hat{\pi}_{i-1}^{(1)}\rangle, \ldots, |\hat{\pi}_{i-1}^{(N)}\rangle$ denote the post-measurements states ▷ Lemma 15
6: Apply quantum walk with Σ_i to evolve the states $|\hat{\pi}_{i-1}^{(1)}\rangle, \ldots, |\hat{\pi}_{i-1}^{(N)}\rangle$ to $|\widetilde{\pi}_i^{(1)}\rangle, \ldots, |\widetilde{\pi}_i^{(N)}\rangle$ ▷ Theorem 11
7: **end for**
8: $x_K^j \leftarrow$ measure the final state $|\widetilde{\pi}_K^{(j)}\rangle$ for $j \in [N]$
9: **return** $\arg\min_{j\in[N]} F(x_K^j)$
10: **end procedure**

Then, consider the query complexity. We have $K = \sqrt{n}\log(n/\epsilon)$ stages in the annealing process. In each iteration, the quantum walk has query complexity $\mathcal{C} = \widetilde{O}(n^{1.5})$ by Theorem 11. Thus, the query cost of Line 5 is $\widetilde{O}(N\mathcal{C}) = \widetilde{O}(n^{2.5})$. Also, the query cost of Line 6 is also $\widetilde{O}(n^{2.5})$. Therefore, the total query complexity of the annealing procedure is

$$K \cdot \widetilde{O}(n^{2.5}) = \widetilde{O}(n^3).$$

□