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

Anonymous Author(s)
Affiliation
Address
email

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 $\sup_{x\in\mathcal{K}} |F(x) - f(x)| \le \epsilon/n$, our quantum algorithm finds an $x^* \in \mathcal{K}$ such that $F(x^*) - \min_{x\in\mathcal{K}} F(x) \le \epsilon$ using $\tilde{O}(n^3)$[1] 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

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

Throughout the paper, we assume that f is L-Lipschitz with respect to ℓ_∞ norm, i.e., $|f(x)-f(y)| \le L\|x-y\|_\infty$ 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 $F(x^*) -$

[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.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

$\min_{x\in\mathcal{K}} F(x) \le \epsilon$ with high probability was proposed by Belloni et al. [8], which takes $\tilde{O}(n^{4.5})$ queries to the zeroth-order oracle. On the other hand, Ref. [32] 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, 32].

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 [40]. 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 [39], 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, 39]:

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

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 this 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:

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

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).*

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* 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.

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*

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

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 stated 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 $\tilde{O}(n^{4.5}\sqrt{T})$ [23], 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. To achieve this, 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 [23] 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 [28].

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

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

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.,

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

In quantum, we can also define the discrete-time quantum walk [34], 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, 34, 35] for more details.

2.3 Hit-and-Run walk

Hit-and-Run walk was introduced by R.L. Smith [33], and has a long line of reserach [1, 7, 9, 16, 24–27, 38] 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 ([24])**.** *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 [34, 35]. 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:

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

where $|\psi_x\rangle := \int_{\mathbb{R}^n}\sqrt{P(y,x)}\,|y\rangle\,\mathrm{d}y$ mixes all the points that can move to 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 [22], 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 require some additional complicated constraints on the Markov chain and distributions (see e.g., [2, 13, 29, 37]). Based on previous studies on quantum walk mixing, we propose the following user-friendly quantum walk framework:

[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.

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:** $d_{\mathrm{TV}}(P_1^{t_0} \cdot \pi_0, \pi_1) \leq \epsilon$ *and* $d_{\mathrm{TV}}(P_0^{t_1} \cdot \pi_1, \pi_0) \leq \epsilon$.
- **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 $\| \, |\widetilde{\pi}_1\rangle - |\pi_1\rangle \, \|_2 \leq \epsilon$ 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 $0 \leq i \leq K-1$,*

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

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 $0 \leq i \leq K-1$,*

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

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 $0 \leq i \leq K$. 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 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:

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

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

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

□

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

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.

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 $f(x_t) - f(x_{\min}) \le Cn\Delta(3r_{t-1}))$ for some global constant $C > 0$. Due to strong convexity, this gives a new radius r_t recursively:

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

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 $\tau \ge \log(1/\Delta)$, there is a quantum algorithm that outputs a mean estimate $\tilde{\mu}$ such that*

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

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). □

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($\mathcal{O}_f^{\mathsf{stoc}}, L/K$) ▷ 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

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

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

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

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

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

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

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:

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

□

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

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

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

[25] 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.

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

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

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

[29] 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.

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

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

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

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

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

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

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

[37] 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.

[38] 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.

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

[40] 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]