Quantum Speedups for Sampling and Non-Convex Optimization with Stochastic Oracles

Guneykan Ozgul
Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA
Global Technology Applied Research, JPMorgan Chase, New York, NY, USA

Xiantao Li
Department of Mathematics, Pennsylvania State University, University Park, PA, USA

Mehrdad Mahdavi
Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA

Chunhao Wang
Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA

Abstract

We present quantum speedups for sampling from distributions of the form $\pi \propto e^{-f}$ on $\mathbb{R}^d$. We consider two stochastic oracle models: a stochastic gradient oracle, where $f = \frac{1}{n}\sum_{i=1}^{n} f_i$ and component gradients $\{\nabla f_i\}_{i\in[n]}$ are available, and a stochastic evaluation oracle, where only noisy values of f are available. Our framework accelerates classical stochastic Langevin Monte Carlo (LMC) and Hamiltonian Monte Carlo (HMC) algorithms by replacing stochastic gradient estimators with variance-controlled quantum mean estimation and gradient estimation subroutines. Unlike quantum walk based approaches, our algorithms do not require reversibility or exact gradients, and they preserve the structure of the underlying Markov chain. In the finite-sum setting, quantum mean estimation combined with classical variance-reduction techniques improves the stochastic gradient-query complexity for the approximate sampling task. In the stochastic zeroth-order setting, we develop gradient estimators robust to noisy function evaluations, yielding improved evaluation complexity for LMC and HMC. These results apply to strongly log-concave and/or non-log-concave distributions satisfying a log-Sobolev inequality, with convergence guarantees in Wasserstein distance and Kullback–Leibler divergence. We also show that faster sampling methods lead to quantum speedups for optimization, including for non-smooth and approximately convex objectives.

2012 ACM Subject Classification Theory of computation → Quantum computation theory; Mathematics of computing → Markov-chain Monte Carlo methods; Mathematics of computing → Stochastic processes

Keywords and phrases Quantum algorithms, sampling, Langevin Monte Carlo, Hamiltonian Monte Carlo, stochastic oracles, non-convex optimization

Digital Object Identifier 10.4230/LIPIcs.TQC.2026.8

Related Version *Full Version*: https://arxiv.org/abs/2504.03626 [37]

Funding *Guneykan Ozgul*: National Science Foundation grant CCF-2238766 (CAREER).
Xiantao Li: National Science Foundation grants DMS-2111221 and CCF-2312456.
Chunhao Wang: National Science Foundation grant CCF-2238766 (CAREER).

1 Introduction

1.1 Motivation

Efficient sampling from complex distributions is a fundamental problem in many scientific and engineering disciplines and it is becoming increasingly important as modern applications deal with high-dimensional data and complex probabilistic models. In machine learning, sampling plays an important role in Bayesian inference, as it facilitates posterior estimation and uncertainty quantification [47, 46, 22, 39]. In non-convex optimization, sampling allows for the exploration of complex energy landscapes and helps avoid local minima, facilitating progress in tasks such as resource allocation, scheduling, and hyperparameter tuning [51, 13]. It is also widely used in other domains of science, such as statistical mechanics [10, 23], and convex geometry [31, 18].

Given a potential function $f : \mathbb{R}^d \to \mathbb{R}$, we consider the problem of approximately sampling from *Gibbs-Boltzmann distribution* $\pi \in L^1(\mathbb{R}^d, \mathrm{d}\mathbf{x})$ of the form

$$\pi(\mathbf{x}) = \frac{e^{-f(\mathbf{x})}}{\int e^{-f(\mathbf{x})}\mathrm{d}\mathbf{x}}. \tag{1}$$

Solving this problem is in general difficult as it encodes NP-hard problems, and efficient sampling is only possible when f is highly structured. Given the challenging nature of the problem, it is natural to ask whether quantum computers can provide any speedup for the sampling task. The state-of-the-art classical algorithms are based on Markov chains, and this is a domain where quantum algorithms have been studied extensively [43, 48, 1]. In the continuous setting, Childs et al. [16] initiated the study of quantum algorithms for sampling from log-concave distributions, assuming that f is strongly convex and that ∇f can be accessed to very high precision. Their construction uses the quantum-walk framework of [43] together with fixed-point amplitude amplification similar to [48]. Although this approach improves the best classical dependence on certain parameters, the speedup does not directly apply in the stochastic setting because the relevant Markov chains are not *reversible*. Ozgul et al. [36] later addressed this limitation by analyzing a reversible proxy chain; as a result, the speedup is obtained relative to a suboptimal version of the best classical sampler.

Moreover, existing quantum results in this line primarily guarantee convergence in total variation distance. Obtaining quantum speedups with guarantees in Wasserstein distance or Kullback-Leibler divergence, metrics that are standard in classical sampling theory, remains largely open. Motivated by these challenges, we ask the following question:

▶ **Question 1.** *Can quantum computers improve the oracle complexity of sampling algorithms such that*

1. *the algorithm can use stochastic gradient and/or noisy evaluation oracles,*
2. *it can be based on fast (potentially nonreversible) Markov chains rather than a reversibilized surrogate, and*
3. *it provides guarantees in Wasserstein distance and Kullback–Leibler divergence?*

We answer Question 1 affirmatively by introducing a framework that replaces stochastic gradient computation with variance-controlled quantum subroutines, yielding provable oracle-complexity improvements while keeping the underlying nonreversible Markov chain unchanged. Conceptually, our contribution is not a new sampler, but a principled way to accelerate existing stochastic samplers by exploiting variance structure (via quantum mean/gradient estimation) rather than relying on restrictive quantum-walk techniques.

Another motivation for this work is to investigate whether quantum speedups for certain optimization problems can be obtained via our sampling algorithms. Sampling and optimization are tightly connected: sampling from a Gibbs distribution with potential βf at large inverse temperature β concentrates around minimizers of f. Although using sampling as a generic optimization method is not always optimal, it can be effective in regimes where gradient information is noisy or the objective is non-smooth/non-convex. To this end, we ask the following question:

▶ **Question 2.** *Can we use the quantum sampling algorithms to recover the existing quantum speedups for certain optimization problems and/or identify new classes of optimization problems that can be accelerated through sampling?*

We also answer Question 2 affirmatively by designing optimization algorithms based on our quantum samplers. In the finite-sum setting, we recover the quantum speedup of [50]; in the zeroth-order setting, we improve some parameter dependencies for non-smooth optimization problems studied in [29, 9]. While most of the resulting improvements are sub-quadratic, the point of our work is not to focus on the degree of the speedup for these particular problems. More broadly, we highlight that viewing optimization through the lens of sampling can be a useful organizing principle for identifying future quantum speedups. The key point is that sampling becomes competitive with standard optimization methods in regimes where (i) gradients are unavailable or prohibitively expensive, (ii) the objective is non-smooth (the gradients are not Lipschitz continuous), or (iii) the landscape is non-convex and escaping shallow traps is important. In these regimes, a low-temperature Gibbs distribution can serve as a powerful tool to approach these problems.

This perspective is also useful from a quantum-algorithmic standpoint for two related reasons. First, many standard first-order optimization routines (e.g., gradient descent) do not admit meaningful generic quantum speedups in the usual black-box models. Second, the sampling problem offers a more natural ground for quantum algorithms compared to deterministic iterative procedures. Moreover, in non-convex settings, which are a primary target for quantum acceleration, sampling-based methods are shown to be particularly effective [32].

We use Langevin Monte Carlo (LMC) and Hamiltonian Monte Carlo (HMC) algorithms as base samplers. LMC is the Euler-Maruyama discretization of the Langevin diffusion

$$\mathrm{d}\mathbf{x}_t = -\nabla f(\mathbf{x}_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t.$$

With a stochastic gradient $\mathbf{g}_k$, the update at each iteration is

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\mathbf{g}_k + \sqrt{2\eta}\boldsymbol{\epsilon}_k, \tag{2}$$

where $\boldsymbol{\epsilon}_k \sim \mathcal{N}(0, I_d)$ is distributed normally. HMC augments the state with a momentum $\mathbf{p}$ and simulates the Hamiltonian $H(\mathbf{x}, \mathbf{p}) = f(\mathbf{x}) + \|\mathbf{p}\|^2/2$. We give more detailed overview on these sampling algorithms in Appendix A. While they are effective, both algorithms are sensitive to gradient noise: smaller variance allows larger steps or fewer iterations, but producing a lower-variance gradient estimator costs more oracle queries.

In the finite-sum setting, a stochastic gradient can be obtained by sampling an index $i \in [n]$ and using $\nabla f_i(\mathbf{x})$ as an approximation to $\nabla f(\mathbf{x})$. This estimator is unbiased, but because of the large variance, the error due to stochasticity may be too large for sampling unless the step size is made very small. In zeroth-order settings, stochastic gradients are often built from finite differences or Gaussian smoothing. Such estimators have an additional dimension-dependent cost because a classical method must probe enough directions to

Algorithm 1 Stochastic-gradient LMC and HMC templates.

input Current point $\mathbf{x}_0$, step size η, number of leapfrog steps S (for HMC only), stochastic gradient routine $\mathbf{g}(\cdot)$.

output One step LMC update or HMC proposal.

1: **LMC step:** sample $\boldsymbol{\epsilon}_k \sim \mathcal{N}(0, I_d)$ and set

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \mathbf{g}(\mathbf{x}_k) + \sqrt{2\eta}\boldsymbol{\epsilon}_k.$$

2: **HMC proposal:** set $\mathbf{q}_0 = \mathbf{x}_0$ and sample $\mathbf{p}_0 \sim \mathcal{N}(0, I_d)$.

3: **for** $s = 0, \ldots, S-1$ **do**

4: Perform the leapfrog updates

$$\mathbf{q}_{s+1} = \mathbf{q}_s + \eta \mathbf{p}_s - \frac{\eta^2}{2}\mathbf{g}(\mathbf{q}_s), \qquad \mathbf{p}_{s+1} = \mathbf{p}_s - \frac{\eta}{2}\mathbf{g}(\mathbf{q}_s) - \frac{\eta}{2}\mathbf{g}(\mathbf{q}_{s+1}).$$

5: **end for**

6: **Return** the LMC point $\mathbf{x}_{k+1}$ or the HMC proposal $\mathbf{q}_S$.

recover a d-dimensional gradient. The quantum algorithms in this paper improve these two bottlenecks: the finite-sum algorithms reduce the cost of estimating $\nabla f = \frac{1}{n}\sum_{i=1}^n \nabla f_i$ using queries to $\{\nabla f_i\}_{i\in[n]}$, while the zeroth-order algorithms reduce the cost of estimating ∇f using queries to an oracle that returns f with some noise.

1.2 Quantum Oracle Model

Here, we describe our quantum input model and assumptions.

- The *stochastic gradient oracle*. In this setting, we assume $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x})$, and we assume access to the gradients of each component, i.e., $\nabla f_i(x)$ for any $i \in [n]$. This is a relevant setting in statistics and machine learning problems, where access to the function f is provided only through sample data [47]. For each $i \in [n]$, we consider the quantum oracle access of the form

$$O_{\nabla f} |\mathbf{x}\rangle |i\rangle |0\rangle \mapsto |\mathbf{x}\rangle |i\rangle |\nabla f_i(\mathbf{x})\rangle . \tag{3}$$

- The *stochastic evaluation oracle*. In this setting, we assume that we have access only to noisy function values $f(\mathbf{x}; \xi)$, where ξ is the random seed that characterizes the noise. This setting is more relevant in contexts where the function value f is replaced by a statistical estimate, perhaps obtained through a numerical simulation [39]. The function is accessed through the binary oracle below.

$$O_f |\mathbf{x}\rangle |\xi\rangle |0\rangle \mapsto |\mathbf{x}\rangle |\xi\rangle |f(\mathbf{x}; \xi)\rangle . \tag{4}$$

  Since this model only uses function values, we use the terms "zeroth-order oracle" and "evaluation oracle" interchangeably. We note that the evaluation oracle requires coherent access to the random seed ξ and that the same seed can be queried at multiple points. Although this model is stronger than a standard classical noisy oracle, we show that it naturally captures non-smooth optimization settings. This assumption is also explicit in the classical baseline work [39].

Both oracles can be implemented by making the corresponding classical oracles reversible with additional registers, so their query cost is comparable to the classical setting. As our study spans multiple settings under various assumptions, we state the precise noise assumptions separately for each setting.

1.3 Main Contributions

We summarize our technical contributions as follows.

Speedups for sampling in the stochastic gradient oracle setting. Using quantum mean estimation together with classical variance-reduction techniques such as stochastic variance-reduced gradients (SVRG) and control variates (CV), we obtain improved query complexities for sampling from both convex and non-convex potentials in Section 2. The main challenge is that these algorithms periodically compute the full gradient to control the variance of the stochastic gradients, and this full-gradient cost can dominate the improvement obtained from quantum mean estimation. To address this, we analyze the variance of the stochastic gradients and show that one can choose the step size and target variance online so that full-gradient computations are needed less frequently when quantum variance reduction is used. We also use these samplers to construct optimization algorithms for strongly convex functions.

Quantum speedups for gradient estimation via stochastic evaluation oracle. The challenge in this setting is that existing gradient estimation algorithms assume that the evaluation oracle is almost exact. However, in the setting we consider, the error between the estimated function value and the true function value can be unbounded. To address this, we develop quantum gradient estimation algorithms under several smoothness assumptions in Section 3. When the potential is smooth, our estimator reduces the number of evaluation queries from $\tilde{\mathcal{O}}(d^2\sigma^2/\epsilon^2)$ to $\tilde{\mathcal{O}}(d\sigma/\epsilon)$ for computing an ϵ-accurate gradient estimate (Theorem 3.4), where σ^2 is the noise variance in Assumption 3.1. When the stochastic functions are also smooth with high probability, we improve the dimension dependence by an additional factor of $d^{1/2}$ (Theorem 3.7). Our gradient estimation algorithms could be useful as independent tools, especially in zeroth-order stochastic optimization.

Speedups for stochastic zeroth-order sampling. In Section 4, we combine our quantum gradient estimation algorithm with HMC and LMC and analyze the convergence of the resulting samplers. We prove that, under the same assumptions, these hybrid algorithms use fewer queries to the evaluation oracle than the best-known classical samplers (Theorems 4.1 and 4.2).

Application to non-smooth non-convex optimization. In Section 5, we extend our quantum sampling methods to optimize non-convex functions with specific structural properties without the smoothness assumption. In particular, we show that Lipschitz continuous and approximately convex functions can be optimized with fewer stochastic-evaluation queries than the best-known classical algorithms in terms of dimension dependence (Theorem 5.4).

To help the readers navigate, we summarize the problems, oracles, assumptions, and results in Table 1.

1.4 Related Work

Non-asymptotic convergence rates for stochastic LMC and HMC have been analyzed extensively by [38, 49, 54, 19] and [12, 52], respectively. In the finite-sum setting, more sophisticated variance-reduction techniques, such as SVRG, SAGA, SARAH, and control variates (CV) [26, 20, 35, 3], have been used to reduce stochastic-gradient variance by using gradient information from previous iterations. Although these methods were originally

Table 1 Overview of the main results. The table separates assumptions on the oracle/noise model from assumptions on the target distribution or optimization objective.

Problem	Oracle	Oracle/noise assumptions	Target or objective assumptions	Results
Sampling	Stochastic gradient	Lipschitz stochastic gradients	Strong convexity and Smoothness	Theorems 2.6 and 2.7
Sampling	Stochastic gradient	Lipschitz stochastic gradients	Log-Sobolev inequality and Smoothness	Theorem 2.3
Sampling	Stochastic evaluation	Bounded noise variance	Strong convexity and Smoothness	Theorem 4.1
Sampling	Stochastic evaluation	Bounded noise variance	Log-Sobolev inequality and Smoothness	Theorem 4.2
Optimization	Stochastic evaluation	Uniform error bound	Approximate convexity and Lipschitz continuity	Theorem 5.4

introduced in the context of optimization, successive works have applied these methods to improve sampling efficiency via LMC [21, 11, 3, 28] and HMC [53, 52]. In particular, [52] has incorporated various variance reduction techniques to SG-HMC and analyzed convergence in Wasserstein distance for smooth and strongly convex potentials. In the non-log-concave setting, [28] has analyzed the convergence of SVRG-LMC and SARAH-LMC for target distributions that satisfy the log-Sobolev inequality and applied their results to optimize structured non-convex objectives.

In quantum sampling, most existing results use quantum walks, which can speed up certain Markov Chain Monte Carlo (MCMC) methods by improving the mixing time of the underlying Markov chain [43, 41, 42, 48, 6]. These methods have been used in several domains to reduce the computational time of sampling-related tasks [33, 1, 16, 6, 29, 8]. For continuous distributions, [16, 36] used the quantum-walk framework to reduce the number of gradient queries needed to approximately sample from strongly log-concave and non-log-concave distributions.

In the context of optimization, quantum algorithms such as multi-dimensional quantum mean estimation [17] and quantum gradient estimation [27, 24] have shown promise in reducing the computational cost associated with gradient-based methods [44, 7, 40, 50, 30]. These techniques are particularly well-suited for addressing challenges in large-scale and noisy settings, as they can provide more accurate gradient estimates with fewer queries. However, these methods have not been considered for sampling tasks, and this paper focuses on integrating these quantum techniques to enhance the efficiency of stochastic gradient-based samplers and alleviate the computational burden inherent in classical methods.

1.5 Notation and Glossary

Bold symbols, such as $\mathbf{x}$ and $\mathbf{y}$, are used to represent vectors, with $\|\cdot\|$ indicating the Euclidean or operator norm depending on the context. Given two scalars a and b, we use $a \wedge b$ to denote $\min\{a, b\}$ and use $a \vee b$ to denote $\max\{a, b\}$. We use $\mathcal{B}_d(c, r)$ to denote the d-dimensional ball centered at c with radius r and $G_d^l(c)$ to denote the d-dimensional grid centered at point c with side length l. We occasionally use G_d^l when the center of the grid is clear from the context. The notation $\tilde{O}$ is used to suppress the polylogarithmic dependencies on d, ϵ, L, μ and α that will be defined later in the text.

We use several metrics to compare probability distributions over a state space $\mathcal{X}$. Let π and μ be two probability distributions on $\mathcal{X}$. The p-Wasserstein distance between π and μ is defined as $\mathrm{W}_p(\pi,\mu) = \left(\inf_{\gamma\in\Gamma(\pi,\mu)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma}\|\mathbf{x}-\mathbf{y}\|^p\right)^{1/p}$ where $\Gamma(\pi,\mu)$ is the set of all joint distributions $\gamma(\mathbf{x},\mathbf{y})$ whose marginals are π and μ. The KL divergence of π with respect to μ is defined as $\mathrm{KL}(\pi\|\mu) = \int_{\mathcal{X}} \mathrm{d}\mathbf{x}\pi(\mathbf{x}) \log\left(\frac{\pi(\mathbf{x})}{\mu(\mathbf{x})}\right)$ and the relative Fisher information is $\mathrm{FI}(\pi\|\mu) = \int_{\mathcal{X}} \mathrm{d}\mathbf{x}\pi(\mathbf{x}) \left\|\nabla \log\left(\frac{\pi(\mathbf{x})}{\mu(\mathbf{x})}\right)\right\|^2$. The total variation distance is defined as $\mathrm{TV}(\pi,\mu) = \sup_{A\subseteq\mathcal{X}} |\pi(A)-\mu(A)| = \frac{1}{2}\int_{\mathcal{X}} \mathrm{d}\mathbf{x}|\pi(\mathbf{x})-\mu(\mathbf{x})|$.

1.6 Organization of the paper

In Section 2, we present our quantum sampling algorithms for stochastic gradient oracles. In Section 3, we introduce the stochastic zeroth-order gradient estimation algorithm that serves as the main technical tool for the sampling results in Section 4. Finally, in Section 5, we show how these sampling results can be used to speed up non-convex optimization.

2 Quantum Speedups for Finite-Sum Sampling and Optimization via Gradient Oracle

Throughout this section, we consider $f = \frac{1}{n}\sum_{i=1}^n f_i$, and use the quantum stochastic-gradient oracle in Equation (3). Our goal is to sample approximately from π using as few oracle queries as possible. We first introduce stochastic-gradient estimators that combine unbiased quantum mean estimation with classical variance-reduction frameworks, yielding improved query complexity for both LMC and HMC. Classically, a stochastic gradient at $\mathbf{x}_t$ is often formed by subsampling a mini-batch $B \subseteq [n]$ and setting $\mathbf{g}_t = |B|^{-1}\sum_{i\in B}\nabla f_i(\mathbf{x}_t)$. Then $\mathbb{E}_B[\mathbf{g}_t] = \nabla f(\mathbf{x}_t)$, so $\mathbf{g}_t$ is an unbiased estimator. If LMC or HMC takes T iterations to converge, the resulting query complexity is $\mathcal{O}(|B|T)$, and there is a tradeoff between the batch size $|B|$ and the number of iterations T.

To decouple these quantities, variance-reduction methods such as SVRG periodically compute a full gradient $\nabla f(\tilde{\mathbf{x}})$ and use it to correct later stochastic gradients. Control variates (CV) use a similar correction based on the initial gradient $\nabla f(\mathbf{x}_0)$. Our quantum LMC algorithms preserve this structure, but replace mini-batch averaging with unbiased quantum mean estimation; see Algorithms 2 and 3.

For `QSVRG`, the epoch length m determines how often the full gradient is computed. The parameter b is the quantum analogue of the batch size and is chosen analytically. In `QCV`, the stochastic gradient is corrected using the initial iterate, as in classical control variates. We also highlight that quantum mean estimation requires a target variance level as input, unlike classical subsampling. Although the optimal target variance changes along the trajectory and depends on quantities that are difficult to compute, `QSVRG` uses $L^2\|\mathbf{x}_k-\tilde{\mathbf{x}}\|^2/b^2$, where $\tilde{\mathbf{x}}$ is the most recent point at which the full gradient was computed (Algorithm 2). Similarly, `QCV` uses $L^2\|\mathbf{x}_k-\mathbf{x}_0\|^2/b^2$, where $\mathbf{x}_0$ is the initial point (Algorithm 3). Note that we only use `QCV` for HMC but (Algorithm 3) uses LMC update for ease of exposition. Changing the update rule with HMC proposal in Algorithm 1 implements `QCV-HMC`.

Although the algorithm is simple, proving the speedup is more delicate because the cost of full-gradient computations must also be reduced. We show that quantum mean estimation enables less frequent full-gradient computations through improved variance reduction. Our analysis quantifies stochastic-gradient variance along the algorithm trajectory and balances the frequency of full-gradient computations with the target variance used in quantum mean

Algorithm 2 `QSVRG-LMC`.

input Stochastic gradient oracle $O_{\nabla f}$, initial point $\mathbf{x}_0$, number of iterations K, step size η, smoothness constant L, variance scale factor b, epoch length m.

output Final iterate $\mathbf{x}_K$.

Set $\tilde{\mathbf{x}} = \mathbf{x}_0$ and $\tilde{\mathbf{g}} = \nabla f(\tilde{\mathbf{x}})$.
for $k = 0, \ldots, K-1$ **do**
  if $k \bmod m = 0$ **then**
    Set $\tilde{\mathbf{x}} = \mathbf{x}_k$ and $\tilde{\mathbf{g}} = \nabla f(\tilde{\mathbf{x}})$.
    Set $\mathbf{g}_k = \tilde{\mathbf{g}}$.
  else
    Define the quantum sampling oracle $O_{\mathrm{SVRG}}^{\mathbf{x}_k, \tilde{\mathbf{x}}}$ by

$$|0\rangle |0\rangle \mapsto \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |\nabla f_i(\mathbf{x}_k) - \nabla f_i(\tilde{\mathbf{x}}) + \tilde{\mathbf{g}}\rangle |i\rangle .$$

    Set $\hat{\sigma}_k^2 = L^2 \|\mathbf{x}_k - \tilde{\mathbf{x}}\|^2 / b^2$.
    Set $\mathbf{g}_k = \texttt{QuantumMeanEstimation}(O_{\mathrm{SVRG}}^{\mathbf{x}_k, \tilde{\mathbf{x}}}, \hat{\sigma}_k^2)$.
  end if
  Sample $\boldsymbol{\epsilon}_k \sim \mathcal{N}(0, I_d)$.
  Set $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \mathbf{g}_k + \sqrt{2\eta}\boldsymbol{\epsilon}_k$.
end for
Return $\mathbf{x}_K$.

estimation. This yields improved complexity bounds in both strongly convex and non-convex settings. We first present the LMC result without assuming convexity; later, we consider HMC with similar stochastic-gradient estimators to further improve some parameter dependencies.

▶ **Assumption 2.1** (Lipschitz Stochastic Gradients)**.** There exists a positive constant L such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and all functions f_i, $i = 1, ..., n$, it holds that

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|. \tag{5}$$

This is a standard assumption in the optimization and sampling literature [45, 32].

2.1 Sampling under Log-Sobolev Inequality via Langevin Monte Carlo

We use the SVRG-LMC algorithm of [28] as the base sampler and replace its stochastic-gradient estimator with the quantum estimator in `QSVRG-LMC`; see Algorithm 2. In this section, we assume that the target distribution satisfies a log-Sobolev inequality.

▶ **Assumption 2.2** (Log-Sobolev Inequality)**.** We say that π satisfies the log-Sobolev inequality with constant α if for all ρ, it holds that

$$\mathrm{KL}(\rho||\pi) \leq \frac{1}{2\alpha}\mathrm{FI}(\rho||\pi). \tag{6}$$

The log-Sobolev inequality is a sampling analog of the Polyak-Łojasiewicz (PL) condition used in optimization [15]. It is standard in the non-log-concave sampling literature [45, 32, 14, 28]. Moreover, if f is μ-strongly convex, then π satisfies an LSI with constant μ. We also note that this assumption is weaker than the dissipative gradient condition [38, 53] which is commonly used in non-log-concave sampling.

Algorithm 3 `QCV-LMC`.

input Stochastic gradient oracle $O_{\nabla f}$, initial point $\mathbf{x}_0$, number of iterations K, step size η, smoothness constant L, variance scale factor b.

output Final iterate $\mathbf{x}_K$.

Set $\mathbf{g}^{(0)} = \nabla f(\mathbf{x}_0)$.
for $k = 0, \dots, K-1$ **do**
  Define the quantum sampling oracle $O_{\mathrm{CV}}^{\mathbf{x}_k, \mathbf{x}_0}$ by

$$|0\rangle |0\rangle \mapsto \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |\nabla f_i(\mathbf{x}_k) - \nabla f_i(\mathbf{x}_0) + \mathbf{g}^{(0)}\rangle |i\rangle .$$

  Set $\hat{\sigma}_k^2 = L^2 \|\mathbf{x}_k - \mathbf{x}_0\|^2 / b^2$.
  Set $\mathbf{g}_k = \texttt{QuantumMeanEstimation}(O_{\mathrm{CV}}^{\mathbf{x}_k, \mathbf{x}_0}, \hat{\sigma}_k^2)$.
  Sample $\boldsymbol{\epsilon}_k \sim \mathcal{N}(0, I_d)$.
  Set $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \mathbf{g}_k + \sqrt{2\eta}\boldsymbol{\epsilon}_k$.
end for
Return $\mathbf{x}_K$.

▶ **Theorem 2.3** (Main Theorem for `QSVRG-LMC`). *Let μ_k be the distribution of $\mathbf{x}_k$ in* `QSVRG-LMC` *algorithm. Suppose that f satisfies Assumptions 2.1 and 2.2. Then for $\eta = \mathcal{O}\left(\frac{\epsilon\alpha}{dL^2} \wedge \frac{\alpha}{L^2 m}\right)$, $K = \tilde{\mathcal{O}}\left(\frac{L^2 \log(\mathrm{KL}(\mu_0 \| \pi))}{\alpha^2}\left(n^{2/3} + \frac{d}{\epsilon}\right)\right)$, $b = \tilde{\mathcal{O}}(n^{1/3})$, and $m = \tilde{\mathcal{O}}(n^{2/3})$ we have*

$$\left\{\mathrm{KL}(\mu_K \| \pi), \mathrm{TV}(\mu_K, \pi)^2, \frac{\alpha}{2}\mathrm{W}_2(\mu_K, \pi)^2\right\} \leq \epsilon.$$

The stochastic-gradient oracle query complexity is $\tilde{\mathcal{O}}\left(\frac{L^2 \log(\mathrm{KL}(\mu_0 \| \pi))}{\alpha^2}\left(nd^{1/2} + \frac{d^{3/2} n^{1/3}}{\epsilon}\right)\right)$.

The query complexity in [28] is $\mathcal{O}(n + n^{1/2}\epsilon^{-1})$, whereas ours is $\mathcal{O}(n + n^{1/3}\epsilon^{-1})$. The coupled term $n^{1/3}\epsilon^{-1}$ is possible because the stochastic-gradient cost Kb and the full-gradient cost nK/m are balanced. In classical SVRG-LMC, the optimal choice is $m = b = n^{1/2}$, where b is the inner batch size; in our algorithm, we have $m = n^{2/3}$ and $b = n^{1/3}$. Thus, quantum gradient estimators allow the exact gradient to be computed less frequently, which is the main mechanism behind the speedup.

2.2 Application to Optimization

In this section, we derive a finite-sum optimization algorithm by running our sampler at low temperature, equivalently by scaling f by a large inverse-temperature parameter β. The following theorem quantifies how large β should be to find an approximate optimizer of f.

▶ **Theorem 2.4** (Quantum Finite-sum Optimization Upper Bound). *Suppose f is a μ-strongly convex function. Then, there exists a quantum algorithm that outputs a point $\widetilde{\mathbf{x}}$ such that $\mathbb{E}[f(\widetilde{\mathbf{x}}) - f(\mathbf{x}^\star)] \leq \epsilon$ using $\tilde{\mathcal{O}}\left(\frac{L^2 d}{\mu^2}\left(nd^{1/2} + \frac{d^{3/2} n^{1/3}}{\epsilon}\right)\right)$ queries to the quantum stochastic gradient oracle.*

This matches the upper bound of [50] in its n-dependence. The remaining parameter dependencies are suboptimal; this is expected because, in the strongly convex setting, sampling is not the most efficient optimization method in terms of parameters such as L and μ. We have simplified the proof by assuming strong convexity, but one can also impose

additional conditions, such as the Morse-type conditions used in [28], to obtain a quantum upper bound of the form $\mathcal{O}(n+n^{1/3}\epsilon^{-1})$ for a broader class of non-convex functions satisfying an LSI.

2.3 Sampling under Strong Convexity via Hamiltonian Monte Carlo

Next, we consider quantum speedups for Hamiltonian Monte Carlo (HMC) for strongly convex potentials. We replace the stochastic gradients in HMC with quantum-estimated stochastic gradients based on the SVRG and CV estimators used in Algorithms 2 and 3.

▶ **Assumption 2.5** (Strong Convexity)**.** There exists a positive constant μ such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ it holds that

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2. \tag{7}$$

For smooth objectives, we occasionally use the condition number $\kappa = L/\mu$ to express the dependence on L and μ more compactly.

Based on Assumptions 2.1 and 2.5, we give the main theorem for the quantum Hamiltonian Monte Carlo algorithm implemented with the quantum SVRG estimator.

▶ **Theorem 2.6** (Main Theorem for `QSVRG-HMC`)**.** *Let μ_k be the distribution of $\mathbf{x}_k$ in* `QSVRG-HMC` *algorithm. Suppose that f satisfies Assumptions 2.1 and 2.5. Given that the initial point $\mathbf{x}_0$ satisfies $\|\mathbf{x}_0 - \arg\min_{\mathbf{x}} f(\mathbf{x})\| \leq \frac{d}{\mu}$, then, for $\eta = \mathcal{O}\left(\frac{\epsilon}{L^{1/2}d^{1/2}\kappa^{3/2}}\right)$, $S = \tilde{\mathcal{O}}\left(\frac{Ld^{1/2}\kappa^{3/2}}{\epsilon}\right)$, $T = \tilde{\mathcal{O}}(1)$, $b = \mathcal{O}\left(\frac{L^{1/8}\epsilon^{1/4}n^{1/2}}{d^{1/8}\kappa^{3/8}} \vee 1\right)$, and $m = n/b$, we have*

$$\mathrm{W}_2(\mu_{ST}, \pi) \leq \epsilon.$$

The total query complexity to the stochastic gradient oracle is $\tilde{\mathcal{O}}\left(\frac{Ld^{1/2}\kappa^{3/2}}{\epsilon} + \frac{L^{9/8}d^{7/8}\kappa^{3/4}n^{1/2}}{\epsilon^{3/4}}\right)$.

We obtain an analogous guarantee for HMC with the quantum control-variate estimator.

▶ **Theorem 2.7** (Main Theorem for `QCV-HMC`)**.** *Let μ_k be the distribution of $\mathbf{x}_k$ in* `QCV-HMC` *algorithm. Suppose that f satisfies Assumptions 2.1 and 2.5. Given that the initial point $\mathbf{x}_0$ satisfies $\|\mathbf{x}_0 - \arg\min_{\mathbf{x}} f(\mathbf{x})\| \leq \frac{d}{\mu}$, then, for $\eta = \mathcal{O}\left(\frac{\epsilon}{L^{1/2}d^{1/2}\kappa^{3/2}}\right)$, $S = \tilde{\mathcal{O}}\left(\frac{Ld^{1/2}\kappa^{3/2}}{\epsilon}\right)$, $T = \tilde{\mathcal{O}}(1)$, and $b = \mathcal{O}\left(\frac{d^{1/4}\kappa^{3/4}}{L^{1/4}\epsilon^{1/2}} \vee 1\right)$, we have*

$$\mathrm{W}_2(\mu_{ST}, \pi) \leq \epsilon.$$

The total query complexity to the stochastic gradient oracle is $\tilde{\mathcal{O}}\left(\frac{Ld^{5/4}\kappa^{9/4}}{\epsilon^{3/2}}\right)$.

The proofs follow the same idea as in the LMC setting. We express the stochastic-gradient variance σ along the HMC trajectory in terms of b and the distance between the current iterate and the last iterate at which the full gradient was computed. We then choose b so that the per-iteration cost of quantum mean estimation, $\tilde{\mathcal{O}}(\sigma/\hat{\sigma})$, balances the amortized full-gradient calculation cost, $\tilde{\mathcal{O}}(n/m)$, since the full gradient is computed only once every m iterations.

Theorems 2.6 and 2.7 imply that when $n = \mathcal{O}(\epsilon^{-1/2})$ the best classical (`SVRG-HMC`) and the best quantum (`QSVRG-HMC`) algorithms have $\tilde{\mathcal{O}}(\epsilon^{-1})$ gradient complexity. On the other hand, when $n = \omega(\epsilon^{-1})$, quantum algorithms have better complexity than the best classical algorithms; the better of `QSVRG-HMC` and `QCV-HMC` depends on the size of n.

Table 2 Summary of the results in the **stochastic gradient oracle setting** (some of the previous results use a different scaling of f and we convert the results to the same scaling as ours in the table). Here, we mainly focus on n and ϵ dependency. See Theorems 2.3, 2.6, and 2.7 for explicit dependencies on L, μ, α, d.

Algorithm	Assumptions	Metric	Gradient Complexity
SG-HMC [52]	Strongly Convex	W_2	$\tilde{\mathcal{O}}(n\epsilon^{-2})$
SVRG-HMC [52]	Strongly Convex	W_2	$\tilde{\mathcal{O}}(n^{2/3}\epsilon^{-2/3}+\epsilon^{-1})$
SAGA-HMC [52]	Strongly Convex	W_2	$\tilde{\mathcal{O}}(n^{2/3}\epsilon^{-2/3}+\epsilon^{-1})$
CV-HMC [52]	Strongly Convex	W_2	$\tilde{\mathcal{O}}(\epsilon^{-2})$
SRVR-HMC [53]	Dissipative Gradients	W_2	$\tilde{\mathcal{O}}(n+n^{1/2}\epsilon^{-2}+\epsilon^{-4})$
SVRG-LMC [28]	LSI	KL	$\tilde{\mathcal{O}}(n+n^{1/2}\epsilon^{-1})$
SARAH-LMC [28]	LSI	KL	$\tilde{\mathcal{O}}(n+n^{1/2}\epsilon^{-1})$
QSVRG-HMC [Theorem 2.6]	Strongly Convex	W_2	$\tilde{\mathcal{O}}(n^{1/2}\epsilon^{-3/4}+\epsilon^{-1})$
QCV-HMC [Theorem 2.7]	Strongly Convex	W_2	$\tilde{\mathcal{O}}(\epsilon^{-3/2})$
QSVRG-LMC [Theorem 2.3]	LSI	KL[1]	$\tilde{\mathcal{O}}(n+n^{1/3}\epsilon^{-1})$

▶ Remark 2.8. Both the classical algorithms in [52] and quantum algorithms in this paper assume that the starting point is (d/μ)-close to the minimizer $\mathbf{x}^\star = \arg\min f(\mathbf{x})$. In case this point is not given, it can be obtained using $\mathcal{O}(n)$ iterations of SGD [3].

3 Quantum Gradient Estimation in Zeroth-Order Stochastic Setting

We characterize the complexity of our algorithms in this section with respect to the stochastic zeroth-order oracle defined in Equation (4). Jordan's quantum gradient estimation algorithm [27] constructs a superposition over d-dimensional N grid points G_d^l centered around the point $\mathbf{x}$ with side length l. Then the following quantum state is obtained by using a phase oracle $O : |\mathbf{x}\rangle \mapsto e^{itf(\mathbf{x})}|\mathbf{x}\rangle$,

$$|\psi\rangle = \frac{1}{\sqrt{N^d}} \sum_{\mathbf{y}\in G_d^l(\mathbf{x})} e^{it[f(\mathbf{x}+\frac{l}{N}(\mathbf{y}-\frac{N}{2}))-f(\mathbf{x})]}|\mathbf{y}\rangle \tag{8}$$

where t is a proper scale factor. Then the algorithm uses quantum Fourier transform to compute the full gradient. This algorithm uses only a constant number of queries to the function f, whereas one needs at least d queries to approximate the gradient classically. We refer the reader to [7, 24] and the full version of this paper for a more detailed review of Jordan's quantum estimation algorithm. We occasionally refer to the version of this algorithm with optimized parameters as `QuantumGradient` given in [7]. We also review this algorithm in Appendix B.2 and give the description in Algorithm 7.

Jordan's quantum gradient estimation algorithm [27] constructs a superposition over a d-dimensional grid $G_d^l(\mathbf{x})$ centered at $\mathbf{x}$, applies a phase oracle $O : |\mathbf{x}\rangle \mapsto e^{itf(\mathbf{x})}|\mathbf{x}\rangle$, and then applies a quantum Fourier transform to recover the gradient. In the deterministic setting, this requires only a constant number of function queries, whereas a classical finite-difference method needs at least d queries.

[1] Convergence in KL divergence implies convergence in squared TV and W_2 distances due to Pinsker's and Talagrand's inequalities.

3.1 Gradient Estimation for Smooth Potentials

We assume the following about the evaluation oracle for f.

▶ **Assumption 3.1** (Bounded Noise)**.** For any $\mathbf{x} \in \mathbb{R}^d$, the stochastic zeroth-order oracle outputs an estimator $f(\mathbf{x};\xi)$ of $f(\mathbf{x})$ such that $\mathbb{E}[f(\mathbf{x};\xi)] = f(\mathbf{x})$, $\mathbb{E}[\nabla f(\mathbf{x};\xi)] = \nabla f(\mathbf{x})$, and $\mathbb{E}\|\nabla f(\mathbf{x};\xi) - \nabla f(\mathbf{x})\|^2 \le \sigma^2$.

We note that gradients of $f(\cdot;\xi)$ are not queried by the algorithm and only appear in the assumption.

▶ **Assumption 3.2** (Smoothness)**.** The potential function $f : \mathbb{R}^d \to \mathbb{R}$ has L-Lipschitz gradients. Specifically, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ it holds that

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|.$$

The assumption is broader than an additive-noise model because it also covers certain multiplicative-noise models. For example, suppose $f : \mathcal{B}_d(0,R) \to \mathbb{R}$ is L-smooth and differentiable, and the stochastic components have the form $f(\mathbf{x};\xi) = \xi f(\mathbf{x})$, where $\mathbb{E}[\xi] = 1$ and $\mathbb{E}[(\xi-1)^2] \le \frac{\sigma^2}{4L^2R^2}$. Then Assumption 3.1 is satisfied.

We assume that f can be queried with the same randomness at two points, so that $f(\mathbf{x};\xi_i)$ and $f(\mathbf{y};\xi_i)$ can be evaluated using the same seed ξ_i. In the finite-sum case, for instance, one can prepare $\frac{1}{\sqrt{n}}\sum_{i=1}^n |0\rangle|i\rangle$ and then, using the binary oracle, prepare $\frac{1}{\sqrt{n}}\sum_{i=1}^n |f(\mathbf{x};i)\rangle|i\rangle$. This requires no QRAM-style data encoding, since the superposition is only over the indices. Classically, the gradient in this two-point setting can be estimated using the Gaussian smoothing technique. This involves sampling random directions from the extended space around the target point and performing two-point evaluations to approximate the gradient. Specifically, the gradient can be approximated as:

$$\mathbf{g}_{\nu,b}(\mathbf{x}) = \frac{1}{b}\sum_{i=1}^{b} \frac{f(\mathbf{x}+\nu\mathbf{u}_i;\xi_i) - f(\mathbf{x};\xi_i)}{\nu}\mathbf{u}_i, \tag{9}$$

where $\mathbf{u}_i \sim \mathcal{N}(0, I_d)$ are independent and identically distributed random vectors. Authors in [4] showed that for any $\mathbf{x} \in \mathbb{R}^d$, the estimator $\mathbf{g}_{\nu,b}$ satisfies $\mathbb{E}\|\mathbf{g}_{\nu,b}(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \le \frac{4(d+5)(\|\nabla f(\mathbf{x})\|^2+\sigma^2)}{b} + \frac{3\nu^2L^2(d+3)^3}{2}$. Although the squared norm of the gradient on the right-hand side is unbounded, it is typically of order $\tilde{\mathcal{O}}(d)$ in expectation throughout the trajectory of LMC. Consequently, this method requires $b = \mathcal{O}(d^2/\epsilon^2)$ function evaluations to achieve an ϵ-accurate gradient estimate in the L_2 norm.

To our knowledge, no existing quantum algorithm estimates the gradient in this particular setting. Our goal is to develop additional tools that allow Jordan's quantum estimation algorithm [27] to be used under this oracle model. The main technical tool is the following robust phase oracle.

▶ **Proposition 3.3.** *Let $X \in \mathbb{R}$ be a random variable such that $\mathbb{E}\|X - \mathbb{E}[X]\|^2 \le \sigma^2$. Given two reals $t \ge 0$ and $\epsilon \in (0,1)$, then there is a unitary operator $P^X_{t,\epsilon} : |0\rangle|0\rangle \mapsto |\phi_X\rangle|0\rangle$ acting on $\mathcal{H}_X \otimes \mathcal{H}_{aux}$ that can be implemented using $\tilde{\mathcal{O}}(t\sigma\log(1/\epsilon))$ quantum experiments and binary oracle queries to X such that*

$$\| |\phi_X\rangle - e^{it\mathbb{E}[X]}|0\rangle \| \le \epsilon,$$

with probability at least $8/9$.

This phase oracle is similar to the oracle implemented in [17], but their construction requires $\|X\| \leq 1$, whereas X may be unbounded in our setting. Thus Proposition 3.3 extends the phase oracle to unbounded random variables by using a sequence of unitaries at different truncation levels. In our gradient estimation algorithm, this oracle replaces the phase oracle in Algorithm 7, with $X = f(\mathbf{x}; \xi)$. Since Jordan's algorithm is biased and succeeds with high probability, we postprocess its output using a multilevel Monte Carlo algorithm (See Appendix B.3 for overview) to obtain a smooth unbiased gradient estimator.

▶ **Theorem 3.4.** *Suppose that the potential function f satisfies Assumptions 3.1 and 3.2 and further suppose that $\|\nabla f(\mathbf{x})\| \leq M^2$ for all $\mathbf{x} \in \mathcal{B}(x^\star, \mathrm{poly}(d))$. Then, given a real $\mathbf{x} \in \mathcal{B}(x^\star, \mathrm{poly}(d))$ and $\hat{\sigma} > 0$, there exists a quantum algorithm that outputs a random vector $\mathbf{g}$ such that*

$$\mathbb{E}[\mathbf{g}] = \nabla f(\mathbf{x}), \quad \textit{and} \quad \mathbb{E}\|\mathbf{g} - \nabla f(\mathbf{x})\|^2 \leq \hat{\sigma}^2$$

using $\tilde{O}(\frac{\sigma d}{\hat{\sigma}})$ queries to the stochastic evaluation oracle.

Although Theorem 3.4 is motivated by the sampling task, it might be of independent interest for various optimization problems even in non-smooth settings. For example, suppose that f_ξ is a non-smooth but locally L-Lipschitz function on the grid G_d^l. We define $f_v(\mathbf{x}) = \mathbb{E}_{\xi \in \Xi, \mathbf{u} \sim \mathcal{B}(0,1)}[f(\mathbf{x} + v\mathbf{u}; \xi)]$. Then, let $\mathbf{y} \in G_d^l$, and we have $\mathbb{E}\|\nabla f(\mathbf{y} + v\mathbf{u}) - \nabla f_v(\mathbf{y})\|^2 \leq 4L^2$. It is known that f_v is a smooth function with smoothness parameter $O(Ld^{1/2}v^{-1})$. Hence, by Theorem 3.4 our algorithm outputs an unbiased estimator $\mathbf{g}$ such that $\mathbb{E}[\mathbf{g}] = \nabla f_v(\mathbf{x})$ and $\mathbb{E}\|\mathbf{g} - \nabla f_v(\mathbf{x})\|^2 \leq \hat{\sigma}^2$ using $\tilde{\mathcal{O}}(\frac{Ld}{\hat{\sigma}})$ queries to f_ξ. This result has recently been established in [30], and it is a special case of Theorem 3.4. Hence, our quantum gradient estimation can give speedups beyond the settings considered in [30], but this is outside the scope of this work.

3.2 Gradient Estimation under Additional Smoothness Assumption

In this section, we impose a slightly stronger smoothness assumption on the stochastic functions f_ξ, which further improves the dimension dependence.

▶ **Assumption 3.5** (Lipschitz Stochastic Gradients)**.** The stochastic component $f(\cdot; \xi) : \mathbb{R}^d \to \mathbb{R}$ has $L(\xi)$-Lipschitz gradients for any $\xi \in \Xi$. Specifically, it holds that

$$\|\nabla f(\mathbf{x}; \xi) - \nabla f(\mathbf{y}; \xi)\| \leq L(\xi)\|\mathbf{x} - \mathbf{y}\|, \tag{10}$$

and the expected Lipschitz constant satisfies $\mathbb{E}[L(\xi)] = L$.

Assumption 3.5 is weaker than assuming that each stochastic function f_ξ has L-Lipschitz gradients, and it is straightforward to show that Assumption 3.5 implies that f has Lipschitz gradients.

In this section, we assume the following sampling oracle,

$$O_\xi : |\mathbf{x}\rangle |0\rangle \mapsto \sum_{\xi \in \Xi} \sqrt{\Pr(\xi)}\, |\mathbf{x}\rangle |\xi\rangle . \tag{11}$$

[2] One can show that the norm of the gradient is bounded by a function of problem parameters in such a large ball throughout the trajectory of HMC or LMC due to smoothness. Since the dependency on M is logarithmic, we do not give an explicit bound on M.

Instead of implementing an accurate phase oracle, one can first estimate $\nabla f(\mathbf{x};\xi)$ and then use quantum mean estimation to compute $\nabla f(\mathbf{x})$. Quantum mean estimation computes the mean of a random variable with quadratically fewer samples than classical mean estimation (Lemma B.1).

However, Assumption 3.5 implies that f_ξ might not be a smooth function (even if f is smooth), which is the requirement in [7]. Hence, Jordan's algorithm might fail to compute the gradient for ∇f_ξ with small probability regardless of the parameter choice in quantum gradient estimation. To address this, we propose a robust version of the quantum gradient estimation algorithm. Our final algorithm achieves $\tilde{\mathcal{O}}(d^{1/2}\epsilon^{-1})$ query complexity to estimate the gradient up to ϵ error.

Algorithm 4 `QuantumStochasticGradient`.

0: **Input**: Access to stochastic functions $f_{\xi\in\Xi}$, variance σ^2, target ϵ, smoothness parameter L, point $\mathbf{x}$.
Define $\beta = \frac{164L\sigma^2}{\epsilon^2}$, $D = \frac{40\sigma^2}{\epsilon}$, $\epsilon' = \frac{\epsilon^2}{\beta^2 d^3 (12000)^2}$.
1: Sample ξ_0 at random from Ξ.
2: Compute $\boldsymbol{s} = \texttt{QuantumGradient}(f_{\xi_0}, \epsilon', M, \beta, \mathbf{x})$ (See Algorithm 7).
3: Let $\mathcal{A}$ be a algorithm that runs $\mathbf{g} = \texttt{QuantumGradient}(f_\xi, \epsilon', M, \beta, \mathbf{x})$ with random $\xi \in \Xi$ and outputs $\mathbf{g}$ if $\|\mathbf{g} - \boldsymbol{s}\| \le D$, otherwise it outputs $\boldsymbol{s}$. Further suppose that $\mathcal{A}$ does not make any measurement.
4: Run $\mathcal{A}$ on the superposition state to obtain $\sum_{\xi\in\Xi} \sqrt{\Pr(\xi)}\,|\mathcal{A}(\mathbf{x})\rangle\,|\xi\rangle$.
5: Output $\boldsymbol{v} = \texttt{QuantumMeanEstimation}(\mathcal{A}, \epsilon/4, \delta)$.

The algorithm first samples ξ, runs the gradient estimation algorithm, and stores the output $\mathbf{s}$. The key idea is to use $\mathbf{s}$ to replace outlier estimates and then show that this replacement does not significantly increase the error with high probability. Next, we run $\mathcal{A}$ in superposition on $|\psi_1\rangle = \sum_{\xi\in\Xi} \sqrt{\Pr(\xi)}\,|\mathbf{x}\rangle\,|\xi\rangle$. The final step of Jordan's algorithm produces the following quantum state:

$$\sum_{\xi\in\Xi} \sqrt{\Pr(\xi)}\,|\tilde{\mathbf{g}}(\mathbf{x};\xi)\rangle\,|\mathbf{x}\rangle\,|\xi\rangle + |\mathcal{X}_1\rangle\,, \tag{12}$$

where $|\mathcal{X}_1\rangle$ is another garbage state with a small amplitude and

$$\tilde{\mathbf{g}}(\mathbf{x},\xi) = \begin{cases} \mathbf{g}(\mathbf{x},\xi) & \text{if } \|\mathbf{g}(\mathbf{x},\xi) - \mathbf{s}\| \le D, \\ \mathbf{s} & \text{otherwise.} \end{cases} \tag{13}$$

is the corrected gradient estimate.

Finally, we estimate the mean of the first register and output the resulting vector $\boldsymbol{v}$ as the gradient estimate. The replacement step works because $\mathbf{s}$ is within a controlled distance of the true gradient with high probability, so truncating large-error estimates changes the expectation by only a small amount. This truncation is nevertheless necessary because rare large errors can otherwise dominate the mean. A rigorous probabilistic analysis gives the following guarantee.

▶ **Lemma 3.6.** *Under Assumptions 3.1 and 3.5, Algorithm 4 returns a vector $\boldsymbol{v}$ such that*

$$\|\boldsymbol{v} - \nabla f(\mathbf{x})\| \le \epsilon \tag{14}$$

with high probability using $\tilde{\mathcal{O}}(\sigma d^{1/2}\epsilon^{-1})$ queries to the stochastic evaluation oracle.

Next, we can postprocess the output of Algorithm 4 using MLMC to obtain a smooth and unbiased estimate.

▶ **Theorem 3.7** (Smooth Gradient). *Suppose that the potential function f satisfies Assumptions 3.1 and 3.5 and further suppose that $\|\nabla f(\mathbf{x})\| \leq M$ for all $\mathbf{x} \in \mathcal{B}(x^\star, \mathrm{poly}(d))$. Then, given a real $\mathbf{x} \in \mathcal{B}(x^\star, \mathrm{poly}(d))$ and $\hat{\sigma} > 0$, there exists a quantum algorithm that outputs a random vector $\mathbf{g}$ such that*

$$\mathbb{E}[\mathbf{g}] = \nabla f(\mathbf{x}), \quad \textit{and} \quad \mathbb{E}\|\mathbf{g} - \nabla f(\mathbf{x})\|^2 \leq \hat{\sigma}^2 \tag{15}$$

using $\tilde{\mathcal{O}}(\frac{\sigma d^{1/2}}{\hat{\sigma}})$ queries to the stochastic evaluation oracle in expectation.

We remark that our focus is on query complexity. Quantum mean estimation and Jordan's algorithm use $\mathcal{O}(\mathrm{poly}(d, \log(1/\epsilon)))$ gates [17, 9], and our algorithm does not change this circuit structure.

4 Quantum Speedups for Sampling via Evaluation Oracle

We apply the quantum gradient estimation algorithms from the previous sections to establish convergence of HMC and LMC with inexact gradients in the strongly convex and LSI settings. The gradient estimation error creates a tradeoff: larger error increases the number of sampler iterations, while smaller error requires more evaluation queries. We optimize this error to minimize the total query cost. We first treat approximate sampling from strongly log-concave distributions using HMC, with error measured in the Wasserstein metric.

▶ **Theorem 4.1** (Main Theorem for `QZ-HMC`). *Let μ_k be the distribution of $\mathbf{x}_k$ in* `QZ-HMC` *algorithm. Suppose that the Assumptions 2.5, 3.1, and 3.2 hold. Given that the initial point $\mathbf{x}_0$ satisfies $\|\mathbf{x}_0 - \arg\min_{\mathbf{x}} f(\mathbf{x})\| \leq \frac{d}{\mu}$, if we set the step size $\eta = \mathcal{O}\left(\frac{\epsilon}{d^{1/2}\kappa^{3/2}}\right)$, $S = \tilde{\mathcal{O}}\left(\frac{L d^{1/2} \kappa^{3/2}}{\epsilon}\right)$, $T = \tilde{\mathcal{O}}(1)$, and $\hat{\sigma}^2 = \mathcal{O}\left(\frac{L^{3/2} d^{1/2} \epsilon}{\kappa^{3/2}}\right)$, we have*

$$\mathrm{W}_2(\mu_{ST}, \pi) \leq \epsilon.$$

In addition, under Assumptions 3.1 and 3.2, the query complexity to the stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^{5/4}\sigma}{\epsilon^{3/2}}\right)$ or under Assumptions 3.1 and 3.5 the query complexity to the stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^{3/4}\sigma}{\epsilon^{3/2}}\right)$.

Because strong convexity and the Wasserstein metric can be restrictive, we next consider non-convex potentials.

▶ **Theorem 4.2** (Main Theorem for `QZ-LMC`). *Suppose that the Assumptions 2.5, 3.1, and 3.2 hold. Let μ_k be the distribution of $\mathbf{x}_k$ in* `QZ-LMC` *algorithm. Then, if we set the step size $\eta = \mathcal{O}\left(\frac{\epsilon\alpha}{dL^2}\right)$, $K = \tilde{\mathcal{O}}\left(\frac{dL^2 \log(\mathrm{KL}(\mu_0||\pi))}{\epsilon\alpha^2}\right)$, and $\hat{\sigma}^2 = \mathcal{O}(\alpha\epsilon)$, we have*

$$\left\{\mathrm{KL}(\mu_K||\pi), \mathrm{TV}(\mu_K, \pi)^2, \frac{\alpha}{2}\mathrm{W}_2(\mu_K, \pi)^2\right\} \leq \epsilon.$$

In addition, under Assumptions 3.1 and 3.2, the query complexity to the stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^2 L^2 \sigma}{\alpha^{5/2}\epsilon^{3/2}}\right)$, or under Assumptions 3.1 and 3.5 the query complexity to the stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^{3/2} L^2 \sigma}{\alpha^{5/2}\epsilon^{3/2}}\right)$.

This result holds for several distance metrics. Compared with classical results, [39] analyzes zeroth-order LMC under Assumptions 3.1 and 3.2 and obtains evaluation complexity $\mathcal{O}(d^3\sigma^2/\epsilon^4)$ for convergence in W_2 (Theorem 3.2 of [39]). Our algorithm uses $\tilde{\mathcal{O}}\frac{d^2\sigma}{\epsilon^{3/2}})$ evaluation queries under the same assumptions, yielding polynomial improvements in d, σ, and ϵ.

5 Application to Non-smooth and Non-convex Optimization

Although the previous sampling results assume smoothness (Lipschitz gradients), they also yield algorithms for non-smooth optimization. We consider objectives satisfying the following approximate-convexity and Lipschitz continuity assumptions.

▶ **Assumption 5.1** (Approximate Convexity)**.** Let f be a differentiable function, we say that $f : \mathbb{R}^d \to \mathbb{R}$ is an ϵ-approximately convex function, if there exists a strongly convex function F such that for all $\mathbf{x}$,

$$|F(\mathbf{x}) - f(\mathbf{x})| \leq \frac{\epsilon}{d}. \tag{16}$$

▶ **Assumption 5.2** (Lipschitz Continuity)**.** For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $f : \mathbb{R}^d \to \mathbb{R}$ satisfies,

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq M\|\mathbf{x} - \mathbf{y}\|. \tag{17}$$

The goal is to find an approximate minimizer $\mathbf{x}^\star$ such that $|f(\mathbf{x}^\star) - \min_{\mathbf{x}} f(\mathbf{x})| \leq \epsilon$. This setting arises naturally in empirical risk minimization (ERM), where the objective is defined through empirical observations. Even if the population objective is smooth and strongly convex, subsampling can make the empirical objective non-smooth and non-convex. Similar settings also arise in the study of escaping local minima, both classically [5] and quantumly [29], under access to a stochastic evaluation oracle. Our approach differs from previous techniques by sampling from a Gibbs distribution associated with a smooth approximation of f.

Since f is not smooth, we consider the smoothed approximation

$$f_v(\mathbf{x}) = \mathbb{E}_{u \sim \mathcal{B}_d(0,1)}[f(\mathbf{x} + v\mathbf{u})] \tag{18}$$

where $\mathcal{B}$ is the unit L_2 ball around the center. The local properties of f_v are known and given by the following proposition.

▶ **Proposition 5.3.** *If f satisfies Assumption 5.2, then f_v satisfies $|f_v(\cdot) - f(\cdot)| \leq vM$ and $|\nabla f_v(\mathbf{x}) - \nabla f_v(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|$, where $L = cM\sqrt{d}v^{-1}$ for some constant $c > 0$.*

First, note that $\mathbb{E}_{\mathbf{u}}\|\nabla f(\mathbf{x} + v\mathbf{u}) - \nabla f_v(\mathbf{x})\|^2 \leq 4M^2$, since $\|\nabla f(\mathbf{x})\| \leq M$ by Lipschitz continuity.

The essential idea is that the Gibbs distribution localizes around the global minimizers of f as β increases. Another challenge in sampling from f_v is that the log-Sobolev constant may depend on the dimension. Using the Holley-Stroock LSI perturbation result [25], we show how this dimension dependence enters the total evaluation complexity. We now state the main result.

▶ **Theorem 5.4.** *Suppose that f satisfies Assumptions 5.1 and 5.2. Then, there exists a quantum algorithm that returns ϵ approximate minimizer for f with high probability using $\tilde{\mathcal{O}}\left(\frac{d^{9/2}}{\epsilon^{3/2}}\right)$ queries to the stochastic evaluation oracle for f.*

The closest result to our setting is [29], whose stochastic query complexity is $\tilde{\mathcal{O}}(d^5/\epsilon)$, although the assumptions are not directly comparable. The authors assume additive sub-Gaussian noise and convexity of F on a bounded domain, but not strong convexity. Because these assumptions differ from ours, the comparison should be interpreted cautiously. Under our assumptions, the algorithm improves the dimension dependence at the cost of worse ϵ-dependence, a common tradeoff for sampling-based optimization methods.

We also note the work by Augustino et al. [2] and Chakrabarti et al. [9], which also investigate zeroth-order stochastic convex optimization under assumptions similar to those in [29]. They propose algorithms with query complexities $\widetilde{\mathcal{O}}(d^{9/2}/\epsilon^7)$ and $\widetilde{\mathcal{O}}(d^3/\epsilon^5)$, respectively. Although both approaches have worse ϵ-dependence than ours, their assumptions and problem settings differ significantly from ours.

References

1 Simon Apers and Alain Sarlette. Quantum fast-forwarding: Markov chains and graph property testing. *Quantum Info. Comput.*, 19(3–4):181–213, March 2019. doi:10.26421/QIC19.3-4-1.

2 Brandon Augustino, Dylan Herman, Enrico Fontana, Junhyung Lyle Kim, Jacob Watkins, Shouvanik Chakrabarti, and Marco Pistoia. Fast convex optimization with quantum gradient methods, 2025. doi:10.48550/arXiv.2503.17356.

3 Jack Baker, Paul Fearnhead, Emily B. Fox, and Christopher Nemeth. Control variates for stochastic gradient MCMC. *Statistics and Computing*, 29(3):599–615, May 2019. doi:10.1007/s11222-018-9826-2.

4 Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-order nonconvex stochastic optimization: Handling constraints, high dimensionality, and saddle points. *Found. Comput. Math.*, 22(1):35–76, February 2022. doi:10.1007/s10208-021-09499-8.

5 Alexandre Belloni, Tengyuan Liang, Hariharan Narayanan, and Alexander Rakhlin. Escaping the local minima via simulated annealing: Optimization of approximately convex functions. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 240–265, Paris, France, 03–06 July 2015. PMLR. URL: https://proceedings.mlr.press/v40/Belloni15.html.

6 Shouvanik Chakrabarti, Andrew M. Childs, Shih-Han Hung, Tongyang Li, Chunhao Wang, and Xiaodi Wu. Quantum algorithm for estimating volumes of convex bodies. *ACM Transactions on Quantum Computing*, 4(3), May 2023. doi:10.1145/3588579.

7 Shouvanik Chakrabarti, Andrew M. Childs, Tongyang Li, and Xiaodi Wu. Quantum algorithms and lower bounds for convex optimization. *Quantum*, 4:221, January 2020. doi:10.22331/q-2020-01-13-221.

8 Shouvanik Chakrabarti, Dylan Herman, Guneykan Ozgul, Shuchen Zhu, Brandon Augustino, Tianyi Hao, Zichang He, Ruslan Shaydulin, and Marco Pistoia. Generalized short path algorithms: Towards super-quadratic speedup over Markov chain search for combinatorial optimization, 2024. doi:10.48550/arXiv.2410.23270.

9 Shouvanik Chakrabarti, Dylan Herman, Jacob Watkins, Enrico Fontana, Brandon Augustino, Junhyung Lyle Kim, and Marco Pistoia. On speedups for convex optimization via quantum dynamics, 2025. doi:10.48550/arXiv.2503.24332.

10 David Chandler. Introduction to modern statistical. *Mechanics. Oxford University Press, Oxford, UK*, 5(449):11, 1987.

11 Niladri Chatterji, Nicolas Flammarion, Yian Ma, Peter Bartlett, and Michael Jordan. On the theory of variance reduction for stochastic gradient Monte Carlo. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 764–773. PMLR, 10–15 July 2018. URL: https://proceedings.mlr.press/v80/chatterji18a.html.

12 Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1683–1691, Bejing, China, 22–24 June 2014. PMLR. URL: https://proceedings.mlr.press/v32/cheni14.html.

13 Xi Chen, Simon S. Du, and Xin T. Tong. On stationary-point hitting time and ergodicity of stochastic gradient langevin dynamics. *Journal of Machine Learning Research*, 21(68):1–41, 2020. URL: http://jmlr.org/papers/v21/19-327.html.

14 Sinho Chewi, Murat A Erdogdu, Mufan Li, Ruoqi Shen, and Shunshi Zhang. Analysis of Langevin Monte carlo from Poincare to log-Sobolev. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 1–2. PMLR, 02–05 July 2022. URL: https://proceedings.mlr.press/v178/chewi22a.html.

15 Sinho Chewi and Austin J. Stromme. The ballistic limit of the log-Sobolev constant equals the Polyak-Łojasiewicz constant, 2024. doi:10.48550/arXiv.2411.11415.

16 Andrew M. Childs, Tongyang Li, Jin-Peng Liu, Chunhao Wang, and Ruizhe Zhang. Quantum algorithms for sampling log-concave distributions and estimating normalizing constants. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23205–23217. Curran Associates, Inc., 2022. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/933e953353c25ec70477ef28e45a2dcc-Paper-Conference.pdf.

17 Arjan Cornelissen, Yassine Hamoudi, and Sofiene Jerbi. Near-optimal quantum algorithms for multivariate mean estimation. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC '22. ACM, June 2022. doi:10.1145/3519935.3520045.

18 Ben Cousins and Santosh Vempala. Gaussian cooling and $O^*(n^3)$ algorithms for volume and Gaussian volume. *SIAM Journal on Computing*, 47(3):1237–1273, 2018. doi:10.1137/15M1054250.

19 Aniket Das, Dheeraj M. Nagaraj, and Anant Raj. Utilising the CLT structure in stochastic gradient based sampling : Improved analysis and faster algorithms. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 4072–4129. PMLR, 12–15 July 2023. URL: https://proceedings.mlr.press/v195/das23b.html.

20 Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/ede7e2b6d13a41ddf9f4bdef84fdc737-Paper.pdf.

21 Kumar Avinava Dubey, Sashank J. Reddi, Sinead A Williamson, Barnabas Poczos, Alexander J Smola, and Eric P Xing. Variance reduction in stochastic gradient langevin dynamics. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/9b698eb3105bd82528f23d0c92dedfc0-Paper.pdf.

22 Alain Durmus and Eric Moulines. High-dimensional Bayesian inference via the unadjusted Langevin algorithm, 2018. doi:10.48550/arXiv.1605.01559.

23 Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*, volume 1 of *Computational Science Series*. Academic Press, San Diego, second edition, 2002.

24 András Gilyén, Srinivasan Arunachalam, and Nathan Wiebe. *Optimizing quantum optimization algorithms via faster quantum gradient computation*, pages 1425–1444. Society for Industrial and Applied Mathematics, January 2019. doi:10.1137/1.9781611975482.87.

25 Richard Holley and Daniel W. Stroock. Logarithmic sobolev inequalities and stochastic ising models. *Journal of Statistical Physics*, 46:1159–1194, 1987. URL: https://link.springer.com/article/10.1007/BF01011161.

26 Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper_files/paper/2013/file/ac1dd209cbcc5e5d1c6e28598e8cbbe8-Paper.pdf.

27 Stephen P. Jordan. Fast quantum algorithm for numerical gradient estimation. *Physical Review Letters*, 95(5), July 2005. doi:10.1103/physrevlett.95.050501.

28 Yuri Kinoshita and Taiji Suzuki. Improved convergence rate of stochastic gradient Langevin dynamics with variance reduction and its application to optimization. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 19022–19034. Curran Associates, Inc., 2022. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/78e839f96568985d18463044a064ea0f-Paper-Conference.pdf.

29 Tongyang Li and Ruizhe Zhang. Quantum speedups of optimizing approximately convex functions with applications to logarithmic regret stochastic convex bandits. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.

30 Chengchang Liu, Chaowen Guan, Jianhao He, and John C.S. Lui. Quantum algorithms for non-smooth non-convex optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

31 László Lovász and Santosh Vempala. Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. *Journal of Computer and System Sciences*, 72(2):392–417, 2006. JCSS FOCS 2003 Special Issue. doi:10.1016/j.jcss.2005.08.004.

32 Yi-An Ma, Yuansi Chen, Chi Jin, Nicolas Flammarion, and Michael I. Jordan. Sampling can be faster than optimization. *Proceedings of the National Academy of Sciences*, 116(42):20881–20885, September 2019. doi:10.1073/pnas.1820003116.

33 Frederic Magniez, Ashwin Nayak, Jeremie Roland, and Miklos Santha. Search via quantum walk. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 575–584, New York, NY, USA, 2007. Association for Computing Machinery. doi:10.1145/1250790.1250874.

34 Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Found. Comput. Math.*, 17(2):527–566, April 2017. doi:10.1007/s10208-015-9296-2.

35 Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2613–2621. PMLR, 06–11 August 2017. URL: https://proceedings.mlr.press/v70/nguyen17b.html.

36 Guneykan Ozgul, Xiantao Li, Mehrdad Mahdavi, and Chunhao Wang. Stochastic quantum sampling for non-logconcave distributions and estimating partition functions. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 38953–38982. PMLR, 21–27 July 2024. URL: https://proceedings.mlr.press/v235/ozgul24a.html.

37 Guneykan Ozgul, Xiantao Li, Mehrdad Mahdavi, and Chunhao Wang. Quantum speedups for sampling and non-convex optimization with stochastic oracles, 2025. doi:10.48550/arXiv.2504.03626.

38 Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic gradient Langevin dynamics: a nonasymptotic analysis. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1674–1703. PMLR, 07–10 July 2017. URL: https://proceedings.mlr.press/v65/raginsky17a.html.

39 Abhishek Roy, Lingqing Shen, Krishnakumar Balasubramanian, and Saeed Ghadimi. Stochastic zeroth-order discretizations of Langevin diffusions for Bayesian inference, 2021. doi:10.48550/arXiv.1902.01373.

40 Aaron Sidford and Chenyi Zhang. Quantum speedups for stochastic optimization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 35300–35330. Curran Associates, Inc., 2023. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/6ed9931d6e1fb6a85efa1b2c014a47e1-Paper-Conference.pdf.

41 R. Somma, S. Boixo, and H. Barnum. Quantum simulated annealing, 2007. doi:10.48550/arXiv.0712.1008.

42 R. D. Somma, S. Boixo, H. Barnum, and E. Knill. Quantum simulations of classical annealing processes. *Phys. Rev. Lett.*, 101:130504, September 2008. doi:10.1103/PhysRevLett.101.130504.

43 M. Szegedy. Quantum speed-up of Markov chain based algorithms. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 32–41, 2004. doi:10.1109/FOCS.2004.53.

44 Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Convex optimization using quantum oracles. *Quantum*, 4:220, January 2020. doi:10.22331/q-2020-01-13-220.

45 Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted Langevin algorithm: Isoperimetry suffices. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/65a99bb7a3115fdede20da98b08a370f-Paper.pdf.

46 Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2493–2502, Lille, France, 07–09 July 2015. PMLR. URL: https://proceedings.mlr.press/v37/wangg15.html.

47 Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 681–688, Madison, WI, USA, 2011. Omnipress. URL: https://icml.cc/2011/papers/398_icmlpaper.pdf.

48 Pawel Wocjan and Anura Abeyesinghe. Speedup via quantum sampling. *Phys. Rev. A*, 78:042336, October 2008. doi:10.1103/PhysRevA.78.042336.

49 Pan Xu, Jinghui Chen, Difan Zou, and Quanquan Gu. Global convergence of Langevin dynamics based algorithms for nonconvex optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/9c19a2aa1d84e04b0bd4bc888792bd1e-Paper.pdf.

50 Yexin Zhang, Chenyi Zhang, Cong Fang, Liwei Wang, and Tongyang Li. Quantum algorithms and lower bounds for finite-sum optimization. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 60244–60270. PMLR, 21–27 July 2024. URL: https://proceedings.mlr.press/v235/zhang24bz.html.

51 Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient Langevin dynamics. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1980–2022. PMLR, 07–10 July 2017. URL: https://proceedings.mlr.press/v65/zhang17b.html.

52 Difan Zou and Quanquan Gu. On the convergence of Hamiltonian Monte Carlo with stochastic gradients. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 13012–13022. PMLR, 18–24 July 2021. URL: https://proceedings.mlr.press/v139/zou21b.html.

53 Difan Zou, Pan Xu, and Quanquan Gu. Stochastic gradient Hamiltonian Monte Carlo methods with recursive variance reduction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/c3535febaff29fcb7c0d20cbe94391c7-Paper.pdf.

54 Difan Zou, Pan Xu, and Quanquan Gu. Faster convergence of stochastic gradient Langevin dynamics for non-log-concave sampling. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1152–1162. PMLR, 27–30 July 2021. URL: https://proceedings.mlr.press/v161/zou21a.html.

A Overview of Classical Sampling Algorithms

One widely used method for sampling from the Gibbs-Boltzmann distribution is to use the Langevin Monte Carlo (LMC) algorithm:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t) + \sqrt{2\eta_t}\boldsymbol{\epsilon}_t, \tag{19}$$

where $\eta_t > 0$ is the step size and $\boldsymbol{\epsilon}_t$ is isotropic Gaussian noise. This update rule is reminiscent of the gradient descent update rule, but there is an additional noise term that makes the algorithm stochastic. LMC results from the *Euler-Maruyama* discretization of a continuous-time Langevin diffusion equation:

$$\mathrm{d}\mathbf{x}_t = -\nabla f(\mathbf{x}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}(t), \tag{20}$$

where $\mathbf{B}(t)$ is the standard Brownian motion. It is not hard to show that the distribution π is actually stationary for continuous dynamics. Therefore, after iterating the LMC algorithm sufficiently long times with a small step size, the iterates converge to the Gibbs-Boltzmann distribution. We also note that this algorithm is referred to as the Unadjusted Langevin Algorithm (ULA). Therefore, we sometimes use ULA and LMC interchangeably. Although effective, the computational cost of each iteration in this algorithm becomes prohibitive when the computation of the gradient is costly (such as in the finite-sum setting) or not directly available (such as in the zeroth-order setting, where one needs to compute the gradient using a finite difference formula). To alleviate the computational burden, stochastic gradient-based samplers such as Stochastic Gradient Langevin Dynamics (SGLD) [47] have been proposed. Instead of computing the full gradient, these algorithms use stochastic approximation to the gradient. In the stochastic gradient oracle setting, a stochastic estimate $\mathbf{g}_t$ can be obtained by randomly sampling a component $i \in [n]$ and computing $\nabla f_i(\mathbf{x}_t)$. In the stochastic zeroth-order setting, the stochastic gradient can also be obtained by using finite difference formulas by evaluating the function at two close points [34].

Another method that is commonly used in sampling is the Hamiltonian Monte Carlo (HMC) algorithm, which uses the principles of Hamiltonian dynamics to propose new states in a Markov Chain. It introduces the Hamiltonian $H(\mathbf{x}, \mathbf{p}) = f(\mathbf{x}) + \frac{1}{2}\|\mathbf{p}\|^2$ with auxiliary momentum variables and updates the position ($\mathbf{x}$) and momentum ($\mathbf{p}$) by simulating Hamiltonian dynamics, which follows the equations:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \frac{\partial H}{\partial \mathbf{p}}, \quad \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -\frac{\partial H}{\partial \mathbf{x}}. \tag{21}$$

Algorithm 5 SG-LMC.

input The stochastic gradient oracle $O_{\nabla f}$, initial point $\mathbf{x}_0$, step size η, number of steps K
output Approximate sample from $\pi \propto e^{-f(\mathbf{x})}$
  for $k = 0$ to K **do**
    Sample $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, I)$
    $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_t k \mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k) + \sqrt{2\eta_k}\boldsymbol{\epsilon}_k$,
  end for
  Return $\mathbf{x}^K$

After a series of updates, the momentum $\mathbf{p}_{k+1}$ is refreshed by sampling from $\mathcal{N}(0, I)$. This discretization ensures symplecticity, preserving volume in phase space and allowing the algorithm to make large, energy-conserving moves through the parameter space. Similar to SGLD, Stochastic Hamiltonian Monte Carlo (SG-HMC) [12] has been proposed, where one can replace the gradients with stochastic gradients (See Algorithm 6).

Algorithm 6 SG-HMC.

input The stochastic gradient oracle $O_{\nabla f}$, initial point $\mathbf{x}_0$, step size η, number of leapfrog steps S, number of HMC proposals T
output Approximate sample from $\pi \propto e^{-f(\mathbf{x})}$
  for $t = 0$ to T **do**
    Sample $\mathbf{p}_{St} \sim \mathcal{N}(0, I)$
    for $s = 0$ to $S - 1$ **do**
      $k = St + s$
      $\mathbf{x}_{k+1} = \mathbf{x}_k + \eta \mathbf{p}_k - \frac{\eta^2}{2}\mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k)$
      $\mathbf{p}_{k+1} = \mathbf{p}_k - \frac{\eta}{2}\mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k) - \frac{\eta}{2}\mathbf{g}(\mathbf{x}_{k+1}, \boldsymbol{\xi}_{k+1/2})$
    end for
  end for
  Return $\mathbf{x}^T$

The stochastic gradients $\mathbf{g}(\mathbf{x}, \xi)$ in Algorithm 6 can be obtained using different techniques. While stochastic gradient methods reduce computation at each iteration, they introduce variance into the gradient estimates, which can degrade the quality of the samples and slow down convergence. To reduce the variance, further techniques have been proposed, such as SVRG (stochastic variance reduced gradients), CV (control variates). In this paper, we will also employ quantum computational techniques to further reduce the variance in the gradients.

B Overview of Quantum Algorithms

B.1 Quantum Mean Estimation

Quantum mean estimation is a technique to estimate the mean of a d-dimensional random variable X up to ϵ accuracy using $\tilde{\mathcal{O}}(d^{1/2}/\epsilon)$ queries, which is a quadratic improvement in ϵ compared to classical algorithms [17]. Although the quantum mean estimation algorithm is biased, [40] developed an unbiased quantum mean estimation algorithm. Specifically, for a multi-dimensional variable with mean μ and variance σ^2, unbiased quantum mean estimation outputs an estimate $\hat{\mu}$ such that $\mathbb{E}[\hat{\mu}] = \mu$ and $\mathbb{E}[\|\hat{\mu} - \mu\|^2] \leq \hat{\sigma}^2$ using $\tilde{\mathcal{O}}(d^{1/2}\sigma/\hat{\sigma})$ queries.

▶ **Lemma B.1** (Unbiased quantum mean estimation [40]). *Let X be a d-dimensional random variable with $\mathrm{Var}[X] \le \sigma^2$, and suppose a quantum sampling oracle for X is available. For any target variance $\hat{\sigma}^2$, there is a procedure that outputs an unbiased estimate of $\mathbb{E}X$ with variance at most $\hat{\sigma}^2$ using $\widetilde{\mathcal{O}}(d^{1/2}\sigma/\hat{\sigma})$ oracle queries.*

B.2 Jordan's Gradient Estimation Algorithm

Jordan's algorithm [27] approximates the gradient using a finite difference formula on a small grid around the point of interest and encodes the estimate into the quantum phase. Then, the algorithm applies an inverse quantum Fourier transform to estimate the gradient. Although Jordan's original analysis implicitly assumes that higher-order derivatives of the function are negligible, Gilyén, Arunachalam, and Wiebe [24] analyzed the algorithm and extended it to handle functions in the Gevrey class, using central difference formulas and a binary oracle model commonly encountered in variational quantum algorithms. The closest analysis of Jordan's algorithm to our setting was provided by [7], who demonstrated that Algorithm 7 achieves constant query complexity for functions with Lipschitz gradients, provided that the function values can be queried with high precision.

Algorithm 7 `QuantumGradient`$(f, \epsilon, L, \beta, x_0)$.

0: **Input**: Function f, evaluation error ϵ, gradient norm bound L, smoothness parameter β, and point $\mathbf{x}_0$.
Define
- $l = 2\sqrt{\epsilon/\beta d}$ to be the size of the grid used,
- $b \in \mathbb{N}$ such that $\frac{24\pi\sqrt{d\epsilon\beta}}{L} \le \frac{1}{2^b} = \frac{1}{N} \le \frac{48\pi\sqrt{d\epsilon\beta}}{L}$,
- $b_0 \in \mathbb{N}$ such that $\frac{N\epsilon}{2Ll} \le \frac{1}{2^{b_0}} = \frac{1}{N_0} \le \frac{N\epsilon}{Ll}$,
- $F(x) = \frac{N}{2Ll}[f(x_0 + \frac{l}{N}(x - N/2)) - f(x_0)]$, and,
- $\gamma : \{0, 1, \ldots, N-1\} \to G := \{-N/2, -N/2+1, \ldots, N/2-1\}$ s.t. $\gamma(x) = x - N/2$.

Let O_F denote a unitary operation acting as $O_F|x\rangle = e^{2\pi i \tilde{F}(x)}|x\rangle$, where $|\tilde{F}(x) - F(x)| \le \frac{1}{N_0}$, with x represented using b bits and $\tilde{F}(x)$ represented using b_0 bits.

1: Start with n b-bit registers set to 0 and Hadamard transform each to obtain

$$\frac{1}{\sqrt{N^n}} \sum_{x_1,\ldots,x_n \in \{0,1,\ldots,N-1\}} |x_1, \ldots, x_n\rangle ;$$

2: Perform the operation O_F and the map $|x\rangle \mapsto |\gamma(x)\rangle$ to obtain

$$\frac{1}{N^{n/2}} \sum_{\mathbf{g} \in G^n} e^{2\pi i \tilde{F}(\mathbf{g})} |\mathbf{g}\rangle ;$$

3: Apply the inverse QFT over G to each of the registers
4: Measure the final state to get $k_1, k_2, \ldots, k_n$ and report $\tilde{\mathbf{g}} = \frac{2L}{N}(k_1, k_2, \ldots, k_n)$ as the result.

The following lemma from [7] demonstrates that Algorithm 7 achieves $\tilde{O}(1)$ query complexity for evaluating the gradient of a β-smooth function with high probability.

▶ **Lemma B.2** (Lemma 2.2 in [7]). *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function that is accessible via an evaluation oracle with error at most ϵ. Assume that $\|\nabla f\| \le L$ and f is β-smooth in $B_\infty(x, 2\sqrt{\epsilon/\beta})$. Let $\tilde{\mathbf{g}}$ be the output of* `QuantumGradient`$(f, \epsilon, M, \beta, \mathbf{x}_0)$ *(as defined in Algorithm 7). Then:*

$$\Pr\left[|\tilde{\mathbf{g}}_i - \nabla f(\mathbf{x})_i| > 1500\sqrt{d\epsilon\beta}\right] < \frac{1}{3}, \quad \forall i \in [d]. \tag{22}$$

Although Algorithm 7 results in an accurate estimate for the gradient with high probability, it is possible to run the algorithm multiple times and take the coordinate-wise median of the outputs to obtain a smooth estimate for the gradient (Lemma 2.3 in [7]) when the norm of the gradient is bounded. To estimate the gradient up to δ error (in $L2$ norm), it is required to have an evaluation oracle with error at most $\mathcal{O}(\delta^2/d^2)$, which might not be feasible if the noisy evaluation oracle is stochastic.

Our algorithms work in the stochastic setting where we prove that we can create an accurate evaluation oracle under Assumptions 3.1 and 3.2. Furthermore, the function f needs to be smooth; however, under Assumptions 3.1 and 3.5, the smoothness constant is not bounded, and this might cause unbounded error. We propose a robust version of Algorithm 7 so that we can still estimate the gradient accurately (See the step-by-step description in Section 3.2). We also note that the oracle O_F is known as the phase oracle. Our oracle (Equation (4)) can be converted to a phase oracle efficiently.

Second, Jordan's quantum gradient estimation algorithm [27, 24, 7] estimates all coordinates of a gradient by querying a phase oracle on a grid around the target point and applying a quantum Fourier transform. In the deterministic setting, if f is sufficiently smooth and the phase oracle is accurate enough, the algorithm estimates $\nabla f(\mathbf{x})$ with polylogarithmic overhead. The stochastic zeroth-order setting requires additional work because the phase oracle must represent $\mathbb{E}_\xi[f(\mathbf{x};\xi)]$ even when the random function value is not bounded.

More concretely, Jordan's algorithm prepares a uniform superposition over a grid $G_d^l(\mathbf{x})$ centered at $\mathbf{x}$, applies phases proportional to finite differences of f on this grid, and then applies an inverse quantum Fourier transform. If the function is nearly linear on the grid, the resulting frequency encodes the gradient. The deterministic analyses show that Lipschitz-gradient functions can be handled by choosing the grid width and evaluation precision appropriately. In our stochastic setting, the same high-level structure is used, but the phase query itself must be synthesized from noisy evaluations. This is the role of the robust phase oracle in Proposition 3.3.

B.3 Overview of Multi-Level Monte Carlo Algorithm

In this section, we give a brief overview of a technique known as the Multi-Level Monte Carlo algorithm. Without using this technique, our gradient estimation algorithms in the zeroth-order setting would not provide an unbiased estimate for the gradient. Suppose that we have an algorithm $\texttt{BiasedStochasticGradient}(\mathbf{x}, \sigma)$ that outputs $\mathbf{v}$ such that $\mathbb{E}\|\mathbf{v} - \nabla f(\mathbf{x})\| \leq \hat{\sigma}^2$ with cost $\tilde{\mathcal{O}}\left(\frac{C}{\hat{\sigma}}\right)$ where C is a function of other problem parameters. Consider the following algorithm.

While stochastic gradient methods reduce computation at each iteration, they introduce variance into the gradient estimates, which can degrade the quality of the samples and slow down convergence.

Algorithm 8 `UnbiasedStochasticGradient`.

Input: Estimator `BiasedStochasticGradient`, target variance $\hat{\sigma}^2$
Output:An unbiased estimate $\mathbf{g}$ of $\nabla f(\mathbf{x})$ with variance at most $\hat{\sigma}^2$
Set $\mathbf{g}_0 \leftarrow$`BiasedStochasticGradient`$(\mathbf{x}, \hat{\sigma}/10)$
Randomly sample $j \sim \text{Geom}\left(\frac{1}{2}\right) \in \mathbb{N}$
$\mathbf{g}_j \leftarrow$`BiasedStochasticGradient`$(\mathbf{x}, 2^{-3j/4}\hat{\sigma}/10)$
$\mathbf{g}_{j-1} \leftarrow$`BiasedStochasticGradient`$(\mathbf{x}, 2^{-3(j-1)/4}\hat{\sigma}/10)$
$\mathbf{g} \leftarrow \mathbf{g}_0 + 2^j(\mathbf{g}_j - \mathbf{g}_{j-1})$
Return $\mathbf{g}$

▶ **Lemma B.3.** *Given access to an algorithm* `BiasedStochasticGradient` *that outputs a random vector* $\mathbf{v}$ *such that* $\mathbb{E}\|\mathbf{v} - \nabla f(\mathbf{x})\| \leq \hat{\sigma}^2$ *with a cost* $\tilde{\mathcal{O}}\left(\frac{C}{\hat{\sigma}}\right)$*, Algorithm 8 outputs a vector* $\mathbf{g}$ *such that* $\mathbb{E}[\mathbf{g}] = \nabla f(\mathbf{x})$ *and* $\mathbb{E}\|\mathbf{g} - \nabla f(\mathbf{x})\| \leq \hat{\sigma}^2$ *with an expected cost* $\tilde{\mathcal{O}}\left(\frac{C}{\hat{\sigma}}\right)$.

Disclaimer

This paper is not a product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates makes any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, without limitation, with respect to the completeness, accuracy, or reliability of the information contained herein and the potential legal, compliance, tax, or accounting effects thereof. This document is not intended as investment research or investment advice, or as a recommendation, offer, or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.