

QUANTUM SPEEDUPS FOR SAMPLING AND NON-CONVEX OPTIMIZATION WITH STOCHASTIC ZEROth ORACLES

Anonymous authors

Paper under double-blind review

ABSTRACT

We propose quantum algorithms with provable speedups for sampling from probability distributions of the form $\pi \propto e^{-f}$, where $f : \mathbb{R}^d \mapsto \mathbb{R}$ is a potential function. In particular, we consider access only to a stochastic evaluation oracle, allowing simultaneous queries of the potential value at two different points under the same stochastic parameter. By introducing novel quantum algorithms for stochastic gradient estimation in this setting, our algorithms improve the evaluation complexities of classical samplers, such as Hamiltonian Monte Carlo (HMC) and Langevin Monte Carlo (LMC) in terms of dimension, precision, and other problem-dependent parameters. Furthermore, we demonstrate that our quantum sampling algorithms can be used to achieve quantum speedups in optimization, particularly for minimizing nonsmooth and approximately convex functions that commonly appear in empirical risk minimization problems.

1 INTRODUCTION

Efficient sampling from complex distributions is a central task across science and engineering, growing in importance as applications involve high-dimensional data and intricate probabilistic models. For example, in probabilistic machine learning, sampling facilitates posterior estimation and quantifies uncertainty in model predictions Welling & Teh (2011); Wang et al. (2015); Durmus & Moulines (2018); Roy et al. (2021). In non-convex optimization, sampling allows for the exploration of complex energy landscapes and helps avoid local minima Zhang et al. (2017); Chen et al. (2020). It is also widely used in other domains of science such as statistical mechanics Chandler (1987); Frenkel & Smit (2002), convex geometry Lovász & Vempala (2006); Cousins & Vempala (2018), etc.

Given a potential function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, we consider the problem of sampling from a probability distribution π of the form

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

This distribution is called the *Gibbs-Boltzmann distribution*, and our goal is to efficiently sample approximately from π while minimizing the number of evaluation queries in the *stochastic zeroth-order setting*. In this setting, we assume that we have only access to noisy function values $f(\mathbf{x}; \xi)$ where ξ is a random variable that characterizes the noise.

Our first approach is based on discretization of continuous Langevin diffusion equation, which follows stochastic differential equation (SDE):

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

where $\mathbf{B}(t)$ is the standard Brownian motion. The Euler-Maruyama discretization of this SDE results in the well-known *Langevin Monte Carlo* (LMC) algorithm:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t) + \sqrt{2\eta_t} \epsilon_t, \quad (3)$$

where $\eta_t > 0$ is the step size and ϵ_t is isotropic Gaussian noise. The second algorithm we consider is the *Hamiltonian Monte Carlo* (HMC) algorithm. HMC introduces the Hamiltonian $H(\mathbf{x}, \mathbf{p}) =$

054 $f(\mathbf{x}) + \frac{1}{2} \|\mathbf{p}\|^2$ and updates the position (\mathbf{x}) and momentum (\mathbf{p}) by simulating Hamiltonian dynamics,
 055 which follows the differential equations:

$$056 \frac{d\mathbf{x}}{dt} = \frac{\partial H}{\partial \mathbf{p}}, \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{x}}. \quad (4)$$

059 Similar to LMC, in practice HMC is simulated by discretizing Eq. (4) and randomly refreshing the
 060 momentum periodically. We refer the reader to appendix A for more details on the discretization
 061 HMC algorithm. Despite their efficient convergence, the computational cost of each iteration in
 062 these algorithms becomes prohibitive when the computation of the gradient is not directly avail-
 063 able. For example, in zeroth order setting one needs to compute the gradient using finite difference
 064 formula. Similarly, one uses mini-batch gradients when the objective function is given through
 065 large number of data points. To alleviate the computational burden, stochastic gradient-based sam-
 066 plers such as Stochastic Gradient Langevin Dynamics (SGLD) Welling & Teh (2011) and Stochastic
 067 Hamiltonian Monte Carlo (SG-HMC) Chen et al. (2014) have also been proposed. Instead of com-
 068 puting the gradient exactly, these algorithms use stochastic approximation to the gradient at each
 069 iteration. For example, the stochastic update for LMC becomes

$$070 \mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \mathbf{g}_t + \sqrt{2\eta_t} \epsilon_t. \quad (5)$$

071 where \mathbf{g} is a random vector that approximates ∇f . Although these algorithms originally proposed
 072 to address finite-sum settings, a stochastic gradient can also be obtained in zeroth order setting by
 073 using finite difference formulas by evaluating the function at two close points Nesterov & Spokoiny
 074 (2017). This technique is typically employed when autodifferentiation is costly due to expensive
 075 backpropagation in training ML models or the function value is only accessible in a black-box
 076 fashion. This scenario has been analyzed under various settings in optimization literature Duchi
 077 et al. (2015); Nesterov & Spokoiny (2017); Balasubramanian & Ghadimi (2022); Lin et al. (2022).
 078 For sampling problems, Roy et al. (2021) has analyzed the convergence of various discretizations
 079 of Langevin diffusion algorithms both for strongly convex and non-convex potentials using the
 080 noisy zeroth-order oracle. Similarly Dalalyan & Karagulyan (2019) has established the conver-
 081 gence of sampling under inexact gradients when the bias and the variance of the inexact estimates
 082 are bounded. Similarly, Yang & Wibisono (2023) analyzed the convergence of the inexact Langevin
 083 algorithm in KL divergence under different assumptions on the potential function.

084 Building on this classical foundation, researchers have started to explore how quantum computa-
 085 tion could accelerate sampling and optimization in the past decade. Quantum algorithms such as
 086 multi-dimensional quantum mean estimation Cornelissen et al. (2022) and quantum gradient esti-
 087 mation Jordan (2005); Gilyén et al. (2019) have shown potential for reducing the query complexity
 088 of gradient-based methods van Apeldoorn et al. (2020); Chakrabarti et al. (2020); Sidford & Zhang
 089 (2023); Zhang et al. (2024); Liu et al. (2024). These techniques are particularly well-suited for ad-
 090 dressing challenges in large-scale and noisy settings, as they can provide more accurate gradient
 091 estimates with asymptotically fewer queries. In this paper, we focus on integrating these quantum
 092 techniques to enhance the efficiency of stochastic gradient-based samplers and alleviate the compu-
 093 tational burden inherent in classical methods.

094 The quest for quantum speedups for optimization has emerged since the discovery of Jordan’s quan-
 095 tum gradient estimation algorithm Jordan (2005) and its refined version Gilyén et al. (2019) which
 096 have generated significant excitement by provable reductions in query complexity against classical
 097 finite-difference methods. Despite the effort in many years, the applicability of quantum gradient
 098 estimation is still rather limited in general settings of optimization problems, which is largely due
 099 to biased estimations, sensitivity to noise, or strong assumptions on the higher derivatives of the
 100 function. In this work, we bridge this technical gap by extending quantum gradient estimation to a
 101 more practical stochastic zeroth-order setting where the function values do not need to be accessed
 102 very precisely and the objective function does not need to satisfy impractical smoothness properties.

103 1.1 MAIN CONTRIBUTIONS

104 We summarize our contributions as follows.

- 105 • **Quantum Speedups for Gradient Estimation via Stochastic Evaluation Oracle:** We develop
 106 novel quantum gradient estimation algorithms under various smoothness assumptions in Section 2
 107 to address the shortcomings of the existing gradient estimation methods. Our algorithm provides

quadratic speedup when the potential function is smooth, reducing the evaluation queries from $\tilde{O}(\frac{d^2\sigma^2}{\epsilon^2})$ to $\tilde{O}(\frac{d\sigma}{\epsilon})$ to compute the gradient up to ϵ accuracy (Theorem 2.4) where σ^2 is the variance of the noise as in Assumption 2.1. Furthermore, when the stochastic functions are also smooth with high probability, we manage to shave off an additional $d^{1/2}$ term (Theorem 2.7). This is achieved by combining quantum mean estimation with Jordan’s quantum gradient estimation in a robust manner. Our gradient estimation algorithms could be useful as independent tools, especially in zeroth-order stochastic optimization.

- **Speedups for Zeroth-Order Sampling:** In Section 3, we combine our new quantum gradient estimation algorithm with HMC and LMC algorithms and analyzed the convergence of the final algorithm. We proved that our hybrid algorithms use fewer number of queries to evaluation oracle than the best known classical samplers under the same assumptions (Theorems 3.2 and 3.4).
- **Application to Non-Convex Optimization:** In Section 4, we extend our quantum sampling methods to optimize non-convex functions with specific structural properties, demonstrating that faster sampling translates to provable speedups in complex optimization tasks. In particular, we show that we can optimize non-smooth and approximately convex functions, i.e. a function that is uniformly close to a strongly convex function, using fewer stochastic evaluation queries than the best known classical algorithms in terms of dimension dependency (Theorem 4.5).

We note that the realization of the contributions in this paper requires a fault-tolerant quantum computer that can implement the procedures above with proper error correction. Therefore, our focus is on establishing the theoretical speedup as a guide for future algorithm design as empirical validation is not possible as of today.

1.2 PRELIMINARIES

Notation: 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.

Quantum Computation: Quantum computation is expressed using linear algebra over complex vector spaces. The computational basis of \mathbb{C}^d is the standard basis $\{\mathbf{e}_0, \dots, \mathbf{e}_{d-1}\}$, where \mathbf{e}_i is the column vector with a 1 in the $(i+1)$ st position and zeros elsewhere. In Dirac notation, we denote \mathbf{e}_i by $|i\rangle$ and its conjugate transpose by $\langle i|$.

The state space of a quantum system with n subsystems is the tensor product space $\mathbb{C}^{d_1} \otimes \dots \otimes \mathbb{C}^{d_n}$. The tensor (Kronecker) product of two vectors $|u\rangle \in \mathbb{C}^{d_1}$ and $|v\rangle \in \mathbb{C}^{d_2}$ is the vector $|u\rangle \otimes |v\rangle \in \mathbb{C}^{d_1 d_2}$, given explicitly by: $|u\rangle \otimes |v\rangle = (u_0 v_0, u_0 v_1, \dots, u_{d_1-1} v_{d_2-1})^\top$.

A single qubit is a normalized vector in \mathbb{C}^2 , written as $\alpha|0\rangle + \beta|1\rangle$, with $|\alpha|^2 + |\beta|^2 = 1$. A system of n qubits lives in the Hilbert space \mathbb{C}^{2^n} , and a general n -qubit state may be entangled, i.e., not expressible as a tensor product of single-qubit states. We often abbreviate tensor products such as $|u\rangle \otimes |v\rangle$ by $|u\rangle |v\rangle$.

Quantum operations correspond to unitary transformations. In the circuit model, a k -qubit gate is a unitary operator $U \in \mathbb{C}^{2^k \times 2^k}$. A universal gate set allows for any n -qubit unitary to be approximated using a sequence of two-qubit gates, up to arbitrarily small error. The gate complexity of a unitary operation refers to the number of such basic gates required in its circuit decomposition.

Measurement is the process of extracting classical information from a quantum system. A projective measurement of state $|\psi\rangle$ in the computational basis $\{|0\rangle, |1\rangle, \dots, |2^n - 1\rangle\}$ yields outcome i with probability $|\langle i|\psi\rangle|^2$, collapsing the state $|\psi\rangle$ to $|i\rangle$. In the quantum framework, a classical probability distribution p can be represented by the quantum state $\sum_{\mathbf{x}} \sqrt{p(\mathbf{x})} |\mathbf{x}\rangle$. When measuring this state, the resulting outcomes are governed by the probability distribution p .

Metrics: 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 $W_p(\pi, \mu) = (\inf_{\gamma \in \Gamma(\pi, \mu)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \|\mathbf{x} - \mathbf{y}\|^p)^{1/p}$ 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 $\text{KL}(\pi\|\mu) = \int_{\mathcal{X}} d\mathbf{x} \pi(\mathbf{x}) \log\left(\frac{\pi(\mathbf{x})}{\mu(\mathbf{x})}\right)$ and the relative Fisher information is $\text{FI}(\pi\|\mu) = \int_{\mathcal{X}} d\mathbf{x} \pi(\mathbf{x}) \left\| \nabla \log\left(\frac{\pi(\mathbf{x})}{\mu(\mathbf{x})}\right) \right\|^2$. The total variation distance is defined as $\text{TV}(\pi, \mu) = \sup_{A \subseteq \mathcal{X}} |\pi(A) - \mu(A)| = \frac{1}{2} \int_{\mathcal{X}} d\mathbf{x} |\pi(\mathbf{x}) - \mu(\mathbf{x})|$.

2 QUANTUM GRADIENT ESTIMATION IN ZERO-ORDER STOCHASTIC SETTING

We consider access to an evaluation oracle for the stochastic components $f_{\xi}(\mathbf{x}) = f(\mathbf{x}; \xi)$, where $\xi \in \Xi$ represents a random seed characterizing the randomness of the noise. Then, the function is accessed through the binary oracle below.

$$O_f |\mathbf{x}\rangle |\xi\rangle |0\rangle \mapsto |\mathbf{x}\rangle |\xi\rangle |f(\mathbf{x}; \xi)\rangle. \quad (6)$$

We characterize the complexity of our algorithms in this section with respect to this oracle. Jordan’s quantum gradient estimation algorithm Jordan (2005) 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 $O : |\mathbf{x}\rangle \mapsto e^{itf(\mathbf{x})} |\mathbf{x}\rangle$,

$$|\psi\rangle = \frac{1}{\sqrt{N^d}} \sum_{\mathbf{y} \in G_d^l(\mathbf{x})} e^{it[f(\mathbf{x} + \frac{1}{N}(\mathbf{y} + \frac{1}{N}) - f(\mathbf{x}))]} |\mathbf{y}\rangle \quad (7)$$

where t is a proper scale factor. Then the algorithm uses quantum Fourier transform to compute the full gradient. This algorithm uses only constant number of queries to the function (Lemma A.4) whereas one needs at least d queries to approximate the gradient classically. We refer the reader to A.4 for more detailed review on Jordan’s quantum estimation algorithm. We occasionally refer to the version of this algorithm with optimized parameters as `QuantumGradient` given in pseudocode 4. Although Jordan’s algorithm is appealing as it only uses a constant number of evaluations to estimate the gradient, its practical use cases remained limited as it requires the function evaluations to be very accurate and the function needs to be very close to linear. In fact when the function is not linear, the speedup in dimension does not always survive Gilyén et al. (2019). In this section, we develop algorithmic techniques to address these obstacles.

2.1 QUANTUM GRADIENT ESTIMATION FOR SMOOTH POTENTIALS

We assume the following about the potential function.

Assumption 2.1 (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 $\mathbb{E}\|\nabla f(\mathbf{x}; \xi) - \nabla f(\mathbf{x})\|^2 \leq \sigma^2$.

Assumption 2.2 (Smoothness). The potential function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ has L -Lipschitz gradients. Specifically, it holds that

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

These assumptions are standard in the zeroth-order sampling and optimization literature Roy et al. (2021); Balasubramanian & Ghadimi (2022). We note that Assumption 2.1 is broader than an additive noise model, as it accommodates models with multiplicative noise. For example, suppose that $f : \mathcal{B}_d(0, R) \mapsto \mathbb{R}$ is an L smooth differentiable function, and that the stochastic components are of the form $f(\mathbf{x}; \xi) = \xi f(\mathbf{x})$, where $\mathbb{E}[\xi] = 1$ and $\mathbb{E}[\xi^2] \leq \frac{\sigma^2}{4L^2 R^2}$. In this case, Assumption 2.1 is satisfied.

Suppose that the function f can be queried with the same randomness at two different points, that is, we can query $f(\mathbf{x}; \xi_i)$ and $f(\mathbf{y}; \xi_i)$ simultaneously¹. For example, in finite-sum case, one can first prepare the quantum state $\frac{1}{\sqrt{n}} \sum_{i=1}^n |0\rangle |i\rangle$ and then prepare the quantum state $\frac{1}{\sqrt{n}} \sum_{i=1}^n |f(\mathbf{x}; i)\rangle |i\rangle$ using binary oracle, since f can be queried at each data point and superposition is over indices 1 to n . Note that this does not require any data encoding (or QRAM structure) as superposition is just over the indices. Classically, the gradient in this two-point setting can be estimated using the Gaussian

¹This is the case in finite-sum and some bandit settings where ξ can be queried explicitly.

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:

$$\mathbf{g}_{\nu,b}(\mathbf{x}) = \frac{1}{b} \sum_{i=1}^b \frac{f(\mathbf{x} + \nu \mathbf{u}_i; \xi_i) - f(\mathbf{x}; \xi_i)}{\nu} \mathbf{u}_i, \quad (8)$$

where $\mathbf{u}_i \sim \mathcal{N}(0, I_d)$ are independent and identically distributed random vectors. Balasubramanian & Ghadimi (2022) showed that for any $\mathbf{x} \in \mathbb{R}^d$, the estimator $\mathbf{g}_{\nu,b}$ satisfies $\mathbb{E} \|\mathbf{g}_{\nu,b}(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \frac{4(d+5)(\|\nabla f(\mathbf{x})\|^2 + \sigma^2)}{b} + \frac{3\nu^2 L^2 (d+3)^3}{2}$. Although the squared norm of the gradient on the right-hand side is unbounded, it is typically of order $\tilde{O}(d)$ in expectation throughout the trajectory of LMC (See Eq. (60)). 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.

As far as we are aware of, currently there does not exist a quantum algorithm to estimate the gradient in this particular setting. Our purpose is to design additional techniques to be able to use Jordan’s algorithm in this setting. The major technical tool in the algorithm is the phase oracle in the following proposition.

Proposition 2.3. *Let $X \in \mathbb{R}$ be a random variable such that $\mathbb{E} \|X - \mathbb{E}[X]\|^2 \leq \sigma^2$. Given two reals $t \geq 0$ and $\epsilon \in (0, 1)$, then there is a unitary operator $P_{t,\epsilon}^X : |0\rangle |0\rangle \mapsto |\phi_X\rangle |0\rangle$ acting on $\mathcal{H}_X \otimes \mathcal{H}_{\text{aux}}$ that can be implemented using $\tilde{O}(t\sigma \log(1/\epsilon))$ quantum experiments and binary oracle queries to X such that*

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

with probability at least $8/9$.

This phase oracle is similar to the oracle implemented in Cornelissen et al. (2022); however, their algorithm requires $\|X\| \leq 1$ whereas $\|X\|$ might be unbounded in our case. Hence, Proposition 2.3 generalizes the phase oracle to the unbounded random variables by constructing a sequence of unitaries for different levels of truncation of the random variable X (See the more detailed description in Appendix D). In particular, this oracle replaces phase oracle (line 4 in Algorithm 4) where the random variable is the stochastic function $f(x; \xi)$ in this case. Since Jordan’s algorithm is biased and succeeds with high probability, we can also obtain smooth and unbiased gradient by post-processing the output using the Multi-Level Monte Carlo technique (Algorithm 3). The preliminaries for the MLMC algorithm can be found in Appendix A.2.

Theorem 2.4. *Suppose that the potential function f satisfies Assumptions 2.1 and 2.2 and further suppose that $\|\nabla f(\mathbf{x})\| \leq M^2$ for all \mathbf{x} . Then, given a real $\hat{\sigma} > 0$, there exists a quantum algorithm that outputs a random vector \mathbf{g} such that*

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

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

Proofs of Proposition 2.3 and Theorem 2.4 are postponed to Appendix D due to limited state.

Although 2.4 is motivated by the sampling task, it might be of an independent tool for various optimization problems even in nonsmooth setting. For example, suppose that f_ξ is a non-smooth but locally L -Lipschitz function around 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 $\mathbf{y} \in G_d^l$, $\mathbb{E} \|\nabla f(\mathbf{y} + v\mathbf{u}) - \nabla f_v(\mathbf{y})\|^2 \leq 4L^2$. It is known that f_v is a smooth function with smoothness parameter $O(Ld^{1/2}v^{-1})$. Hence, by Theorem 2.4 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 $\tilde{O}(\frac{Ld}{\hat{\sigma}})$ queries to f_ξ . This result has recently been established in Liu et al. (2024), and it is a special case of Theorem 2.4. Hence, our quantum gradient estimation can give speedups beyond the settings considered in Liu et al. (2024) but this is outside of the scope of this work.

²One can show that the norm of the gradient is bounded by a function of problem parameters 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 .

2.2 QUANTUM GRADIENT ESTIMATION UNDER ADDITIONAL SMOOTHNESS ASSUMPTION

In this section, we consider a setting that imposes a slightly stronger smoothness assumption on the stochastic functions f_ξ to be able to improve the dimension dependency further.

Assumption 2.5 (Lipschitz Stochastic Gradients). The stochastic component $f(\cdot; \xi) : \mathbb{R}^d \rightarrow \mathbb{R}$ has $L(\xi)$ -Lipschitz gradients for any $\xi \in \Xi$. Specifically, it holds that

$$\|\nabla f(\mathbf{x}; \xi) - \nabla f(\mathbf{y}; \xi)\| \leq L(\xi)\|\mathbf{x} - \mathbf{y}\|, \quad (9)$$

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

Assumption 2.5 is weaker than the assumption that each stochastic function f_ξ has L -Lipschitz gradients and it is straightforward to show that Assumption 2.5 implies that f has Lipschitz gradients.

In this section, we assume the following sampling oracle,

$$O_\xi : |\mathbf{x}\rangle \mapsto \sum_{\xi \in \Xi} \sqrt{\Pr(\xi)} |\mathbf{x}\rangle |\xi\rangle. \quad (10)$$

As opposed to implementing an accurate phase oracle, one can estimate the gradient $\nabla f(\mathbf{x}; \xi)$ and then use the quantum mean estimation algorithm to compute $\nabla f(\mathbf{x})$. Quantum mean estimation is a technique to compute the mean of a random variable quadratically faster than classical techniques (See A.3 overview of quantum mean estimation).

However, Assumption 2.5 implies that f_ξ might not be a smooth function (even if f is smooth), which is the requirement in Lemma A.4. Hence, Jordan’s algorithm might fail to compute the gradient for ∇f_ξ with small probability no matter how large we set β in Algorithm 4. To address this, we propose a robust version of the quantum gradient estimation algorithm. Our final algorithm achieves $\tilde{O}(d^{1/2}\epsilon^{-1})$ query complexity to estimate the gradient up to ϵ error.

Algorithm 1 QuantumStochasticGradient

- 0: **Input:** stochastic functions f_Ξ , 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 $\mathbf{s} = \text{QuantumGradient}(f_{\xi_0}, \epsilon', M, \beta, \mathbf{x})$ (See algorithm 4).
 - 3: Let \mathcal{A} be a randomized algorithm that runs $\mathbf{g} = \text{QuantumGradient}(f_\xi, \epsilon', M, \beta, \mathbf{x})$ with random $\xi \in \Xi$ and outputs \mathbf{g} if $\|\mathbf{g} - \mathbf{s}\| \leq D$, otherwise it outputs \mathbf{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 $\mathbf{v} = \text{QuantumMeanEstimation}(\mathcal{A}, \epsilon/4, \delta)$.
-

The algorithm (line 2) first selects a random ξ and computes runs the gradient estimation algorithm and stores the output \mathbf{s} . The crucial idea is to use this value to replace very bad estimates and then show that this procedure does not increase the error with high probability. Next, we run the algorithm \mathcal{A} in superposition on $|\psi_1\rangle = \sum_{\xi \in \Xi} \sqrt{\Pr(\xi)} |\mathbf{x}\rangle |\xi\rangle$. Then the last step in Jordan’s algorithm is the following quantum state,

$$\sum_{\xi \in \Xi} \sqrt{\Pr(\xi)} |\tilde{\mathbf{g}}(\mathbf{x}; \xi)\rangle |\mathbf{x}\rangle |\xi\rangle + |\mathcal{X}_1\rangle, \quad (11)$$

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

$$\tilde{\mathbf{g}}(\mathbf{x}, \xi) = \begin{cases} \mathbf{g}(\mathbf{x}, \xi) & \text{if } \|\mathbf{g}(\mathbf{x}, \xi) - \mathbf{s}\| \leq D, \\ \mathbf{s} & \text{otherwise.} \end{cases} \quad (12)$$

is the corrected gradient estimate.

Finally, we estimate the mean of the first register to compute \mathbf{v} , which is output as the gradient estimate. Note that the reason why replacement with $\tilde{\mathbf{g}}$ works is the fact that the mediocre estimate \mathbf{s} is one standard deviation away from the true gradient with high probability. Therefore, replacing the

324 tails of the distribution (the estimates with very high error) actually does not change the expectation
 325 too much. However, this procedure is necessary as very erroneous values can change the expectation
 326 without replacement. A more detailed analysis by more rigorous probabilistic arguments gives the
 327 following results.

328 **Lemma 2.6.** *Under Assumptions 2.1 and 2.5, Algorithm 1 returns a vector \mathbf{v} such that*

$$329 \quad \|\mathbf{v} - \nabla f(\mathbf{x})\| \leq \epsilon \quad (13)$$

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

333 Next, we can postprocess the output of Algorithm 1 using MLMC to obtain a smooth and unbiased
 334 estimate.

335 **Theorem 2.7** (Smooth Gradient). *Suppose that the potential function f satisfies Assumptions 2.1
 336 and 2.5 and further suppose that $\|\nabla f(\mathbf{x})\| \leq M$ for all \mathbf{x} . Then, given a real $\hat{\sigma} > 0$, there exists a
 337 quantum algorithm that outputs a random vector \mathbf{g} such that*

$$338 \quad \mathbb{E}[\mathbf{g}] = \nabla f(\mathbf{x}), \quad \text{and} \quad \mathbb{E}\|\mathbf{g} - \nabla f(\mathbf{x})\|^2 \leq \hat{\sigma}^2 \quad (14)$$

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

342 The mathematical details of Lemma 2.6 and Theorem 2.7 are postponed to Appendix D. We remark
 343 our focus is on the query complexity and it can be shown that quantum mean estimation and Jordan’s
 344 algorithm use $\mathcal{O}(\text{poly}(d, \log(1/\epsilon)))$ gates Cornelissen et al. (2022); Chakrabarti et al. (2025) and our
 345 algorithm does not change the circuit structure.

347 3 QUANTUM SPEEDUPS FOR SAMPLING VIA ZERO-ORDER ORACLE

349 We apply our quantum gradient estimation algorithm to establish the convergence of both HMC and
 350 LMC in strongly convex and LSI settings, respectively. In particular, at each iteration, we use the
 351 inexact gradients computed by our quantum gradient estimation algorithms introduced in previous
 352 sections. Our proof techniques involve establishing the convergence rates of HMC and LMC with
 353 inexact gradients. As large error in the gradient requires more iterations and smaller error requires
 354 more function evaluations by the gradient estimation algorithm, we optimize the error in the gradient
 355 to obtain the total optimal cost. We first present our results for approximate sampling from strongly
 356 logconcave distributions using HMC algorithm where the error metric is Wasserstein metric.

357 **Assumption 3.1** (Strong Convexity). *There exists a positive constant μ such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$
 358 it holds that*

$$359 \quad f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{x} - \mathbf{y}\|^2. \quad (15)$$

360 We define the *condition number* $\kappa := \frac{L}{\mu}$.

362 **Theorem 3.2** (Main Theorem for QZ-HMC). *Let μ_k be the distribution of \mathbf{x}_k in QZ-HMC al-
 363 gorithm. Suppose that f satisfies Assumption 3.1. Given that the initial point \mathbf{x}_0 satisfies
 364 $\|\mathbf{x}_0 - \arg \min_{\mathbf{x}} f(\mathbf{x})\| \leq \frac{d}{\mu}$, if we set the step size $\eta = \mathcal{O}(\frac{\epsilon}{d^{1/2} \kappa^{3/2}})$, $S = \tilde{\mathcal{O}}(\frac{L d^{1/2} \kappa^{3/2}}{\epsilon})$,
 365 $T = \tilde{\mathcal{O}}(1)$, and $\hat{\sigma}^2 = \mathcal{O}(\frac{L^{3/2} d^{1/2} \epsilon}{\kappa^{3/2}})$, we have*

$$366 \quad \mathbb{W}_2(\mu_{ST}, \pi) \leq \epsilon.$$

369 *In addition, under Assumptions 2.1 and 2.2, the query complexity to the stochastic evaluation oracle
 370 is $\tilde{\mathcal{O}}(\frac{d^{5/4} \sigma}{\epsilon^{3/2}})$ or under Assumptions 2.1 and 2.5 the query complexity to the stochastic evaluation
 371 oracle is $\tilde{\mathcal{O}}(\frac{d^{3/4} \sigma}{\epsilon^{3/2}})$.*

374 We note that if the initial point does not satisfy the closeness condition, it can be obtained using $\mathcal{O}(1)$
 375 iterations of SGD Baker et al. (2019). The proof of this theorem is postponed to Appendix B.1. The
 376 closest result in the classical setting is given by Roy et al. (2021) for Kinetic LMC algorithm which is
 377 obtained by setting the inner iterations to 1 in HMC algorithm. Their classical evaluation complexity
 under Assumptions 2.1 and 2.2 is $\tilde{\mathcal{O}}(d^2 \sigma^2 / \epsilon^2)$ for convergence in \mathbb{W}_2 distance (Theorem 2.2 in Roy

et al. (2021)). Our algorithm uses $\tilde{\mathcal{O}}(d^{5/4}\sigma/\epsilon^{3/2})$ evaluation queries providing speedup both in d , ϵ , and σ . It is worth noting that quantum acceleration of sampling in the zeroth-order setting for strongly convex functions has been studied in Childs et al. (2022) when the error in the function evaluation is ϵ ; however, their results do not apply to bounded variance setting. Similarly the results by Dalalyan & Karagulyan (2019); Yang & Wibisono (2023) either use different assumptions on f or the access model.

As strong convexity and Wasserstein metric might be restrictive, we consider the sampling problem under for nonconvex potential functions. We assume that the target distribution satisfies the log-Sobolev inequality.

Assumption 3.3 (Log-Sobolev Inequality). We say that π satisfies the Log-Sobolev inequality with constant α if for all ρ , it holds that

$$\text{KL}(\rho||\pi) \leq \frac{1}{2\alpha} \text{FI}(\rho||\pi). \quad (16)$$

This is a sampling analog of the PL (Polyak-Łojasiewicz) condition commonly used in optimization Chewi & Stromme (2024) and standard in non-log-concave sampling literature Vempala & Wibisono (2019); Ma et al. (2019); Chewi et al. (2022); Kinoshita & Suzuki (2022). We note that LSI relaxes strong convexity in the sense that for any μ strongly convex function f , π satisfies the Log-Sobolev inequality with constant $\frac{\mu}{2}$. This assumption is weaker than the dissipative gradient condition Raginsky et al. (2017); Zou et al. (2019) which is used commonly in non-log-concave sampling. We first present the main result and defer the proof to the appendix.

Theorem 3.4 (Main Theorem for QZ-LMC). *Under Assumption 3.3, let μ_k be the distribution of \mathbf{x}_k in QZ-LMC algorithm. Then, if we set the step size $\eta = \mathcal{O}\left(\frac{\epsilon\alpha}{dL^2}\right)$, $K = \tilde{\mathcal{O}}\left(\frac{dL^2 \log(\text{KL}(\mu_0||\pi))}{\epsilon\alpha^2}\right)$, and $\hat{\sigma}^2 = \mathcal{O}(\alpha\epsilon)$, we have*

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

In addition, under Assumptions 2.1 and 2.2, 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 2.1 and 2.5 the query complexity to the stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^{3/2} L^2 \sigma}{\alpha^{5/2} \epsilon^{3/2}}\right)$.

Note that this result holds for various distance metrics. Comparing to the classical results, Roy et al. (2021) analyzed the convergence of LMC in the zeroth-order setting under Assumptions 2.1 and 2.2 and established evaluation complexity $\mathcal{O}(d^3\sigma^2/\epsilon^4)$ for convergence in \mathbf{W}_2 distance (Theorem 3.2 in Roy et al. (2021)). Our algorithm uses $\tilde{\mathcal{O}}(d^2\sigma/\epsilon^3)$ evaluation queries under the same assumptions giving polynomial speedups in d, σ and ϵ .

4 APPLICATION TO NONSMOOTH AND NONCONVEX OPTIMIZATION

Although we assumed smoothness for f in previous sections, our results also help in nonsmooth sampling and optimization problems. In this section, we consider the following problem. Suppose that f satisfies the following assumptions.

Assumption 4.1 (Approximate-Convexity). Let f be a differentiable function, we say that $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is an ϵ -approximately convex function, if there exists a strongly convex function F such that for all \mathbf{x} ,

$$|F(\mathbf{x}) - f(\mathbf{x})| \leq \frac{\epsilon}{d}. \quad (17)$$

Instead of smoothness, we only assume that f is Lipschitz continuous.

Assumption 4.2. For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $f : \mathbb{R}^d \rightarrow \mathbb{R}$ satisfies,

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq M \|\mathbf{x} - \mathbf{y}\|. \quad (18)$$

The goal is to find an approximate minimizer \mathbf{x}^* such that $|f(\mathbf{x}^*) - \min_{\mathbf{x}} f(\mathbf{x})| \leq \epsilon$. Although these assumptions are very strong, this problem commonly appears in empirical risk minimization

(ERM) tasks where f is given by empirical observations. Even the population function is smooth and strongly convex, the empirical optimization function loses its smoothness and convexity due to subsampling. Similar settings have also appeared in the context of escaping from local minima both in classical Belloni et al. (2015) and quantum settings Li & Zhang (2024) with access to a stochastic evaluation oracle. Diverging from previous techniques, we address these problems by sampling from Gibbs sampling by using smooth approximation of f as proxy. Since f is not smooth, we consider the smoothed approximation

$$f_v(\mathbf{x}) = \mathbb{E}_{\mathbf{u} \sim \mathcal{B}_d(0,1)}[f(\mathbf{x} + v\mathbf{u})] \quad (19)$$

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

Proposition 4.3. *If f satisfies Assumption 4.2, then f_v satisfies*

- $|f_v(\cdot) - f(\cdot)| \leq vM$ and $|f_v(\mathbf{x}) - f_v(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|$,
- $|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})| \leq cM\sqrt{d}v^{-1}$ for some constant $c > 0$.

First we notice that,

$$\mathbb{E}_{\mathbf{u}}\|\nabla f(\mathbf{x} + v\mathbf{u}) - \nabla f_v(\mathbf{x})\|^2 \leq 4M^2 \quad (20)$$

as $\|\nabla f(\mathbf{x})\| \leq M$ because of Lipschitz continuity. Hence, Assumption 2.1 holds with $\sigma^2 = 4M^2$ and Assumption 2.2 holds with $L = \frac{cM\sqrt{d}}{v}$. Therefore, using Theorem 3.4, we can sample from the Gibbs-Boltzmann distribution with potential f_v faster than the classical algorithms. This is possible due to the fact that our quantum gradient estimation can compute the gradient of f_v if we consider $f(\mathbf{x} + v\mathbf{u})$ as the stochastic evaluation oracle. Then the seed set $\Xi = \mathcal{B}(0, 1)$ for which the sampling oracle can be prepared very efficiently. Since initial goal is to optimize f rather than to sample from the Gibbs distribution, we use the following lemma that describes a method to turn a sampling algorithm into an optimizer.

Lemma 4.4. *Let $\pi_v^\beta = \frac{e^{-\beta f_v(\mathbf{x})}}{\int e^{-\beta f_v(\mathbf{x})} d\mathbf{x}}$. If $\beta = \Theta(d/\epsilon)$ and $v \leq \mathcal{O}(\frac{\epsilon}{Md})$, then sampling from π_v^β returns ϵ approximate optimizer for f with probability at least 0.1.*

The essential idea is that Gibbs distribution gets localized around the global minima of f as β increases. One other challenge to sample from f_v is that the Log-Sobolev constant in general might depend on dimension. However, by using Holley–Stroock LSI perturbation result, we showed the explicit dimension dependence of α to characterize the total evaluation complexity in E. Next, we give our main result.

Theorem 4.5. *Suppose that f satisfies Assumptions 4.1 and 4.2. Then, there exists a quantum algorithm that returns ϵ approximate minimizer for f with probability at least 0.1 using $\tilde{\mathcal{O}}\left(\frac{d^{9/2}}{\epsilon^{3/2}}\right)$ queries to the stochastic evaluation oracle for f .*

The proof of Lemma 4.4 and Theorem 4.5 are postponed to Appendix E. The closest result to our setting is given by Li & Zhang (2024) and their query complexity in the stochastic setting is $\tilde{\mathcal{O}}(d^5/\epsilon)$ although their assumptions are slightly different. First, they assume that the noise is sub-Gaussian and additive. Furthermore, they assume F is convex in a bounded domain but not necessarily strongly convex. Noting that these differences might possibly make the classical results loose, our algorithm seems to give a speedup in dimension dependence with a small performance drop in terms of ϵ . However, this is a known trade-off in sampling algorithms. Since their algorithm uses a reversible sampler (hit-and-run walk), their ϵ dependence only comes from the quantum mean estimation. On the other hand, our algorithm uses a non-reversible sampler (also referred to as a low accuracy sampler) which typically gives better dependency on dimension but worse on accuracy. We also note that the classical algorithm by Belloni et al. (2015) takes $\tilde{\mathcal{O}}(\frac{d^{7.5}}{\epsilon^2})$ queries to the stochastic evaluation oracle.

Upon completion of this work, we became aware of recent studies by Augustino et al. Augustino et al. (2025) and Chakrabarti et al. Chakrabarti et al. (2025), which also investigate zeroth-order stochastic convex optimization under assumptions similar to those in Li & Zhang (2024). They propose algorithms with query complexities of $\tilde{\mathcal{O}}(d^{9/2}/\epsilon^7)$ and $\tilde{\mathcal{O}}(d^3/\epsilon^5)$, respectively. While both approaches exhibit worse dependence on ϵ compared to ours, we emphasize that the assumptions and problem settings differ significantly than ours.

REFERENCES

- 486
487
488 Brandon Augustino, Dylan Herman, Enrico Fontana, Junhyung Lyle Kim, Jacob Watkins, Shou-
489 vanik Chakrabarti, and Marco Pistoia. Fast convex optimization with quantum gradient methods,
490 2025. URL <https://arxiv.org/abs/2503.17356>.
- 491 Jack Baker, Paul Fearnhead, Emily B. Fox, and Christopher Nemeth. Control variates for stochastic
492 gradient mcmc. *Statistics and Computing*, 29(3):599–615, May 2019. ISSN 0960-3174. doi: 10.
493 1007/s11222-018-9826-2. URL <https://doi.org/10.1007/s11222-018-9826-2>.
- 494 Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-order nonconvex stochastic optimiza-
495 tion: Handling constraints, high dimensionality, and saddle points. *Found. Comput. Math.*,
496 22(1):35–76, February 2022. ISSN 1615-3375. doi: 10.1007/s10208-021-09499-8. URL
497 <https://doi.org/10.1007/s10208-021-09499-8>.
- 498 Alexandre Belloni, Tengyuan Liang, Hariharan Narayanan, and Alexander Rakhlin. Escaping the
499 local minima via simulated annealing: Optimization of approximately convex functions. In Peter
500 Grünwald, Elad Hazan, and Satyen Kale (eds.), *Proceedings of The 28th Conference on Learning*
501 *Theory*, volume 40 of *Proceedings of Machine Learning Research*, pp. 240–265, Paris, France,
502 03–06 Jul 2015. PMLR. URL [https://proceedings.mlr.press/v40/Belloni15.](https://proceedings.mlr.press/v40/Belloni15.html)
503 [html](https://proceedings.mlr.press/v40/Belloni15.html).
- 504 Shouvanik Chakrabarti, Andrew M. Childs, Tongyang Li, and Xiaodi Wu. Quantum algorithms and
505 lower bounds for convex optimization. *Quantum*, 4:221, January 2020. ISSN 2521-327X. doi: 10.
506 22331/q-2020-01-13-221. URL <https://doi.org/10.22331/q-2020-01-13-221>.
- 507 Shouvanik Chakrabarti, Dylan Herman, Jacob Watkins, Enrico Fontana, Brandon Augustino, Jun-
508 hyung Lyle Kim, and Marco Pistoia. On speedups for convex optimization via quantum dynamics,
509 2025. URL <https://arxiv.org/abs/2503.24332>.
- 510 David Chandler. Introduction to modern statistical. *Mechanics. Oxford University Press, Oxford,*
511 *UK*, 5(449):11, 1987.
- 512 Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In
513 Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Ma-*
514 *chine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1683–1691, Be-
515 jing, China, 22–24 Jun 2014. PMLR. URL [https://proceedings.mlr.press/v32/](https://proceedings.mlr.press/v32/chen14.html)
516 [chen14.html](https://proceedings.mlr.press/v32/chen14.html).
- 517 Xi Chen, Simon S. Du, and Xin T. Tong. On stationary-point hitting time and ergodicity of stochastic
518 gradient langevin dynamics. *Journal of Machine Learning Research*, 21(68):1–41, 2020. URL
519 <http://jmlr.org/papers/v21/19-327.html>.
- 520 Sinho Chewi and Austin J. Stromme. The ballistic limit of the log-sobolev constant equals the
521 polyak-lojasiewicz constant, 2024. URL <https://arxiv.org/abs/2411.11415>.
- 522 Sinho Chewi, Murat A Erdogdu, Mufan Li, Ruoqi Shen, and Shunshi Zhang. Analysis of langevin
523 monte carlo from poincare to log-sobolev. In Po-Ling Loh and Maxim Raginsky (eds.), *Pro-*
524 *ceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine*
525 *Learning Research*, pp. 1–2. PMLR, 02–05 Jul 2022. URL [https://proceedings.mlr.](https://proceedings.mlr.press/v178/chewi22a.html)
526 [press/v178/chewi22a.html](https://proceedings.mlr.press/v178/chewi22a.html).
- 527 Andrew M. Childs, Tongyang Li, Jin-Peng Liu, Chunhao Wang, and Ruizhe Zhang. Quantum
528 algorithms for sampling log-concave distributions and estimating normalizing constants. In
529 S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in*
530 *Neural Information Processing Systems*, volume 35, pp. 23205–23217. Curran Associates, Inc.,
531 2022. URL [https://proceedings.neurips.cc/paper_files/paper/2022/](https://proceedings.neurips.cc/paper_files/paper/2022/file/933e953353c25ec70477ef28e45a2dcc-Paper-Conference.pdf)
532 [file/933e953353c25ec70477ef28e45a2dcc-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/933e953353c25ec70477ef28e45a2dcc-Paper-Conference.pdf).
- 533 Arjan Cornelissen, Yassine Hamoudi, and Sofiene Jerbi. Near-optimal quantum algorithms for
534 multivariate mean estimation. In *Proceedings of the 54th Annual ACM SIGACT Symposium*
535 *on Theory of Computing*, STOC ’22. ACM, June 2022. doi: 10.1145/3519935.3520045. URL
536 <http://dx.doi.org/10.1145/3519935.3520045>.

- 540 Ben Cousins and Santosh Vempala. Gaussian cooling and $o^*(n^3)$ algorithms for volume and gaussian
541 volume. *SIAM Journal on Computing*, 47(3):1237–1273, 2018. doi: 10.1137/15M1054250.
542 URL <https://doi.org/10.1137/15M1054250>.
543
- 544 Arnak S. Dalalyan and Avetik Karagulyan. User-friendly guarantees for the langevin monte
545 carlo with inaccurate gradient. *Stochastic Processes and their Applications*, 129(12):5278–
546 5311, 2019. ISSN 0304-4149. doi: <https://doi.org/10.1016/j.spa.2019.02.016>. URL <https://www.sciencedirect.com/science/article/pii/S0304414918304824>.
547
- 548 John C. Duchi, Michael I. Jordan, Martin J. Wainwright, and Andre Wibisono. Optimal rates for
549 zero-order convex optimization: The power of two function evaluations. *IEEE Trans. Inf. Theor.*,
550 61(5):2788–2806, May 2015. ISSN 0018-9448. doi: 10.1109/TIT.2015.2409256. URL <https://doi.org/10.1109/TIT.2015.2409256>.
551
- 552 Alain Durmus and Eric Moulines. High-dimensional bayesian inference via the unadjusted langevin
553 algorithm, 2018. URL <https://arxiv.org/abs/1605.01559>.
554
- 555 Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Appli-*
556 *cations*, volume 1 of *Computational Science Series*. Academic Press, San Diego, second edition,
557 2002.
558
- 559 András Gilyén, Srinivasan Arunachalam, and Nathan Wiebe. *Optimizing quantum optimization*
560 *algorithms via faster quantum gradient computation*, pp. 1425–1444. Society for Industrial and
561 Applied Mathematics, January 2019. ISBN 9781611975482. doi: 10.1137/1.9781611975482.87.
562 URL <http://dx.doi.org/10.1137/1.9781611975482.87>.
563
- 564 Richard Holley and Daniel W. Stroock. Logarithmic sobolev inequalities and stochastic ising mod-
565 els. *Journal of Statistical Physics*, 46:1159–1194, 1987. URL <https://link.springer.com/article/10.1007/BF01011161>.
566
- 567 Stephen P. Jordan. Fast quantum algorithm for numerical gradient estimation. *Physical Review*
568 *Letters*, 95(5), July 2005. ISSN 1079-7114. doi: 10.1103/physrevlett.95.050501. URL <http://dx.doi.org/10.1103/PhysRevLett.95.050501>.
569
- 570 Yuri Kinoshita and Taiji Suzuki. Improved convergence rate of stochastic gradient langevin
571 dynamics with variance reduction and its application to optimization. In S. Koyejo,
572 S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural*
573 *Information Processing Systems*, volume 35, pp. 19022–19034. Curran Associates, Inc.,
574 2022. URL [https://proceedings.neurips.cc/paper_files/paper/2022/](https://proceedings.neurips.cc/paper_files/paper/2022/file/78e839f96568985d18463044a064ea0f-Paper-Conference.pdf)
575 [file/78e839f96568985d18463044a064ea0f-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/78e839f96568985d18463044a064ea0f-Paper-Conference.pdf).
576
- 577 Tongyang Li and Ruizhe Zhang. Quantum speedups of optimizing approximately convex functions
578 with applications to logarithmic regret stochastic convex bandits. In *Proceedings of the 36th*
579 *International Conference on Neural Information Processing Systems*, NIPS ’22, Red Hook, NY,
580 USA, 2024. Curran Associates Inc. ISBN 9781713871088.
- 581 Tianyi Lin, Zeyu Zheng, and Michael Jordan. Gradient-free methods for determinis-
582 tic and stochastic nonsmooth nonconvex optimization. In S. Koyejo, S. Mohamed,
583 A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Infor-*
584 *mation Processing Systems*, volume 35, pp. 26160–26175. Curran Associates, Inc.,
585 2022. URL [https://proceedings.neurips.cc/paper_files/paper/2022/](https://proceedings.neurips.cc/paper_files/paper/2022/file/a78f142aec481e68c75276756e0a0d91-Paper-Conference.pdf)
586 [file/a78f142aec481e68c75276756e0a0d91-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/a78f142aec481e68c75276756e0a0d91-Paper-Conference.pdf).
587
- 588 Chengchang Liu, Chaowen Guan, Jianhao He, and John C.S. Lui. Quantum algorithms for non-
589 smooth non-convex optimization. In *The Thirty-eighth Annual Conference on Neural Information*
590 *Processing Systems*, 2024.
- 591 László Lovász and Santosh Vempala. Simulated annealing in convex bodies and an $o^*(n^4)$ volume
592 algorithm. *Journal of Computer and System Sciences*, 72(2):392–417, 2006. ISSN 0022-0000.
593 doi: <https://doi.org/10.1016/j.jcss.2005.08.004>. URL <https://www.sciencedirect.com/science/article/pii/S0022000005000966>. JCSS FOCS 2003 Special Issue.

- 594 Yi-An Ma, Yuansi Chen, Chi Jin, Nicolas Flammarion, and Michael I. Jordan. Sampling can
595 be faster than optimization. *Proceedings of the National Academy of Sciences*, 116(42):
596 20881–20885, September 2019. ISSN 1091-6490. doi: 10.1073/pnas.1820003116. URL
597 <http://dx.doi.org/10.1073/pnas.1820003116>.
598
- 599 Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex func-
600 tions. *Found. Comput. Math.*, 17(2):527–566, April 2017. ISSN 1615-3375. doi: 10.1007/
601 s10208-015-9296-2. URL <https://doi.org/10.1007/s10208-015-9296-2>.
- 602 Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic
603 gradient langevin dynamics: a nonasymptotic analysis. In Satyen Kale and Ohad Shamir (eds.),
604 *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine
605 Learning Research*, pp. 1674–1703. PMLR, 07–10 Jul 2017. URL [https://proceedings.
606 mlr.press/v65/raginsky17a.html](https://proceedings.mlr.press/v65/raginsky17a.html).
- 607
608 Abhishek Roy, Lingqing Shen, Krishnakumar Balasubramanian, and Saeed Ghadimi. Stochastic
609 zeroth-order discretizations of langevin diffusions for bayesian inference, 2021. URL [https://
610 arxiv.org/abs/1902.01373](https://arxiv.org/abs/1902.01373).
- 611 Aaron Sidford and Chenyi Zhang. Quantum speedups for stochastic optimization. In A. Oh,
612 T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neu-
613 ral Information Processing Systems*, volume 36, pp. 35300–35330. Curran Associates, Inc.,
614 2023. URL [https://proceedings.neurips.cc/paper_files/paper/2023/
615 file/6ed9931d6e1fb6a85efalb2c014a47e1-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2023/file/6ed9931d6e1fb6a85efalb2c014a47e1-Paper-Conference.pdf).
- 616
617 Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Convex optimization
618 using quantum oracles. *Quantum*, 4:220, January 2020. ISSN 2521-327X. doi: 10.22331/
619 q-2020-01-13-220. URL <https://doi.org/10.22331/q-2020-01-13-220>.
- 620 Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted langevin algorithm:
621 Isoperimetry suffices. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox,
622 and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Cur-
623 ran Associates, Inc., 2019. URL [https://proceedings.neurips.cc/paper_files/
624 paper/2019/file/65a99bb7a3115fdede20da98b08a370f-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/65a99bb7a3115fdede20da98b08a370f-Paper.pdf).
- 625
626 Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. Privacy for free: Posterior sampling and
627 stochastic gradient monte carlo. In Francis Bach and David Blei (eds.), *Proceedings of the
628 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine
629 Learning Research*, pp. 2493–2502, Lille, France, 07–09 Jul 2015. PMLR. URL [https://
630 proceedings.mlr.press/v37/wangg15.html](https://proceedings.mlr.press/v37/wangg15.html).
- 631
632 Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics.
633 In *Proceedings of the 28th International Conference on International Conference on Machine
634 Learning*, ICML’11, pp. 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.
- 635
636 Kaylee Yingxi Yang and Andre Wibisono. Convergence of the inexact langevin algorithm and score-
637 based generative models in kl divergence, 2023. URL [https://arxiv.org/abs/2211.
638 01512](https://arxiv.org/abs/2211.01512).
- 639
640 Yexin Zhang, Chenyi Zhang, Cong Fang, Liwei Wang, and Tongyang Li. Quantum algorithms
641 and lower bounds for finite-sum optimization. In Ruslan Salakhutdinov, Zico Kolter, Kather-
642 ine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Pro-
643 ceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceeed-
644 ings of Machine Learning Research*, pp. 60244–60270. PMLR, 21–27 Jul 2024. URL [https://
645 proceedings.mlr.press/v235/zhang24bz.html](https://proceedings.mlr.press/v235/zhang24bz.html).
- 646
647 Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient
langevin dynamics. In Satyen Kale and Ohad Shamir (eds.), *Proceedings of the 2017 Conference
on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pp. 1980–2022.
PMLR, 07–10 Jul 2017. URL [https://proceedings.mlr.press/v65/zhang17b.
html](https://proceedings.mlr.press/v65/zhang17b.html).

648 Difan Zou and Quanquan Gu. On the convergence of hamiltonian monte carlo with stochastic
649 gradients. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International*
650 *Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*,
651 pp. 13012–13022. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/zou21b.html>.
652

653 Difan Zou, Pan Xu, and Quanquan Gu. Stochastic gradient hamiltonian monte carlo methods with
654 recursive variance reduction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc,
655 E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32.
656 Curran Associates, Inc., 2019. URL [https://proceedings.neurips.cc/paper_](https://proceedings.neurips.cc/paper_files/paper/2019/file/c3535febaaff29fcb7c0d20cbe94391c7-Paper.pdf)
657 [files/paper/2019/file/c3535febaaff29fcb7c0d20cbe94391c7-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/c3535febaaff29fcb7c0d20cbe94391c7-Paper.pdf).
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701

A BACKGROUND

In this section, we give background on some classical and quantum algorithms for various tasks that are repeatedly referred in the main text.

A.1 OVERVIEW OF HAMILTONIAN MONTE CARLO ALGORITHM

Hamiltonian Monte Carlo (HMC) is an advanced sampling technique designed to efficiently explore high-dimensional probability distributions by introducing auxiliary momentum variables. Given a target distribution $\pi(\mathbf{x}) \propto e^{-f(\mathbf{x})}$, HMC augments the state space with momentum variables \mathbf{p} and defines the Hamiltonian $H(\mathbf{x}, \mathbf{p}) = f(\mathbf{x}) + \frac{1}{2}\|\mathbf{p}\|^2$ where $\mathbf{p} \sim \mathcal{N}(0, I)$.

HMC alternates between updating the position \mathbf{x} and momentum \mathbf{p} by simulating Hamiltonian dynamics Eq. (4). In practice, Hamiltonian dynamics is simulated using the leapfrog integrator, which discretizes the continuous equations of motion. The key advantage of HMC is that it allows for large, efficient moves through the parameter space by leveraging gradient information and auxiliary momentum. This reduces the correlation between successive samples, particularly in high-dimensional spaces, resulting in faster convergence compared to simple random-walk methods like the Metropolis-Hastings algorithm. In practice, Hamiltonian dynamics are simulated using the leapfrog integrator, which discretizes the continuous equations of motion. The leapfrog method proceeds in three steps:

$$\begin{aligned} \mathbf{p}_{k+\frac{1}{2}} &= \mathbf{p}_k - \frac{\eta}{2} \nabla f(\mathbf{x}_k), \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \eta \mathbf{p}_{k+\frac{1}{2}}, \\ \mathbf{p}_{k+1} &= \mathbf{p}_{k+\frac{1}{2}} - \frac{\eta}{2} \nabla f(\mathbf{x}_{k+1}), \end{aligned}$$

where η is the step size. 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.

Algorithm 2 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**

$k = St + s$

$\mathbf{x}_{k+1} = \mathbf{x}_k + \eta \mathbf{p}_k - \frac{\eta^2}{2} \mathbf{g}(\mathbf{x}_k, \xi_k)$

$\mathbf{p}_{k+1} = \mathbf{p}_k - \frac{\eta}{2} \mathbf{g}(\mathbf{x}_k, \xi_k) - \frac{\eta}{2} \mathbf{g}(\mathbf{x}_{k+1}, \xi_{k+1/2})$

end for

end for

Return \mathbf{x}^T

Similar to SGLD, one can replace the gradients with stochastic gradients resulting in SG-HMC (See Algorithm 2). The stochastic gradients $\mathbf{g}(\mathbf{x}, \xi)$ in Algorithm 2 can be obtained using different techniques such as mini-batch, SVRG, CV, or even zeroth-order methods. In this case, we use quantum variance reduction techniques to compute $\mathbf{g}(\mathbf{x}, \xi)$.

A.2 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 would not provide an unbiased estimate for the gradient. Suppose that we have an algorithm `BiasedStochasticGradient`(\mathbf{x}, σ) that outputs \mathbf{v} such that $\mathbb{E}\|\mathbf{v} - \nabla f(\mathbf{x})\| \leq \hat{\sigma}^2$ with cost $\tilde{O}\left(\frac{C}{\hat{\sigma}}\right)$ where C is a function of other problem parameters. Consider the following algorithm.

Algorithm 3 UnbiasedStochasticGradient

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

Lemma A.1. *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 $\tilde{O}(\frac{C}{\hat{\sigma}})$, the algorithm `UnbiasedStochasticGradient` 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 $\tilde{O}(\frac{C}{\hat{\sigma}})$.*

Proof. We repeat the proof in Sidford & Zhang (2023).

$$\mathbf{g} = \mathbf{g}_0 + 2^J(\mathbf{g}_J - \mathbf{g}_{J-1}), \quad J \sim \text{Geom}\left(\frac{1}{2}\right) \in \mathbb{N}. \quad (21)$$

Given that $\Pr(J = j) = 2^{-j}$, we have

$$\mathbb{E}[\mathbf{g}] = \mathbb{E}[\mathbf{g}_0] + \sum_{j=1}^{\infty} \Pr(J = j) 2^j (\mathbb{E}[\mathbf{g}_j] - \mathbb{E}[\mathbf{g}_{j-1}]) = \mathbb{E}[\mathbf{g}_{\infty}] = \nabla f(\mathbf{x}). \quad (22)$$

As for the variance, using the inequality $(a + b)^2 \leq 2a^2 + 2b^2$, we have

$$\mathbb{E}\|\mathbf{g} - \nabla f(\mathbf{x})\|^2 \leq 2\mathbb{E}\|\mathbf{g} - \mathbf{g}_0\|^2 + 2\mathbb{E}\|\mathbf{g}_0 - \nabla f(\mathbf{x})\|^2 \quad (23)$$

where

$$\mathbb{E}\|\mathbf{g} - \mathbf{g}_0\|^2 = \sum_{j=1}^{\infty} \Pr(J = j) 2^{2j} \mathbb{E}\|\mathbf{g}_j - \mathbf{g}_{j-1}\|^2 = \sum_{j=1}^{\infty} 2^j \mathbb{E}\|\mathbf{g}_j - \mathbf{g}_{j-1}\|^2, \quad (24)$$

and for each j we have

$$\mathbb{E}\|\mathbf{g}_j - \mathbf{g}_{j-1}\|^2 \leq 2\mathbb{E}\|\mathbf{g}_j - \nabla f(\mathbf{x})\|^2 + 2\mathbb{E}\|\mathbf{g}_{j-1} - \nabla f(\mathbf{x})\|^2. \quad (25)$$

By assumption on `BiasedStochasticGradient`,

$$\mathbb{E}\|\mathbf{g}_j - \nabla f(\mathbf{x})\|^2 \leq \frac{\hat{\sigma}^2}{100 \cdot 2^{3j/2}}, \quad \forall j \geq 0, \quad (26)$$

which leads to

$$\mathbb{E}\|\mathbf{g}_j - \mathbf{g}_{j-1}\|^2 \leq \frac{\hat{\sigma}^2}{50 \cdot 2^{3(j-1)/2}} + \frac{\hat{\sigma}^2}{50 \cdot 2^{3j/2}} \leq \frac{\hat{\sigma}^2}{10 \cdot 2^{3j/2}}, \quad (27)$$

and

$$\mathbb{E}\|\mathbf{g} - \mathbf{g}_0\|^2 = \frac{\hat{\sigma}^2}{10} \sum_{j=1}^{\infty} \frac{1}{2^{j/2}} \leq \frac{1}{3} \hat{\sigma}^2. \quad (28)$$

Hence,

$$\mathbb{E}\|\mathbf{g} - \nabla f(\mathbf{x})\|^2 \leq 2\mathbb{E}\|\mathbf{g} - \mathbf{g}_0\|^2 + 2\mathbb{E}\|\mathbf{g}_0 - \nabla f(\mathbf{x})\|^2 \leq \hat{\sigma}^2, \quad (29)$$

Moreover, the expected cost is

$$\tilde{O}\left(\frac{C}{\hat{\sigma}}\right) \cdot \left(1 + \sum_{j=1}^{\infty} \Pr\{J = j\} \cdot \left(2^{3j/4} + 2^{3(j-1)/4}\right)\right) = \tilde{O}\left(\frac{C}{\hat{\sigma}}\right). \quad (30)$$

□

810 A.3 QUANTUM MEAN ESTIMATION

811 We assume access to the following oracle for a probability distribution.

812 **Definition A.2** (Quantum Sampling Oracle). Quantum sampling oracle O_X of a random variable
813 $X \in \Omega$ is given by $O_X |0\rangle |0\rangle \mapsto \sum_{X \in \Omega} \sqrt{\Pr(X)} |X\rangle |\text{garbage}(X)\rangle$.

814 Here, the second register contains $|\text{garbage}(X)\rangle$, which depends on X . The state in the (auxiliary)
815 garbage register is usually generated in some intermediate step of computing X in the first register. It
816 is important to note that the state in this quantum sampling oracle differs from the coherent quantum
817 sample state, as the former is entangled and we cannot simply discard the garbage register.

818 Quantum mean estimation is a technique to estimate the mean of a d -dimensional random variable
819 X up to ϵ accuracy using $\tilde{O}(d^{1/2}/\epsilon)$ queries, which is a quadratic improvement in ϵ compared to
820 classical algorithms Cornelissen et al. (2022). Although the quantum mean estimation algorithm
821 is biased, Sidford & Zhang (2023) developed an unbiased quantum mean estimation algorithm.
822 Specifically, for a multi-dimensional variable with mean μ and variance σ^2 , unbiased quantum mean
823 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 $\tilde{O}(d^{1/2}\sigma/\hat{\sigma})$
824 queries.

825 **Lemma A.3** (Unbiased Quantum Mean Estimation Sidford & Zhang (2023)). For a d -dimensional
826 random variable X with $\text{Var}[X] \leq \sigma^2$ and some $\hat{\sigma} \geq 0$, suppose we are given access
827 to its quantum sampling oracle O_X (as in Definition A.2). Then, there is a procedure
828 $\text{QuantumMeanEstimation}(O_X, \hat{\sigma})$ that uses $\tilde{O}\left(\frac{d^{1/2}\sigma}{\hat{\sigma}}\right)$ queries to O_X and outputs an unbi-
829 ased estimate $\hat{\mu}$ of the expectation μ satisfying $\text{Var}[\hat{\mu}] \leq \hat{\sigma}^2$.

833 A.4 OVERVIEW OF JORDAN’S ALGORITHM

834 Jordan’s algorithm Jordan (2005) approximates the gradient using a finite difference formula on a
835 small grid around the point of interest and encodes the estimate into the quantum phase. Then,
836 the algorithm applies an inverse quantum Fourier transform to estimate the gradient. Although
837 Jordan’s original analysis implicitly assumes that higher-order derivatives of the function are neg-
838 ligible, Gilyén, Arunachalam, and Wiebe Gilyén et al. (2019) analyzed the algorithm and extended
839 it to handle functions in the Gevrey class, using central difference formulas and a binary oracle
840 model commonly encountered in variational quantum algorithms. The closest analysis of Jordan’s
841 algorithm to our setting was provided by Chakrabarti et al. (2020), who demonstrated that Algo-
842 rithm 4 achieves constant query complexity for functions with Lipschitz gradients, provided that the
843 function values can be queried with high precision.

844 The following lemma from Chakrabarti et al. (2020) demonstrates that Algorithm 4 achieves $\tilde{O}(1)$
845 query complexity for evaluating the gradient of a β -smooth function with high probability.

846 **Lemma A.4** (Lemma 2.2 in Chakrabarti et al. (2020)). Let $f: \mathbb{R}^d \rightarrow \mathbb{R}$ be a function that is acces-
847 sible via an evaluation oracle with error at most ϵ . Assume that $\|\nabla f\| \leq L$ and f is β -smooth in
848 $B_\infty(x, 2\sqrt{\epsilon/\beta})$. Let $\tilde{\mathbf{g}}$ be the output of $\text{QuantumGradient}(f, \epsilon, M, \beta, \mathbf{x}_0)$ (as defined in Algo-
849 rithm 4). Then:

$$850 \Pr \left[|\tilde{\mathbf{g}}_i - \nabla f(\mathbf{x})_i| > 1500\sqrt{d\epsilon\beta} \right] < \frac{1}{3}, \quad \forall i \in [d]. \quad (31)$$

851 Although Algorithm 4 results in an accurate estimate for the gradient with high probability, it is
852 possible to run the algorithm multiple times and take the coordinate-wise median of the outputs to
853 obtain a smooth estimate for the gradient (Lemma 2.3 in Chakrabarti et al. (2020)) when the norm
854 of the gradient is bounded. To estimate the gradient up to δ error (in L_2 norm), it is required to have
855 an evaluation oracle with error at most $\mathcal{O}(\delta^2/d^2)$ which might not be feasible if the noisy evaluation
856 oracle is stochastic.

857 Our algorithms work in the stochastic setting where we prove that we can create an accurate eval-
858 uation oracle under Assumptions 2.1 and 2.2. Furthermore, the function f needs to be smooth;
859 however, under Assumptions 2.1 and 2.5 the smoothness constant is not bounded and this might
860 cause unbounded error. We propose a robust version of Algorithm 4 so that we can still estimate the
861 gradient accurately (See the step-by-step description in Section 2.2). We also note that the oracle
862 O_F is known as the phase oracle. Our oracle (Eq. (6)) can be converted to phase oracle efficiently.
863

Algorithm 4 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 $\frac{24\pi\sqrt{d\epsilon\beta}}{L} \leq \frac{1}{2^b} = \frac{1}{N} \leq \frac{48\pi\sqrt{d\epsilon\beta}}{L}$,
- $b_0 \in \mathbb{N}$ such that $\frac{N\epsilon}{2Ll} \leq \frac{1}{2^{b_0}} = \frac{1}{N_0} \leq \frac{N\epsilon}{Ll}$,
- $F(x) = \frac{N}{2Ll}[f(x_0 + \frac{l}{N}(x - N/2)) - f(x_0)]$, and,
- $\gamma : \{0, 1, \dots, N-1\} \rightarrow G := \{-N/2, -N/2+1, \dots, N/2-1\}$ 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 $|\tilde{F}(x) - F(x)| \leq \frac{1}{N_0}$, 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

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

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, \dots, k_n and report $\tilde{\mathbf{g}} = \frac{2L}{N}(k_1, k_2, \dots, k_n)$ as the result.

B PROOFS FOR HAMILTONIAN MONTE CARLO IN STRONGLY CONVEX CASE

We start with the following result in Zou & Gu (2021) that quantifies the convergence of the stochastic gradient Hamiltonian Monte Carlo algorithm in Wasserstein distance.

Theorem B.1 (Theorem 4.4 in Zou & Gu (2021)). *Under Assumptions 2.2 and 3.1, let $D = \|\mathbf{x}^0 - \arg \min_{\mathbf{x}}(f(\mathbf{x}))\|$ and μ_T be the distribution of the iterate \mathbf{x}^T , then if the step size satisfies $\eta = O(L^{1/2}\sigma^{-2}\kappa^{-1} \wedge L^{-1/2})$ and $K = 1/(4\sqrt{L}\eta)$, the output of HMC satisfies*

$$W_2(\mu_T, \pi) \leq (1 - (128\kappa)^{-1})^{\frac{T}{2}} (2D + 2d/\mu)^{1/2} + \Gamma_1\eta^{1/2} + \Gamma_2\eta, \quad (32)$$

where $\Gamma_1^2 = O(L^{-3/2}\sigma^2\kappa^2)$ and $\Gamma_2^2 = O(\kappa^2(LD + \kappa d + L^{-1/2}\sigma^2\eta))$ where $\sigma^2 = \max_{t \leq T} \mathbb{E}\|\mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k) - \nabla f(\mathbf{x}_k)\|^2$ is the upper bound on the variance of the gradients in the trajectory of SG-HMC algorithm.

This is a generic result that applies to any HMC algorithm under Assumptions 2.2 and 3.1 that uses stochastic gradients with variance upper bounded by σ^2 . Note that we do not assume a uniform upper bound for σ that is independent of problem parameters. Instead, the variance upper bound depends on the trajectory of the algorithm, which can be characterized using theoretical analysis. The analysis in Zou & Gu (2021) assumes Lipschitzness of the stochastic gradients, and this follows from the smoothness of f in this case as we set the final error in our gradient estimation to ϵ so that the gradients are very close to gradient of ∇f .

B.1 PROOF OF QZ-HMC

Theorem 3.2 (Main Theorem for QZ-HMC). *Let μ_k be the distribution of \mathbf{x}_k in QZ-HMC algorithm. Suppose that f satisfies Assumption 3.1. Given that the initial point \mathbf{x}_0 satisfies*

$$\|\mathbf{x}_0 - \arg \min_{\mathbf{x}} f(\mathbf{x})\| \leq \frac{d}{\mu}, \text{ if we set the step size } \eta = \mathcal{O}\left(\frac{\epsilon}{d^{1/2}\kappa^{3/2}}\right), S = \tilde{\mathcal{O}}\left(\frac{Ld^{1/2}\kappa^{3/2}}{\epsilon}\right),$$

$T = \tilde{\mathcal{O}}(1)$, and $\hat{\sigma}^2 = \mathcal{O}\left(\frac{L^{3/2}d^{1/2}\epsilon}{\kappa^{3/2}}\right)$, we have

$$W_2(\mu_{ST}, \pi) \leq \epsilon.$$

In addition, under Assumptions 2.1 and 2.2, 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 2.1 and 2.5 the query complexity to the stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^{3/4}\sigma}{\epsilon^{3/2}}\right)$.

Proof. By Theorem B.1 for $\eta = \mathcal{O}(L^{1/2}\sigma^{-2}\kappa^{-1} \wedge L^{-1/2})$ and $K = \frac{1}{4\sqrt{L}\eta}$, we have

$$\mathbf{W}_2(\mu_T, \pi) \leq (1 - (128\kappa)^{-1})^{\frac{T}{2}} (2D + 2d/\mu)^{1/2} + \Gamma_1\eta^{1/2} + \Gamma_2\eta, \quad (33)$$

where

$$\Gamma_1 = \mathcal{O}\left(L^{-3/2}\hat{\sigma}^2\kappa^2\right), \quad (34)$$

$$\Gamma_2 = \mathcal{O}\left(\kappa^3d\right). \quad (35)$$

The first term in error is $\mathcal{O}(\epsilon)$ when $T = \tilde{\mathcal{O}}(\log(1/\epsilon))$. The last two terms become $\mathcal{O}(L^{-3/4}\hat{\sigma}\eta^{1/2} + d^{1/2}\kappa^{3/2}\eta)$. For $\hat{\sigma} = \mathcal{O}(L^{3/4}d^{1/4}\kappa^{-3/4}\epsilon^{1/2} \wedge \sigma)$ and $\eta = \mathcal{O}(\epsilon d^{-1/2}\kappa^{-3/2})$, the bias term becomes $\mathcal{O}(\epsilon)$. Then, under Assumptions 2.1 and 2.2, the number of calls to evaluation oracle scale as $\tilde{\mathcal{O}}(d^{1/2}\kappa^{3/2}\epsilon^{-1} + \sigma d^{3/4}\kappa^{3/4}\epsilon^{-3/2}) = \tilde{\mathcal{O}}(d^{3/4}\kappa^{3/4}\epsilon^{-3/2})$. Similarly, under Assumptions 2.1 and 2.5 the evaluation complexity is $\tilde{\mathcal{O}}(\sigma d^{5/4}\kappa^{3/4}\epsilon^{-3/2})$. \square

C PROOFS FOR LSI CASE

Lemma C.1 (Stochastic-LMC One Step Convergence). *Let μ_k be the distribution of the iterate \mathbf{x}_k , then if the step size satisfies $\eta = \frac{2}{3\alpha}$,*

$$\text{KL}(\mu_{k+1}|\pi) \leq e^{-3\alpha\eta/2} \left[\left(1 + \frac{32\eta^3 L^4}{\alpha}\right) \text{KL}(\mu_k|\pi) + 6\eta\sigma_k^2 + 16\eta^2 dL^2 \right], \quad (36)$$

where $\sigma_k^2 = \mathbb{E}_{\mathbf{x}_k, \boldsymbol{\xi}_k} \|\mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k) - \nabla f(\mathbf{x}_k)\|^2$.

Proof. We compare one step of LMC starting at \mathbf{x}_k with stochastic gradients $\mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k)$ to the output of continuous Langevin SDE (Eq. (2)) starting at \mathbf{x}_k with true gradient $\nabla f(\mathbf{x}_t)$ after time η . This technique has been used to establish the convergence of unadjusted Langevin algorithm with full gradients under isoperimetry by Vempala & Wibisono (2019). We extend the analysis by Vempala & Wibisono (2019) to the stochastic gradient LMC. Assume that the initial point \mathbf{x}_k and $\mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k)$ obey the joint distribution μ_0 . The randomness on $\mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k)$ depends both on the randomness on \mathbf{x}_k and the randomness in the quantum mean estimation algorithm. Then, one step update of LMC algorithm with stochastic gradient yields,

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

Alternatively, \mathbf{x}_{k+1} can be written as the solution of the following SDE at time $t = \eta$,

$$d\mathbf{x}_t = -\mathbf{g}_k dt + \sqrt{2}d\mathbf{W}_t$$

where $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k)$ and \mathbf{W}_t is the standard Brownian motion starting at $\mathbf{W}_0 = 0$. Let $\mu_t(\mathbf{x}_k, \mathbf{g}_k, \mathbf{x}_t)$ be the joint distribution of $\mathbf{x}_k, \mathbf{g}_k$, and \mathbf{x}_t at time t . Each expectation in the proof is over this joint distribution unless specified otherwise.

Consider the following stochastic differential equation

$$d\mathbf{X} = \mathbf{v}(\mathbf{X})dt + \sqrt{2}d\mathbf{W},$$

where \mathbf{v} is a smooth vector field and \mathbf{W} is the Brownian motion with $\mathbf{W}_0 = 0$. The Fokker-Planck equation describes the evolution of probability density function μ_t as follows:

$$\frac{\partial\mu_t}{\partial t} = -\nabla \cdot (\mu_t\mathbf{v}) + \Delta\mu_t, \quad (37)$$

where $\nabla \cdot$ is the divergence operator and Δ is the Laplacian. Then, the Fokker Planck equation gives the following evolution for the marginal density $\mu_t(\mathbf{x}|\mathbf{x}_k, \mathbf{g}_k) = \mu_t(\mathbf{x}_t = \mathbf{x}|\mathbf{x}_k, \mathbf{g}_k)$,

$$\frac{\partial \mu_t(\mathbf{x}|\mathbf{x}_k, \mathbf{g}_k)}{\partial t} = \nabla \cdot (\mu_t(\mathbf{x}|\mathbf{x}_k, \mathbf{g}_k)\mathbf{g}_k) + \Delta \mu_t(\mathbf{x}|\mathbf{x}_k, \mathbf{g}_k). \quad (38)$$

Taking the expectation over both sides with respect to $(\mathbf{x}_k, \mathbf{g}_k) \sim \mu_0$,

$$\frac{\partial \mu_t(\mathbf{x})}{\partial t} = \mathbb{E}_{(\mathbf{x}_k, \mathbf{g}_k) \sim \mu_0} [\nabla \cdot (\mu_t(\mathbf{x}|\mathbf{x}_k)\mathbf{g}_k)] + \mathbb{E}_{(\mathbf{x}_k, \mathbf{g}_k) \sim \mu_0} [\Delta \mu_t(\mathbf{x}|\mathbf{x}_k)] \quad (39)$$

$$= \int_{\mathbb{R}^d} \nabla \cdot (\mu_t(\mathbf{x}|\mathbf{x}_k, \mathbf{g}_k)\mathbf{g}_k) \mu_0(\mathbf{x}_k, \mathbf{g}_k) d\mathbf{x}_k d\mathbf{g}_k + \int_{\mathbb{R}^d} \Delta \mu_t(\mathbf{x}|\mathbf{x}_k, \mathbf{g}_k) \mu_0(\mathbf{x}_k, \mathbf{g}_k) d\mathbf{x}_k d\mathbf{g}_k \quad (40)$$

$$= \int_{\mathbb{R}^d} \nabla \cdot (\mu_t(\mathbf{x})\mu(\mathbf{x}_k, \mathbf{g}_k|\mathbf{x}_t = \mathbf{x})\mathbf{g}_k) d\mathbf{x}_k d\mathbf{g}_k + \Delta \mu_t(\mathbf{x}) \quad (41)$$

$$= \nabla \cdot \left(\mu_t(\mathbf{x}) \mathbb{E}[\mathbf{g}_k - \nabla f(\mathbf{x}_k)|\mathbf{x}_t = \mathbf{x}] + \mu_t(\mathbf{x}) \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right). \quad (42)$$

Consider the time derivative of KL divergence between μ_t and π ,

$$\frac{d}{dt} \text{KL}(\mu_t || \pi) = \frac{d}{dt} \int_{\mathbb{R}^d} \mu_t(\mathbf{x}) \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) d\mathbf{x} \quad (43)$$

$$= \int_{\mathbb{R}^d} \frac{\partial \mu_t(\mathbf{x})}{\partial t} \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) d\mathbf{x}_t + \int_{\mathbb{R}^d} \mu_t(\mathbf{x}) \frac{\partial}{\partial t} \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) d\mathbf{x} \quad (44)$$

$$= \int_{\mathbb{R}^d} \frac{\partial \mu_t(\mathbf{x})}{\partial t} \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) d\mathbf{x}_t + \int_{\mathbb{R}^d} \frac{\partial \mu_t(\mathbf{x})}{\partial t} d\mathbf{x} \quad (45)$$

$$= \int_{\mathbb{R}^d} \frac{\partial \mu_t(\mathbf{x})}{\partial t} \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) d\mathbf{x}_t. \quad (46)$$

The last term in the third equality vanishes since the μ_t is probability distribution and its L_1 norm is always 1. Then the KL divergence evolves as

$$\frac{d}{dt} \text{KL}(\mu_t || \pi) = \int_{\mathbb{R}^d} \nabla \cdot \left(\mu_t(\mathbf{x}) \mathbb{E}[\mathbf{g}_k - \nabla f(\mathbf{x})|\mathbf{x}_t = \mathbf{x}] + \mu_t(\mathbf{x}) \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right) \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) d\mathbf{x} \quad (47)$$

$$= - \int_{\mathbb{R}^d} \mu_t(\mathbf{x}) \left\langle \mathbb{E}[\mathbf{g}_k - \nabla f(\mathbf{x})|\mathbf{x}_t = \mathbf{x}] + \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right), \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right\rangle d\mathbf{x} \quad (48)$$

$$= - \int_{\mathbb{R}^d} \mu_t(\mathbf{x}) \left\| \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right\|^2 d\mathbf{x} + \mathbb{E} \left\langle \nabla f(\mathbf{x}_t) - \mathbf{g}_k, \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right\rangle. \quad (49)$$

The second term can be bounded as follows:

$$\mathbb{E} \left\langle \nabla f(\mathbf{x}_t) - \mathbf{g}_k, \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right\rangle \leq \mathbb{E} \left[\|\nabla f(\mathbf{x}_t) - \mathbf{g}_k\|^2 + \frac{1}{4} \left\| \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right\|^2 \right] \quad (50)$$

$$= \mathbb{E} \|\nabla f(\mathbf{x}_t) - \mathbf{g}_k\|^2 + \frac{1}{4} \text{FI}(\mu_t|\pi) \quad (51)$$

$$= \mathbb{E} \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 + \frac{1}{4} \text{FI}(\mu_t|\pi) \quad (52)$$

$$\leq 2\mathbb{E} \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_k)\|^2 + 2\mathbb{E}_{\mu_t(\mathbf{x}_t, \mathbf{x}_k)} \|\nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 \quad (53)$$

$$+ \frac{1}{4} \text{FI}(\mu_t|\pi). \quad (54)$$

The first inequality holds since $\langle a, b \rangle \leq a^2 + \frac{b^2}{4}$. The last line follows from Young's inequality. Furthermore, using Lipschitzness of gradients of f , we have

$$\mathbb{E} \|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_k)\|^2 \leq L^2 \mathbb{E} \|\mathbf{x}_t - \mathbf{x}_k\|^2 \quad (55)$$

$$\leq L^2 \mathbb{E} \|\mathbf{x}_t - \mathbf{x}_k + \sqrt{2t} \boldsymbol{\epsilon}_k\|^2 \quad (56)$$

$$= t^2 L^2 \mathbb{E}_{\mu_0} \|\mathbf{g}_k\|^2 + 2tdL^2. \quad (57)$$

Plugging back these into the time derivative of KL divergence, we have

$$\frac{d}{dt} \text{KL}(\mu_t|\pi) \leq -\frac{3}{4} \text{FI}(\mu_t|\pi) + 2t^2 L^2 \mathbb{E}_{\mu_0} \|\mathbf{g}_k\|^2 + 2\mathbb{E}_{\mu_0} \|\nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 + 4tdL^2 \quad (58)$$

$$\leq -\frac{3}{4} \text{FI}(\mu_t|\pi) + (4t^2 L^2 + 2) \mathbb{E}_{\mu_0} \|\nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 + 4t^2 L^2 \mathbb{E}_{\mu_0} \|\nabla f(\mathbf{x}_k)\|^2 + 4tdL^2. \quad (59)$$

The third term can be bounded as follows: We choose an optimal coupling $\mathbf{x}_k \sim \mu_0(\mathbf{x}_k)$ and $\mathbf{x}^* \sim \pi$ so that $\mathbb{E} \|\mathbf{x}_k - \mathbf{x}^*\| = \mathbf{W}_2(\mu_0, \pi)^2$, then using Young's inequality and smoothness of f ,

$$\mathbb{E}_{\mu_0} \|\nabla f(\mathbf{x}_k)\|^2 \leq 2\mathbb{E}_{\mu_0} \|\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}^*)\|^2 + 2\mathbb{E}_{\mu_0} \|\nabla f(\mathbf{x}^*)\|^2 \quad (60)$$

$$\leq 2L^2 \mathbb{E}_{\mu_0} \|\mathbf{x}_k - \mathbf{x}^*\|^2 + 2\mathbb{E}_{\mu_0} \|\nabla f(\mathbf{x}^*)\|^2 \quad (61)$$

$$\leq 2L^2 \mathbf{W}_2(\mu_0, \pi)^2 + 2dL \quad (62)$$

$$\leq \frac{4L^2}{\alpha} \text{KL}(\mu_0|\pi) + 2dL. \quad (63)$$

The last inequality follows from Talgrand's inequality. Hence for $t \leq \eta$ and $\eta \leq \frac{1}{2L}$, we have

$$\frac{d}{dt} \text{KL}(\mu_t|\pi) \leq -\frac{3}{4} \text{FI}(\mu_t|\pi) + (4t^2 L^2 + 2) \mathbb{E}_{\mu_0} \|\nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 + \frac{16t^2 L^4}{\alpha} \text{KL}(\mu_0|\pi) + 4tdL^2 + 8t^2 dL^3 \quad (64)$$

$$\leq -\frac{3\alpha}{2} \text{KL}(\mu_t|\pi) + (4t^2 L^2 + 2) \mathbb{E}_{\mu_0} \|\nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 + \frac{16t^2 L^4}{\alpha} \text{KL}(\mu_0|\pi) + 4tdL^2 + 8t^2 dL^3 \quad (65)$$

$$\leq -\frac{3\alpha}{2} \text{KL}(\mu_t|\pi) + 3\mathbb{E}_{\mu_0} \|\nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 + \frac{16\eta^2 L^4}{\alpha} \text{KL}(\mu_0|\pi) + 8\eta dL^2 \quad (66)$$

$$\leq -\frac{3\alpha}{2} \text{KL}(\mu_t|\pi) + 3\sigma_k^2 + \frac{16\eta^2 L^4}{\alpha} \text{KL}(\mu_0|\pi) + 8\eta dL^2. \quad (67)$$

The second inequality is due to Eq. (16). Equivalently, we can write,

$$\frac{d}{dt} (e^{3\alpha t/2} \text{KL}(\mu_t|\pi)) \leq e^{3\alpha t/2} \left(3\sigma_k^2 + \frac{16\eta^2 L^4}{\alpha} \text{KL}(\mu_0|\pi) + 8\eta dL^2 \right). \quad (68)$$

Integrating from $t = 0$ to $t = \eta$ gives,

$$e^{3\alpha\eta/2} \text{KL}(\mu_\eta|\pi) - \text{KL}(\mu_0|\pi) \leq 6\eta\sigma_k^2 + \frac{32\eta^3 L^4}{\alpha} \text{KL}(\mu_0|\pi) + 16\eta^2 dL^2 \quad (69)$$

1080 for $\eta \leq \frac{2}{3\alpha}$. Rearranging the terms,

$$1081 \text{KL}(\mu_\eta|\pi) \leq e^{-3\alpha\eta/2} \left[\left(1 + \frac{32\eta^3 L^4}{\alpha} \right) \text{KL}(\mu_0|\pi) + 6\eta\sigma_k^2 + 16\eta^2 dL^2 \right]. \quad (70)$$

1084 Renaming $\mu_0 = \mu_k$ and $\mu_\eta = \mu_{k+1}$, we obtain the result in the statement.

□

1088 The statement in Lemma C.1 is generic and can be applied to any LMC algorithm with stochastic
1089 gradients with bounded variance on the trajectory of the algorithm. Note that this is different from
1090 assuming that the variance is uniformly upper bounded. Instead, we set inner loop and variance
1091 reduction parameters so that the variance does not explode along the trajectory of the algorithm.

1092 C.1 PROOF OF QZ-LMC

1094 **Theorem 3.4** (Main Theorem for QZ-LMC). *Under Assumption 3.3, let μ_k be the distribution of \mathbf{x}_k
1095 in QZ-LMC algorithm. Then, if we set the step size $\eta = \mathcal{O}\left(\frac{\epsilon\alpha}{dL^2}\right)$, $K = \tilde{\mathcal{O}}\left(\frac{dL^2 \log(\text{KL}(\mu_0|\pi))}{\epsilon\alpha^2}\right)$, and
1096 $\hat{\sigma}^2 = \mathcal{O}(\alpha\epsilon)$, we have*

$$1097 \left\{ \text{KL}(\mu_K|\pi), \text{TV}(\mu_K, \pi)^2, \frac{\alpha}{2} \mathbf{W}_2(\mu_K, \pi)^2 \right\} \leq \epsilon.$$

1101 *In addition, under Assumptions 2.1 and 2.2, the query complexity to the stochastic evaluation oracle
1102 is $\tilde{\mathcal{O}}\left(\frac{d^2 L^2 \sigma}{\alpha^{5/2} \epsilon^{3/2}}\right)$, or under Assumptions 2.1 and 2.5 the query complexity to the stochastic evaluation
1103 oracle is $\tilde{\mathcal{O}}\left(\frac{d^{3/2} L^2 \sigma}{\alpha^{5/2} \epsilon^{3/2}}\right)$.*

1105 *Proof.* By Lemma C.1, one-step equation can be written as

$$1107 \text{KL}(\mu_{k+1}|\pi) \leq e^{-3\alpha\eta/2} \left[\left(1 + \frac{32\eta^3 L^4}{\alpha} \right) \text{KL}(\mu_k|\pi) + 6\eta\hat{\sigma}^2 + 16\eta^2 dL^2 \right] \quad (71)$$

$$1109 \leq e^{-\alpha\eta} \text{KL}(\mu_k|\pi) + 6\eta\hat{\sigma}^2 + 16\eta^2 dL^2. \quad (72)$$

1111 Since for $\eta \leq \frac{\alpha}{8L^2}$, $1 + \frac{32\eta^3 L^4}{\alpha} \leq 1 + \frac{\alpha\eta}{2} \leq e^{\alpha\eta/2}$. Unrolling the recursion, we have

$$1113 \text{KL}(\mu_k|\pi) \leq e^{-\alpha\eta k} \text{KL}(\mu_0|\pi) + \frac{6\eta\hat{\sigma}^2 + 16\eta^2 dL^2}{1 - e^{-\alpha\eta}} \quad (73)$$

$$1115 \leq e^{-\alpha\eta k} \text{KL}(\mu_0|\pi) + \frac{8\hat{\sigma}^2 + 32\eta dL^2}{\alpha} \quad (74)$$

$$1117 \leq e^{-\alpha\eta k} \text{KL}(\mu_0|\pi) + \frac{8\hat{\sigma}^2 + 32\eta dL^2}{\alpha}. \quad (75)$$

1120 The second inequality is due to the fact that for $\eta \leq \frac{\alpha}{8L^2}$, $1 - e^{-\alpha\eta} \geq \frac{3}{4}\alpha\eta$ when $\alpha\eta \leq \frac{1}{4}$. We
1121 set $\eta \leq \frac{\epsilon\alpha}{128dL^2}$ and $\hat{\sigma}^2 \leq \frac{\alpha\epsilon}{32}$ and $k \geq \frac{1}{\alpha\eta} \log\left(\frac{2\text{KL}(\mu_0|\pi)}{\epsilon}\right)$ so that $\text{KL}(\mu_k|\pi) \leq \epsilon$. The number
1122 of calls to the stochastic evaluation oracle under Assumptions 2.1 and 2.2 to achieve $\hat{\sigma}^2 \leq \frac{\alpha\epsilon}{32}$ at
1123 each iteration is $\mathcal{O}\left(\frac{d\sigma}{\alpha^{1/2}\epsilon^{1/2}}\right)$ by Theorem 2.4. Hence, the total number of calls to the stochastic
1124 evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^2 L^2 \sigma}{\alpha^{5/2} \epsilon^{3/2}}\right)$. Similarly, under Assumptions 2.1 and 2.5 the number of calls to
1125 stochastic evaluation at each iteration is $\mathcal{O}\left(\frac{d^{1/2}\sigma}{\alpha^{1/2}\epsilon^{1/2}}\right)$ by Theorem 2.7. Hence, the total number of
1126 calls to stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^{3/2} L^2 \sigma}{\alpha^{5/2} \epsilon^{3/2}}\right)$. □

1130 D PROOFS FOR GRADIENT ESTIMATION

1131 **Proposition 2.3.** *Let $X \in \mathbb{R}$ be a random variable such that $\mathbb{E}\|X - \mathbb{E}[X]\|^2 \leq \sigma^2$. Given two reals
1132 $t \geq 0$ and $\epsilon \in (0, 1)$, then there is a unitary operator $P_{t,\epsilon}^X : |0\rangle|0\rangle \mapsto |\phi_X\rangle|0\rangle$ acting on $\mathcal{H}_X \otimes \mathcal{H}_{aux}$*

1134 that can be implemented using $\tilde{O}(t\sigma \log(1/\epsilon))$ quantum experiments and binary oracle queries to
 1135 X such that

$$1136 \quad \|\phi_X - e^{it\mathbb{E}[X]}|0\rangle\| \leq \epsilon,$$

1137 with probability at least 8/9.

1138
 1139 *Proof.* The proof constructs a sequence of unitary operators using the binary-to-phase conversion
 1140 algorithm for different quantiles of X . We begin by randomly drawing a classical sample s from the
 1141 distribution that generates X . By Chebyshev's inequality,

$$1142 \quad \Pr[|s - \mathbb{E}[X]| \geq 3\sigma] \leq \frac{1}{9}. \quad (76)$$

1143 We consider the case $|s - \mathbb{E}[X]|$ is smaller than 3σ which holds with probability 8/9. Next, we define
 1144 the random variable $Y = X - s$. Additionally, we introduce a random variable $Y_{a,b}$, a truncated
 1145 version of Y , where values of Y outside the interval $[a, b]$ are set to zero. The expectation $\mathbb{E}[Y_{0,\infty}]$
 1146 can be expressed as a sum:

$$1147 \quad \mathbb{E}[Y_{0,\infty}] = \mathbb{E}[Y_{0,1}] + \sum_{k=1}^K 2^k \mathbb{E}\left[\frac{Y_{2^{k-1}, 2^k}}{2^k}\right] + \mathbb{E}[Y_{2^K,\infty}]. \quad (77)$$

1148 We define the unitary operator $P_{t,\epsilon}^{Y_{a,b}}$, which implements the phase oracle for $\mathbb{E}[Y_{a,b}]$ with an error
 1149 of at most ϵ . The unitary $P_{t,\epsilon/2}^{Y_{0,\infty}}$ can be implemented as the following product:

$$1150 \quad P_{t,\epsilon/2}^{Y_{0,\infty}} = P_{t,\epsilon/6}^{Y_{0,1}} \left(\prod_{k=1}^K P_{t,\epsilon/6K}^{Y_{2^{k-1}, 2^k}} \right) P_{t,\epsilon/6}^{Y_{2^K,\infty}}. \quad (78)$$

1151 When $K = \log\left(\frac{120\sigma^2 t}{\epsilon}\right)$, the operator $P_{t,\epsilon/6}^{Y_{2^K,\infty}}$ is effectively the identity operator, as:

$$1152 \quad \left| |0\rangle - e^{it\mathbb{E}[Y_{2^K,\infty}]}|0\rangle \right| \leq t\mathbb{E}[Y_{2^K,\infty}] \leq \frac{\epsilon}{6}. \quad (79)$$

1153 The last inequality holds because:

$$1154 \quad \mathbb{E}[Y_{2^K,\infty}] = \sum_{Y \geq 2^K} \Pr(Y)Y \leq \sum_Y \frac{1}{2^K} \Pr(Y)Y^2 = \frac{\mathbb{E}\|Y\|^2}{2^K} \quad (80)$$

$$1155 \quad \leq \frac{2\mathbb{E}\|X - \mathbb{E}[X]\|^2 + 2\|s - \mathbb{E}[X]\|^2}{2^K} \quad (81)$$

$$1156 \quad \leq \frac{20\sigma^2}{2^K} = \frac{\epsilon}{6t}, \quad (82)$$

1157 where the inequality in the second line follows from the definition of Y and Young's inequality.
 1158 Since $X_{0,1}$ is bounded between 0 and 1, we can implement $P_{t,\epsilon/6}^{Y_{0,1}}$ using $\tilde{O}(1)$ queries to X via the
 1159 binary-to-phase conversion algorithm (Lemma 2.12 in Cornelissen et al. (2022)). We need to show
 1160 how to implement $P_{t,\epsilon/6K}^{Y_{a,b}}$ when $b > 1$. We start by defining the unitary operator:

$$1161 \quad V_{a,b} : |0\rangle|0\rangle \mapsto \sum_Y \sqrt{\Pr(Y)} |Y_{a,b}/b\rangle|0\rangle \quad (83)$$

$$1162 \quad \mapsto \sum_Y \sqrt{\Pr(Y)} |Y_{a,b}/b\rangle \left(\sqrt{Y_{a,b}/b} |0\rangle + \sqrt{1 - Y_{a,b}/b} |1\rangle \right) \quad (84)$$

$$1163 \quad = \sqrt{\mathbb{E}[Y_{a,b}/b]} |\psi_0\rangle|0\rangle + \sqrt{1 - \mathbb{E}[Y_{a,b}/b]} |\psi_1\rangle|1\rangle, \quad (85)$$

1164 where the $|\psi_0\rangle$ and $|\psi_1\rangle$ are normalized quantum states. Noting that

$$1165 \quad \mathbb{E}[Y_{a,b}/b] \leq \frac{1}{b} \sum_{a \leq Y \leq b} \Pr(Y)Y \leq \frac{1}{ab} \sum_{a \leq Y \leq b} \Pr(Y)Y^2 \quad (86)$$

$$1166 \quad = \frac{1}{ab} \mathbb{E}\|Y\|^2 \leq \frac{\sigma^2}{ab}, \quad (87)$$

we can apply the linear amplitude amplification algorithm (see (Cornelissen et al., 2022, Proposition 2.10)) to implement the operator:

$$W_{a,b} : |0\rangle |0\rangle \mapsto \sqrt{p_{a,b}} |\psi_0\rangle |0\rangle + \sqrt{1-p_{a,b}} |\psi_1\rangle |1\rangle, \quad (88)$$

such that,

$$\left| \sqrt{p_{a,b}} - \sqrt{\frac{\mathbb{E}[Y_{a,b}/b]}{\sigma^2/(ab)}} \right| \leq \frac{\epsilon}{24Ktb} \quad (89)$$

using $\tilde{O}(\sqrt{ab}/\sigma)$ calls to $V_{a,b}$. Let $t' = t\sigma^2/a$. Using the binary-to-phase conversion algorithm, we then implement $|\phi_{a,b}\rangle = e^{it\mathbb{E}[Y_{a,b}]} |0\rangle$ with $\tilde{O}(t\sigma^2/a)$ calls to $W_{a,b}$ up to an operator norm error of at most $\frac{\epsilon}{12K}$. By using the triangular inequality,

$$\|W_{a,b} |0\rangle - e^{it\mathbb{E}[Y_{a,b}]} |0\rangle\| = \|e^{it'p_{a,b}} |0\rangle - e^{it\mathbb{E}[Y_{a,b}]} |0\rangle\| \quad (90)$$

$$\leq t' \left| p_{a,b} - \frac{\mathbb{E}[Y_{a,b}/b]}{\sigma^2/(ab)} \right| + \frac{\epsilon}{12K} \quad (91)$$

$$\leq 2t' \left| \sqrt{p_{a,b}} - \sqrt{\frac{\mathbb{E}[Y_{a,b}/b]}{\sigma^2/(ab)}} \right| + \frac{\epsilon}{12K} \quad (92)$$

$$\leq \frac{\epsilon}{6K}. \quad (93)$$

Thus, the total implementation of $P_{t,\epsilon/6K}^{Y_{a,b}}$ requires $\tilde{O}(t\sigma\sqrt{a/b})$ calls to $V_{a,b}$. This implies that each term in the product can be implemented using $\tilde{O}(t\sigma)$ quantum experiments and binary query oracles to Y . Finally, we apply the phase e^{its} to the resulting state to implement $P_{t,\epsilon/2}^{X_{0,\infty}}$. Similarly, we use the same method to implement $P_{t,\epsilon/2}^{X_{-\infty,0}}$, and take the product:

$$P_{t,\epsilon}^X = P_{t,\epsilon/2}^{X_{0,\infty}} P_{t,\epsilon/2}^{X_{-\infty,0}}. \quad (94)$$

This concludes the proof. \square

Lemma D.1. *Suppose we run Algorithm 4 with the phase oracle in Proposition 2.3 with evaluation accuracy $\epsilon' = \frac{\epsilon^2}{d^2\beta}$ to $f(\mathbf{x}, \xi)$. Let $\tilde{\mathbf{g}}$ denote the output. Then, under Assumptions 2.1 and 2.2,*

$$\|\tilde{\mathbf{g}} - \nabla f(\mathbf{x})\| \leq \epsilon,$$

with probability at least 5/9 using $\tilde{O}(\frac{\sigma d}{\epsilon})$ queries to $f(\mathbf{x}; \xi)$.

Proof. To be able to run the quantum gradient estimation algorithm, we need to implement O_F that maps

$$O_F |\mathbf{x}\rangle \mapsto e^{i\mathbb{E}_\xi[F(\mathbf{x}, \xi)]} |\mathbf{x}\rangle, \quad (95)$$

where $F(\mathbf{x}; \xi) = \frac{N}{2Ll} (f(\mathbf{x}_0 + \frac{l}{N}(\mathbf{x} - N/2); \xi) - f(\mathbf{x}_0; \xi))$. Let $\mathbf{y} = \frac{l}{N}(\mathbf{x} - N/2)$, the variance of $F(\mathbf{x}, \xi)$ is

$$\mathbb{E}\|F(\mathbf{x}; \xi) - \mathbb{E}[F(\mathbf{x}; \xi)]\|^2 = \mathbb{E}\left\| \int_0^1 \langle \nabla f(\mathbf{x} + t\mathbf{y}; \xi) - \nabla f(\mathbf{x} + t\mathbf{y}), \mathbf{y} \rangle dt \right\|^2 \quad (96)$$

$$\leq \|\mathbf{y}\|^2 \int_0^1 \mathbb{E}\|\nabla f(\mathbf{x} + t\mathbf{y}; \xi) - \nabla f(\mathbf{x} + t\mathbf{y})\|^2 dt \quad (97)$$

$$\leq \sigma^2 l^2 d. \quad (98)$$

Hence, implementing $e^{i\mathbb{E}[F(\mathbf{x}, \xi)]}$ takes $\tilde{O}(\sigma l d^{1/2} \frac{N}{Ll}) = \tilde{O}(\frac{\sigma}{\epsilon^{1/2} \beta^{1/2}}) = \tilde{O}(\frac{\sigma d}{\epsilon})$ queries to stochastic zeroth-order oracle and succeeds with probability 8/9. Since Algorithm 4 uses $\tilde{O}(1)$ queries to O_F by Lemma A.4 and succeeds with probability 2/3, the total query complexity is $\tilde{O}(\frac{\sigma d}{\epsilon})$ and success probability is at least 5/9 due to union bound. \square

Theorem 2.4. Suppose that the potential function f satisfies Assumptions 2.1 and 2.2 and further suppose that $\|\nabla f(\mathbf{x})\| \leq M^3$ for all \mathbf{x} . Then, given a real $\hat{\sigma} > 0$, there exists a quantum algorithm that outputs a random vector \mathbf{g} such that

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

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

Proof. Suppose that we run Algorithm 4 in Lemma D.1 T times with target accuracy $\frac{\hat{\sigma}}{2}$, then compute the median (coordinate-wise) of these outputs. If the result has norm smaller than M , we output this vector. Otherwise, we output all 0 vector. Let \mathbf{v} be the output of this algorithm. Since the algorithm in Lemma D.1 outputs a vector $\tilde{\mathbf{g}}$ such that $\|\tilde{\mathbf{g}} - \nabla f(\mathbf{x})\| \leq \frac{\hat{\sigma}}{2}$ with high probability, then by Chernoff bound and union bound over each dimension, at least $\frac{T}{2}$ of the outputs satisfy $\|\tilde{\mathbf{g}} - \nabla f(\mathbf{x})\| \leq \hat{\sigma}$ with probability at least $1 - 2\exp(-T^2/24)$. Since the norm of the gradient is M , when the condition fails, the error is $\|\tilde{\mathbf{g}} - \nabla f(\mathbf{x})\| \leq M$. Then in expectation,

$$\mathbb{E}\|\mathbf{v} - \nabla f(\mathbf{x})\|^2 \leq \frac{\hat{\sigma}^2}{4} + 2\exp(-T^2/24)M^2. \quad (99)$$

Setting $T^2 = 24 \log\left(\frac{8M^2}{3\hat{\sigma}^2}\right)$ gives $\mathbb{E}\|\mathbf{v} - \nabla f(\mathbf{x})\|^2 \leq \hat{\sigma}^2$. Hence, the overhead to Lemma D.1 to make the output smooth is at most logarithmic. Finally, we can use this algorithm as the biased stochastic gradient estimator in Algorithm 3 and obtain an unbiased estimator \mathbf{g} . \square

Lemma 2.6. Under Assumptions 2.1 and 2.5, Algorithm 1 returns a vector \mathbf{v} such that

$$\|\mathbf{v} - \nabla f(\mathbf{x})\| \leq \epsilon \quad (13)$$

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

Proof. As the algorithm essentially computes the expectation of $\mathbb{E}_\xi[\tilde{\mathbf{g}}(\mathbf{x}, \xi)]$, we need to prove that $\mathbb{E}_\xi[\tilde{\mathbf{g}}(\mathbf{x}, \xi)]$ is close to $\nabla f(\mathbf{x})$. We consider the case that Algorithm 4 returns $\epsilon/8$ accurate estimate whenever the function f behaves like β smooth inside the grid points. Furthermore, we consider the case $\|\mathbf{s} - \nabla f(\mathbf{x})\| \leq 2\sigma$. Both conditions are in fact achieved with high probability. Let $S \subseteq \Xi$ be a set such that the output of quantum gradient estimation (Algorithm 4) \mathbf{g} satisfies $\|\mathbf{g} - \nabla f(\mathbf{x}, \xi)\| \leq \frac{\epsilon}{8}$. Let $S' = \Xi - S$. We can consider the difference in L_2 norm separately for S and S' using triangular inequality.

$$\|\mathbb{E}_\xi \tilde{\mathbf{g}}(\mathbf{x}, \xi) - \nabla f(\mathbf{x})\| \leq \|\mathbb{E}_S(\tilde{\mathbf{g}}(\mathbf{x}, \xi) - \nabla f(\mathbf{x}; \xi))\| + \|\mathbb{E}_{S'}(\tilde{\mathbf{g}}(\mathbf{x}, \xi) - \nabla f(\mathbf{x}; \xi))\|. \quad (100)$$

We first analyze the first term. The contribution to the first term is either due to gradient estimation error $\frac{\epsilon}{8}$ or it is due to the fact that \mathbf{g} is replaced by \mathbf{s} because $\|\mathbf{g} - \mathbf{s}\| > D$. Suppose that $S_1 = \{\xi \in \Xi : \|\mathbf{g}(\mathbf{x}; \xi) - \mathbf{s}\| \leq D\}$ and $S_2 = S - S_1$. We can separate the error further for both cases using triangular inequality.

$$\|\mathbb{E}_S(\tilde{\mathbf{g}}(\mathbf{x}, \xi) - \nabla f(\mathbf{x}; \xi))\| \leq \mathbb{E}_{\xi \in S_1} \|(\mathbf{g}(\mathbf{x}, \xi) - \nabla f(\mathbf{x}, \xi))\| + \mathbb{E}_{\xi \in S_2} \|(\mathbf{s} - \nabla f(\mathbf{x}, \xi))\| \quad (101)$$

$$\leq \mathbb{E}_{\xi \in S_1} \|(\mathbf{g}(\mathbf{x}, \xi) - \nabla f(\mathbf{x}, \xi))\| + \mathbb{E}_{\xi \in S_2} \|(\mathbf{s} - \mathbf{g}(\mathbf{x}; \xi))\| \quad (102)$$

$$+ \mathbb{E}_{\xi \in S_2} \|(\mathbf{g}(\mathbf{x}; \xi) - \nabla f(\mathbf{x}, \xi))\| \quad (103)$$

$$\leq \frac{\epsilon}{8} + \frac{\mathbb{E}\|\mathbf{s} - \nabla f(\mathbf{x}, \xi)\|^2}{D} + \frac{\epsilon}{8}. \quad (104)$$

The first inequality is due to the fact that for any $\xi \in S_2$, Algorithm 1 replaces \mathbf{g} by \mathbf{s} . The last inequality follows from the fact that $\|\mathbf{g}(\mathbf{x}; \xi) - \nabla f(\mathbf{x}; \xi)\| \leq \frac{\epsilon}{8}$ for any $\xi \in S$ and $\mathbb{E}_{\xi \in S_2} \|\mathbf{s} - \mathbf{g}(\mathbf{x}; \xi)\| \leq \frac{\mathbb{E}\|\mathbf{s} - \mathbf{g}(\mathbf{x}; \xi)\|^2}{D}$ since for any $\xi \in S_2$ we have $\|\mathbf{g}(\mathbf{x}; \xi) - \mathbf{s}\| > D$. As $\|\mathbf{s} - \nabla f(\mathbf{x})\| \leq 2\sigma$,

$$\mathbb{E}\|\mathbf{s} - \mathbf{g}(\mathbf{x}; \xi)\|^2 \leq 2\mathbb{E}\|\mathbf{s} - \nabla f(\mathbf{x}; \xi)\|^2 + 2\mathbb{E}\|\nabla f(\mathbf{x}; \xi) - \mathbf{g}(\mathbf{x}; \xi)\|^2 \quad (105)$$

$$\leq 10\sigma^2. \quad (106)$$

³One can show that the norm of the gradient is bounded by a function of problem parameters 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 .

Then, for $D = \frac{40\sigma^2}{\epsilon}$, we have $\frac{\mathbb{E}\|\mathbf{s} - \mathbf{g}(\mathbf{x}; \xi)\|^2}{D} \leq \frac{\epsilon}{4}$. Therefore, $\|\mathbb{E}_{S'}(\tilde{\mathbf{g}}(\mathbf{x}, \xi) - \nabla f(\mathbf{x}; \xi))\| \leq \frac{\epsilon}{2}$.

The term due to S' comes from the case where gradient estimation fails. Notice that whenever gradient estimation fails, we have $\|\tilde{\mathbf{g}}(\mathbf{x}; \xi) - \nabla f(\mathbf{x})\| \leq \max(D, 2\sigma)$. Gradient estimation only fails when $f(\mathbf{x}; \xi)$ has smoothness constant larger than β . Using Markov's inequality this happens with probability at most $\frac{L}{\beta}$. Then,

$$\mathbb{E}_{S'}\|\tilde{\mathbf{g}}(\mathbf{x}, \xi) - \nabla f(\mathbf{x})\| \leq \frac{L}{\beta} \max(D, 2\sigma) \leq \frac{\epsilon}{4} \quad (107)$$

for $\beta = \frac{160L\sigma^2}{\epsilon^2}$ and $\sigma \geq \epsilon$. This implies that non-smooth branches do not affect the expectation by replacing \mathbf{g} with $\tilde{\mathbf{g}}$. Furthermore, the variance of $\tilde{\mathbf{g}}(\mathbf{x})$ is

$$\mathbb{E}_{\xi}\|\tilde{\mathbf{g}}(\mathbf{x}, \xi) - \mathbb{E}[\tilde{\mathbf{g}}(\mathbf{x}, \xi)]\|^2 \leq 2\mathbb{E}\|\tilde{\mathbf{g}}(\mathbf{x}, \xi) - \nabla f(\mathbf{x})\|^2 + 2\|\mathbb{E}[\tilde{\mathbf{g}}(\mathbf{x}, \xi)] - \nabla f(\mathbf{x})\|^2 \quad (108)$$

$$\leq 2\mathbb{E}_{S'}\|\tilde{\mathbf{g}}(\mathbf{x}, \xi) - \nabla f(\mathbf{x})\|^2 + 2\mathbb{E}_S\|\tilde{\mathbf{g}}(\mathbf{x}, \xi) - \nabla f(\mathbf{x})\|^2 + 2\epsilon^2 \quad (109)$$

$$\leq \frac{2Ld}{\beta} \max(D^2, 4\sigma^2) + 2\mathbb{E}\|\nabla f(\mathbf{x}; \xi) - \nabla f(\mathbf{x})\|^2 + 2\mathbb{E}\|\mathbf{s} - \nabla f(\mathbf{x})\|^2 + 3\epsilon^2 \quad (110)$$

$$= \mathcal{O}(\sigma^2). \quad (111)$$

Therefore we can use quantum mean estimation to output ϵ accurate vector \mathbf{v} such that $\|\mathbf{v} - \nabla f(\mathbf{x})\| \leq \epsilon$ using $\tilde{\mathcal{O}}(\sigma d^{1/2}/\epsilon)$ calls to algorithm \mathcal{A} . Since algorithm \mathcal{A} uses $\tilde{\mathcal{O}}(1)$ queries to evaluation oracle, total stochastic evaluation complexity is $\tilde{\mathcal{O}}(\sigma d^{1/2}/\epsilon)$. \square

Theorem 2.7 (Smooth Gradient). *Suppose that the potential function f satisfies Assumptions 2.1 and 2.5 and further suppose that $\|\nabla f(\mathbf{x})\| \leq M$ for all \mathbf{x} . Then, given a real $\hat{\sigma} > 0$, there exists a quantum algorithm that outputs a random vector \mathbf{g} such that*

$$\mathbb{E}[\mathbf{g}] = \nabla f(\mathbf{x}), \quad \text{and} \quad \mathbb{E}\|\mathbf{g} - \nabla f(\mathbf{x})\|^2 \leq \hat{\sigma}^2 \quad (14)$$

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

Proof. Suppose that we run Algorithm 1 T times with target accuracy $\frac{\hat{\sigma}}{2}$, then compute the median (coordinate-wise) of these outputs. If the result has norm smaller than M , we output this vector. Otherwise, we output all 0 vector. Let \mathbf{v} be the output of this algorithm. Since Algorithm 1 outputs a gradient \mathbf{v} such that $\|\mathbf{v} - \nabla f(\mathbf{x})\| \leq \hat{\sigma}/2$ with high probability (say $2/3$), then by Chernoff bound and union bound over each dimension, at least $\frac{T}{2}$ of the outputs satisfy $\|\mathbf{v} - \nabla f(\mathbf{x})\| \leq \hat{\sigma}$ with probability at least $1 - 2\exp(-T^2/24)$. Since the norm of the gradient is M , when the condition fails the error is $\|\mathbf{v} - \nabla f(\mathbf{x})\| \leq M$. Then in expectation,

$$\mathbb{E}\|\mathbf{v} - \nabla f(\mathbf{x})\|^2 \leq \frac{\hat{\sigma}^2}{4} + 2\exp(-T^2/24)M^2. \quad (112)$$

Setting $T^2 = 24 \log\left(\frac{8M^2}{3\hat{\sigma}^2}\right)$ gives $\mathbb{E}\|\mathbf{v} - \nabla f(\mathbf{x})\|^2 \leq \hat{\sigma}^2$. Hence, the overhead is at most logarithmic. Finally, we run Algorithm 3 to obtain an unbiased estimator \mathbf{g} . \square

E PROOFS FOR OPTIMIZATION

To be able to characterize the run-time of the algorithm, we first need to characterize the Log-Sobolev constant of f_v . To achieve this, we use the following LSI perturbation lemma by Holley-Stroock Holley & Stroock (1987).

Lemma E.1. *Let ρ be the Log-Sobolev constant of the Gibbs distribution with potential F . Then, the Log-Sobolev constant of f satisfies,*

$$\alpha \geq \rho e^{-|\sup_x(f(x) - F(x)) - \inf_x(f(x) - F(x))|}. \quad (113)$$

Next we present the proofs of Lemma 4.4 and Theorem 4.5.

Lemma 4.4. Let $\pi_v^\beta = \frac{e^{-\beta f_v(\mathbf{x})}}{\int e^{-\beta f_v(\mathbf{x})} d\mathbf{x}}$. If $\beta = \Theta(d/\epsilon)$ and $v \leq \mathcal{O}(\frac{\epsilon}{Md})$, then sampling from π_v^β returns ϵ approximate optimizer for f with probability at least 0.1.

Proof. Consider the following quantity that quantifies the optimization error with respect to original distribution π^β .

$$\varepsilon(\beta) = \mathbb{E}_{\pi^\beta}[f(\mathbf{x})] \quad (114)$$

which can be written explicitly,

$$\varepsilon(\beta) = \frac{\int_{\mathbb{R}^d} f(\mathbf{x}) \exp(-\beta f(\mathbf{x})) d\mathbf{x}}{\int_{\mathbb{R}^d} \exp(-\beta f(\mathbf{x})) d\mathbf{x}} \quad (115)$$

Consider the derivative of $\varepsilon(\beta)$ with respect to β .

$$\frac{d\varepsilon(\beta)}{d\beta} = \frac{\beta \left(\int_{\mathbb{R}^d} f(\mathbf{x}) \exp(-\beta f(\mathbf{x})) d\mathbf{x} \right)^2 - \beta \left(\int_{\mathbb{R}^d} f(\mathbf{x}) \exp(-\beta f^2(\mathbf{x})) d\mathbf{x} \right) \left(\int_{\mathbb{R}^d} \exp(-\beta f(\mathbf{x})) d\mathbf{x} \right)}{\left(\int_{\mathbb{R}^d} \exp(-\beta f(\mathbf{x})) d\mathbf{x} \right)^2} \quad (116)$$

$$= -\text{Var}(\beta^{1/2} f(\mathbf{x})) \quad (117)$$

Since variance is positive quantity, $\frac{d\varepsilon(\beta)}{d\beta}$ is negative and its absolute value is bounded by

$$\left| \frac{d\varepsilon(\beta)}{d\beta} \right| \leq \text{Var}(\beta^{1/2} f(\mathbf{x})) \leq M\beta \text{Var}(\mathbf{x}) \leq \mathcal{O}(d) \quad (118)$$

where the last inequality is due to the fact that tails of π^β is upper bounded by a Gaussian with variance $\Omega(1/\beta)$. Since $\varepsilon(\infty) = 0$, we can write

$$\varepsilon(\beta) = \lim_{\beta_u \rightarrow \infty} \int_{\beta_u}^{\beta} \frac{d\varepsilon(\beta)}{d\beta} d\beta \leq \mathcal{O}(d/\beta) \quad (119)$$

Let $f(x^*) = \min_{\mathbf{x}} f = 0$ without loss of generality. Then, for $\beta > \Theta(d/\epsilon)$, $\mathbb{E}_{\pi^\beta}[f(\mathbf{x}) - f(\mathbf{x}^*)] = \varepsilon(\beta) \leq \epsilon$. By Markov's inequality

$$\Pr[f(\mathbf{x}) - f(\mathbf{x}^*) \geq \epsilon] \leq 0.05 \quad (120)$$

For $\nu \leq \mathcal{O}(\frac{\epsilon}{Md})$,

$$\beta(f_\nu(\mathbf{x}) - f(\mathbf{x})) \leq \beta M\nu \leq \mathcal{O}(1) \quad (121)$$

Therefore, the total variation distance between π^β and π_v^β can be made smaller than 0.05 by scaling ν by a constant. Then,

$$\Pr[f(\mathbf{x}) - f(\mathbf{x}^*) \geq \epsilon] \leq 0.1 \quad (122)$$

This concludes the proof. \square

Theorem 4.5. Suppose that f satisfies Assumptions 4.1 and 4.2. Then, there exists a quantum algorithm that returns ϵ approximate minimizer for f with probability at least 0.1 using $\tilde{\mathcal{O}}\left(\frac{d^{9/2}}{\epsilon^{3/2}}\right)$ queries to the stochastic evaluation oracle for f .

Proof. We consider the potential function $\beta f_v(x)$ where β is the inverse temperature parameter. By Lemma 4.4, sampling from $\pi_v^\beta \propto e^{-\beta f_v}$ returns $\frac{\epsilon}{2}$ approximate minimizer for f with high probability (say 0.9) for sufficiently large $\beta = \mathcal{O}(\frac{d}{\epsilon})$. Suppose that we sample from a probability distribution μ such that

$$\text{TV}(\mu, \pi_v^\beta) \leq 0.1. \quad (123)$$

Then, the sample must be $\frac{\epsilon}{2}$ minimizer for f with probability at least 0.8. Therefore, it is sufficient to sample from π_v^β up to a constant TV distance.

We need to characterize the sampling complexity from π_v^β . From Bakry Emery theorem, Log Sobolev constant ρ of βF satisfies $\rho \geq \frac{\beta\mu}{2}$ where μ is the strong convexity constant of F . Let

1404 $M' = \max(M, 1)$ and take $v = \frac{\epsilon}{2M'd}$. Then using LSI perturbation result Lemma E.1 by Holley-
1405 Stroock, we have $\alpha \geq \frac{\beta\mu}{2}e^{-3\beta\epsilon/d} = \Omega(\frac{\mu d}{\epsilon})$. The uniform perturbation holds due to the inequality
1406 $|f_v - F| \leq |f_v - f| + |f - F| \leq vM + \frac{\epsilon}{d} \leq \frac{3\epsilon}{2d}$. Since βf_v is a smooth function with smoothness
1407 constant $L = \mathcal{O}(\frac{\beta M\sqrt{d}}{v}) = \mathcal{O}(\frac{d^{5/2}M^2}{\epsilon^2})$ by Proposition 4.3, the number of calls to stochastic evaluation
1408 oracle to sample from π_v is $\tilde{\mathcal{O}}(\frac{L^2 d^2}{\alpha^{5/2}}) = \tilde{\mathcal{O}}(\frac{M^4 d^{9/2}}{\mu^{5/2} \epsilon^{3/2}})$ by Theorem 3.4. Hence, we can optimize
1409 f in polynomial time. \square

1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457