Quantum Algorithms for Sampling Log-Concave Distributions and Estimating Normalizing Constants

Andrew M. Childs
Joint Center for Quantum Information and Computer Science,
Department of Computer Science, and
Institute for Advanced Computer Studies
University of Maryland
amchilds@umd.edu

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

Jin-Peng Liu
Simons Institute and
Department of Mathematics
UC Berkeley
jliu1219@terpmail.umd.edu

Chunhao Wang
Department of Computer Science and Engineering
Pennsylvania State University
cwang@psu.edu

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

Abstract

Given a convex function $f\colon \mathbb{R}^d \to \mathbb{R}$, the problem of sampling from a distribution $\propto e^{-f(x)}$ is called log-concave sampling. This task has wide applications in machine learning, physics, statistics, etc. In this work, we develop quantum algorithms for sampling log-concave distributions and for estimating their normalizing constants $\int_{\mathbb{R}^d} e^{-f(x)}\mathrm{d}x$. First, we use underdamped Langevin diffusion to develop quantum algorithms that match the query complexity (in terms of the condition number κ and dimension d) of analogous classical algorithms that use gradient (first-order) queries, even though the quantum algorithms use only evaluation (zeroth-order) queries. For estimating normalizing constants, these algorithms also achieve quadratic speedup in the multiplicative error ϵ. Second, we develop quantum Metropolis-adjusted Langevin algorithms with query complexity $\widetilde{O}(\kappa^{1/2}d)$ and $\widetilde{O}(\kappa^{1/2}d^{3/2}/\epsilon)$ for log-concave sampling and normalizing constant estimation, respectively, achieving polynomial speedups in κ, d, ϵ over the best known classical algorithms by exploiting quantum analogs of the Monte Carlo method and quantum walks. We also prove a $1/\epsilon^{1-o(1)}$ quantum lower bound for estimating normalizing constants, implying near-optimality of our quantum algorithms in ϵ.

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

1 Introduction

Sampling from a given distribution is a fundamental computational problem. For example, in statistics, samples can determine confidence intervals or explore posterior distributions. In machine learning, samples are used for regression and to train supervised learning models. In optimization, samples from well-chosen distributions can produce points near local or even global optima.

Sampling can be nontrivial even when the distribution is known. Indeed, efficient sampling is often a challenging computational problem, and bottlenecks the running time in many applications. Many efforts have been made to develop fast sampling methods. Among those, one of the most successful tools is Markov Chain Monte Carlo (MCMC), which uses a Markov chain that converges to the desired distribution to (approximately) sample from it.

Here we focus on the fundamental task of *log-concave sampling*, i.e., sampling from a distribution proportional to e^{-f} where $f\colon \mathbb{R}^d \to \mathbb{R}$ is a convex function. This covers many practical applications such as multivariate Gaussian distributions and exponential distributions. Provable performance guarantees for log-concave sampling have been widely studied [14]. A closely related problem is estimating the normalizing constants of log-concave distributions, which also has many applications [15].

Quantum computing has been applied to speed up many classical algorithms based on Markov processes, so it is natural to investigate quantum algorithms for log-concave sampling. If we can prepare a quantum state whose amplitudes are the square roots of the corresponding probabilities, then measurement yields a random sample from the desired distribution. In this approach, the number of required qubits is only poly-logarithmic in the size of the sample space. Unfortunately, such a quantum state probably cannot be efficiently prepared in general, since this would imply $\mathsf{SZK} \subseteq \mathsf{BQP}$ [1]. Nevertheless, in some cases, quantum algorithms can achieve polynomial speedup over classical algorithms. Examples include uniform sampling on a 2D lattice [35], estimating partition functions [4, 21, 31, 45, 46], and estimating volumes of convex bodies [6]. However, despite the importance of sampling log-concave distributions and estimating normalizing constants, we are not aware of any previous quantum speedups for general instances of these problems.

Formulation In this paper, we consider a d-dimensional convex function $f\colon \mathbb{R}^d \to \mathbb{R}$ which is L-smooth and μ-strongly convex, i.e., $\mu, L > 0$ and for any $x, y \in \mathbb{R}^d$, $x \neq y$,

$$\frac{f(y) - f(x) - \langle \nabla f(x), y - x\rangle}{\|x - y\|_2^2/2} \in [\mu, L]. \tag{1.1}$$

We denote by $\kappa := L/\mu$ the condition number of f. The corresponding log-concave distribution has probability density $\rho_f\colon \mathbb{R}^d \to \mathbb{R}$ with

$$\rho_f(x) := \frac{e^{-f(x)}}{Z_f}, \tag{1.2}$$

where the normalizing constant is

$$Z_f := \int_{x\in\mathbb{R}^d} e^{-f(x)}\,\mathrm{d}x. \tag{1.3}$$

When there is no ambiguity, we abbreviate ρ_f and Z_f as ρ and Z, respectively. Given an $\epsilon \in (0, 1)$,

- the goal of *log-concave sampling* is to output a random variable with distribution $\widetilde{\rho}$ such that $\|\widetilde{\rho} - \rho\| \leq \epsilon$, and
- the goal of *normalizing constant estimation* is to output a value $\widetilde{Z}$ such that with probability at least $2/3$, $(1-\epsilon)Z \leq \widetilde{Z} \leq (1+\epsilon)Z$.

Here $\|\cdot\|$ is a certain norm. We consider the general setting where the function f is specified by an oracle. In particular, we consider the quantum evaluation oracle O_f, a standard model in the quantum computing literature [3, 6, 7, 50]. The evaluation oracle acts as

$$O_f|x, y\rangle = |x, f(x) + y\rangle \quad \forall x \in \mathbb{R}^d, y \in \mathbb{R}. \tag{1.4}$$

(Quantum computing notations are briefly explained in Section 2.) We also consider the quantum gradient oracle $O_{\nabla f}$ with

$$O_{\nabla f}|x, z\rangle = |x, \nabla f(x) + z\rangle \quad \forall x, z \in \mathbb{R}^d. \tag{1.5}$$

In other words, we allow superpositions of queries to both function evaluations and gradients. The essence of quantum speedup is the ability to compute with carefully designed superpositions.

Contributions Our main results are quantum algorithms that speed up log-concave sampling and normalizing constant estimation.

Theorem 1.1 (Main log-concave sampling result)**.** *Let ρ denote the log-concave distribution (1.2). There exist quantum algorithms that output a random variable distributed according to $\widetilde{\rho}$ such that*

- *$W_2(\widetilde{\rho}, \rho) \leq \epsilon$ where W_2 is the Wasserstein 2-norm (2.4), using $\widetilde{O}(\kappa^{7/6} d^{1/6} \epsilon^{-1/3} + \kappa d^{1/3} \epsilon^{-2/3})$ queries to the quantum evaluation oracle (1.4); or*
- *$\|\widetilde{\rho} - \rho\|_{TV} \leq \epsilon$ where $\|\cdot\|_{TV}$ is the total variation distance (2.3), using $\widetilde{O}\big(\kappa^{1/2} d\big)$ queries to the quantum gradient oracle (1.5), or $\widetilde{O}\big(\kappa^{1/2} d^{1/4}\big)$ queries when the initial distribution is warm (formally defined in Appendix C.2.1).*

In the above results, the query complexity $\widetilde{O}(\kappa^{7/6} d^{1/6} \epsilon^{-1/3} + \kappa d^{1/3} \epsilon^{-2/3})$ is achieved by our quantum ULD-RMM algorithm. Although the quantum query complexity is the same as the best known classical result [37], we emphasize that our quantum algorithm uses a zeroth-order oracle while [37] uses a first-order oracle. The query complexity $\widetilde{O}\big(\kappa^{1/2} d\big)$ is achieved by our quantum MALA algorithm that uses a first-order oracle (as in classical algorithms). This is a quadratic speedup in κ compared with the best known classical algorithm [27]. With a warm start, our quantum speedup is even more significant: we achieve quadratic speedups in κ and d as compared with the best known classical algorithm with a warm start [47].

Theorem 1.2 (Main normalizing constant estimation result)**.** *There exist quantum algorithms that estimate the normalizing constant by $\widetilde{Z}$ within multiplicative error ϵ with probability at least $3/4$,*

- *using $\widetilde{O}(\kappa^{7/6} d^{7/6} \epsilon^{-1} + \kappa d^{4/3} \epsilon^{-1})$ queries to the quantum evaluation oracle (1.4); or*
- *using $\widetilde{O}(\kappa^{1/2} d^{3/2} \epsilon^{-1})$ queries to the quantum gradient oracle (1.5).*

Furthermore, this task has quantum query complexity at least $\Omega(\epsilon^{-1+o(1)})$ (Theorem 5.1).

Our query complexity $\widetilde{O}(\kappa^{7/6} d^{7/6} \epsilon^{-1} + \kappa d^{4/3} \epsilon^{-1})$ for normalizing constant estimation achieves a quadratic speedup in precision compared with the best known classical algorithm [15]. More remarkably, our quantum ULD-RMM algorithm again uses a zeroth-order oracle while the slower best known classical algorithm uses a first-order oracle [15]. Our quantum algorithm working with a first-order oracle achieves polynomial speedups in all parameters compared with the best known classical algorithm [15]. Moreover, the precision-dependence of our quantum algorithms is nearly optimal, which is quadratically better than the classical lower bound in $1/\epsilon$ [15].

To the best of our knowledge, these are the first quantum algorithms with quantum speedup for the fundamental problems of log-concave sampling and estimating normalizing constants. We explore multiple classical techniques including the underdamped Langevin diffusion (ULD) method [11–13, 43], the randomized midpoint method for underdamped Langevin diffusion (ULD-RMM) [36, 37], and the Metropolis adjusted Langevin algorithm (MALA) [8, 10, 14, 27, 28, 47], and achieve quantum speedups. Our main contributions are as follows.

- *Log-concave sampling.* For this problem, our quantum algorithms based on ULD and ULD-RMM have the same query complexity as the best known classical algorithms, but our quantum algorithms only use a zeroth-order (evaluation) oracle, while the classical algorithms use the first-order (gradient) oracle. For MALA, this improvement on the order of oracles is nontrivial, but we can use the quantum gradient oracle in our quantum MALA algorithm to achieve a quadratic speedup in the condition number κ. Furthermore, given a warm-start distribution, our quantum algorithm achieves a quadratic speedup in all parameters.
- *Normalizing constant estimation.* For this problem, our quantum algorithms provide larger speedups. In particular, our quantum algorithms based on ULD and ULD-RMM achieve quadratic

speedup in the multiplicative precision ϵ (while using a zeroth-order oracle) compared with the corresponding best-known classical algorithms (using a first-order oracle). Our quantum algorithm based on MALA achieves polynomial speedups in all parameters. Furthermore, we prove that our quantum algorithm is nearly optimal in terms of ϵ.

We summarize our results and compare them to previous classical algorithms in Table 1 and Table 2. See Appendix A for more detailed comparisons to related classical and quantum work.

Table 1: Summary of the query complexities of classical and quantum algorithms for sampling a d-dimensional log-concave distribution. Here $\kappa = L/\mu$ in (1.1) and ϵ is the error in the designated norm.

Method	Oracle	Complexity	Norm
ULD [9]	gradient	$\widetilde{O}\left(\kappa^2 d^{1/2}\epsilon^{-1}\right)$	W_2
ULD-RMM [37]	gradient	$\widetilde{O}\left(\kappa^{7/6}d^{1/6}\epsilon^{-1/3} + \kappa d^{1/3}\epsilon^{-2/3}\right)$	W_2
MALA [27]	gradient	$\widetilde{O}(\kappa d)$	TV
MALA with warm start [47]	gradient	$\widetilde{O}\left(\kappa d^{1/2}\right)$	TV
Quantum Inexact ULD (Theorem C.1)	evaluation	$\widetilde{O}\left(\kappa^2 d^{1/2}\epsilon^{-1}\right)$	W_2
Quantum Inexact ULD-RMM (Theorem C.2)	evaluation	$\widetilde{O}\left(\kappa^{7/6}d^{1/6}\epsilon^{-1/3} + \kappa d^{1/3}\epsilon^{-2/3}\right)$	W_2
Quantum MALA (Theorem C.7)	gradient	$\widetilde{O}\left(\kappa^{1/2} d\right)$	TV
Quantum MALA (warm start) (Theorem C.8)	gradient	$\widetilde{O}\left(\kappa^{1/2} d^{1/4}\right)$	TV

Table 2: Summary of the query complexities of classical and quantum algorithms for estimating the normalizing constant of a d-dimensional log-concave distribution. Here $\kappa = L/\mu$ in (1.1) and ϵ is the multiplicative error.

Method	Oracle	Complexity
Multilevel ULD [15]	gradient	$\widetilde{O}\left(\kappa^2 d^{3/2}\epsilon^{-2}\right)$
Multilevel ULD-RMM [15]	gradient	$\widetilde{O}\left(\kappa^{7/6}d^{7/6}\epsilon^{-2} + \kappa d^{4/3}\epsilon^{-2}\right)$
MALA [15]	gradient	$\widetilde{O}\left(\kappa d^2\epsilon^{-2}\max\{1, \frac{\kappa}{d}\}\right)$
Multilevel Quantum Inexact ULD (Theorem D.3)	evaluation	$\widetilde{O}\left(\kappa^2 d^{3/2}\epsilon^{-1}\right)$
Multilevel Quantum Inexact ULD-RMM (Theorem D.4)	evaluation	$\widetilde{O}\left(\kappa^{7/6}d^{7/6}\epsilon^{-1} + \kappa d^{4/3}\epsilon^{-1}\right)$
Quantum annealing with Quantum MALA (Theorem D.2)	gradient	$\widetilde{O}\left(\kappa^{1/2} d^{3/2}\epsilon^{-1}\right)$

Techniques In this work, we develop a systematic approach for studying the complexity of quantum walk mixing and show that for any reversible classical Markov chain, we can obtain quadratic speedup for the mixing time as long as the initial distribution is warm. In particular, we apply the quantum walk and quantum annealing in the context of Langevin dynamics and achieve polynomial quantum speedups.

The technical ingredients of our quantum algorithms are highlighted below.

- *Quantum simulated annealing (Lemma 3.2).* Our quantum algorithm for estimating normalizing constants combines the quantum simulated annealing framework of [45] and the quantum mean estimation algorithm of [31]. For each type of Langevin dynamics (which are random walks), we build a corresponding quantum walk. Crucially, the spectral gap of the random walk is quadratically amplified in the phase gap of the corresponding quantum walk. This allows us to use a Grover-like procedure to produce the stationary distribution state given a sufficiently good initial state. In the simulated annealing framework, this initial state is the stationary distribution state of the previous Markov chain.
- *Effective spectral gap (Lemma C.7).* We show how to leverage a "warm" initial distribution to achieve a quantum speedup for sampling. Classically, a warm start leads to faster mixing even if

the spectral gap is small. Quantumly, we generalize the notion of "effective spectral gap" [6, 26, 34] to our more general sampling problem. We show that with a bounded warmness parameter, quantum algorithms can achieve a quadratic speedup in the mixing time. By viewing the sampling problem as a simulated annealing process with only one Markov chain, we prove a quadratic speedup for quantum MALA by analyzing the effective spectral gap.

- *Quantum gradient estimation (Lemma C.1).* We adapt Jordan's quantum gradient algorithm [23] to the ULD and ULD-RMM algorithms and give rigorous proofs to bound the sampling error due to gradient estimation errors.

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

- Can we achieve quantum speedup in d and κ for unadjusted Langevin algorithms such as ULD and ULD-RMM? The main difficulty is that ULD and ULD-RMM are irreversible, while most available quantum walk techniques only apply to reversible Markov chains. New techniques might be required to resolve this question.
- Can we achieve further quantum speedup for estimating normalizing constants with a warm start distribution? This might require a more refined version of quantum mean estimation.
- Can we give quantum algorithms for estimating normalizing constants with query complexity sublinear in d? Such a result would give a provable quantum-classical separation due to the $\Omega(d^{1-o(1)}/\epsilon^{2-o(1)})$ classical lower bound proved in [15].

Limitations and societal impacts Researchers working on theoretical aspects of quantum computing or Monte Carlo methods may benefit from our results. In the long term, once fault-tolerant quantum computers have been built, our results may find practical applications in MCMC methods arising in the real world. As far as we are aware, our work does not have negative societal impacts.

2 Preliminaries

Basic definitions of quantum computation Quantum mechanics is formulated in terms of linear algebra. The *computational basis* of $\mathbb{C}^d$ is $\{\vec{e}_0, \ldots, \vec{e}_{d-1}\}$, where $\vec{e}_i = (0, \ldots, 1, \ldots, 0)^\top$ with the 1 in the $(i+1)^{\text{st}}$ position. We use *Dirac notation*, writing $|i\rangle$ (called a "ket") for $\vec{e}_i$ and $\langle i|$ (a "bra") for $\vec{e}_i^{\top}$.

The *tensor product* of quantum states is their Kronecker product: if $|u\rangle \in \mathbb{C}^{d_1}$ and $|v\rangle \in \mathbb{C}^{d_2}$, then we have $|u\rangle \otimes |v\rangle \in \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$ with

$$|u\rangle \otimes |v\rangle = (u_0 v_0, u_0 v_1, \ldots, u_{d_1-1} v_{d_2-1})^\top. \tag{2.1}$$

The basic element of quantum information is a *qubit*, a quantum state in $\mathbb{C}^2$, which can be written as $a|0\rangle + b|1\rangle$ for some $a, b \in \mathbb{C}$ with $|a|^2 + |b|^2 = 1$. An n-qubit tensor product state can be written as $|v_1\rangle \otimes \cdots \otimes |v_n\rangle \in \mathbb{C}^{2^n}$, where for any $i \in [n]$, $|v_i\rangle$ is a one-qubit state. Note however that most states in $\mathbb{C}^{2^n}$ are not product states. We sometimes abbreviate $|u\rangle \otimes |v\rangle$ as $|u\rangle|v\rangle$.

Operations on quantum states are *unitary transformations*. They are typically stated in the circuit model, where a *k-qubit gate* is a unitary matrix in $\mathbb{C}^{2^k}$. Two-qubit gates are *universal*, i.e., every n-qubit gate can be decomposed into a product of gates that act as the identity on $n-2$ qubits and as some two-qubit gate on the other 2 qubits. The *gate complexity* of an operation refers to the number of two-qubit gates used in a quantum circuit for realizing it.

Quantum access to a function, referred to as a *quantum oracle*, must be reversible and allow access to different values of the function in *superposition* (i.e., for linear combinations of computational basis states). For example, consider the unitary evaluation oracle O_f defined in (1.4). Given a probability distribution $\{p_i\}_{i=1}^n$ and a set of points $\{x_i\}_{i=1}^n$, we have

$$O_f \sum_{i=1}^n \sqrt{p_i}|x_i\rangle|0\rangle = \sum_{i=1}^n \sqrt{p_i}|x_i\rangle|f(x_i)\rangle. \tag{2.2}$$

Then a measurement would give $f(x_i)$ with probability p_i. However, a quantum oracle can not only simulate random sampling, but can enable uniquely quantum behavior through interference. Examples include amplitude amplification—the main idea behind Grover's search algorithm [19] and

the amplitude estimation procedure used in this paper—and many other quantum algorithms relying on coherent quantum access to a function. Similar arguments apply to the quantum gradient oracle (1.5). If a classical oracle can be computed by an explicit classical circuit, then the corresponding quantum oracle can be implemented by a quantum circuit of approximately the same size. Therefore, these quantum oracles provide a useful framework for understanding the quantum complexity of log-concave sampling and normalizing constant estimation.

To sample from a distribution π, it suffices to prepare the state $|\pi\rangle := \sum_x \sqrt{\pi_x}|x\rangle$ and then measure it. For a Markov chain specified by a transition matrix P with stationary distribution π, one can construct a corresponding *quantum walk* operator $W(P)$. Intuitively, quantum walks can be viewed as applying a sequence of quantum unitaries on a quantum state encoding the initial distribution to rotate it to the subspace of stationary distribution $|\pi\rangle$. The number of rotations needed (i.e., the angle between the initial distribution and stationary distribution) depends on the spectral gap of P, and a quantum algorithm can achieve a quadratic speedup via *quantum phase estimation* and *amplification* algorithms. More background on quantum walk is given in Appendix C.2.2.

Notations Throughout the paper, the big-O notations $O(\cdot)$, $o(\cdot)$, $\Omega(\cdot)$, and $\Theta(\cdot)$ follow common definitions. The $\tilde{O}$ notation omits poly-logarithmic terms, i.e., $\tilde{O}(f) := O(f\mathrm{poly}(\log f))$. We say a function f is *L-Lipschitz continuous* at x if $|f(x) - f(y)| \le L\|x - y\|$ for all y sufficiently near x. The total variation distance (TV-distance) between two functions $f, g\colon \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\|f - g\|_{\mathrm{TV}} := \int_{\mathbb{R}^d} |f(x) - g(x)| \,\mathrm{d}x. \tag{2.3}$$

Let $\mathcal{B}(\mathbb{R}^d)$ denote the Borel σ-field of $\mathbb{R}^d$. Given probability measures μ and ν on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, a *transference plan* ζ between μ and ν is defined as a probability measure on $(\mathbb{R}^d \times \mathbb{R}^d, \mathcal{B}(\mathbb{R}^d) \times \mathcal{B}(\mathbb{R}^d))$ such that for any $A \subseteq \mathbb{R}^d$, $\zeta(A \times \mathbb{R}^d) = \mu(A)$ and $\zeta(\mathbb{R}^d \times A) = \nu(A)$. We let $\Gamma(\mu, \nu)$ denote the set of all transference plans. We let

$$W_2(\mu, \nu) := \left(\inf_{\zeta \in \Gamma(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_2^2 \,\mathrm{d}\zeta(x, y) \right)^{\frac{1}{2}} \tag{2.4}$$

denote the Wasserstein 2-norm between μ and ν.

3 Quantum Algorithm for Log-Concave Sampling

In this section, we describe several quantum algorithms for sampling log-concave distributions.

Quantum inexact ULD and ULD-RMM We first show that the gradient oracle in the classical ULD and ULD-RMM algorithms can be efficiently simulated by the quantum evaluation oracle via quantum gradient estimation. Suppose we are given access to the evaluation oracle (1.4) for $f(x)$. Then by Jordan's algorithm [23] (see Lemma C.1 for details), there is a quantum algorithm that can compute $\nabla f(x)$ with a polynomially small ℓ_1-error by querying the evaluation oracle $O(1)$ times. Using this, we can prove the following theorem (see Appendix C.1 for details).

Theorem 3.1 (Informal version of Theorem C.1 and Theorem C.2)**.** *Let $\rho \propto e^{-f}$ be a d-dimensional log-concave distribution with f satisfying (1.1). Given a quantum evaluation oracle for f,*

- *the quantum inexact ULD algorithm uses $\widetilde{O}(\kappa^2 d^{1/2}\epsilon^{-1})$ queries, and*
- *the quantum inexact ULD-RMM algorithm uses $\widetilde{O}(\kappa^{7/6}d^{1/6}\epsilon^{-1/3} + \kappa d^{1/3}\epsilon^{-2/3})$ queries,*

to quantumly sample from a distribution that is ϵ-close to ρ in W_2-distance.

We note that the query complexities of our quantum algorithms using a *zeroth-order* oracle match the state-of-the-art classical ULD [9] and ULD-RMM [37] complexities with a *first-order* oracle. The main technical difficulty of applying the quantum gradient algorithm is that it produces a *stochastic gradient oracle* in which the output of the quantum algorithm $\mathbf{g}$ satisfies $\|\mathbb{E}[\mathbf{g}] - \nabla f(x)\|_1 \le d^{-\Omega(1)}$. In particular, the randomness of the gradient computation is "entangled" with the randomness of the Markov chain. We use the classical analysis of ULD and ULD-RMM processes [36] to prove that the stochastic gradient will not significantly slow down the mixing of ULD processes, and that the error caused by the quantum gradient algorithm can be controlled.

Quantum MALA We next propose two quantum algorithms with lower query complexity than classical MALA, one with a Gaussian initial distribution and another with a warm-start distribution. The main technical tool we use is a quantum walk in continuous space.

The classical MALA (i.e., Metropolized HMC) starts from a Gaussian distribution $\mathcal{N}(0, L^{-1}I_d)$ and performs a leapfrog step in each iteration. It is well-known that the initial Gaussian state

$$|\rho_0\rangle = \int_{\mathbb{R}^d} \left(\frac{L}{2\pi}\right)^{d/4} e^{-\frac{L}{4}\|z-x^\star\|_2^2} |z\rangle \, \mathrm{d}z \tag{3.1}$$

can be efficiently prepared. We show that the quantum walk update operator

$$U := \int_{\mathbb{R}^d} \mathrm{d}x \int_{\mathbb{R}^d} \mathrm{d}y \, \sqrt{p_{x\to y}} |x\rangle\langle x| \otimes |y\rangle\langle 0| \tag{3.2}$$

can be efficiently implemented, where $p_{x\to y} := p(x,y)$ is the transition density from x to y, and the density p satisfies $\int_{\mathbb{R}^d} p(x,y)\,\mathrm{d}y = 1$ for any $x \in \mathbb{R}^d$.

Lemma 3.1 (Informal version of Lemma C.6)**.** *The continuous-space quantum walk operator corresponding to the MALA Markov chain can be implemented with $O(1)$ gradient and evaluation queries.*

In general, it is difficult to quantumly speed up the mixing time of a classical Markov chain, which is upper bounded by $O(\delta^{-1}\log(\rho_{\min}^{-1}))$, where δ is the spectral gap. However, [45] shows that a quadratic speedup is possible when following a sequence of *slowly-varying* Markov chains. More specifically, let $\rho_0, \ldots, \rho_r$ be the stationary distributions of the *reversible* Markov chains $\mathcal{M}_0, \ldots, \mathcal{M}_r$ and let $|\rho_0\rangle, \ldots, |\rho_r\rangle$ be the corresponding quantum states. Suppose $|\langle\rho_i|\rho_{i+1}\rangle| \geq p$ for all $i \in \{0, \ldots, r-1\}$, and suppose the spectral gaps of $\mathcal{M}_0, \ldots, \mathcal{M}_r$ are lower-bounded by δ. Then we can prepare a quantum state $|\widetilde{\rho_r}\rangle$ that is ϵ-close to $|\rho_r\rangle$ using $\tilde{O}\big(\delta^{-1/2} r p^{-1}\big)$ quantum walk steps. To fulfill the slowly-varying condition, we consider an annealing process that goes from $\rho_0 = \mathcal{N}(0, L^{-1}I_d)$ to the target distribution $\rho_{M+1} = \rho$ in $M = \widetilde{O}(\sqrt{d})$ stages. At the ith stage, the stationary distribution is $\rho_i \propto e^{-f_i}$ with $f_i := f + \frac{1}{2}\sigma_i^{-2}\|x\|^2$. By properly choosing $\sigma_1 \leq \cdots \leq \sigma_M$, we prove that this sequence of Markov chains is slowly varying.

Lemma 3.2 (Informal version of Lemma B.6)**.** *If we take $\sigma_1^2 = \frac{\epsilon}{2dL}$ and $\sigma_{i+1}^2 = (1 + \frac{1}{\sqrt{d}})\sigma_i^2$, then for $0 \leq i \leq M$, we have $|\langle\rho_i|\rho_{i+1}\rangle| \geq \Omega(1)$.*

Combining Lemma 3.1, Lemma 3.2, and results on the mixing time of MALA [27], we have:

Theorem 3.2 (Informal version of Theorem C.7)**.** *Let $\rho \propto e^{-f}$ be a d-dimensional log-concave distribution with f satisfying (1.1). There is a quantum algorithm (Algorithm 1) that prepares a state $|\widetilde{\rho}\rangle$ with $\||\widetilde{\rho}\rangle - |\rho\rangle\| \leq \epsilon$ using $\widetilde{O}(\kappa^{1/2}d)$ gradient and evaluation oracle queries.*

Algorithm 1: QUANTUMMALAFORLOG-CONCAVESAMPLING (Informal)

Input: Evaluation oracle $\mathcal{O}_f$, gradient oracle $\mathcal{O}_{\nabla f}$, smoothness parameter L, convexity parameter μ

Output: Quantum state $|\widetilde{\rho}\rangle$ close to the stationary distribution state $\int_{\mathbb{R}^d} e^{-f(x)/2}\,\mathrm{d}|x\rangle$

Compute the cooling schedule parameters $\sigma_1, \ldots, \sigma_M$

Prepare the state $|\rho_0\rangle \propto \int_{\mathbb{R}^d} e^{-\frac{1}{4}\|x\|^2/\sigma_1^2}\,\mathrm{d}|x\rangle$

for $i \leftarrow 1, \ldots, M$ **do**

  Construct $\mathcal{O}_{f_i}$ and $\mathcal{O}_{\nabla f_i}$ where $f_i(x) = f(x) + \frac{1}{2}\|x\|^2/\sigma_i^2$

  Construct quantum walk update unitary U with $\mathcal{O}_{f_i}$ and $\mathcal{O}_{\nabla f_i}$

  Implement the quantum walk operator and the approximate reflection $\widetilde{R}_i$

  Prepare $|\rho_i\rangle$ by performing $\frac{\pi}{3}$-amplitude amplification with $\widetilde{R}_i$ on the state $|\rho_{i-1}\rangle|0\rangle$

return $|\rho_M\rangle$

For the classical MALA with a Gaussian initial distribution, it was shown by [28] that the mixing time is at least $\widetilde{\Omega}(\kappa d)$. Theorem 3.2 quadratically reduces the κ dependence.

Note that Algorithm 1 uses a first-order oracle, instead of the zeroth-order oracle used in the quantum ULD algorithms. The technical barrier to applying the quantum gradient algorithm (Lemma C.1) in the quantum MALA is to analyze the classical MALA with a stochastic gradient oracle. We currently do not know whether the "entangled randomness" dramatically increases the mixing time.

More technical details and proofs are provided in Appendix C.

4 Quantum Algorithm for Estimating Normalizing Constants

In this section, we apply our quantum log-concave sampling algorithms to the normalizing constant estimation problem. A very natural approach to this problem is via MCMC, which constructs a multi-stage annealing process and uses a sampler at each stage to solve a mean estimation problem. We show how to quantumly speed up these annealing processes and improve the query complexity of estimating normalizing constants.

Quantum speedup for the standard annealing process We first consider the standard annealing process for log-concave distributions, as already applied in the previous section. Recall that we pick parameters $\sigma_1 < \sigma_2 < \cdots < \sigma_M$ and construct a sequence of Markov chains with stationary distributions $\rho_i \propto e^{-f_i}$, where $f_i = f + \frac{1}{2\sigma_i^2}\|x\|^2$. Then, at the ith stage, we estimate the expectation

$$\mathbb{E}_{\rho_i}[g_i] \;\text{ where }\; g_i = \exp\left(\frac{1}{2}(\sigma_i^{-2} - \sigma_{i+1}^{-2})\|x\|^2\right). \tag{4.1}$$

If we can estimate each expectation with relative error at most $O(\epsilon/M)$, then the product of these M quantities estimates the normalizing constant $Z = \int_{\mathbb{R}^d} e^{-f(x)}\,\mathrm{d}x$ with relative error at most ϵ.

For the mean estimation problem, [31] showed that when the relative variance $\frac{\mathbf{Var}_{\rho_i}[g_i]}{\mathbb{E}_{\rho_i}[g_i]^2}$ is constant, there is a quantum algorithm for estimating the expectation $\mathbb{E}_{\rho_i}[g_i]$ within relative error at most ϵ using $\widetilde{O}(1/\epsilon)$ quantum samples from the distribution ρ_i. Our annealing schedule satisfies the bounded relative variance condition. Therefore, by the quantum mean estimation algorithm, we improve the sampling complexity of the standard annealing process from $\widetilde{O}(M^2\epsilon^{-2})$ to $\widetilde{O}(M\epsilon^{-1})$.

To further improve the query complexity, we consider using the quantum MALAs developed in the previous section to generate samples. Observe that Algorithm 1 outputs a quantum state corresponding to some distribution that is close to ρ_i, instead of an individual sample. If we can estimate the expectation without destroying the quantum state, then we can reuse the state and evolve it for the $(i+1)$st Markov chain. Fortunately, we can use non-destructive mean estimation to estimate the expectation and restore the initial states. A detailed error analysis of this algorithm can be found in [6, 21]. We first prepare $\widetilde{O}(M\epsilon^{-1})$ copies of initial states corresponding to the Gaussian distribution $\mathcal{N}(0, L^{-1}I_d)$. Then, for each stage, we apply the non-destructive mean estimation algorithm to estimate the expectation $\mathbb{E}_{\rho_i}[g_i]$ and then run quantum MALA to evolve the states $|\rho_i\rangle$ to $|\rho_{i+1}\rangle$. This gives our first quantum algorithm for estimating normalizing constants.

Theorem 4.1 (Informal version of Theorem D.2)**.** *Let Z be the normalizing constant in (1.3). There is a quantum algorithm (Algorithm 2) that outputs an estimate $\widetilde{Z}$ with relative error at most ϵ using $\widetilde{O}\big(d^{3/2}\kappa^{1/2}\epsilon^{-1}\big)$ queries to the quantum gradient and evaluation oracles.*

Quantum speedup for MLMC Now we consider using multilevel Monte Carlo (MLMC) as the annealing process and show how to achieve quantum speedup. MLMC was originally developed by [22] for parametric integration; then [16] applied MLMC to simulate stochastic differential equations (SDEs). The idea of MLMC is natural: we choose a different number of samples at each stage based on the cost and variance of that stage.

To estimate normalizing constants, a variant of MLMC was proposed in [15]. Unlike the standard MLMC for bounding the mean-squared error, they upper bound the bias and the variance separately, and the analysis is technically difficult. The first quantum algorithm based on MLMC was subsequently developed by [2] based on the quantum mean estimation algorithm. Roughly speaking, the quantum algorithm can quadratically reduce the ϵ-dependence of the sample complexity compared with classical MLMC.

Algorithm 2: QUANTUMMALAFORESTIMATINGNORMALIZINGCONSTANT (Informal)

Input: Evaluation oracle $\mathcal{O}_f$, gradient oracle $\mathcal{O}_{\nabla f}$

Output: Estimate $\widetilde{Z}$ of Z with relative error at most ϵ

$M \leftarrow \widetilde{O}(\sqrt{d})$, $K \leftarrow \widetilde{O}(\epsilon^{-1})$
Compute the cooling schedule parameters $\sigma_1, \ldots, \sigma_M$
for $j \leftarrow 1, \ldots, K$ **do**
  Prepare the state $|\rho_{1,j}\rangle \propto \int_{\mathbb{R}^d} e^{-\frac{1}{4}\|x\|^2/\sigma_1^2}|x\rangle \mathrm{d}x$
$\widetilde{Z} \leftarrow (2\pi\sigma_1^2)^{d/2}$
for $i \leftarrow 1, \ldots, M$ **do**
  $\widetilde{g}_i \leftarrow$ Non-destructive mean estimation for g_i using $\{|\rho_{i,0}\rangle, \ldots, |\rho_{i,K}\rangle\}$
  $\widetilde{Z} \leftarrow \widetilde{Z}\widetilde{g}_i$
  for $j \leftarrow 1, \ldots, K$ **do**
    $|\rho_{i+1,j}\rangle \leftarrow$ QUANTUMMALA$(\mathcal{O}_{f_{i+1}}, \mathcal{O}_{\nabla, f_{i+1}}, |\rho_{i,j}\rangle)$
return $\widetilde{Z}$

In this work, we apply the quantum accelerated MLMC (QA-MLMC) scheme [2] to simulate underdamped Langevin dynamics as the SDE. One challenge in using QA-MLMC is that g_i in our setting is not Lipschitz. Fortunately, as suggested by [15], this issue can be resolved by truncating large x and replacing g_i by $h_i := \min\{g_i, \exp(\frac{(r_i^+)^2}{\sigma_i^2(1+\alpha^{-1})})\}$, with the choice

$$\alpha = \widetilde{O}\left(\frac{1}{\sqrt{d}\log(1/\epsilon)}\right) \qquad r_i^+ = \mathbb{E}_{\rho_{i+1}}\|x\| + \Theta(\sigma_i\sqrt{(1+\alpha)\log(1/\epsilon)}) \tag{4.2}$$

to ensure $\frac{h_i}{\mathbb{E}_{\rho_i} g_i}$ is $O(\sigma_i^{-1})$ Lipschitz. Furthermore, $\left|\mathbb{E}_{\rho_i}(h_i - g_i)\right| < \epsilon$ by Lemmas C.7 and C.8 in [15]. For simplicity, we regard g_i as a Lipschitz continuous function in our main results. We present QA-MLMC in Algorithm 3, where the sampling algorithm A can be chosen to be quantum inexact ULD/ULD-RMM or quantum MALA.

Algorithm 3: QA-MLMC (Informal)

Input: Evaluation oracle $\mathcal{O}_f$, function g, error ϵ, a quantum sampler $\mathrm{A}(x_0, f, \eta)$ for ρ

Output: An estimate of $\widetilde{R} = \mathbb{E}_\rho h$

$K \leftarrow \widetilde{O}(\epsilon^{-1})$
Compute the initial point x_0 and the step size η_0
Compute the number of samples $N_1, \ldots, N_K$
for $j \leftarrow 1, \ldots, K$ **do**
  Let $\eta_j = \eta/2^{j-1}$
  for $i \leftarrow 1, \ldots, N_j$ **do**
    Sample $X_i^{\eta_j}$ by $\mathrm{A}(f, x_0, \eta_j)$, and sample $X_i^{\eta_j/2}$ by $\mathrm{A}(f, x_0, \eta_j/2)$
    $\widetilde{G}_i^- \leftarrow$ QMEANEST$(\{g(X_i^{\eta_j})\}_{i\in[N_j]})$, and $\widetilde{G}_i^+ \leftarrow$ QMEANEST$(\{g(X_i^{\eta_j/2})\}_{i\in[N_j]})$
return $\widetilde{R} = \widetilde{G}_0 + \sum_{j=0}^K(\widetilde{G}_i^- - \widetilde{G}_i^+)$

This QA-MLMC framework reduces the ϵ-dependence of the sampling complexity for estimating normalizing constants from ϵ^{-2} to ϵ^{-1} in both the ULD and ULD-RMM cases, as compared with the state-of-the-art classical results [15].

Using the quantum inexact ULD and ULD-RMM algorithms (Theorem 3.1) to generate samples, we obtain our second quantum algorithm for estimating normalizing constants (see Appendix D for proofs).

Theorem 4.2 (Informal version of Theorem D.3 and Theorem D.4)**.** *Let Z be the normalizing constant in (1.3). There exist quantum algorithms for estimating Z with relative error at most ϵ using*

- *quantum inexact ULD with $\widetilde{O}(d^{3/2}\kappa^2\epsilon^{-1})$ queries to the evaluation oracle, and*

- *quantum inexact ULD-RMM with* $\widetilde{O}((d^{7/6}\kappa^{7/6} + d^{4/3}\kappa)\epsilon^{-1})$ *queries to the evaluation oracle.*

5 Quantum Lower Bound

Finally, we lower bound the quantum query complexity of normalizing constant estimation.

Theorem 5.1. *For any fixed positive integer* k*, given query access (1.4) to a function* $f\colon \mathbb{R}^k \to \mathbb{R}$ *that is 1.5-smooth and 0.5-strongly convex, the quantum query complexity of estimating the partition function* $Z = \int_{\mathbb{R}^k} e^{-f(x)}\,\mathrm{d}x$ *within multiplicative error* ϵ *with probability at least* $2/3$ *is* $\Omega(\epsilon^{-\frac{1}{1+4/k}})$.

The proof of our quantum lower bound is inspired by the construction in Section 5 of [15]. They consider a log-concave function whose value is negligible outside a hypercube centered at 0. The interior of the hypercube is decomposed into cells of two types. The function takes different values on each type, and the normalizing constant estimation problem reduces to determining the number of cells of each type. Quantumly, we follow the same construction and reduce the cell counting problem to the *Hamming weight problem*: given an n-bit Boolean string and two integers $\ell < \ell'$, decide whether the Hamming weight (i.e., the number of ones) of this string is ℓ_1 or ℓ_2. This problem has a known quantum query lower bound [32], which implies the quantum hardness of estimating the normalizing constant. The full proof of Theorem 5.1 appears in Appendix E.

Acknowledgements

AMC acknowledges support from the Army Research Office (grant W911NF-20-1-0015); the Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing program; and the National Science Foundation (grant CCF-1813814). TL was supported by a startup fund from Peking University, and the Advanced Institute of Information Technology, Peking University. JPL was supported by the National Science Foundation (grant CCF-1813814), an NSF Quantum Information Science and Engineering Network (QISE-NET) triplet award (DMR-1747426), a Simons Foundation award (No. 825053), and the Simons Quantum Postdoctoral Fellowship. RZ was supported by the University Graduate Continuing Fellowship from UT Austin.

References

[1] Dorit Aharonov and Amnon Ta-Shma, *Adiabatic quantum state generation and statistical zero knowledge*, Proceedings of the 35th Annual ACM Symposium on Theory of Computing, pp. 20–29, 2003, arXiv:quant-ph/0301023.

[2] Dong An, Noah Linden, Jin-Peng Liu, Ashley Montanaro, Changpeng Shao, and Jiasu Wang, *Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance*, Quantum **5** (2021), 481, arXiv:2012.06283.

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

[4] Srinivasan Arunachalam, Vojtech Havlicek, Giacomo Nannicini, Kristan Temme, and Pawel Wocjan, *Simpler (classical) and faster (quantum) algorithms for Gibbs partition functions*, 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 112–122, IEEE, 2021, arXiv:2009.11270.

[5] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp, *Quantum amplitude amplification and estimation*, Contemporary Mathematics **305** (2002), 53–74, arXiv:quant-ph/0005055.

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

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

[8] Yuansi Chen, Raaz Dwivedi, Martin J. Wainwright, and Bin Yu, *Fast mixing of Metropolized Hamiltonian Monte Carlo: Benefits of multi-step gradients*, Journal of Machine Learning Research **21** (2020), 1–72, arXiv:1905.12247.

[9] Xiang Cheng, Niladri S. Chatterji, Peter L. Bartlett, and Michael I. Jordan, *Underdamped langevin mcmc: A non-asymptotic analysis*, Conference on learning theory, pp. 300–323, PMLR, 2018, arXiv:1707.03663.

[10] Sinho Chewi, Chen Lu, Kwangjun Ahn, Xiang Cheng, Thibaut Le Gouic, and Philippe Rigollet, *Optimal dimension dependence of the metropolis-adjusted langevin algorithm*, Conference on Learning Theory, pp. 1260–1300, PMLR, 2021, arXiv:2012.12810.

[11] Arnak S. Dalalyan and Avetik Karagulyan, *User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient*, Stochastic Processes and their Applications **129** (2019), no. 12, 5278–5311, arXiv:1710.00095.

[12] Alain Durmus, Szymon Majewski, and Błażej Miasojedow, *Analysis of Langevin Monte Carlo via convex optimization*, The Journal of Machine Learning Research **20** (2019), no. 1, 2666–2711, arXiv:1802.09188.

[13] Alain Durmus and Eric Moulines, *Nonasymptotic convergence analysis for the unadjusted Langevin algorithm*, The Annals of Applied Probability **27** (2017), no. 3, 1551–1587, arXiv:1507.05021.

[14] Raaz Dwivedi, Yuansi Chen, Martin J. Wainwright, and Bin Yu, *Log-concave sampling: Metropolis-Hastings algorithms are fast!*, Conference on learning theory, pp. 793–797, PMLR, 2018, arXiv:1801.02309.

[15] Rong Ge, Holden Lee, and Jianfeng Lu, *Estimating normalizing constants for log-concave distributions: Algorithms and lower bounds*, Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, pp. 579–586, 2020, arXiv:1911.03043.

[16] Michael B. Giles, *Multilevel Monte Carlo path simulation*, Operations Research **56** (2008), no. 3, 607–617.

[17] Michael B. Giles, *Multilevel Monte Carlo methods*, Acta Numerica **24** (2015), 259–328.

[18] Lov Grover and Terry Rudolph, *Creating superpositions that correspond to efficiently integrable probability distributions*, 2002, arXiv:quant-ph/0208112.

[19] Lov K. Grover, *A fast quantum mechanical algorithm for database search*, Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, pp. 212–219, ACM, 1996, arXiv:quant-ph/9605043.

[20] Yassine Hamoudi and Frédéric Magniez, *Quantum Chebyshev's inequality and applications*, Proceedings of the 46th International Colloquium on Automata, Languages, and Programming, Leibniz International Proceedings in Informatics, vol. 132, pp. 69:1–69:16, 2019, arXiv:1807.06456.

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

[22] Stefan Heinrich, *Multilevel Monte Carlo methods*, International Conference on Large-Scale Scientific Computing, pp. 58–67, Springer, 2001.

[23] Stephen P. Jordan, *Fast quantum algorithm for numerical gradient estimation*, Physical Review Letters **95** (2005), no. 5, 050501.

[24] Ravi Kannan, László Lovász, and Ravi Montenegro, *Blocking conductance and mixing in random walks*, Comb. Probab. Comput. **15** (2006), no. 4, 541–570.

[25] Phillip Kaye and Michele Mosca, *Quantum networks for generating arbitrary quantum states*, International Conference on Quantum Information, p. PB28, Optical Society of America, 2001, arXiv:quant-ph/0407102.

[26] Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy, *Quantum query complexity of state conversion*, 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, pp. 344–353, IEEE, 2011.

[27] Yin Tat Lee, Ruoqi Shen, and Kevin Tian, *Logsmooth gradient concentration and tighter runtimes for metropolized Hamiltonian Monte Carlo*, Conference on Learning Theory, pp. 2565–2597, PMLR, 2020, arXiv:2002.04121.

[28] Yin Tat Lee, Ruoqi Shen, and Kevin Tian, *Lower bounds on Metropolized sampling methods for well-conditioned distributions*, Advances in Neural Information Processing Systems, vol. 34, pp. 18812–18824, 2021, arXiv:2106.05480.

[29] David A. Levin and Yuval Peres, *Markov chains and mixing times*, vol. 107, American Mathematical Soc., 2017.

[30] Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang, *Quantum inference on Bayesian networks*, Physical Review A **89** (2014), no. 6, 062315, arXiv:1402.7359.

[31] Ashley Montanaro, *Quantum speedup of Monte Carlo methods*, Proceedings of the Royal Society A **471** (2015), no. 2181, 20150301, arXiv:1504.06987.

[32] Ashwin Nayak and Felix Wu, *The quantum query complexity of approximating the median and related statistics*, Proceedings of the 31st Annual ACM Symposium on Theory of Computing, pp. 384–393, 1999, arXiv:quant-ph/9804066.

[33] Maris Ozols, Martin Roetteler, and Jérémie Roland, *Quantum rejection sampling*, ACM Transactions on Computation Theory (TOCT) **5** (2013), no. 3, 1–33, arXiv:1103.2774.

[34] Ben W. Reichardt, *Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function*, 2009 50th Annual IEEE Symposium on Foundations of Computer Science, pp. 544–551, IEEE, 2009, arXiv:0904.2759.

[35] Peter C. Richter, *Quantum speedup of classical mixing processes*, Physical Review A **76** (2007), no. 4, 042306, arXiv:quant-ph/0609204.

[36] Abhishek Roy, Lingqing Shen, Krishnakumar Balasubramanian, and Saeed Ghadimi, *Stochastic zeroth-order discretizations of Langevin diffusions for Bayesian inference*, 2019, arXiv:1902.01373.

[37] Ruoqi Shen and Yin Tat Lee, *The randomized midpoint method for log-concave sampling*, Proceedings of the 33rd International Conference on Neural Information Processing Systems, pp. 2100–2111, 2019, arXiv:1909.05503.

[38] Rolando D. Somma, Sergio Boixo, and Howard Barnum, *Quantum simulated annealing*, 2007, arXiv:0712.1008.

[39] Rolando D. Somma, Sergio Boixo, Howard Barnum, and Emanuel Knill, *Quantum simulations of classical annealing processes*, Physical Review Letters **101** (2008), no. 13, 130504, arXiv:0804.1571.

[40] Daniel Štefankovič, Santosh Vempala, and Eric Vigoda, *Adaptive simulated annealing: A near-optimal connection between sampling and counting*, Journal of the ACM **56** (2009), no. 3, 1–36, arXiv:cs/0612058.

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

[42] Kristan Temme, Tobias J. Osborne, Karl G. Vollbrecht, David Poulin, and Frank Verstraete, *Quantum Metropolis sampling*, Nature **471** (2011), no. 7336, 87–90, arXiv:0911.3635.

[43] Santosh Vempala and Andre Wibisono, *Rapid convergence of the unadjusted Langevin algorithm: Isoperimetry suffices*, Advances in Neural Information Processing Systems **32** (2019), 8094–8106, arXiv:1903.08568.

[44] Nathan Wiebe and Christopher Granade, *Can small quantum systems learn?*, 2015, arXiv:1512.03145.

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

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

[47] Keru Wu, Scott Schmidler, and Yuansi Chen, *Minimax mixing time of the Metropolis-adjusted Langevin algorithm for log-concave sampling*, Journal of Machine Learning Research **23** (2022), no. 270, 1–63, arXiv:2109.13055.

[48] Man-Hong Yung and Alán Aspuru-Guzik, *A quantum–quantum Metropolis algorithm*, Proceedings of the National Academy of Sciences **109** (2012), no. 3, 754–759, arXiv:1011.1468.

[49] Christof Zalka, *Simulating quantum systems on a quantum computer*, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **454** (1998), no. 1969, 313–322, arXiv:quant-ph/9603026.

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

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Related Work

A.1 Classical MCMC methods

Our quantum algorithms are inspired by a major class of classical MCMC algorithms based on *Langevin dynamics*. There has been extensive work on non-asymptotic error bounds for the mixing times of Langevin-type algorithms for sampling [8, 14, 27, 37, 47]. One commonly used type of algorithm is based on the mixing time of Langevin dynamics, including the underdamped Langevin diffusion process described by the stochastic differential equations

$$\mathrm{d}v_t = -\gamma v_t \,\mathrm{d}t - u\nabla f(x_t)\,\mathrm{d}t + \sqrt{2\gamma u}\,\mathrm{d}W_t \tag{A.1}$$

$$\mathrm{d}x_t = v_t \,\mathrm{d}t \tag{A.2}$$

with parameters γ, u, where $W_t \sim \mathcal{N}(0,t)$ is a standard Wiener process. The coefficients of (A.1) are Lipschitz continuous since f is L-smooth; and the overdamped Langevin diffusion process

$$\mathrm{d}x_t = -u\nabla f(x_t)\,\mathrm{d}t + \sqrt{2u}\,\mathrm{d}W_t \tag{A.3}$$

is obtained by taking $\gamma \to \infty$ and $t \to t/\gamma$.

It can be shown that taking $\gamma = 2$ and $u = 1/L$, the stationary distribution of the underdamped Langevin diffusion (A.1) is proportional to $e^{-(f(x)+L\|v\|^2/2)}$, and the marginal distribution of x is proportional to $e^{-f(x)}$. When $\gamma \to \infty$, the stationary distribution of the overdamped version (A.3) is proportional to $e^{-f(x)}$. The numerical discretization of (A.3) is used in unadjusted Langevin algorithms, while sampling algorithms based on the discretization of (A.1) can have a better dependence on d and ϵ.

We now introduce a few common classical sampling algorithms: the underdamped Langevin diffusion (ULD) method; the randomized midpoint method for underdamped Langevin diffusion (ULD-RMM), with the best known dependence on d; and the Metropolis adjusted Langevin algorithm (MALA), with the best known dependence on κ and ϵ. To simulate the random process in discrete time, ULD can be viewed as the first-order forward Euler discretization of the continuous process (A.1). In particular, ULD takes $\widetilde{O}\big(\kappa^2\sqrt{d}/\epsilon\big)$ steps to approximate the stationary distribution $e^{-f(x)}$ within ϵ in the Wasserstein 2-norm [9], where κ is the condition number of f, and d is the dimension. ULD-RMM approximates the integral of the random process (A.1) by randomly choosing the midpoint in the integral, which reduces the bias in the accumulation of the integration. As a more accurate approximation, ULD-RMM converges in the Wasserstein 2-distance ϵ with $\widetilde{O}\Big(\frac{\kappa^{7/6}d^{1/6}}{\epsilon^{1/3}} + \frac{\kappa d^{1/3}}{\epsilon^{2/3}}\Big)$ steps [37], a polynomial reduction in κ, d, ϵ over ULD. As an alternative approach, MALA also constructs the Euler discretization of (A.1), and then applies the Metropolis-Hastings acceptance/rejection step to ensure convergence to the correct stationary distribution. It was first shown that MALA converges in the total variation distance ϵ with $\widetilde{O}\big(\kappa d \max\{1, \kappa/d\}\log(\kappa d/\epsilon)\big)$ steps for Gaussian initial distributions [8, 14]. Later, this result was improved to $\widetilde{O}\big(\kappa d \log(\kappa d/\epsilon)\big)$ based on an improved non-asymptotic analysis of the mixing time [27]. For warm-start distributions, the complexity of MALA can be further reduced to $\widetilde{O}\big(\kappa d^{1/2}\log(\kappa d/\epsilon)\big)$ [47]. Compared to ULD and ULD-RMM, this exponentially improves the dependence on ϵ, and polynomially improves the dependence on κ, while it suffers from a worse dependence on d. We introduce the algorithms and complexities of ULD and ULD-RMM in Appendix B.1, and introduce these results of MALA in Appendix C.2.1.

For the task of estimating the normalizing constant (1.3), the state-of-the-art classical results are given by [15]. That work applies the classical sampling algorithms described above with an annealing strategy. The normalizing constant is estimated by a sequence of telescoping sums, each of which can be approximated by a Monte Carlo method that samples from a log-concave distribution. We introduce this annealing procedure in Appendix B.2. Reference [15] employed the mixing time of MALA for Gaussian initial distributions developed by [8, 14] with the annealing procedure, achieving the overall complexity $\widetilde{O}\Big(\frac{\kappa d^2}{\epsilon^2}\max\{1, \kappa/d\}\Big)$ for estimating the normalizing constant. They also combined ULD and ULD-RMM with the annealing and the multilevel Monte Carlo (MLMC) method to achieve complexities of $\widetilde{O}\Big(\frac{\kappa^2 d^{3/2}}{\epsilon^2}\Big)$ and $\widetilde{O}\Big(\frac{\kappa^{7/6}d^{7/6}}{\epsilon^2} + \frac{\kappa d^{4/3}}{\epsilon^2}\Big)$, respectively. Here MLMC, introduced in Appendix D.2, is utilized to resolve the worse dependence on ϵ in

ULD and ULD-RMM, resulting in the same $\widetilde{O}(1/\epsilon^2)$ scaling of the error compared to the annealing with MALA. Annealing with MLMC and ULD/ULD-RMM also has a better dependence on d over annealing with MALA, while they suffer from a worse dependence on κ.

A.2 Quantum computing

Previous literature developed alternative approaches to generating quantum states corresponding to classical probability distributions on a quantum computer, sometimes referred to as quantum sampling (or qsampling) from a distribution. References [49], [18], and [25] propose direct state generation approaches using controlled rotations. However, this approach is limited to the regime in which the distribution is efficiently integrable. As an alternative, [1] develops an adiabatic approach to qsampling. They apply adiabatic evolution techniques to qsample the stationary distributions of a sequence of slowly varying Markov chains, a technique referred to as quantum simulated annealing (QSA) in subsequent literature [21, 38, 39, 45, 48]. The time complexity of Aharanov and Ta-Shma's approach is $O(1/\delta)$ as a function of the spectral gap δ, comparable to the running time of analogous classical sampling methods. Reference [45] adopted Szegedy's quantum walks [41] and amplitude amplification [5] to improve the time complexity of this qsampling procedure to $O(1/\sqrt{\delta})$, achieving a quadratic speedup in the spectral gap. As a generalization, [42] proposes a quantum Metropolis sampling method that extends qsampling to quantum Hamiltonians, with time complexity $O(1/\delta)$. Reference [48] combines quantum Metropolis sampling with QSA to achieve time complexity $O(1/\sqrt{\delta})$. Another alternative approach is quantum rejection sampling [30, 33, 44], which provides a method for transforming an initial superposition of desired and undesired states into the desired state using amplitude amplification. Reference [44] employs semi-classical Bayesian updating to achieve time complexity $O(1/\sqrt{\epsilon})$ as a function of the approximation error ϵ. The quantum rejection sampling approach is generally less efficient than the QSA approach, as the latter can achieve $O(\log 1/\epsilon)$ by choosing proper slowly varying MCs that mix rapidly.

Previous quantum computing literature on partition function estimation mainly focused on discrete systems with

$$Z(\beta) = \sum_{x\in\Omega} e^{-\beta H(x)}, \tag{A.4}$$

where β is an inverse temperature and H is a Hamiltonian function of x over a finite state space Ω. The space Ω is usually assumed to be a simple discrete set, such as $\{0,1\}^n$, and H is assumed to be a sum of local terms. For instance, [31] considers H taking integer values $\{0,1,\ldots,n\}$, and [21] assumes $0 \le H(x) \le n$ for all x.

To estimate $Z = Z(\infty)$ in (A.4), [31] considers a classical Chebyshev cooling schedule $0 = \beta_0 < \beta_1 < \ldots \beta_l = \infty$ for Z with the length $l = O(\sqrt{\log|\Omega|}\log\log|\Omega|)$ [40]. Reference [31] applies fast qsampling algorithms to estimate Z with $\widetilde{O}(l^2/\sqrt{\delta}\epsilon) = \widetilde{O}(\log|\Omega|/\sqrt{\delta}\epsilon)$ quantum walk steps to sample from Gibbs distributions $\pi_i(x) = \frac{1}{Z(\beta_i)}e^{-\beta_i H(x)}$, whereas a corresponding classical algorithm takes $\widetilde{O}(l^2/\delta\epsilon^2) = \widetilde{O}(\log|\Omega|/\delta\epsilon^2)$ random walk steps. Here ϵ denotes the relative error for estimating Z, and δ denotes the spectral gap of the Markov chains with stationary distributions $\pi_i(x)$. Reference [31] also points out that this quantum algorithm relies on classical Markov chain Monte Carlo for computing the Chebyshev cooling schedule, introducing an overhead of $\widetilde{O}(\log|\Omega|/\delta)$ [40]. Hence, the overall cost is $\widetilde{O}(\log|\Omega|/\sqrt{\delta}(\epsilon+\sqrt{\delta}))$, a quadratic reduction with respect to ϵ over classical algorithms. Reference [21] develops a fully quantized version of the Chebyshev cooling schedule that only requires additional cost $\widetilde{O}(\log|\Omega|/\sqrt{\delta})$. This results in overall cost $\widetilde{O}(\log|\Omega|/\sqrt{\delta}\epsilon)$, a quadratic speedup in terms of δ over [31] and classical algorithms. Reference [4] constructs a shorter Chebyshev cooling schedule by using a paired-product estimator with length $l = O(\sqrt{\log|\Omega|})$, eliminating the $l = O(\log\log|\Omega|)$ factors in the previous schedule [40]. Reference [4] develops a fully quantized version of this shorter schedule, almost matching the same overall cost $\widetilde{O}(\log|\Omega|/\sqrt{\delta}(\epsilon+\sqrt{\delta}))$ of [21].

Estimating the partition function of a discrete system corresponds to a discrete counting problem, with applications such as counting colorings or matchings of a graph and estimating statistics of Ising models, while estimating partition functions of continuous systems is relevant to the volume estimation problem.

B Tools from Classical MCMC Algorithms

B.1 ULD and ULD-RMM

We now describe underdamped Langevin diffusion (ULD) and the randomized midpoint method for underdamped Langevin diffusion (ULD-RMM), as introduced in [15] with Lipschitz continuous constants. We consider the underdamped Langevin diffusion with parameters γ, u:

$$\mathrm{d}v_t = -\gamma v_t \,\mathrm{d}t - \nabla f(x_t)\,\mathrm{d}t + \sqrt{2\gamma u}\,\mathrm{d}W_t, \tag{B.1}$$

$$\mathrm{d}x_t = v_t \,\mathrm{d}t, \tag{B.2}$$

The discrete dynamics of ULD can be described by

$$\mathrm{d}v_t^h = -\gamma v_t^h \,\mathrm{d}t - u\nabla f(x^h_{\lfloor t/h \rfloor h})\,\mathrm{d}t + \sqrt{2\gamma u}\,\mathrm{d}W_t, \tag{B.3}$$

$$\mathrm{d}x_t^h = v_t^h \,\mathrm{d}t. \tag{B.4}$$

According to [15], taking $\gamma = 2$ and $u = 1/L$, the explicit discrete-time update of ULD is integrated as

$$v^h_{t+h} = e^{-2h} v_t^h + \frac{1}{2L}(1 - e^{-2h})\nabla f(x_t^h) + \frac{2}{\sqrt{L}} W^h_{1,t}, \tag{B.5}$$

$$x^h_{t+h} = x_t^h + \frac{1}{2}(1 - e^{-2h})v_t^h + \frac{1}{2L}[h - (1 - e^{-2h})]\nabla f(x_t^h) + \frac{1}{\sqrt{L}} W^h_{2,t}, \tag{B.6}$$

where

$$W^h_{1,t} = \int_0^h e^{2(s-h)} \mathrm{d}B_{t+s}, \tag{B.7}$$

$$W^h_{2,t} = \int_0^h (1 - e^{2(s-h)}) \mathrm{d}B_{t+s}. \tag{B.8}$$

$W^h_{1,t}$ and $W^h_{2,t}$ can be obtained by sampling the d-dimensional standard Brownian motion B_t.

The ULD algorithm is presented in Algorithm 4. The convergence of ULD has been established by Theorem 1 of [9], which was restated by Theorem C.3 of [15] as follows.

Algorithm 4: Underdamped Langevin Dynamics (ULD)

Input: Function f, step size h, time T, and a sample x_0 from a starting distribution ρ_0
Output: Sequence $x_h^h, x_{2h}^h, \ldots, x^h_{\lfloor T \rfloor + 1}$

Compute $x_0^h \leftarrow x_0$
for $t = 0, h, \ldots, \lfloor T \rfloor$ **do**
  Draw $W^h_{1,t} = \int_0^h e^{2(s-h)} \mathrm{d}B_{t+s}$, $W^h_{2,t} = \int_0^h (1 - e^{2(s-h)}) \mathrm{d}B_{t+s}$
  Compute $v^h_{t+h} = e^{-2h} v_t^h + \frac{1}{2L}(1 - e^{-2h})\nabla f(x_t^h) + \frac{2}{\sqrt{L}} W^h_{1,t}$,
   $x^h_{t+h} = x_t^h + \frac{1}{2}(1 - e^{-2h})v_t^h + \frac{1}{2L}[h - (1 - e^{-2h})]\nabla f(x_t^h) + \frac{1}{\sqrt{L}} W^h_{2,t}$

Lemma B.1 (Theorem 1 of [9])**.** *Assume the target distribution ρ is strongly log-concave with L-smooth and μ-strongly convex negative log-density. Let ρ_n be the distribution of the underdamped Langevin diffusion with the initial point x_0 satisfying $\|x_0 - x^*\| \le D$, step size $h \le \frac{\epsilon}{104\kappa}\sqrt{\frac{1}{d/\mu + D^2}}$, and time $T \ge \frac{\kappa}{2}\log\Big(\frac{24\sqrt{d/\mu + D^2}}{\epsilon}\Big)$. Then ULD achieves*

$$\mathbb{E}\Big(\|\widehat{X}_n - X_T\|^2\Big) \le \widetilde{O}\Big(\frac{d^2\kappa^2 h^2}{\mu}\Big), \tag{B.9}$$

$$W_2(\rho_n, \rho) \le \epsilon, \tag{B.10}$$

using

$$\frac{T}{h} = \widetilde{\Theta}\Big(\frac{\kappa^2\sqrt{d}}{\epsilon}\Big) \tag{B.11}$$

queries to ∇f.

According to [15], the explicit discrete-time update of ULD-RMM is integrated as

$$v_{t+h}^h = e^{-2h} v_t^h + \frac{h}{L} e^{-2(1-\alpha)h} \nabla f(y_t^h) + \frac{2}{\sqrt{L}} W_{1,t}^h, \tag{B.12}$$

$$x_{t+h}^h = x_t^h + \frac{1}{2}(1 - e^{-2h}) v_t^h + \frac{h}{2L}(1 - e^{-2(1-\alpha)h}) \nabla f(y_t^h) + \frac{1}{\sqrt{L}} W_{2,t}^h, \tag{B.13}$$

$$y_{t+h}^h = x_t^h + \frac{1}{2}(1 - e^{-2\alpha h}) v_t^h + \frac{1}{2L}[\alpha h - (1 - e^{-2\alpha h})] \nabla f(x_t^h) + \frac{1}{\sqrt{L}} W_{3,t}^h, \tag{B.14}$$

where

$$W_{1,t}^h = \int_0^h e^{2(s-h)} \mathrm{d}B_{t+s}, \tag{B.15}$$

$$W_{2,t}^h = \int_0^h (1 - e^{2(s-h)}) \mathrm{d}B_{t+s}, \tag{B.16}$$

$$W_{3,t}^h = \int_0^{\alpha h} (1 - e^{2(s-h)}) \mathrm{d}B_{t+s}. \tag{B.17}$$

$W_{1,t}^h$, $W_{2,t}^h$, and $W_{3,t}^h$ can be obtained by sampling the d-dimensional standard Brownian motion B_t.

The ULD-RMM algorithm is presented in Algorithm 5. The convergence of ULD-RMM has been established by Theorem 3 of [37], which was restated by Theorem C.5 of [15] as follows.

Algorithm 5: Underdamped Langevin Dynamics with Randomized Midpoint Method (ULD-RMM)

Input: Function f, step size h, time T, and a sample x_0 from a starting distribution ρ_0
Output: Sequence $x_h^h, x_{2h}^h, \ldots, x_{\lfloor T \rfloor + 1}^h$

Compute $x_0^h \leftarrow x_0$, $y_0^h \leftarrow x_0$
for $t = 0, h, \ldots, \lfloor T \rfloor$ **do**
  Draw $W_{1,t}^h = \int_0^h e^{2(s-h)} \mathrm{d}B_{t+s}$, $W_{2,t}^h = \int_0^h (1 - e^{2(s-h)}) \mathrm{d}B_{t+s}$,
  $W_{3,t}^h = \int_0^{\alpha h} (1 - e^{2(s-h)}) \mathrm{d}B_{t+s}$
  Compute $v_{t+h}^h = e^{-2h} v_t^h + \frac{h}{L} e^{-2(1-\alpha)h} \nabla f(y_t^h) + \frac{2}{\sqrt{L}} W_{1,t}^h$,
  $x_{t+h}^h = x_t^h + \frac{1}{2}(1 - e^{-2h}) v_t^h + \frac{h}{2L}(1 - e^{-2(1-\alpha)h}) \nabla f(y_t^h) + \frac{1}{\sqrt{L}} W_{2,t}^h$,
  $y_{t+h}^h = x_t^h + \frac{1}{2}(1 - e^{-2\alpha h}) v_t^h + \frac{1}{2L}[\alpha h - (1 - e^{-2\alpha h})] \nabla f(x_t^h) + \frac{1}{\sqrt{L}} W_{3,t}^h$

Lemma B.2 (Theorem 3 of [37])**.** *Assume the target distribution ρ is strongly log-concave with L-smooth and μ-strongly convex negative log-density. Let ρ_n be the distribution of the randomized midpoint method for underdamped Langevin diffusion with the initial point x_0, step size $h \leq \min\Big\{ \frac{\epsilon^{1/3} \mu^{1/6}}{\kappa^{1/6} d^{1/6} \log^{1/6}\left(\frac{\sqrt{d/\mu}}{\epsilon}\right)}, \frac{\epsilon^{2/3} \mu^{1/3}}{d^{1/3} \log^{1/3}\left(\frac{\sqrt{d/\mu}}{\epsilon}\right)} \Big\}$, and time $T \geq 2\kappa \log\left(\frac{20d/\mu}{\epsilon^2}\right)$. Then ULD-RMM achieves*

$$\mathbb{E}\Big(\|\widehat{X}_n - X_T\|^2\Big) \leq \widetilde{O}\Big(\frac{d\kappa h^6}{\mu} + \frac{dh^3}{\mu}\Big), \tag{B.18}$$

$$W_2(\rho_n, \rho) \leq \epsilon, \tag{B.19}$$

using

$$\frac{2T}{h} = \widetilde{\Theta}\Big(\frac{\kappa^{7/6} d^{1/6}}{\epsilon^{1/3}} + \frac{\kappa d^{1/3}}{\epsilon^{2/3}}\Big) \tag{B.20}$$

queries to ∇f.

B.2 Annealing for Estimating the Normalizing Constant

Having described the sampling procedure for a log-concave function, we now move to the problem of estimating the normalizing constant

$$Z = \int_{x \in \mathbb{R}^d} e^{-f(x)} \mathrm{d}x. \tag{B.21}$$

We consider a sequence of auxiliary distributions, given by

$$f_i(x) = \frac{1}{2}\frac{\|x\|^2}{\sigma_i^2} + f(x) \tag{B.22}$$

for $i \in [M]$, where $\sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_M$. We define $\sigma_{M+1} = \infty$ and $f_{M+1} = f$ for convenience. We consider the sequence of distributions

$$\rho_i(\mathrm{d}x) = Z_i^{-1} e^{-f_i(x)} \mathrm{d}x, \tag{B.23}$$

where Z_i is the normalizing constant

$$Z_i = \int_{x \in \mathbb{R}^d} e^{-f_i(x)} \mathrm{d}x. \tag{B.24}$$

Then Z is estimated by the telescoping product

$$Z = Z_{M+1} = Z_1 \prod_{i=1}^{M} \frac{Z_{i+1}}{Z_i}. \tag{B.25}$$

In (B.25), we first approximate Z_1 by the normalizing constant of the Gaussian distribution with variance σ_1^2, which is bounded by the following lemma.

Lemma B.3 (Lemma 3.1 of [15])**.** *Letting $\sigma_1^2 = \frac{\epsilon}{2dL}$, we have*

$$\left(1 - \frac{\epsilon}{2}\right) \int_{x \in \mathbb{R}^d} e^{-\frac{1}{2}\frac{\|x\|^2}{\sigma_1^2}} \mathrm{d}x \leq Z_1 \leq \int_{x \in \mathbb{R}^d} e^{-\frac{1}{2}\frac{\|x\|^2}{\sigma_1^2}} \mathrm{d}x. \tag{B.26}$$

We then approximate $\frac{Z_{i+1}}{Z_i}$ by sampling the distribution ρ_i, with

$$\frac{Z_{i+1}}{Z_i} = \mathbb{E}_{\rho_i}(g_i), \tag{B.27}$$

where

$$g_i = \exp\Big(\frac{1}{2}\big(\frac{1}{\sigma_i^2} - \frac{1}{\sigma_{i+1}^2}\big)\|x\|^2\Big). \tag{B.28}$$

If $X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(K)}$ are i.i.d. samples generated according to the distribution ρ_i, then

$$\frac{Z_{i+1}}{Z_i} \approx \frac{1}{K}\sum_{k=1}^{K} g_i(X_i^{(k)}). \tag{B.29}$$

For the sequence of σ_i^2 with the annealing schedule $\frac{\sigma_{i+1}^2}{\sigma_i^2} = 1 + \alpha$, we aim to bound the relative variance of $\frac{Z_{i+1}}{Z_i}$. First, for $\frac{Z_{M+1}}{Z_M}$, we have the following lemma.

Lemma B.4 (Lemma 3.2 of [15])**.** *For $\sigma_M^2 \geq \frac{2}{\mu}$, we have*

$$\frac{\mathbb{E}_{\rho_M}(g_M^2)}{\mathbb{E}_{\rho_M}(g_M)^2} \leq \exp\left(\frac{4d}{\mu\sigma_M^4}\right). \tag{B.30}$$

When $\sigma_M^2 \geq \frac{2\sqrt{d}}{\mu}$, and assuming $\mu < 1$, we have $\frac{\mathbb{E}_{\rho_M}(g_M^2)}{\mathbb{E}_{\rho_M}(g_M)^2} \leq e$.

Second, for $\frac{Z_{i+1}}{Z_i}$ with $i \in [M-1]$, the relative variance of can be bounded by the following lemma.

Lemma B.5 ([Lemma 3.3 of [15])**.** *Let ρ be a log-concave distribution. For $\alpha \leq \frac{1}{2}$, we have*

$$\frac{\mathbb{E}_{\rho_i}(g_i^2)}{\mathbb{E}_{\rho_i}(g_i)^2} = \frac{\mathbb{E}_{\rho}\left[\exp\left(-\frac{1+\alpha}{2}\frac{\|x\|^2}{\sigma^2}\right)\right] \cdot \mathbb{E}_{\rho}\left[\exp\left(-\frac{1-\alpha}{2}\frac{\|x\|^2}{\sigma^2}\right)\right]}{\mathbb{E}_{\rho}\left[\exp\left(-\frac{1}{2}\frac{\|x\|^2}{\sigma^2}\right)\right]^2} \leq \exp\big(4\alpha^2 d\big). \tag{B.31}$$

Therefore, if we choose the annealing schedule

$$\frac{\sigma_{i+1}^2}{\sigma_i^2} = 1 + \frac{1}{\sqrt{d}}, \tag{B.32}$$

then $\frac{\mathbb{E}_{\rho_i}(g_i^2)}{\mathbb{E}_{\rho_i}(g_i)^2} \leq e^4$.

The estimate of the normalizing constant (1.3) relies on the above annealing framework and the sampling algorithms for the log-concave distribution ρ_i including ULD, ULD-RMM, and MALA. In the following sections, we discuss the quantum speedup for (1.3) using MALA and annealing, and using multilevel ULD/ULD-RMM and annealing.

B.3 Annealing Markov chains are slowly varying

The goal of this subsection is to prove the following lemma.

Lemma B.6 (Slowly varying MCs)**.** *Let $f_0(x) = \frac{\|x\|^2}{2\sigma_1^2}$ and let $\mathrm{d}\pi_0 = (2\pi\sigma_1^2)^{d/2} \cdot e^{-f_0(x)}\mathrm{d}x$ be the Gaussian distribution. For $i \in \{1, \ldots, M\}$, let $f_i(x) = f(x) + \frac{\|x\|^2}{2\sigma_i^2}$ and let $\mathrm{d}\pi_i = Z_i^{-1}e^{-f_i(x)}\mathrm{d}x$ be its stationary distribution. Let $f_{M+1}(x) = f(x)$ and let $\mathrm{d}\pi_{M+1}$ be the target log-concave distribution. Define the qsample state*

$$|\pi_i\rangle = \int_\Omega \mathrm{d}x \sqrt{\pi_i(x)}|x\rangle \quad \forall\, 0 \leq i \leq M+1. \tag{B.33}$$

Then, for $0 \leq i \leq M$, we have

$$|\langle \pi_i | \pi_{i+1}\rangle| \geq \Omega(1). \tag{B.34}$$

Proof. First, we consider the case when $i = 0$. Note that $|\langle \pi_0|\pi_1\rangle|$ can be written as

$$|\langle \pi_0|\pi_1\rangle| = \int_\Omega \mathrm{d}x \cdot (2\pi\sigma_1^2)^{-d/4} e^{-\frac{1}{2}f_0(x)} \cdot Z_1^{-1/2} e^{-\frac{1}{2}f_1(x)} \tag{B.35}$$

$$= \frac{\int_\Omega e^{-\frac{1}{2}f(x) - \frac{\|x\|^2}{2\sigma_1^2}}\mathrm{d}x}{(2\pi\sigma_1^2)^{d/4} \cdot \sqrt{Z_1}}. \tag{B.36}$$

Since $0 \leq f(x) \leq \frac{1}{2}L\|x\|^2$, the numerator can be lower bounded by

$$\int_\Omega e^{-\frac{1}{2}f(x) - \frac{\|x\|^2}{2\sigma_1^2}}\mathrm{d}x \geq \int_\Omega e^{-\frac{1}{2}(L/2 + \sigma_1^{-2})\|x\|^2}\mathrm{d}x = \left(2\pi(L/2 + \sigma_1^{-2})^{-1}\right)^{d/2} \tag{B.37}$$

and the denominator can be upper bounded by

$$(2\pi\sigma_1^2)^{d/4} \cdot \sqrt{\int_\Omega e^{-f(x) - \frac{1}{2}\|x\|^2/\sigma_1^2}\mathrm{d}x} \leq (2\pi\sigma_1^2)^{d/4} \cdot \sqrt{\int_\Omega e^{-\frac{1}{2}\|x\|^2/\sigma_1^2}\mathrm{d}x} = (2\pi\sigma_1^2)^{d/2}. \tag{B.38}$$

Therefore

$$|\langle \pi_0|\pi_1\rangle| \geq \frac{\left(2\pi(L/2 + \sigma_1^{-2})^{-1}\right)^{d/2}}{(2\pi\sigma_1^2)^{d/2}} = (1 + \sigma_1^2 L/2)^{-d/2} \geq e^{-\sigma_1^2 dL/4}. \tag{B.39}$$

By our choice of $\sigma_1^2 = \frac{\epsilon}{2dL}$, we have $|\langle \pi_0|\pi_1\rangle| \geq e^{-\epsilon/8} = \Omega(1)$.

Now consider the case where $1 \leq i \leq M-1$. The inner product between $|\pi_i\rangle$ and $|\pi_{i+1}\rangle$ can be written as

$$|\langle \pi_i|\pi_{i+1}\rangle| = \int_\Omega \mathrm{d}x \cdot Z_i^{-1/2} e^{-\frac{1}{2}f_i(x)} \cdot Z_{i+1}^{-1/2} e^{-\frac{1}{2}f_{i+1}(x)} \tag{B.40}$$

$$= \frac{\int_\Omega e^{-f(x) - \frac{\|x\|^2}{4}(\sigma_i^{-2} + \sigma_{i+1}^{-2})}\mathrm{d}x}{\sqrt{Z_i Z_{i+1}}}. \tag{B.41}$$

Let $\sigma^2 = \sigma_{i+1}^2$ and $\sigma^2/(1+\alpha) = \sigma_i^2$. Also, let ρ be the log-concave distribution $\rho(\mathrm{d}x) = Z^{-1}e^{-f(x)}\mathrm{d}x$. Then we have

$$\int_\Omega e^{-f(x)-\frac{\|x\|^2}{4}(\sigma_i^{-2}+\sigma_{i+1}^{-2})}\mathrm{d}x = Z \cdot \mathbb{E}_\rho\left[e^{-\frac{1+\alpha/2}{2\sigma^2}\|x\|^2}\right]. \tag{B.42}$$

Similarly,

$$Z_i = Z \cdot \mathbb{E}_\rho\left[e^{-\frac{1+\alpha}{2\sigma^2}\|x\|^2}\right] \quad \text{and} \quad Z_{i+1} = Z \cdot \mathbb{E}_\rho\left[e^{-\frac{1}{2\sigma^2}\|x\|^2}\right]. \tag{B.43}$$

Hence,

$$|\langle \pi_i | \pi_{i+1} \rangle| = \frac{\mathbb{E}_\rho\left[e^{-\frac{1+\alpha/2}{2\sigma^2}\|x\|^2}\right]}{\mathbb{E}_\rho\left[e^{-\frac{1+\alpha}{2\sigma^2}\|x\|^2}\right]^{1/2} \cdot \mathbb{E}_\rho\left[e^{-\frac{1}{2\sigma^2}\|x\|^2}\right]^{1/2}}. \tag{B.44}$$

Let $\alpha' := \frac{\alpha}{\alpha+2}$ and $\sigma'^2 := \frac{\sigma^2}{1+\alpha/2}$. Then

$$|\langle \pi_i | \pi_{i+1} \rangle| = \frac{\mathbb{E}_\rho\left[e^{-\frac{1}{2\sigma'^2}\|x\|^2}\right]}{\mathbb{E}_\rho\left[e^{-\frac{1+\alpha'}{2\sigma'^2}\|x\|^2}\right]^{1/2} \cdot \mathbb{E}_\rho\left[e^{-\frac{1-\alpha'}{2\sigma'^2}\|x\|^2}\right]^{1/2}} \tag{B.45}$$

$$\geq e^{-2\alpha'^2 d}, \tag{B.46}$$

where the last step follows from Lemma B.5. Since we choose $\alpha = d^{-1/2}$, we have $\alpha' = \frac{1}{1+2\sqrt{d}} = O(d^{-1/2})$, which implies that $e^{-2\alpha^2 d} = \Omega(1)$.

Finally, we consider the case where $i = M$. The inner product can be written as

$$|\langle \pi_M | \pi_{M+1} \rangle| = \frac{\int_\Omega \mathrm{d}x \cdot e^{-\frac{1}{2}f(x)-\frac{1}{4}\|x\|^2/\sigma_M^2} \cdot e^{-\frac{1}{2}f(x)}}{\sqrt{Z_M}\sqrt{Z}} \tag{B.47}$$

$$= \frac{\int_\Omega \mathrm{d}x \cdot e^{-f(x)-\frac{1}{4}\|x\|^2/\sigma_M^2}}{\sqrt{Z_M}\sqrt{Z}}. \tag{B.48}$$

Let ρ' be a log-concave distribution with density proportional to $e^{-f(x)-\frac{1}{4}\|x\|^2/\sigma_M^2}$. Then

$$\frac{\int_\Omega \mathrm{d}x \cdot e^{-f(x)-\frac{1}{4}\|x\|^2/\sigma_M^2}}{\sqrt{Z_M}\sqrt{Z}} = \mathbb{E}_{\rho'}\left[e^{-\frac{1}{4}\|x\|^2/\sigma_M^2}\right]^{-1/2} \cdot \mathbb{E}_{\rho'}\left[e^{\frac{1}{4}\|x\|^2/\sigma_M^2}\right]^{-1/2} \tag{B.49}$$

$$\geq e^{-\frac{d}{2\mu\sigma_M^4}}, \tag{B.50}$$

where the second step follows from the proof of Lemma B.4 in [15]. Since we take $\sigma_M^2 = \Theta(\frac{\sqrt{d}}{\mu})$, we find that $|\langle \pi_M | \pi_{M+1} \rangle| \geq e^{-\Theta(1)} = \Omega(1)$.

Combining the three cases, the proof is complete. □

C Quantum Algorithm for Log-Concave Sampling: Details

In this section, we provide several quantum algorithms for sampling log-concave distributions. In Appendix C.1, we show that the classical underdamped Langevin diffusion (ULD) and the randomized midpoint method for underdamped Langevin diffusion (ULD-RMM) can be improved by replacing the first-order oracle by the zeroth-order quantum oracle, while achieving the same efficiency and accuracy guarantees. In Appendix C.2, we show that the Metropolis adjusted Langevin algorithm (MALA) can be quantumly sped up in terms of query complexity, for both Gaussian initial distributions and warm-start distributions.

C.1 Quantum Inexact ULD and ULD-RMM

In the quantum setting, we can estimate $\nabla f(x)$ by using Jordan's algorithm with queries to the quantum zeroth-order evaluation oracle (1.4). The following lemma provides an ℓ_1-error guarantee.

Lemma C.1 (Lemma 2.3 in [7])**.** *Let f be a convex, L_0-Lipschitz continuous function that is specified by an evaluation oracle with error at most ϵ. Suppose f is L-smooth in $B_\infty(x, 2\sqrt{\epsilon/L})$. Let*

$$\widetilde{g} = \textsf{SmoothQuantumGradient}(f, \epsilon, L_0, L, x). \tag{C.1}$$

Then for any $i \in [d]$, we have $|\widetilde{g}_i| \le L_0$ and $\mathbb{E}|\widetilde{g}_i - \nabla f(x)_i| \le 3000\sqrt{d\epsilon L}$; hence

$$\mathbb{E}\|\widetilde{g} - \nabla f(x)\|_1 \le 3000 d^{1.5}\sqrt{\epsilon L}. \tag{C.2}$$

If L_0, $1/L$, and $1/\epsilon$ are $\mathrm{poly}(d)$, the SmoothQuantumGradient *algorithm uses $O(1)$ queries to the quantum evaluation oracle and $\widetilde{O}(d)$ gates.*

We then introduce inexact ULD and ULD-RMM by using a stochastic zeroth-order oracle as follows.

Lemma C.2 (Theorem 2.2 of [36])**.** *Let ρ_n be the distribution of the underdamped Langevin diffusion with the initial point x_0 satisfying $\|x_0 - x^*\| \le D$, step size $h \le \frac{\epsilon}{104\kappa}\sqrt{\frac{1}{d/\mu + D^2}}$, and time $T \ge \frac{\kappa}{2}\log\Big(\frac{24\sqrt{d/\mu + D^2}}{\epsilon}\Big)$. Assume there is a stochastic zeroth-order oracle that provides an unbiased evaluation of $\nabla f(x)$ with bounded variance $\mathbb{E}\|\widetilde{g} - \nabla f(x)\|^2 \le \sigma^2$. Then inexact ULD achieves $W_2(\rho_n, \rho) \le \epsilon$ using*

$$\frac{T}{h} = \widetilde{\Theta}\Big(\frac{\kappa^2\sqrt{d}}{\epsilon}\Big) \tag{C.3}$$

iterations and

$$b = \frac{d^{1.5}\max\{1, \sigma^2\}}{\epsilon} \tag{C.4}$$

queries to the zeroth-order oracle per iteration. The total number of calls is $\frac{bT}{h}$.

Lemma C.3 (Theorem 2.3 of [36])**.** *Let ρ_n be the distribution of the randomized midpoint method for underdamped Langevin diffusion with the initial point x_0, step size $h \le \min\Big\{\frac{\epsilon^{1/3}\mu^{1/6}}{\kappa^{1/6}d^{1/6}\log^{1/6}\big(\frac{\sqrt{d/\mu}}{\epsilon}\big)}, \frac{\epsilon^{2/3}\mu^{1/3}}{d^{1/3}\log^{1/3}\big(\frac{\sqrt{d/\mu}}{\epsilon}\big)}\Big\}$, and time $T \ge 2\kappa\log\Big(\frac{20d/\mu}{\epsilon^2}\Big)$. Assume there is a stochastic zeroth-order oracle that provides an unbiased evaluation of $\nabla f(x)$ with bounded variance $\mathbb{E}\|\widetilde{g} - \nabla f(x)\|^2 \le \sigma^2$. Then inexact ULD-RMM achieves $W_2(\rho_n, \rho) \le \epsilon$ using*

$$\frac{2T}{h} = \widetilde{\Theta}\Big(\frac{\kappa^{7/6}d^{1/6}}{\epsilon^{1/3}} + \frac{\kappa d^{1/3}}{\epsilon^{2/3}}\Big) \tag{C.5}$$

iterations and

$$b = \frac{d\kappa}{h^3} \tag{C.6}$$

queries to the zeroth-order oracle per iteration. The total number of calls is $\frac{bT}{h}$.

As a quantum counterpart, we are able to reduce the number of queries from $O(b)$ to $O(1)$ for each iteration in Lemma C.2 and Lemma C.3 based on Lemma C.1. Here we are able to choose $\epsilon = O(\frac{\sigma^2}{d^3 L})$ to preserve the condition

$$\mathbb{E}\|\widetilde{g} - \nabla f(x)\|^2 \le \mathbb{E}\|\widetilde{g} - \nabla f(x)\|_1^2 \le \sigma^2 \tag{C.7}$$

used in Lemma C.2 and Lemma C.3 with $O(1)$ additional quantum queries. The total number of calls is $O(\frac{T}{h})$ in Lemma C.2 and Lemma C.3, the same scaling as in Lemma B.1 and Lemma B.2. The query complexities of ULD and ULD-RMM are as follows.

Theorem C.1. *Assume the target distribution ρ is strongly log-concave with L-smooth and μ-strongly convex negative log-density. Let ρ_n be the distribution of the underdamped Langevin diffusion with the initial point x_0 satisfying $\|x_0 - x^*\| \le D$, step size $h \le \frac{\epsilon}{104\kappa}\sqrt{\frac{1}{d/\mu+D^2}}$, and time $T \ge \frac{\kappa}{2}\log\Big(\frac{24\sqrt{d/\mu+D^2}}{\epsilon}\Big)$. Then quantum inexact ULD (Algorithm 6) achieves*

$$\mathbb{E}\Big(\|\widehat{X}_n - X_T\|^2\Big) \le \widetilde{O}\Big(\frac{d^2\kappa^2h^2}{\mu}\Big), \tag{C.8}$$

$$W_2(\rho_n, \rho) \le \epsilon, \tag{C.9}$$

using

$$\frac{T}{h} = \widetilde{\Theta}\Big(\frac{\kappa^2\sqrt{d}}{\epsilon}\Big) \tag{C.10}$$

queries to the quantum evaluation oracle.

Proof. By Lemma C.2, we know that the number of iterations of ULD with an inexact gradient oracle is $\widetilde{O}(\kappa^2\sqrt{d}/\epsilon)$, as long as the oracle satisfies $\mathbb{E}\|\widetilde{g} - \nabla f(x)\|^2 \le \sigma^2$. By Lemma C.1, this condition can be achieved by the quantum gradient algorithm such that each gradient computation takes $O(1)$ queries to the quantum evaluation oracle. Therefore, the total number of queries is $\widetilde{O}(\kappa^2\sqrt{d}/\epsilon)$ for the quantum inexact ULD. □

Theorem C.2. *Assume the target distribution ρ is strongly log-concave with L-smooth and μ-strongly convex negative log-density. Let ρ_n be the distribution of the randomized midpoint method for underdamped Langevin diffusion with initial point x_0, step size $h \le \min\Big\{\frac{\epsilon^{1/3}\mu^{1/6}}{\kappa^{1/6}d^{1/6}\log^{1/6}\left(\frac{\sqrt{d/\mu}}{\epsilon}\right)}, \frac{\epsilon^{2/3}\mu^{1/3}}{d^{1/3}\log^{1/3}\left(\frac{\sqrt{d/\mu}}{\epsilon}\right)}\Big\}$, and time $T \ge 2\kappa\log\Big(\frac{20d/\mu}{\epsilon^2}\Big)$. Then quantum inexact ULD-RMM (Algorithm 7) achieves*

$$\mathbb{E}\Big(\|\widehat{X}_n - X_T\|^2\Big) \le \widetilde{O}\Big(\frac{d\kappa h^6}{\mu} + \frac{dh^3}{\mu}\Big), \tag{C.11}$$

$$W_2(\rho_n, \rho) \le \epsilon, \tag{C.12}$$

using

$$\frac{2T}{h} = \widetilde{\Theta}\Big(\frac{\kappa^{7/6}d^{1/6}}{\epsilon^{1/3}} + \frac{\kappa d^{1/3}}{\epsilon^{2/3}}\Big) \tag{C.13}$$

queries to the quantum evaluation oracle.

The proof is almost the same as Theorem C.1, so we omit it here.

Algorithm 6: Quantum Inexact Underdamped Langevin Dynamics (Quantum IULD)

Input: Function f, step size h, time T, and a sample x_0 from a starting distribution ρ_0, evaluation error ϵ, Lipschitz constant L, smoothness parameter β

Output: Sequence $x_h^h, x_{2h}^h, \ldots, x_{\lfloor T\rfloor+1}^h$

Compute $x_0^h \leftarrow x_0$

Compute $\widetilde{g}(x_0) \leftarrow \mathsf{SmoothQuantumGradient}(f, \epsilon, L, \beta, x_0)$

for $t = 0, h, \ldots, \lfloor T\rfloor$ **do**

  Draw $W_{1,t}^h = \int_0^h e^{2(s-h)}\mathrm{d}B_{t+s}$, $W_{2,t}^h = \int_0^h (1 - e^{2(s-h)})\mathrm{d}B_{t+s}$

  Compute $v_{t+h}^h = e^{-2h}v_t^h + \frac{1}{2L}(1 - e^{-2h})\widetilde{g}(x_t^h) + \frac{2}{\sqrt{L}}W_{1,t}^h$

  $x_{t+h}^h = x_t^h + \frac{1}{2}(1 - e^{-2h})v_t^h + \frac{1}{2L}[h - (1 - e^{-2h})]\widetilde{g}(x_t^h) + \frac{1}{\sqrt{L}}W_{2,t}^h$

  Compute $\widetilde{g}(x_{t+h}^h) \leftarrow \mathsf{SmoothQuantumGradient}(f, \epsilon, L, \beta, x_{t+h}^h)$

Algorithm 7: Quantum Inexact Underdamped Langevin Dynamics with Randomized Midpoint Method (Quantum IULD-RMM)

Input: Function f, step size h, time T, and a sample x_0 from a starting distribution ρ_0
Output: Sequence $x_h^h, x_{2h}^h, \ldots, x_{\lfloor T \rfloor+1}^h$

Compute $x_0^h \leftarrow x_0$, $y_0^h \leftarrow x_0$
Compute $\widetilde{g}(x_0^h) \leftarrow \mathsf{SmoothQuantumGradient}(f, \epsilon, L, \beta, x_0^h)$
$\widetilde{g}(y_0^h) \leftarrow \mathsf{SmoothQuantumGradient}(f, \epsilon, L, \beta, y_0^h)$
for $t = 0, h, \ldots, \lfloor T \rfloor$ **do**
  Draw $W_{1,t}^h = \int_0^h e^{2(s-h)} \mathrm{d}B_{t+s}$, $W_{2,t}^h = \int_0^h (1 - e^{2(s-h)}) \mathrm{d}B_{t+s}$, $W_{3,t}^h = \int_0^{\alpha h} (1 - e^{2(s-h)}) \mathrm{d}B_{t+s}$
  Compute $v_{t+h}^h = e^{-2h} v_t^h + \frac{h}{L} e^{-2(1-\alpha)h} \widetilde{g}(y_t^h) + \frac{2}{\sqrt{L}} W_{1,t}^h$
  $x_{t+h}^h = x_t^h + \frac{1}{2}(1 - e^{-2h}) v_t^h + \frac{h}{2L}(1 - e^{-2(1-\alpha)h}) \widetilde{g}(y_t^h) + \frac{1}{\sqrt{L}} W_{2,t}^h$
  $y_{t+h}^h = x_t^h + \frac{1}{2}(1 - e^{-2\alpha h}) v_t^h + \frac{1}{2L}[\alpha h - (1 - e^{-2\alpha h})] \widetilde{g}(x_t^h) + \frac{1}{\sqrt{L}} W_{3,t}^h$
  Compute $\widetilde{g}(x_{t+h}^h) \leftarrow \mathsf{SmoothQuantumGradient}(f, \epsilon, L, \beta, x_{t+h}^h)$
  $\widetilde{g}(y_{t+h}^h) \leftarrow \mathsf{SmoothQuantumGradient}(f, \epsilon, L, \beta, y_{t+h}^h)$

C.2 Quantum MALA

In Appendix C.2.1, we introduce several classical results on the mixing of MALA. Then, in Appendix C.2.2, we describe how to implement a quantum walk for MALA and describe the quantum speedup for MALA with a Gaussian initial distribution. Finally, in Appendix C.2.3, we discuss quantum MALA with a warm start distribution.

C.2.1 Mixing time and spectral gap of MALA

The Metropolis adjusted Langevin algorithm (MALA) is a key method for sampling log-concave distributions. Classically, the state-of-the-art mixing time bound of MALA was proven by [27]. They show that MALA is equivalent to the Metropolized Hamiltonian Monte Carlo method (HMC). Then, they consider the following Metropolized HMC algorithm (Algorithm 8) and use the blocking conductance analysis of [24] to upper bound the mixing time.

Algorithm 8: Metropolized HMC: HMC(x_0, η)

Input: Initial point $x_0 \in \mathbb{R}^d$, step size η
Output: Sequence $\{x_k\}, k \geq 0$

for $k \geq 0$ **do**
  Draw $v_k \sim \mathcal{N}(0, I_d)$
  $(\widetilde{x}_k, \widetilde{v}_k) \leftarrow$ LEAPFROG(η, x_k, v_k)
  Draw $u \sim \mathcal{U}([0,1])$
  if $u \leq \min\{1, \exp(\mathcal{H}(x_k, v_k) - \mathcal{H}(\widetilde{x}_k, \widetilde{v}_k))\}$ **then**
    $x_{k+1} \leftarrow \widetilde{x}_k$
  else
    $x_{k+1} \leftarrow x_k$

Algorithm 9: LEAPFROG(η, x, v)

Input: Points $x, v \in \mathbb{R}^d$, step size η
Output: Points $\widetilde{x}, \widetilde{v} \in \mathbb{R}^d$

$v' \leftarrow v - \frac{\eta}{2} \nabla f(x)$
$\widetilde{x} \leftarrow x + \eta v'$
$\widetilde{v} \leftarrow v' - \frac{\eta}{2} \nabla f(x)$

Define $\mathcal{H}(x,v) := f(x) + \frac{1}{2}\|v\|_2^2$. Let $\mathrm{d}\pi^\star$ denote the target distribution, i.e., $\mathrm{d}\pi^\star(x)/\mathrm{d}x \propto \exp(-f(x))$. Then the Markov chain defined by Algorithm 8 has the following property.

Lemma C.4 ([27])**.** *The Markov chain of Algorithm 8 is reversible, and its stationary distribution is* $\mathrm{d}\pi^\star$.

The main result of [27] is the following theorem on the mixing time of Algorithm 8.

Theorem C.3 (Mixing of Hamiltonian Monte Carlo, Theorem 4.7 of [27])**.** *There is an algorithm initialized from a point drawn from* $\mathcal{N}(x^\star, L^{-1}I_d)$ *that iterates Algorithm 8*

$$O(\kappa d \log(\kappa/\epsilon)\log(d\log(\kappa/\epsilon))\log(1/\epsilon)) \tag{C.14}$$

times and produces a point from a distribution ρ *such that* $\|\rho - \pi^\star\|_{TV} \le \epsilon$.

The algorithm in the above theorem defines a new Markov chain where in each step, we draw an integer j uniformly from 0 to $O(\kappa d \log(\kappa/\epsilon)\log(d\log(\kappa/\epsilon)))$ and run Algorithm 8 for j iterations. One step of this Markov chain gives a distribution with TV-distance from $\pi^\star$ at most $(2e)^{-1}$ [27]. Hence, if we run for $\log(1/\epsilon)$ steps, we get ϵ TV-distance.

We also use a well-known relation between the mixing time and the spectral gap of Markov chain (see e.g. Chapter 12 in [29]).

Theorem C.4 (Mixing time and spectral gap)**.** *For a reversible, irreducible, and aperiodic Markov chain, let* $t_{mix}(\epsilon)$ *denote the* ϵ*-mixing time and let* δ *denote the spectral gap. Then*

$$t_{mix}(\epsilon) \ge (\delta^{-1} - 1)\log(1/(2\epsilon)). \tag{C.15}$$

It is easy to verify that the Markov chain of Algorithm 8 is reversible, irreducible, and aperiodic. Hence, together with Theorem C.3, we know that the spectral gap of the Markov chain satisfies

$$\delta^{-1} \le O(\kappa d \log(\kappa/\epsilon)\log(d\log(\kappa/\epsilon))). \tag{C.16}$$

Furthermore, we can show that MALA converges faster under a certain warm start condition [10, 14, 47]. We say the initial distribution ρ_0 is β*-warm* if there is a constant β independent of κ, d such that

$$\sup_{S\in\mathcal{B}(\mathbb{R}^d)} \frac{\rho_n(S)}{\rho(S)} \le \beta. \tag{C.17}$$

The warmness of the Gaussian $\rho_0 = \mathcal{N}(x^*, \frac{1}{L}I)$ satisfies $\beta \le \kappa^{d/2}$ [14, 47].

Given a β-warm initial distribution, MALA has the following improved convergence.

Lemma C.5 (Theorem 1 of [47])**.** *Assume the target distribution* ρ *is strongly log-concave with* L*-smooth and* μ*-strongly convex negative log-density. Let* ρ_n *be the distribution of the* $\frac{1}{2}$*-lazy version of MALA with* β*-warm initial distribution* ρ_0 *and step size* $h = c_0(Ld\log^2(\max\{\kappa, d, \frac{\beta}{\epsilon}, c_2\}))^{-1}$. *There exist universal constants* $c_0, c_1, c_2 > 0$*, such that MALA achieves*

$$d_{TV}(\rho_n, \rho) \le \epsilon \tag{C.18}$$

after

$$n \ge c_1\kappa\sqrt{d}\log^3(\max\{\kappa, d, \frac{\beta}{\epsilon}, c_2\}) \tag{C.19}$$

steps.

C.2.2 Quantum walks for MALA

The goal of this section is to show quantum speedup for the Metropolis adjusted Langevin algorithm (MALA) using the continuous-space quantum walks defined by [6], which generalize the discrete-time quantum walk of [41] to continuous space.

Given a transition density function p, the quantum walk is characterized by the states

$$|\phi_x\rangle := |x\rangle \otimes \int_\Omega \mathrm{d}y\, \sqrt{p_{x\to y}}|y\rangle \qquad \forall x \in \mathbb{R}^n, \tag{C.20}$$

where $p_{x\to y} := p(x,y)$.

Now, denote

$$U := \int_\Omega \mathrm{d}x\, |\phi_x\rangle(\langle x| \otimes \langle 0|), \quad \Pi := \int_\Omega \mathrm{d}x\, |\phi_x\rangle\langle\phi_x|, \quad S := \int_\Omega \int_\Omega \mathrm{d}x\, \mathrm{d}y\, |x,y\rangle\langle y,x|. \tag{C.21}$$

A single step of the quantum walk is defined as the unitary operator

$$W := S(2\Pi - I). \tag{C.22}$$

The following theorem characterizes the eigenvalues of the quantum walk operator.

Theorem C.5 (Theorem 3.1 of [6])**.** *Let*

$$D := \int_\Omega \int_\Omega \mathrm{d}x\, \mathrm{d}y\, \sqrt{p_{x\to y} p_{y\to x}} |x\rangle\langle y| \tag{C.23}$$

denote the discriminant operator *of p. Let Λ be the set of eigenvalues of D, so that $D = \int_\Lambda \mathrm{d}\lambda\, \lambda |\lambda\rangle\langle\lambda|$. Then the eigenvalues of the quantum walk operator W in (C.22) are ± 1 and $\lambda \pm i\sqrt{1-\lambda^2}$ for all $\lambda \in \Lambda$.*

Furthermore, Ref. [6] shows that, for a reversible Markov chain with unique stationary distribution ρ, the state

$$|\rho_W\rangle := \int_\Omega \mathrm{d}x \sqrt{\rho_x} |\phi_x\rangle \tag{C.24}$$

is the unique eigenvalue-1 eigenstate of the quantum walk operator W restricted to the subspace $\mathrm{span}_{\lambda\in\Lambda}\{T|\lambda\rangle, ST|\lambda\rangle\}$. Hence, the stationary distribution corresponds to an eigenstate with eigenphase 0, while the other eigenstates have eigenphase at least $\sqrt{2\delta}$, where δ is the spectral gap of P. Thus, by the quantum phase estimation algorithm with $O(1/\sqrt{\delta})$ calls to W, we can distinguish the stationary state from other eigenstates, achieving quadratic speedup over the classical mixing time $O(1/\delta)$.

To implement a quantum version of Algorithm 8, we prepare the initial state $|\pi_0\rangle$ and implement the quantum walk operator W.

Initial state. For the initial state $|\rho_0\rangle$, by Theorem C.3, it suffices to take $\rho_0 = \mathcal{N}(x^\star, L^{-1} I_d)$, where $x^\star$ is the minimum point of $f(x)$. Suppose we already have $x^\star$. Appendix A.3 of [6] shows that the state

$$\int_{\mathbb{R}^d} \left(\frac{L}{2\pi}\right)^{d/4} e^{-\frac{L}{4}\|z\|_2^2} |z\rangle \mathrm{d}z \tag{C.25}$$

can be efficiently prepared by applying a Box-Muller transformation to the state corresponding to the uniform distribution (i.e., an equal superposition of points). Then, for the ith register, we apply the shift operation U_{shift} with $U_{\text{shift}}|x_i\rangle = |x_i + x_i^\star\rangle$. The resulting state is

$$|\rho_0\rangle = \int_{\mathbb{R}^d} \left(\frac{L}{2\pi}\right)^{d/4} e^{-\frac{L}{4}\|z-x^\star\|_2^2} |z\rangle \mathrm{d}z. \tag{C.26}$$

Quantum walk operator. The quantum walk operator $W(P)$ can be implemented[1] using the quantum walk update unitary U that maps each point $|x\rangle$ to the superposition $\int_{\mathbb{R}^d} \mathrm{d}y \sqrt{p_{x\to y}} |y\rangle$. We show how to efficiently implement U.

We first use d ancilla registers to prepare a standard Gaussian state

$$|v\rangle := \int_{\mathbb{R}^d} \left(\frac{L}{2\pi}\right)^{d/4} e^{-\frac{L}{4}\|z\|_2^2} |z\rangle \mathrm{d}z. \tag{C.27}$$

[1] As shown in [45], $W(P) = U^\dagger SURU^\dagger SUR$, where S is the swap gate and R is a reflection operator with respect to the state space $\mathrm{span}\{|x\rangle|0\rangle : x \in \mathbb{R}^d\}$.

Then, we implement a unitary U_{LF} defined by LEAPFROG(η, x, v) (Algorithm 9) such that for two points $x, v \in \mathbb{R}^d$, $U_{\mathrm{LF}}|x, v\rangle|0\rangle = |x, v\rangle|\widetilde{x}, \widetilde{v}\rangle$, by querying the gradient oracle of f twice. Then we use another ancilla register to prepare the state

$$|u\rangle := \int_{[0,1]} \mathrm{d}|z\rangle. \tag{C.28}$$

Based on the registers $x, v, \widetilde{x}, \widetilde{v}, u$, we can decide using two queries to the evaluation oracle for f whether the target register y should be x or $\widetilde{x}$. Overall, this process implements the following mapping (up to a hidden normalization factor):

$$|x\rangle|0\rangle \mapsto |x\rangle \int_{\mathbb{R}^d} \int_{[0,1]} \mathrm{d}v\mathrm{d}u \exp\big(-\|v\|_2^2/4\big)|v\rangle|\widetilde{x}\rangle|\widetilde{v}\rangle|u\rangle|y\rangle. \tag{C.29}$$

Finally, uncomputing the $v, \widetilde{x}, \widetilde{v}$ registers using $U_{\mathrm{LF}}^\dagger$ and the inverse of the unitary preparing $|v\rangle$, we obtain a superposition of points with the correct transition density.

Algorithm 10: QUANTUMUPDATEUNITARY

Input: Quantum state $|\rho\rangle = \int_{\mathbb{R}^d} \mathrm{d}x \sqrt{\rho_x}|x\rangle$.
Output: Quantum state $|\phi\rangle = \int_{\mathbb{R}^d} \mathrm{d}x \sqrt{\rho_x}|x\rangle \int_{\mathbb{R}^d} \mathrm{d}y \sqrt{p_{x\to y}}|y\rangle$.

1 Prepare $|\rho\rangle|v\rangle$ where $|v\rangle$ is a d-dimensional Gaussian state
2 Apply the leap-frog process to $|\rho\rangle|v\rangle$: $\int_{\mathbb{R}^d} \mathrm{d}x\, \mathrm{d}v|x, v\rangle|\widetilde{x}, \widetilde{v}\rangle$; `/* Query` $\mathcal{O}_{\nabla f}$ `twice */`
3 Prepare $\int_{\mathbb{R}^d} \mathrm{d}x\, \mathrm{d}v|x, v\rangle|\widetilde{x}, \widetilde{v}\rangle \int_{[0,1]} \mathrm{d}u|u\rangle$
4 Compute the target point y: $\int_{\mathbb{R}^d} \mathrm{d}x\, \mathrm{d}v \int_{[0,1]} \mathrm{d}u|x, v\rangle|\widetilde{x}, \widetilde{v}\rangle|u\rangle|y\rangle$; `/* Query` $\mathcal{O}_f$ `twice */`
5 Uncompute the $v, \widetilde{x}, \widetilde{v}, u$ registers: $\int \mathrm{d}x \sqrt{\rho_x}|x\rangle \int_{\mathbb{R}^d} \mathrm{d}y \sqrt{p_{x\to y}}|y\rangle$

Algorithm 11: QUANTUMMALA

Input: Evaluation oracle $\mathcal{O}_f$, gradient oracle $\mathcal{O}_{\nabla f}$, initial state $|\rho_0\rangle$
Output: Quantum state $|\widetilde{\rho}\rangle$ close to the stationary distribution state $\int_{\mathbb{R}^d} e^{-f(x)}\mathrm{d}|x\rangle$

1 Construct quantum walk update unitary U from QUANTUMUPDATEUNITARY (Algorithm 10) with $\mathcal{O}_f$ and $\mathcal{O}_{\nabla f}$
2 Implement the quantum walk operator $W(P)$
3 Perform $\frac{\pi}{3}$-amplitude amplification with $W(P)$ on the state $|\rho_0\rangle|0\rangle$
4 **return** the resulting state $|\widetilde{\rho}\rangle$

Lemma C.6 (Continuous-space quantum walk implementation)**.** *The Markov chain of Algorithm 8 can be implemented as a continuous-space quantum walk where the quantum walk unitary for one step can be implemented with 2 queries to the gradient oracle, 2 queries to the evaluation oracle, and $O(d)$ quantum gates.*

If we have a sequence of slowly varying log-concave distributions $\rho_0, \ldots, \rho_r$, we can quantumly sample from ρ_r via MALA with a quadratic speedup.

Theorem C.6 (Quantum speedup for slowly varying Markov chains [45])**.** *Let $M_0, \ldots, M_r$ be classical reversible Markov chains with stationary distributions $\rho_0, \ldots, \rho_r$ such that each chain has spectral gap at least δ^{-1}. Assume that $|\langle\rho_i|\rho_{i+1}\rangle| \geq p$ for some $p > 0$ and all $i \in \{0, \ldots, r-1\}$, and that we can prepare the state $|\rho_0\rangle$. Then, for any $\epsilon > 0$, there is a quantum algorithm which produces a quantum state $|\widetilde{\rho_r}\rangle$ such that $\||\widetilde{\rho}_r\rangle - |\rho_r\rangle|0^a\rangle\| \leq \epsilon$, for some integer a. The algorithm uses*

$$\tilde{O}\left(\delta^{-1/2} \cdot \frac{r}{p}\right) \tag{C.30}$$

applications of the quantum walk operators W_i corresponding to the chains M_i for $i \in [r]$.

Combining Theorem C.6 with the lower bound for the spectral gap of MALA (Eq. (C.16)), we find the following.

Corollary C.1 (Quantum speedup for slowly varying MALAs)**.** *Let $\rho_0, \ldots, \rho_r$ be a sequence of log-concave distributions such that $|\langle \rho_i | \rho_{i+1} \rangle| \geq p$ for some $p > 0$ and for all $i \in \{0, \ldots, r-1\}$. Suppose we can prepare the initial state $|\rho_0\rangle$. Then, for any $\epsilon > 0$, there is a quantum procedure to produce a state $|\widetilde{\rho}_r\rangle$ such that $\| |\widetilde{\rho}_r\rangle - |\rho_r\rangle \| \leq \epsilon$ using*

$$\widetilde{O}(\kappa^{1/2} d^{1/2} \cdot (r/p)) \tag{C.31}$$

applications of the quantum walk operators W_i corresponding to the MALA procedure for ρ_i for $i \in [r]$.

Algorithm 12: QUANTUMMALAFORLOG-CONCAVESAMPLING

Input: Evaluation oracle $\mathcal{O}_f$, gradient oracle $\mathcal{O}_{\nabla f}$, smoothness parameter L, convexity parameter μ

Output: Quantum state $|\widetilde{\rho}\rangle$ close to the stationary distribution state $\int_{\mathbb{R}^d} e^{-f(x)} \mathrm{d}|x\rangle$

Compute the cooling schedule parameters $\sigma_1, \ldots, \sigma_M$

Prepare the state $|\rho_0\rangle \propto \int_{\mathbb{R}^d} e^{-\frac{1}{4}\|x\|^2/\sigma_1^2} \mathrm{d}|x\rangle$

for $i \leftarrow 1, \ldots, M$ **do**

  Construct $\mathcal{O}_{f_i}$ and $\mathcal{O}_{\nabla f_i}$ where $f_i(x) = f(x) + \frac{1}{2}\|x\|^2/\sigma_i^2$

  $|\rho_i\rangle \leftarrow$ QUANTUMMALA$(\mathcal{O}_{f_i}, \mathcal{O}_{\nabla f_i}, |\rho_{i-1}\rangle)$ (Algorithm 11)

return $|\rho_M\rangle$

Theorem C.7 (Quantum MALA for log-concave sampling)**.** *Assume the target distribution $\rho \propto e^{-f}$ is strongly log-concave with $f \colon \mathbb{R}^d \to \mathbb{R}_+$ being L-smooth and μ- strongly convex. Let $|\rho\rangle$ be the quantum state corresponding to the distribution ρ. Then, for any $\epsilon > 0$, there is a quantum algorithm (Algorithm 12) that prepares a state $|\widetilde{\rho}\rangle$ such that $\| |\widetilde{\rho}\rangle - |\rho\rangle \| \leq \epsilon$ using $\widetilde{O}(\kappa^{1/2} d)$ queries to the evaluation oracle $\mathcal{O}_f$ and gradient oracle $\mathcal{O}_{\nabla f}$.*

Proof. By Lemma B.6, we know that the cooling schedule $\sigma_1, \ldots, \sigma_M$ gives a sequence of slowly-varying Markov chains with overlap $p = \Omega(1)$. We also know that the length of the schedule is $M = \tilde{O}(d^{1/2})$.

Hence, by Corollary C.1 with $r = \tilde{O}(d^{1/2})$ and $p = \Omega(1)$, $|\widetilde{\rho}\rangle$ can be prepared by $\tilde{O}(\kappa^{1/2} d)$ quantum walk steps. By Lemma C.6, each step queries $\mathcal{O}_f$ and $\mathcal{O}_{\nabla f}$ twice. The result follows. □

C.2.3 Quantum MALA with a warm start

We now show that quantum MALA can achieve further speedup in query complexity when the initial distribution is a warm start (Theorem C.8).

The following proposition shows that when the initial distribution is warm, then the eigenvalues whose eigenspaces have large overlap with the initial state are bounded away from 1. Our result improves Proposition 4.2 of [7], which required the initial distribution to satisfy an L_2-norm condition, by instead relying only on the standard warmness of the initial distribution.

Lemma C.7 (Effective spectral gap for warm start)**.** *Let $M = (\Omega, p)$ be an ergodic reversible Markov chain with a transition operator P and unique stationary state with a corresponding density ρ. Let $\{(\lambda_i, f_i)\}$ be the set of eigenvalues and eigenfunctions of P, and let $|u_i\rangle$ be the eigenvectors of the corresponding quantum walk operator W. Let ρ_0 be a probability density that is a warm start for ρ and mixes up to TV-distance ϵ in t steps of M. Furthermore, assume that ρ_0 is a β-warm start of ρ. Let $|\phi_{\rho_0}\rangle$ be the state obtained by applying the quantum walk update operator U to the state $|\rho_0\rangle$:*

$$|\phi_{\rho_0}\rangle = \int_\Omega \sqrt{\rho_0(x)} \int_\Omega \sqrt{p_{x \to y}} |x\rangle |y\rangle \, \mathrm{d}x \, \mathrm{d}y. \tag{C.32}$$

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

Proof. Let $S := \left\{x \in \mathbb{R}^d : \frac{\rho(x)}{\rho_0(x)} \geq \frac{1}{\epsilon}\right\}$. Since $\mathbb{E}_{\rho_0}\left[\frac{\rho(x)}{\rho_0(x)}\right] = 1$, by Markov's inequality, we have

$$\int_S \rho_0(x)\mathrm{d}x = \Pr_{\rho_0}[x \in S] \leq \epsilon. \tag{C.33}$$

Then we define a quantum state $|\rho_1\rangle$ such that $\langle\rho_1|x\rangle = \langle\rho_0|x\rangle$ for $x \notin S$, and $\langle\rho_1|x\rangle = 0$ for $x \in S$. Furthermore, let $|\phi_{\rho_1}\rangle := U|\rho_1\rangle$.

We have

$$\||\phi_{\rho_0}\rangle - |\phi_{\rho_1}\rangle\| = \left\|\int_S \sqrt{\rho_0(x)}T|x\rangle\mathrm{d}x\right\| = \left|\int_S \rho_0(x)\mathrm{d}x\right|^{1/2} \leq \sqrt{\epsilon}, \tag{C.34}$$

where T is the isometry

$$T := \int_\Omega \int_\Omega \sqrt{p_{x\to y}}|x, y\rangle\langle x|\,\mathrm{d}x\,\mathrm{d}y.$$

Moreover, by Eqs. (4.35) and (4.36) in [6], if $1 > \lambda_i \geq 1 - O(1/t)$, we have

$$|\langle\phi_{\rho_1}|u_i\rangle| \leq 2|\langle\rho_1|v_i\rangle| = 2\left|\int_{\overline{S}} \sqrt{\frac{\rho(x)}{\rho_0(x)}}\frac{\rho_0(x)f_i(x)}{\rho(x)}\mathrm{d}x\right| \leq \frac{2\langle\rho_0, f_i\rangle_\rho}{\epsilon^{1/2}} = O(\beta\sqrt{\epsilon}), \tag{C.35}$$

where the third step follows from $\frac{\rho(x)}{\rho_0(x)} \leq \frac{1}{\epsilon}$ for $x \notin S$ and the Cauchy-Schwarz inequality, and the last step follows from the claim that $\langle\rho_0, f_i\rangle_\rho = O(\beta\epsilon)$.

Combining these observations, we find

$$|\langle\phi_{\rho_0}|u_i\rangle| \leq |\langle\phi_{\rho_0} - \phi_{\rho_1}|u_i\rangle| + |\langle\phi_{\rho_1}|u_i\rangle| \leq \sqrt{\epsilon} + O(\beta\sqrt{\epsilon}) = O(\beta\sqrt{\epsilon}) \tag{C.36}$$

when $1 > \lambda_i \geq 1 - O(1/t)$, which gives the desired result.

Now, it remains to prove the claim that $\langle\rho_0, f_i\rangle_\rho = O(\beta\epsilon)$ for $1 > \lambda_i > 1 - O(1/t)$. Suppose ρ_0 can be decomposed in the eigenbasis of P as $\rho_0 = \rho + \sum_{i=2}^\infty \langle\rho_0, f_i\rangle_\rho f_i$. Then $P^t\rho_0 = \rho + \sum_{i=2}^\infty \lambda_i^t\langle\rho_0, f_i\rangle_\rho f_i$, where λ_i is the eigenvalue of f_i. Since $\|P^t\rho_0 - \rho\|_1 \leq \epsilon$, by Fact C.1, we have $\|P^t\rho_0 - \rho\|_\rho \leq \beta\epsilon$. Hence, by the orthogonality of f_i, we have $\lambda_i^t\langle a, f_i\rangle_\rho \leq \beta\epsilon$. Therefore, when $1 > \lambda_i > 1 - O(1/t)$, we have $\lambda_i^t = \Omega(1)$, which implies that $\langle\rho_0, f_i\rangle_\rho = O(\beta\epsilon)$. □

Fact C.1. *Let ρ_0 be a β-warm start of a Markov chain with transition operator P and stationary distribution ρ. If $\|P^t\rho_0 - \rho\|_1 \leq \epsilon$ for some $t > 0$, then $\|P^t\rho_0 - \rho\|_\rho \leq \beta \cdot \epsilon$.*

Proof. We expand $\|P^t\rho_0 - \rho\|_\rho$ in terms of its definition, giving

$$\|P^t\rho_0 - \rho\|_\rho = \int_{\mathbb{R}^d} \frac{(P^t\rho_0 - \rho)^2(x)}{\rho(x)}\mathrm{d}x = \int_{\mathbb{R}^d} \frac{(P^t\rho_0)^2(x)}{\rho(x)}\mathrm{d}x - 1 = \chi^2(P^t\rho_0, \rho), \tag{C.37}$$

where the second step follows since $P^t\rho_0$ and ρ are distributions, and the last step follows from the definition of χ^2-distance. Let $\rho_1 := P^t\rho_0$. By Lemma 27 in [10], we know that ρ_1 is also β-warm. Furthermore, by Lemma 28 in [10], the χ-square distance can be upper bounded by

$$\chi^2(P^t\rho_0, \rho) \leq \beta \cdot \|P^t\rho_0 - \rho\|_1 \leq \beta\epsilon \tag{C.38}$$

as claimed. □

Theorem C.8 (Quantum MALA with warm start)**.** *Let $|\rho_0\rangle$ be a β-warm start with respect to the log-concave distribution $\rho \propto e^{-f}$. Let $\mathcal{T}$ be the mixing time of classical MALA with initial distribution ρ_0, i.e., $\|P^{\mathcal{T}}\rho_0 - \rho\|_{\mathrm{TV}} \leq \epsilon$. Then there is a quantum algorithm that prepares a state $|\widetilde{\rho}\rangle$ that is ϵ-close to $|\rho\rangle$ using*

$$\widetilde{O}\left(\sqrt{\mathcal{T}\log(\beta)}\right) \tag{C.39}$$

queries to the evaluation oracle $\mathcal{O}_f$ and gradient oracle $\mathcal{O}_{\nabla f}$.

Proof. Let $|\rho_0\rangle$ be the initial state corresponding to a distribution ρ_0 that is β-warm. Then we know that $t = \widetilde{O}(\mathcal{T}\log(\beta))$ steps of the classical MALA random walk suffice to achieve $\|P^t\rho_0 - \rho\|_1 \leq \epsilon/\beta^2$.

By Lemma C.7, we have $|\rho_0\rangle = |\rho_0'\rangle + |e\rangle$, where $|\rho_0'\rangle$ lies in the space of eigenvectors $|v_i\rangle$ of W such that $\lambda_i = 1$ or $\lambda_i \leq 1 - \widetilde{\Omega}(\mathcal{T}^{-1}\log^{-1}(\beta))$, and $\||e\rangle\| \leq \epsilon_1$.

Hence, by Corollary 4.1 in [7], the approximate reflection in the quantum walk $\widetilde{R}$ can be implemented using $\widetilde{O}(\mathcal{T}^{1/2}\log^{1/2}(\beta))$ calls to the controlled-W operator. Furthermore, the approximated stationary state $|\widetilde{\rho}\rangle$ can be prepared via $O(\log(1/\epsilon_2))$ recursive levels of $\frac{\pi}{3}$-amplitude amplification such that $\||\rho\rangle - |\widetilde{\rho}\rangle\| \leq \epsilon_2$.

By choosing $\epsilon_1 = \epsilon/(2\log(2/\epsilon))$ and $\epsilon_2 = \epsilon/2$, we achieve a final approximation error of $O(\epsilon_1\log(\epsilon_2) + \epsilon_2) \leq \epsilon$. By Lemma C.6, each controlled-W operator takes a constant number of queries to $\mathcal{O}_f$ and $\mathcal{O}_{\nabla f}$, which gives the desired result. □

Remark C.1. *The above theorem is quite general and works for both polynomially-warm and Gaussian initial distributions.*

- *When ρ_0 is a β-warm start for $\beta \leq \mathrm{poly}(d)$[2], by Lemma C.5, we have $\mathcal{T} = \widetilde{O}(\kappa d^{1/2})$. Then the query complexity of quantum MALA is $\widetilde{O}(\kappa^{1/2}d^{1/4})$.*
- *When $\rho_0 = \mathcal{N}(0, L^{-1}I_d)$, the warmness is $\kappa^{d/2}$, and by Theorem C.3, we have $\mathcal{T} = \widetilde{O}(\kappa d)$. Hence, the query complexity of quantum MALA is $\widetilde{O}(\kappa^{1/2}d)$, recovering Theorem C.7.*

Algorithm 13: QUANTUMMALAWITHWARMSTART

Input: Evaluation oracle $\mathcal{O}_f$, gradient oracle $\mathcal{O}_{\nabla f}$, smoothness parameter L, convexity parameter μ, warm-start state $|\rho_0\rangle$

Output: Quantum state $|\widetilde{\rho}\rangle$ close to the stationary distribution state $\int_{\mathbb{R}^d} e^{-f(x)}\mathrm{d}|x\rangle$

$|\widetilde{\rho}\rangle \leftarrow$ QUANTUMMALA$(\mathcal{O}_{f_i}, \mathcal{O}_{\nabla f_i}, |\rho_0\rangle)$ (Algorithm 11)

return $|\widetilde{\rho}\rangle$

D Quantum Algorithm for Estimating Normalizing Constants: Details

We now come back to the problem of estimating the normalizing constant.

D.1 Quantum MALA and annealing

In this section, we first describe a quantum speedup for the annealing process via a quantum-accelerated Monte Carlo method, which quadratically improves the ϵ-dependence of the sampling complexity of the classical algorithm. Then we further reduce the κ- and d-dependence of the query complexity using the quantum MALA procedure developed in Appendix C.2.

D.1.1 Quantum speedup for the standard annealing process

Reference [31] developed a quantum-accelerated Monte Carlo method for mean estimation with B-bounded relative variance.[3] We state the result as follows.

Lemma D.1 (Theorem 6 of [31])**.** *Assume there is an algorithm $\mathcal{A}$ such that $v(\mathcal{A}) \geq 0$ and $\frac{\mathrm{Var}(v(\mathcal{A}))}{\mathbb{E}[v(\mathcal{A})]^2} \leq B$ for some $B \geq 1$, and an accuracy $\epsilon < 27B/4$. Then there is a quantum algorithm which outputs an estimate $\widetilde{\mu}$ such that*

$$\Pr\Big[|\widetilde{\mu} - \mathbb{E}[v(\mathcal{A})]| \geq \epsilon\mathbb{E}[v(\mathcal{A})]\Big] \leq \frac{1}{4}, \tag{D.1}$$

[2]We assume that we can access an arbitrary number of copies of the quantum state $|\rho_0\rangle$.

[3]Reference [4] improves the scaling from $\widetilde{O}(B/\epsilon)$ to $\widetilde{O}(\sqrt{B}/\epsilon)$. Such a result also follows from the quantum Chebyshev inequality of [20]. Since $B = O(1)$ in our case, we apply the original algorithm of [31].

with

$$O\Big(\frac{B}{\epsilon}\log^{3/2}\Big(\frac{B}{\epsilon}\Big)\log\log\Big(\frac{B}{\epsilon}\Big)\Big) \tag{D.2}$$

queries to $\mathcal{A}$.

Lemma D.2 (Lemma 1 of [31])**.** *Let $\mathcal{A}$ be a (classical or quantum) algorithm which aims to estimate some quantity μ, and whose output $\tilde{\mu}$ satisfies $|\mu-\tilde{\mu}|\leq\epsilon$ except with probability γ, for some fixed $\gamma<1/2$. Then, for any $\delta>0$, it suffices to repeat $\mathcal{A}$ $O(\log 1/\delta)$ times and take the median to obtain an estimate which is accurate to within ϵ with probability at least $1-\delta$.*

This result provides a way of estimating the telescoping product (B.25). The following theorem and its proof closely follows Theorem 8 of [31], while the definitions of the partition function and the cooling schedule are different.

Theorem D.1 (Quantum speedup of annealing)**.** *Let Z be the normalizing constant in (1.3). Consider a sequence of values g_i as in (B.28), with $\frac{\mathbb{E}_{\rho_i}(g_i^2)}{\mathbb{E}_{\rho_i}(g_i)^2}=O(1)$. Further assume that we have the ability to sample ρ_i for $i\in[M]$. Then there is a quantum algorithm which outputs an estimate $\widetilde{Z}$, such that*

$$\Pr[(1-\epsilon)Z\leq\widetilde{Z}\leq(1+\epsilon)Z]\geq\frac{3}{4}, \tag{D.3}$$

using

$$O\Big(\frac{M^2}{\epsilon}\log^{3/2}\Big(\frac{M}{\epsilon}\Big)\log\log\Big(\frac{M}{\epsilon}\Big)\Big)=\widetilde{O}\Big(\frac{M^2}{\epsilon}\Big) \tag{D.4}$$

samples in total.

Proof. For $i\in[M]$, we estimate $\mathbb{E}_{\rho_i}(g_i)$ with output $\widetilde{g}_i$ up to additive error $(\epsilon/2M)\mathbb{E}_{\rho_i}(g_i)$ with failure probability $1/4M$. We output as a final estimate

$$\widetilde{Z}=\widetilde{Z}_1\prod_{i=1}^{M}\widetilde{g}_i, \tag{D.5}$$

where $\widetilde{Z}_1$ is the normalizing constant of the Gaussian distribution with variance σ_1^2 as in Lemma B.3. Assuming that all the estimates are indeed accurate, we have

$$1-\epsilon\leq(1-\frac{\epsilon}{2})\Big(1-\frac{\epsilon}{2M}\Big)^M\leq\frac{\widetilde{Z}}{Z}\leq\Big(1+\frac{\epsilon}{2M}\Big)^M\leq e^{\epsilon/2}\leq 1+\epsilon. \tag{D.6}$$

Thus $|\widetilde{Z}-Z|\leq\epsilon Z$ with probability at least

$$\Big(1-\frac{1}{4M}\Big)^M\geq 1-\frac{1}{4}=\frac{3}{4}. \tag{D.7}$$

Based on Lemma B.4 and Lemma B.5, $\frac{\mathbb{E}_{\rho_i}(g_i^2)}{\mathbb{E}_{\rho_i}(g_i)^2}=O(1)$, so $\frac{\mathrm{Var}_{\rho_i}(g_i)}{\mathbb{E}_{\rho_i}(g_i)^2}=O(1)$. By Lemma D.1, each requires

$$O\Big(\frac{M}{\epsilon}\log^{3/2}\Big(\frac{M}{\epsilon}\Big)\log\log\Big(\frac{M}{\epsilon}\Big)\Big) \tag{D.8}$$

samples from ρ_i, and the total number of samples is

$$O\Big(\frac{M^2}{\epsilon}\log^{3/2}\Big(\frac{M}{\epsilon}\Big)\log\log\Big(\frac{M}{\epsilon}\Big)\Big). \tag{D.9}$$

This completes the proof. □

D.1.2 Quantum MALA for estimating the normalizing constant

We now describe how to combine quantum annealing with quantum MALA to reduce the query complexity of estimating the normalizing constants.

We begin with the following lemma on non-destructive mean estimation.

Lemma D.3 (Non-destructive mean estimation with quantum MALA)**.** *For $\epsilon < 1$, given $\widetilde{O}(\epsilon^{-1})$ copies of a state $|\widetilde{\rho}_{i-1}\rangle$ such that $\||\widetilde{\rho}_{i-1}\rangle - |\rho_{i-1}\rangle\| \leq \epsilon$, there exists a quantum procedure that outputs $\widetilde{g}_i$ such that*

$$|\widetilde{g}_i - \mathbb{E}_{\rho_i}[g_i]| \leq \epsilon \cdot \mathbb{E}_{\rho_i}[g_i] \tag{D.10}$$

with success probability $1 - o(1)$ using

$$\widetilde{O}(\kappa^{1/2} d^{1/2} \epsilon^{-1}) \tag{D.11}$$

steps of the quantum walk operator corresponding to the MALA with stationary distribution ρ_i, where δ is the spectral gap of the Markov chain. The quantum procedure also returns $\widetilde{O}(\epsilon^{-1})$ copies of the state $|\widetilde{\rho}_i\rangle$ such that $\||\widetilde{\rho}_i\rangle - |\rho_i\rangle\| \leq \epsilon$.

Proof sketch. This lemma is a variant of Lemma 4.4 in [7]. Notice that Corollary C.1 implies that we can prepare $|\widetilde{\rho}_i\rangle$ from $|\widetilde{\rho}_{i-1}\rangle$ using $\widetilde{O}(\kappa^{1/2} d^{1/2} p^{-1})$ quantum walk steps, where $p \leq |\langle \rho_i | \rho_{i-1}\rangle|$. By Lemma B.6, we have $p = \Omega(1)$. The lemma follows by properly choosing the parameters in Lemma 4.4 in [7]. □

Theorem D.2 (Quantum speedup using MALA, annealing, and quantum walk)**.** *Let Z be the normalizing constant in (1.3). Assume we are given the access to query the quantum gradient oracle (1.5). Then there is a quantum algorithm which outputs an estimate $\widetilde{Z}$, such that*

$$\Pr[(1-\epsilon)Z \leq \widetilde{Z} \leq (1+\epsilon)Z] \geq \frac{3}{4}, \tag{D.12}$$

using

$$\widetilde{O}\Big(d^{3/2} \kappa^{1/2} \epsilon^{-1}\Big) \tag{D.13}$$

queries to the quantum gradient oracle in total.

Proof. The number of annealing stages is $M = \widetilde{O}(\sqrt{d})$. At the ith stage, we estimate $\mathbb{E}_{\rho_i}[g_i]$ with relative error ϵ/M. Hence, we can apply Lemma D.3 M times, where each application takes

$$\widetilde{O}(\kappa^{1/2} d^{1/2} (\epsilon/M)^{-1}) = \widetilde{O}(\kappa^{1/2} d \epsilon^{-1}) \tag{D.14}$$

MALA quantum walk steps. This process takes

$$M \cdot \widetilde{O}(\kappa^{1/2} d \epsilon^{-1}) = \widetilde{O}(\kappa^{1/2} d^{3/2} \epsilon^{-1}) \tag{D.15}$$

steps in total.

By Lemma C.6, each step of the quantum walk operator can be implemented by querying the gradient oracle and the evaluation oracle $O(1)$ times. Therefore, we can estimate Z with relative error ϵ using $\widetilde{O}(\kappa^{1/2} d^{3/2} \epsilon^{-1})$ queries to the gradient and evaluation oracles. □

D.2 Quantum multilevel Langevin algorithms

We now consider an alternative approach for estimating the normalizing constant, by replacing MALA by a multilevel Langevin approach. More concretely, for each sample we perform the underdamped Langevin diffusion (ULD) or the randomized midpoint method for underdamped Langevin diffusion (ULD-RMM) that has an improved dependence on the dimension, and apply the multilevel Monte Carlo (MLMC) to preserve the dependence on the accuracy.

Multilevel Monte Carlo methods have attracted extensive attention in stochastic simulations and financial models [16, 17]. This approach was originally developed by [22] for parametric integration, and used to simulate SDEs in [16]. Considering a general random variable P, MLMC gives a

sequence of estimators $P_0, P_l, \ldots, P_L$ for approximating P with increasing accuracy and cost, and uses the telescoping sum of $\mathbb{E}[P_l - P_{l-1}]$ to estimate $\mathbb{E}[P]$. For $P_l - P_{l-1}$ with smaller variance but larger cost, MLMC performs fewer samples to reach a given error tolerance, reducing the overall complexity. MLMC has been widely discussed and improved under many settings, and has been used in various applications [17].

To estimate normalizing constants, a variant of MLMC has been proposed by [15, Lemmas C.1 and C.2]. Unlike standard MLMC for bounding the mean-squared error, this approach upper bounds the bias and the variance separately, making the analysis more technically involved. The first quantum algorithm based on MLMC was developed by [2, Theorem 2]. They upper bound the additive error with high probability (as is common for quantum algorithms). They also observe that the mean-squared error can control both the bias and the variance [2, Section 2.2] and that the mean-squared error is almost equivalent to the additive error with high probability [2, Appendix A]. Considering this, we still use the additive error scenario for estimating normalizing constants, both for convenience and for compatibility with the quantum annealing speedup of Theorem D.1.

We first introduce the general quantum speedup of MLMC as described in [2], and then apply these results to our problem.

D.2.1 Quantum-accelerated multilevel Monte Carlo method

We begin by describing the following general result on quantum-accelerated multilevel Monte Carlo (QA-MLMC).

Lemma D.4 (Theorem 2 of [2])**.** *Let P denote a random variable, and let P_l (for $l \in \{0, 1, \ldots, L\}$) denote a sequence of random variables such that P_l approximates P at level l. Also define $P_{-1} = 0$. Let C_l be the cost of sampling from P_l, and let V_l be the variance of $P_l - P_{l-1}$. If there exist positive constants $\alpha, \beta = 2\hat{\beta}, \gamma$ such that $\alpha \geq \min(\hat{\beta}, \gamma)$ and*

- $|\mathbb{E}[P_l - P]| = O(2^{-\alpha l})$,
- $V_l = O(2^{-\beta l}) = O(2^{-2\hat{\beta} l})$, *and*
- $C_l = O(2^{\gamma l})$,

then for any $\epsilon < 1/e$ there is a quantum algorithm that estimates $\mathbb{E}[P]$ up to additive error ϵ with probability at least 0.99, and with cost

$$\begin{cases} O\Big(\epsilon^{-1}(\log 1/\epsilon)^{3/2}(\log\log 1/\epsilon)^2\Big), & \hat{\beta} > \gamma, \\ O\Big(\epsilon^{-1}(\log 1/\epsilon)^{7/2}(\log\log 1/\epsilon)^2\Big), & \hat{\beta} = \gamma, \\ O\Big(\epsilon^{-1-(\gamma-\hat{\beta})/\alpha}(\log 1/\epsilon)^{3/2}(\log\log 1/\epsilon)^2\Big), & \hat{\beta} < \gamma. \end{cases} \tag{D.16}$$

We apply this result to the payoff model of general stochastic differential equations (SDEs) as discussed in [2]. Consider an SDE

$$\mathrm{d}X_t = \mu(X_t, t)\,\mathrm{d}t + \sigma(X_t, t)\,\mathrm{d}W_t \tag{D.17}$$

for $t \in [0, T]$, where we assume μ and σ are Lipschitz continuous. Given an initial condition X_0 and an evolution time $T > 0$, we aim to compute

$$\mathbb{E}[\mathcal{P}(X_T)], \tag{D.18}$$

where $\mathcal{P}(X)$ is the so-called payoff function as a functional of X. In Lemma D.4, we denote $\mathcal{P}(X_T)$ as P, and the goal is to estimate $\mathbb{E}[P]$.

We also consider a numerical scheme that produces $\widehat{X}_k$ with time step size $h = T/n$. We say the scheme is of strong order r if for any $m \in \{1, 2\}$, there exists a constant C_m such that

$$\mathbb{E}\Big(\|\widehat{X}_n - X_T\|^m\Big) \leq C_m h^{rm}. \tag{D.19}$$

Note that it suffices to verify this condition for $m = 2$ since $(\mathbb{E}\|\widehat{X}_n - X_T\|)^2 \leq \mathbb{E}\|\widehat{X}_n - X_T\|^2$. We further assume the coefficients of the scheme are Lipschitz continuous. For the discretization $n_l = 2^l$ with step size $h = T/2^l$, we let P_l denote $\mathcal{P}(\widehat{X}_{n_l})$, an estimator of P.

Finally, we assume $\mathcal{P}(X)$ is L_P-Lipschitz continuous. Thus, we have satisfied the three assumptions of Proposition 2 of [2], which estimates the rates of $|\mathbb{E}[P_l - P]|, V_l, C_l$ in Lemma D.4. We state a simpler version as follows.

Lemma D.5 (Proposition 2 of [2])**.** *Given an SDE and a scheme of strong order r with Lipschitz continuous constants, and given a Lipschitz continuous payoff function $\mathcal{P}$, we have $\alpha = r$, $\beta = 2r$, and $\gamma = 1$.*

Note that while we relax the definition of a scheme of strong order r, our definition (D.19) is sufficient to prove Lemma D.5. More concretely, in the proof of Proposition 2 of [2], we have the following simplified inequalities:

$$|\mathbb{E}[P_l - P]| \leq \mathbb{E}|\mathcal{P}(\widehat{X}_n) - \mathcal{P}(X_T)| \leq L_P \mathbb{E}\|\widehat{X}_n - X_T\| \leq L_P C_1 h^r = O(2^{-rl}), \tag{D.20}$$

$$V_l \leq \mathbb{E}|\mathcal{P}(\widehat{X}_n) - \mathcal{P}(X_T)|^2 \leq L_P \mathbb{E}\|\widehat{X}_n - X_T\|^2 \leq L_P C_2 h^{2r} = O(2^{-2rl}), \tag{D.21}$$

$$C_l = O(n_l) = O(2^l), \tag{D.22}$$

and therefore $\alpha = r$, $\beta = 2r$, and $\gamma = 1$.

Finally, we can characterize the performance of QA-MLMC as follows.

Lemma D.6 (Theorem 3 of [2])**.** *Given an SDE and a scheme of strong order r with Lipschitz continuous constants, and given a Lipschitz continuous payoff function $\mathcal{P}$, QA-MLMC estimates $\mathbb{E}[\mathcal{P}]$ up to additive error ϵ with probability at least 0.99 with cost*

$$O\Big(\epsilon^{-1}(\log 1/\epsilon)^{3/2}(\log\log 1/\epsilon)^2\Big), \quad r > 1, \tag{D.23}$$

$$O\Big(\epsilon^{-1}(\log 1/\epsilon)^{7/2}(\log\log 1/\epsilon)^2\Big), \quad r = 1, \tag{D.24}$$

$$O\Big(\epsilon^{-1/r}(\log 1/\epsilon)^{3/2}(\log\log 1/\epsilon)^2\Big), \quad r < 1. \tag{D.25}$$

Note that we can amplify the success probability to $1 - \delta$ for arbitrarily small $\delta > 0$ using the powering lemma (Lemma D.2).

D.2.2 Quantum-accelerated multilevel Langevin method

We have described ULD and ULD-RMM in Algorithm 4 and Algorithm 5, respectively. We now apply these schemes to simulate the underdamped Langevin dynamics as the SDE. According to (D.19), ULD and ULD-RMM are schemes of order 1 and 1.5, respectively.

Let the payoff function $\mathcal{P}$ be g_i as defined in (B.28). Our goal is to estimate the mean of $\mathcal{P}(\widehat{X}_{n_l}) = g_i(\widehat{X}_{n_l})$ using several samples $\widehat{X}_{n_l}$ produced by ULD or ULD-RMM. If g is assumed to be L_g-Lipschitz as in Lemma C.2 of [15], we have a Lipschitz continuous payoff function $\mathcal{P}$ with $L_P = L_g$. Although $g_i = \exp\big(\frac{\|x\|^2}{\sigma_i^2(1+\alpha^{-1})}\big)$ is not Lipschitz, according to Section 4.3 of [15], we can truncate at large x and replace g_i by $h_i := \min\big\{g_i, \exp\big(\frac{(r_i^+)^2}{\sigma_i^2(1+\alpha^{-1})}\big)\big\}$ with

$$\alpha = \widetilde{O}\bigg(\frac{1}{\sqrt{d}\log(1/\epsilon)}\bigg), \tag{D.26}$$

$$r_i^+ = \mathbb{E}_{\rho_{i+1}}\|x\| + \Theta(\sigma_i\sqrt{(1+\alpha)\log(1/\epsilon)}), \tag{D.27}$$

to ensure $\frac{h_i}{\mathbb{E}_{\rho_i} g_i}$ is $O(\sigma_i^{-1})$ Lipschitz and $\big|\mathbb{E}_{\rho_i}(h_i - g_i)\big| < \epsilon$ by Lemmas C.7 and C.8 of [15]. For simplicity, as in Section 4.2 of [15], we regard g_i as a Lipschitz continuous payoff function in our main results.

Thus, using Lemma D.6 to estimate $\frac{Z_{i+1}}{Z_i} = \mathbb{E}_{\rho_i}(g_i)$, QA-MLMC using either ULD or ULD-RMM can reduce the ϵ-dependence of the number of steps to $\widetilde{O}(\epsilon^{-1})$. Then each step of ULD or ULD-RMM uses the value of $\nabla f(x)$ about $O(1)$ times as shown in Algorithm 4 and Algorithm 5.

Having described the implementations of quantum inexact ULD and ULD-RMM, we now state the quantum speedup for estimating normalizing constants using multilevel ULD and annealing, or multilevel ULD-RMM and annealing, as follows.

Theorem D.3 (Quantum speedup using multilevel ULD and annealing)**.** *Let Z be the normalizing constant in (1.3). Assume we are given access to the quantum gradient oracle (1.5). Then there is a quantum algorithm which outputs an estimate $\widetilde{Z}$ such that*

$$\Pr[(1-\epsilon)Z \le \widetilde{Z} \le (1+\epsilon)Z] \ge \frac{3}{4} \tag{D.28}$$

using

$$\widetilde{O}\Big(\frac{d^{3/2}\kappa^2}{\epsilon}\Big) \tag{D.29}$$

queries to the quantum gradient oracle.

Proof. As in Theorem D.1 and Theorem D.2, for $i \in [M]$ with M stages, we estimate $\mathbb{E}_{\rho_i}(g_i)$ with output $\widetilde{g}_i$ up to additive error $(\epsilon/2M)\mathbb{E}_{\rho_i}(g_i)$ with failure probability $1/4M$, which ensures $|\widetilde{Z} - Z| \le \epsilon Z$ with probability at least $\frac{3}{4}$.

According to Lemma D.6, each sample of ρ_i using multilevel ULD uses $\widetilde{O}(\frac{M\kappa^2\sqrt{d}}{\epsilon})$ queries to the quantum evaluation oracle (1.4) used in Algorithm 6 or Algorithm 7. For $M = \widetilde{O}(\sqrt{d})$ stages, the query complexity of estimating the normalizing constant is $\widetilde{O}(\frac{M^2\kappa^2\sqrt{d}}{\epsilon}) = \widetilde{O}(\frac{d^{3/2}\kappa^2}{\epsilon})$. □

Theorem D.4 (Quantum speedup using multilevel ULD-RMM and annealing)**.** *Let Z be the normalizing constant in (1.3). Assume we are given the access to the quantum gradient oracle (1.5). Then there is a quantum algorithm which outputs an estimate $\widetilde{Z}$ such that*

$$\Pr[(1-\epsilon)Z \le \widetilde{Z} \le (1+\epsilon)Z] \ge \frac{3}{4} \tag{D.30}$$

using

$$\widetilde{O}\Big(\frac{d^{7/6}\kappa^{7/6} + d^{4/3}\kappa}{\epsilon}\Big) \tag{D.31}$$

queries to the quantum gradient oracle.

Proof. As above, for $i \in [M]$ with M stages, we estimate $\mathbb{E}_{\rho_i}(g_i)$ with output $\widetilde{g}_i$ up to additive error $(\epsilon/2M)\mathbb{E}_{\rho_i}(g_i)$ with failure probability $1/4M$.

According to Lemma D.6 and Lemma B.2, each sample of ρ_i using multilevel ULD uses $\widetilde{O}(\frac{M(\kappa^{7/6}d^{1/6}+\kappa d^{1/3})}{\epsilon})$ queries to the quantum gradient oracle (1.5) or the quantum evaluation oracle (1.4) (with additional $\widetilde{O}(1)$ cost). For $M = \widetilde{O}(\sqrt{d})$ stages, the query complexity of estimating the normalizing constant is $\widetilde{O}(\frac{M^2(\kappa^{7/6}d^{1/6}+\kappa d^{1/3})}{\epsilon}) = \widetilde{O}(\frac{d^{7/6}\kappa^{7/6}+d^{4/3}\kappa}{\epsilon})$. □

E Proof of the Quantum Lower Bound

To prove Theorem 5.1, we use the following quantum query lower bound on the Hamming weight problem.

Proposition E.1 (Theorem 1.3 of [32])**.** *For $x = (x_1, \dots, x_n) \in \{0,1\}^n$, let $\|x\|_1 = \sum_{i=1}^n x_i$ be the Hamming weight of x. Furthermore, let ℓ, ℓ' be integers such that $0 \le \ell < \ell' \le n$. Define the partial boolean function $f_{\ell,\ell'}$ on $\{0,1\}^n$ as*

$$f_{\ell,\ell'}(x) = \begin{cases} 0 & \text{if } \|x\|_1 = \ell \\ 1 & \text{if } \|x\|_1 = \ell'. \end{cases} \tag{E.1}$$

Let $m \in \{\ell, \ell'\}$ be such that $|\frac{n}{2} - m|$ is maximized, and let $\Delta = \ell' - \ell$. Then given the quantum query oracle

$$O_x|i\rangle|b\rangle = |i, b \oplus x_i\rangle \quad \forall i \in [n], b \in \{0,1\}, \tag{E.2}$$

the quantum query complexity of computing the function $f_{\ell,\ell'}$ is $\Theta(\sqrt{n/\Delta} + \sqrt{m(n-m)}/\Delta)$.

Now we prove Theorem 5.1 using a construction motivated by Section 5 of [15].

Proof. We start from a basic function $f_0(x) = \frac{\|x\|^2}{2}$. The partition function of f_0 is

$$\int_{\mathbb{R}^k} e^{-f_0(x)} \mathrm{d}x = (2\pi)^{k/2}. \tag{E.3}$$

We then construct n cells in $\mathbb{R}^d$. Without loss of generality we assume that $n^{1/k}$ is an integer, and let $l := 1/(\sqrt{k} n^{1/k})$. We partition $[-1/\sqrt{k}, 1/\sqrt{k}]$ into $n^{1/k}$ intervals, each having length $2l$. Let I_i denote the ith interval, where $i \in [n^{1/k}]$. Each cell will thus be represented as a k-tuple $(i_1, \ldots, i_k) \in \{1, 2, \ldots, n^{1/k}\}^k$ corresponding to $I_{i_1} \times \cdots \times I_{i_k} \subset \mathbb{R}^k$.

Next, each cell $\tau = (i_1, \ldots, i_k)$ with center denoted v_τ is assigned one of two types (as detailed below), and we let

$$f(x) = \begin{cases} f_0(x) & \text{if cell } \tau \text{ is of type 1} \\ f_0(x) + c_\tau q(\frac{1}{l}(x - v_\tau)) & \text{if cell } \tau \text{ is of type 2.} \end{cases} \tag{E.4}$$

The function q and the normalizing constant c_τ are carefully chosen, following Lemma D.1 in [15], such that

- $f(x)$ is 1.5-smooth and 0.5-strongly convex; and
- the partition function $Z_f = \int_{\mathbb{R}^k} e^{-f(x)} \mathrm{d}x = (2\pi)^{k/2} - C \cdot \frac{n_2}{n}$, where n_2 is the number of type-2 cells, and C is at least $\Omega(l^2)$.

With these properties, we consider two functions as follows. We choose δ such that $\epsilon = \Theta(\delta^{1+4/k})$. One of the functions has a $1/2 + \delta$ fraction of its cells of type 1 (and a $1/2 - \delta$ fraction of type 2). The other function has a $1/2 - \delta$ fraction of its cells of type 1 (and a $1/2 + \delta$ fraction of type 2). Note that one query to the quantum evaluation oracle (1.4) can be implemented using one quantum query to the binary information indicating the type of the corresponding cell. In addition, by Proposition E.1 with $\ell = (1 - \delta)n/2$ and $\ell' = (1 + \delta)n/2$, it takes $\Omega(1/\delta)$ quantum queries to distinguish whether there are $(1 + \delta)n/2$ or $(1 - \delta)n/2$ cells of type 1. Since $C = \Omega(l^2)$, the partition functions of the two functions differ by a multiplicative factor of at least $1 + \Omega(l^2\delta)$, where $l = \Theta(n^{-1/k}) = \Theta(\delta^{2/k})$, and hence $l^2\delta = \Theta(\delta^{1+4/k}) = \Theta(\epsilon)$. The quantum query complexity is therefore $\Omega(1/\delta) = \Omega(\epsilon^{-\frac{1}{1+4/k}})$ as claimed. □