The Pennsylvania State University
The J.Jeffrey and Ann Marie Fox Graduate School

QUANTUM ALGORITHMS FOR MARKOV CHAIN METHODS IN MACHINE LEARNING AND OPTIMIZATION

A Dissertation in
Computer Science and Engineering
by
Guneykan Ozgul

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2025

The dissertation of Guneykan Ozgul was reviewed and approved by the following:

Mehrdad Mahdavi
Associate Professor of Computer Science and Engineering
Dissertation Co-Advisor
Chair of Committee

Chunhao Wang
Assistant Professor of Computer Science and Engineering
Dissertation Co-Advisor

Xiantao Li
Professor of Mathematics

Sean Hallgren
Professor of Computer Science and Engineering

Chitaranjan Das
Distinguished Professor of Computer Science and Engineering
Head of the Graduate Programs in Computer Science and Engineering

Abstract

Markov chains are fundamental tools for solving sampling and optimization problems that arise across machine learning, statistical physics, computational geometry, finance, and other fields. However, classical Markov chain methods often suffer from slow convergence, particularly in high-dimensional or multimodal landscapes, limiting their scalability in modern applications. In large-scale learning tasks, even a single step of certain chains may be computationally expensive due to the need to process large datasets. Moreover, in optimization and search problems, multiple independent runs of the Markov chain are typically required to locate the target object, further compounding the total runtime.

Quantum computers have the potential to accelerate such methods, either by improving the mixing times through quantum walks or by reducing the number of required Markov chain runs via quantum amplitude amplification. Yet, many existing approaches rely on restrictive assumptions, such as reversibility of the underlying chain, log-concavity of the target distribution, or easy access to the gradient of the log-density. Furthermore, speedups are often at most quadratic and demonstrated only against naive baselines like hypercube-walk.

This dissertation overcomes these limitations by advancing quantum algorithmic techniques along three interconnected directions:

First, we extend quantum walk frameworks to sample efficiently from Gibbs distributions for non-logconcave potentials using non-reversible dynamics, such as the unadjusted Langevin algorithm. We tailored a new annealing schedule for non-logconcave distributions and developed a robust perturbation analysis to overcome non-reversibility to establish polynomial quantum speedups over classical non-reversible samplers, which are often more practical to implement than their reversible counterparts. These methods also integrate stochastic gradient oracles, making them well-suited for large-scale applications. We further apply these techniques to partition function estimation in non-logconcave settings, demonstrating their practical utility.

Second, we address high per-step computational cost of classical samplers in large-scale learning, where gradient evaluations can be expensive. By combining quantum mean estimation with classical variance reduction and advanced samplers such as Langevin and Hamiltonian Monte Carlo, we design quantum-enhanced algorithms that achieve near-quadratic reductions in gradient query complexity in the finite-sum setting. In the zeroth-order regime—where only noisy function evaluations are available—we construct robust quantum gradient estimators that enable efficient sampling and optimization under assumptions common in machine learning. These results are supported by rigorous theoretical analyses under both finite-sum and zeroth-order oracle models, providing

improved convergence guarantees for both sampling and optimization. In particular, we use these methods to solve structured non-convex optimization problems, which are prevalent in modern machine learning applications such as empirical risk minimization.

Finally, we turn to investigate quantum algorithms for certain optimization problems, where classical Markov chain methods such as simulated annealing or tempering often encounter exponential-time bottlenecks. Building on Hastings' short-path framework[Quantum 2, 78 (2018)], we develop a generalized quantum strategy that achieves super-quadratic speedups over a classical algorithm that uses sampling from stationary distribution of a Markov chain, going beyond the quantum amplitude amplification, by exploiting structural features of the underlying Markov chain and the cost function. This approach leads to provable improvements over both classical Gibbs sampling and existing quantum methods on hard combinatorial instances, including Max-Bisection, Max Independent Set, the Ising Model, and the Sherrington Kirkpatrick Model—supported by both theoretical analysis and empirical evidence.

Table of Contents

List of Figures

List of Tables

Acknowledgments

This dissertation grew out of my early curiosity about the potential of quantum algorithms to tackle optimization problems in machine learning. Over time, this curiosity deepened and took shape as my understanding and interests evolved. I am grateful to my advisors, Prof. Mehrdad Mahdavi and Prof. Chunhao Wang, for their patience and guidance during this process, especially during times when I struggled to navigate the interdisciplinary challenges of situating this work at the intersection of quantum computing, machine learning, and optimization. I am also thankful to Prof. Xiantao Li for sharing his expertise and making this work possible. I also thank my other committee member, Prof. Sean Hallgren, for his valuable feedback.

I owe heartfelt thanks to my girlfriend, Ege, whose constant support carried me through the demands and difficulties of graduate school and life in general. She has always believed in my unyielding determination to overcome challenges, and without her belief and encouragement, I would not have found the motivation to persevere and stay true to that attitude. I am also grateful to my friends and family, especially my brother Deniz, for his presence and support at every stage of my life.

Finally, I am thankful to my collaborators at JPMorgan Chase for their support during my summer internship, which made the final chapter of this dissertation possible.

The research described in this dissertation was supported in part by the seed grant from the Institute of Computational and Data Science (ICDS) and National Science Foundation grant CCF-2238766 (CAREER) to Chunhao Wang. The findings and conclusions presented do not necessarily reflect the views of the funding agency.

Chapter 1
Introduction

1.1 Overview

Sampling and optimization problems are foundational to a wide range of disciplines, including machine learning, optimization, computational geometry, and statistical physics. In probabilistic machine learning, sampling plays an important role in Bayesian inference, as it facilitates posterior estimation and quantifies uncertainty in model predictions [WT11, WFS15, DM18, RSBG21]. In non-convex optimization, sampling allows for the exploration of complex energy landscapes and helps avoid local minima, facilitating progress in tasks such as resource allocation, scheduling, and hyperparameter tuning [BLNR15, ZLC17, CDT20]. Similarly techniques such as simulated annealing plays an important role in practice for solving combinatorial optimization problems [KGV83]. In convex geometry, it helps in approximating volumes and studying high-dimensional structures [LV06, CV18]. In statistical mechanics, sampling is used to analyze the thermodynamic properties of materials by exploring configurations of particle systems [Cha87, FS02]. Many of these problems can be formalized in terms of sampling from *Gibbs-Boltzmann* distributions of the form

$$\pi(x) \propto \exp\left(-\frac{f(x)}{T}\right), \tag{1.1}$$

where the function $f : \mathcal{X} \to \mathbb{R}$ encodes an energy landscape or objective function, and $\mathcal{X}$ may be either continuous (e.g., $\mathbb{R}^d$) or discrete (e.g., subsets of graph vertices) and T is the temperature parameter. In this framework, two core computational tasks arise: (i) sampling from the distribution π, and (ii) finding configurations $x \in \mathcal{X}$ that approximately or exactly minimize f. The latter task can be viewed as a limiting case of the former as $T \to 0$.

Classical approaches to both sampling and optimization frequently rely on Markov chain methods when more direct methods such as rejection sampling or exhaustive search are not feasible. Markov chains are a class of stochastic processes that provide a mathematical framework for modeling systems that evolve in time and possess the memoryless property: the future state of the system depends only on the present state, not on the sequence of events that preceded it. This simple yet powerful property makes Markov chains an essential tool in probability theory and its applications. Their long-term behavior is of central interest: under suitable conditions, such as irreducibility and aperiodicity, a Markov chain converges to a unique stationary distribution π, allowing the drawing of samples from π. For example, Langevin Monte Carlo (LMC) or Hamiltonian Monte Carlo (HMC) chains are used to simulate ergodic processes that converge to the Gibbs distribution for continuous domains [WT11, BGJM11]. In discrete domains, techniques such as Glauber dynamics, Metropolis–Hastings walks, and simulated annealing construct ergodic chains over combinatorial spaces such as the set of independent sets in a graph. The effectiveness of these algorithms depends on their convergence rate or mixing time, which in turn is governed by geometric and spectral properties of the chain, such as the *spectral gap* defined as the difference between the largest eigenvalues of its transition operator.

Quantum computing offers a fundamentally different model of computation from its classical counterpart, leveraging unitary evolution and quantum superposition to enable algorithmic primitives that have no classical analog. Originally proposed by Feynman [Fey82] as a framework for simulating quantum systems, quantum computation has since proved powerful for a variety of other tasks. An influential result by Shor [Sho97] demonstrated that quantum algorithms can solve the integer factorization problem in polynomial time via the quantum Fourier transform, in contrast to the best known classical algorithms which require superpolynomial time. Shor also provided quantum algorithms for discrete logarithm, and period-finding problem which are instances of *hidden subgroup problem (HSP)* over finite Abelian groups. The later extensions of HSP to infinite Abelian groups (the real numbers) provided solutions for problems in algebraic number theory such as solving Pell's equation [Hal07, Joz03]. Another key result for an unrelated task is Grover's algorithm [Gro96a], which provides a quadratic speedup for unstructured search problems. Grover's search algorithm is an instance of a more generic procedure called *quantum amplitude amplification* [BHMT02] that finds a marked element x^* using $\mathcal{O}\left(\frac{1}{\sqrt{p_{x^*}}}\right)$ queries to an oracle that prepares the quantum state $|\psi\rangle = \sum_{x\in\mathcal{X}} \sqrt{p_x}\,|x\rangle$.

These breakthroughs have spurred the development of several new quantum algorithmic primitives, including quantum walks [AAKV01, Sze04], quantum gradient estimation [Jor05], quantum linear system solvers [HHL09a], and quantum singular value transformation [GSLW19]. Despite growing body of such algorithmic primitives, exponential quantum speedups remain rare and typically arise only in contrived problems with highly structured inputs [AA14]. Even when such a speedup is demonstrated, it often lacks practical significance. For instance, Childs et al. [CCD$^+$03] constructed an artificial graph known as the *glued trees* and showed that a continuous-time quantum walk could reach a marked vertex exponentially faster than any classical algorithm in oracular (i.e., black box) setting. While this result is theoretically striking, the underlying problem does not correspond to a task of direct practical interest. Moreover, establishing exponential separations between quantum and classical algorithms might be difficult even in the black box setting. Not only must one identify a problem with the right structure to expose a quantum advantage, but one must also rigorously rule out the possibility of classical dequantization—where a classical algorithm mimics the quantum behavior efficiently to achieve a similar query complexity (See [Aar22] for more discussion). For example, it was shown in [FGG02,Rei04] that optimizing Hemming weight potential with a spike takes the polynomial time for quantum adiabatic algorithm in oracular setting whereas classical simulated annealing takes exponential time. However, later [CH16] showed that a classical algorithm called *quantum simulated annealing* can also solve this problem in polynomial time. Similar dequantization has also been achieved for quantum recommendation systems by [Tan19] for the quantum algorithm proposed in [KP17].

As a result, both the discovery of problems for which quantum algorithms provide exponential speedups and the formal proof of an exponential separation remain significant theoretical challenges. In contrast, polynomial quantum speedups are more common and often apply to problems of broader computational relevance. These speedups typically arise from more general-purpose quantum primitives that can be integrated into existing algorithmic frameworks. In the context of optimization, for example, techniques such as quantum walks [LZ24] and quantum gradient estimation [Jor05, GAW19, CCLW20, SZ23] have led to improvements over classical baselines in a variety of settings. Although these gains are more modest, they can still play a significant role in enabling practical quantum applications for large scale problems.

Quantum algorithms have also been studied in the context of *sampling* from the stationary distributions of a reversible Markov chain. The goal in this setting is to

efficiently prepare the quantum state called *qsample*:

$$|\pi\rangle = \sum_{x\in\mathcal{X}} \sqrt{\pi(x)}\,|x\rangle\,, \tag{1.2}$$

from which a measurement yields a classical sample distributed according to a target distribution π. This is also known as *quantum sampling problem*, and a general solution to this problem would imply $\mathsf{SZK} \subseteq \mathsf{BQP}$ [ATS03] and the relationship between SZK and BQP remains an open question in complexity theory. Nevertheless, in certain structured settings, quantum algorithms have been shown to achieve polynomial speedups over classical mixing procedures [WA08], particularly when the sampling task involves a sequence of slowly evolving Markov chains.

For the classical Markov chains, the classical mixing time scales as $\mathcal{O}\left(\frac{1}{\delta}\log(1/\pi_{\min})\right)$, where δ denotes the spectral gap of the transition matrix of the chain, and $\pi_{\min}$ is the minimum stationary probability. It is also possible to prepare $|\pi\rangle$ by combining quantum phase estimation and amplitude amplification and this quantum algorithm has runtime of $\mathcal{O}\left(\frac{1}{\sqrt{\delta\pi_{\min}}}\right)$ [MNRS07]. This raises a natural question: can one generically obtain a quantum runtime of $\mathcal{O}\left(\frac{1}{\sqrt{\delta}}\log(1/\pi_{\min})\right)$, having similar classical dependence on $\log(1/\pi_{\min})$ while achieving a quadratic speedup in the spectral gap? In a special case, Richter [Ric07] achieved this target runtime for *uniform* mixing by performing repeated quantum measurements at random time intervals. However, this technique does not extend to general, non-uniform stationary distributions. As a result, a universal quadratic speedup for Markov chain mixing—applicable to arbitrary reversible chains—remains an open challenge.

In the context of sampling from continuous Gibbs distributions, [CLL$^+$22] obtained a quantum speedup over classical Metropolis adjusted Langevin algorithm (MALA) with respect to dimension (d) and condition number(κ) by using slowly varying Markov chains originally proposed in [WA08]. Their construction allows one to exploit quantum walks that have quadratically larger phase gap than the spectral gap of classical MALA. This faster sampling algorithm also allows faster partition function estimation when combined with quantum mean estimation. However, these results rely on idealized assumptions, such as convexity of the potential, reversibility of the Markov chain, or access to warm initial states (a state with large overlap with $|\pi\rangle$). Although generalizing log-concavity to non-logconcave distributions is possible by using smoothness and isoperimetry, the reversibility assumption causes a more fundamental problem. A Markov chain is said to be time reversible (or detailed balanced) if it holds for all $x, y \in \mathcal{X}$,

$\pi(x)P(x,y) = \pi(y)P(y,x)$. In quantum computing terms, reversibility allows to convert the Markov chain kernel to an Hermitian operator

$$D(P) = \pi^{1/2} P \pi^{-1/2}. \tag{1.3}$$

This operator is called the *discriminant operator* and it is symmetric and $|\pi\rangle$ is the ground state of $-D$ with unique eigenvalue 1. The spectral properties of D is investigated in [Sze04] in more detail. Therefore, one can prepare a reflection operator around $|\pi\rangle$ using quantum phase estimation and apply quantum amplitude amplification. However, this operator is no longer Hermitian if P is not reversible and its spectrum is not straightforward to analyze. The previous works in [Sze04, WA08] have not made significant efforts to relax the reversibility assumption mainly because in practice a non-reversible Markov chain can be made reversible by adding a filter called Metropolis rejection step. However, this reversibilization is not always trivial. The reversibilization step often requires to evaluate f very accurately which is very costly in machine learning applications when f is given as a finite-sum. Moreover, reversible chains are often slower than their non-reversible counterparts. Therefore, a quantum speedup for reversible chains does not imply speedup over best classical algorithm because it is often the case that there is a classical non-reversible Markov chain that is faster than the quantum algorithm. The requirement of a warm start can be equally restrictive—generating such a state is often as hard as preparing $|\pi\rangle$ itself. [WA08] mitigated this obstacle by introducing an annealing schedule that gradually reduces the temperature, avoiding the exponential slow-down associated with cold starts. Unfortunately, the schedule must be carefully tailored to each specific task, and a *universal speed-up* for general non-log-concave targets via non-reversible remains open and we will return to this problem in Chapter 2.

Quantum algorithms for sampling from continuous distributions may also benefit from quantum gradient estimation or mean estimation techniques, as many state-of-the-art classical samplers—such as the Metropolis-Adjusted Langevin Algorithm (MALA)—rely on gradient information from samples at each iteration. These techniques have been well-studied in the context of optimization [Jor05, GAW19, CCLW20, ZZF+24], where they yield asymptotic speedups under various assumptions such as smoothness, convexity and/or Lipschitzness of the objective function. However, despite this close connection, these techniques have not yet been systematically explored for sampling applications. The zeroth-order quantum algorithm proposed in [CLL+22] achieves the same asymptotic runtime as its classical counterpart, despite relying only on function evaluations to

approximate gradients. A key challenge to reduce the run-time is that existing quantum gradient estimation methods typically assume access to high-precision function evaluations. In practice—particularly in machine learning and statistical inference—function values are often noisy due to stochasticity, subsampling, or model approximation. As a result, naively applying these techniques in such settings may not yield improved runtimes. Overcoming this challenge will require the design of robust quantum gradient estimators that can tolerate imprecise or noisy oracles while still delivering provable speedups in sampling tasks. Our focus in Chapter 3 is to address these challenges in more detail and design quantum gradient estimation algorithms that would provide accelerated rates for certain sampling tasks.

One fundamental limitation of these existing quantum algorithms is that the speedups are typically quadratic at most by construction. Recent research has highlighted significant barriers to achieving quadratic quantum speedups in practice, owing to constant-factor slowdowns relative to classical hardware. These slowdowns arise from a variety of sources, including lower clock speeds of quantum processors, the substantial overhead introduced by quantum error correction, and the limited potential for parallelization in many quantum algorithms. As a result, realistic estimates of the resources required to execute quantum algorithms at scale often predict runtimes that exceed several days, even when the asymptotic complexity suggests a polynomial advantage. Crucially, the feasibility of realizing a practical quantum advantage depends not only on the existence of a polynomial speedup but also on its magnitude. The larger the degree of the speedup, the more robust it is to constant-factor overheads. For example, the resource analysis conducted in [BMN$^+$21] indicates that a quartic speedup, when all architectural and algorithmic overheads are taken into account, could reduce the required runtime from multiple days to just a few hours—making it significantly more viable in real-world settings.

Recent works by [Has18c, DPCB23] managed to obtain super-quadratic speedups over brute force search algorithms for combinatorial optimization problems under more assumptions on the spectrum of the cost function. Although the speedups over Grover's algorithm are very small in terms of practicality, their frameworks provide promising directions to obtain such *super-quadratic speedups*. The idea to obtain such speedups is to consider adiabatic Hamiltonian,

$$H_b = \frac{-X}{n} + bg_\eta\left(\frac{H}{|E^*|}\right) \tag{1.4}$$

where $X = \sum_{i=1}^{n} X_i$ is the Pauli mixer, H is a target Hamiltonian encoding the optimization problem, E^* is the ground state energy of H and g_η is a filter function that truncates high energy states. The spectral analysis of this Hamiltonian shows that there exists a $b^* = \Omega(1)$ whose ground state can be prepared efficiently and the ground state has better overlap than the uniform quantum state. This algorithm is called short-path algorithm due to its avoiding nature from the exponentially vanishing gaps.

However, a remaining issue is that these quantum algorithms provide accelerated rates compared to Grover's search, which is not necessarily optimal even classically. For example, when the objective function has an exploitable structure, the optimal classical strategy may involve local search or tailored heuristics. In such cases, Grover's algorithm provides only a quadratic speedup over exhaustive search, but may not offer any advantage over more efficient structure-aware algorithms. Hence, realizing meaningful quantum advantages requires designing quantum algorithms that respect and exploit the structure present in the problem, rather than treating all inputs uniformly.

A good candidate direction to improve upon these sub-optimal algorithms is to develop super-quadratic speedups over classical Markov chain search algorithms that are known to perform better than brute-force search algorithms due to their ability to exploit local structures. A Markov Chain Search algorithm simply runs the chain to draw samples from the stationary distribution π and keeps track of the running minimum of the samples in terms of the cost function H. This minimum (and the corresponding sample) serves as an estimate of the global minimum (and minimizer) of H on $\mathcal{X}$. Let π^* denote the total probability that a sample from π is a global minimizer of H. It follows that after $\mathcal{O}\left((\pi^*)^{-1}\log(\epsilon^{-1})\right)$ samples from π, a global minimizer is encountered with probability at least $1-\epsilon$. In Chapter 4, we provide quantum algorithms with runtime $\mathcal{O}\left((\pi^*)^{-\frac{1}{2}+c}\right)$ where $c > 0$ is a positive function by generalizing the short path framework to Markov Chain Search. Our analysis also provides evidence that the short path algorithm is more powerful than a quantum algorithm that provides a quadratic speedup over classical Gibbs samplers.

In summary, there has been a major progress in obtaining quantum algorithmic frameworks for search and optimization problems. Although these primitives also show promising results for the future of quantum computing in the context of Markov chain algorithms to obtain provable speedups, their applicability is limited due to practical concerns. This dissertation generalizes these foundational quantum algorithmic frameworks to settings that fall outside their standard assumptions to address the open problems listed above. It generalizes quantum walks to operate on non-reversible and

non-logconcave dynamics, develops quantum sampling techniques for distributions with weaker regularity, and adapts quantum gradient estimation to noisy or finite-sum oracle models. In the combinatorial domain, it introduces a generalization of the short-path framework that surpasses the limits of amplitude amplification by explicitly leveraging structural properties of classical Markov chains to obtain super-quadratic speedup over classical algorithms that repeatedly sample from local Markov chains such as Glauber dynamics. These contributions yield provable quantum speedups in terms of sample complexity, mixing time, or oracle efficiency for optimization and sampling tasks that remain challenging for both classical and quantum algorithms.

1.2 Summary of Contributions

In Chapter 2, we focus on quantum algorithms for sampling from continuous distributions with particular focus on non-reversible Markov chains. We consider distributions $\pi(x) \propto \exp(-f(x))$ where f may be nonconvex, but it satisfies weaker regularity conditions, such as dissipativity or smoothness. Our contributions in first chapter can be summarized as follows.

- **Quantum MALA for Non-logconcave Distributions:** We analyze the mixing time of the quantum MALA algorithm for non-logconcave distributions, extending the work done in [CLL+22]. Using the conductance analysis in [ZXG21], we characterized the phase gap of the corresponding quantum walk operator. Next, we showed that by using isoperimetric inequalities, the length of the annealing schedule is $\tilde{O}(\sqrt{d})$ similar to the logconcave case.

- **Quantum Speedup via ULA:** We propose quantum ULA algorithm and analyze its runtime using a novel perturbation analysis with respect to quantum MALA to show that quantum computers can provide speedups even for non-reversible Markov chains. Since quantum MALA is time-reversible and asymptotically unbiased, it converges to the target Gibbs distribution, allowing us to express our algorithm's error with respect to the Gibbs distribution. Our results imply polynomial speedups compared to the mixing time of classical ULA [VW19].

- **Stochastic Quantum Sampling:** We further incorporate stochastic gradient oracle to make the implementation of quantum walk efficient and provide the mixing time of our stochastic quantum sampling algorithm. In addition to the error due to the lack of Metropolis-Hastings filter, the stochastic algorithm introduces

additional errors because of the noisy gradients. We use concentration techniques to show that even with stochastic gradients, the quantum algorithm gives the correct distribution with high probability. Our algorithms improve the gradient complexity of the classical SGLD [ZXG21].

- **Partition Function Estimation:** Finally, we combine our sampling algorithms with recently developed efficient quantum product estimator [CH23] and proposed algorithms for approximating the partition function for non-logconcave distributions.

In Chapter 3, we address the high per-step cost of sampling or optimization algorithms. In large-scale machine learning problems, the objective f often arises as a finite sum over data points, and computing the full gradient is expensive. In other settings, such as black-box optimization or bandit feedback models, gradient information is not directly available. To address both regimes, we develop quantum-enhanced estimators for gradients using quantum mean estimation and Jordan's algorithm respectively. In the finite-sum setting, we integrate these estimators with classical variance-reduction techniques and show that the improved gradient query complexity leads to overall polynomial gains in n, the number of data points. For the zeroth-order setting, we develop a robust version of quantum gradient estimation algorithm that tolerates bounded evaluation noise and prove convergence guarantees for both Hamiltonian and Langevin-type dynamics. Our analysis demonstrates that nearly quadratic improvements in query complexity are achievable under mild smoothness and noise assumptions. In particular, our contributions include:

- **Speedups for Finite Sum Potentials**: We propose novel quantum algorithms to sample from Gibbs distribution for finite-sum potentials implemented via quantum variance reduction techniques. We prove that our algorithms improve the dependency on n(number of terms in the sum) compared to classical state-of-the-art algorithms such as stochastic HMC and LMC [ZG21,KS22] to approximately sample from strongly convex and non-convex potentials, respectively.

- **Quantum Speedups for Gradient Estimation via Stochastic Evaluation Oracle**: In the zeroth-order setting, where only stochastic evaluations of the potential function are available, we develop new quantum gradient estimation algorithms under various smoothness assumptions. Our algorithm provides quadratic speedup when the potential function is smooth, reducing the evaluation queries from $\tilde{\mathcal{O}}(\frac{d^2\sigma^2}{\epsilon^2})$ to $\tilde{\mathcal{O}}(\frac{d\sigma}{\epsilon})$ to compute the gradient in d dimension up to ϵ accuracy where σ^2 is the variance of the noise. Furthermore, when the stochastic functions are also smooth

with high probability, we manage to shave off an additional $d^{1/2}$ term . This is achieved by combining quantum mean estimation with Jordan's quantum gradient estimation in a robust manner. Our gradient estimation algorithms could be useful as independent tools, especially in zeroth-order stochastic optimization.

- **Speedups for Zeroth-Order Sampling**: Next, we combine our new quantum gradient estimation algorithm with the gradient based sampling algorithms and show that the final algorithm uses fewer number of queries to evaluation oracle than the best known classical samplers under the same assumptions in [RSBG21].

- **Application to Non-Convex Optimization**: Finally, we extend our quantum sampling methods to optimize non-convex functions with specific structural properties, demonstrating that faster sampling translates to provable speedups in complex optimization tasks. In particular, we show that we can optimize non-smooth and approximately convex functions, i.e. a function that is uniformly close to a strongly convex function, using fewer stochastic evaluation queries than the best known classical algorithms in terms of dimension dependency.

The final part of the dissertation turns to *combinatorial optimization problems*, where the domain $\mathcal{X}$ is finite and structured (e.g., subsets of vertices in a graph), and the function f encodes a constraint or objective (e.g., Max-Bisection, Max Independent Set, the Ising Model, and the Sherrington Kirkpatrick Model). Our primary contribution in Chapter 4 is to generalize the short path framework and to identify some conditions under which this generalized algorithm obtains the runtime $\mathcal{O}(\left(\pi^*\right)^{-(0.5-c)}\right)$ for constant $c > 0$. Our generalization is simple to describe and is motivated by previous studies of the quantization of discrete time Markov Chains. The short path framework is based on computing the ground states of the a Hamiltonian $H_b = -X/n + g_\eta(H/|E^*|)$ parameterized by $b \in \mathbb{R}^+$ where H is our cost Hamiltonian, E^* the energy of its ground state, and $g_\eta = \min(0, (x + 1 - \eta)/\eta)$ is a clipping function. We replace $-X/n$ in the above Hamiltonian with $-D(P)$, where $D_p = \text{diag}(\pi)^{1/2} P \text{diag}(\pi)^{-1/2}$ is the *discriminant matrix* corresponding to the Markov Chain. The remainder of the algorithm takes a similar form to [DPCB23], however, extra care is needed to construct efficient algorithms for ground state preparation with the redefined Hamiltonians. The three technical questions concern the isoperimetry of the stationary state π, the concentration of the cost function on states sampled from π, and the expected increase in cost when P is applied to a solution. Equipped with our main result, we identify examples of various settings where the conditions for speedup are satisfied. We focus on two types of stationary

states, the uniform distribution over a constrained space, or Gibbs distributions. The applications are the following:

1. **Optimization with Fixed Hamming Weight:** We first consider optimization problems where only strings of a fixed Hamming weight are considered to be in-constraint. Specifically we consider the following versions of Maximum Cut on hamming Weight k spaces.

$$\mathcal{C}^*_{\frac{n}{2}} := \min_{x\in\{-1,1\}^n} \left\{ -\frac{1}{2}\sum_{i<j} e_{ij}(1-x_ix_j) : |x| = \frac{n}{2} \right\}. \qquad \text{(MaxBisection)}$$

$$\mathcal{C}^*_k := \min_{x\in\{-1,1\}^n} \left\{ -\frac{1}{2}\sum_{i<j} e_{ij}(1-x_ix_j) : |x| = k \right\}. \qquad \text{(MaxCut-Hamming)}$$

   The corresponding Markov Chain considered is the transposition walk, each step of which swaps a uniformly random pair of indices with opposite spins. Our techniques yield runtimes of $\binom{n}{k}^{0.5-c(n)}$ where $c(n)$ is constant for $k = \Theta(n)$ and decays with n otherwise. For $k = \Theta(n)$, the speedup is super-quadratic over uniformly searching the constrained space. We note that the original short-path framework with penalty terms added to ensure the hamming weight constraints cannot obtain a super-quadratic speedup over uniformly searching the constrained space, due to the technical conditions for speedup becoming impossible to satisfy if a sufficiently large penalty term is added.

2. **Glauber Dynamics:** We consider next the setting where the stationary state used for Markov Chain search is the Gibbs distribution at some inverse temperature β corresponding to the cost function. The Glauber Dynamics offers a convenient and well-studied Markov Chain that samples from these distributions. Markov Chain search offers two types of advantages over brute force search: the Gibbs distribution at any $\beta \in \mathbb{R}$ places non-zero probability only on in-constraint states, which in many cases boosts the probability of the optimum compared to distributions with support on all states. Secondly, Gibbs distributions at $\beta > 0$ always have higher overlap on the optimum than 2^{-n}. Our results are for three different classes of cost Hamiltonians, and apply to any inverse temperature β where the corresponding Glauber Dynamics has been shown to mix in polynomial time.

   (a) For the Maximum Independent Set problem on regular graphs, we obtain super-quadratic speedups of degree $0.5 - c$ for constant c.

(b) For the Ising Model on regular graphs, we obtain super-quadratic speedups of degree $0.5 - c$ for constant c.

(c) For the Sherrington-Kirkpatrick Model, we obtain super-quadratic speedups of degree $0.5 - c(n)$ where $c(n) = o_n(1)$.

Finally, we provide evidence that it is not generically possible to construct classical algorithms that are at most quadratically slower than short-path algorithms. We first show that for maximum independent set on regular graphs of sufficiently high degree, Markov Chain Search with the Glauber Dynamics can be faster than the best known classical algorithm for all graphs (with a runtime of 1.1996^n ([XN17]). In this case, the short path algorithm is super-quadratically faster than the best classical algorithm (that is based on very non-trivial backtracking analysis). Then, the speedup for MIS is valid for any β where polynomial time Gibbs sampling is possible. Therefore, the generalized short-path algorithm is super-quadratically faster than *any* classical algorithm based on polynomial time Gibbs sampling (whether or not that algorithm uses the Glauber dynamics).

The material in this dissertation will be mostly based on the works in [OLMW24, OLMW25, CHO+24].

1.3 Preliminaries

1.3.1 Notation

Bold symbols, such as $\mathbf{x}$ and $\mathbf{y}$, are used to represent vectors, with $\|\cdot\|$ indicating the Euclidean or operator norm depending on the context. Given two scalars a and b, we use $a \wedge b$ to denote $\min\{a, b\}$ and use $a \vee b$ to denote $\max\{a, b\}$. We use $\mathcal{B}_d(c, r)$ to denote the d dimensional ball centered at c with radius r and $G_d^l(c)$ to denote the d dimensional grid centered at point c with side length l. We occasionally use G_d^l when the center of the grid is clear from the context.

For Markov chains, we use the notation $P(x, \cdot)$ to denote the transition probability distribution for point $x \in \mathcal{X}$, whereas we use $P(x, y)$ or p_{xy} to denote the probability of transitioning from point x to y. For a distribution $p(x)$ and a function $q(x)$, the notation $p(x) \propto q(x)$ means p is proportional to q up to a normalization factor.

In the quantum framework, a classical probability distribution p over $\mathcal{X}$ can be represented by the quantum state $\sum_{x \in \mathcal{X}} \sqrt{p(x)}\,|x\rangle$. When measuring this state, the resulting outcomes are governed by the probability distribution p. The ket notation

$|\nu\rangle$ is sometimes referred to the coherent quantum state corresponding to probability distribution ν and is not explicitly stated when it is clear from the context.

We might occasionally deviate from the notation above for some quantities to keep the notation similar to the previous works. Such quantities will be defined explicitly in the corresponding chapters.

Definition 1.3.1 (Order Estimates)**.** We define $\mathcal{O}(\cdot)$ as

$$f(x) = \mathcal{O}(g(x)) \iff \exists \ell \in \mathbb{R}, \alpha \in \mathbb{R}_+, \text{ such that } f(x) \leq \alpha g(x) \quad \forall x > \ell.$$

Similarly,

$$f(x) = o(g(x)) \iff \exists \ell \in \mathbb{R}, \text{ such that } f(x) < \alpha g(x) \quad \forall x > \ell \text{ and } \forall \alpha \in \mathbb{R}_+.$$

We write $f(x) = \Omega(g(x)) \iff g(x) = \mathcal{O}(f(x))$. If there exists positive constants α_1 and α_2 such that

$$\alpha_1 g(x) \leq f(x) \leq \alpha_2 g(x) \quad \forall x > 0,$$

then we write $f(x) = \Theta(g(x))$. We also define $\widetilde{\mathcal{O}}(f(x)) = \mathcal{O}(f(x) \cdot \text{polylog}(f(x)))$ and $\mathcal{O}^*(2^{f(x)}) = \mathcal{O}(2^{f(x)} \cdot \text{poly}(x))$.

1.3.2 Probability Toolbox

In what follows let $\mathcal{X}$ be a finite set satisfying $|\mathcal{X}| = V$. A *Markov chain* $\mathcal{M}$ is a random process that defines movements between elements of $\mathcal{X}$. Transitions between states are determined according to a fixed probability distribution, and can be represented by an $V \times V$ (though not necessarily symmetric) *transition matrix* P. The entry $P_{kj} := P(k, j)$ is the probability of making a transition from k to j, and the rows of P sum to 1 to preserve normalization $\sum_{j \in \mathcal{X}} P(k, j) = 1$; we say that such a matrix is *stochastic*. One step in the chain obeys

$$\mu^{(t)} P = \mu^{(t+1)} \quad \forall t \geq 0. \tag{1.5}$$

For any initial distribution $\mu^{(0)} \in \mathbb{R}^n$ over $\mathcal{X}$, the distribution after t steps of the walk is

$$\mu^{(0)} P^t = \mu^{(t)} \quad \forall t \geq 0. \tag{1.6}$$

We say that a distribution π over $\mathcal{X}$ is a *stationary distribution* if

$$\pi P = \pi. \tag{1.7}$$

For a function $f : \mathcal{X} \mapsto \mathbb{R}$, we define

$$\mathbb{E}_{y \underset{P}{\sim} x}[f(x)] := (Pf)(x) = \sum_{y \in \mathcal{X}} P(x, y) f(y), \tag{1.8}$$

and for two such functions f, g their π inner product is

$$\langle f, g \rangle_\pi = \sum_{x \in \mathcal{X}} \pi(x) f(x) g(x). \tag{1.9}$$

Definition 1.3.2 (Time-reversal of a Markov Chain)**.** The *time reversal* of $\mathcal{M}$ is defined as the Markov chain $\mathcal{M}^* = (\mathcal{X}, P^*, \pi)$, which shares the stationary distribution π of $\mathcal{M}$, and P^* is defined by the equation:

$$\pi(x) P(x, y) = \pi(y) P^*(y, x). \tag{1.10}$$

Definition 1.3.3 (Reversibility)**.** The chain $\mathcal{M}$ is called *reversible* if $P^* = P$.

Definition 1.3.4 (Distance Metrics)**.** The *total variation distance* between two probability distributions μ and ν on $\mathcal{X}$ is defined by

$$\mathrm{TV}(\mu, \nu) := \frac{1}{2} \|\mu - \nu\|_1. \tag{1.11}$$

It also satisfies the following variational formula

$$\mathrm{TV}(\mu, \nu) = \sup_{f : \|f\| \leq 1} \frac{1}{2} (\mathbb{E}_\mu[f(x)] - \mathbb{E}_\nu[f(x)]). \tag{1.12}$$

Kullback–Leibler (KL) divergence (or, relative entropy) of two probability distributions μ and ν over $\mathcal{X}$:

$$\mathrm{KL}(\mu \| \nu) := \sum_{x \in \mathcal{X}} \mu(x) \ln \left(\frac{\mu(x)}{\nu(x)} \right).$$

We also have the Donsker and Varadhan's variational formula [DV83] for $\mathrm{KL}(\mu \| \nu)$:

$$\mathrm{KL}(\mu \| \nu) = \sup_{f \in \mathcal{F}} \left(\mathbb{E}_\mu[f(x)] - \ln \left(\mathbb{E}_\nu[\exp(f(x))] \right) \right), \tag{1.13}$$

where $\mathcal{F}$ denotes the set of all measurable functions.
The p-Wasserstein distance between μ and ν on $\mathcal{X}$ is defined as

$$\mathrm{W}_p(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu, \nu)} \mathbb{E}_{(x,y) \sim \gamma} \|x - y\|^p \right)^{1/p}, \tag{1.14}$$

where $\Gamma(\mu, \nu)$ is the set of all joint distributions $\gamma(x, y)$ whose marginals are μ and ν. The relative Fisher information between μ and ν on $\mathcal{X}$ is defined as

$$\mathrm{FI}(\pi\|\mu) = \sum_{x\in\mathcal{X}} \pi(x) \left\| \nabla \log\left(\frac{\pi(x)}{\mu(x)}\right)\right\|^2. \tag{1.15}$$

Finally the Hellinger distance between μ and ν on $\mathcal{X}$ is defined as

$$\mathrm{H}(\mu, \nu) \coloneqq \left(\frac{1}{2}\sum_{x\in\mathcal{X}} \left(\sqrt{\mu(x)} - \sqrt{\nu(x)}\right)^2\right)^{1/2}. \tag{1.16}$$

Definition 1.3.5 (Cheeger constant)**.** Let ν be a probability measure on Ω. Then ν satisfies the isoperimetric inequality with Cheeger constant ρ if for any $A \subseteq \mathcal{X}$, it holds that

$$\lim_{h\to 0^+} \inf \frac{\nu(A_h) - \nu(A)}{h} \geq \rho \min\{\nu(A), 1 - \nu(A)\}, \tag{1.17}$$

where $A_h = \{x \in \mathcal{X} : \exists y \in A, \|x - y\| \leq h\}$.

Definition 1.3.6 (log-Sobolev inequality for measures)**.** Let ν be a probability measure on $\mathcal{X}$. We say that ν satisfies log-Sobolev inequality with constant c_{LSI} if for any smooth function g on $\mathbb{R}^d$, satisfying $\int_x g(x)\nu(x)\,\mathrm{d}x = 1$, it holds that

$$\int g(x)\log(g(x))\nu(x)\,\mathrm{d}x \leq \frac{1}{2c_{\mathrm{LSI}}}\int \frac{\|\nabla g(x)\|^2}{g(x)}\nu(x)\,\mathrm{d}x. \tag{1.18}$$

The Cheeger constant measures the bottleneck of a measure and Log Sobolev Inequality is a sampling analog of the PL (Polyak-Łojasiewicz) condition commonly used in optimization [CS24] and standard in non-log-concave sampling literature [VW19, MCJ+19, CEL+22, KS22]. LSI relaxes strong convexity in the sense that for any μ strongly convex function f, π satisfies the Log-Sobolev inequality with constant $\frac{\mu}{2}$. This inequaloty is weaker than the dissipative gradient condition [RRT17, ZXG19] which is used commonly in non-log-concave sampling.

Definition 1.3.7 (Mixing time)**.** The *mixing time* of a Markov chain is the amount of time it takes for the distance to stationarity to be small:

$$t_{\mathrm{mix}}(\varepsilon) := \min\left\{t : d(t) \leq \varepsilon\right\}, \tag{1.19}$$

where $d(t) := \sup_\mu \mathrm{TV}(\mu P^t, \pi)$.

The mixing time of a *reversible* Markov chain is related spectral properties of P. In the reversible case, the matrix P is similar to a symmetric matrix, and thus diagonalizable. The eigenvalues of P can be ordered as

$$1 \geq \lambda_1 \geq \cdots \geq \lambda_N \geq -1. \tag{1.20}$$

It is known that $\lambda_1 = 1$ and $\lambda_2 < 1$. We define the *spectral gap* of a reversible Markov chain to be:

$$\delta := 1 - \max_{\{\lambda_2,\ldots,\lambda_N\}} \{|\lambda| : \lambda \neq \pm 1\}. \tag{1.21}$$

The relationship between mixing time and the spectral gap can be expressed as

$$t_{\text{mix}}(\varepsilon) = \widetilde{\mathcal{O}}_{\frac{1}{\varepsilon}}\left(\delta^{-1}\right). \tag{1.22}$$

A larger spectral gap therefore implies faster mixing, meaning that the Markov chain more rapidly converges to the stationary distribution.

Next we define the *Discriminant matrix* of a Markov chain, which is a useful tool for analyzing random walks.

Definition 1.3.8 (Discriminant matrix)**.** For a Markov chain $\mathcal{M} = (\mathcal{X}, P, \pi)$, the discriminant matrix is the operator with elements

$$D(P)_{ji} := \sqrt{P_{ij} \circ P^*_{ji}} = \left(\text{diag}(\pi)^{1/2} P \text{diag}(\pi)^{-1/2}\right)_{ij}. \tag{1.23}$$

Furthermore, if P is reversible, then $D(P)$ is symmetric, and the following hold:

1. The unique, maximum eigenvalue eigenstate of $D(P)$ is $|\sqrt{\pi}\rangle$ with eigenvalue 1.
2. The spectral gap of $-D(P)$ (and equivalently P) is

$$\delta := 1 - \max\{|\lambda| : \lambda \in \sigma(D(P)), \lambda \neq \pm 1\}. \tag{1.24}$$

3. $\|D(P)\|_2 = 1$.
4. If P is symmetric, then $D(P) = P$.

Definition 1.3.9 (Markov functionals)**.** Let $f : \mathcal{X} \mapsto \mathbb{R}$. The Dirichlet form, $\mathcal{D}(f, f)$, generated by a Markov chain $\mathcal{M} = (\mathcal{X}, P, \pi)$ is defined by

$$\mathcal{D}(f, f) := \langle f, (I - P)f \rangle_\pi, \tag{1.25}$$

and if P is reversible, then $I - P$ is symmetric with respect to the π-inner product, and so $\mathcal{D}$ extends to an inner product for functions f, g:

$$\mathcal{D}(f, g) = \langle f, (I - P)g \rangle_\pi = \frac{1}{2}\mathbb{E}_{x\sim\pi}\mathbb{E}_{y\underset{P}{\sim}x}[(f(x) - f(y))(g(x) - g(y))]. \tag{1.26}$$

The π-Variance of f is defined as

$$\mathrm{Var}_\pi(f) := \mathbb{E}_\pi[f^2] - (\mathbb{E}_\pi[f])^2, \tag{1.27}$$

and the π-Entropy of f is defined as

$$\mathrm{Ent}_\pi(f) := \mathbb{E}_\pi[f \ln(f)] - \mathbb{E}_\pi[f \ln(\mathbb{E}[f])]. \tag{1.28}$$

One can relate the variance to the Dirichlet form and the spectral gap using a *Poincaré inequality*:

Definition 1.3.10 (Poincaré inequality)**.** A Markov chain $\mathcal{M} = (\mathcal{X}, P, \pi)$ satisfies a Poincaré inequality with constant δ if

$$\mathcal{D}(f, f) \geq \delta \mathrm{Var}_\pi(f). \tag{1.29}$$

For reversible Markov chains, the Poincaré constant is equal to the spectral gap. It is possible to obtain better bounds on the mixing time of a Markov chain using the so-called *logarithmic Sobolev* inequalities.

Definition 1.3.11 (log-Sobolev inequality for Markov chains)**.** A Markov chain $\mathcal{M} = (\mathcal{X}, P, \pi)$ satisfies a log-Sobolev inequality with constant $\omega := \omega_{LS}$ if

$$\mathcal{D}(f, f) \geq \omega \mathrm{Ent}_\pi(f^2). \tag{1.30}$$

Definition 1.3.12 (modified log-Sobolev inequality)**.** A Markov chain $\mathcal{M} = (\mathcal{X}, P, \pi)$ satisfies a log-Sobolev inequality with constant ω_{MLS} if

$$\mathcal{D}(f, \ln f) \geq \omega_{\mathrm{MLS}} \mathrm{Ent}_\pi(f). \tag{1.31}$$

The following chain of inequalities is well-known:

$$\delta \geq \omega_{\mathrm{MLS}} \geq \omega_{\mathrm{LS}}. \tag{1.32}$$

The Poincaré and log-Sobolev inequalities belong to a group known as the functional inequalities. They are well-defined for non-reversible chains, although Poincaré now bounds the singular-value gap, and enable one to bound the corresponding mixing time [Cha23].

Definition 1.3.13 (Fokker-Planck Equation)**.** Consider the following stochastic differential equation

$$dX = v(X)dt + \sqrt{2}dW, \tag{1.33}$$

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

$$\frac{\partial \mu_t}{\partial t} = -\nabla \cdot (\mu_t v a) + \Delta \mu_t, \tag{1.34}$$

where $\nabla\cdot$ is the divergence operator and Δ is the Laplacian.

Finally, we define the P-pseudo Lipschitz norm, which measures the smoothness of a function with respect to the transition probabilities of a Markov chain.

Definition 1.3.14 (P-pseudo Lipschitz norm)**.** Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a Markov chain. The P-pseudo Lipschitz norm of $f : \mathcal{X} \mapsto \mathbb{R}$ is defined to be

$$\|f\|_P := \max_{x \in \mathcal{X}} \mathbb{E}_{y \underset{P}{\sim} x}[(f(x) - f(y))^2]. \tag{1.35}$$

1.3.3 Quantum Computing

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

The state space of a quantum system with n subsystems is the tensor product space $\mathbb{C}^{d_1} \otimes \cdots \otimes \mathbb{C}^{d_n}$. The tensor (Kronecker) product of two vectors $|u\rangle \in \mathbb{C}^{d_1}$ and $|v\rangle \in \mathbb{C}^{d_2}$ is the vector $|u\rangle \otimes |v\rangle \in \mathbb{C}^{d_1 d_2}$, given explicitly by:

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

A single qubit is a normalized vector in $\mathbb{C}^2$, written as $\alpha|0\rangle + \beta|1\rangle$, with $|\alpha|^2 + |\beta|^2 = 1$. A system of n qubits lives in the Hilbert space $\mathbb{C}^{2^n}$, and a general n-qubit state may

be entangled, i.e., not expressible as a tensor product of single-qubit states. We often abbreviate tensor products such as $|u\rangle \otimes |v\rangle$ by $|u\rangle |v\rangle$.

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

Measurement is the process of extracting classical information from a quantum system. A projective measurement in the computational basis $\{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$ yields outcome i with probability $p_i = |\langle i|\psi\rangle|^2$, collapsing the state $|\psi\rangle$ to $|i\rangle$.

Realization of the contributions in this dissertation requires a fault-tolerant quantum computer that can implement the described procedures with proper error correction as opposed to NISQ era computers that are designed to run with a few number of qubits without error correction.

Chapter 2
Quantum Speedups for Sampling from Continuous Non-Logconcave Distributons via Non-Reversible Markov Chains

In this chapter, we present quantum algorithms for sampling from possibly non-logconcave probability distributions expressed as $\pi(\mathbf{x}) \propto \exp(-\beta f(\mathbf{x}))$ as well as quantum algorithms for estimating the partition function for such distributions. We also incorporate a stochastic gradient oracle that implements the quantum walk operators inexactly by only using mini-batch gradients when f can be written as a finite sum. One challenge of quantizing the resulting Markov chains is that they do not satisfy the detailed balance condition in general. Consequently, the mixing time of the algorithm cannot be expressed in terms of the spectral gap of the transition density matrix, making the quantum algorithms nontrivial to analyze. We overcome these challenges by first building a reference reversible Markov chain that converges to the target distribution, then controlling the discrepancy between our algorithm's output and the target distribution by using the reference Markov chain as a bridge to establish the total complexity. We prove that our quantum algorithms exhibit polynomial speedups in terms of dimension or precision dependencies when compared to best-known classical algorithms under similar assumptions.

This chapter is based on [OLMW24], joint with Xiantao Li, Mehrdad Mahdavi and Chunhao Wang.

2.1 Introduction

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

$$\pi(\mathbf{x}) = \frac{e^{-f(\mathbf{x})}}{\int e^{-f(\mathbf{x})} \mathrm{d}\mathbf{x}}. \tag{2.1}$$

This distribution is called the Boltzmann-Gibbs distribution, and our goal is to efficiently sample approximately from π while minimizing the number of gradient queries in the finite-sum setting, i.e., $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$.

A well-known classical method for Gibbs sampling is Markov Chain Monte Carlo (MCMC) method, where a Markov chain with desired stationary density is constructed. Then, the samples can be generated by running the Markov chain for a sufficiently long time (See Section 1.3.2 for more details on Markov chains). One such Markov chain can be obtained through careful discretization of Langevin diffusion equation and this technique inspired a large family of gradient-based sampling algorithms.

Langevin diffusion, often referred to as Langevin dynamics, is a fundamental stochastic differential equation that describes the dynamics of a particle undergoing random motion in a fluid or complex environment. It is widely used in various scientific disciplines, including physics, chemistry, and biology, to model systems exhibiting Brownian motion or other forms of random behavior. It also provides a probabilistic approach to optimization by simulating the motion of particles under the influence of both deterministic gradient forces and random noise. This allows the optimization process to explore the parameter space more extensively, potentially escaping local optima and reaching a broader range of solutions. By simulating Langevin dynamics, machine learning practitioners can also sample from the posterior distribution of the model parameters, enabling Bayesian inference and uncertainty estimation. The dynamics follows the solution to the following stochastic differential equation (SDE):

$$d\mathbf{x}_t = -\nabla f(\mathbf{x}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \tag{2.2}$$

where $\mathbf{B}_t$ is the standard Brownian motion. This continuous dynamics converges to the Gibbs distribution under mild conditions on the potential function f. The Euler-Maruyama discretization of this SDE results in the well-known Langevin Monte Carlo

(LMC) algorithm:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \nabla f(\mathbf{x}_k) + \sqrt{2\eta_k}\boldsymbol{\epsilon}_t, \tag{2.3}$$

where η_k is the step size and $\boldsymbol{\epsilon}_k$ is isotropic Gaussian noise.

Algorithm 1 Unadjusted Langevin Algorithm (ULA)

1: **Input:** $\mathbf{x}_0, (\eta_{[0,K-1]} > 0)$.
2: **Output:** An approximate sample from π.
3: **for** $k = 0, ..., K-1$ **do**
4: $\quad \mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \nabla f(\mathbf{x}_k) + \sqrt{2\eta_k}\boldsymbol{\epsilon}_k$
5: **end for**
6: Return $\mathbf{x}_K$

Due to finite-sized discretization, the Markov chain corresponding to LMC is asymptotically biased. That is, it only converges to the neighborhood of the desired Gibbs distribution, prohibiting one from using large step sizes because of this discrepancy. To overcome this bias, one can adjust the Markov chain by introducing a Metropolis-Hastings filter, which is used as a conditional rejection to guarantee that the chain is time-reversible. A Markov chain that satisfies the detailed balance condition is known as time-reversible and it is a fundamental property for establishing the mixing time in terms of spectral gap or conductance. Mathematically, it can be expressed as:

$$P(\mathbf{x}, \mathbf{y})\pi(\mathbf{x}) = P(\mathbf{y}, \mathbf{x})\pi(\mathbf{y}). \tag{2.4}$$

where P is the transition kernel of the Markov chain. This condition also guarantees that the Markov chain will converge to a stationary distribution.

The algorithm with the Metropolis-Hastings filter is sometimes referred to as Metropolis-adjusted Langevin algorithm (MALA), and the algorithm without the rejection step is conventionally called unadjusted Langevin algorithm (ULA).

In the past decade, notable progress has been witnessed in the theoretical development of quantum algorithms for various machine learning and optimization problems. It is natural to expect that quantum computers also provide provable speedups for general sampling problems. If we could prepare a quantum state whose amplitudes correspond to some desired distribution, then measuring this state yields a random sample from this probability distribution on Ω. That is a quantum state of the following form

$$|\pi\rangle = \sum_{\mathbf{x}\in\Omega} \sqrt{\pi(\mathbf{x})}\, |\mathbf{x}\rangle\,. \tag{2.5}$$

Algorithm 2 Metropolis Adjusted Langevin Algorithm (MALA)

Input: $\mathbf{x}_0, (\eta_{[0,K-1]} > 0)$.
Output: An approximate sample from π.
for $k = 0, ..., K-1$ **do**
  $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \nabla f(\mathbf{x}_k) + \sqrt{2\eta_k}\boldsymbol{\epsilon}_k$
  $\alpha = \frac{p(\mathbf{x}_k|\mathbf{x}_{k+1})\pi(\mathbf{x}_k)}{p(\mathbf{x}_{k+1}|\mathbf{x}_k)\pi(\mathbf{x}_{k+1})}$
  $u \sim \mathcal{U}[0,1]$
  if $\alpha < u$ **then**
    $\mathbf{x}_{k+1} = \mathbf{x}_k$
  end if
end for
Return $\mathbf{x}_K$

Then a measurement on this state yields the basis $|\mathbf{x}'\rangle$ with probability $\pi(\mathbf{x}')$. Unfortunately, quantum speedups in such sampling models probably do not hold in general as this will imply $\mathsf{SZK} \subseteq \mathsf{BQP}$ [ATS03]. While the hardness barrier exists for a quantum speedup for general sampling problems, in some special cases, it has been shown that quantum algorithms can achieve polynomial speedups over classical algorithms. Such examples include quantum algorithms for uniform sampling on a 2D lattice [Ric07], for estimating partition functions [WA08, WCNA09, Mon15, HW20, AHN$^+$21, CH23], and for estimating volumes of convex bodies [CCH$^+$23].

Recently, a quantum MALA algorithm based on quantum simulated annealing is introduced [CLL$^+$22], which leverages the fact that a coherent quantum state corresponding to desired logconcave distribution can be prepared using fewer number of calls to gradient and evaluation oracle than the classical counterparts.

Inspired by this, an interesting question arises: Can we attain quantum speedups for more general distributions, such as non-logconcave distributions? Moreover, one intriguing open question posed in [CLL$^+$22] was the possibility of speeding up unadjusted Langevin algorithm using similar techniques. The *main challenge* for analyzing quantum version of ULA is that the transition density does not satisfy the detailed balance condition due to lack of the Metropolis-Hastings filter. Hence the Markov chain is not time reversible which is the main assumption for almost all quantum walk based algorithms [Sze04, WA08, MNRS07, AS19]. The current quantum walk frameworks leverage the fact that a symmetric discriminant matrix D (See Definition 1.3.8 for the definition) can be related to the spectrum of the classical transition matrix P. Then the eigenstate of D with unique singular value 1 encodes the coherent quantum state whose amplitudes are the desired Gibbs density. Then, by extracting this eigenstate using a quantum

computer can prepare the Gibbs state quadratically faster than the classical computers in spectral gap parameter. However, for non-reversible Markov chains, this extraction is straightforward. In fact, the discriminant matrix for non-reversible chains might have zero singular value gap [MNRS07], which breaks down the entire quantum algorithm. We also touch upon this technical difficulty of quantizing non-reversible Markov chains in Section 2.4 in more detail.

2.1.1 Main Contributions

- We analyze the mixing time of the quantum MALA algorithm (Theorem 2.4.5) for non-logconcave distributions, extending the work done in [CLL$^+$22]. The main challenge in analyzing quantum MALA for non-logconcave distributions is to characterize the phase gap of the quantum walk and to show the existence of a quantum annealing schedule that guarantees a large overlap between successive distributions since the target distribution does not satisfy the concentration inequalities as in log-concave case. By using the conductance analysis done in [ZXG21], we characterize this phase gap. Next, we show that by using isoperimetric inequalities, the length of the annealing schedule is $\tilde{\mathcal{O}}(\sqrt{d})$ similar to non-logconcave case (Section 2.3).

- Next, we analyze the quantum ULA algorithm (Theorem 2.4.10) using a novel perturbation analysis with respect to quantum MALA to show that quantum computers can provide speedups even for non-reversible Markov chains. Since quantum MALA is time-reversible (as it satisfies Equation (2.4)) and asymptotically unbiased, it converges to target distribution, allowing us to express our algorithms' error with respect to Gibbs distribution. In the construction of our algorithms, we use standard quantum simulated annealing techniques as in [CLL$^+$22] while the underlying Markov chain is not reversible. Although perturbation techniques have been used in classical analysis of Markov chains [ZXG21, RRT17, XCZG18], these results cannot be transferred to quantum setting as the quantum walk algorithms are fundamentally different. That is, while classical algorithms run in an iterative fashion to generate candidate samples, quantum algorithms use linear algebraic techniques to rotate the input state towards the eigenvector in the invariant subspace. We believe that this technique can be useful for the analysis of other non-reversible Markov chains as an independent tool.

- We further incorporate stochastic gradient oracle to make the implementation of quantum walk efficient and provide the mixing time of our stochastic quantum sampling

algorithm in Theorem 2.4.14. In addition to the error due to the lack of Metropolis-Hastings filter, the stochastic algorithm introduces additional errors because of the noisy gradients. We use concentration techniques to show that even with unitaries implemented via stochastic gradients, the quantum algorithm gives the correct distribution with high probability.

- Finally, we combine our sampling algorithms with recently developed efficient quantum product estimator [CH23] and proposed algorithms for computing the partition function for non-logconcave distributions in Section 2.5.

2.1.2 Problem formulation

We focus on designing and analyzing quantum algorithms for sampling from the Gibbs density $\pi(\mathbf{x}) \propto e^{-\beta f(\mathbf{x})}$ where f is not necessarily convex. We note that sometimes we write the β, known as inverse temperature, explicitly to express dependency of the mixing time on β.

An important scenario in machine learning is when f admits a decomposition,

$$f(\mathbf{x}) = \frac{1}{N}\sum_{k=1}^{N} f_k(\mathbf{x}), \tag{2.6}$$

where $N \gg 1$ is large. One typical example is where $\mathbf{x}$ comes from the model parameter, and f is the empirical loss defined on a large data set. Clearly, this will cause a significant slowdown of quantum MALA algorithm due to function and gradient evaluations when f is given in this finite sum form.

We further make the following assumptions on f. These assumptions are realistic as they are satisfied in many applications and are widely assumed in the non-logconcave sampling literature [RRT17, ZXG21].

Assumption 2.1.1 (Smoothness)**.** There exists a positive constant L such that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and all functions $f_k(\mathbf{x}), k \in [N]$, it holds that

$$\|\nabla f_k(\mathbf{x}) - \nabla f_k(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|. \tag{2.7}$$

Assumption 2.1.2 (Dissipativeness)**.** There are absolute constants $m > 0$ and $b \geq 0$ such that

$$\langle \nabla f(\mathbf{x}), \mathbf{x} \rangle \geq m\|\mathbf{x}\|^2 - b. \tag{2.8}$$

The first assumption ensures that small changes in the input parameters result in bounded changes in gradients whereas the second one implies that f grows like a quadratic form outside a ball.

2.1.3 Oracle model

We assume that we have the access to the following quantum oracles to implement our algorithm. These oracles are virtually classical oracles while empowering superposition access. We first define the *full gradient oracle* for f as follows:

$$O_{\nabla f} |\mathbf{x}\rangle |0\rangle = |\mathbf{x}\rangle |\nabla f(\mathbf{x})\rangle . \tag{2.9}$$

Similarly, we define a *stochastic gradient oracle*,

$$O_{\tilde{\nabla} f} |\mathbf{x}\rangle |0\rangle = |\mathbf{x}\rangle \ |\tilde{\nabla} f(\mathbf{x})\rangle . \tag{2.10}$$

where $\tilde{\nabla} f(\mathbf{x}) = \frac{1}{B} \sum_{k \in S} \nabla f_k(\mathbf{x})$ where S is a subset of size B data samples chosen randomly without replacement. Note that $O_{\tilde{\nabla} f}$ possibly outputs a different state for the same input state depending on the internal random batch. Finally, the *evaluation oracle* is defined by

$$O_f |\mathbf{x}\rangle |0\rangle = |\mathbf{x}\rangle |f(\mathbf{x})\rangle . \tag{2.11}$$

We note that although we quantified the complexity of our algorithm in terms of the number of calls to these oracles, the evaluation oracle and full gradient oracle are slower than the stochastic gradient oracle for finite sum form due to evaluation of N terms. One of our contributions is to use a stochastic gradient oracle to make quantum walk implementation more efficient.

We also emphasize that our gradient oracles, including those for stochastic gradients, operate classically with superposition access. Since classical circuits can be simulated by quantum circuits with a constant overhead, implementation cost of these oracles is on par with the classical oracles. Therefore, any speedups with respect to the number of calls to these oracles are not suppressed by their implementation cost. Furthermore, our quantum algorithms are robust to a small error in these oracles. This is because a small error in gradient will introduce a perturbation of the Markov chain and the resulting quantum walk. The analysis in this work, which is precisely based on quantifying the difference between two Markov chains (a time-irreversible and a time-reversible chain), can quantify how the gradient error can propagate in the algorithm. In fact, our analysis

for quantum ULA with stochastic gradients show that the algorithm works for even noisy gradients.

It is worth mentioning that one particular reason to analyze ULA and stochastic ULA is that implementing one step of MALA requires N function and gradient evaluations when f can be decomposed into a sum of N terms, whereas ULA only uses N gradient evaluations as it does not do any adjustment. We prove that stochastic ULA only needs $\tilde{\mathcal{O}}(d)$ gradient evaluations to converge which is a significant improvement especially when $d \ll N$. Therefore, we believe each algorithm is suited to specific use cases, depending on the size of the data set, problem dimension, and hardness of function and gradient evaluations.

Remark 2.1.3. Classical works used in Table 2.1 has slightly different assumptions than Assumptions 2.1.1 and 2.1.2. In the work of [MCJ+19], the function f is assumed to be strongly convex outside a sufficiently large ball with radius. This shares the same intuition with the dissipative condition, i.e., sufficiently fast growth in the far field. Moreover it implies the dissipative condition. The isoperimetry condition in [VW19] relies on log-Sobolev inequality, which can be proved from the dissipative condition together with the Lipschitz condition (See proposition 3.2 in [RRT17]). In fact, this is why we do not add log-Sobolev equality as an additional assumption.

2.2 Prior Work

Extensive research has been conducted to understand the non-asymptotic dynamics of Langevin based algorithms for both log-concave and non-logconcave densities under various settings. This section reviews a selection of significant works to provide context for our study, given the extensive literature available.

For log-concave distributions, a significant body of research has been conducted to understand the dynamics of the Langevin Monte Carlo (LMC) based algorithms [BEL15, Dal17b, Dal17a, DMM19, LZT22]. Sampling from non-logconcave distributions under various assumptions have also been analyzed broadly [LRG18, VW19, MCJ+19, XCZG18]. The convergence of LMC under the condition that the target density satisfies isoperimetry condition is shown by [VW19]. Although the gradient descent methods are known to be superior to sampling-based optimization in convex cases, [MCJ+19] showed that sampling-based methods could provide speedups over local optimization methods in non-convex setting which motivates us to explore the quantum algorithms for non-logconcave densities. The stochastic extension of the algorithm (SGLD) in non-logconcave setting has

Table 2.1. Comparison of our sampling algorithm to classical results with similar assumptions, focusing on the dependencies on d and ϵ.

Algorithm	**Complexity**	**Oracle**	**Assumptions**
ULA [MCJ+19]	$\tilde{\mathcal{O}}(d/\epsilon^2)$	Full Gradient	Local non-convex
MALA [MCJ+19]	$\tilde{\mathcal{O}}(d^2)$	Full Gradient & Evaluation	Local non-convex
ULA [VW19]	$\tilde{\mathcal{O}}(d/\epsilon^2)$	Full Gradient	Isoperimetry
SGLD [ZXG21]	$\tilde{\mathcal{O}}(d^4/\epsilon^2)$	Stochastic Gradient	Dissipative Gradients
Quantum MALA (Theorem 2.4.5)	$\tilde{\mathcal{O}}(d)$	Full Gradient & Evaluation	Dissipative Gradients
Quantum ULA (Theorem 2.4.10)	$\tilde{\mathcal{O}}(d^{3/2}/\epsilon)$	Full Gradient	Dissipative Gradients
Stochastic Quantum ULA (Theorem 2.4.14)	$\tilde{\mathcal{O}}(d^{5/2}/\epsilon^2)$	Stochastic Gradient	Dissipative Gradients

been investigated recently in several works. The hitting time of the stochastic Langevin dynamics to a neighborhood of the minima is analyzed in [ZLC17] and they showed that SGLD can escape suboptimal local minima that only exist in the empirical risk function. More notably [ZXG21] and [XCZG18] analyzed the mixing time of SGLD to the stationary distribution in total variation distance using similar techniques to ours. They used perturbation analysis to show that output of SGLD is closed to a reversible chain. Unfortunately, their result does not transfer to the quantum setting due to fundamental difference between quantum and classical Monte Carlo algorithms.

We also note the classical sampling algorithms that uses more sophisticated techniques to improve the mixing time in terms of various distances. One such popular technique is called Hamiltonian Monte Carlo method [BGJM11] which uses the momentum and leapfrog integrator to reduce the error of discretization which improves the sampling time. Based on underdamped Langevin Monte Carlo algorithm, [SL19] proposed randomized midpoint method to sample from log-concave distributions with better dependencies compared to unadjusted Langevin algorithm. Furthermore, [FYC23] used proximal sampling algorithm to improve the dimension dependence to $d^{1/2}$ under the log-Sobolev inequality. However, their assumption is ℓ_1 smoothness which differs from our Assumption 2.1.1. As we will explore these samplers in the next chapter, more details will not be given here.

We only compare our results to the classical ones that use the same or very similar assumptions and only claim our polynomial speedups with respect to these results. The comparison is summarized in Table 2.1.

The *quantum walk* operators used in this work is developed in [Sze04]. This technique has been shown to provide speedups for reversible Markov Chain Monte Carlo (MCMC) methods by improving the mixing time of the underlying Markov chain [Sze04, SBB07, WA08, WA08, CCH$^+$23]. These methods have been incorporated into various domains to improve the computation time of various tasks [AS19, CLL$^+$22, CCH$^+$23, LZ24, OLMW24, CHO$^+$24]. The speedup of non-reversible Markov chains has been discussed by [MNRS07] in the context of quantum search. However, their construction implements time reversal of the Markov chain $P^*(\mathbf{y}, \mathbf{x}) = P(\mathbf{x}, \mathbf{y})\pi(\mathbf{x})/\pi(\mathbf{y})$ which requires N function evaluations in our case. Therefore, this construction would lead to an algorithm somewhat similar to our quantum MALA algorithm. After the completion of this chapter, [CPM25] proposed a similar technique to ours to obtain quantum speedups for non-reversible Markov chains, using the idea of geometric reversibilization with respect to the so-called "most reversible" distribution. Although their result applies to a broader class of Markov chains, it still requires bounding certain quantities, such as the spectral gap of the geometric reversibilization and the overlap between the stationary distribution and the most reversible distribution.

2.3 Annealing Schedule for Non-Logconcave Distributions

A known technique to speedup classical Markov chains is to design a sequence of slowly changing Markov chains [WA08]. The construction is first to define a series of Markov chains with stationary distributions $\pi_1, ..., \pi_M$. Then for each Markov chain, a projector is constructed to iteratively drive the initial state $|\pi_1\rangle$ to the final state $|\pi_M\rangle$ using amplitude amplification. If the overlap $|\langle\pi_i|\pi_{i+1}\rangle| \geq \Omega(1)$ for all $i \in [1, M-1]$, then the cost of the algorithm becomes M times the cost of implementing quantum walk for each Markov chain.

To implement slowly varying Markov chains for non-logconcave distributions, we prove the following lemma to construct the annealing schedule. It shows that there exists an annealing schedule of length $\tilde{\mathcal{O}}(\sqrt{d})$ such that the adjacent quantum states have large overlap. Our construction is similar to [GLL20] in that we start from quantum Gaussian state and slowly decrease the Gaussian component of the distribution so that final state is very close to target Gibbs state.

Lemma 2.3.1 (Quantum Annealing)**.** *Under Assumptions 2.1.1 and 2.1.2, there exists a series of quantum states* $|\mu_0\rangle, |\mu_1\rangle, \ldots, |\mu_M\rangle$ *satisfying the following properties:*

1. *There exists an efficient quantum algorithm to prepare initial state* $|\mu_0\rangle$ *without using any function queries.*

2. *For all* $i \in \{0, \ldots, M-1\}$ *,* $|\mu_i\rangle$ *and* $|\mu_{i+1}\rangle$ *has at least constant overlap, i.e.,*

$$|\langle \mu_i | \mu_{i+1} \rangle| \geq \Omega(1). \tag{2.12}$$

3. *The final state* $|\mu_M\rangle$ *has at least constant overlap with the target Gibbs state* $|\pi\rangle$*,*

$$|\langle \mu_M | \pi \rangle| \geq \Omega(1). \tag{2.13}$$

4. *The number of quantum states* $M \leq \tilde{\mathcal{O}}(c_{\mathrm{LSI}}^{-1} d^{1/2})$*.*

We first restate the following useful lemmas from previous works as we use them repeatedly in our proofs. The first one lower bounds f by a quadratic function whereas the second one upper bounds the norm of the gradient by a linear function. We refer the readers to the original papers for their proofs and we don't repeat here for readability.

Lemma 2.3.2 (Lemma A.1 in [ZXG21])**.** *Under Assumption 2.1.2, the objective function* $f(\mathbf{x})$ *satisfies,*

$$f(\mathbf{x}) \geq \frac{m}{4}\|\mathbf{x}\|^2 + f(\mathbf{x}^*) - b/2, \tag{2.14}$$

where $f(\mathbf{x}^*) = \min_{\mathbf{x}} f(\mathbf{x})$*.*

Lemma 2.3.3 (Lemma 3.1 in [RRT17])**.** *Under Assumption 2.1.1, there exists a constant* $G = \max_{k\in[N]} \|\nabla f_k(0)\|$ *such that for any* $\mathbf{x} \in \mathbb{R}^d$ *and* $k \in [n]$*, it holds that,*

$$\|\nabla f_k(\mathbf{x})\| \leq L\|\mathbf{x}\| + G. \tag{2.15}$$

The next three technical lemmas are presented here to make the proof of the annealing schedule concise. From a more technical perspective, these lemmas generalize the work done in [GLL20] for non-logconcave distributions under Assumption 2.1.1 and Assumption 2.1.2.

Lemma 2.3.4. *Suppose* $\pi(x) \propto e^{-f(\mathbf{x})}$ *is a Gibbs measure and* f *satisfies Assumptions 2.1.1 and 2.1.2. Then, we have*

$$\mathbb{E}_\pi\left[\exp\left(-s\|\mathbf{x}\|^2\right)\right]\mathbb{E}_\pi\left[\exp\left(s\|\mathbf{x}\|^2\right)\right] \leq \mathcal{O}(\exp\left(dLs^2/(mc_s^2)\right)), \tag{2.16}$$

where c_s^2 *is the log-Sobolev constant of the distribution* $\pi_s \propto \pi e^{s\|\mathbf{x}\|^2}$*.*

Proof. Let $h(s) = \mathbb{E}_\pi\left[\exp(-s\|\mathbf{x}\|^2)\right]\mathbb{E}_\pi\left[\exp(s\|\mathbf{x}\|^2)\right]$, then

$$\frac{h'(s)}{h(s)} = \left(\frac{\mathbb{E}_\pi\left[\|\mathbf{x}\|^2\exp(s\|\mathbf{x}\|^2)\right]}{\mathbb{E}_\pi\left[\exp(s\|\mathbf{x}\|^2)\right]} - \frac{\mathbb{E}_\pi\left[\|\mathbf{x}\|^2\exp(-s\|\mathbf{x}\|^2)\right]}{\mathbb{E}_\pi\left[\exp(-s\|\mathbf{x}\|^2)\right]}\right) \tag{2.17}$$

$$= \int_{-s}^{s} v'(t), \mathrm{d}t, \tag{2.18}$$

where $v(t)$ is defined as,

$$v(t) = \frac{\mathbb{E}_\pi\left[\|\mathbf{x}\|^2\exp(t\|\mathbf{x}\|^2)\right]}{\mathbb{E}_\pi\left[\exp(t\|\mathbf{x}\|^2)\right]}. \tag{2.19}$$

Computing $v'(t)$ gives,

$$v'(t) = \frac{\mathbb{E}_\pi\left[\|\mathbf{x}\|^4\exp(t\|\mathbf{x}\|^2)\right]\mathbb{E}_\pi\left[\exp(t\|\mathbf{x}\|^2)\right] - \left(\mathbb{E}_\pi\left[\|\mathbf{x}\|^2\exp(t\|\mathbf{x}\|^2)\right]\right)^2}{\left(\mathbb{E}_\pi\left[\exp(t\|\mathbf{x}\|^2)\right]\right)^2} \tag{2.20}$$

$$= \mathrm{Var}_{\pi_t}\|\mathbf{x}\|^2, \tag{2.21}$$

where π_t is a distribution defined as,

$$\pi_t(\mathbf{x}) \propto \pi(\mathbf{x})\exp\Big(t\|\mathbf{x}\|^2\Big). \tag{2.22}$$

Suppose π satisfies the log-Sobolev inequality with constant c_{LSI}, it also satisfies the Poincare inequality with the same constant (e.g [Goe04]).

$$\mathrm{Var}_{\pi_t}[\|\mathbf{x}\|^2] \le \frac{1}{c_t}\mathbb{E}_{\pi_t}[\|\mathbf{x}\|^2] \le \mathcal{O}(Ld/(mc_t^2)), \tag{2.23}$$

where c_t is LSI constant of π_t and the second inequality is due to Lemma 2.3.6. Therefore,

$$\frac{h'(s)}{h(s)} = \int_{-s}^{s} v'(t)\,\mathrm{d}t = \mathcal{O}(dLs/(mc_s^2)). \tag{2.24}$$

Hence,

$$\log(h(s)) - \log(h(0)) = \int_0^s \frac{h'(t)}{h(t)}\,\mathrm{d}t = \mathcal{O}(dLs^2/(mc_s^2)). \tag{2.25}$$

Since $h(0) = 1$, we conclude the proof. □

Lemma 2.3.5. *Suppose $\pi(x) \propto e^{-f(\mathbf{x})}$ is a Gibbs measure and f satisfies the log-Sobolev*

inequality with constant c_{LSI}. Then under Assumptions 2.1.1 and 2.1.2,

$$\frac{\mathbb{E}_\pi\left[\exp(-(1+\alpha)\|\mathbf{x}\|^2)\right]\mathbb{E}_\pi\left[\exp(-(1-\alpha)\|\mathbf{x}\|^2)\right]}{(\mathbb{E}_\pi[\exp(-\|\mathbf{x}\|^2)])^2} \leq \mathcal{O}(\exp\Big(dL\alpha^2/(c_{\mathrm{LSI}}^2 m)\Big)) \tag{2.26}$$

for $0 \leq \alpha \leq 1/2$.

Proof. This follows from Lemma 2.3.4, by setting $\tilde{\pi} \propto \pi \exp(-\|\mathbf{x}\|^2)$. Then,

$$\begin{aligned}
&\frac{\mathbb{E}_\pi\left[\exp(-(1+\alpha)\|\mathbf{x}\|^2)\right]\mathbb{E}_\pi\left[\exp(-(1-\alpha)\|\mathbf{x}\|^2)\right]}{(\mathbb{E}_\pi[\exp(-\|\mathbf{x}\|^2)])^2} && (2.27)\\
&= \mathbb{E}_{\tilde{\pi}}\left[\exp\Big(-\alpha\|\mathbf{x}\|^2\Big)\right]\mathbb{E}_{\tilde{\pi}}\left[\exp\Big(\alpha\|\mathbf{x}\|^2\Big)\right] && (2.28)\\
&\leq \mathcal{O}(\exp\Big(dL\alpha^2/(mc_\alpha^2)\Big)) && (2.29)\\
&\leq \mathcal{O}(\exp\Big(dL\alpha^2/(mc_{\mathrm{LSI}}^2)\Big)) && (2.30)
\end{aligned}$$

with c_α is LSI constant of $\pi_\alpha \propto \pi \exp(-(1-\alpha)\|\mathbf{x}\|^2)$. The last step follows from the fact that $c_\alpha \geq c_{\mathrm{LSI}}$ for $\alpha \leq 1/2$. □

Lemma 2.3.6. *Suppose $\pi(x) \propto e^{-f(\mathbf{x})}$ is a Gibbs measure and f satisfies Assumptions 2.1.1 and 2.1.2. Then*

$$\mathbb{E}_{\pi_s(\mathbf{x})}[e^{s\|\mathbf{x}\|^2}\|\mathbf{x}\|^2] \leq \mathcal{O}(Ld/(mc_s)), \tag{2.31}$$

where π_s is a probability distribution proportional to $\pi(\mathbf{x})e^{s\|\mathbf{x}\|^2}$ for a constant $s \leq \frac{m}{8}$ and c_s is the log-Sobolev constant of π_s.

Proof. Our proof follows the idea presented in proof of Lemma 6 in [MCJ+19] without the assumption of local non-convexity. We choose an auxiliary random variable $\mathbf{x}'$ following the law of $p \propto e^{-(L-s)\|\mathbf{x}\|^2}$ and couples optimally with $\mathbf{x}_s \sim \pi_s : (\mathbf{x}_s, \mathbf{x}') \sim \gamma \in \Gamma_{opt}(\pi_s, p)$.

$$\begin{aligned}
\mathbb{E}_{\pi_s}\|\mathbf{x}\|^2 &= \mathbb{E}_{(\mathbf{x}_s,\mathbf{x}'\sim\gamma)}\|\mathbf{x}' - \mathbf{x}' + \mathbf{x}_s\|^2 && (2.32)\\
&\leq 2\mathbb{E}_p\|\mathbf{x}'\|^2 + 2\mathbb{E}_{(\mathbf{x}_s,\mathbf{x}'\sim\gamma)}\|\mathbf{x}' - \mathbf{x}_s\|^2 && (2.33)\\
&= \frac{2d}{L-s} + 2\mathrm{W}_2^2(p, \pi_s) && (2.34)\\
&\leq \frac{2d}{L-s} + \frac{2}{c_s}\mathrm{KL}(p, \pi_s), && (2.35)
\end{aligned}$$

where c_{π_s} is LSI constant of π_s. The first inequality follows from Young's inequality and second inequality is due to generalized Talagrand inequality [OV00]. KL divergence can

be bounded,

$$\mathrm{KL}(p, \pi_s) = \int_{\mathbf{x}} \mathrm{d}\mathbf{x} \log\left(\frac{p(\mathbf{x})}{\pi_s(\mathbf{x})}\right) p(\mathbf{x}) \tag{2.36}$$

$$\leq \sup_{\mathbf{x}} \log\left(\frac{p(\mathbf{x})}{\pi_s(\mathbf{x})}\right) \int_{\mathbf{x}} \mathrm{d}\mathbf{x}\, p(\mathbf{x}) \tag{2.37}$$

$$= \sup_{\mathbf{x}} \log\left(\frac{p(\mathbf{x})}{\pi_s(\mathbf{x})}\right). \tag{2.38}$$

We can further bound $\frac{p(\mathbf{x})}{\pi_s(\mathbf{x})}$ for any $\mathbf{x} \in \Omega$,

$$\frac{p(\mathbf{x})}{\pi_s(\mathbf{x})} = \frac{e^{-(L-s)\|\mathbf{x}\|^2}}{\int_{\mathbf{x}} \mathrm{d}\mathbf{x}\, e^{-(L-s)\|\mathbf{x}\|^2}} \frac{\int \mathrm{d}\mathbf{x}\, e^{-f(\mathbf{x})} e^{s\|\mathbf{x}\|^2}}{e^{-f(\mathbf{x})} e^{s\|\mathbf{x}\|^2}} \tag{2.39}$$

$$= \frac{\int \mathrm{d}\mathbf{x}\, e^{s\|\mathbf{x}\|^2 - f(\mathbf{x})}}{\int \mathrm{d}\mathbf{x} e^{-(L-s)\|\mathbf{x}\|^2}} e^{-L\|\mathbf{x}\|^2 + f(\mathbf{x})} \tag{2.40}$$

$$\leq \frac{\int \mathrm{d}\mathbf{x}\, e^{s\|\mathbf{x}\|^2 - m\|\mathbf{x}\|^2/4 + b/2 - f(\mathbf{x}^*)}}{\int \mathrm{d}\mathbf{x} e^{-(L-s)\|\mathbf{x}\|^2}} e^{-L\|\mathbf{x}\|^2 + f(\mathbf{x})} \tag{2.41}$$

$$\leq \frac{\int \mathrm{d}\mathbf{x}\, e^{s\|\mathbf{x}\|^2 - m\|\mathbf{x}\|^2/4 + b/2 - f(\mathbf{x}^*)}}{\int \mathrm{d}\mathbf{x} e^{-(L-s)\|\mathbf{x}\|^2}} e^{L\|\mathbf{x}^*\|^2 + f(\mathbf{x}^*)} \tag{2.42}$$

$$= e^{b/2 + L\|\mathbf{x}^*\|^2} \frac{(L-s)^{d/2}}{(m/4 - s)^{d/2}}, \tag{2.43}$$

where the first inequality is due to Assumption 2.1.2 and Lemma 2.3.2. Second inequality follows from Equation (2.46). Hence, KL divergence is bounded by,

$$\mathrm{KL}(p, \pi_s) \leq \sup_{\mathbf{x}} \left(\frac{p(\mathbf{x})}{\pi_s(\mathbf{x})}\right) \leq b/2 + L\|\mathbf{x}^*\| + \frac{d}{2} \log\left(\frac{L-s}{m/2 - 2s}\right). \tag{2.44}$$

This implies that,

$$\mathbb{E}_{\pi_s} \|\mathbf{x}\|^2 \leq \frac{2d}{L-s} + \frac{2}{c_s}(b/2 + L\|\mathbf{x}^*\|^2 + \frac{d}{2} \log\left(\frac{L-s}{m/2 - 2s}\right) = \mathcal{O}(Ld/(mc_s)) \tag{2.45}$$

for $s \leq m/8$. □

Finally we are ready to prove our result for quantum annealing procedure. The key idea in this proofs is to show that two consequent Gibbs distributions in our annealing scheme are close to each other so that the Markov chains become slowly changing. Furthermore, we show that the final distribution is close to desired Gibbs distribution.

Proof. Our construction and analysis are similar to the annealing scheme used in [CLL+22], however our proof does not require any convexity assumption in f. The construction is as follows:

1. $|\mu_0\rangle = \sum_{\mathbf{x}} \sqrt{p_0(\mathbf{x})}\,|\mathbf{x}\rangle$, where $p_0(\mathbf{x}) = \frac{\exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma_1^2}\right)}{Z_0}$.

2. For all $i \in [1, M-1]$, $|\mu_i\rangle = \sum_{\mathbf{x}\in\Omega} \sqrt{p_i(\mathbf{x})}\,|\mathbf{x}\rangle$, where $p_i(\mathbf{x}) = \frac{\exp\left(-f(\mathbf{x})-\frac{\|\mathbf{x}\|^2}{2\sigma_i^2}\right)}{Z_i}$ such that $\sigma_{i+1}^2 = \sigma_i^2(1+\alpha)$ with $\alpha = \tilde{\mathcal{O}}(d^{-1/2}c_{\mathrm{LSI}})$.

Here, $Z_0 = \int \mathrm{d}\mathbf{x}\, \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma_1^2}\right)$ and $Z_i = \int \mathrm{d}\,\mathbf{x} \exp\left(-f(\mathbf{x}) - \frac{\|\mathbf{x}\|^2}{2\sigma_i^2}\right)$. The first property in the lemma statement holds, since p_0 corresponds to a Gaussian distribution and the coherent quantum state corresponding to Gaussian distributions can be efficiently prepared by using Box-Muller technique without using any evaluation of f or ∇f. Next, we prove the second property. We first start with $i = 0$ as the base case: $|\,\langle\mu_0|\mu_1\rangle\,| \geq \Omega(1)$. To prove this, Let $f(\mathbf{x}^*) = \min_{\mathbf{x}\in\Omega} f(\mathbf{x})$. We fix $\beta = 1$ without loss of generality. Then, we can write,

$$f(\mathbf{x}) \leq f(\mathbf{x}^*) + \langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^*\rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^*\|^2 \leq f(\mathbf{x}^*) + L\|\mathbf{x}^*\|^2 + L\|\mathbf{x}\|^2, \quad (2.46)$$

where the first inequality is well known due to Assumption 2.1.1 (see [Nes18]) and second inequality is due to Young's inequality. Using this upper bound on f, we have,

$$\begin{aligned}
|\,\langle\mu_0|\mu_1\rangle\,| &= \frac{\int \mathrm{d}\mathbf{x} \exp\left(-\frac{1}{2}f(\mathbf{x}) - \frac{\|\mathbf{x}\|^2}{2\sigma_1^2}\right)}{(2\pi\sigma_1^2)^{d/4}\sqrt{Z_1}} && (2.47)\\
&\geq \frac{\int \mathrm{d}\mathbf{x} \exp\left(-\frac{1}{2}f(\mathbf{x}^*) - \frac{1}{2}L\|\mathbf{x}\|^2 - \frac{1}{2}L\|\mathbf{x}^*\|^2 - \frac{\|\mathbf{x}\|^2}{2\sigma_1^2}\right)}{(2\pi\sigma_1^2)^{d/4}\sqrt{\int \mathrm{d}\mathbf{x} \exp\left(-f(\mathbf{x}^*) - \frac{\|\mathbf{x}\|^2}{4\sigma_1^2}\right)}} && (2.48)\\
&= \frac{\exp\left(-\frac{L}{2}\|\mathbf{x}^*\|^2\right)}{(2\pi\sigma_1^2)^{d/4}} \frac{\pi^{d/2}(L/2 + 1/(2\sigma_1^2))^{-d/2}}{(2\pi\sigma_1^2)^{d/4}} && (2.49)\\
&= \exp\left(-\frac{L}{2}\|\mathbf{x}^*\|^2\right)(L\sigma_1^2 + 1)^{-d/2} && (2.50)\\
&\geq \exp\left(-\frac{L}{2}\|\mathbf{x}^*\|^2 - \frac{dL\sigma_1^2}{2}\right). && (2.51)
\end{aligned}$$

Choosing $\sigma_1^2 = \frac{\epsilon}{2dL}$ yields $|\,\langle\mu_0|\mu_1\rangle\,| \geq \Omega(1)$. Next, we consider $1 \leq i \leq M-1$. Letting

$\sigma^2 = \sigma_{i+1}^2$, we have

$$|\langle \mu_i | \mu_{i+1} \rangle| = \int \mathrm{d}\mathbf{x} \frac{\exp(-f_i(\mathbf{x})/2)}{\sqrt{Z_i}} \frac{\exp(-f_{i+1}(\mathbf{x})/2)}{\sqrt{Z_{i+1}}} \tag{2.52}$$

$$= \int \mathrm{d}\mathbf{x} \frac{\exp\left(-f(\mathbf{x}) - \frac{\|\mathbf{x}\|^2}{4\sigma_i^2} - \frac{\|\mathbf{x}\|^2}{4\sigma_{i+1}^2}\right)}{\sqrt{Z_i Z_{i+1}}} \tag{2.53}$$

$$= \frac{\mathbb{E}_\pi \left[\exp\left(-\frac{1+\alpha/2}{2\sigma^2}\|\mathbf{x}\|^2\right)\right]}{\mathbb{E}_\pi \left[\exp\left(\frac{-1+\alpha}{2\sigma^2}\|\mathbf{x}\|^2\right)\right]^{1/2} \mathbb{E}_\pi \left[\exp\left(\frac{-1}{2\sigma^2}\|\mathbf{x}\|^2\right)\right]^{1/2}}, \tag{2.54}$$

where the last step follows from the fact that the numerator can be written as,

$$\int \mathrm{d}\mathbf{x} \exp\left(-f(\mathbf{x}) - \frac{\|\mathbf{x}\|^2}{4\sigma_i^2} - \frac{\|\mathbf{x}\|^2}{4\sigma_{i+1}^2}\right) = \frac{Z \int \mathrm{d}\mathbf{x} \exp\left(-f(\mathbf{x}) - \frac{1+\alpha/2}{2\sigma^2}\|\mathbf{x}\|^2\right)}{Z} \tag{2.55}$$

$$= Z\mathbb{E}_\pi \left[\exp\left(-\frac{1+\alpha/2}{2\sigma^2}\|\mathbf{x}\|^2\right)\right], \tag{2.56}$$

and similarly, Z_i and Z_{i+1} can be simplified as,

$$Z_i = \int \mathrm{d}\mathbf{x} e^{-f(\mathbf{x}) - \frac{1}{2\sigma_i^2}\|\mathbf{x}\|^2} = \frac{Z \int \mathrm{d}\mathbf{x} \exp\left(-f(\mathbf{x}) - \frac{(1+\alpha)}{2\sigma^2}\|\mathbf{x}\|^2\right)}{Z} \tag{2.57}$$

$$= Z\mathbb{E}_\pi \left[\exp\left(-\frac{(1+\alpha)}{2\sigma^2}\|\mathbf{x}\|^2\right)\right] \tag{2.58}$$

$$\tag{2.59}$$

and,

$$Z_{i+1} = \int \mathrm{d}\mathbf{x} e^{-f(\mathbf{x}) - \frac{1}{2\sigma_{i+1}^2}\|\mathbf{x}\|^2} = \frac{Z \int \exp\left(-f(\mathbf{x}) - \frac{1}{2\sigma^2}\|\mathbf{x}\|^2\right)}{Z} \tag{2.60}$$

$$= Z\mathbb{E}_\pi \left[\exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right)\right]. \tag{2.61}$$

Defining $\alpha' = \frac{\alpha}{\alpha+2}$ and $\sigma'^2 = \frac{\sigma^2}{1+\alpha/2}$, we have

$$|\langle \mu_i | \mu_{i+1} \rangle| = \frac{\mathbb{E}_\pi \left[\exp\left(\frac{-1}{2\sigma'^2}\|\mathbf{x}\|^2\right)\right]}{\mathbb{E}_\pi \left[\exp\left(-\frac{1+\alpha'}{2\sigma'^2}\|\mathbf{x}\|^2\right)\right]^{1/2} \mathbb{E}_\pi \left[\exp\left(-\frac{1-\alpha'}{2\sigma'^2}\|\mathbf{x}\|^2\right)\right]^{1/2}} \tag{2.62}$$

$$\geq \Omega(\exp\left(-2dL\alpha'^2/(mc_{\mathrm{LSI}}^2)\right)), \tag{2.63}$$

where the last inequality is due to *Lemma* 2.3.5. Setting $\alpha^2 = \tilde{\mathcal{O}}(c_{\mathrm{LSI}}^2 m/(dL))$, we have $|\langle \mu_i | \mu_{i+1} \rangle| \geq \Omega(1)$. Having established the second property, we move on to the third property.

$$|\langle \mu_M | \pi \rangle| = \int \mathrm{d}\mathbf{x} \frac{\exp\left(-f(\mathbf{x}) - \frac{\|\mathbf{x}\|^2}{4\sigma_M^2}\right)}{\sqrt{Z_M}\sqrt{Z}} \tag{2.64}$$

$$= \mathbb{E}_{\rho'}\left[\exp\left(-\frac{1}{4\sigma_M^2}\|\mathbf{x}\|^2\right)\right]^{-1/2} \mathbb{E}_{\rho'}\left[\exp\left(\frac{1}{4\sigma_M^2}\|\mathbf{x}\|^2\right)\right]^{-1/2} \tag{2.65}$$

$$\geq 1 - \Omega(dL/(m\sigma_M^4 c_{\mathrm{LSI}}^2)), \tag{2.66}$$

where $\rho' \propto \pi(\mathbf{x}) \exp\left(-\frac{\|\mathbf{x}\|^2}{4\sigma_M^2}\right)$. The last step is due to *Lemma* 2.3.4. Setting $\sigma_M^2 = \sqrt{dL/(mc_{\mathrm{LSI}}^2)}$ satisfies $|\langle \mu_M | \pi \rangle| \geq \Omega(1)$. The final property follows from the fact that $\alpha = \tilde{\mathcal{O}}(\sqrt{c_{\mathrm{LSI}}^2 m/(dL)})$, since,

$$\sigma_M = \sigma_0(1+\alpha)^M, \tag{2.67}$$

and solving this for M yields $M = \tilde{\mathcal{O}}(\sqrt{dL/(mc_{\mathrm{LSI}}^2)})$. □

We believe that our analysis for this annealing schedule for non-logconcave distributions can have other applications. For instance, [LZ24] used an annealing schedule to optimize approximately-convex functions and they showed applications for stochastic bandits. In their construction, they assumed that the objective function can be written as a uniform perturbation of a convex function in the entire domain. Our construction can allow design of optimization algorithms for more general non-convex functions as Assumption 2.1.1 and Assumption 2.1.2 are not too restrictive.

2.4 Quantum Algorithms for Sampling

We review the discrete quantum walk framework to *quantize* Markov chains.

2.4.1 Quantizing Markov Chains

Classical Markov chains can be quantized on a quantum computer using Szgedy's quantum walk operators introduced in [Sze04] by constructing a unitary operator on $\mathcal{H} = \mathbb{C}^N \otimes \mathbb{C}^N$. A quantum walk operator associated with a Markov chain with transition density P and

stationary density π is a mapping

$$|\mathbf{x}\rangle |0\rangle \mapsto |\psi_{\mathbf{x}}\rangle \tag{2.68}$$

where

$$|\psi_{\mathbf{x}}\rangle = \int_{\mathbf{y}} d\mathbf{y} \sqrt{P(\mathbf{x}, \mathbf{y})} |\mathbf{x}\rangle |\mathbf{y}\rangle \tag{2.69}$$

The unitary operator U can be realized as:

$$U \coloneqq S\Big(2 \int_{\mathbf{x}} |\psi_{\mathbf{x}}\rangle \langle\psi_{\mathbf{x}}| - I\Big), \tag{2.70}$$

where $S = \int_{\mathbf{xy}} |\mathbf{x}\rangle |\mathbf{y}\rangle \langle\mathbf{x}| \langle\mathbf{y}|$ is the swap operator. For reversible Markov chains, the spectrum of U can be related to the spectrum of P in the sense that each eigenvalue of P is mapped to a point on the unit disk. Then, using a simple geometry once can show that the phase gap of U is quadratically larger than the spectral gap of P. To understand how U is related to P, we use the discriminant matrix $D(P)$ associated with P. Let $\pi^{1/2}$ denote the diagonal matrix with entries $\sqrt{\pi(x)}$. We recall that the discriminant matrix D is defined by,

$$D(P) = \pi^{1/2} P \pi^{-1/2}. \tag{2.71}$$

It is straightforward to show that the quantum state $|\pi\rangle$ is an singular vector of D with singular value (left or right) 1.

$$D(P) |\pi\rangle = |\pi\rangle . \tag{2.72}$$

Therefore, the spectrum of P matches the spectrum of D since we obtain D by a similarity transformation. For reversible Markov chains, D is a symmetric matrix, hence its eigenvalues and singular values match. This follows from the detailed balance condition as follows:

$$D_{\mathbf{xy}} = \sum_{\mathbf{z},\mathbf{q}} \pi_{\mathbf{xz}}^{1/2} P_{\mathbf{zq}} \pi_{\mathbf{qy}}^{-1/2} \tag{2.73}$$

$$= \sqrt{\pi(\mathbf{x})} P_{\mathbf{xy}} \sqrt{\pi(\mathbf{y})} \tag{2.74}$$

$$= \sqrt{P_{\mathbf{xy}} P_{\mathbf{yx}}}. \tag{2.75}$$

Since D is symmetric, its eigenvectors form an orthogonal basis and from Perron-Frobenius theorem it has unique eigenvalue 1.

Then a reflection around the eigenvector of D with eigenvalue 1 on the relevant

subspace can be prepared using singular value transformations [GSLW19] or phase estimation [WCNA09] using $\frac{1}{\sqrt{\delta}}$ calls to a unitary operator. In fact, from a singular value processing perspective a quantum walk operator can be seen as a block-encoding operator for D. Once we have the appropriate reflection operators, we can use fixed point amplitude amplification technique introduced in [Gro05] to drive the initial state to the target state by applying reflection operators iteratively. It is also possible to use quantum Zeno effect to keep the quantum state close to the Gibbs density by using projective measurements after each phase estimation [SBB07, SBBK08], it results in worse dependency on the schedule length than the amplitude amplification.

For non-reversible Markov chains, the discriminant matrix is no longer a symmetric matrix. Therefore, there is no tight relation between the mixing time and the spectral gap of P. Furthermore, block-encoding for D cannot be constructed in general. Therefore, in general, one cannot expect to prepare $|\pi\rangle$ using quantum linear algebra techniques in a similar fashion. Alternatively, one can implement $\tilde{D}_{\mathbf{x},\mathbf{y}} = \sqrt{P_{\mathbf{xy}} P_{\mathbf{yx}}}$, however, this operator's spectrum is not related to P and $|\pi\rangle$ is no longer an eigenvector introducing additional challenges.

2.4.2 Implementing Quantum Walk Operators for Gibbs Sampling

We describe the implementation of quantum walk operator for stochastic case here. For full gradient case, we just need to replace the oracle to full gradient oracle. We use the stochastic gradient oracle $O_{\tilde{\nabla} f}$ to prepare the following state,

$$|\mathbf{x}\rangle |0\rangle \mapsto |\mathbf{x}\rangle \left|\tilde{\nabla} f(\mathbf{x})\right\rangle . \tag{2.76}$$

Since the transition density of unadjusted Langevin algorithm is Gaussian, then one step of the walk can be implemented efficiently on a quantum computer using first Box-Muller transformation [CCH+23] and applying a shift operation based on the gradient.

$$|\mathbf{x}\rangle |0\rangle \mapsto |\mathbf{x}\rangle \int_{\mathbb{R}^d} \mathrm{d}y \, \sqrt{p_{\mathbf{xy}}} \, |\mathbf{y}\rangle , \tag{2.77}$$

where

$$p_{\mathbf{xy}} = \left(\frac{1}{2\pi}\right)^{d/2} e^{-\frac{1}{2}\left\|\mathbf{y}-\eta\tilde{\nabla} f(\mathbf{x})\right\|_2^2}. \tag{2.78}$$

The query complexity of this operation is $\mathcal{O}(B)$ due to B gradient evaluations required to implement the oracle in Equation (2.76). Note that the we present the quantum states and operators in the continuous-space representation. The analysis in continuous-space simplifies the analysis, while the implementations are always in a discretized space (as we only have finite bits of precision for real numbers). We refer to [CCH$^+$23] for the error analysis caused by the discretization, which is not dominating other errors. For quantum MALA, we update the target register conditionally similar to [CLL$^+$22].

Remark 2.4.1. As opposed to a family of quantum machine learning algorithms where the data needs to be encoded into a quantum state, our sampling algorithms uses the data by the gradient and evaluation oracles which are simulated classically. Since the unitary evolutions only use constant number of calls to these oracles, our speedups are not suppressed by other hidden costs such as input preparations etc. However, the overall implementation of quantum sampling algorithm still requires a fault-tolerant quantum computer that can implement the phase estimation circuit with high fidelity. It is an open question whether these speedups can be implemented in near term quantum computers.

Next, we present our results for the sampling algorithms for continuous Gibbs distributions.

2.4.3 Quantum Metropolis Adjusted Langevin Algorithm

The following theorem establishes the query complexity of quantum MALA algorithm. Since it is a time reversible Markov chain, this result is obtained by characterizing the spectral gap of its transition density using conductance analysis. Then the phase gap of the quantum MALA algorithm scales as $\eta^{-1/2}$ for sufficiently small step size η. Once the phase gap is characterized, the rest of the proof is to combine it with the annealing schedule.

For technical reasons, we set the domain $\Omega = \mathbb{R}^d \cap B(0, R)$ where R is sufficiently large enough to show that the truncated distribution π^* in Ω is ϵ close to the original Gibbs distribution. More specifically, we work on sufficiently large but bounded domain to show that the norm of the gradients are bounded and derive our results in terms of R. The truncation is done by only considering the sum of projectors up to $\|x\| \leq R$ in the implementation of the quantum walk. Then we use the following lemma to characterize R,

Lemma 2.4.2 (Lemma 6 in [ZXG21])**.** *For any $\epsilon \in (0, 1)$ set $R = \bar{R}(\epsilon/12)$ and let π^* be the truncated distribution in Ω. Then the total variation distance between π^* and π is*

upper bounded by $\|\pi^* - \pi\| \leq \epsilon/4$, *where*

$$\bar{R}(z) = \left[\max\left\{\frac{625d\log(4/z)}{m\beta}, \frac{4d\log(4L/m)}{m\beta}, \frac{4d + 8\sqrt{d\log(1/z)} + 8\log(1/z)}{m\beta})\right\}\right]^{1/2}. \tag{2.79}$$

The lemma below characterizes the conductance parameter of the classical MALA algorithm constructed with stochastic gradients under given assumptions. Though similar results are given for full gradient case in [MCJ+19], we use the stochastic version and remove B dependent condition on the step size when we apply this lemma in full gradient case by setting $B \gg d$.

Lemma 2.4.3 (Lemma 6.5 in [ZXG21])**.** *Under Assumptions 2.1.1 and 2.1.2, if the step size meets the condition* $\eta \leq \min\{35(Ld + (LR+G)^2\beta d/B)]^{-1}, [25\beta(LR+G)^2]^{-1}\}$, *then there exists absolute constant* c_0 *such that, the conductance parameter* ϕ *for Metropolis adjusted Stochastic Langevin Algorithm satisfies,*

$$\phi \geq c_0\rho\sqrt{\eta/\beta}, \tag{2.80}$$

where ρ *is the Cheeger constant of the truncated distribution* π^*.

The following lemma is useful to characterize the phase gap of quantum walk operator for a reversible Markov chain in terms of its conductance parameter and it is the source of the quantum speed up for mixing time for reversible chains.

Lemma 2.4.4. *Let* Q *be a reversible Markov chain with conductance parameter* $\phi(Q)$ *and let eigenvalues for the transition density of* Q *be* $\lambda_0 = 1 > |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_m|$. *Let* W *be a unitary quantum walk operator constructed with the transition density of* Q. *Then the phase gap* $\Delta(W) \coloneqq 2\arccos|\lambda_1|$ *is lower bounded by,*

$$\Delta(W) \geq \sqrt{2}\phi(Q). \tag{2.81}$$

Proof. Let $\gamma(Q) = 1 - \lambda_1$ denote the spectral gap of Q. Using Cheeger's inequality [Che71], $\gamma(Q)$ can be bounded in terms of the conductance parameter,

$$\sqrt{2\gamma(Q)} \leq \phi(Q). \tag{2.82}$$

Let $\theta = \arccos|\lambda_1|$. Then we can write,

$$\Delta(W) \geq |1 - e^{2i\theta}| = 2\sqrt{1-\lambda_1^2} \geq 2\sqrt{\gamma(Q)}. \tag{2.83}$$

By combining Equation (2.82) and Equation (2.83), we obtain $\Delta(Q) \geq \sqrt{2}\phi(Q)$. □

Having established the phase gap of the quantum walk operator associated with quantum MALA algorithm, we are now ready to prove the following theorem.

Theorem 2.4.5 (Quantum MALA)**.** *Let $\pi \propto e^{-\beta f(\mathbf{x})}$ denote a probability distribution with inverse temperature $\beta > 0$ such that $f(\mathbf{x})$ satisfies Assumptions 2.1.1 and 2.1.2. Then, there exists a quantum algorithm that outputs a random variable distributed according to μ such that,*

$$\mathrm{TV}(\mu, \pi) \leq \epsilon, \tag{2.84}$$

using $\tilde{\mathcal{O}}\left(\beta d \rho^{-1} c_{\mathrm{LSI}}^{-1}\right)$ queries to $O_{\nabla f}$ and O_f.

Proof. Let $|\mu_0\rangle, |\mu_1\rangle, ..., |\mu_{M-1}\rangle$ be the series of quantum states described in Lemma 2.3.1. We start with the preparation of the initial Gaussian state $|\mu_0\rangle$ which can be done efficiently by applying the Box-Muller transformation to the uniform distribution state (see Appendix A.3 in [CCH$^+$23] for more details). Then, for each $i \in [0, M-2]$, we drive each state $|\mu_i\rangle$ to $|\mu_{i+1}\rangle$ using $\pi/3$-fixed-point amplitude amplification algorithm [Gro05]. The amplitude amplification uses the following reflection operators,

$$V_i = e^{i\pi/3} |\mu_i\rangle \langle\mu_i| + (I - |\mu_i\rangle \langle\mu_i|), \tag{2.85}$$

$$V_{i+1} = e^{i\pi/3} |\mu_{i+1}\rangle \langle\mu_{i+1}| + (I - |\mu_{i+1}\rangle \langle\mu_{i+1}|). \tag{2.86}$$

Each state $|\mu_i\rangle$ is the unique eigenvector of quantum MALA operator U_i^* for $f_i(\mathbf{x}) = f(\mathbf{x}) + \frac{\|\mathbf{x}\|^2}{2\sigma_i^2}$ since the classical MALA is time reversible and its stationary distribution is μ_i. Therefore, the operator $|\mu_i\rangle \langle\mu_i|$ is a projector operator to the eigenstate of U_i^* with eigenphase 0. Then, by Corollary 4.1 in [CCH$^+$23], the operator V_i can be implemented with ϵ accuracy using $\tilde{\mathcal{O}}(1/\Delta(U_i^*))$ calls to controlled-U^* operators where $\Delta(\cdot)$ is the phase gap. By Lemma 2.4.3, Lemma 2.4.2, and Lemma 2.4.4, $\Delta \geq \rho\sqrt{2\eta/\beta}$ for step size smaller than $\mathcal{O}(\min\{d^{-1}, \beta^{-1}\})$. Then, using $\tilde{\mathcal{O}}(\rho^{-1}\eta^{-1/2}\beta)$ calls to controlled- U_i^* operators, we can implement a quantum reflection $\tilde{V}_i$ such that,

$$\|\tilde{V}_i - V_i\| \leq \epsilon, \tag{2.87}$$

for each i. Then, given $|\mu_i\rangle$, we can drive $|\mu_i\rangle$ to $|\tilde{\mu}_{i+1}\rangle$ using constant number of $\tilde{V}_i$ operators because,

$$|\langle \mu_i|\tilde{\mu}_{i+1}\rangle| \geq \Omega(1), \tag{2.88}$$

such that $\| |\tilde{\mu}_{i+1}\rangle - |\mu_{i+1}\rangle \| \leq \epsilon$. Then we apply the same steps M times to drive μ_0 to π with at most error ϵ. Note that the error in each step does not accumulate linearly. This is because we can drive $\tilde{\mu}_i$ to $\tilde{\mu}_{i+1}$ with logarithmic cost in applying reflection operators. Since $M = \tilde{\mathcal{O}}(c_{\text{LSI}}^{-1}\sqrt{d})$, the total complexity of the annealing procedure is $\tilde{\mathcal{O}}(d^{1/2}c_{\text{LSI}}^{-1}\rho^{-1}\eta^{-1/2}\beta) = \tilde{\mathcal{O}}(c_{\text{LSI}}^{-1}\rho^{-1}\beta d)$. Each U^* operator can be implemented using constant number of calls to full gradient and evaluation oracles, the algorithm uses $\tilde{\mathcal{O}}(c_{\text{LSI}}^{-1}\rho^{-1}\beta d)$ full gradient and function evaluations. □

2.4.4 Quantum Unadjusted Langevin Algorithm

Since unadjusted Langevin algorithm is not a reversible chain, we cannot follow the same procedure since there is no direct relation between the spectral gap and the mixing time of the quantum algorithm. Our quantum algorithm follows the same procedure as in quantum MALA algorithm, however we implement the quantum walk operators using the transition density of ULA algorithm instead of MALA algorithm. Let U^* and U be the quantum walk operator associated with quantum MALA and quantum ULA respectively. The key idea in our proof is to show that for sufficiently small step size η, the operator norm of the difference $\|U^* - U\| \leq \tilde{\mathcal{O}}(\eta d)$. Then, using this one step error between quantum walks and due to the fact that error accumulates at most linearly with K, the total discrepancy between two algorithms becomes $\tilde{\mathcal{O}}(\eta d K)$ where K is the total number of calls to U^* in quantum MALA algorithm with the same step size. Finally, we set the step size sufficiently small so that the total error between two algorithms are smaller than ϵ. Since K is proportional to $1/\eta^{1/2}$, this allows us to characterize K.

Lemma 2.4.6. *Let U and $\tilde{U}$ be two quantum walk operators associated with two classical Markov chains on Ω with transition densities P and $\tilde{P}$, respectively. Then,*

$$\|U - \tilde{U}\| \leq 4\sqrt{2}\max_{\mathbf{x}} \mathrm{H}(P(\mathbf{x},\cdot), \tilde{P}(\mathbf{x},\cdot)), \tag{2.89}$$

where $\mathrm{H}(P(\mathbf{x},\cdot), \tilde{P}(\mathbf{x},\cdot))$ denotes the Hellinger distance between the probability densities $P(\mathbf{x},\cdot)$ and $\tilde{P}(\mathbf{x},\cdot)$ for any $\mathbf{x} \in \Omega$.

Proof. Let $p_{\mathbf{xy}} = P(\mathbf{x},\mathbf{y})$ and $\tilde{p}_{\mathbf{xy}} = \tilde{P}(\mathbf{x},\mathbf{y}) =$. We first define the following quantum

states,

$$|\psi_{\mathbf{x}}\rangle = \sum_{y} \sqrt{p_{\mathbf{xy}}}\, |\mathbf{x}\rangle\, |\mathbf{y}\rangle\,, \tag{2.90}$$

$$|\tilde{\psi}_{\mathbf{x}}\rangle = \sum_{y} \sqrt{\tilde{p}_{\mathbf{xy}}}\, |\mathbf{x}\rangle\, |\mathbf{y}\rangle\,. \tag{2.91}$$

Then, using the definition of quantum walk operators, the spectral norm of difference of operators can be bounded as,

$$\|U - \tilde{U}\| = \|S(2\sum_{\mathbf{x}\in\Omega} |\psi_{\mathbf{x}}\rangle\langle\psi_{\mathbf{x}}| - I) - S(2\sum_{\mathbf{x}\in\Omega} |\tilde{\psi}_{\mathbf{x}}\rangle\langle\tilde{\psi}_{\mathbf{x}}| - I)\| \tag{2.92}$$

$$\leq 2\Big\|\sum_{\mathbf{x}\in\Omega} |\psi_{\mathbf{x}}\rangle\langle\psi_{\mathbf{x}}| - \sum_{\mathbf{x}\in\Omega} \Big|\tilde{\psi}_{\mathbf{x}}\Big\rangle\Big\langle\tilde{\psi}_{\mathbf{x}}\Big|\Big\| \tag{2.93}$$

$$\leq 2\Big\|\sum_{\mathbf{x}\in\Omega} (|\psi_{\mathbf{x}}\rangle - |\tilde{\psi}_{\mathbf{x}}\rangle)\langle\psi_{\mathbf{x}}| + \sum_{\mathbf{x}\in\Omega} |\tilde{\psi}_{\mathbf{x}}\rangle\,(\langle\psi_{\mathbf{x}}| - \langle\tilde{\psi}_{\mathbf{x}}|)\Big\| \tag{2.94}$$

$$\leq 2\Big\|\sum_{\mathbf{x}\in\Omega} (|\psi_{\mathbf{x}}\rangle - |\tilde{\psi}_{\mathbf{x}}\rangle)\langle\psi_{\mathbf{x}}|\Big\| + 2\Big\|\sum_{\mathbf{x}\in\Omega} (|\psi_{\mathbf{x}}\rangle - |\tilde{\psi}_{\mathbf{x}}\rangle)\langle\tilde{\psi}_{\mathbf{x}}|\Big\|, \tag{2.95}$$

where the first inequality is due to unitarity of S and the third inequality is due to triangular inequality. Let $|\phi\rangle$ and $|\phi'\rangle$ are the states defined as the maximizers,

$$\Big\|\sum_{\mathbf{x}\in\Omega} (|\psi_{\mathbf{x}}\rangle - |\tilde{\psi}_{\mathbf{x}}\rangle)\langle\psi_{\mathbf{x}}|\Big\| = \max_{|\phi\rangle} \Big\|\sum_{\mathbf{x}\in\Omega} (|\psi_{\mathbf{x}}\rangle - |\tilde{\psi}_{\mathbf{x}}\rangle)\langle\psi_{\mathbf{x}}|\phi\rangle\Big\|, \tag{2.96}$$

and

$$\Big\|\sum_{\mathbf{x}\in\Omega} (|\psi_{\mathbf{x}}\rangle - |\tilde{\psi}_{\mathbf{x}}\rangle)\langle\tilde{\psi}_{\mathbf{x}}|\Big\| = \max_{|\phi'\rangle} \Big\|\sum_{\mathbf{x}\in\Omega} (|\psi_{\mathbf{x}}\rangle - |\tilde{\psi}_{\mathbf{x}}\rangle)\langle\tilde{\psi}_{\mathbf{x}}|\phi'\rangle\Big\|. \tag{2.97}$$

Notice that, for any $\mathbf{x} \in \Omega$, we have $\langle\tilde{\psi}_{\mathbf{x}}|\tilde{\psi}_{\mathbf{y}}\rangle = \delta_{\mathbf{xy}}$ and $\langle\psi_{\mathbf{x}}|\psi_{\mathbf{y}}\rangle = \delta_{\mathbf{xy}}$. Therefore, we can write $|\phi\rangle = \sum_{\mathbf{x}\in\Omega} c_{\mathbf{x}}\,|\psi_{\mathbf{x}}\rangle + |\xi\rangle$ and $|\phi'\rangle = \sum_{\mathbf{x}\in\Omega} \tilde{c}_{\mathbf{x}}\,|\tilde{\psi}_{\mathbf{x}}\rangle + |\tilde{\xi}\rangle$ where $\langle\tilde{\xi}|\tilde{\psi}_{\mathbf{x}}\rangle = \langle\xi|\psi_{\mathbf{x}}\rangle = 0$ for all $\mathbf{x} \in \Omega$. Hence,

$$\|U - \tilde{U}\| \leq 2\Big\|\sum_{\mathbf{x}} c_{\mathbf{x}}(\,|\tilde{\psi}_{\mathbf{x}}\rangle - |\psi_{\mathbf{x}}\rangle)\Big\| + 2\Big\|\sum_{\mathbf{x}} \tilde{c}_{\mathbf{x}}(\,|\tilde{\psi}_{\mathbf{x}}\rangle - |\psi_{\mathbf{x}}\rangle)\Big\| \tag{2.98}$$

$$\leq 4\max_{\mathbf{x}} \|\,|\tilde{\psi}_{\mathbf{x}}\rangle - |\psi_{\mathbf{x}}\rangle\,\|. \tag{2.99}$$

Finally, we can write,

$$\| |\tilde{\psi}_{\mathbf{x}}\rangle - |\psi_{\mathbf{x}}\rangle \| = \left\| \sum_{y} (\sqrt{p_{\mathbf{xy}}} - \sqrt{\tilde{p}_{\mathbf{xy}}}) \, |\mathbf{x}\rangle \, |\mathbf{y}\rangle \right\| \tag{2.100}$$

$$= \left(\sum_{\mathbf{y}} (\sqrt{p_{\mathbf{xy}}} - \sqrt{\tilde{p}_{\mathbf{xy}}})^2 \right)^{1/2} \tag{2.101}$$

$$\leq \sqrt{2} \, \mathrm{H}(P(\mathbf{x}, .), \tilde{P}(\mathbf{x}, .)), \tag{2.102}$$

where the last step follows from the definition of Hellinger distance. □

To be able to apply Lemma 2.4.6, we bound the Hellinger distance between the probability distributions of MALA and ULA algorithm through the following lemma.

Lemma 2.4.7. *Let P and P^* be the transition densities for Unadjusted Langevin Algorithm (ULA) and Metropolis Adjusted Langevin Algorithm (MALA) respectively. Then under Assumptions 2.1.1 and 2.1.2 and for step size $\eta \leq d(\beta(LR+G)^2)^{-1}$,*

$$\max_{\mathbf{x}} \mathrm{H}(P(\mathbf{x}, .), \tilde{P}(\mathbf{x}, .)) \leq 4\eta d L, \tag{2.103}$$

where G is a positive constant that satisfies $\|\nabla f(0)\| \leq G$.

Proof. Let $p_{\mathbf{xy}} = P(\mathbf{x}, \mathbf{y})$ and $p^*_{\mathbf{xy}} = P^*(\mathbf{x}, \mathbf{y}) =$. For the sake of the proof, we use the lazy version of the Markov chains as it does not change the stationary density. Let $q_{\mathbf{xy}} = \frac{1}{(4\pi\eta/\beta)^{d/2}} \exp\Big(-\frac{\|\mathbf{y}-\mathbf{x}+\eta\nabla f(\mathbf{x})\|^2}{2\eta/\beta}\Big)$, then we can write

$$p_{\mathbf{xy}} = \frac{1}{2}\delta_{\mathbf{xy}} + \frac{1}{2} q_{\mathbf{xy}}, \tag{2.104}$$

and

$$p^*_{\mathbf{xy}} = \left\{ \begin{array}{ll} \alpha_{\mathbf{x}}(\mathbf{y}) p_{\mathbf{xy}}, & \text{if } \mathbf{x} \neq \mathbf{y} \\ p_{\mathbf{xy}} + \sum\limits_{\mathbf{z} \in \Omega} p_{\mathbf{xz}} (1 - \alpha_x(\mathbf{z})) & \text{if } \mathbf{x} = \mathbf{y} \end{array} \right\}, \tag{2.105}$$

where $\delta_{\mathbf{xy}}$ is Kronecker delta function and $\alpha_{\mathbf{x}}(\mathbf{y})$ is the acceptance probability given by

$$\alpha_{\mathbf{x}}(\mathbf{y}) = \min \left(1, \frac{\exp\Big(-\beta f(\mathbf{y}) - \frac{\|\mathbf{x}-\mathbf{y}+\eta\nabla f(\mathbf{y})\|^2}{4\eta/\beta}\Big)}{\exp\Big(-\beta f(\mathbf{x}) - \frac{\|\mathbf{y}-\mathbf{x}+\eta\nabla f(\mathbf{x})\|^2}{4\eta/\beta}\Big)} \right). \tag{2.106}$$

By this definition, $\alpha_{\mathbf{x}}(\mathbf{y}) \leq 1$. Suppose $\alpha_{\mathbf{x}}(\mathbf{y}) \geq 1 - e(\mathbf{x}, \mathbf{y})$. Then for $\mathbf{x} \neq \mathbf{y}$,

$$(\sqrt{p^*_{\mathbf{xy}}} - \sqrt{p_{\mathbf{xy}}})^2 = p_{\mathbf{xy}} (1 - \sqrt{\alpha_{\mathbf{xy}}})^2 \tag{2.107}$$

$$\leq p_{\mathbf{xy}}(1-\sqrt{1-e(\mathbf{x},\mathbf{y})})^2 \tag{2.108}$$
$$\leq p_{\mathbf{xy}} e(\mathbf{x},\mathbf{y})^2, \tag{2.109}$$

where the second inequality is due to the fact that for $0 \leq u \leq 1$,

$$1-\sqrt{1-u} = \frac{(1-\sqrt{1-u})(1+\sqrt{1-u})}{(1+\sqrt{1+u})} = \frac{1-(1-u)}{1+\sqrt{1+u}} = \frac{u}{1+\sqrt{1+u}} \leq u. \tag{2.110}$$

For $\mathbf{x}=\mathbf{y}$,

$$(\sqrt{p^*_{\mathbf{xy}}}-\sqrt{p_{\mathbf{xy}}})^2 = p_{\mathbf{xy}}\left(\sqrt{1+\frac{1-\mathbb{E}_{p_{\mathbf{xy}}}(\alpha_{\mathbf{x}}(\mathbf{y}))}{p_{\mathbf{xy}}}}-1\right)^2 \tag{2.111}$$
$$\leq p_{\mathbf{xy}}\left(1+\frac{1-\mathbb{E}_{p_{\mathbf{xy}}}(\alpha_{\mathbf{x}}(\mathbf{y}))}{2p_{\mathbf{xy}}}-1\right)^2 \tag{2.112}$$
$$\leq \frac{(1-\mathbb{E}_{p_{\mathbf{xy}}}(\alpha_{\mathbf{x}}(\mathbf{y})))^2}{4p_{\mathbf{xy}}} \tag{2.113}$$
$$\leq \frac{\mathbb{E}_{p_{\mathbf{xy}}}(e(\mathbf{x},\mathbf{y}))^2}{2}, \tag{2.114}$$

where the second inequality follows from $\sqrt{1+u} \leq 1+\frac{u}{2}$ for $u \geq 0$ and the third inequality holds since $p_{\mathbf{xy}} \geq \frac{1}{2}$ for $\mathbf{x}=\mathbf{y}$ because of laziness of the Markov chains. Therefore,

$$\int_{\mathbf{y}\in\Omega}(\sqrt{p^*_{\mathbf{xy}}}-\sqrt{p}_{\mathbf{xy}})^2\,\mathrm{d}\mathbf{y} = \int_{\mathbf{y}\in\Omega}\delta_{\mathbf{xy}}(\sqrt{p^*_{\mathbf{xy}}}-\sqrt{p}_{\mathbf{xy}})^2\,\mathrm{d}\mathbf{y} + \int_{\mathbf{y}\in\Omega}(1-\delta_{\mathbf{xy}})(\sqrt{p^*_{\mathbf{xy}}}-\sqrt{p}_{\mathbf{xy}})^2\,\mathrm{d}\mathbf{y} \tag{2.115}$$
$$\leq \mathbb{E}_{p_{\mathbf{xy}}}(e(\mathbf{x},\mathbf{y})^2) + \frac{\mathbb{E}_{p_{\mathbf{xy}}}(e(\mathbf{x},\mathbf{y}))^2}{2} \tag{2.116}$$
$$\leq \frac{\mathbb{E}_{q_{\mathbf{xy}}}(e(\mathbf{x},\mathbf{y})^2)}{2} + \frac{\mathbb{E}_{q_{\mathbf{xy}}}(e(\mathbf{x},\mathbf{y}))^2}{8}, \tag{2.117}$$

where the extra factors of 1/2 and 1/4 in the second inequality comes from the laziness of the chain. Now, we need to bound $e(\mathbf{x},\mathbf{y})$. Starting from

$$\alpha_{\mathbf{x}}(\mathbf{y}) \geq \frac{\exp\left(-\beta f(\mathbf{y}) - \frac{\|\mathbf{x}-\mathbf{y}+\eta\nabla f(\mathbf{y})\|^2}{4\eta/\beta}\right)}{\exp\left(-\beta f(\mathbf{x}) - \frac{\|\mathbf{y}-\mathbf{x}+\eta\nabla f(\mathbf{x})\|^2}{4\eta/\beta}\right)} \tag{2.118}$$
$$= \exp\left(-\beta(f(\mathbf{y})-f(\mathbf{x}) - \frac{2\eta\langle\mathbf{y}-\mathbf{x},\nabla f(\mathbf{y})+\nabla f(\mathbf{x})\rangle + \eta^2\|\nabla f(\mathbf{y})\|^2 - \eta^2\|\nabla f(\mathbf{x})\|^2}{4\eta})\right) \tag{2.119}$$

$$\geq \exp\left(-\frac{\beta L\|\mathbf{x}-\mathbf{y}\|^2}{2} - \frac{\beta\eta^2\|\nabla f(\mathbf{y})\|^2 - \eta^2\|\nabla f(\mathbf{x})\|^2}{4\eta}\right)) \tag{2.120}$$

$$\geq \exp\left(-\frac{\beta L\|\mathbf{x}-\mathbf{y}\|^2}{2} - \frac{\beta\eta L(LR+G)\|\mathbf{x}-\mathbf{y}\|}{2}\right) \tag{2.121}$$

$$\geq 1 - \frac{\beta L\|\mathbf{x}-\mathbf{y}\|^2}{2} - \frac{\beta\eta L(LR+G)\|\mathbf{x}-\mathbf{y}\|}{2}. \tag{2.122}$$

The second inequality holds because of the smoothness of $f(\mathbf{x})$ since,

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \langle \mathbf{y}-\mathbf{x}, \nabla f(\mathbf{x})\rangle + \frac{L\|\mathbf{x}-\mathbf{y}\|^2}{2}, \tag{2.123}$$

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \mathbf{x}-\mathbf{y}, \nabla f(\mathbf{y})\rangle + \frac{L\|\mathbf{x}-\mathbf{y}\|^2}{2}, \tag{2.124}$$

which implies the following inequality

$$|f(\mathbf{y}) - f(\mathbf{x}) - \frac{1}{2}\langle \mathbf{y}-\mathbf{x}, \nabla f(\mathbf{x}) + \nabla f(\mathbf{y})\rangle| \leq \frac{L\|\mathbf{x}-\mathbf{y}\|^2}{2}. \tag{2.125}$$

To obtain the third inequality, we use Lemma 2.3.3 to show that,

$$\|\nabla f(\mathbf{x})\| \leq G + L\|\mathbf{x}\| \leq LR + G, \tag{2.126}$$

where the last inequality is due to fact that the domain is a ball with radius R. Then,

$$\|\nabla f(\mathbf{x})\|^2 - \|\nabla f(\mathbf{y})\|^2 = \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|\|\nabla f(\mathbf{x}) + \nabla f(\mathbf{y})\| \leq 2(LR+G)L\|\mathbf{x}-\mathbf{y}\|. \tag{2.127}$$

Consequently, $e(\mathbf{x},\mathbf{y}) \leq \frac{\beta L\|\mathbf{x}-\mathbf{y}\|^2}{2} + \frac{\beta\eta L(LR+G)\|\mathbf{x}-\mathbf{y}\|}{2}$. Finally, we need to bound

$$\int (\sqrt{p^*_{\mathbf{xy}}} - \sqrt{p}_{\mathbf{xy}})^2 \,\mathrm{d}\mathbf{y} \leq \frac{\mathbb{E}_{q(\mathbf{x},\cdot)}(e(x,y)^2)}{2} + \frac{\mathbb{E}_{q(\mathbf{x},\cdot)}(e(\mathbf{x},\mathbf{y}))^2}{8} \tag{2.128}$$

$$\leq \frac{5}{8}\mathbb{E}_{q_{\mathbf{xy}}}(e(\mathbf{x},\mathbf{y})^2) \tag{2.129}$$

$$\leq \frac{5}{8}\mathbb{E}_{q_{\mathbf{xy}}}\left(\frac{\beta L\|\mathbf{x}-\mathbf{y}\|^2}{2} + \frac{\beta\eta L(LR+G)\|\mathbf{x}-\mathbf{y}\|}{2}\right)^2 \tag{2.130}$$

$$\leq \frac{5}{8}\beta^2 L^2\mathbb{E}_{q_{\mathbf{xy}}}\|\mathbf{x}-\mathbf{y}\|^4 + \frac{5}{8}\beta^2\eta^2 L^2(LR+G)^2\mathbb{E}_{q_{\mathbf{xy}}}\|\mathbf{x}-\mathbf{y}\|^2, \tag{2.131}$$

where the second inequality uses Jensen's inequality due to convexity of $e(\mathbf{x},\mathbf{y})$ and the last inequality is due to Young's inequality. Next, we need to compute the expectation

values. Notice that since $q_{\mathbf{xy}}$ is a Gaussian, the variable $\frac{\beta\|\mathbf{y}-\mathbf{x}+\nabla f(\mathbf{x})\|^2}{\eta}$ is a chi-squared distributed random variable with mean d and variance $2d$.

$$\begin{aligned}
\mathbb{E}_{q_{\mathbf{xy}}}\|\mathbf{x}-\mathbf{y}\|^2 &= \mathbb{E}_{q_{\mathbf{xy}}}\|\mathbf{x}-\mathbf{y}+\eta\nabla f(\mathbf{x})-\eta\nabla f(\mathbf{x})\|^2 && (2.132)\\
&\leq 2\mathbb{E}_{q_{\mathbf{xy}}}\|\mathbf{x}-\mathbf{y}-\eta\nabla f(\mathbf{x})\|^2+2\eta^2\mathbb{E}_{q_{\mathbf{xy}}}\|\nabla f(\mathbf{x})\|^2 && (2.133)\\
&\leq 2\eta d/\beta+2\eta^2(LR+G)^2 && (2.134)\\
&\leq 4\eta d/\beta, && (2.135)
\end{aligned}$$

since the mean of chi squared distribution is d and $\eta \leq \frac{d}{\beta(LR+G)^2}$. Furthermore,

$$\begin{aligned}
\mathbb{E}_{q_{\mathbf{xy}}}\|\mathbf{x}-\mathbf{y}\|^4 &= \mathrm{Var}_{q_{\mathbf{xy}}}\|\mathbf{x}-\mathbf{y}\|^2+(\mathbb{E}_{q_{\mathbf{xy}}}\|\mathbf{x}-\mathbf{y}\|^2)^2 && (2.136)\\
&\leq 2d\eta^2/\beta^2+(4\eta d/\beta)^2 && (2.137)\\
&\leq 2d\eta^2/\beta^2+16\eta^2d^2/\beta^2, && (2.138)
\end{aligned}$$

since variance of chi squared distribution is $2d$. Putting things together, we have for any $\mathbf{x}\in\Omega$,

$$\begin{aligned}
\mathrm{H}(P(\mathbf{x},\cdot),\tilde{P}(\mathbf{x},\cdot))^2 &\leq \frac{5d\eta^2L^2}{4}+10\eta^2d^2L^2+\frac{5\eta^3d\beta L^2(LR+G)^2}{2} && (2.139)\\
&\leq 16\eta^2L^2d^2, && (2.140)
\end{aligned}$$

for $\eta \leq \frac{d}{\beta(LR+G)^2}$. Hence, $\mathrm{H}(P(\mathbf{x},\cdot),\tilde{P}(\mathbf{x},\cdot)) \leq 4\eta dL$. □

As the quantum walk operator is the basic building block of the reflection operators used in amplitude amplification, we present the following result to relate the error in quantum walk operator to the projection operators.

Lemma 2.4.8. *Let W be a unitary operator with phase gap Δ and assume that W has a unique eigenvector $|\psi_0\rangle$ with eigenvalue 1. Suppose that we have $\tilde{W}$ such that,*

$$\|W-\tilde{W}\| \leq \delta. \tag{2.141}$$

Let $\Pi_{<\Delta}$ and $\tilde{\Pi}_{<\Delta}$ be operators that project any quantum state onto the space of eigenvectors of W and $\tilde{W}$ with phases smaller than Δ respectively. Then,

$$\|\Pi_{<\Delta}-\tilde{\Pi}_{<\Delta}\| \leq \frac{\delta\pi}{4\Delta}. \tag{2.142}$$

Proof. Let $W = \sum_m e^{2i\phi_m} |\psi_m\rangle \langle\psi_m|$ where $\phi_0 = 0$. Similarly, let $\tilde{W} = \sum_m e^{2i\tilde{\phi}_m} |\tilde{\psi}_m\rangle \langle\tilde{\psi}_m|$.

$$\|W\psi_0 - \tilde{W}\psi_0\|^2 = \left\| \sum_m \left(1 - e^{2i\tilde{\phi}_m}\right) |\tilde{\psi}_m\rangle \langle\tilde{\psi}_m|\psi_0\rangle \right\|^2 \tag{2.143}$$

$$= \sum_m |1 - e^{2i\tilde{\phi}_m}|^2 \left| \langle\tilde{\psi}_m|\psi_0\rangle \right|^2 \tag{2.144}$$

$$\geq \sum_{m:\tilde{\phi}_m \geq \Delta} |1 - e^{2i\tilde{\phi}_m}|^2 \left| \langle\tilde{\psi}_m|\psi_0\rangle \right|^2 \tag{2.145}$$

$$\geq 16\Delta^2/\pi^2 \sum_{m:\tilde{\phi}_m \geq \Delta} \left| \langle\tilde{\psi}_m|\psi_0\rangle \right|^2, \tag{2.146}$$

where the second inequality is due to $|1 - e^{ix}| \geq 2|x|/\pi$ whenever $-\pi \leq x \leq \pi$. Since $\|W - \tilde{W}\| \leq \delta$, we have

$$\sum_{m:\tilde{\phi}_m < \Delta} \left| \langle\tilde{\psi}_m|\psi_0\rangle \right|^2 \geq 1 - \frac{\delta^2\pi^2}{16\Delta^2}. \tag{2.147}$$

Let $|\chi\rangle = \alpha_0 |\psi_0\rangle + \alpha_1 \left|\psi_0^\perp\right\rangle$ be an arbitrary quantum state such that $\alpha_1, \alpha_2 \in \mathbb{C}$ and $|\alpha_1|^2 + |\alpha_2|^2 = 1$. Then due to triangular inequality

$$\|\Pi_{<\Delta} |\chi\rangle - \tilde{\Pi}_{<\Delta} |\chi\rangle\| \leq |\alpha_0| \|\Pi_{<\Delta} |\psi_0\rangle - |\tilde{\Pi}_{<\Delta} |\psi_0\rangle\| + |\alpha_1| \|\Pi_{<\Delta} \left|\psi_0^\perp\right\rangle - \tilde{\Pi}_{<\Delta} \left|\psi_0^\perp\right\rangle\|. \tag{2.148}$$

We first focus on the first term:

$$\|\Pi_{<\Delta} |\psi_0\rangle - \tilde{\Pi}_{<\Delta} |\psi_0\rangle\| = \| |\psi_0\rangle - \sum_{m:\tilde{\phi}_m < \Delta} |\tilde{\psi}_m\rangle \langle\tilde{\psi}_m|\psi_0\rangle \| \tag{2.149}$$

$$= \left(2 - 2 \sum_{m:\tilde{\phi}_m < \Delta} \left| \langle\tilde{\psi}_m|\psi_0\rangle \right|^2\right)^{1/2} \tag{2.150}$$

$$\leq \frac{\delta\pi}{4\Delta}. \tag{2.151}$$

Similarly, for the second term,

$$\| |\Pi_{<\Delta} \left|\psi_0^\perp\right\rangle - \tilde{\Pi}_{<\Delta} \left|\psi_0^\perp\right\rangle \| = \left\| \sum_{m:\tilde{\phi}_m < \Delta} |\tilde{\psi}_m\rangle \langle\tilde{\psi}_m|\psi_0^\perp\rangle \right\| \tag{2.152}$$

$$= \left(\sum_{m:\tilde{\phi}_m < \Delta} \left| \left\langle\tilde{\psi}_m \middle| \psi_0^\perp\right\rangle \right|^2 \right)^{1/2} \tag{2.153}$$

$$= \left(1 - \sum_{m:\tilde{\phi}_m \geq \Delta} |\left\langle \tilde{\psi}_m \middle| \psi_0 \right\rangle|^2\right)^{1/2} \tag{2.154}$$

$$\leq \frac{\delta\pi}{4\Delta}. \tag{2.155}$$

Since both terms are smaller than $\frac{\delta\pi}{4\Delta}$, we conclude that for any state $|\chi\rangle$, the projectors are at most $\delta\pi/(4\Delta)$ apart in spectral norm. □

Next lemma quantifies the number of required controlled-U operators to implement the reflection operators.

Lemma 2.4.9. *Let U be the quantum walk operator associated with Unadjusted Langevin algorithm. Under Assumptions 2.1.1 and 2.1.2 the reflection operator $V = e^{i\pi/3} |\pi\rangle \langle\pi| + (I - |\pi\rangle \langle\pi|)$ can be implemented with ϵ accuracy in spectral norm using $\tilde{\mathcal{O}}(\rho^{-1}\beta d L \epsilon^{-1})$ controlled-U operators.*

Proof. Let P^* and P be the transition density of Metropolis Adjusted Langevin algorithm and Unadjusted Langevin algorithm respectively. Let U^* and U be the quantum walk operators built using P^* and P respectively. We can write U^* in spectral form:

$$U^* = \sum_m e^{2i\phi_m} |\psi_m\rangle \langle\psi_m| . \tag{2.156}$$

The phase gap Δ of U^* is defined to be $2|\phi_1|$. Since P^* is a reversible Markov chain, U^* accepts $|\pi\rangle$ as its eigenvector with eigenvalue 1. Furthermore, $|\pi\rangle$ is the unique eigenvector of U^* with eigenvalue 1 (see [MNRS07] for more details). Notice that, R can be written as,

$$V = e^{i\pi/3}\Pi^*_\Delta + (I - \Pi^*_\Delta), \tag{2.157}$$

where Π^*_Δ is the projector that projects any quantum state onto the eigenstate of U^* with eigenphase smaller than Δ. This is because the only eigenvector of U^* with phase smaller than Δ is $|\pi\rangle$. This operator can be implemented in ϵ accuracy using techniques such as quantum singular value transformation technique introduced in [GSLW19] or phase estimation based method ([MNRS07]) using $\tilde{\mathcal{O}}(1/\Delta)$ calls to quantum walk operator. Suppose that we replaced each U^* with U and implement the following operator instead:

$$\tilde{V} = e^{i\pi/3}\Pi_\Delta + (I - \Pi_\Delta), \tag{2.158}$$

where Π is the projector similarly defined for U which is the quantum walk operator constructed for the unadjusted Langevin algorithm. Therefore, we can characterize the

error,

$$\|V - \tilde{V}\| \leq 2\|\Pi^*_\Delta - \Pi_\Delta\| \tag{2.159}$$

$$\leq \frac{\|U - U^*\|}{2\Delta/\pi}. \tag{2.160}$$

The last inequality follows from Lemma 2.4.8. By Lemma 2.4.4 and Lemma 2.4.3, $\Delta(U^*) \geq c_0\rho\sqrt{2\eta/\beta}$ for step size smaller than $O(d^{-1}\beta^{-1})$. Therefore,

$$\|V - \tilde{V}| \leq c_0 16\sqrt{2}\pi\eta dL/\Delta \tag{2.161}$$

$$\leq c_0 16\pi\sqrt{2\eta\beta}dL/\rho, \tag{2.162}$$

where the first inequality is due to *Lemma* 2.4.6 and Lemma 2.4.7. Therefore, by setting $\eta \leq \frac{\epsilon^2\rho^2}{c_0 16\sqrt{2}\pi d^2L^2\beta}$, we have

$$\|V - \tilde{V}\| \leq \epsilon. \tag{2.163}$$

The total number of calls to U is $\tilde{\mathcal{O}}(1/\Delta) = \tilde{\mathcal{O}}(\rho^{-1}\eta^{-1/2}/\beta^{-1/2}) = \tilde{\mathcal{O}}(\rho^{-1}\beta dL/\epsilon)$. □

We are now ready to prove the query complexity of quantum ULA algorithm.

Theorem 2.4.10 (Quantum ULA)**.** *Let $\pi \propto e^{-\beta f(\mathbf{x})}$ denote a probability distribution with inverse temperature $\beta > 0$ such that $f(\mathbf{x})$ satisfies Assumptions 2.1.1 and 2.1.2. Then, there exists a quantum algorithm that outputs a random variable distributed according to μ such that,*

$$\mathrm{TV}(\mu, \pi) \leq \epsilon, \tag{2.164}$$

using $\tilde{\mathcal{O}}\left(\beta d^{3/2}\epsilon^{-1}\rho^{-1}c_{\mathrm{LSI}}^{-1}\right)$ queries to $O_{\nabla f}$.

Proof. Let $P^*(\mathbf{x}, \mathbf{y}) = p^*_{\mathbf{xy}}$ and $P(\mathbf{x}, \mathbf{y}) = p_{\mathbf{xy}}$ denote the transition densities of MALA and ULA algorithms respectively. Similarly, let U^* and U be the quantum walk operators associated with P^* and P constructed. We use the same algorithm described in proof of quantum MALA algorithm. That is, we iteratively drive each state $|\mu_i\rangle$ to $|\mu_{i+1}\rangle$ using $\pi/3$ fixed point amplitude amplification algorithm. However, since accessing U^* requires evaluation oracle, we instead use U to implement the reflection operator inexactly. The reflection operators can be implemented using $\tilde{\mathcal{O}}(\rho^{-1}\beta dL\epsilon^{-1})$ calls to controlled U operator by Lemma 2.4.9. Since the length of annealing schedule in Lemma 2.3.1 is

$\tilde{\mathcal{O}}(c_{\mathrm{LSI}}^{-1}\sqrt{d})$, the total complexity is $\tilde{\mathcal{O}}(c_{\mathrm{LSI}}^{-1}\rho^{-1}d^{3/2}\beta\epsilon^{-1})$. Implementing U only requires full gradient oracle constant number of times, we establish the result. □

2.4.5 Quantum Unadjusted Langevin Algorithm with Stochastic Gradients

The construction for the stochastic quantum ULA algorithm is similar to quantum ULA. The stochastic ingredient here is realized by replacing full gradient ∇f with a stochastic gradient $g_\ell = \frac{1}{B}\sum_{k\in S^\ell}\nabla f_k$ in implementing the quantum walk operator where S_ℓ is a batch randomly uniformly drawn from the set $\{A \subseteq [N] : |A| = B\}$. The next theorem quantifies the query complexity of stochastic quantum Langevin algorithm with respect to stochastic gradient oracle. The proof of the query complexity is similar to ULA, however, due to noisy gradients, we need to use matrix concentration to show that the quantum walks are close to each other with high probability for sufficiently small step size.

The next lemma characterizes the expectation value of U_ℓ over ℓ with respect to a deterministic unitary U.

Lemma 2.4.11. *Let $U_\ell = S\Big(2\sum_{\mathbf{x}} |\psi_{\mathbf{x}}^{(\ell)}\rangle\langle\psi_{\mathbf{x}}^{(\ell)}| - I\Big)$ be a quantum walk operator where $|\psi_{\mathbf{x}}^{(\ell)}\rangle = \sum_{\mathbf{y}}\sqrt{p_{\mathbf{xy}}^{(\ell)}}\,|\mathbf{y}\rangle$ is a quantum state constructed with stochastic gradient g_ℓ. Let $U = S\Big(2\sum_{\mathbf{x}}|\psi_{\mathbf{x}}\rangle\langle\psi_{\mathbf{x}}| - I\Big)$. Then, we have*

$$\|\mathbb{E}_\ell U_\ell - U\| \leq 6\max_{\mathbf{x}\in\Omega}\|\mathbb{E}_\ell\,|\psi_{\mathbf{x}}^{(\ell)}\rangle - |\psi_{\mathbf{x}}\rangle\,\|. \tag{2.165}$$

Proof.

$$\|\mathbb{E}_\ell U_\ell - U\| \leq 2\Big\|\mathbb{E}_\ell\sum_{\mathbf{x}}|\psi_{\mathbf{x}}^{(\ell)}\rangle\,\langle\psi_{\mathbf{x}}^{(\ell)}| - \sum_{\mathbf{x}}|\psi_{\mathbf{x}}\rangle\langle\psi_{\mathbf{x}}|\Big\| \tag{2.166}$$

$$= 2\Big\|\mathbb{E}_\ell\sum_{\mathbf{x}}|\psi_{\mathbf{x}}^{(\ell)}\rangle\,(\,\langle\psi_{\mathbf{x}}^{(\ell)}| - \langle\psi_{\mathbf{x}}|) + \sum_{\mathbf{x}}(\,|\psi_{\mathbf{x}}^{(\ell)}\rangle - |\psi_{\mathbf{x}}\rangle)\,\langle\psi_{\mathbf{x}}|\Big\| \tag{2.167}$$

$$\leq 2\Big\|\mathbb{E}_\ell\sum_{\mathbf{x}}|\psi_{\mathbf{x}}^{(\ell)}\rangle\,(\,\langle\psi_{\mathbf{x}}^{(\ell)}| - \langle\psi_{\mathbf{x}}|)\Big\| + 2\Big\|\sum_{\mathbf{x}}(\,|\psi_{\mathbf{x}}^{(\ell)}\rangle - |\psi_{\mathbf{x}}\rangle)\,\langle\psi_{\mathbf{x}}|\Big\|, \tag{2.168}$$

where the second inequality follows from triangular inequality. First, we focus on the second term,

$$\Big\|\mathbb{E}_\ell\sum_{\mathbf{x}}(\,|\psi_{\mathbf{x}}^{(\ell)}\rangle - |\psi_{\mathbf{x}}\rangle)\,\langle\psi_{\mathbf{x}}|\Big\| = \max_{|\phi\rangle}\Big\|\mathbb{E}_\ell\sum_{\mathbf{x}}(\,|\psi_{\mathbf{x}}^{(\ell)}\rangle - |\psi_{\mathbf{x}}\rangle)\,\langle\psi_{\mathbf{x}}|\phi\rangle\Big\|. \tag{2.169}$$

We can expand the state that maximizes this equation as $|\phi\rangle = \sum_{\mathbf{x}} c_{\mathbf{x}} |\psi_{\mathbf{x}}\rangle + |\xi\rangle$ where $\langle \xi|\psi_{\mathbf{x}}\rangle = 0$ for any $\mathbf{x} \in \Omega$. This is true because $\langle \psi_{\mathbf{x}}|\psi_{\mathbf{y}}\rangle = \delta_{\mathbf{xy}}$. Therefore, $\langle \psi_{\mathbf{x}}|\phi\rangle = c_{\mathbf{x}}$. Then,

$$\left\| \mathbb{E}_\ell \sum_{\mathbf{x}} (\, |\psi_{\mathbf{x}}^{(\ell)}\rangle - |\psi_{\mathbf{x}}\rangle) \, \langle \psi_{\mathbf{x}}| \right\| = \left\| \mathbb{E}_\ell \sum_{\mathbf{x}} c_{\mathbf{x}} (\, |\psi_{\mathbf{x}}^{(\ell)}\rangle - |\psi_{\mathbf{x}}\rangle) \right\| \tag{2.170}$$

$$= \left(\sum_{\mathbf{x}} |c_{\mathbf{x}}|^2 \| \mathbb{E}_\ell \, |\psi_{\mathbf{x}}^{(\ell)}\rangle - |\psi_{\mathbf{x}}\rangle \, \|^2 \right)^{1/2} \tag{2.171}$$

$$\leq \max_{\mathbf{x}} \left(\| \mathbb{E}_\ell \, |\psi_{\mathbf{x}}^{(\ell)}\rangle - |\psi_{\mathbf{x}}\rangle \, \|^2 \right)^{1/2} \tag{2.172}$$

$$= \max_{\mathbf{x}} \| \mathbb{E}_\ell \, |\psi_{\mathbf{x}}^{(\ell)}\rangle - |\psi_{\mathbf{x}}\rangle \, \|. \tag{2.173}$$

Again, the first equality is due to $\langle \psi_{\mathbf{x}}|\psi_{\mathbf{y}}\rangle = \delta_{\mathbf{xy}}$ and the first inequality is due to fact that $\sum_{\mathbf{x}} |c_{\mathbf{x}}|^2 \leq 1$. The first term can be written as

$$2 \left\| \mathbb{E}_\ell \sum_{\mathbf{x}} \, |\psi_{\mathbf{x}}^{(\ell)}\rangle \, (\, \langle \psi_{\mathbf{x}}^{(\ell)}| - \langle \psi_{\mathbf{x}}|) \right\| = 2 \left\| \mathbb{E}_\ell \sum_{\mathbf{x}} (\, |\psi_{\mathbf{x}}^{(\ell)}\rangle - |\psi_{\mathbf{x}}\rangle) (\, \langle \psi_{\mathbf{x}}^{(\ell)}| - \langle \psi_{\mathbf{x}}|) \right. \tag{2.174}$$

$$\left. + \, \mathbb{E}_\ell \sum_{\mathbf{x}} |\psi_{\mathbf{x}}\rangle \, (\, \langle \psi_{\mathbf{x}}^{(\ell)}| - \langle \psi_{\mathbf{x}}|) \right\| \tag{2.175}$$

$$\leq 2 \left\| \mathbb{E}_\ell \sum_{\mathbf{x}} (\, |\psi_{\mathbf{x}}^{(\ell)}\rangle - |\psi_{\mathbf{x}}\rangle) (\, \langle \psi_{\mathbf{x}}^{(\ell)}| - \langle \psi_{\mathbf{x}}|) \right\| \tag{2.176}$$

$$+ \, 2 \left\| \mathbb{E}_\ell \sum_{\mathbf{x}} |\psi_{\mathbf{x}}\rangle \, (\, \langle \psi_{\mathbf{x}}^{(\ell)}| - \langle \psi_{\mathbf{x}}|) \right\|. \tag{2.177}$$

The second term is bounded by $\max_{\mathbf{x}} \|\psi_{\mathbf{x}} - \mathbb{E}_\ell \psi_{\mathbf{x}}^{(\ell)}\|$ and the first term,

$$\left\| \mathbb{E}_\ell \sum_{\mathbf{x}} (|\psi_{\mathbf{x}}\rangle - \, |\psi_{\mathbf{x}}^{(\ell)}\rangle)(\langle \psi_{\mathbf{x}}| - \, \langle \psi_{\mathbf{x}}^{(\ell)}|) \right\| \tag{2.178}$$

$$\leq \max_{\mathbf{x}} \left\| \mathbb{E}_\ell (|\psi_{\mathbf{x}}\rangle - \, |\psi_{\mathbf{x}}^{(\ell)}\rangle)(\langle \psi_{\mathbf{x}}| - \, \langle \psi_{\mathbf{x}}^{(\ell)}|) \right\| \tag{2.179}$$

$$= \max_{\mathbf{x}} \max_{|\phi\rangle} \| \mathbb{E}_\ell (|\psi_{\mathbf{x}}\rangle - \, |\psi_{\mathbf{x}}^{(\ell)}\rangle)(\langle \psi_{\mathbf{x}}| - \, \langle \psi_{\mathbf{x}}^{(\ell)}|) \, |\phi\rangle \, \| \tag{2.180}$$

$$\leq \max_{\mathbf{x}} \| \mathbb{E}_\ell (|\psi_{\mathbf{x}}\rangle - \, |\psi_{\mathbf{x}}^{(\ell)}\rangle)\|, \tag{2.181}$$

the first inequality is due to the fact that for different $\mathbf{x}, \mathbf{y} \in \Omega$, $(\langle \psi_{\mathbf{x}}| - \, \langle \psi_{\mathbf{x}}^{(\ell)}|)(|\psi_{\mathbf{y}}\rangle - |\psi_{\mathbf{y}}^{(\ell)}\rangle) = 0$ and the last inequality is because $|(\langle \psi_{\mathbf{x}}| - \, \langle \psi_{\mathbf{x}}^{(\ell)}|) \, |\phi\rangle \, | \leq 1$.

□

The next lemma is the application of Lemma 2.4.11 on quantum Langevin algorithms.

Lemma 2.4.12. *Let U be the quantum walk operator for unadjusted Langevin algorithm computed using exact gradients. Let U_ℓ be a quantum walk operator for unadjusted Langevin algorithm constructed by computing the gradient on random mini batch ℓ of size B. Then, under Assumptions 2.1.1 and 2.1.2, we have*

$$\|\mathbb{E}_\ell U_\ell - U\| \leq 6\sqrt{2}\eta\beta(LR+G)d^{1/2}/B^{1/2}, \tag{2.182}$$

where G is a positive constant that satisfies $\|\nabla f(0)\| \leq G$.

Proof.

$$\|U - \mathbb{E}_\ell U_\ell\|^2 \tag{2.183}$$
$$\leq 36 \max_{\mathbf{x}\in\Omega} \|\psi_{\mathbf{x}} - \mathbb{E}_\ell \psi_{\mathbf{x}}^{(\ell)}\|^2 \tag{2.184}$$
$$\leq 36 \max_{\mathbf{x}\in\Omega} \| \int_{\mathbf{y}\in\mathbb{R}^d} \mathrm{d}\mathbf{y}\, (\sqrt{p_{\mathbf{xy}}} - \mathbb{E}_\ell \sqrt{p_{\mathbf{xy}}^\ell})\, |\mathbf{x}\rangle\, |\mathbf{y}\rangle \|^2 \tag{2.185}$$
$$= 36 \max_{\mathbf{x}\in\Omega} \left(\int_{\mathbf{y}\in\mathbb{R}^d} \mathrm{d}\mathbf{y}\, p_{\mathbf{xy}} + \int_{\mathbf{y}\in\mathbb{R}^d} \mathrm{d}\mathbf{y}\, \left(\mathbb{E}_\ell \sqrt{p_{\mathbf{xy}}^\ell}\right)^2 - 2\mathbb{E}_\ell \int_{\mathbf{y}\in\mathbb{R}^d} \mathrm{d}\mathbf{y}\, \sqrt{p_{\mathbf{xy}} p_{\mathbf{xy}}^\ell} \right) \tag{2.186}$$
$$\leq 36 \max_{\mathbf{x}\in\Omega} \left(\int_{\mathbf{y}\in\mathbb{R}^d} \sqrt{p_{\mathbf{xy}}} + \mathbb{E}_\ell \int_{\mathbf{y}\in\mathbb{R}^d} \mathrm{d}\mathbf{y}\, p_{\mathbf{xy}}^\ell - 2\mathbb{E}_\ell \int_{\mathbf{y}\in\mathbb{R}^d} \mathrm{d}\mathbf{y}\, \sqrt{p_{\mathbf{xy}} p_{\mathbf{xy}}^\ell} \right) \tag{2.187}$$
$$= 36 \max_{\mathbf{x}\in\Omega} \left(2 - 2\mathbb{E}_\ell \int_{\mathbf{y}\in\mathbb{R}^d} \mathrm{d}\mathbf{y}\, \sqrt{p_{\mathbf{xy}} p_{\mathbf{xy}}^\ell} \right), \tag{2.188}$$

where the first inequality is due to *Lemma* 2.4.11 and the second inequality is due to Jensen's inequality since square root is a concave function.

$$\int_{\mathbf{y}\in\mathbb{R}^d} \mathrm{d}\mathbf{y}\, \sqrt{p_{\mathbf{xy}} p_{\mathbf{xy}}^\ell} \tag{2.189}$$
$$= \frac{1}{(4\pi\eta/\beta)^{d/2}} \int_{\mathbf{y}\in\mathbb{R}^d} \mathrm{d}\mathbf{y}\, \exp\left(-\frac{\|\mathbf{y}-\mathbf{x}+\eta\nabla f(\mathbf{x})\|^2}{4\eta/\beta} \right) \exp\left(-\frac{\|\mathbf{y}-\mathbf{x}+\eta g_\ell(\mathbf{x})\|^2}{4\eta/\beta} \right) \tag{2.190}$$
$$= \frac{1}{(4\pi\eta/\beta)^{d/2}} \int_{\mathbf{y}\in\mathbb{R}^d} \mathrm{d}\mathbf{y}\, \exp\left(-\frac{2\|\mathbf{y}-\mathbf{x}\|^2 + 2\eta\langle \mathbf{y}-\mathbf{x}, \nabla f(\mathbf{x}) + g_\ell(\mathbf{x})\rangle + \eta^2\|\nabla f(\mathbf{x})\|^2 + \eta^2 g_\ell(\mathbf{x})^2}{4\eta/\beta} \right) \tag{2.191}$$
$$= \frac{1}{(4\pi\eta/\beta)^{d/2}} \int_{\mathbf{y}\in\mathbb{R}^d} \mathrm{d}\mathbf{y}\, \exp\left(-\frac{\|\mathbf{y}-\mathbf{x}+\eta(\nabla f(\mathbf{x}) + g_\ell(\mathbf{x}))/2\|^2}{2\eta/\beta} \right) \exp\left(-\frac{\eta^2\|\nabla f(\mathbf{x}) - g_\ell(\mathbf{x})\|^2}{2\eta/\beta} \right) \tag{2.192}$$
$$= \exp\left(-\frac{\eta^2\|\nabla f(\mathbf{x}) - g_\ell(x)\|^2}{2\eta/\beta} \right), \tag{2.193}$$

where $\mathbb{E}[g_\ell] = \nabla f$. Therefore,

$$\|U - \mathbb{E}U_\ell\|^2 \leq 36 \max_{\mathbf{x}\in\Omega} \left(2 - 2\mathbb{E}\exp\left(-\frac{\eta^2\|\nabla f(\mathbf{x}) - g_\ell(\mathbf{x})\|^2}{2\eta/\beta}\right)\right) \tag{2.194}$$

$$\leq 36(2 - 2\exp\Big(-\eta^2\beta^2(LR+G)^2/B\Big)) \tag{2.195}$$

$$\leq 72\eta^2 d\beta^2(LR+G)^2/B, \tag{2.196}$$

where the first inequality follows from lemma B.2 from [ZXG21],

$$\mathbb{E}\exp(\langle a, g_\ell(\mathbf{x}) - \nabla f\rangle) \leq \exp\Big(M^2\|a\|_2^2/B\Big), \tag{2.197}$$

where M is the upper bound on $\|g_\ell(\mathbf{x}) - \nabla f(\mathbf{x})\|$ with batch size B. □

The next lemma upper bounds the difference of two random unitary quantum walk operators.

Lemma 2.4.13. *Let U_{ℓ_1} and U_{ℓ_2} be two random quantum walk operators constructed with two different stochastic gradients g_{ℓ_1} and g_{ℓ_2} for unadjusted Langevin algorithm. Then, under Assumptions 2.1.1 and 2.1.2, we have*

$$\|U_{\ell_1} - U_{\ell_2}\| \leq 8\sqrt{\eta\beta(LR+G)^2}, \tag{2.198}$$

where G is a positive constant that satisfies $\|\nabla f(0)\| \leq G$.

Proof. By Lemma 2.4.6, the difference of quantum walk operators is bounded by,

$$\|U_{\ell_1} - U_{\ell_2}\|^2 \leq 32 \max_{\mathbf{x}} \mathrm{H}(P_{\ell_1}, P_{\ell_2})^2, \tag{2.199}$$

where P_{ℓ_1} and P_{ℓ_2} are Gaussian transition densities of ULA computed with gradients on mini batches ℓ_1 and ℓ_2. This is squared Hellinger distance between two Gaussian distributions with the same variance and different mean. This is a known result [Par18] and equal to following.

$$\mathrm{H}(P_{\ell_1}, P_{\ell_2}) = 1 - \exp\left(-\frac{\eta^2\|g_{\ell_1}(\mathbf{x}) - g_{\ell_2}(\mathbf{x})\|^2}{2\eta/\beta}\right) \leq \frac{\eta^2\|g_{\ell_1}(\mathbf{x}) - g_{\ell_2}(\mathbf{x})\|^2}{2\eta/\beta}. \tag{2.200}$$

Since $\|\nabla f(\mathbf{x})\| \leq L\|\mathbf{x}\| + G \leq LR + G$, $\|g_{\ell_1}(\mathbf{x}) - g_{\ell_2}(\mathbf{x})\| \leq 2(LR+G)$, therefore for any $\mathbf{x} \in \Omega$,

$$\|U_{\ell_1} - U_{\ell_2}\|^2 \leq 64(LR+G)^2\eta\beta. \tag{2.201}$$

Taking the square root, we obtain the result in the statement. □

Finally we prove the following theorem to conclude the analysis of stochastic quantum sampling algorithm.

Theorem 2.4.14 (Quantum ULA with stochastic gradient)**.** *Let $\pi \propto e^{-\beta f(\mathbf{x})}$ denote a probability distribution with inverse temperature $\beta > 0$ such that $f(\mathbf{x}) = \frac{1}{N}\sum_{k=1}^{N} f_k(\mathbf{x})$ satisfies Assumptions 2.1.1 and 2.1.2. Then, there exists a quantum algorithm that outputs a random variable distributed according to μ such that,*

$$\mathrm{TV}(\mu, \pi) \leq \epsilon, \tag{2.202}$$

using $\tilde{\mathcal{O}}\left(\beta^2 d^{3/2}\epsilon^{-2}\rho^{-2}c_{\mathrm{LSI}}^{-1}\right)$[1] queries to $O_{\tilde{\nabla}f}$ and each $O_{\tilde{\nabla}f}$ involves $\mathcal{O}(d)$ gradient calculations.

Proof. Let U_ℓ be a unitary quantum walk operator defined as,

$$U_\ell = S\Big(2\sum_x |\psi_{\mathbf{x}}^{(\ell)}\rangle \langle\psi_{\mathbf{x}}^{(\ell)}| - I\Big), \tag{2.203}$$

where $|\psi_{\mathbf{x}}^{(\ell)}\rangle$ is the state,

$$|\psi_{\mathbf{x}}^{(\ell)}\rangle = \sum_y \sqrt{p_{\mathbf{xy}}^{(\ell)}}\,|x\rangle\,|\mathbf{y}\rangle\,, \tag{2.204}$$

where $p_{\mathbf{xy}}^{(\ell)} = \frac{1}{(4\pi\eta/\beta)}\exp\Big(-\frac{\|\mathbf{y}-\mathbf{x}+g_\ell(\mathbf{x})\|^2}{2\eta/\beta}\Big)$, and g_ℓ is the stochastic gradient computed on randomly selected data points of size B, i.e.,

$$g_\ell(\mathbf{x}) \coloneqq \frac{1}{B}\sum_{i\in S_\ell\subseteq[N]} \nabla f_i(\mathbf{x}). \tag{2.205}$$

The number of gradient evaluations for implementing unitary U_ℓ is $\mathcal{O}(B)$ since we only need to compute gradient on B data points. The key idea in proof of the quantum ULA is the fact that the following operator can be implemented using controlled-U operators:

$$V = e^{i\pi/3}\Pi_\Delta + \Pi_\Delta^\perp, \tag{2.206}$$

where Δ is the phase gap of quantum MALA walk operator. Suppose that we replace every controlled-U operator with a unitary U_ℓ. Note that each U in the circuit might be

[1]As each $O_{\tilde{\nabla}f}$ uses $\tilde{\mathcal{O}}(d)$ gradient calculations, the number of total gradient calculations scale as $d^{5/2}$ as shown Table 2.1.

possibly replaced by different unitary due to randomness of stochastic gradients. Let's denote this circuit by $\tilde{V}$. Now, we show that with high probability $\|V - \tilde{V}\| \leq \epsilon$ for sufficiently small step size. Since the algorithm uses $1/\Delta(U^*)$ calls to U,

$$\|V - \mathbb{E}(\tilde{V})\| \leq \frac{1}{\Delta}\|U^* - \mathbb{E}_\ell U_\ell\| \tag{2.207}$$

$$\leq \frac{1}{\Delta}\|U - U^*\| + \|U - \mathbb{E}_\ell U_\ell\| \tag{2.208}$$

$$\leq (\rho^{-1}\sqrt{\beta/\eta})\eta d L + (\rho^{-1}\sqrt{\beta/\eta})(\eta\beta(LR+G)\sqrt{d/B}) \tag{2.209}$$

$$= \rho^{-1}\eta^{1/2}\beta^{1/2} d L + \rho^{-1}\beta^{3/2}\eta^{1/2}(LR+G)d^{1/2}/B^{1/2}. \tag{2.210}$$

Setting $\eta \leq \min\left(\frac{\epsilon^2\rho^2}{2\beta d^2 L^2}, \frac{\epsilon^4\rho^2 B}{4\beta^3 d(LR+G)^2}\right)$ and $B = d$, we guarantee that,

$$\|V - \mathbb{E}\tilde{V}\| \leq \epsilon/2. \tag{2.211}$$

Next, we use the McDiarmid's inequality to obtain high probability bound:

$$\mathrm{P}(\|\tilde{V} - \mathbb{E}\tilde{V}\| \geq \epsilon/2) \leq 2\exp\left(-\frac{\epsilon^2\Delta}{2\|U_{\ell_1} - U_{\ell_2}\|^2}\right) \tag{2.212}$$

$$\leq 2\exp\left(-\frac{\epsilon^2\rho\eta^{1/2}}{2\beta^{1/2}\|U_{\ell_1} - U_{\ell_2}\|^2}\right) \tag{2.213}$$

$$\leq 2\exp\left(-\frac{\epsilon^2\rho\eta^{1/2}}{128\beta^{1/2}\eta\beta(LR+G)^2}\right) \tag{2.214}$$

$$= 2\exp\left(-\frac{\epsilon^2\rho}{128\beta^{3/2}\eta^{1/2}(LR+G)^2}\right). \tag{2.215}$$

Setting $\eta \leq \frac{\epsilon^4\rho^2}{128^2(LR+G)^4\beta^3}$, guarantees that with at least constant probability,

$$\|\tilde{V} - \mathbb{E}\tilde{V}\| \leq \epsilon/2. \tag{2.216}$$

The probability can be boosted in logarithmic number of steps to obtain high probability. Therefore, with high probability,

$$\|V - \tilde{V}\| \leq \|V - \mathbb{E}\tilde{V}\| + \|\mathbb{E}\tilde{V} - \tilde{V}\| \leq \epsilon. \tag{2.217}$$

Then, to implement the operator V up to ϵ accuracy with high probability, we need $\tilde{\mathcal{O}}(1/\Delta) = (\rho^{-1}\eta^{1/2}\beta^{-1/2}) = \tilde{\mathcal{O}}(\rho^{-2}d\beta/\epsilon^2)$ calls to U_ℓ. Since each U_ℓ requires $B = d$ gradient computations and we need to prepare $\tilde{\mathcal{O}}(c_{\mathrm{LSI}}^{-1}\sqrt{d})$ reflections, the total gradient

complexity is $\tilde{\mathcal{O}}(c_{\mathrm{LSI}}^{-1}\rho^{-2}d^{5/2}/\epsilon^2)$. □

Remark 2.4.15. In a special scenario where an initial quantum state that has at least constant overlap with $|\pi\rangle$ is provided (e.g. a constant warm state), it is possible to obtain an additional speed up in d dependence by saving up to $\mathcal{O}(d^{1/2}c_{\mathrm{LSI}}^{-1})$ using a single Markov chain instead of using simulated annealing.

Remark 2.4.16. The dependency on the isoperimetric constants for quantum MALA, ULA and stochastic ULA are $\rho^{-1}c_{\mathrm{LSI}}^{-1}, \rho^{-1}c_{\mathrm{LSI}}^{-1}$ and $\rho^{-2}c_{\mathrm{LSI}}^{-1}$ respectively as given in the Theorems 2.4.5, 2.4.10 and 2.4.14. On the other hand, the dependency for the classical algorithms in Table 2.1 are $c_{\mathrm{LSI}}^{-2}, c_{\mathrm{LSI}}^{-2}$ and ρ^{-4}. Unfortunately, there is no tight relation between c_{LSI} and ρ and this makes hard to make comparison without further structure on f. Although it is not fully rigorous, it is still possible to make a comparison by converting both Cheeger constants and log-Sobolev constants to Poincare constant (c_p) which is another way of expressing the global properties of the function landscape. Using $\rho \geq \Omega(d^{-1/2}c_p)$, and $c_{\mathrm{LSI}} \geq c_p$ [Bus82], we can show that quantum algorithms have the complexity $d^{3/2}c_p^{-2}$, $d^{5/2}c_p^{-2}$, $d^{7/2}c_p^{-3}$ for quantum MALA, ULA and stochastic ULA. On the other hand, the classical complexities become $d^2c_p^{-2}, dc_p^{-2}$ and $d^6c_p^{-4}$. Note that this conversion does not change ϵ dependency. Hence, our quantum algorithm have the same dependency on c_p for MALA and ULA, whereas it has better dependency (c_p^{-3} vs c_p^{-4}) for stochastic ULA. As claimed in the main text, by expressing the bounds in terms of c_p, we also maintain the improvement in d for MALA and stochastic ULA. Unfortunately, Buser's inequality is not always tight and this argument both loosens classical and quantum bounds. For the sake of keeping the bounds sharp, we did not include this kind of comparison in Table 2.1.

We also note that in the most general case, the isoperimetric constants may be exponentially small on d, L and b. Therefore, in stochastic case, we might obtain a significant speedup. However, since the runtime is dominated by c_{LSI}^{-1} and ρ^{-1}, the dimension speedup for quantum MALA may not be as important. However, for certain non-convex functions encountered in machine learning, the dependency on d might not be exponential. For example, for locally non-convex function, which models the Gaussian mixtures, considered in [MCJ+19], c_{LSI}^{-1} scales as $\mathcal{O}(\exp(LR^2))$ where R is the radius of the non-convex region.

2.5 Partition Function Estimation

Computing the partition functions in low temperature regime is a challenging problem that has applications in convex geometry [CCH+23], linear algebra [JSV04], and graph theory [ŠVV09]. Even though computing the partition function exactly is a $\#P$ hard problem, it can be approximated up to a multiplicative constant using MCMC methods.

In this section, we describe the method and analysis for estimating the partition function for a non-logconcave distribution defined as,

$$Z = \int_{\mathbf{x}\in\mathbb{R}^d} e^{-f(\mathbf{x})}\,\mathrm{d}\mathbf{x}. \tag{2.218}$$

The partition function can be estimated using the following telescoping product:

$$Z = Z_1 \prod_{i=1}^{M} \frac{Z_{i+1}}{Z_i}, \tag{2.219}$$

where Z_i is the normalizing constant of the distribution $\mu_i \propto \exp\left(-\frac{-\|\mathbf{x}\|^2}{2\sigma_i^2} + f(\mathbf{x})\right)$ and $Z_{M+1} = Z$ where $\sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_M$ and $\sigma_{M+1} = \infty$. We then approximate $Z_{M+1} = Z_1 \prod\limits_{i=1}^{M} \frac{Z_{i+1}}{Z_i} = Z_1 \prod_{i=1}^{M} \mathbb{E}_{\mu_i}[g_i]$, where

$$g_i = \exp\left(\frac{1}{2}\left(\frac{1}{\sigma_i^2} - \frac{1}{\sigma_{i+1}^2}\right)\|\mathbf{x}\|^2\right) \tag{2.220}$$

for $i \in [M]$ where σ_i is defined in proof of Lemma 2.3.1. To be able estimate this product, we use the technique proposed by [CH23]. Their idea is to estimate each expectation in the product using nearly unbiased quantum mean estimation. Since each term in the product can be estimated faster on a quantum computer than the classical counterparts, the overall algorithm both exploits the fast mean estimation and sampling.

Theorem 2.5.1. *Let $Z = \int_x e^{-f(\mathbf{x})}\,\mathrm{d}\mathbf{x}$ be the partition with $f(\mathbf{x})$ function satisfying assumptions Assumptions 2.1.1 and 2.1.2. Then, there exists quantum algorithms that output an estimate $\tilde{Z}$ such that,*

$$(1-\epsilon)Z \leq \tilde{Z} \leq (1+\epsilon)Z \tag{2.221}$$

with probability at least $3/4$ using,

- $\tilde{\mathcal{O}}\left(d^{5/4}\epsilon^{-1}\rho^{-1}c_{\mathrm{LSI}}^{-1}\right)$ *queries to $O_{\nabla f}$ and O_f, or*

- $\tilde{\mathcal{O}}\left(d^{7/4}\epsilon^{-2}\rho^{-1}c_{\mathrm{LSI}}^{-1}\right)$ *queries to* $O_{\nabla f}$*, or*
- $\tilde{\mathcal{O}}\left(d^{11/4}\epsilon^{-3}\rho^{-2}c_{\mathrm{LSI}}^{-1}\right)$ *queries to* $O_{\tilde{\nabla} f}$*.*

Before we prove the main theorem for partition functions, we give the following lemmas. The following lemma shows that Z_1 can be estimated up to ϵ multiplicative constant from a Gaussian.

Lemma 2.5.2 (Lemma 3.1 of [GLL20])**.** *Letting* $\sigma_1^2 = \frac{\epsilon}{2dL}$*, it holds that*

$$\left(1-\frac{\epsilon}{2}\right)\int_{\mathbf{x}\in\mathbb{R}^d}\exp\left(\frac{-\|\mathbf{x}\|^2}{2\sigma_1^2}\right)\mathrm{d}\mathbf{x} \leq Z_1 \leq \int_{\mathbf{x}\in\mathbb{R}^d}\exp\left(\frac{-\|\mathbf{x}\|^2}{2\sigma_1^2}\right)\mathrm{d}\mathbf{x}. \tag{2.222}$$

The next lemma uses unbiased quantum mean estimation to compute the product of ℓ random variables.

Lemma 2.5.3 (Theorem 3.3 of [CH23])**.** *Let* $B > 1$ *and* $\epsilon \in (0,1)$*. Consider a sequence* $X_1, \cdots, X_\ell$ *of* ℓ *independent random variables with support size* n*, bounded relative second moment* $\frac{\mathbb{E}[X_i^2]}{\mathbb{E}[X_i]^2} \leq B$ *and bounded fidelity* $|\left\langle \pi_{X_i} \middle| \pi_{X_{i+1}} \right\rangle|^2 \geq 1/B$ *for all* i*. Denote their product as* $X = X_1 \ldots X_\ell$*. Then, there exists a quantum algorithm that outputs a multiplicative-error estimate* $\tilde{p}$ *such that*

$$\left|\tilde{p} - \mathbb{E}\Big[\prod_{i=1}^{\ell} X_i\Big]\right| \leq \epsilon \mathbb{E}\Big[\prod_{i=1}^{\ell} X_i\Big] \tag{2.223}$$

with probability at least 2/3. It uses $\mathcal{O}(B)$ *copies of* $|\pi_{X_1}\rangle$ *and* $\tilde{\mathcal{O}}(B^2\ell^{3/2}/\epsilon + B\ell\log(n))$ *reflections through the states* $|\mu_{X_1}\rangle, \cdots, |\pi_{X_\ell}\rangle$ *in expectation.*

Finally, we combine our sampling algorithms with the annealing schedule and product estimator to obtain our result for the partition function estimation.

Proof. By Lemma 2.5.2, we can estimate Z_1 with ϵ accuracy using normalization constant of Gaussian distribution with variance σ_1^2. Then, we show that g_i has constant relative variance for all $i \in [M]$. Since the partition function can be written in telescoping product given in Equation (2.219), we can use Lemma 2.5.3 to estimate the remaining product up to ϵ multiplicative constant with high probability. First, for g_M,

$$\frac{\mathbb{E}_{\mu_M}[g_M^2]}{\mathbb{E}_{\mu_M}[g_M]^2} = \mathbb{E}_\pi\left[\exp\left(-\frac{1}{2\sigma_M^2}\|\mathbf{x}\|^2\right)\right]\mathbb{E}_\pi\left[\exp\left(\frac{1}{2\sigma_M^2}\|\mathbf{x}\|^2\right)\right] \tag{2.224}$$

$$\leq \mathcal{O}(\exp\Big(dL/(m\sigma_M^4 c_{\mathrm{LSI}}^2)\Big)) \tag{2.225}$$

by Lemma 2.3.4. Setting $\sigma_M^2 = \Omega(\sqrt{dL/(mc_{\mathrm{LSI}}^2)})$ implies $\frac{\mathbb{E}_{\mu_M}[g_M^2]}{\mathbb{E}_{\mu_M}[g_M]^2} \leq O(1)$. Similarly, for $i \in [1, M-1]$,

$$\frac{\mathbb{E}_{\mu_i}[g_i^2]}{\mathbb{E}_{\mu_i}[g_i]^2} = \frac{\mathbb{E}_\pi\left[\exp\left(-\frac{(1+\alpha)}{2}\|\mathbf{x}\|^2\right)\right]\mathbb{E}_\pi\left[\exp\left(-\frac{(1-\alpha)}{2}\|\mathbf{x}\|^2\right)\right]}{(\mathbb{E}_\pi[\exp(-\|\mathbf{x}\|^2/2)])^2} \tag{2.226}$$

$$\leq \mathcal{O}(\exp\left(dL\alpha^2/(mc_{\mathrm{LSI}}^2)\right) \tag{2.227}$$

by Lemma 2.3.5. Therefore, for $\alpha^2 = \tilde{\mathcal{O}}(mc_{\mathrm{LSI}}^2/(dL))$, $\frac{\mathbb{E}_{\mu_i}[g_i^2]}{\mathbb{E}_{\mu_i}[g_i]^2} \leq \mathcal{O}(1)$. Having established that the relative variance is constant for all g_i, by Lemma 2.5.3, we conclude that the product form can be estimated by using $\tilde{\mathcal{O}}(M^{3/2}/\epsilon) = \tilde{\mathcal{O}}(d^{3/4}/\epsilon)$ reflection operators. We can choose quantum MALA, quantum ULA or stochastic quantum ULA algorithms to implement the reflection operators. Since the complexities given in Theorem 2.4.5, Theorem 2.4.10, Theorem 2.4.14 are the complexities of implementing reflection operator times M, we just need to multiply these results with $\tilde{\mathcal{O}}(M^{1/2}) = \tilde{\mathcal{O}}(d^{1/4})$ to conclude the proof. □

Unfortunately, we are not aware of any classical algorithm for computing the partition function under the same assumptions as ours, therefore we are unable to make a solid comparison.

2.6 Conclusion

We have analyzed algorithms for quantum sampling and estimating partition functions for non-logconcave distributions by quantizing popular techniques in classical sampling literature. We believe our techniques and analysis can be useful tools for developing future quantum Monte Carlo algorithms especially based on non-reversible chains. We list the following theoretical open problems for future work.

- Our quantum algorithms utilize the first order sampling methods used in classical literature. It is known that underdamped Langevin algorithm is the accelerated variant of sampling similar to Nesterov's acceleration in optimization [MCC+19]. It is an interesting direction to analyze the possible quantum speedups using such sophisticated classical techniques. Analyzing these possible quantum algorithms in terms of other distance metrics such as Wasserstein distance or KL divergence is also another challenge as these distance metrics are not invariant under unitary transformations.

- Langevin Monte Carlo algorithm is obtained by discretization of the continuous stochastic differential equation known as Langevin diffusion. Analyzing the continuous SDE in quantum domain directly might be another way of getting around reversibility issue and we might obtain more efficient quantum algorithms.
- Fast forwarding of quantum Markov chains to obtain the transient dynamics rather than its stationary density is also interesting direction and [AS19] proposed a quantum algorithm to solve this problem for reversible chains. Using similar perturbation analysis can potentially be used to show that non-reversible chains can also be fast-forwarded under special settings faster than classical counterparts.

Chapter 3
Speedups for Sampling and Optimization via Quantum Gradients

In Chapter 2, we showed how to use quantum walk framework to obtain speedups for Gibbs sampling using simple discretization of Langevin dynamics. In this chapter, we follow a different approach to obtain speedups for the Gibbs sampling problem. Instead of using quantum walks, we exploit quantum computers to speedup gradient estimation which constitutes a major bottleneck for sampling for some potential functions. Our first approach considers Gibbs sampling for finite-sum potentials in the stochastic setting, employing an oracle that provides gradients of individual functions. In the second setting, we consider access only to a stochastic evaluation oracle, allowing simultaneous queries at two points of the potential function under the same stochastic parameter. By introducing novel techniques for stochastic gradient estimation, our algorithms improve the gradient and evaluation complexities of classical samplers, such as Hamiltonian Monte Carlo (HMC) and Langevin Monte Carlo (LMC) in terms of dimension, precision, and other problem-dependent parameters. Furthermore, we achieve quantum speedups in optimization, particularly for minimizing non-smooth and approximately convex functions that commonly appear in empirical risk minimization problems.

This chapter is based on [OLMW25], joint with Xiantao Li, Mehrdad Mahdavi and Chunhao Wang.

3.1 Introduction

We continue considering the problem of sampling from a probability distribution π of the form

$$\pi(\mathbf{x}) = \frac{e^{-f(\mathbf{x})}}{\int e^{-f(\mathbf{x})} \mathrm{d}\mathbf{x}}. \tag{3.1}$$

Our goal in this chapter is to efficiently sample approximately from π while minimizing the number of gradient queries in the finite-sum setting, i.e., $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$, and minimizing the number of stochastic evaluation queries in the zeroth-order setting, where we have only access to noisy function values.

Recall that The Euler-Maruyama discretization of this Langevin diffusion equation results in Langevin Monte Carlo (LMC) algorithm:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \nabla f(\mathbf{x}_t) + \sqrt{2\eta_t}\boldsymbol{\epsilon}_t, \tag{3.2}$$

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

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \frac{\partial H}{\partial \mathbf{p}}, \quad \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = -\frac{\partial H}{\partial \mathbf{x}}. \tag{3.3}$$

Similar to LMC, in practice HMC is simulated by discretizing Equation (3.3). Although effective, the computational cost of each iteration in these algorithms becomes prohibitive when the computation of the gradient is costly, such as in the finite sum or zeroth-order setting. To alleviate the computational burden, stochastic gradient-based samplers such as Stochastic Gradient Langevin Dynamics (SGLD) [WT11] and Stochastic Hamiltonian Monte Carlo (SG-HMC) [CFG14] have been proposed. Instead of computing the full gradient, these algorithms use stochastic approximation to the gradient. For example, the stochastic update for LMC becomes

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \mathbf{g}_t + \sqrt{2\eta_t}\boldsymbol{\epsilon}_t. \tag{3.4}$$

In the finite sum form, $\mathbf{g}_t$ can be obtained by randomly sampling a component $i \in [n]$ and

computing $\nabla f_i(\mathbf{x}_t)$. In the zeroth-order scenario, a stochastic gradient can be obtained by using finite difference formulas by evaluating the function at two close points [NS17].

While stochastic gradient methods reduce computation at each iteration, they introduce variance into the gradient estimates, which can degrade the quality of the samples and slow down convergence. Non-asymptotic convergence rates for SGLD and SG-HMC have been analyzed extensively by [RRT17, XCZG18, ZXG21, DNR23] and [CFG14, ZG21] respectively. In the finite sum setting, more sophisticated variance reduction techniques such as SVRG [JZ13], SAGA [DBLJ14], SARAH [NLST17], and Control Variates (CV) [BFFN19] have been used to reduce the variance of stochastic gradients by leveraging the gradient information from previous iterations. Although these methods were originally introduced in the context of optimization, successive works have applied these methods to improve sampling efficiency via LMC [DJRW$^+$16, CFM$^+$18, BFFN19, KS22] and HMC [ZXG19, ZG21]. In particular, [ZG21] has incorporated various variance reduction techniques to SG-HMC and analyzed convergence in Wasserstein distance for smooth and strongly convex potentials. In the non-log-concave setting, [KS22] has analyzed the convergence of SVRG-LMC and SARAH-LMC for target distributions that satisfy the Log-Sobolev inequality and applied their results to optimize structured non-convex objectives.

In certain problems where the gradient is either unavailable or computationally too expensive to query, one must often rely on noisy function evaluations, which can significantly degrade performance due to the inherent difficulty in accurately estimating the gradient from noisy function values. This scenario has been analyzed under various settings in optimization literature [DJWW15, NS17, BG22, LZJ22]. For sampling problems, [RSBG21] has analyzed the convergence of various discretizations of Langevin diffusion algorithms both for strongly convex and non-convex potentials using the noisy zeroth-order oracle. It is also worth noting that [DK19] has established the convergence of sampling under inexact gradients; however, their analysis only applies when the bias and the variance of the inexact estimates are bounded, which does not always hold in the zeroth-order setting. Similarly, [YW23] analyzed the convergence of the inexact Langevin algorithm in KL divergence under different assumptions on the score function.

Quantum computing has emerged as a powerful tool for tackling problems in computational science, offering potential speedups in various domains, including sampling and optimization. In the context of optimization, quantum algorithms such as multi-dimensional quantum mean estimation [CHJ22] and quantum gradient estimation [Jor05, GSLW19] have shown promise in reducing the computational cost associated with gradient-based

methods [vAGGdW20, CCLW20, SZ23, ZZF+24, LGHL24]. These techniques are particularly well-suited for addressing challenges in large-scale and noisy settings, as they can provide more accurate gradient estimates with fewer queries. This work focuses on integrating these quantum techniques to enhance the efficiency of stochastic gradient-based samplers and alleviate the computational burden inherent in classical methods.

3.1.1 Main Contributions

- **Speedups for Finite Sum Potentials**: We propose novel quantum algorithms to sample from Gibbs distribution for finite-sum potentials implemented via quantum variance reduction techniques (Section 3.3). We prove that our algorithms improve the dependency on n compared to classical state-of-the-art algorithms such as HMC (Theorems 3.3.6 and 3.3.8) and LMC (Theorem 3.3.15) to approximately sample from strongly convex and non-convex potentials, respectively (See Table 3.1).
- **Quantum Speedups for Gradient Estimation via Stochastic Evaluation Oracle**: In the zeroth-order setting, where only stochastic evaluations of the potential function are available, we develop new quantum gradient estimation algorithms under various smoothness assumptions in Section 3.4. Our algorithm provides quadratic speedup when the potential function is smooth, reducing the evaluation queries from $\tilde{\mathcal{O}}(\frac{d^2\sigma^2}{\epsilon^2})$ to $\tilde{\mathcal{O}}(\frac{d\sigma}{\epsilon})$ to compute the gradient up to ϵ accuracy (Theorem 3.4.7) where σ^2 is the variance of the noise as in Assumption 3.4.3. Furthermore, when the stochastic functions are also smooth with high probability, we manage to shave off an additional $d^{1/2}$ term (Theorem 3.4.12). This is achieved by combining quantum mean estimation with Jordan's quantum gradient estimation in a robust manner. Our gradient estimation algorithms could be useful as independent tools, especially in zeroth-order stochastic optimization.
- **Speedups for Zeroth-Order Sampling**: In Section 3.5, we combine our new quantum gradient estimation algorithm with HMC and LMC algorithms and show that the final algorithm uses fewer number of queries to evaluation oracle than the best known classical samplers under the same assumptions (Theorems 3.5.1 and 3.5.2).
- **Application to Non-Convex Optimization**: In Section 3.6, we extend our quantum sampling methods to optimize non-convex functions with specific structural properties, demonstrating that faster sampling translates to provable speedups in complex optimization tasks. In particular, we show that we can optimize non-smooth and approximately convex functions, i.e. a function that is uniformly close to a strongly convex function,

using fewer stochastic evaluation queries than the best known classical algorithms in terms of dimension dependency (Theorem 3.6.6).

3.2 Quantum Mean Estimation

Quantum mean estimation is a technique to estimate the mean of a d-dimensional random variable X up to ϵ accuracy using $\tilde{\mathcal{O}}(d^{1/2}/\epsilon)$ queries, which is a quadratic improvement in ϵ compared to classical algorithms [CHJ22]. Although the quantum mean estimation algorithm is biased, [SZ23] developed an unbiased quantum mean estimation algorithm. Specifically, for a multi-dimensional variable with mean μ and variance σ^2, unbiased quantum mean estimation outputs an estimate $\hat{\mu}$ such that $\mathbb{E}[\hat{\mu}] = \mu$ and $\mathbb{E}[\|\hat{\mu}-\mu\|^2] \leq \hat{\sigma}^2$ using $\tilde{\mathcal{O}}(d^{1/2}\sigma/\hat{\sigma})$ queries.

Definition 3.2.1 (Quantum Sampling Oracle)**.** Quantum sampling oracle O_X of a random variable $X \in \Omega$ is given by $O_X |0\rangle |0\rangle \mapsto \sum_{X\in\Omega} \sqrt{\Pr(X)} |X\rangle |\text{garbage}(X)\rangle$.

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

The following lemma shows that the mean $\mathbb{E}[X]$ for a random variable X can be computed quadratically faster than classical mean estimation with respect to oracle O_X.

Lemma 3.2.2 (Unbiased Quantum Mean Estimation [SZ23])**.** *For a d-dimensional random variable X with $\text{Var}[X] \leq \sigma^2$ and some $\hat{\sigma} \geq 0$, suppose we are given access to its quantum sampling oracle O_X. Then, there is a procedure* `QuantumMeanEstimation`$(O_X, \hat{\sigma})$ *that uses $\tilde{\mathcal{O}}\left(\frac{d^{1/2}\sigma}{\hat{\sigma}}\right)$ queries to O_X and outputs an unbiased estimate $\hat{\mu}$ of the expectation μ satisfying $\text{Var}[\hat{\mu}] \leq \hat{\sigma}^2$.*

In the next section, we analyze the trade-off between the error due to stochastic gradients and discretization to quantify how much quantum mean estimation techniques can provide speedups when combined with classical variance reduction methods such as SVRG and CV.

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

Table 3.1. Summary of the results (some of the previous results use different scaling of f and we convert the results to the same scaling as ours in the table). Here, we mainly focus on n and ϵ dependency. See Theorems 3.3.6, 3.3.8 and 3.3.15 for dependency on L, μ, α, d.

Algorithm	Assumptions	Metric	Gradient Complexity
SG-HMC [ZG21]	Strongly Convex	W_2	$\tilde{\mathcal{O}}(n\epsilon^{-2})$
SVRG-HMC [ZG21]	Strongly Convex	W_2	$\tilde{\mathcal{O}}(n^{2/3}\epsilon^{-2/3} + \epsilon^{-1})$
SAGA-HMC [ZG21]	Strongly Convex	W_2	$\tilde{\mathcal{O}}(n^{2/3}\epsilon^{-2/3} + \epsilon^{-1})$
CV-HMC [ZG21]	Strongly Convex	W_2	$\tilde{\mathcal{O}}(\epsilon^{-2})$
SRVR-HMC [ZXG19]	Dissipative Gradients	W_2	$\tilde{\mathcal{O}}(n + n^{1/2}\epsilon^{-2} + \epsilon^{-4})$
SVRG-LMC [KS22]	LSI	KL	$\tilde{\mathcal{O}}(n + n^{1/2}\epsilon^{-1})$
SARAH-LMC [KS22]	LSI	KL	$\tilde{\mathcal{O}}(n + n^{1/2}\epsilon^{-1})$
QSVRG-HMC [Theorem 3.3.6]	Strongly Convex	W_2	$\tilde{\mathcal{O}}(n^{1/2}\epsilon^{-3/4} + \epsilon^{-1})$
QCV-HMC [Theorem 3.3.8]	Strongly Convex	W_2	$\tilde{\mathcal{O}}(\epsilon^{-3/2})$
QSVRG-LMC [Theorem 3.3.15]	LSI	KL [1]	$\tilde{\mathcal{O}}(n + n^{1/3}\epsilon^{-1})$

3.3 Quantum Speedups for Finite-Sum Sampling via Gradient Oracle

In this section, we consider a finite sum potential $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$. We assume that we have a stochastic gradient oracle that takes $i \in [n]$ and $\mathbf{x} \in \mathbb{R}^d$ and returns $\nabla f_i(\mathbf{x})$. That is,

$$O_{\nabla f} |\mathbf{x}\rangle |i\rangle |0\rangle \mapsto |\mathbf{x}\rangle |i\rangle |\nabla f_i(\mathbf{x})\rangle \tag{3.5}$$

Computing the exact gradient takes $\mathcal{O}(n)$ queries to this oracle and dominates the sampling complexity, especially when $n \gg d$. The goal is to approximately sample from π by using as few gradient computations as possible without deteriorating the convergence.

3.3.1 Sampling under Strong Convexity via Hamiltonian Monte Carlo

First, we consider quantum speedups for Hamiltonian Monte Carlo (HMC) algorithm using quantum variance reduction techniques.

Hamiltonian Monte Carlo (HMC) is an advanced sampling technique designed to

Algorithm 3 `QSVRG/QCV`

input $O_{\nabla f}$, current iterate $\mathbf{x}_k$, smoothness constant L, variance scale factor b, epoch length m.

output Quantum variance reduced stochastic gradient $\mathbf{g}$.

QSVRG:
if $k \mod m = 0$ **then**
  $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$
  $\tilde{\mathbf{x}} = \mathbf{x}_k$
else
  Define oracle $O_{\mathrm{SVRG}}^{\mathbf{x}_k}$:

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

  $\hat{\sigma}^2 = L^2 \|\mathbf{x}_k - \tilde{\mathbf{x}}\|^2 / b^2$
  $\mathbf{g}_k = \texttt{QuantumMeanEstimation}(O_{\mathrm{SVRG}}^{\mathbf{x}_k}, \hat{\sigma}^2)$
end if

QCV:
Define oracle $O_{\mathrm{CV}}^{\mathbf{x}_k}$:

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

$\hat{\sigma}^2 = L^2 \|\mathbf{x}_k - \mathbf{x}_0\|^2 / b^2$
$\mathbf{g}_k = \texttt{QuantumMeanEstimation}(O_{\mathrm{CV}}^{\mathbf{x}_k}, \hat{\sigma}^2)$

Return $\mathbf{g}_k$

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

HMC alternates between updating the position $\mathbf{x}$ and momentum $\mathbf{p}$ by simulating Hamiltonian dynamics Equation (3.3). In practice, Hamiltonian dynamics is simulated using the leapfrog integrator, which discretizes the continuous equations of motion. The key advantage of HMC is that it allows for large, efficient moves through the parameter space by leveraging gradient information and auxiliary momentum. This reduces the correlation between successive samples, particularly in high-dimensional spaces, resulting in faster convergence compared to simple random-walk methods like the Metropolis-

Hastings algorithm. In practice, Hamiltonian dynamics are simulated using the leapfrog integrator, which discretizes the continuous equations of motion. The leapfrog method proceeds in three steps:

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

where η is the step size. After a series of updates, the momentum $\mathbf{p}_{k+1}$ is refreshed by sampling from $\mathcal{N}(0, I)$. This discretization ensures symplecticity, preserving volume in phase space and allowing the algorithm to make large, energy-conserving moves through the parameter space.

Algorithm 4 `SG-HMC`

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

Similar to SGLD, one can replace the gradients with stochastic gradients resulting in SG-HMC (See Algorithm 4). The stochastic gradients $\mathbf{g}(\mathbf{x}, \xi)$ in Algorithm 4 can be obtained using different techniques such as mini-batch, SVRG, CV, or even zeroth-order methods. In this case, we use quantum variance reduction techniques to compute $\mathbf{g}(\mathbf{x}, \xi)$.

We propose to replace the gradients in HMC (See Algorithm 4 in appendix) with quantum gradients computed via Algorithm 3. Essentially Algorithm 3 combines the classical variance reduction techniques with the unbiased quantum mean estimation algorithm in Lemma 3.2.2 to reduce the variance further. The epoch length m for `QSVRG` determines the period where the full gradient needs to be computed. The parameter b is the quantum analog of batch size and will be determined analytically. To establish the convergence of the new samplers, we make the following assumptions in this section.

Assumption 3.3.1 (Strong Convexity)**.** There exists a positive constant μ such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ it holds that

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2. \tag{3.6}$$

Assumption 3.3.2 (Lipschitz Stochastic Gradients)**.** There exists a positive constant L such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and all functions f_i, $i = 1, ..., n$, it holds that

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|. \tag{3.7}$$

We also define the condition number $\kappa = \frac{L}{\mu}$. These assumptions are standard and used in the classical analysis of HMC [ZG21].

We start the convergence analysis with the following result in [ZG21] that quantifies the convergence of the stochastic Hamiltonian Monte Carlo algorithm in Wasserstein distance.

Theorem 3.3.3 (Theorem 4.4 in [ZG21])**.** *Under Assumptions 3.3.1 and 3.3.2, let* $D = \|\mathbf{x}^0 - \arg\min_{\mathbf{x}}(f(\mathbf{x}))\|$ *and* μ_T *be the distribution of the iterate* $\mathbf{x}^T$*, then if the step size satisfies* $\eta = O(L^{1/2}\sigma^{-2}\kappa^{-1} \wedge L^{-1/2})$ *and* $K = 1/(4\sqrt{L}\eta)$*, the output of HMC satisfies*

$$\mathrm{W}_2(\mu_T, \pi) \leq (1 - (128\kappa)^{-1})^{\frac{T}{2}}(2D + 2d/\mu)^{1/2} + \Gamma_1\eta^{1/2} + \Gamma_2\eta, \tag{3.8}$$

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

This is a generic result that applies to any HMC algorithm under Assumptions 3.3.1 and 3.3.2 that uses stochastic gradients with variance upper bounded by σ^2. Note that we do not assume a uniform upper bound for σ that is independent of problem parameters. Instead, the variance upper bound depends on the trajectory of the algorithm, which can be characterized using theoretical analysis.

The following auxiliary lemma is useful for the analysis in the later proofs.

Lemma 3.3.4. *Under Assumption 3.3.2, if the initial point satisfies* $\|\mathbf{x}^0 - \mathbf{x}^*\| \leq \frac{d}{\mu}$*, then it holds that*

$$\mathbb{E}_i\|\nabla f_i(\mathbf{x}_k) - \nabla f(\mathbf{x}_k)\|^2 \leq L^2\|\mathbf{x}_k - \widetilde{\mathbf{x}}\|^2, \tag{3.9}$$

where $\widetilde{\mathbf{x}} = \mathbf{x}_{k' < k}$ *is the last iteration the full gradient is computed.*

Proof. The proof simply follows from the definition of variance in the SVRG algorithm and the smoothness of each component.

$$\mathbb{E}_i\|\nabla f_i(\mathbf{x}_k) - \nabla f(\mathbf{x}_k)\|^2 \le \mathbb{E}_i\|\nabla f_i(\mathbf{x}_k) - \nabla f_i(\widetilde{\mathbf{x}}) + f(\widetilde{\mathbf{x}}) - \nabla f(\mathbf{x}_k)\|^2 \tag{3.10}$$

$$\le \mathbb{E}_i\|\nabla f_i(\mathbf{x}_k) - \nabla f_i(\widetilde{\mathbf{x}})\|^2 \tag{3.11}$$

$$\le L^2\|\mathbf{x}_k - \widetilde{\mathbf{x}}\|^2. \tag{3.12}$$

□

Lemma 3.3.4 allows us to set the target variance in quantum mean estimation to be $L^2\|\mathbf{x}_k - \widetilde{\mathbf{x}}\|/b^2$. Hence, each mean estimation call takes $\mathcal{O}(d^{1/2}b)$ gradient evaluations by Lemma 3.2.2.

The following lemma characterizes the variance of the stochastic gradients along the trajectory of `QHMC-SVRG`.

Lemma 3.3.5 (Modified Lemma C.2 in [ZG21])**.** *Let $\mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k)$ be the vector computed using the unbiased quantum mean estimation algorithm in* `QHMC-SVRG`*. Then, under Assumption 3.3.2,*

$$\mathbb{E}\|\mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k) - \nabla f(\mathbf{x}_k)\|^2 \le \frac{768m^2L^2\eta^2\kappa d}{b^2}, \tag{3.13}$$

where the expectation is over both the iterate $\mathbf{x}_k$ and the noise in quantum mean estimation $\boldsymbol{\xi}_k$.

Next, we state and prove the main theorem for `QSVRG-HMC`.

Theorem 3.3.6 (Main Theorem for `QSVRG-HMC`)**.** *Let μ_k be the distribution of $\mathbf{x}_k$ in* `QSVRG-HMC` *algorithm. Suppose that f satisfies Assumptions 3.3.1 and 3.3.2. Given that the initial point $\mathbf{x}_0$ satisfies $\|\mathbf{x}_0 - \arg\min_{\mathbf{x}} f(\mathbf{x})\| \le \frac{d}{\mu}$, then, for $\eta = \mathcal{O}\left(\frac{\epsilon}{L^{1/2}d^{1/2}\kappa^{3/2}}\right)$, $S = \tilde{\mathcal{O}}\left(\frac{Ld^{1/2}\kappa^{3/2}}{\epsilon}\right)$, $T = \tilde{\mathcal{O}}(1)$, $b = \mathcal{O}\left(\frac{L^{1/8}\epsilon^{1/4}n^{1/2}}{d^{1/8}\kappa^{3/8}} \vee 1\right)$, and $m = n/b$, we have*

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

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

Proof. By the choice of η in the theorem statement and the variance upper bound in Lemma 3.3.13, $\eta = \mathcal{O}(L^{1/2}\sigma^{-2}\kappa^{-1} \wedge L^{-1/2})$. Therefore, by Theorem 3.3.3, for $K = \frac{1}{4\sqrt{L}\eta}$, we have

$$\mathrm{W}_2(\mu_T, \pi) \le (1 - (128\kappa)^{-1})^{\frac{T}{2}}(2D + 2d/\mu)^{1/2} + \Gamma_1\eta^{1/2} + \Gamma_2\eta \tag{3.14}$$

where,

$$\Gamma_1^2 = \mathcal{O}\left(\frac{L^{1/2}m^2\kappa^3 d\eta^2}{b^2}\right), \tag{3.15}$$

$$\Gamma_2^2 = \mathcal{O}\left(\kappa^3 d + \frac{L^{3/2}m^2\kappa^3 d\eta^3}{b^2}\right). \tag{3.16}$$

We set $bm = \mathcal{O}(n)$. The first term in Equation (3.14) is $\mathcal{O}(\epsilon)$ when $T = \tilde{\mathcal{O}}(\log(1/\epsilon))$. The last two terms in Equation (3.8) for `QSVRG-HMC` become $\mathcal{O}\left(\frac{L^{1/4}d^{1/2}\kappa^{3/2}\eta^{3/2}n}{b^2} + d^{1/2}\kappa^{3/2}\eta\right)$. For $b = \mathcal{O}(d^{-1/8}\kappa^{-3/8}\epsilon^{1/4}n^{1/2}L^{1/8} \vee 1)$ and $\eta = \mathcal{O}(\epsilon\kappa^{-3/2}d^{-1/2})$, the bias term becomes $\mathcal{O}(\epsilon)$. Using Lemma 3.2.2, the number of gradient calculations scales as $\tilde{\mathcal{O}}(Ld^{1/2}\kappa^{3/2}\epsilon^{-1} + L^{9/8}d^{7/8}\kappa^{3/4}\epsilon^{-3/4}n^{1/2})$. □

Next we establish the complexity for quantum Hamiltonian Monte Carlo algorithm implemented with `QCV` technique. The lemma below characterizes the variance of the stochastic gradients along the trajectory of `QCV-HMC`.

Lemma 3.3.7 (Modified Lemma C.4 in [ZG21])**.** *Let $\mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k)$ be the vector computed using the unbiased quantum mean estimation algorithm in* `QCV-HMC`*. Then, under Assumption 3.3.2,*

$$\mathbb{E}\|\mathbf{g}(\mathbf{x}_k, \boldsymbol{\xi}_k) - \nabla f(\mathbf{x}_k)\|^2 \leq \frac{688Ld\kappa}{b^2},$$

where the expectation is over both the iterate $\mathbf{x}_k$ and the noise in quantum mean estimation $\boldsymbol{\xi}_k$.

Using this lemma, we can state and prove the main result for `QCV-HMC`.

Theorem 3.3.8 (Main Theorem for `QCV-HMC`)**.** *Let μ_k be the distribution of $\mathbf{x}_k$ in* `QCV-HMC` *algorithm. Suppose that f satisfies Assumptions 3.3.1 and 3.3.2. Given that the initial point $\mathbf{x}_0$ satisfies $\|\mathbf{x}_0 - \arg\min_{\mathbf{x}} f(\mathbf{x})\| \leq \frac{d}{\mu}$, then, for $\eta = \mathcal{O}\left(\frac{\epsilon}{L^{1/2}d^{1/2}\kappa^{3/2}}\right)$, $S = \tilde{\mathcal{O}}\left(\frac{Ld^{1/2}\kappa^{3/2}}{\epsilon}\right)$, $T = \tilde{\mathcal{O}}(1)$, and $b = \mathcal{O}\left(\frac{d^{1/4}\kappa^{3/4}}{L^{1/4}\epsilon^{1/2}} \vee 1\right)$, we have*

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

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

Proof. By the choice of η in the theorem statement and the variance upper bound in Lemma 3.3.13, $\eta = \mathcal{O}(L^{1/2}\sigma^{-2}\kappa^{-1} \wedge L^{-1/2})$. Therefore, by Theorem 3.3.3, for $K = \frac{1}{4\sqrt{L}\eta}$, we have

$$\mathrm{W}_2(\mu_T, \pi) \leq (1 - (128\kappa)^{-1})^{\frac{T}{2}}(2D + 2d/\mu)^{1/2} + \Gamma_1\eta^{1/2} + \Gamma_2\eta, \tag{3.17}$$

where,

$$\Gamma_1 = \mathcal{O}\left(\frac{L^{-1/2}\kappa^3 d}{b^2}\right), \tag{3.18}$$

$$\Gamma_2 = \mathcal{O}\left(\kappa^3 d\right). \tag{3.19}$$

The first term in Equation (3.17) is $\mathcal{O}(\epsilon)$ when $T = \tilde{\mathcal{O}}(1)$. The last two terms in Equation (3.17) for `QCV-HMC` become $\mathcal{O}\left(\frac{L^{-1/4}d^{1/2}\kappa^{3/2}\eta^{1/2}}{b^2} + d^{1/2}\kappa^{3/2}\eta\right)$. For $b = \mathcal{O}(L^{-1/4}d^{1/4}\kappa^{3/4}\epsilon^{-1/2} \vee 1)$ and $\eta = \mathcal{O}(\epsilon d^{-1/2}\kappa^{-3/2})$, the bias term becomes $\mathcal{O}(\epsilon)$. Using Lemma 3.2.2, the number of gradient calculations scales as $\tilde{O}(Ld^{1/2}\kappa^{3/2}\epsilon^{-1} + L^{3/4}d^{5/4}\kappa^{9/4}\epsilon^{-3/2}) = \tilde{O}(Ld^{5/4}\kappa^{9/4}\epsilon^{-3/2})$. □

Theorems 3.3.6 and 3.3.8 imply that when $n = \mathcal{O}(\epsilon^{-1/2})$ the best classical (`SVRG-HMC`) and the best quantum (`QSVRG-HMC`) algorithms have $\tilde{\mathcal{O}}(\epsilon^{-1})$ gradient complexity. On the other hand, when $n = \omega(\epsilon^{-1})$, quantum algorithms have better complexity than the best classical algorithms, where the race between `QSVRG-HMC` and `QCV-HMC` depends on how large n is.

Remark 3.3.9. Both the classical algorithms in [ZG21] and quantum algorithms in this work assume that the starting point is (d/μ)-close to the minimizer $\mathbf{x}^* = \arg\min f(\mathbf{x})$. In case this point is not given, it can be obtained using $\mathcal{O}(n)$ iterations of SGD [BFFN19].

3.3.2 Sampling under Log-Sobolev Inequality via Langevin Monte Carlo

We use SVRG-LMC for the base algorithm in [KS22] and replace the stochastic gradient calculation with unbiased quantum mean estimation. This section generalizes the strong convexity assumption with the following LSI assumption, which is common in non-log-concave sampling.

Assumption 3.3.10 (Log-Sobolev Inequality)**.** We say that π satisfies the Log-Sobolev inequality with constant α if for all ρ, it holds that

$$\mathrm{KL}(\rho||\pi) \leq \frac{1}{2\alpha}\mathrm{FI}(\rho||\pi). \tag{3.20}$$

We start by proving the following lemma that shows one-step convergence of LMC with stochastic gradients.

Lemma 3.3.11 (Stochastic-LMC One Step Convergence)**.** *Let μ_k be the distribution of the iterate $\mathbf{x}_k$, then if the step size satisfies $\eta = \frac{2}{3\alpha}$,*

$$\mathrm{KL}(\mu_{k+1}||\pi) \leq e^{-3\alpha\eta/2}\left[\left(1 + \frac{32\eta^3 L^4}{\alpha}\right)\mathrm{KL}(\mu_k||\pi) + 6\eta\sigma_k^2 + 16\eta^2 dL^2\right], \tag{3.21}$$

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

Proof. We compare one step of LMC starting at $\mathbf{x}_k$ with stochastic gradients $\mathbf{g}(\mathbf{x}_k,\boldsymbol{\xi}_k)$ to the output of continuous Langevin SDE (Equation (2.2)) starting at $\mathbf{x}_k$ with true gradient $\nabla f(\mathbf{x}_t)$ after time η. This technique has been used to establish the convergence of unadjusted Langevin algorithm with full gradients under isoperimetry by [VW19]. We extend the analysis by [VW19] to the stochastic gradient LMC. Assume that the initial point $\mathbf{x}_k$ and $\mathbf{g}(\mathbf{x}_k,\boldsymbol{\xi}_k)$ obey the joint distribution μ_0. The randomness on $\mathbf{g}(\mathbf{x}_k,\boldsymbol{\xi}_k)$ depends both on the randomness on $\mathbf{x}_k$ and the randomness in the quantum mean estimation algorithm. Then, one step update of LMC algorithm with stochastic gradient yields,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\mathbf{g}(\mathbf{x}_k,\boldsymbol{\xi}_k) + \sqrt{2\eta}\boldsymbol{\epsilon}_k. \tag{3.22}$$

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

$$d\mathbf{x}_t = -\mathbf{g}_k dt + \sqrt{2}d\boldsymbol{W}_t \tag{3.23}$$

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

Then, the Fokker Planck equation gives the following evolution for the marginal density $\mu_t(\mathbf{x}|\mathbf{x}_k,\mathbf{g}_k) = \mu_t(\mathbf{x}_t = \mathbf{x}|\mathbf{x}_k,\mathbf{g}_k)$,

$$\frac{\partial\mu_t(\mathbf{x}|\mathbf{x}_k,\mathbf{g}_k)}{\partial t} = \nabla\cdot(\mu_t(\mathbf{x}|\mathbf{x}_k,\mathbf{g}_k)\mathbf{g}_k) + \Delta\mu_t(\mathbf{x}|\mathbf{x}_k,\mathbf{g}_k). \tag{3.24}$$

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

$$\begin{aligned}
&\frac{\partial\mu_t(\mathbf{x})}{\partial t} \\
&= \mathbb{E}_{(\mathbf{x}_k,\mathbf{g}_k)\sim\mu_0}[\nabla\cdot(\mu_t(\mathbf{x}|\mathbf{x}_k)\mathbf{g}_k)] + \mathbb{E}_{(\mathbf{x}_k,\mathbf{g}_k)\sim\mu_0}[\Delta\mu_t(\mathbf{x}|\mathbf{x}_k)] &(3.25)\\
&= \int_{\mathbb{R}^d}\nabla\cdot(\mu_t(\mathbf{x}|\mathbf{x}_k,\mathbf{g}_k)\mathbf{g}_k)\mu_0(\mathbf{x}_k,\mathbf{g}_k)d\mathbf{x}_k d\mathbf{g}_k + \int_{\mathbb{R}^d}\Delta\mu_t(\mathbf{x}|\mathbf{x}_k,\mathbf{g}_k)\mu_0(\mathbf{x}_k,\mathbf{g}_k)d\mathbf{x}_k d\mathbf{g}_k &(3.26)
\end{aligned}$$

$$= \int_{\mathbb{R}^d} \nabla \cdot (\mu_t(\mathbf{x})\mu(\mathbf{x}_k, \mathbf{g}_k|\mathbf{x}_t = \mathbf{x})\mathbf{g}_k) d\mathbf{x}_k d\mathbf{g}_k + \Delta\mu_t(\mathbf{x}) \tag{3.27}$$

$$= \nabla \cdot \left(\mu_t(\mathbf{x}) \mathbb{E}[\mathbf{g}_k - \nabla f(\mathbf{x}_k)|\mathbf{x}_t = \mathbf{x}] + \mu_t(\mathbf{x}) \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right). \tag{3.28}$$

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

$$\frac{d}{dt} \mathrm{KL}(\mu_t||\pi) = \frac{d}{dt} \int_{\mathbb{R}^d} \mu_t(\mathbf{x}) \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) d\mathbf{x} \tag{3.29}$$

$$= \int_{\mathbb{R}^d} \frac{\partial \mu_t(\mathbf{x})}{\partial t} \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) d\mathbf{x}_t + \int_{\mathbb{R}^d} \mu_t(\mathbf{x}) \frac{\partial}{\partial t} \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) d\mathbf{x} \tag{3.30}$$

$$= \int_{\mathbb{R}^d} \frac{\partial \mu_t(\mathbf{x})}{\partial t} \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) d\mathbf{x}_t + \int_{\mathbb{R}^d} \frac{\partial \mu_t(\mathbf{x})}{\partial t} d\mathbf{x} \tag{3.31}$$

$$= \int_{\mathbb{R}^d} \frac{\partial \mu_t(\mathbf{x})}{\partial t} \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) d\mathbf{x}_t. \tag{3.32}$$

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

$$\frac{d}{dt} \mathrm{KL}(\mu_t||\pi)$$

$$= \int_{\mathbb{R}^d} \nabla \cdot \left(\mu_t(\mathbf{x}) \mathbb{E}[\mathbf{g}_k - \nabla f(\mathbf{x})|\mathbf{x}_t = \mathbf{x}] + \mu_t(\mathbf{x}) \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right) \log \left(\frac{\mu_t(x)}{\pi(\mathbf{x})} \right) d\mathbf{x} \tag{3.33}$$

$$= -\int_{\mathbb{R}^d} \mu_t(\mathbf{x}) \left\langle \mathbb{E}[\mathbf{g}_k - \nabla f(\mathbf{x})|\mathbf{x}_t = \mathbf{x}] + \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right), \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right\rangle d\mathbf{x} \tag{3.34}$$

$$= -\int_{\mathbb{R}^d} \mu_t(\mathbf{x}) \left\| \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right\|^2 d\mathbf{x} + \mathbb{E} \left\langle \nabla f(\mathbf{x}_t) - \mathbf{g}_k, \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right\rangle. \tag{3.35}$$

The second term can be bounded as follows:

$$\mathbb{E} \left\langle \nabla f(\mathbf{x}_t) - \mathbf{g}_k, \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right\rangle \tag{3.36}$$

$$\leq \mathbb{E} \left[\|\nabla f(\mathbf{x}_t) - \mathbf{g}_k\|^2 + \frac{1}{4} \left\| \nabla \log \left(\frac{\mu_t(\mathbf{x})}{\pi(\mathbf{x})} \right) \right\|^2 \right] \tag{3.37}$$

$$= \mathbb{E}\|\nabla f(\mathbf{x}_t) - \mathbf{g}_k\|^2 + \frac{1}{4} \mathrm{FI}(\mu_t||\pi) \tag{3.38}$$

$$= \mathbb{E}\|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 + \frac{1}{4} \mathrm{FI}(\mu_t||\pi) \tag{3.39}$$

$$\leq 2\mathbb{E}\|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_k)\|^2 + 2\mathbb{E}_{\mu_t(\mathbf{x}_t,\mathbf{x}_k)}\|\nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 \tag{3.40}$$

$$+ \frac{1}{4}\mathrm{FI}(\mu_t||\pi). \tag{3.41}$$

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

$$\mathbb{E}\|\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_k)\|^2 \leq L^2\mathbb{E}\|\mathbf{x}_t - \mathbf{x}_k\|^2 \tag{3.42}$$

$$\leq L^2\mathbb{E}\| - t\mathbf{g}_k + \sqrt{2t}\boldsymbol{\epsilon}_k\|^2 \tag{3.43}$$

$$= t^2L^2\mathbb{E}_{\mu_0}\|\mathbf{g}_k\|^2 + 2tdL^2. \tag{3.44}$$

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

$$\frac{d}{dt}\,\mathrm{KL}(\mu_t||\pi) \leq -\frac{3}{4}\mathrm{FI}(\mu_t||\pi) + 2t^2L^2\mathbb{E}_{\mu_0}\|\mathbf{g}_k\|^2 + 2\mathbb{E}_{\mu_0}\|\nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 + 4tdL^2 \tag{3.45}$$

$$\leq -\frac{3}{4}\mathrm{FI}(\mu_t||\pi) + (4t^2L^2 + 2)\mathbb{E}_{\mu_0}\|\nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 \tag{3.46}$$

$$+ 4t^2L^2\mathbb{E}_{\mu_0}\|\nabla f(\mathbf{x}_k)\|^2 + 4tdL^2. \tag{3.47}$$

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

$$\mathbb{E}_{\mu_0}\|\nabla f(\mathbf{x}_k)\|^2 \leq 2\mathbb{E}_{\mu_0}\|\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}^*)\|^2 + 2\mathbb{E}_{\mu_0}\|\nabla f(\mathbf{x}^*)\|^2 \tag{3.48}$$

$$\leq 2L^2\mathbb{E}_{\mu_0}\|\mathbf{x}_k - \mathbf{x}_0\|^2 + 2\mathbb{E}_{\mu_0}\|\nabla f(\mathbf{x}^*)\|^2 \tag{3.49}$$

$$\leq 2L^2\mathrm{W}_2(\mu_0, \pi)^2 + 2dL \tag{3.50}$$

$$\leq \frac{4L^2}{\alpha}\,\mathrm{KL}(\mu_0||\pi) + 2dL. \tag{3.51}$$

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

$$\frac{d}{dt}\,\mathrm{KL}(\mu_t||\pi) \leq -\frac{3}{4}\mathrm{FI}(\mu_t||\pi) + (4t^2L^2 + 2)\mathbb{E}_{\mu_0}\|\nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 \tag{3.52}$$

$$+ \frac{16t^2L^4}{\alpha}\,\mathrm{KL}(\mu_0||\pi) + 4tdL^2 + 8t^2dL^3 \tag{3.53}$$

$$\leq -\frac{3\alpha}{2}\,\mathrm{KL}(\mu_t||\pi) + (4t^2L^2 + 2)\mathbb{E}_{\mu_0}\|\nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 \tag{3.54}$$

$$+ \frac{16t^2L^4}{\alpha}\,\mathrm{KL}(\mu_0||\pi) + 4tdL^2 + 8t^2dL^3 \tag{3.55}$$

$$\leq -\frac{3\alpha}{2}\,\mathrm{KL}(\mu_t||\pi) + 3\mathbb{E}_{\mu_0}\|\nabla f(\mathbf{x}_k) - \mathbf{g}_k\|^2 \tag{3.56}$$

$$+ \frac{16\eta^2 L^4}{\alpha}\,\mathrm{KL}(\mu_0||\pi) + 8\eta d L^2 \tag{3.57}$$

$$\leq -\frac{3\alpha}{2}\,\mathrm{KL}(\mu_t||\pi) + 3\sigma_k^2 + \frac{16\eta^2 L^4}{\alpha}\,\mathrm{KL}(\mu_0||\pi) + 8\eta d L^2. \tag{3.58}$$

The second inequality is due to Equation (3.20). Equivalently, we can write,

$$\frac{d}{dt}(e^{3\alpha t/2}\,\mathrm{KL}(\mu_t||\pi)) \leq e^{3\alpha t/2}\left(3\sigma_k^2 + \frac{16\eta^2 L^4}{\alpha}\,\mathrm{KL}(\mu_0||\pi) + 8\eta d L^2\right). \tag{3.59}$$

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

$$e^{3\alpha\eta/2}\,\mathrm{KL}(\mu_\eta||\pi) - \mathrm{KL}(\mu_0||\pi) \leq 6\eta\sigma_k^2 + \frac{32\eta^3 L^4}{\alpha}\,\mathrm{KL}(\mu_0||\pi) + 16\eta^2 d L^2 \tag{3.60}$$

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

$$\mathrm{KL}(\mu_\eta||\pi) \leq e^{-3\alpha\eta/2}\left[\left(1 + \frac{32\eta^3 L^4}{\alpha}\right)\mathrm{KL}(\mu_0||\pi) + 6\eta\sigma_k^2 + 16\eta^2 d L^2\right]. \tag{3.61}$$

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

□

The statement in Lemma 3.3.11 is generic and can be applied to any LMC algorithm with stochastic gradients with bounded variance on the trajectory of the algorithm. Note that this is different from assuming that the variance is uniformly upper bounded. Instead, we set inner loop and variance reduction parameters so that the variance does not explode along the trajectory of the algorithm.

We continue with the following lemma that characterizes the variance of the quantum stochastic gradients in `QSVRG-LMC` in terms of the distance between the current iterate and the reference point where the full gradient is computed.

Lemma 3.3.12. *Let $\tilde{\mathbf{x}}$ be any iteration where* `QSVRG-LMC` *computes the full gradient. Then under Assumption 3.3.2, the quantum stochastic gradient $\mathbf{g}_k$ at $\mathbf{x}_k$ that is computed using $\tilde{\mathbf{x}}$ as a reference point in* `QSVRG-LMC` *satisfies*

$$\mathbb{E}[\|\mathbf{g}_k - \nabla f(\mathbf{x}_k)\|^2] \leq \frac{L^2\|\mathbf{x}_k - \tilde{\mathbf{x}}\|^2}{b^2} \tag{3.62}$$

using $\tilde{O}(d^{1/2}b)$ gradient computations.

Proof. Recall that SVRG computes the stochastic gradient $\tilde{\mathbf{g}}$ at $\mathbf{x}_k$ by the following.

$$\tilde{\mathbf{g}}_k = \nabla f_i(\mathbf{x}_k) - \nabla f_i(\tilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}}), \tag{3.63}$$

where $\tilde{\mathbf{x}}$ is the last iteration the full gradient is computed and i is a component randomly chosen from $[n]$. Let $\sigma_k^2 = \mathbb{E}\|\tilde{\mathbf{g}}_k - \nabla f(\mathbf{x}_k)\|^2$. Then, σ_k^2 can be bounded in terms of the distance between $\mathbf{x}_k$ and $\tilde{\mathbf{x}}$.

$$\begin{aligned}
\sigma_k^2 &= \mathbb{E}[\|\nabla f_i(\mathbf{x}_k) - \nabla f_i(\tilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}}) - \nabla f(\mathbf{x}_k)\|^2] && (3.64)\\
&= \mathbb{E}[\|\nabla f_i(\mathbf{x}_k) - \nabla f_i(\tilde{\mathbf{x}})\|^2] - (\mathbb{E}[\nabla f_i(\mathbf{x}_k) - \nabla f_i(\tilde{\mathbf{x}})])^2 && (3.65)\\
&\leq \mathbb{E}[\|\nabla f_i(\mathbf{x}_k) - \nabla f_i(\tilde{\mathbf{x}})\|^2] && (3.66)\\
&\leq L^2\|\mathbf{x}_k - \tilde{\mathbf{x}}\|^2, && (3.67)
\end{aligned}$$

where the equality follows from the fact that ∇f_i is an unbiased estimator for ∇f and the last line follows from Assumption 3.3.2. Hence, using unbiased quantum mean estimation in Lemma 3.2.2, we can obtain a random vector $\mathbf{g}_k$ such that,

$$\mathbb{E}\|\mathbf{g}_k - \nabla f(\mathbf{x}_k)\|^2 \leq \frac{L^2\|\mathbf{x}_k - \tilde{\mathbf{x}}\|^2}{b^2} \tag{3.68}$$

by using $\tilde{O}(d^{1/2}b)$ calls to the gradient oracle. □

To be able to apply Lemma 3.3.11, we need to characterize the expected upper bound on the variance of the stochastic gradients over the algorithm trajectory for SVRG.

Lemma 3.3.13 (`QSVRG-LMC` Variance Lemma)**.** *Let $k' < k$ be the last iteration where the full gradient is computed in* `QSVRG-LMC` *and $\sigma_k^2 = \mathbb{E}\|\mathbf{g}_k - \nabla f(\mathbf{x}_k)\|^2$. Then, for $\eta^2 \leq \frac{1}{6L^2m^2}$,*

$$\sigma_{k'+l}^2 \leq \frac{16L^4\eta^2}{\alpha}\sum_{r=1}^{l}\mathrm{KL}(\mu_{k'+r-1}||\pi) + \frac{8\eta dmL^2}{b^2}. \tag{3.69}$$

Proof. Let $\tilde{\mathbf{x}} = \mathbf{x}_{k'}$. Then, by Lemma 3.3.12, quantum stochastic gradient $\mathbf{g}_k$ satisfies

$$\mathbb{E}[\|\mathbf{g}_k - \nabla f(\mathbf{x}_k)\|^2] \leq \frac{L^2\mathbb{E}\|\mathbf{x}_k - \tilde{\mathbf{x}}\|^2}{b^2}. \tag{3.70}$$

Let $\tilde{\mathbf{x}} = \mathbf{y}_0$ and $\mathbf{x}_k = \mathbf{y}_k$, then using the update rule of Langevin Monte Carlo,

$$\mathbb{E}[\|\mathbf{x}_k - \tilde{\mathbf{x}}\|^2] = \mathbb{E}\left[\left\|\sum_{r=1}^{l}(\mathbf{y}_r - \mathbf{y}_{r-1})\right\|^2\right] = \mathbb{E}\left[\left\|\sum_{r=1}^{l} -\eta\mathbf{g}_{r-1} + \sqrt{2\eta}\boldsymbol{\epsilon}_{r-1})\right\|^2\right] \tag{3.71}$$

$$\leq \mathbb{E}\left[2\eta^2\left\|\sum_{r=1}^{l}\mathbf{g}_{r-1}\right\|^2 + 4\eta\left\|\sum_{r=1}^{l}\boldsymbol{\epsilon}_{r-1}\right\|^2\right] \tag{3.72}$$

$$\leq 2\eta^2 m\sum_{r=1}^{l}\mathbb{E}\left\|\mathbf{g}_{r-1}\right\|^2 + 4\eta\sum_{r=1}^{l}\left\|\boldsymbol{\epsilon}_{r-1}\right\|^2 \tag{3.73}$$

$$\leq 2\eta^2 m\sum_{r=1}^{l}\mathbb{E}\left\|\mathbf{g}_{r-1}\right\|^2 + 4\eta dm. \tag{3.74}$$

The first inequality is due to Young's inequality and the second inequality follows from the fact that the Gaussian noises at different iterations are independent and the fact that $l \leq m$. Defining $\sigma^2_{\max} = \max_k \mathbb{E}\|\sigma_k\|^2$, we can write the first term on the right-hand side in terms of $\sigma^2_{\max}$,

$$\mathbb{E}[\|\mathbf{g}_r\|^2] = \mathbb{E}\|\mathbf{g}_r - \nabla f(\mathbf{x}_r) + \nabla f(\mathbf{x}_r)\|^2 \tag{3.75}$$

$$\leq 2\mathbb{E}\|\mathbf{g}_r - \nabla f(\mathbf{x}_r)\|^2 + 2\|\nabla f(\mathbf{x}_r)\|^2 \tag{3.76}$$

$$\leq 2\sigma^2_{\max} + \frac{8L^2}{\alpha}\,\mathrm{KL}(\mu_r||\pi) + 4dL, \tag{3.77}$$

and using Equation (3.70),

$$\sigma^2_{\max} \leq \frac{4L^2m^2\eta^2\sigma^2_{\max}}{b^2} + \frac{16L^4\eta^2 m}{b^2\alpha}\sum_{r=1}^{l}\mathrm{KL}(\mu_{r-1}||\pi) + \frac{8dL^3\eta^2m^2}{b^2} + \frac{4\eta dmL^2}{b^2}. \tag{3.78}$$

If we set $\eta^2 \leq \frac{1}{6L^2m^2}$, we obtain

$$\sigma^2_{k'+l} \leq \frac{32L^4\eta^2 m}{b^2\alpha}\sum_{r=1}^{l}\mathrm{KL}(\mu_{r-1}||\pi) + \frac{8\eta dmL^2}{b^2}. \tag{3.79}$$

□

Theorem 3.3.14 (Convergence theorem for `QSVRG-LMC`)**.** *Assume that $m \leq b^2$. Then, for $\eta \leq \frac{\alpha^2}{24L^2m}$, the iterates in* `QSVRG-LMC` *satisfy,*

$$\mathrm{KL}(\mu_k||\pi) \leq e^{-\alpha\eta k}\,\mathrm{KL}(\mu_0\|\pi) + \frac{64m\eta dL^2}{\alpha b^2} + \frac{24\eta dL^2}{\alpha}. \tag{3.80}$$

Proof. Let $l < k$ be the last iteration the full gradient is computed. Then, using Lemmas 3.3.11 and 3.3.13, we can write one step bound as follows.

$$\mathrm{KL}(\mu_{k+1}||\pi) \leq e^{-3\alpha\eta/2}\Bigg[\left(1+\frac{32\eta^3L^4}{\alpha}\right)\mathrm{KL}(\mu_k||\pi) + \frac{192m\eta^3L^4}{b^2\alpha}\sum_{r=l}^{k}\mathrm{KL}(\mu_r||\pi) + \frac{48m\eta^2dL^2}{b^2} + 16\eta^2dL^2\Bigg]. \tag{3.81}$$

First, we claim that the following inequality is true.

$$\mathrm{KL}(\mu_{k+1}||\pi) \leq e^{-\alpha\eta k}\,\mathrm{KL}(\mu_0\|\pi) + \frac{48m\eta^2dL^2 + 16\eta^2dL^2b^2}{b^2(1-e^{-\alpha\eta})}. \tag{3.82}$$

To prove Equation (3.82), we use induction. For $k = 1$, the statement holds due to Equation (3.81). That is,

$$\begin{aligned}
\mathrm{KL}(\mu_1||\pi) &\leq e^{-3\alpha\eta/2}\left[\left(1+\frac{224\eta^3L^4}{\alpha}\right)\mathrm{KL}(\mu_0||\pi) + \frac{48m\eta^2dL^2}{b^2} + 16\eta^2dL^2\right] && (3.83)\\
&\leq e^{-\alpha\eta}\,\mathrm{KL}(\mu_0\|\pi) + \frac{48m\eta^2dL^2}{b^2} + 16\eta^2dL^2 && (3.84)\\
&\leq e^{-\alpha\eta}\,\mathrm{KL}(\mu_0\|\pi) + \frac{48m\eta^2dL^2 + 16\eta^2dL^2b^2}{b^2(1-e^{-\alpha\eta})}. && (3.85)
\end{aligned}$$

The first inequality is due to the fact that $m \leq b^2$. The second inequality holds since $\left(1+\frac{224\eta^3L^4}{\alpha}\right) \leq \left(1+\frac{\eta\alpha}{2}\right) \leq e^{\alpha\eta/2}$ since $\eta \leq \frac{\alpha}{24L^2m}$. The third inequality follows from the fact that $1-e^{-\alpha\eta} \leq 1$. Next, assume that the statement holds for $k-1$, and then we prove the k-th step of induction.

$$\begin{aligned}
\mathrm{KL}(\mu_k||\pi) &\leq e^{-3\alpha\eta/2}\Bigg[\left(1+\frac{32\eta^3L^4}{\alpha}\right)\mathrm{KL}(\mu_{k-1}||\pi) + \frac{192\eta^3L^4}{\alpha}\sum_{r=\ell}^{k-1}\mathrm{KL}(\mu_\ell||\pi)\\
&\quad + \frac{48m\eta^2dL^2 + 16\eta^2dL^2b^2}{b^2}\Bigg] && (3.86)\\
&\leq e^{-3\alpha\eta/2}\left(1+\frac{32\eta^3L^4}{\alpha}\right)\left(e^{-\alpha\eta(k-1)}\,\mathrm{KL}(\mu_0\|\pi) + \frac{48m\eta^2dL^2 + 16\eta^2dL^2b^2}{b^2(1-e^{-\alpha\eta})}\right)\\
&\quad + e^{-3\alpha\eta/2}\frac{192\eta^3L^4}{\alpha}\sum_{r=l}^{k-1}\left(e^{-\alpha\eta r}\,\mathrm{KL}(\mu_0\|\pi) + \frac{48m\eta^2dL^2 + 16\eta^2dL^2b^2}{b^2(1-e^{-\alpha\eta})}\right)
\end{aligned}$$

$$+ \frac{48m\eta^2 dL^2 + 16\eta^2 dL^2 b^2}{b^2} \tag{3.87}$$

$$\leq e^{-3\alpha\eta/2}\left(1 + \frac{32\eta^3 L^4}{\alpha}\right)\left(e^{-\alpha\eta(k-1)}\,\mathrm{KL}(\mu_0\|\pi) + \frac{48m\eta^2 dL^2 + 16\eta^2 dL^2 b^2}{b^2(1-e^{-\alpha\eta})}\right)$$
$$+ e^{-3\alpha\eta/2}\frac{192m\eta^3 L^4}{\alpha}e^{m\alpha\eta}\left(e^{-\alpha\eta(k-1)}\,\mathrm{KL}(\mu_0\|\pi) + \frac{48m\eta^2 dL^2 + 16\eta^2 dL^2 b^2}{b^2(1-e^{-\alpha\eta})}\right)$$
$$+ \frac{48m\eta^2 dL^2 + 16\eta^2 dL^2 b^2}{b^2} \tag{3.88}$$

$$\leq e^{-3\alpha\eta/2}\left(1 + \frac{32\eta^3 L^4}{\alpha}\right)\left(e^{-\alpha\eta(k-1)}\,\mathrm{KL}(\mu_0\|\pi) + \frac{48m\eta^2 dL^2 + 16\eta^2 dL^2 b^2}{b^2(1-e^{-\alpha\eta})}\right)$$
$$+ e^{-3\alpha\eta/2}\frac{96m\eta^3 L^4}{\alpha}\left(e^{-\alpha\eta(k-1)}\,\mathrm{KL}(\mu_0\|\pi) + \frac{48m\eta^2 dL^2 + 16\eta^2 dL^2 b^2}{b^2(1-e^{-\alpha\eta})}\right)$$
$$+ \frac{48m\eta^2 dL^2 + 16\eta^2 dL^2 b^2}{b^2} \tag{3.89}$$

$$\leq e^{-3\alpha\eta/2}\left(1 + \frac{128\eta^3 L^4}{\alpha}\right)\left(e^{-\alpha\eta(k-1)}\,\mathrm{KL}(\mu_0\|\pi) + \frac{48m\eta^2 dL^2 + 16\eta^2 dL^2 b^2}{b^2(1-e^{-\alpha\eta})}\right)$$
$$+ \frac{48m\eta^2 dL^2 + 16\eta^2 dL^2 b^2}{b^2} \tag{3.90}$$

$$\leq e^{-\alpha\eta}\left(e^{-\alpha\eta(k-1)}\,\mathrm{KL}(\mu_0\|\pi) + \frac{48m\eta^2 dL^2 + 16\eta^2 dL^2 b^2}{b^2(1-e^{-\alpha\eta})}\right)$$
$$+ \frac{48m\eta^2 dL^2 + 16\eta^2 dL^2 b^2}{b^2} \tag{3.91}$$

$$\leq e^{-\alpha\eta k}\,\mathrm{KL}(\mu_0\|\pi) + \frac{48m\eta^2 dL^2 + 16\eta^2 dL^2 b^2}{b^2(1-e^{-\alpha\eta})} \tag{3.92}$$

$$\leq e^{-\alpha\eta k}\,\mathrm{KL}(\mu_0\|\pi) + \frac{64m\eta dL^2 + 24\eta dL^2 b^2}{\alpha b^2}. \tag{3.93}$$

The first two inequalities are due to Equation (3.81). The third and fourth inequality follow from the fact that $k - l \leq m$ and $e^{m\alpha\eta} \leq e^{\frac{\alpha^2}{8L^2}} \leq e^{\frac{1}{8}} \leq \frac{1}{2}$ for $\eta \leq \frac{\alpha}{8mL^2}$ and the fifth inequality holds since $\left(1 + \frac{128\eta^3 L^4}{\alpha}\right) \leq \left(1 + \frac{\eta\alpha}{2}\right) \leq e^{\alpha\eta/2}$ for $\eta \leq \frac{\alpha}{24L^2 m}$. The final inequality follows from the fact that $1 - e^{-\alpha\eta} \geq \frac{3}{4}\alpha\eta$ when $\alpha\eta \leq \frac{1}{4}$. This concludes the proof. □

Theorem 3.3.15 (Main Theorem for `QSVRG-LMC`)**.** *Let μ_k be the distribution of $\mathbf{x}_k$ in* `QSVRG-LMC` *algorithm. Suppose that f satisfies Assumptions 3.3.2 and 3.3.10. Then for $\eta = \mathcal{O}\left(\frac{\epsilon\alpha}{dL^2} \wedge \frac{\alpha}{L^2 m}\right)$, $K = \tilde{\mathcal{O}}\left(\frac{L^2 \log(\mathrm{KL}(\mu_0\|\pi))}{\alpha^2}\left(n^{2/3} + \frac{d}{\epsilon}\right)\right)$, $b = \tilde{\mathcal{O}}(n^{1/3})$, and $m = \tilde{\mathcal{O}}(n^{2/3})$ we have*

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

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

Proof. Setting $b = \tilde{\mathcal{O}}(n^{1/3})$ and $m = \tilde{\mathcal{O}}(n^{2/3})$ and $\eta \leq \frac{\epsilon\alpha}{176dL^2}$ the second term on the right hand side of Theorem 3.3.14 becomes smaller than $\epsilon/2$. By the step size requirement of Theorem 3.3.14, we have $\eta \leq \frac{\epsilon\alpha}{176dL^2} \wedge \frac{\alpha}{24L^2m}$. The first term in Theorem 3.3.14 is smaller than $\epsilon/2$ when $K \leq \frac{\log(2\,\mathrm{KL}(\mu_0\|\pi)/\epsilon)}{\alpha\eta}$. Hence TV distance is smaller than ϵ. The results for W_2 distance and TV distance hold due to Talagrand's inequality [OV00] and Pinsker's inequality [Tsy09] respectively. The total gradient complexity is $bK = \tilde{\mathcal{O}}\left(\frac{L^2\,\mathrm{KL}(\mu_0\|\pi)}{\alpha^2}\left(nd^{1/2} + \frac{d^{3/2}n^{1/3}}{\epsilon}\right)\right)$. □

Our algorithm improves the dominant term in gradient complexity from $\tilde{\mathcal{O}}(n^{1/2}\epsilon^{-1})$ to $\tilde{\mathcal{O}}(n^{1/3}\epsilon^{-1})$. It is also worth mentioning that recently [HZD+24] proposed a proximal sampling algorithm that uses $\tilde{\mathcal{O}}(\sigma^2\epsilon^{-1})$ gradient queries in the LSI setting when the stochastic gradients have bounded variance σ^2. However, this assumption is different from our setting since the variance in the stochastic gradients is not uniformly bounded by a constant, but it is bounded throughout the trajectory by a function of problem parameters such as d, b, m, L, α (See Lemma 3.3.13).

3.4 Quantum Gradient Estimation in Zeroth-Order Stochastic Setting

In this section, we assume access to a zeroth-order oracle rather than a gradient oracle. This approach is useful in scenarios where gradients are not available or where computing gradients is more expensive than evaluating the function. Specifically, we consider access to an evaluation oracle for the stochastic components $f_\xi(\mathbf{x}) = f(\mathbf{x};\xi)$, where $\xi \in \Xi$ represents a random seed characterizing the stochasticity. Then, the stochastic evaluation oracle is given by

$$O_f \left|\mathbf{x}\right\rangle \left|\xi\right\rangle \left|0\right\rangle \mapsto \left|\mathbf{x}\right\rangle \left|\xi\right\rangle \left|f(\mathbf{x};\xi)\right\rangle . \tag{3.94}$$

We characterize the complexity of our algorithms in this section with respect to this oracle. Before presenting our algorithms, we give a brief overview of Jordan's algorithm in the next section.

3.4.1 Overview of Jordan's algorithm

Jordan's algorithm [Jor05] approximates the gradient using a finite difference formula on a small grid around the point of interest and encodes the estimate into the quantum phase.

Then, the algorithm applies an inverse quantum Fourier transform to estimate the gradient. Although Jordan's original analysis implicitly assumes that higher-order derivatives of the function are negligible, Gilyén, Arunachalam, and Wiebe [GSLW19] analyzed the algorithm and extended it to handle functions in the Gevrey class, using central difference formulas and a binary oracle model commonly encountered in variational quantum algorithms. The closest analysis of Jordan's algorithm to our setting was provided by [CCLW20], who demonstrated that Algorithm 5 achieves constant query complexity for functions with Lipschitz gradients, provided that the function values can be queried with high precision.

Algorithm 5 `QuantumGradient`$(f, \epsilon, L, \beta, x_0)$

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

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

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

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

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

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

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

The following lemma from [CCLW20] demonstrates that Algorithm 5 achieves $\tilde{O}(1)$ query complexity for evaluating the gradient of a β-smooth function with high probability.

Lemma 3.4.1 (Lemma 2.2 in [CCLW20])**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function that is accessible via an evaluation oracle with error at most ϵ. Assume that $\|\nabla f\| \leq L$ and f is β-smooth*

in $B_\infty(x, 2\sqrt{\epsilon/\beta})$. *Let* $\tilde{\mathbf{g}}$ *be the output of* `QuantumGradient`$(f, \epsilon, M, \beta, \mathbf{x}_0)$ *(as defined in Algorithm 5). Then:*

$$\Pr\left[|\tilde{\mathbf{g}}_i - \nabla f(\mathbf{x})_i| > 1500\sqrt{d\epsilon\beta}\right] < \frac{1}{3}, \quad \forall\, i \in [d]. \tag{3.95}$$

Although Algorithm 5 results in an accurate estimate for the gradient with high probability, it is possible to run the algorithm multiple times and take the coordinate-wise median of the outputs to obtain a smooth estimate for the gradient (Lemma 2.3 in [CCLW20]) when the norm of the gradient is bounded. To estimate the gradient up to δ error (in $L2$ norm), it is required to have an evaluation oracle with error at most $\mathcal{O}(\delta^2/d^2)$ which might not be feasible if the noisy evaluation oracle is stochastic.

Our algorithms work in the stochastic setting where we prove that we can create an accurate evaluation oracle under Assumptions 3.4.3 and 3.4.4. Furthermore, the function f needs to be smooth; however, under Assumptions 3.4.3 and 3.4.10 the smoothness constant is not bounded and this might cause unbounded error. We propose a robust version of Algorithm 5 so that we can still estimate the gradient accurately (See the step-by-step description in Section 3.4.4). We also note that the oracle O_F is known as the phase oracle. Our oracle (Equation (3.94)) can be converted to phase oracle efficiently.

3.4.2 Overview of Multi-Level Monte Carlo Algorithm

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

Algorithm 6 `UnbiasedStochasticGradient`

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

Lemma 3.4.2. *Given access to an algorithm* `BiasedStochasticGradient` *that outputs a random vector* $\mathbf{v}$ *such that* $\mathbb{E}\|\mathbf{v} - \nabla f(\mathbf{x})\| \leq \hat{\sigma}^2$ *with a cost* $\tilde{\mathcal{O}}\left(\frac{C}{\hat{\sigma}}\right)$*, the algorithm* `UnbiasedStochasticGradient` *outputs a vector* $\mathbf{g}$ *such that* $\mathbb{E}[\mathbf{g}] = \nabla f(\mathbf{x})$ *and* $\mathbb{E}\|\mathbf{g} - \nabla f(\mathbf{x})\| \leq \hat{\sigma}^2$ *with an expected cost* $\tilde{\mathcal{O}}\left(\frac{C}{\hat{\sigma}}\right)$*.*

Proof. We repeat the proof in [SZ23].

$$\mathbf{g} = \mathbf{g}_0 + 2^J(\mathbf{g}_J - \mathbf{g}_{J-1}), \qquad J \sim \text{Geom}\left(\frac{1}{2}\right) \in \mathbb{N}. \tag{3.96}$$

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

$$\mathbb{E}[\mathbf{g}] = \mathbb{E}[\mathbf{g}_0] + \sum_{j=1}^{\infty} \Pr(J = j)2^j(\mathbb{E}[\mathbf{g}_j] - \mathbb{E}[\mathbf{g}_{j-1}]) = \mathbb{E}[\mathbf{g}_\infty] = \nabla f(\mathbf{x}). \tag{3.97}$$

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

$$\mathbb{E}\|\mathbf{g} - \nabla f(\mathbf{x})\|^2 \leq 2\mathbb{E}\|\mathbf{g} - \mathbf{g}_0\|^2 + 2\mathbb{E}\|\mathbf{g}_0 - \nabla f(\mathbf{x})\|^2 \tag{3.98}$$

where

$$\mathbb{E}\|\mathbf{g} - \mathbf{g}_0\|^2 = \sum_{j=1}^{\infty} \Pr(J = j)2^{2j}\mathbb{E}\|\mathbf{g}_j - \mathbf{g}_{j-1}\|^2 = \sum_{j=1}^{\infty} 2^j\mathbb{E}\|\mathbf{g}_j - \mathbf{g}_{j-1}\|^2, \tag{3.99}$$

and for each j we have

$$\mathbb{E}\|\mathbf{g}_j - \mathbf{g}_{j-1}\|^2 \leq 2\mathbb{E}\|\mathbf{g}_j - \nabla f(\mathbf{x})\|^2 + 2\mathbb{E}\|\mathbf{g}_{j-1} - \nabla f(\mathbf{x})\|^2. \tag{3.100}$$

By assumption on `BiasedStochasticGradient`,

$$\mathbb{E}\|\mathbf{g}_j - \nabla f(\mathbf{x})\|^2 \le \frac{\hat{\sigma}^2}{100 \cdot 2^{3j/2}}, \quad \forall j \ge 0, \tag{3.101}$$

which leads to

$$\mathbb{E}\|\mathbf{g}_j - \mathbf{g}_{j-1}\|^2 \le \frac{\hat{\sigma}^2}{50 \cdot 2^{3(j-1)/2}} + \frac{\hat{\sigma}^2}{50 \cdot 2^{3j/2}} \le \frac{\hat{\sigma}^2}{10 \cdot 2^{3j/2}}, \tag{3.102}$$

and

$$\mathbb{E}\|\mathbf{g} - \mathbf{g}_0\|^2 = \frac{\hat{\sigma}^2}{10} \sum_{j=1}^{\infty} \frac{1}{2^{j/2}} \le \frac{1}{3}\hat{\sigma}^2 . \tag{3.103}$$

Hence,

$$\mathbb{E}\|\mathbf{g} - \nabla f(\mathbf{x})\|^2 \le 2\mathbb{E}\|\mathbf{g} - \mathbf{g}_0\|^2 + 2\mathbb{E}\|\mathbf{g}_0 - \nabla f(\mathbf{x})\|^2 \le \hat{\sigma}^2, \tag{3.104}$$

Moreover, the expected cost is

$$\tilde{\mathcal{O}}\left(\frac{C}{\hat{\sigma}}\right) \cdot \left(1 + \sum_{j=1}^{\infty} \Pr\{J = j\} \cdot \left(2^{3j/4} + 2^{3(j-1)/4}\right)\right) = \tilde{\mathcal{O}}\left(\frac{C}{\hat{\sigma}}\right). \tag{3.105}$$

□

3.4.3 Gradient Estimation for Smooth Potentials

Assumption 3.4.3 (Bounded Noise)**.** For any $\mathbf{x} \in \mathbb{R}^d$, the stochastic zeroth-order oracle outputs an estimator $f(\mathbf{x};\xi)$ of $f(\mathbf{x})$ such that $\mathbb{E}[f(\mathbf{x};\xi)] = f(\mathbf{x})$, $\mathbb{E}[\nabla f(\mathbf{x};\xi)] = \nabla f(\mathbf{x})$, and $\mathbb{E}\|\nabla f(\mathbf{x};\xi) - \nabla f(\mathbf{x})\|^2 \le \sigma^2$.

Assumption 3.4.4 (Smoothness)**.** The potential function $f : \mathbb{R}^d \to \mathbb{R}$ has L-Lipschitz gradients. Specifically, it holds that

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

These assumptions are standard in the zeroth-order sampling and optimization literature [RSBG21, BG22]. We note that Assumption 3.4.3 is broader than an additive noise model, as it accommodates models with multiplicative noise. For example, suppose that $f : \mathcal{B}_d(0, R) \mapsto \mathbb{R}$ is an L smooth differentiable function, and that the stochastic

components are of the form $f(\mathbf{x};\xi) = \xi f(\mathbf{x})$, where $\mathbb{E}[\xi] = 1$ and $\mathbb{E}[\xi^2] \leq \frac{\sigma^2}{4L^2R^2}$. In this case, Assumption 3.4.3 is satisfied. Suppose that the function f can be queried with the same randomness at two different points, that is, we can query $f(\mathbf{x};\xi_i)$ and $f(\mathbf{y};\xi_i)$ simultaneously [2]. Classically, the gradient in this two-point setting can be estimated using the Gaussian smoothing technique. This involves sampling random directions from the extended space around the target point and performing two-point evaluations to approximate the gradient. Specifically, the gradient can be approximated as:

$$\mathbf{g}_{\nu,b}(\mathbf{x}) = \frac{1}{b}\sum_{i=1}^{b} \frac{f(\mathbf{x}+\nu\mathbf{u}_i;\xi_i) - f(\mathbf{x};\xi_i)}{\nu}\mathbf{u}_i, \tag{3.106}$$

where $\mathbf{u}_i \sim \mathcal{N}(0, I_d)$ are independent and identically distributed random vectors. [BG22] showed that for any $\mathbf{x} \in \mathbb{R}^d$, the estimator $\mathbf{g}_{\nu,b}$ satisfies $\mathbb{E}\|\mathbf{g}_{\nu,b}(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \frac{4(d+5)(\|\nabla f(\mathbf{x})\|^2+\sigma^2)}{b} + \frac{3\nu^2L^2(d+3)^3}{2}$. Although the squared norm of the gradient on the right-hand side is unbounded, it is typically of order $\tilde{\mathcal{O}}(d)$ in expectation throughout the trajectory of LMC (See Equation (3.48)). Consequently, this method requires $b = \mathcal{O}(d^2/\epsilon^2)$ function evaluations to achieve an ϵ-accurate gradient estimate in the L_2 norm. We show that we can estimate the gradient of f up to ϵ accurate in L_2 norm using quantum gradient estimation techniques quadratically faster using Jordan's gradient estimation algorithm [Jor05], which we implement through a proper phase oracle (See Proposition 3.4.5) using the stochastic evaluation oracle for f.

Although Jordan's algorithm is appealing as it only uses a constant number of evaluations to estimate the gradient (See Lemma 3.4.1), its practical use cases are limited as it requires the function evaluations to be very accurate. In particular, to be able to use the quantum gradient estimation algorithm, we need to be able to implement the phase oracle (line 4 in Algorithm 5) $O_F|\mathbf{x}\rangle \to \frac{1}{N^{d/2}}\sum_{\mathbf{x}\in G_d^l} e^{2\pi i F(\mathbf{x})}|\mathbf{x}\rangle$ where $F(\mathbf{x}) = \frac{N}{2Ml}[f(\mathbf{x}_0 + \frac{l}{N}(\mathbf{x} - \frac{N}{2})) - f(\mathbf{x}_0)]$. We show that even with the stochastic evaluation oracle, this oracle can be implemented accurately using additional techniques under Assumption 3.4.3. We prove in Proposition 3.4.5 that given the sampling oracle O_X, a sufficiently accurate phase oracle that maps $|0\rangle \to e^{t\mathbb{E}[X]}|0\rangle$ for $t \geq 0$ can be implemented using $\tilde{O}(\sigma t)$ stochastic evaluation queries. Next, we incorporate this oracle into Jordan's algorithm. Since Jordan's algorithm is biased and succeeds with high probability, we postprocess the output using the Multi-Level Monte Carlo technique (Algorithm 6) to make the output smooth and unbiased. The preliminaries for the MLMC algorithm can be found in Section 3.4.2.

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

Proposition 3.4.5. *Let $X \in \mathbb{R}$ be a random variable such that $\mathbb{E}\|X - \mathbb{E}[X]\|^2 \leq \sigma^2$. Given two reals $t \geq 0$ and $\epsilon \in (0,1)$, then there is a unitary operator $P^X_{t,\epsilon} : |0\rangle |0\rangle \mapsto |\phi_X\rangle |0\rangle$ acting on $\mathcal{H}_X \otimes \mathcal{H}_{aux}$ that can be implemented using $\tilde{\mathcal{O}}(t\sigma \log(1/\epsilon))$ quantum experiments and binary oracle queries to X such that*

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

with probability at least $8/9$.

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

$$\Pr[|s - \mathbb{E}[X]]| \geq 3\sigma] \leq \frac{1}{9}. \tag{3.107}$$

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

$$\mathbb{E}[Y_{0,\infty}] = \mathbb{E}[Y_{0,1}] + \sum_{k=1}^{K} 2^k \mathbb{E}\left[\frac{Y_{2^{k-1},2^k}}{2^k}\right] + \mathbb{E}[Y_{2^K,\infty}]. \tag{3.108}$$

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

$$P^{Y_{0,\infty}}_{t,\epsilon/2} = P^{Y_{0,1}}_{t,\epsilon/6} \left(\prod_{k=1}^{K} P^{Y_{2^{k-1},2^k}}_{t,\epsilon/6K}\right) P^{Y_{2^K,\infty}}_{t,\epsilon/6}. \tag{3.109}$$

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

$$\left||0\rangle - e^{it\mathbb{E}[Y_{2^K,\infty}]} |0\rangle\right| \leq t\mathbb{E}[Y_{2^K,\infty}] \leq \frac{\epsilon}{6}. \tag{3.110}$$

The last inequality holds because:

$$\mathbb{E}[Y_{2^K,\infty}] = \sum_{Y \geq 2^K} \Pr(Y)Y \leq \sum_{Y} \frac{1}{2^K} \Pr(Y)Y^2 = \frac{\mathbb{E}\|Y\|^2}{2^K} \tag{3.111}$$

$$\leq \frac{2\mathbb{E}\|X - \mathbb{E}[X]\|^2 + 2\|s - \mathbb{E}[X]\|^2}{2^K} \tag{3.112}$$

$$\leq \frac{20\sigma^2}{2^K} = \frac{\epsilon}{6t}, \tag{3.113}$$

where the inequality in the second line follows from the definition of Y and Young's inequality. Since $X_{0,1}$ is bounded between 0 and 1, we can implement $P_{t,\epsilon/6}^{Y_{0,1}}$ using $\tilde{\mathcal{O}}(1)$ queries to X via the binary-to-phase conversion algorithm (Lemma 2.12 in [CHJ22]). We need to show how to implement $P_{t,\epsilon/6K}^{Y_{a,b}}$ when $b > 1$. We start by defining the unitary operator:

$$V_{a,b} : |0\rangle |0\rangle \mapsto \sum_Y \sqrt{\Pr(Y)} |Y_{a,b}/b\rangle |0\rangle \tag{3.114}$$

$$\mapsto \sum_Y \sqrt{\Pr(Y)} |Y_{a,b}/b\rangle \left(\sqrt{Y_{a,b}/b} |0\rangle + \sqrt{1 - Y_{a,b}/b} |1\rangle\right) \tag{3.115}$$

$$= \sqrt{\mathbb{E}[Y_{a,b}/b]} |\psi_0\rangle |0\rangle + \sqrt{1 - \mathbb{E}[Y_{a,b}/b]} |\psi_1\rangle |1\rangle , \tag{3.116}$$

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

$$\mathbb{E}[Y_{a,b}/b] \leq \frac{1}{b} \sum_{a \leq Y \leq b} \Pr(Y) Y \leq \frac{1}{ab} \sum_{a \leq Y \leq b} \Pr(Y) Y^2 \tag{3.117}$$

$$= \frac{1}{ab} \mathbb{E}\|Y\|^2 \leq \frac{\sigma^2}{ab}, \tag{3.118}$$

we can apply the linear amplitude amplification algorithm (see [CHJ22, Proposition 2.10]) to implement the operator:

$$W_{a,b} : |0\rangle |0\rangle \mapsto \sqrt{p_{a,b}} |\psi_0\rangle |0\rangle + \sqrt{1 - p_{a,b}} |\psi_1\rangle |1\rangle , \tag{3.119}$$

such that,

$$\left| \sqrt{p_{a,b}} - \sqrt{\frac{\mathbb{E}[Y_{a,b}/b]}{\sigma^2/(ab)}} \right| \leq \frac{\epsilon}{24Ktb} \tag{3.120}$$

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

$$\|W_{a,b} |0\rangle - e^{it\mathbb{E}[Y_{a,b}]} |0\rangle \| = \|e^{it' p_{a,b}} |0\rangle - e^{it\mathbb{E}[Y_{a,b}]} |0\rangle \| \tag{3.121}$$

$$\leq t' \left| p_{a,b} - \frac{\mathbb{E}[Y_{a,b}/b]}{\sigma^2/(ab)} \right| + \frac{\epsilon}{12K} \tag{3.122}$$

$$\leq 2t' \left| \sqrt{p_{a,b}} - \sqrt{\frac{\mathbb{E}[Y_{a,b}/b]}{\sigma^2/(ab)}} \right| + \frac{\epsilon}{12K} \tag{3.123}$$

$$\leq \frac{\epsilon}{6K}. \tag{3.124}$$

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

$$P^{X}_{t,\epsilon} = P^{X_{0,\infty}}_{t,\epsilon/2} P^{X_{-\infty,0}}_{t,\epsilon/2}. \tag{3.125}$$

This concludes the proof. □

This phase oracle is similar to the oracle implemented in [CHJ22]; however, their algorithm requires $\|X\| \leq 1$ whereas $\|X\|$ might be unbounded in our case. Hence, Proposition 3.4.5 generalizes the phase oracle to the unbounded random variables by constructing a sequence of unitaries for different levels of truncation of the random variable X.

Lemma 3.4.6. *Suppose we run Algorithm 5 with the phase oracle in Proposition 3.4.5 with evaluation accuracy $\epsilon' = \frac{\epsilon^2}{d^2\beta}$ to $f(\mathbf{x}, \xi)$. Let $\tilde{\mathbf{g}}$ denote the output. Then, under Assumptions 3.4.3 and 3.4.4,*

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

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

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

$$O_F \left|\mathbf{x}\right\rangle \mapsto e^{i\mathbb{E}_\xi[F(\mathbf{x},\xi)]} \left|\mathbf{x}\right\rangle, \tag{3.126}$$

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

$$\mathbb{E}\|F(\mathbf{x}; \xi) - \mathbb{E}[F(\mathbf{x}; \xi)]\|^2 = \mathbb{E}\left\| \int_0^1 \langle \nabla f(\mathbf{x} + t\mathbf{y}; \xi) - \nabla f(\mathbf{x} + t\mathbf{y}), \mathbf{y} \rangle dt \right\|^2 \tag{3.127}$$

$$\leq \|y\|^2 \int_0^1 \mathbb{E}\|\nabla f(\mathbf{x} + t\mathbf{y}; \xi) - \nabla f(\mathbf{x} + t\mathbf{y})\|^2 dt \tag{3.128}$$

$$\leq \sigma^2 l^2 d. \tag{3.129}$$

Hence, implementing $e^{i\mathbb{E}[F(\mathbf{x},\xi)]}$ takes $\tilde{\mathcal{O}}(\sigma l d^{1/2}\frac{N}{Ll}) = \tilde{\mathcal{O}}(\frac{\sigma}{\epsilon'^{1/2}\beta^{1/2}}) = \tilde{\mathcal{O}}(\frac{\sigma d}{\epsilon})$ queries to stochastic zeroth-order oracle and succeeds with probability 8/9. Since Algorithm 5 uses $\tilde{\mathcal{O}}(1)$ queries to O_F by Lemma 3.4.1 and succeeds with probability 2/3, the total query complexity is $\tilde{\mathcal{O}}(\frac{\sigma d}{\epsilon})$ and success probability is at least 5/9 due to union bound. □

Theorem 3.4.7. *Suppose that the potential function f satisfies Assumptions 3.4.3 and 3.4.4 and further suppose that $\|\nabla f(\mathbf{x})\| \leq M$ for all $\mathbf{x}$. Then, given a real $\hat{\sigma} > 0$, there exists a quantum algorithm that outputs a random vector $\mathbf{g}$ such that*

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

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

Proof. Suppose that we run Algorithm 5 in Lemma 3.4.6 T times with target accuracy $\frac{\hat{\sigma}}{2}$, then compute the median (coordinate-wise) of these outputs. If the result has norm smaller than M, we output this vector. Otherwise, we output all 0 vector. Let $\mathbf{v}$ be the output of this algorithm. Since the algorithm in Lemma 3.4.6 outputs a vector $\tilde{\mathbf{g}}$ such that $\|\tilde{\mathbf{g}} - \nabla f(\mathbf{x})\| \leq \frac{\hat{\sigma}}{2}$ with high probability, then by Chernoff bound and union bound over each dimension, at least $\frac{T}{2}$ of the outputs satisfy $\|\tilde{\mathbf{g}} - \nabla f(\mathbf{x})\| \leq \hat{\sigma}$ with probability at least $1 - 2\exp(-T^2/24)$. Since the norm of the gradient is M, when the condition fails, the error is $\|\tilde{\mathbf{g}} - \nabla f(\mathbf{x})\| \leq M$. Then in expectation,

$$\mathbb{E}\|\mathbf{v} - \nabla f(\mathbf{x})\|^2 \leq \frac{\hat{\sigma}^2}{4} + 2\exp\Big(-T^2/24\Big)M^2. \tag{3.130}$$

Setting $T^2 = 24\log\Big(\frac{8M^2}{3\hat{\sigma}^2}\Big)$ gives $\mathbb{E}\|\mathbf{v} - \nabla f(\mathbf{x})\|^2 \leq \hat{\sigma}^2$. Hence, the overhead to Lemma 3.4.6 to make the output smooth is at most logarithmic. Finally, we can use this algorithm as the biased stochastic gradient estimator in Algorithm 6 and obtain an unbiased estimator $\mathbf{g}$. □

Remark 3.4.8. One can show that the norm of the gradient is bounded by a function of problem parameters throughout the trajectory of HMC or LMC due to smoothness. Since the dependency on M is logarithmic, we do not give an explicit bound on M.

Remark 3.4.9. Suppose that f_ξ is a non-smooth but locally L-Lipschitz function around the grid G_d^l. We define $f_v(\mathbf{x}) = \mathbb{E}_{\xi\in\Xi,\mathbf{u}\sim\mathcal{B}(0,1)}[f(\mathbf{x} + v\mathbf{u};\xi)]$. Then, let $\mathbf{y} \in G_d^l$, $\mathbb{E}\|\nabla f(\mathbf{y} + v\mathbf{u}) - \nabla f_v(\mathbf{y})\|^2 \leq 4L^2$. It is known that f_v is a smooth function with smoothness

parameter $O(Ld^{1/2}v^{-1})$. Hence, by Theorem 3.4.7 our algorithm outputs an unbiased estimator $\mathbf{g}$ such that $\mathbb{E}[\mathbf{g}] = \nabla f_v(\mathbf{x})$ and $\mathbb{E}\|\mathbf{g} - \nabla f_v(\mathbf{x})\|^2 \leq \hat{\sigma}^2$ using $\tilde{\mathcal{O}}(\frac{Ld}{\hat{\sigma}})$ queries to f_ξ. This result has recently been established in [LGHL24], and it is a special case of Theorem 3.4.7.

3.4.4 Gradient Estimation under Additional Smoothness Assumption

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

Assumption 3.4.10 (Lipschitz Stochastic Gradients)**.** The stochastic component $f(\cdot;\xi) : \mathbb{R}^d \to \mathbb{R}$ has $L(\xi)$-Lipschitz gradients for any $\xi \in \Xi$. Specifically, it holds that

$$\|\nabla f(\mathbf{x};\xi) - \nabla f(\mathbf{y};\xi)\| \leq L(\xi)\|\mathbf{x} - \mathbf{y}\|, \tag{3.131}$$

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

Assumption 3.4.10 is weaker than the assumption that each stochastic function f_ξ has L-Lipschitz gradients and it is straightforward to show that Assumption 3.4.10 implies that f has Lipschitz gradients.

As opposed to implementing an accurate phase oracle, one can estimate the gradient $\nabla f(\mathbf{x};\xi)$ and then use the quantum mean estimation algorithm to compute $\nabla f(\mathbf{x})$. However, Assumption 3.4.10 implies that f_ξ might not be a smooth function (even if f is smooth), which is the requirement in Lemma 3.4.1. Hence, Jordan's algorithm might fail to compute the gradient for ∇f_ξ with small probability no matter how large we set β in Algorithm 5. To address this, we propose a robust version of the quantum mean estimation algorithm such that we can still estimate the mean of a random variable X even when X is corrupted with small probability, which corresponds to the case Jordan's algorithm fails. Our final algorithm achieves $\tilde{\mathcal{O}}(d^{1/2}\epsilon^{-1})$ query complexity to estimate the gradient up to ϵ error.

We give a step-by-step description of Algorithm 7. The algorithm begins with the application of the oracle O_ξ, which creates the following superposition state:

$$|\psi_1\rangle = O_\xi |\mathbf{x}\rangle |0\rangle = \sum_{\xi\in\Xi} \sqrt{\Pr(\xi)}\, |\mathbf{x}\rangle |\xi\rangle . \tag{3.132}$$

We then construct a superposition over d-dimensional grid points, G_d^l, centered around

Algorithm 7 `QuantumStochasticGradient`

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

$\mathbf{x}$ with side length l, using the oracle $O_{G_d^l}$:

$$|\psi_2\rangle = O_{G_d^l} |0\rangle |\psi_1\rangle = \frac{1}{\sqrt{N^d}} \sum_{\xi\in\Xi} \sum_{\mathbf{y}\in G_d^l} \sqrt{\Pr(\xi)} |\mathbf{y}\rangle |\mathbf{x}\rangle |\xi\rangle . \tag{3.133}$$

Next, the evaluation oracle O_f is applied, resulting in the state:

$$|\psi_3\rangle = O_f |\psi_2\rangle = \frac{1}{\sqrt{N^d}} \sum_{\xi\in\Xi} \sum_{\mathbf{y}\in G_d^l} \sqrt{\Pr(\xi)} e^{2\pi i \frac{N}{2Ml}[f(\mathbf{x}+\frac{l}{N}(\mathbf{y}-N/2);\xi)-f(\mathbf{x};\xi)]} |\mathbf{y}\rangle |\mathbf{x}\rangle |\xi\rangle . \tag{3.134}$$

Note that this oracle is different than the oracle Proposition 3.4.5. Here, we have superposition over the randomness whereas Proposition 3.4.5 implements the expectation over the randomness to the phase.

Applying the inverse QFT and scaling the resulting vector by M/N, we estimate a vector $\mathbf{g}(\mathbf{x};\xi)$ for each ξ:

$$|\psi_4\rangle = \mathrm{QFT}^{-1} |\psi_3\rangle = \sum_{\xi\in\Xi} \sqrt{\Pr(\xi)} |\mathbf{g}(\mathbf{x};\xi)\rangle |\mathbf{x}\rangle |\xi\rangle + |\mathcal{X}_1\rangle , \tag{3.135}$$

where $|\mathcal{X}_1\rangle$ represents a garbage state with a small amplitude arising from the failure probability in gradient estimation. The scaling by M/N compensates for the scale factor introduced in the phase. When the deviation from linearity is quadratic and sufficient precision is chosen by N and l, as shown in Lemma 3.4.1, $\mathbf{g}(\mathbf{x};\xi)$ is an accurate estimate for $\nabla f(\mathbf{x};\xi)$. However, the small deviation condition might not hold under Assumption 3.4.10 for a subset of Ξ.

Next, we sample from $|\psi_4\rangle$ and measure the first register, obtaining an output $\mathbf{s}$. Using the previously defined steps, we recreate $|\psi_4\rangle$. At this point, we define a corrected

gradient estimate:

$$\tilde{\mathbf{g}}(\mathbf{x},\xi) = \begin{cases} \mathbf{g}(\mathbf{x},\xi) & \text{if } \|\mathbf{g}(\mathbf{x},\xi) - \mathbf{s}\| \leq D, \\ \mathbf{s} & \text{otherwise.} \end{cases} \tag{3.136}$$

Next, we construct the following quantum state by applying a controlled operation and undoing the ancillary registers:

$$|\psi_5\rangle = U_x |\psi_4\rangle = \sum_{\xi\in\Xi} \sqrt{\Pr(\xi)}\, |\tilde{\mathbf{g}}(\mathbf{x};\xi)\rangle\, |\mathbf{x}\rangle\, |\xi\rangle + |\mathcal{X}_2\rangle\,, \tag{3.137}$$

where $|\mathcal{X}_2\rangle$ is another garbage state with a small amplitude.

Finally, we estimate the mean of the first register to compute $\boldsymbol{v}$, which is output as the gradient estimate. Note that $\mathbf{g}(\mathbf{x};\xi)$ after the inverse Fourier transform may not be ϵ-accurate for all $f(\mathbf{x},\xi)$. In particular, for some ξ, the error in the gradient could be unbounded because the deviation from linearity may not be small for every f_ξ. To address this, the subsequent step replaces such erroneous estimates with the mediocre estimate $\mathbf{s}$, ensuring robustness.

Lemma 3.4.11. *Under Assumptions 3.4.3 and 3.4.10, Algorithm 7 returns a vector $\boldsymbol{v}$ such that*

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

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

Proof. As the algorithm essentially computes the expectation of $\mathbb{E}_\xi[\tilde{\mathbf{g}}(\mathbf{x},\xi)]$, we need to prove that $\mathbb{E}_\xi[\tilde{\mathbf{g}}(\mathbf{x},\xi)]$ is close to $\nabla f(\mathbf{x})$. We consider the case that Algorithm 5 returns $\epsilon/8$ accurate estimate whenever the function f behaves like β smooth inside the grid points. Furthermore, we consider the case $\|\mathbf{s} - \nabla f(\mathbf{x})\| \leq 2\sigma$. Both conditions are in fact achieved with high probability. Let $S \subseteq \Xi$ be a set such that the output of quantum gradient estimation (Algorithm 5) $\mathbf{g}$ satisfies $\|\mathbf{g} - \nabla f(\mathbf{x},\xi)\| \leq \frac{\epsilon}{8}$. Let $S' = \Xi - S$. We can consider the difference in L_2 norm separately for S and S' using triangular inequality.

$$\|\mathbb{E}_\xi \tilde{\mathbf{g}}(\mathbf{x},\xi) - \nabla f(\mathbf{x})\| \leq \|\mathbb{E}_S(\tilde{\mathbf{g}}(\mathbf{x},\xi) - \nabla f(\mathbf{x};\xi))\| + \|\mathbb{E}_{S'}(\tilde{\mathbf{g}}(\mathbf{x},\xi) - \nabla f(\mathbf{x};\xi))\|. \tag{3.138}$$

We first analyze the first term. The contribution to the first term is either due to gradient estimation error $\frac{\epsilon}{8}$ or it is due to the fact that $\mathbf{g}$ is replaced by $\mathbf{s}$ because $\|\mathbf{g} - \mathbf{s}\| > D$.

Suppose that $S_1 = \{\xi \in \Xi : \|\mathbf{g}(\mathbf{x};\xi) - s\| \leq D\}$ and $S_2 = S - S_1$. We can separate the error further for both cases using triangular inequality.

$$
\begin{aligned}
&\|\mathbb{E}_S(\tilde{\mathbf{g}}(\mathbf{x},\xi) - \nabla f(\mathbf{x};\xi))\| \\
&\leq \mathbb{E}_{\xi\in S_1}\|(\mathbf{g}(\mathbf{x},\xi) - \nabla f(\mathbf{x},\xi))\| + \mathbb{E}_{\xi\in S_2}\|(\mathbf{s} - \nabla f(\mathbf{x},\xi))\| && (3.139)\\
&\leq \mathbb{E}_{\xi\in S_1}\|(\mathbf{g}(\mathbf{x},\xi) - \nabla f(\mathbf{x},\xi))\| + \mathbb{E}_{\xi\in S_2}\|(\mathbf{s} - \mathbf{g}(\mathbf{x};\xi)\| && (3.140)\\
&+ \mathbb{E}_{\xi\in S_2}\|(\mathbf{g}(\mathbf{x};\xi) - \nabla f(\mathbf{x},\xi))\| && (3.141)\\
&\leq \frac{\epsilon}{8} + \frac{\mathbb{E}\|\mathbf{s} - \nabla f(\mathbf{x},\xi))\|^2}{D} + \frac{\epsilon}{8}. && (3.142)
\end{aligned}
$$

The first inequality is due to the fact that for any $\xi \in S_2$, Algorithm 7 replaces $\mathbf{g}$ by $\mathbf{s}$. The last inequality follows from the fact that $\|\mathbf{g}(\mathbf{x};\xi) - \nabla f(\mathbf{x};\xi)\| \leq \frac{\epsilon}{8}$ for any $\xi \in S$ and $\mathbb{E}_{\xi\in S_2}\|\mathbf{s} - \mathbf{g}(\mathbf{x};\xi)\| \leq \frac{\mathbb{E}\|\mathbf{s}-\mathbf{g}(\mathbf{x};\xi)\|^2}{D}$ since for any $\xi \in S_2$ we have $\|\mathbf{g}(\mathbf{x};\xi) - \mathbf{s}\| > D$. As $\|\mathbf{s} - \nabla f(\mathbf{x})\| \leq 2\sigma$,

$$
\begin{aligned}
\mathbb{E}\|\mathbf{s} - \mathbf{g}(\mathbf{x};\xi)\|^2 &\leq 2\mathbb{E}\|\mathbf{s} - \nabla f(\mathbf{x};\xi)\|^2 + 2\mathbb{E}\|\nabla f(\mathbf{x};\xi) - \mathbf{g}(\mathbf{x};\xi)\|^2 && (3.143)\\
&\leq 10\sigma^2. && (3.144)
\end{aligned}
$$

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

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

$$
\mathbb{E}_{S'}\|(\tilde{\mathbf{g}}(\mathbf{x},\xi) - \nabla f(\mathbf{x}))\| \leq \frac{L}{\beta}\max(D, 2\sigma) \leq \frac{\epsilon}{4} \tag{3.145}
$$

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

$$
\begin{aligned}
&\mathbb{E}_\xi\|\tilde{\mathbf{g}}(\mathbf{x},\xi) - \mathbb{E}[\tilde{\mathbf{g}}(\mathbf{x},\xi)]\|^2 \\
&\leq 2\mathbb{E}\left\|\tilde{\mathbf{g}}(\mathbf{x},\xi) - \nabla f(x)\right\|^2 + 2\left\|\mathbb{E}[\tilde{\mathbf{g}}(\mathbf{x},\xi)] - \nabla f(x)\right\|^2 && (3.146)\\
&\leq 2\mathbb{E}_{S'}\left\|\tilde{\mathbf{g}}(\mathbf{x},\xi) - \nabla f(x)\right\|^2 + 2\mathbb{E}_S\left\|\tilde{\mathbf{g}}(\mathbf{x},\xi) - \nabla f(x)\right\|^2 + 2\epsilon^2 && (3.147)\\
&\leq \frac{2Ld}{\beta}\max(D^2, 4\sigma^2) + 2\mathbb{E}\left\|\nabla f(\mathbf{x};\xi) - \nabla f(\mathbf{x})\right\|^2 + 2\mathbb{E}\left\|\mathbf{s} - \nabla f(\mathbf{x})\right\|^2 + 3\epsilon^2 && (3.148)
\end{aligned}
$$

$$= \mathcal{O}(\sigma^2). \tag{3.149}$$

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

Next, we postprocess the output of Algorithm 7 to obtain a smooth and unbiased estimate.

Theorem 3.4.12 (Smooth Gradient)**.** *Suppose that the potential function f satisfies Assumptions 3.4.3 and 3.4.10 and further suppose that $\|\nabla f(\mathbf{x})\| \leq M$ for all $\mathbf{x}$. Then, given a real $\hat{\sigma} > 0$, there exists a quantum algorithm that outputs a random vector $\mathbf{g}$ such that*

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

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

Proof. Suppose that we run Algorithm 7 T times with target accuracy $\frac{\hat{\sigma}}{2}$, then compute the median (coordinate-wise) of these outputs. If the result has norm smaller than M, we output this vector. Otherwise, we output all 0 vector. Let $\mathbf{v}$ be the output of this algorithm. Since Algorithm 7 outputs a gradient $\mathbf{v}$ such that $\|\mathbf{v} - \nabla f(\mathbf{x})\| \leq \hat{\sigma}/2$ with high probability (say 2/3), then by Chernoff bound and union bound over each dimension, at least $\frac{T}{2}$ of the outputs satisfy $\|\mathbf{v} - \nabla f(\mathbf{x})\| \leq \hat{\sigma}$ with probability at least $1 - 2\exp(-T^2/24)$. Since the norm of the gradient is M, when the condition fails the error is $\|\mathbf{v} - \nabla f(\mathbf{x})\| \leq M$. Then in expectation,

$$\mathbb{E}\|\mathbf{v} - \nabla f(\mathbf{x})\|^2 \leq \frac{\hat{\sigma}^2}{4} + 2\exp\Big(-T^2/24\Big)M^2. \tag{3.150}$$

Setting $T^2 = 24\log\Big(\frac{8M^2}{3\hat{\sigma}^2}\Big)$ gives $\mathbb{E}\|\mathbf{v} - \nabla f(\mathbf{x})\|^2 \leq \hat{\sigma}^2$. Hence, the overhead is at most logarithmic. Finally, we run Algorithm 6 to obtain an unbiased estimator $\mathbf{g}$. □

3.5 Quantum Speedups for Sampling via Evaluation Oracle

We apply our quantum gradient estimation algorithm to establish the convergence of both HMC and LMC in strongly convex and LSI settings, respectively. In particular, at each

iteration, we use the inexact gradients computed by our quantum gradient estimation algorithms introduced in previous sections.

3.5.1 Zeroth Order Sampling under Strong Convexity

Theorem 3.5.1 (Main Theorem for `QZ-HMC`). *Let μ_k be the distribution of $\mathbf{x}_k$ in* `QZ-HMC` *algorithm. Suppose that f satisfies Assumption 3.3.1. Given that the initial point $\mathbf{x}_0$ satisfies $\|\mathbf{x}_0 - \arg\min_{\mathbf{x}} f(\mathbf{x})\| \leq \frac{d}{\mu}$, if we set the step size $\eta = \mathcal{O}\left(\frac{\epsilon}{d^{1/2}\kappa^{3/2}}\right)$, $S = \tilde{\mathcal{O}}\left(\frac{Ld^{1/2}\kappa^{3/2}}{\epsilon}\right)$, $T = \tilde{\mathcal{O}}(1)$, and $\hat{\sigma}^2 = \mathcal{O}\left(\frac{L^{3/2}d^{1/2}\epsilon}{\kappa^{3/2}}\right)$, we have*

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

In addition, under Assumptions 3.4.3 and 3.4.4, the query complexity to the stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^{5/4}\sigma}{\epsilon^{3/2}}\right)$ or under Assumptions 3.4.3 and 3.4.10 the query complexity to the stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^{3/4}\sigma}{\epsilon^{3/2}}\right)$.

Proof. By Theorem 3.3.3 for $\eta = \mathcal{O}(L^{1/2}\sigma^{-2}\kappa^{-1} \wedge L^{-1/2})$ and $K = \frac{1}{4\sqrt{L}\eta}$, we have

$$\mathrm{W}_2(\mu_T, \pi) \leq (1 - (128\kappa)^{-1})^{\frac{T}{2}}(2D + 2d/\mu)^{1/2} + \Gamma_1\eta^{1/2} + \Gamma_2\eta, \tag{3.151}$$

where

$$\Gamma_1 = \mathcal{O}\left(L^{-3/2}\hat{\sigma}^2\kappa^2\right), \tag{3.152}$$

$$\Gamma_2 = \mathcal{O}\left(\kappa^3 d\right). \tag{3.153}$$

The first term in error is $\mathcal{O}(\epsilon)$ when $T = \tilde{\mathcal{O}}(\log(1/\epsilon))$. The last two terms become $\mathcal{O}\left(L^{-3/4}\hat{\sigma}\eta^{1/2} + d^{1/2}\kappa^{3/2}\eta\right)$. For $\hat{\sigma} = \mathcal{O}(L^{3/4}d^{1/4}\kappa^{-3/4}\epsilon^{1/2} \wedge \sigma)$ and $\eta = \mathcal{O}(\epsilon d^{-1/2}\kappa^{-3/2})$, the bias term becomes $\mathcal{O}(\epsilon)$. Then, under Assumptions 3.4.3 and 3.4.4, the number of calls to evaluation oracle scale as $\tilde{O}(d^{1/2}\kappa^{3/2}\epsilon^{-1} + \sigma d^{3/4}\kappa^{3/4}\epsilon^{-3/2}) = \tilde{\mathcal{O}}(d^{3/4}\kappa^{3/4}\epsilon^{-3/2})$. Similarly, under Assumptions 3.4.3 and 3.4.10 the evaluation complexity is $\tilde{\mathcal{O}}(\sigma d^{5/4}\kappa^{3/4}\epsilon^{-3/2})$. □

The closest result in the classical setting is given by [RSBG21] for Kinetic LMC algorithm which is obtained by setting the inner iterations to 1 in HMC algorithm. Their classical evaluation complexity under Assumptions 3.4.3 and 3.4.4 is $\tilde{\mathcal{O}}(d^2\sigma^2/\epsilon^2)$ for convergence in W_2 distance (Theorem 2.2 in [RSBG21]). Our algorithm uses $\tilde{\mathcal{O}}(d^{5/4}\sigma/\epsilon^{3/2})$ evaluation queries providing speedup both in d, ϵ, and σ.

3.5.2 Zeroth Order Sampling under Log-Sobolev Inequality

In this section, we consider the sampling problem under the Log-Sobolev inequality using gradients computed via stochastic evaluation oracle. We first present the main result and defer the proof to the appendix.

Theorem 3.5.2 (Main Theorem for `QZ-LMC`)**.** *Under Assumption 3.3.10, let μ_k be the distribution of $\mathbf{x}_k$ in* `QZ-LMC` *algorithm. Then, if we set the step size $\eta = \mathcal{O}\left(\frac{\epsilon\alpha}{dL^2}\right)$, $K = \tilde{\mathcal{O}}\left(\frac{dL^2 \log(\mathrm{KL}(\mu_0||\pi))}{\epsilon\alpha^2}\right)$, and $\hat{\sigma}^2 = \mathcal{O}(\alpha\epsilon)$, we have*

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

In addition, under Assumptions 3.4.3 and 3.4.4, the query complexity to the stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^2L^2\sigma}{\alpha^{5/2}\epsilon^{3/2}}\right)$, or under Assumptions 3.4.3 and 3.4.10 the query complexity to the stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^{3/2}L^2\sigma}{\alpha^{5/2}\epsilon^{3/2}}\right)$.

Proof. By Lemma 3.3.11, one-step equation can be written as

$$\mathrm{KL}(\mu_{k+1}||\pi) \leq e^{-3\alpha\eta/2}\left[\left(1 + \frac{32\eta^3L^4}{\alpha}\right)\mathrm{KL}(\mu_k||\pi) + 6\eta\hat{\sigma}^2 + 16\eta^2dL^2\right] \tag{3.154}$$

$$\leq e^{-\alpha\eta}\,\mathrm{KL}(\mu_k||\pi) + 6\eta\hat{\sigma}^2 + 16\eta^2dL^2. \tag{3.155}$$

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

$$\mathrm{KL}(\mu_k||\pi) \leq e^{-\alpha\eta k}\,\mathrm{KL}(\mu_0||\pi) + \frac{6\eta\hat{\sigma}^2 + 16\eta^2dL^2}{1 - e^{-\alpha\eta}} \tag{3.156}$$

$$\leq e^{-\alpha\eta k}\,\mathrm{KL}(\mu_0||\pi) + \frac{8\hat{\sigma}^2 + 32\eta dL^2}{\alpha} \tag{3.157}$$

$$\leq e^{-\alpha\eta k}\,\mathrm{KL}(\mu_0||\pi) + \frac{8\hat{\sigma}^2 + 32\eta dL^2}{\alpha}. \tag{3.158}$$

The second inequality is due to the fact that for $\eta \leq \frac{\alpha}{8L^2}$, $1 - e^{-\alpha\eta} \geq \frac{3}{4}\alpha\eta$ when $\alpha\eta \leq \frac{1}{4}$. We set $\eta \leq \frac{\epsilon\alpha}{128dL^2}$ and $\hat{\sigma}^2 \leq \frac{\alpha\epsilon}{32}$ and $k \geq \frac{1}{\alpha\eta}\log\left(\frac{2\,\mathrm{KL}(\mu_0||\pi)}{\epsilon}\right)$ so that $\mathrm{KL}(\mu_k||\pi) \leq \epsilon$. The number of calls to the stochastic evaluation oracle under Assumptions 3.4.3 and 3.4.4 to achieve $\hat{\sigma}^2 \leq \frac{\alpha\epsilon}{32}$ at each iteration is $\mathcal{O}\left(\frac{d\sigma}{\alpha^{1/2}\epsilon^{1/2}}\right)$ by Theorem 3.4.7. Hence, the total number of calls to the stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^2L^2\sigma}{\alpha^{5/2}\epsilon^{3/2}}\right)$. Similarly, under Assumptions 3.4.3 and 3.4.10 the number of calls to stochastic evaluation at each iteration is $\mathcal{O}\left(\frac{d^{1/2}\sigma}{\alpha^{1/2}\epsilon^{1/2}}\right)$ by Theorem 3.4.12. Hence, the total number of calls to stochastic evaluation oracle is $\tilde{\mathcal{O}}\left(\frac{d^{3/2}L^2\sigma}{\alpha^{5/2}\epsilon^{3/2}}\right)$. □

Comparing to the classical results, [RSBG21] analyzed the convergence of LMC in the zeroth-order setting under Assumptions 3.4.3 and 3.4.4 and established evaluation complexity $\mathcal{O}(d^3\sigma^2/\epsilon^4)$ for convergence in W_2 distance (Theorem 3.2 in [RSBG21]). Our algorithm uses $\tilde{\mathcal{O}}(d^2\sigma/\epsilon^3)$ evaluation queries under the same assumptions.

3.6 Application in Optimization

Optimizing non-convex objectives arises frequently in machine learning, particularly in empirical risk minimization (ERM), where the goal is to minimize a loss function f that approximates the population risk F based on empirical observations. While F is sometimes assumed to be smooth and strongly convex, the empirical objective f, defined as

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \tag{3.159}$$

can lose the smoothness and convexity due to small perturbations introduced by finite sample effects. Such perturbations often result in f containing numerous local minima; therefore, traditional gradient-based methods like gradient descent or stochastic gradient descent (SGD) are prone to getting trapped in local minima, limiting their ability to find the global minimum of f. On the other hand, Langevin type algorithms are more robust to such local minima that only appear in the empirical objective caused by small perturbations. For example, [ZLC17] showed that stochastic Langevin algorithm can escape from such local minima efficiently due to the noise term that scales with $\eta^{1/2}$, whereas SGD gets trapped as the noise scales as η. Motivated by this, we investigate whether our quantum Langevin algorithms can provide a way to obtain quantum speedup for optimizing non-convex empirical objectives. To be more precise, we make the following assumptions.

Assumption 3.6.1 (Approximate-Convexity)**.** Let f be a differentiable function, we say that $f : \mathbb{R}^d \to \mathbb{R}$ is an ϵ-approximately convex function, if there exists a strongly convex function F such that for all $\mathbf{x}$,

$$|F(\mathbf{x}) - f(\mathbf{x})| \leq \frac{\epsilon}{d}. \tag{3.160}$$

Since f is usually not smooth, we only assume that f is Lipschitz continuous.

Assumption 3.6.2. For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}$, $f : \mathbb{R}^d \to \mathbb{R}$ satisfies,

$$|f(\mathbf{x}) - f(\mathbf{y})| \leq M\|\mathbf{x} - \mathbf{y}\|. \tag{3.161}$$

The goal is to find an approximate minimizer $\mathbf{x}^*$ such that $|f(\mathbf{x}^*) - \min_{\mathbf{x}} f(\mathbf{x})| \leq \epsilon$. Similar settings have been analyzed in the context of escaping from local minima both in classical [BLNR15] and quantum settings [LZ24] with access to a stochastic evaluation oracle. Since f is not Lipschitz smooth, we consider the smoothed approximation $f_v(\mathbf{x}) = \mathbb{E}_{u\sim\mathcal{B}_d(0,1)}[f(\mathbf{x} + v\mathbf{u})]$ and run the sampling algorithm using `QZ-LMC` on potential βf_v. By setting v sufficiently small and β sufficiently large, we make sure that the Gibbs distribution is concentrated around the global minimum of f. The local properties of f_v are known and given by the following proposition.

Proposition 3.6.3. *If f satisfies Assumption 3.6.2, then f_v satisfies*

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

First we notice that,

$$\mathbb{E}_{\mathbf{u}}\|\nabla f(\mathbf{x} + v\mathbf{u}) - \nabla f_v(\mathbf{x})\|^2 \leq 4M^2 \tag{3.162}$$

as $\|\nabla f(x)\| \leq M$ because of Lipschitz continuity. Hence, Assumption 3.4.3 holds with $\sigma^2 = 4M^2$ and Assumption 3.4.4 holds with $L = \frac{cM\sqrt{d}}{v}$. Therefore, using Theorem 3.3.15, we can sample from the Gibbs-Boltzmann distribution with potential f_v. Since our initial goal is to optimize f rather than to sample from the Gibbs distribution, we use the following lemma that describes a method to turn the sampling algorithm into an optimizer.

Lemma 3.6.4. *Let $\pi_v^\beta = \frac{e^{-\beta f_v(\mathbf{x})}}{\int e^{-\beta f_v(\mathbf{x})}d\mathbf{x}}$. If $\beta = \mathcal{O}(d/\epsilon)$ and $v \leq \frac{\epsilon}{Md}$, then sampling from π_v^β returns ϵ approximate optimizer for f with high probability.*

Proof. Without loss of generality, assume that $\min_{\mathbf{x}} F(\mathbf{x}) = 0$. Then, using the fact that F is convex,

$$\mathbb{E}_{\pi_v^\beta}[F(\mathbf{x})] = \frac{\int F(\mathbf{x}) \exp(-\beta f_v(\mathbf{x}))d\mathbf{x}}{\int \exp(-\beta f_v(\mathbf{x}))d\mathbf{x}} \tag{3.163}$$

$$\leq \frac{\int F(\mathbf{x}) \exp(-\beta F(\mathbf{x}))d\mathbf{x}}{\int \exp(-\beta F(\mathbf{x}))d\mathbf{x}} \exp(2v\beta M + 2\beta\epsilon/d) \tag{3.164}$$

$$\leq (d+1)/\beta \exp(2v\beta M + 2\beta\epsilon/d). \tag{3.165}$$

Therefore, $\mathbb{E}_{\pi_v^\beta}[F(\mathbf{x})] - \min_{\mathbf{x}} F(\mathbf{x}) \leq (d+1)/\beta \exp(2v\beta M + 2\beta\epsilon/d) \leq \mathcal{O}(\epsilon)$ for $v \leq \frac{\epsilon}{Md}$. Since F is uniformly close to f, the Gibbs distribution returns an ϵ optimizer for f with high probability due to Markov's inequality.

□

To be able to characterize the run-time of the algorithm, we need to characterize the Log-Sobolev constant of f_v. To achieve this, we use the following lemma by Halley-Stroock [HS87].

Lemma 3.6.5. *Let ρ be the Log-Sobolev constant of the Gibbs distribution with potential F. Then, the Log-Sobolev constant of f satisfies,*

$$\alpha \geq \rho e^{-|\sup_x (f(x)-F(x)) - \inf_x (f(x)-F(x))|}. \tag{3.166}$$

Next, we give our main result.

Theorem 3.6.6. *Suppose that f satisfies Assumptions 3.6.1 and 3.6.2. Then, there exists a quantum algorithm that returns ϵ approximate minimizer for f with high probability using $\tilde{\mathcal{O}}(\frac{d^{9/2}}{\epsilon^{3/2}})$ queries to the stochastic evaluation oracle for f.*

Proof. We consider the potential function $\beta f_v(x)$ where β is the inverse temperature parameter. By Lemma 3.6.4, sampling from $\pi_v^\beta \propto e^{-\beta f_v}$ returns $\frac{\epsilon}{2}$ approximate minimizer for f with high probability (say 0.9) for sufficiently large $\beta = \mathcal{O}(\frac{d}{\epsilon})$. Suppose that we sample from a probability distribution μ such that

$$\mathrm{TV}(\mu, \pi_v^\beta) \leq 0.1. \tag{3.167}$$

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

We need to characterize the sampling complexity from π_v^β. From Bakry Emery theorem, Log Sobolev constant ρ of βF satisfies $\rho \geq \frac{\beta\mu}{2}$ where μ is the strong convexity constant of F. Let $M' = \max(M, 1)$ and take $v = \frac{\epsilon}{2M'd}$. Then using Lemma 3.6.5, we have $\alpha \geq \frac{\beta\mu}{2} e^{-3\beta\epsilon/d} = \Omega(\frac{\mu d}{\epsilon})$ since $|f_v - F| \leq |f_v - f| + |f - F| \leq vM + \frac{\epsilon}{d} \leq \frac{3\epsilon}{2d}$. Since βf_v is a smooth function with smoothness constant $L = \mathcal{O}(\frac{\beta M\sqrt{d}}{v}) = \mathcal{O}(\frac{d^{5/2}M^2}{\epsilon^2})$ by Proposition 3.6.3, the number of calls to stochastic evaluation oracle to sample from π_v is $\tilde{\mathcal{O}}(\frac{L^2 d^2}{\alpha^{5/2}}) = \tilde{\mathcal{O}}(\frac{M^4 d^{9/2}}{\mu^{5/2}\epsilon^{3/2}})$ by Theorem 3.5.2. Hence, we can optimize f in polynomial time. □

The closest result to our setting is given by [LZ24] and their query complexity in the stochastic setting is $\tilde{\mathcal{O}}(d^5/\epsilon)$ although their assumptions are slightly different. First, they assume that the noise is sub-Gaussian and additive. Furthermore, they assume F is convex in a bounded domain but not necessarily strongly convex. Noting that these differences might possibly make the classical results loose, our algorithm seems to give a speedup in dimension dependence with a small performance drop in terms of ϵ. However, this is a known trade-off in sampling algorithms. Since their algorithm uses a reversible sampler (hit-and-run walk), their ϵ dependence only comes from the quantum mean estimation. On the other hand, our algorithm uses a non-reversible sampler (also referred to as a low accuracy sampler) which typically gives better dependency on dimension but worse on accuracy. We also note that the classical algorithm by [BLNR15] takes $\tilde{\mathcal{O}}(\frac{d^{7.5}}{\epsilon^2})$ queries to the stochastic evaluation oracle.

Upon completion of this work, we became aware of recent studies by Augustino et al. [AHF$^+$25] and Chakrabarti et al. [CHW$^+$25], which also investigate zeroth-order stochastic convex optimization under assumptions similar to those in [LZ24]. They propose algorithms with query complexities of $\widetilde{\mathcal{O}}(d^{9/2}/\epsilon^7)$ and $\widetilde{\mathcal{O}}(d^3/\epsilon^5)$, respectively. While both approaches exhibit worse dependence on ϵ compared to ours, we emphasize that the assumptions and problem settings are not identical to ours.

3.7 Conclusion

In this chapter, we developed quantum algorithms that provide provable advantages in sampling and optimization tasks by improving per-iteration cost of various samplers as opposed improving mixing time as we have in Chapter 2. By leveraging quantum variance reduction and gradient estimation techniques, we demonstrate improved query complexities over classical sampling methods for both finite-sum and zeroth-order settings. In addition, our algorithms enable faster optimization of approximately convex functions that arise in empirical risk minimization. Beyond their specific applications, the tools introduced in this work, particularly our stochastic gradient estimation methods, may serve as broadly useful primitives for a wider class of stochastic optimization problems. We leave the identification of such problems and applications of these techniques as an open problem for further research.

Chapter 4
Super-quadratic Speedup over Classical Markov Chain Search for Optimization

In Chapter 3, we transformed fast mixing into faster optimization for structured problems by repeatedly sampling from low-temperature Gibbs distributions. However, rapid mixing does not by itself ensure efficient optimization in general: Once equilibrium is reached, a classical algorithm still requires approximately $N = \mathcal{O}\left(\frac{1}{\pi(x^*)}\right)$ independent samples from the stationary distribution π to observe a global minimizer x^* with constant probability. When $\pi(x^*)$ is small, this sampling cost can dominate the total runtime and diminish the benefits of fast mixing. Quantum amplitude amplification reduces the sample complexity to $\mathcal{O}\left(\sqrt{N}\right)$, offering a quadratic improvement.

In this chapter, we go beyond that quadratic reduction and analyze generalizations of algorithms based on the short-path framework first proposed by Hastings [*Quantum* 2, 78 (2018)], which has been extended and shown by Dalzell et al. [STOC '23] to achieve super-Grover speedups for certain binary optimization problems. We demonstrate that, under some commonly satisfied technical conditions, an appropriate generalization can achieve super-quadratic speedups not only over unstructured search but also over a classical optimization algorithm that searches for the optimum by drawing samples from the stationary distribution of a Markov Chain. We employ this framework to obtain algorithms for problems including variants of Max-Bisection, Max Independent Set, the Ising Model, and the Sherrington Kirkpatrick Model, whose runtimes are asymptotically faster than those obtainable from previous short path techniques. For random regular graphs of sufficiently high degree, our algorithm is super-quadratically faster than the best rigorously proven classical runtimes for regular graphs. Our results also shed light

on the quantum nature of short path algorithms, by identifying a setting where our algorithm is super-quadratically faster than any polynomial time Gibbs sampler, unless NP = RP. This provides evidence that a classical algorithm that is only quadratically slower cannot be constructed from a fast mixing Markov Chain that converges to Gibbs distribution. We conclude our analysis with a numerical analysis that guides the choice of parameters for short path algorithms and raises the possibility of super-quadratic speedups in settings that are currently beyond our theoretical analysis.

Although our applications are drawn from discrete combinatorial problems, the same framework can be extended—through appropriate discretizations—to continuous optimization tasks such as optimizing non-convex functions.

This chapter is based on [CHO+24], joint with Shouvanik Chakrabarti, Dylan Herman, Shuchen Zhu, Brandon Augustino, Tianyi Hao, Zichang He, Ruslan Shaydulin, Marco Pistoia.

4.1 Introduction

4.1.1 Motivation

The prospect of quantum algorithmic speedups for combinatorial optimization has been heavily studied for more than two decades [FGGS00, FGG14, Has18b, Mon18, Mon20]. This interest is partially motivated by practical considerations, since combinatorial optimization problems are ubiquitous in scientific and industrial applications, and are a major source of computational bottlenecks [AAA+24, HGL+23, DMB+23]. A second principled motivation is that there are reasons to expect such a speedup, arising from the existence of quantum algorithms such as Grover's search algorithm [Gro96b], which enjoys a quadratic quantum speedup over brute force search. Since the best classical algorithms with rigorously provable runtimes for combinatorial optimization often reduce to (possibly structured) search over a large space of possible solutions, one may expect algorithms building on Grover Search to provide speedups for these algorithms. In recent years, this intuition has been largely confirmed with the development of quantum-accelerated versions that offer quadratic speedups for backtracking [Mon18, AK17] and branch-and-bound [Mon20, CMYP22], two of the main classical meta-algorithms used to obtain provable runtime guarantees. There has been some success in obtaining sub-quadratic speedups for combinatorial optimization algorithms based on Markov Chain Monte Carlo [WA08], and dynamic programming [ABI+19].

Despite this progress, there remain challenges towards leveraging quantum algorithms for combinatorial optimization. Firstly, there are many problems for which quadratic quantum speedups over the state-of-the-art classical approaches have not been demonstrated, including cases where the best algorithm is based on dynamic programming [FGS06], MCMC, and local search [Sch02, ST13]. The second challenge is more fundamental, as recent research has identified challenges towards the realization of quadratic quantum speedups due to constant-factor slowdowns compared to classical hardware manifesting from slower clock speeds, the overhead of error-correction, and the limited parallelizability of quantum algorithms. Realistic estimates for the resources required to execute a quantum algorithm on a scale where it can break-even with classical computing can result in quantum runtimes exceeding many days. The viability of practical realization of a polynomial speedup increases with the degree of the speedup. For example, the resource analysis in [BMN+21] indicates that the outlook for realizing a quartic speedup, when considering all overheads, can be much more realistic, requiring quantum runtimes on the scale of hours instead of days. It is thus of fundamental importance to investigate whether it is possible to obtain *super-quadratic* speedups for combinatorial optimization.

The general quantum speedups for backtracking [Mon18], branch-and-bound [Mon20], dynamic programming [ABI+19] and MCMC methods either rely directly on Grover Search, or closely related methods like amplitude estimation, quantum minimum finding, or discrete time quantum walks. These frameworks therefore admit at most quadratic speedups by construction. Furthermore, in the case of unstructured search, backtracking and branch-and-bound, quadratic speedups can be shown to be the best one can hope for in the oracle setting. The study of super-quadratic speedups necessitates the investigation of mechanisms for quantum speedup beyond Grover Search. It is also likely that these speedups must leverage problem specific structure to circumvent the aforementioned lower bounds in the oracle setting.

A simpler, but very non-trivial question, is whether there are quantum algorithms for combinatorial optimization that achieve *super-Grover* speedups. That is, a super-quadratic speedup over unstructured search, but not necessarily the best classical algorithm. An important first step towards rigorously obtaining positive results in this direction was the *short path* quantum algorithm from Hastings [Has18b], which was demonstrated to solve combinatorial optimization problems with runtime $\mathcal{O}^*\left(2^{(0.5-c(n))n}\right)$[1], where $c(n)$ is a positively valued function of n. A limitation of this result was that the rigorously established bounds on $c(n)$ asymptote to 0 as n increases, leading to

[1]We use the notation $\mathcal{O}^*(2^{h(n)})$ to indicate an upper bound on the runtime of the form $\mathcal{O}(\text{poly}(n)2^{h(n)})$.

sub-polynomial improvements over Grover search. In a followup work that built on the framework of Hastings (but using a significantly modified algorithm and analysis), Dalzell et al. [DPCB23] gave an algorithm that obtains *strictly* super-Grover speedups for several combinatorial optimization problems, i.e. the algorithm achieves a runtime $\mathcal{O}^*\left(2^{(0.5-c)n}\right)$, where c is a positive parameter independent of n. Our work largely builds on the algorithm of Dalzell et al. [DPCB23], which we henceforth refer to an improved short path algorithm due to its connection with Hastings' original work.

The successful demonstration of super-Grover speedups leads to natural optimism that similar techniques could be used in principle to show super-quadratic speedups over the best known classical algorithm. There remain several challenges towards such a demonstration. On one hand, the speedups shown in [DPCB23] are only larger than quadratic by very small factors. On the other hand, the best performing classical algorithms for well-studied combinatorial optimization problems (while still exponential-time) are significantly faster than unstructured search. For instance, the well-known 3-SAT problem can be solved in time $\mathcal{O}^*\left(2^{0.39n}\right)$, the maximum independent set of an n-vertex graph can be found in time $\mathcal{O}^*(2^{0.258n})$, and the exact ground state of the Sherrington-Kirkpatrick model can be determined in time $\mathcal{O}^*\left(2^{0.45n}\right)$. As a consequence, the runtimes established in [DPCB23] are in most cases slower than the best classical algorithm. An exception to this is the problem of minimizing the energy of k-spin models, for which there has been limited study of classical algorithms. It is important to note that the mathematical analysis of [DPCB23] does not make much effort to optimize the parameters of the algorithm, and the actual runtime is predicted to be better than the theoretical predictions. However, we provide results from a numerical simulation of the algorithm in [DPCB23] which indicate that, even when the parameters are optimized, the speedup is likely to be insufficient by itself. For Maximum Independent Set on graphs with up to 21 vertices, the best runtime scaling (using a penalty term to enforce the constraints) achieved in our experiments is around $\mathcal{O}^*\left(2^{0.400n}\right)$. This is slower than the best classical algorithm despite optimizing the parameters, indicating that the frameworks in [Has18b, DPCB23] must be further generalized if genuine super-quadratic speedups are the goal.

A second consideration is that many combinatorial optimization problems of industrial importance are constrained, including well-studied examples such as Maximum Independent Set, Maximum/Minimum Bisection, Vertex and Set Cover, Portfolio Optimization, and Hamiltonian Cycles. The current short path algorithms can only incorporate constraints by adding penalty terms to the cost function, an approach rarely used by the

state-of-the-art algorithms for combinatorial optimization. Furthermore, the runtime of the short path algorithm scales with the number of bit-strings defined on the unconstrained solution space, which can often be much larger than the number of bit-strings that satisfy all constraints in the problem formulation. As an example, consider a combinatorial optimization problem with a constraint requiring solutions to have Hamming weight $\lfloor n^\alpha \rfloor$ for some $0 < \alpha < 1$. The number of *feasible* bit-strings is about $2^{(1-\alpha)n^\alpha \log(n)}$, which is asymptotically smaller than 2^n by a significant margin. For such constrained problems, the short path algorithm cannot even offer a super-quadratic speedup over unstructured search (if the search is restricted to the feasible region).

We seek to address these limitations by analyzing a generalized version of short-path algorithms that, under some technical conditions, obtain super-quadratic speedups over classical algorithms that search the space of solutions using samples from the stationary distribution of a Markov Chain. We refer to such algorithms as *Markov Chain Search*[2], and the framework can be simply described as follows. Suppose that our aim is to minimize a real-valued cost function $H \colon \mathcal{X} \mapsto \mathbb{R}$ for a finite set $\mathcal{X} \subset \{-1, 1\}^n$, and let P be the transition matrix of a Markov Chain that mixes to a stationary distribution π supported on $\mathcal{X}$ in poly(n) steps. A Markov Chain Search algorithm using P simply runs the chain to draw samples from π, and keeps track of the running minimum of the samples in terms of the cost function H. This minimum (and the corresponding sample) serves as an estimate of the global minimum (and minimizer) of H on $\mathcal{X}$. To bound the expected runtime of this algorithm, it is sufficient to bound the expected number of samples before a global minimum is encountered. Letting π^* denote the total probability that a sample from π is a global minimizer of H, it follows that drawing $\mathcal{O}\left((\pi^*)^{-1} \log(\epsilon^{-1})\right)$ samples from π suffices to ensure that we encounter the global minimizer with probability at least $1 - \epsilon$. We outline this framework in Algorithm 8.

Markov Chain Search is a natural extension of unstructured search, with several advantages. Firstly, it is often possible to classically sample from distributions over the feasible set that favor lower cost solutions. A common example is sampling from the Gibbs distribution corresponding to the cost function H, where a solution $z \in \mathcal{X}$ is sampled with probability proportional to $\exp(-\beta H(z))$ for some parameter β (usually called the inverse temperature). Gibbs distributions are the stationary distributions of well known chains such as the Glauber Dynamics [Gla63] (also referred to as Metropolis sampling), and the mixing times of these Markov Chains have been extensively studied in Physics and Computer Science for decades. It is evident that in a Gibbs distribution

[2]We refer the reader to Section 4.2 for background on Markov Chains.

with $\beta > 0$, low cost solutions are more likely than in the uniform distribution. If the global minima has significantly lower cost than most of the ensemble, this can lead to algorithms that are polynomially faster than unstructured search. We note that Dalzell et al. [DPCB23] make reference to precisely this framework when examining the possibility of faster classical algorithms for k-spin problems. A second advantage of Markov Chain Search arises when the feasible set is asymptotically much smaller than 2^n as discussed previously. Clearly, if there is a chain $\mathcal{M}$ that mixes in polynomial time to the uniform distribution over $|\mathcal{X}|$, the Markov Chain Search algorithm with $\mathcal{M}$ is faster than unstructured search over the unconstrained space. Even if we can only prepare a distribution (not necessarily uniform) whose support is restricted to $\mathcal{X}$, this may result in a runtime that is substantially better than unstructured search. While the classical analysis of precise runtimes using Markov Chain Search is typically challenging, we later give explicit examples of settings where this separation from unstructured search is realized.

4.1.2 Contributions

Our primary contribution is the formulation of the generalized short-path algorithm and the demonstration that, under certain technical conditions, it obtains a super-quadratic speedup over unstructured search. In particular we obtain quantum runtimes of $\mathcal{O}^*\left(T(n)^{0.5-c(n)}\right)$ where $T(n)$ is the (exponential in n) runtime of the classical Markov Chain Search, and $c(n) > 0$ is a positive parameter. When $c(n)$ is bounded below by a constant $c > 0$ we say we have a true super-quadratic speedup, and when $c(n) > 0$ for $n < \infty$ but $c(n) = o(1)$ we say we have an *asymptotically decaying* advantage over quadratic speedup (in the vein of Hastings' results [Has18b, Has18a, Has19]). We go on to discuss how these conditions may be established and demonstrate explicit results in two settings.

1. As an example of constrained optimization, we focus on optimization over strings of fixed Hamming Weight, using a Markov Chain based on random transpositions. In this case, Markov Chain search reduces simply to a brute force search over *feasible* strings, i.e. those that satisfy the Hamming weight constraint. Correspondingly, we identify conditions on the cost function for which a super-quadratic speedup using the short path framework can be obtained over brute-force search restricted to feasible states. As an explicit example, we consider the MaxBisection problem. We note that the new runtime by itself is not particularly interesting as the number of

feasible solutions (i.e., the number of n-bit binary strings of Hamming weight $n/2$) is smaller than 2^n by a polynomial factor and thus, the runtime for Markov Chain Search is not notably faster than unstructured search for this problem. However, applying the generalized framework allows for an algorithm that is completely restricted to feasible states, and the corresponding runtime cannot be obtained using the existing frameworks found in [Has18b, DPCB23].

2. Our next application is to Markov Chain search with non-uniform sampling, particularly to search with Gibbs distributions prepared using the Glauber Dynamics. This analysis yields conditions for super-quadratic speedups for problems such as the Maximum Independent Set problem, the Sherrington Kirkpatrick Model, and the Ising Model on graphs of bounded degree, for which Markov Chain search outperforms unstructured search. For the maximum independent set problem on regular graphs, we demonstrate that if the degree is a sufficiently high constant, Markov Chain search with the Glauber dynamics outperforms the best known theoretical classical algorithms for this problem. We also identify a number of cases where our analysis yields algorithms with a super-quadratic advantage over Markov Chain search where the advantage is asymptotically quadratic for large n (in the vein of Hastings' original results [Has18b]).

The following theorem informally summarizes the applications of the framework.

Theorem 4.1.1 (Applications of Generalized Short-Path Framework (informal))**.** *The following optimization problems exhibit a quantum runtime of* $\mathcal{O}^*\left(T(n)^{0.5-c(n)}\right)$*, where* $T(n)$ *is the runtime of a classical algorithm based on Markov Chain Search, and* $c > 0$ *quantifies the degree of improvement over quadratic speedup:*

1. *Maximum Bisection on random graphs with constant average degree, where the Markov Chain is the transposition walk and* $c(n) = \Theta(1)$*.*

2. *Maximum Independent Set and Ising Model on graphs of constant maximum degree, where the Markov Chain is the Glauber Dynamics at any temperature that permits polynomial mixing, and* $c(n) = \Theta(1)$*.*

3. *A generalization of Maximum Bisection where the smaller partition is constrained to have* k *nodes (where* $k \leq n/2$*). The Markov Chain is the transposition walk and* $c(n) = \Theta\left(1/\log(n/k)\right)$*.*

4. *The Sherrington-Kirkpatrick Model where the Markov Chain is the Glauber Dynamics at any temperature that permits polynomial mixing, and* $c(n) = \Theta\left(1/\log(n)\right)$.

The first two items above represent a constant improvement over quadratic speedup, whereas in the latter two, the speedup decays to 0 *as* n *increases. The Markov Chain Search considered is faster than brute force search on Maximum Independent Set for random regular graphs of sufficiently high degree, and on all instances of the other problems.*

From a technical point of view, while the skeleton of the framework follows that of [DPCB23] quite closely, the intermediate conditions must all be uplifted to incorporate the Markov Chain used for the base classical algorithm, and significant care is needed to obtain the generalized results. These generalizations are crucial to leverage the two main advantages of Markov Chain search over unstructured search, the ability to sample from non-uniform distributions and distributions with support restricted to feasible solutions. As we will later show, these cannot be achieved by specializations of existing frameworks. The uplift of the conditions also clarifies some aspects of the role they play in the argument. We found that approaching the original results of [DPCB23] from this new perspective led to some additional insights, which may be implicit, but are not explicitly documented in other works. We note also that the algorithm of [DPCB23] follows directly as a special case of our generalized framework by considering the bit-flip walk (or the random walk on the vertices of the hypercube). Finally, the generalized analysis in this chapter allows us to greatly simplify some of the statistical mechanics arguments made in [DPCB23], and instead rely directly on some standard results from the theory of Markov Chains and this may be of separate interest.

Our techniques also allow us to shed more light on a fundamental question about the viability of true super-quadratic speedups with the short path framework. The algorithms in this chapter as well as the earlier frameworks, rely on preparing a quantum state whose overlap with the global minimizers is larger than that of some easily prepared starting state and jumping to the global minimum from that state. It is apparent that if there existed a classical algorithm to sample in polynomial time from a distribution with overlap that matches that of this intermediate state, then there is a classical algorithm that finds the global minimum only quadratically slower than the short path algorithm. The advantage over search is then essentially *dequantized* and there is no hope for true super-quadratic speedups. It is therefore important to understand to what degree classical sampling techniques can approach the overlap of the intermediate state, as discussed in [DPCB23]. Most natural sampling algorithms are based on the analysis of Markov

Chains and so our framework provides a useful tool to probe this question. Our results on the Maximum Independent Set for bounded degree graphs yield some concrete evidence that the advantage of the short path framework over search cannot simply be removed by classical Gibbs sampling. Specifically, we demonstrate a concrete optimization problem for which the intermediate state prepared by the short path algorithm has higher overlap than any Gibbs distribution that can be prepared in polynomial time, unless NP = RP.

4.1.3 Related Works

As discussed at length in the paragraph above, the primary inspiration for our work comes from the earlier short path algorithms developed in the series of works [Has18b, Has19, Has18a, DPCB23]. We now discuss other related quantum algorithms and their connections to our results. The most closely related family of algorithms are based on discrete time quantum walks. The framework of Magniez at al [MNRS07] considers a framework that closely matches the one considered here. In [MNRS07], the authors seek to accelerate a classical algorithm, that first prepares in time S a sample from the stationary distribution of a Markov Chain with spectral gap δ, that has ϵ overlap with some marked state that can be checked in time C, and then runs the Markov Chain to repeatedly draw and check samples from the stationary distribution. The classical algorithm finds a marked element in time $\mathcal{O}\left(\mathrm{S} + \epsilon^{-1}(\delta^{-1}\mathrm{U} + \mathrm{C})\right)$ where U is the cost of simulating one step of the chain. The quantum algorithm obtains a runtime of $\mathcal{O}\left(\mathrm{S} + \epsilon^{-1/2}(\delta^{-1/2}\mathrm{U} + \mathrm{C})\right)$.

In our setting ϵ corresponds to $\pi(E^*)$ and falls exponentially, whereas $\mathrm{S}, \mathrm{U}, \mathrm{C}, \delta^{-1}$ all grow polynomially. Applying the [MNRS07] framework, we obtain a quantum runtime of $\mathcal{O}^*(\pi(E^*)^{-0.5})$. If the conditions for our framework are met, we obtain a runtime of $\mathcal{O}^*(\pi(E^*)^{-0.5+c(n)})$. In this setting, our algorithm accelerates the [MNRS07] framework in a manner analogous to how [DPCB23] accelerates Grover search. Another common framework for analyzing search via quantum walk is that of Szegedy [Sze04] where the quadratic speedup is over the hitting time of a marked vertex. However, since the Hitting Time HT from stationary distribution satisfies $\epsilon^{-1} \leq \mathrm{HT} \leq \epsilon^{-1}\delta^{-1}$, we have $\mathrm{HT} = \mathcal{O}^*(\epsilon^{-1})$ in our setting where ϵ^{-1} is exponentially larger than δ^{-1}, and the runtime of both quantum walk frameworks match up to polynomial factors.

Aside from the works we have already mentioned, quantum algorithms have also been successfully leveraged to obtain speedups for solving linear systems of equations [HHL09b, CKS17, CGJ19], computing partition functions [HW20, CH23], estimating the volume of convex bodies [CCH$^+$23], as well as sampling from both log-

concave [CLL+22] and non-logconcave [OLMW24] distributions. We note that our work is primarily concerned with exponential time search algorithms instead of the randomized approximation schemes considered in these papers. It would be interesting to understand whether our methods can be used to obtain super-quadratic speedups for exponential time counting algorithms [GLR21]. Along the line of super-quadratic quantum speedups, there is also recent work providing positive results for Tensor Principle Component Analysis [SOKB24], and approximation algorithms for combinatorial optimization [JSW+24].

4.2 Preliminaries

We write log and ln to indicate logarithms base 2 and base e, respectively. We denote the i-th element of a vector $x \in \mathbb{C}^n$ by x_i, and the ij-th element of a matrix $A \in \mathbb{C}^{m\times n}$ by A_{ij}. For a vector $x \in \mathbb{C}^n$, the matrix $\mathrm{diag}(x) \in \mathbb{C}^{n\times n}$ takes the values of x on its diagonal and zero elsewhere. We write $A \succeq 0$ ($A \succ 0$) to indicate that a matrix $A \in \mathbb{C}^{n\times n}$ is positive semidefinite (positive definite), i.e., all of its eigenvalues are nonnegative (positive). For two $m \times n$ matrices A and B, we write $A \circ B$ to indicate their Hadamard (or, element-wise) product. Note that when we say the phrase *with high probability* (w.h.p. for short), we imply that a result holds asymptotically in the problem size n with probability 1.

4.2.1 Short path algorithms

Let $H : \{-1,1\}^n \mapsto \mathbb{R}$ be a classical cost function satisfying $\sum_z H(z) = 0$ for H with no constant term. Consider the combinatorial optimization problem

$$E^* := \min_{z\in\{-1,1\}^n} H(z), \tag{4.1}$$

where E^* is the optimal objective value. Let Π^* denote the orthogonal projector onto subspace spanned by optimal assignments $|z^*\rangle$.

Let $X = \sum_{i\in[n]} X_i$ be the transverse-field operator, where X_i denotes the Pauli-X operator acting on qubit $i \in [n]$. A well-known approach to determine some $|z^*\rangle$ is the *quantum adiabatic algorithm* (QAA) [FGGS00]. The QAA finds a $|z^*\rangle$ by considering the adiabatic time evolution of

$$H_b^{(\mathrm{QAA})} = -(1-b)X + bH \tag{4.2}$$

as b is tuned from $b = 0$ to $b = 1$. However, QAA is known to suffer from certain *localization* issues, which can be viewed as a quantum analogue of getting trapped in local minima, and can result in run times that are exponentially worse than classical brute-force search [AKR10]. This manifests as a result of the *avoided crossing* phenomenon, or first-order phase transition that can lead to exponentially (or even super-exponentially) small spectral gaps.

Recently, Hastings [Has18b] and Dalzell et al. [DPCB23] proposed a new paradigm for avoiding the first-order phase transition problem with the QAA. Following the approach of [DPCB23], prototypical adiabatic optimization is eschewed through two modifications. First, the term H is replaced with $g_\eta\left(\frac{H}{|E^*|}\right)$ for a piecewise-linear function $g_\eta : [-1, \infty) \mapsto [-1, 0]$ parameterized by $\eta \in [0, 1)$:

$$g_\eta(x) := \min\left(0, \frac{x + 1 - \eta}{\eta}\right), \tag{4.3}$$

leading to the Hamiltonian:

$$H_b = -\frac{X}{n} + b g_\eta\left(\frac{H}{|E^*|}\right), \tag{4.4}$$

where X has been normalized by its spectral norm, and $b \in [0, \infty)$. Second, rather than slowly evolve from $-\frac{X}{n}$ to $\frac{H}{|E^*|}$ as in the QAA, we jump from $-\frac{X}{n}$ to the ground state $|\psi_b\rangle$ of H_b for some value of b that is independent of n (where the spectral gap is guaranteed to be large), and then jump from H_b to the ground-state space of $\frac{H}{|E^*|}$. Note in [DPCB23] they also allow for scaling H by an overestimate of $|E^*|$, for simplicity we just stick with scaling by $|E^*|$.

The jumps are accomplished using a unitary U, which combines phase estimation and amplitude amplification. For a high-level understanding, suppose we seek to enact a jump between two Hamiltonians H_1 and H_2, each acting on n qubits. Denote the ground state of H_1 by $|\psi_1\rangle$, and let Π_2 be the projector onto the ground space of H_2. The unitary U first employs phase estimation to implement reflection operators R_1 and R_2 that reflect about the state $|\psi_1\rangle$, and the groundspace of H_2, respectively. If δ_j is the spectral gap of Hamiltonian H_j, the operator R_j can be implemented up to error ε using $\delta_j^{-1} \log(1/\varepsilon)$ calls to a block-encoding of H_j, and often realizable using $\mathrm{poly}(n)$ gates. When H_j is a classical Hamiltonian, R_j can be implemented using $\mathrm{poly}(n)$ gates irrespective of δ_j. From here, the unitary U employs fixed-point amplitude amplification [YLC14] to implement $\frac{\Pi_2|\psi_1\rangle}{\|\Pi_2|\psi_1\rangle\|}$, requiring at most $\mathcal{O}\left(\|\Pi_2 |\psi_1\rangle\|^{-1} \log(1/\varepsilon)\right)$ applications each of R_1

and R_2.

The algorithm is initialized to $|+\rangle := |+\rangle^{\otimes n}$, the ground state of $-\frac{X}{n}$. Then, the ground state $|\psi_b\rangle$ of H_b is prepared by jumping from $-X/n$ to H_b. Finally, we prepare $\frac{\Pi^*|\psi_b\rangle}{\|\Pi^*|\psi_b\rangle\|}$ by jumping from H_b to the classical Hamiltonian $\frac{H}{|E^*|}$. The state $\Pi^*|\psi_b\rangle$ is a superposition of optimal solutions to $\min_{z\in\{\pm1\}^n} H(z)$, and thus measurement in the computational basis yields an optimal bit-string z^* with high probability. The first jump is small (in the sense that the success probability is nearly 1), whereas the second jump is large (the success probability is exponentially small). The resulting time complexity scales as $\mathcal{O}^*\left(2^{\left(\frac{1}{2}-c\right)n}\right)$, indicating a super-Grover speedup when $c > 0$. In [Has18b], the order of short and long jump steps is reversed. We refer to both approaches [Has18b, DPCB23] as *short path algorithms*.

4.3 Technical Overview

4.3.1 Framework

Define $\mathcal{X} \subseteq \{-1, 1\}^n$ and let $H\colon \mathcal{X} \mapsto \mathbb{R}$ be a cost function. We are interested in (exactly) determining $z^* \in \mathcal{X}$ such that

$$z^* \in \arg\min_{z\in\mathcal{X}} H(z). \tag{4.5}$$

We also use H to refer to a diagonal Hamiltonian in a Hilbert space with a basis indexed by $z \in \mathcal{X}$, with $\langle z|H|z\rangle$ for the Hamiltonian identified with $H(z)$. Whether we are referring to the quantum Hamiltonian or the function will be clear from context. We define $E^* := \min_{z\in\mathcal{X}} H(z)$ and assume everywhere that the cost is scaled to ensure $E^* < 0$. We will further assume that $|\mathcal{X}|$ is super-polynomial, as our primary concern is with super-quadratic speedups over exponential time algorithms. If π is a distribution such that $\pi(E^*)$ is the probability that a sample from π is a global minimizer and there exists a Markov Chain with transition matrix P that mixes to stationary distribution π such that the mixing time is bounded by $t_{\text{mix}}(\varepsilon) = \text{poly}(n, \log(\varepsilon^{-1}))$.

Under these conditions, it follows that Algorithm 8 finds the global minimizer of H in time $\mathcal{O}^*\left(\pi(E^*)^{-1}\right)$. Our framework seeks to accelerate this runtime, and so we assume the same setting for the quantum algorithm. In order to define the quantum framework we must access H, P, and π. We assume the existence of the following subroutines:

Assumption 4.3.1 (Quantum Input Assumptions)**.** The following subroutines are used as primitives in the Generalized Quantum Short Path framework.

Algorithm 8 MarkovChainSearch

Require: Solution space $\mathcal{X} \subset \{-1,1\}^n$, Cost function $H\colon \mathcal{X} \mapsto \mathbb{R}$, distribution π such that $\pi(E^*)$ is the probability that a sample from π is a global minimizer, a Markov Chain with transition matrix P and mixing time $t_{\text{mix}}(\pi(E^*)/2) = \mathcal{O}(\text{poly}(n))$.

input : Description of poly(n) time procedures to evaluate H and perform a step of the Markov Chain described by P, failure probability δ.

output A global minimizer z^* of $H(z)$ over $\mathcal{X}$.

Set $i = 0$, $z^{(0)}$ to an arbitrary point in $\mathcal{X}$.
while $i \leq \frac{2}{\pi(E^*)} \log\left(\frac{1}{\delta}\right)$ **do**
  Simulate $t_{\text{mix}}(n)$ steps of P to obtain sample $\tilde{z}$
  if $H(\tilde{z}) \leq H(z^{(i)})$ **then**
    Set $z^{(i+1)} = \tilde{z}$ and $i \leftarrow i + 1$.
  else
    Set $z^{(i+1)} = z^{(i)}$ and $i \leftarrow i + 1$.
  end if
end while
Output $z^{(i)}$.

1. **Initial State Preparation:** We assume the existence of a unitary U_π implementable using poly(n) gates, such that $U_\pi|0\rangle = |\sqrt{\pi}\rangle \coloneqq \sum_{z\in\mathcal{X}} \sqrt{\pi(z)}|z\rangle$.

2. **Block-encoding of Markov Chain:** Suppose that P is the transition matrix of a reversible Markov chain, then the discriminant operator $D(P)$ (see Definition 1.3.8) is Hermitian. We assume the existence of a unitary $U_{D(P)}$, implementable with poly(n) gates, that is a (κ_1, a) block-encoding of $D(P)$ for $\kappa_1 = \mathcal{O}(\text{poly}(n))$.

3. **Block-encoding of Cost Function:** We assume the existence of a unitary U_H, implementable with poly(n) gates, that is a (κ_2, a) block-encoding of H for $\kappa_2 = \mathcal{O}(\text{poly}(n))$.

We will justify Assumption 4.3.1 for each application of the framework. Note that preparing a block-encoding of the cost function is straightforward given a poly(n) size circuit to evaluate it at a single point, and we do not make this analysis in every case. We also do not explicitly mention these input assumptions in each of our results to avoid cluttering the presentation, but they are prerequisites for the input model in each case. With this setup, we can define the generalized short path framework. We define a *generalized short path Hamiltonian* H_b as

$$H_b = -D(P) + bg_\eta\left(\frac{H}{|E^*|}\right), \tag{4.6}$$

where $D(P)$ is the Discriminant matrix corresponding to P and g_η is defined similarly to [DPCB23], by

$$g_\eta(x) := \min\left(0, \frac{x+1-\eta}{\eta}\right).$$

The block-encoding for H_b can be constructed using the linear-combination-of-block-encodings technique [GSLW19] using $U_{D(P)}$ and U_H.

The framework is specified in Algorithm 9. The implementation of the jumps uses the framework from [DPCB23, Proposition 21] that performs fixed point amplitude amplification using reflections constructed from the block-encodings of the Hamiltonians and the Quantum Singular Value Transform. The overall runtime of Algorithm 9, in terms of the number of queries to U_π and block-encodings of H and $D(P)$, is

$$[\min(\mathrm{Gap}(-D(P)), \mathrm{Gap}(H_b))]^{-1}\left(|\langle\sqrt{\pi}|\psi_b\rangle|^{-1} + \|\Pi^*|\psi_b\rangle\|_2^{-1}\right), \tag{4.7}$$

where Π^* is the projector onto the ground subspace of H. As in previous papers, we refer to one of the steps in the algorithm as the short jump and another as the long jump. The reason for this is that the choices of b, η made for applications of the framework will always ensure that the short jump can be carried out with a polynomial number of queries to $U_\pi, U_{D(P)}$, and a block-encoding of H_b. Thus when including Assumption 4.3.1, the runtime of the algorithm is therefore primarily determined by the long jump, and under the appropriate conditions is bounded by $\mathcal{O}^*(\pi(E^*)^{-(0.5-c)})$, leading to a super-quadratic speedup over Markov Chain Search.

Algorithm 9 GeneralizedShortPathAlgorithm

input Algorithmic parameters b, η, Problem parameters H, P, π, E^*, which define H_b in Equation (4.6).

output an optimal assignment z^* for H.

1: Prepare $|\sqrt{\pi}\rangle$, the ground state of $-D(P)$.
2: **Short Jump:** Prepare $|\psi_b\rangle$ up to exponentially small error with jump $-D(P) \to H_b$.
3: **Long Jump:** Prepare $\frac{\Pi^*|\psi_b\rangle}{\|\Pi^*|\psi_b\rangle\|}$ up to exponentially small error with jump $H_b \to \frac{H}{|E^*|}$.

Our bounds on the runtime rely on two conditions, that we view as uplifted versions of corresponding notions in [DPCB23] to the case of general Markov Chains. Our first condition captures the *smoothness* of the cost function under applications of the transition matrix.

Definition 4.3.2 (Δ_P stability)**.** Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a Markov chain. We say that

the cost Hamiltonian H is $\Delta_P(\eta)$ stable under $\mathcal{M}$ if

$$\mathbb{E}_{y \underset{P}{\sim} x}[h_\eta\left(H(y)\right)] \leq h_\eta\left(H(x) + \Delta_P(\eta)\right), \quad \forall x \in \mathcal{X} \tag{4.8}$$

where $h_\eta(x) := g_\eta\left(\frac{x}{|E^*|}\right)$. If the conditions holds for all $0 < \eta < 1$ we omit it and simply say H is Δ_P stable under $\mathcal{M}$.

The analysis in the following sections will clarify that Δ_P-stable is a generalization of the α-subdepolarizing condition introduced in [DPCB23]. In fact, it is equivalent to a more syntactically obvious generalization of the α-subdepolarizing condition, we state Definition 4.3.2 as the primary condition as it is easier to demonstrate and interpret in most cases.

The next condition is a generalization of the spectral density condition of [DPCB23]. We capture the idea that the measure (according to π) of the set of solutions z for which $g_\eta\left(\frac{H(z)}{|E^*|}\right) \neq 0$ is polynomially related to the measure of the global minimizer, for some value of η. In other words, sampling from π does not allow one to approximately minimize H to arbitrary constant relative error, super-polynomially faster than finding the exact minimum. If this condition is violated, the problem admits a simple classical sub-exponential time approximation scheme. We define this condition as follows:

Definition 4.3.3 (γ Spectral Density)**.** The cost Hamiltonian H is said to satisfy the γ spectral density condition with respect to the stationary distribution π if:

$$\pi(E \leq (1-\eta)E^*) \leq \pi(E^*)^\gamma.$$

We are now ready to state the main results concerning our framework. These results are subject to further technical conditions on the Markov Chain used for the search, and are formulated in terms of conditions that lead to efficient mixing time bounds. We have two variants of our result, the first relies on a logarithmic-Sobolev inequality for P. We have the following result:

Theorem 4.3.4 (informal)**.** *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a reversible Markov chain, and let $H : \mathcal{X} \mapsto \mathbb{R}$ be a diagonal, Δ_P-stable Hamiltonian with ground state energy E^*, that satisfies the γ spectral density condition for some parameters η. In addition, suppose $\mathcal{M}$ satisfies an ω log-Sobolev inequality. If $\omega^{-1} = \Theta(\ln(1/\pi(E^*))$, then there exists a constant b, such that under Assumption 4.3.1, Algorithm 9 determines the ground state*

of H over $\mathcal{X}$ with running time

$$\mathcal{O}\left(\text{poly}(n)\omega^{-1}[\pi(E^*)^{-1}]^{\left(\frac{1}{2}-\frac{\eta(1-\eta)|E^*|b}{2\ln(1/\pi(E^*))\Delta_P}\right)}\right).$$

We also present a variant of the above result that only relies on the weaker Poincaré inequality.

Theorem 4.3.5 (informal)**.** *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a reversible Markov chain, and let $H : \mathcal{X} \mapsto \mathbb{R}$ be a diagonal, Δ_P-stable Hamiltonian with ground state energy E^*, that satisfies a spectral density condition. In addition, suppose $\mathcal{M}$ satisfies a δ Poincaré inequality. If δ is independent of the problem size, then there exists a constant b, such that under Assumption 4.3.1, Algorithm 9 determines the ground state of H over $\mathcal{X}$ with running time*

$$\mathcal{O}\left(\text{poly}(n)\delta^{-1}[\pi(E^*)^{-1}]^{\left(\frac{1}{2}-\frac{\eta(1-\eta)|E^*|b}{2\ln(1/\pi(E^*))\Delta_P}\right)}\right).$$

The spectral density condition with respect to non-uniform starting states presents a technical challenge, since unlike the uniform distribution over all bitstrings of length n, they are no longer product measures. Fortunately, the condition that π is the ground state of a fast mixing Markov Chain allows for simplification via concentration inequalities for Markov chains, e.g. the so-called Herbst's argument [Lal13], that allows the spectral density conditions to be established as long as the cost function has an appropriately bounded pseudo-Lipschitz norm $\|H\|_P$. Theorems 4.3.4 and 4.3.5 are proven in Section 4.4, in order to establish specializations that rely on a bounded pseudo-Lipschitz norm.

4.3.2 Applications

The fact that our main results in Theorems 4.3.4 and 4.3.5 rely on functional inequalities implies many possibilities for interesting algorithmic speedups. Once a Markov Chain with the right properties (log-Sobolev or Poincaré with the proper parameters) is identified, we can derive conditions on cost functions for which we have super-quadratic speedup over Markov Chains. As a simple toy example, consider an expander graph of size 2^n, where we are given access to a polynomial time oracle that outputs the edges incident on any vertex. Since the graph is an expander, the graph random walk satisfies a Poincaré inequality with constant δ. Consider any assignment of costs to the nodes such that the difference in cost between any two endpoints of an edge is bounded above by a constant Δ. It follows from our results, that the generalized short path framework can find the

node with minimum cost with super-quadratically fewer queries than searching with the random walk on the graph. A systematic study of cost functions that yield a speedup for various Markov Chains may lead to some interesting insights. For this chapter, however, we focus on identifying connections to problems of general interest.

4.3.2.1 Optimization with Fixed Hamming Weight

We first consider optimization problems for which feasible solutions are bitstrings of fixed Hamming Weight. A well-studied example of this setting is Max-Bisection [FJ97, DMS17], defined as follows

$$\mathcal{C}^*_{\frac{n}{2}} := \min_{x\in\{-1,1\}^n} \left\{ -\frac{1}{2}\sum_{i<j} e_{ij}(1 - x_i x_j) : |x| = \frac{n}{2} \right\}, \tag{MaxBisection}$$

The algorithm of [DPCB23] does not directly yield useful results for such a problem. Firstly, the framework does not naturally incorporate constraints. More importantly, although one could attempt to enforce the constraints by means of penalty terms, this prohibits the possibility of super-Grover speedups. To see why this is the case, recall that the algorithm of [DPCB23] is simply our Algorithm 9 with the Markov Chain chosen to be the random walk on the edges of the hypercube. Since every transition of such a walk changes the hypercube, the best possible value of Δ_P for the stability condition to be satisfied is of the same order as the penalty terms. On the other hand, the penalty terms must be of the same order as the cost function in order to guarantee that constraints are satisfied. By inspection of Theorem 4.3.4 we observe that no super-Grover speedup is possible via the penalty-based approach (more details on the penalty approach are in Appendix 4.9). We overcome this challenge by employing a generalized framework that uses a Markov Chain, specifically the *Bernoulli-Laplace diffusion* or *transposition walk*, which preserves Hamming weights and transitions from a starting string of weight k to the equal superposition over all such strings. In Section 4.5.1, we present a condition on cost functions over Hamming Weight Slices k for which we obtain runtimes of the form $\text{poly}(n)\binom{n}{k}^{0.5-c}$ for a constant c.

We study a generalization of MaxBisection which we term MaxCut-Hamming, defined as

$$\mathcal{C}^*_k := \min_{x\in\{-1,1\}^n} \left\{ -\frac{1}{2}\sum_{i<j} e_{ij}(1 - x_i x_j) : |x| = k \right\}. \tag{MaxCut-Hamming}$$

In Section 4.5.1 we prove that the Generalized Short Path Framework achieves an

overall runtime of $\mathcal{O}^*\left(\binom{n}{k}^{0.5-c}\right)$ for MaxCut-Hamming on Erdős-Rényi random graphs when $k = \Theta(n)$ (which includes MaxBisection as a special case). We also demonstrate that under the assumption of spectral density, our approach achieves a runtime of $\mathcal{O}^*\left(\binom{n}{k}^{0.5-c(n)}\right)$ with $c(n) = \Theta\left(\log(n/k)^{-1}\right)$, and thus the super-quadratic advantage over Markov Chain search decays as $n \to \infty$ (similar to the results of [Has18b]).

4.3.2.2 Glauber Dynamics

Glauber Dynamics [Gla63] is a well known sampling algorithm designed to sample from the Gibbs measures corresponding to Hamiltonians such as the Ising or Hardcore models. Since sampling from a Gibbs measure at arbitrarily high inverse temperatures solves the exact optimization, for most hard problems there exists a critical threshold beyond which the Glauber dynamics no longer mixes efficiently. Performing Markov Chain Search with Glauber dynamics has two advantages, if the problem is constrained then it provides a natural way to search with a distribution whose support is restricted to feasible solutions only. On the other hand, if the Glauber dynamics mixes for positive inverse temperatures, then low-energy solutions are more favored compared to the uniform distribution and the result in Markov Chain Search is asymptotically faster than unstructured search (see Lemma 4.5.20). In each case, we will consider classical search algorithms that use the Glauber dynamics at a temperature slightly higher the critical threshold where mixing takes exponential time. We demonstrate in Section 4.5.2 that for three models of interest, we obtain super-quadratic speedups over polynomially-mixing Glauber dynamics. These models are:

1. **The Maximum Independent Set problem (or hardcore model) on graphs of constant maximum degree:** For this problem the Glauber dynamics is shown to mix only up to critical temperatures that are negative. This means that our starting distribution favors small sets compared to large sets. However, there is the advantage that the Gibbs distribution has support only on independent sets (which are usually much fewer in number than 2^n, which is the total number of subsets). In the case of random regular graphs of sufficiently high degree, we show that this Markov Chain Search algorithm is faster than unstructured search as well as the best known combinatorial algorithms for Maximum Independent Set.

2. **The Ising Model on random regular graphs of constant maximum degree:** The Ising Model is an unconstrained optimization problem and there are 2^n feasible solutions. In this setting, the Glauber dynamics mixes up to a positive critical

inverse-temperature, thus the starting stationary distribution favors low energy solutions and the Markov Chain search is faster than unstructured search.

3. **The Sherrington Kirkpatrick Model:** Like the Ising Model, the Sherrington Kirkpatrick model is also unconstrained and the Glauber dynamics mixes up to a positive inverse temperature. In this case, however, the exponent of our advantage over quadratic speedup falls with n. Specifically, we show a quantum runtime of $\mathcal{O}^*\left(\pi(E^*)^{-0.5+c(n)}\right)$ where $c(n) = \Theta(1/\log(n))$.

4.3.2.3 Super-Quadratic Speedup Over any Polynomial Time Gibbs Sampler

Our result regarding the Maximum Independent Set (or, hardcore model) on graphs of bounded degree allows us to go a step further and argue that it is very likely that the generalized short path can achieve a super-Grover speedup over all polynomial time Gibbs samplers (whether or not they are based on Glauber dynamics). The key observation is that for the hardcore model there is a critical fugacity $\lambda_c(d) = \exp(-\beta_c(d)) = \frac{(d-1)^{d-1}}{(d-2)^{d-2}}$ such that for graphs of maximum degree d, Glauber dynamics mixes in time $\mathcal{O}(n\log(n))$ for any $\lambda < \lambda_c(d)$. However, it has been shown that computing the partition function of the hardcore model is NP-hard for any fugacity $\lambda > \lambda_c(d)$ [Sly10, SS14]. Due to the well known reduction between sampling and counting, it must therefore be NP-hard to sample from the corresponding Gibbs distribution. We therefore have an almost complete classification of the Gibbs measures corresponding to the hardcore model at different fugacity: it is either NP-hard to sample from the distribution, or the Glauber dynamics mixes in polynomial time. Our results in Section 4.5.2.1 establish that in the latter setting, the *short jump* of Algorithm 9 takes only polynomial time and produces a state $|\psi_b\rangle$ such that it is NP-hard to sample from any Gibbs distribution π obeying $\pi(z^*) = \Omega\left(|\langle\psi_b|z^*\rangle|^2\right)$.

Consequently, there exists a quantitative separation between the ground states of the short path Hamiltonian and Gibbs distributions. It is notable that the Markov Chain methods are obstructed at the critical fugacity due to the development of long range correlations. Such long range correlations in the ground state are also likely to lead to a vanishing gap for the short path Hamiltonian. This indicates that the ground state of the short path algorithm increases the overlap with the ground state, without creating long range correlations. Further understanding of the qualitative differences between the ground state and classical measures such as the Gibbs distribution, may shed light on the mechanisms of the short path algorithm.

We note that we resort to computational assumptions above only to show that the ground state $|\psi_b\rangle$ has higher overlap with the optimal solution than an efficiently sampleable Gibbs distribution. It follows unconditionally from our analysis that the ground state itself does not encode a Gibbs distribution beyond the critical mixing threshold (or indeed any distribution using which we can sample from such a Gibbs distribution by rejection sampling). In particular, Lemma 4.4.20 shows that for any ground state prepared by a short jump that is covered by our analysis, the trace distance from the starting state, and hence the total variation distance from the starting distribution is exponentially small.

4.4 Generalized Short Path Framework

This section provides a simplified and generalized analysis of the short path algorithm presented in [DPCB23]. It also highlights limitations of the current method of analysis, and describes a general recipe for the determining a super-quadratic speedup.

4.4.1 Summary of main results

In what follows, let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a reversible Markov chain over a finite set $\mathcal{X}$. Here π denotes a stationary distribution that is uniform over $\mathcal{X}$ and P. It is assumed that the spectral gap of P is lower-bounded by δ. Using the discriminant matrix of P, we can define a more general short-path Hamiltonian H_b that allows one to work with mixing operators other than $-\frac{X}{n}$.

Definition 4.4.1 (Short-path Hamiltonian H_b)**.** Consider a reversible Markov chain $\mathcal{M} = (\mathcal{X}, P, \pi)$. Let $H : \mathcal{X} \mapsto \mathbb{R}$ be a cost Hamiltonian. The short-path Hamiltonian H_b is given by

$$H_b := -D(P) + bg_\eta\left(\frac{H}{|E^*|}\right),$$

where $D(P)$ is the discriminant matrix of P, and

$$g_\eta(x) := \min\left(0, \frac{x+1-\eta}{\eta}\right).$$

More generally $g_\eta : [-1, \infty) \mapsto [-1, 0]$ can be a non-decreasing, concave function that is differentiable at every point where it is non-zero. However, the specific choice

we make is sufficient for our purposes. We will also sometimes refer use the notation $G_\eta := g_\eta\left(\frac{H}{|E^*|}\right)$.

One major component of the analysis of Algorithm 9 is determining an upper bound on b for which the spectral gap of the short-path Hamiltonian H_b is still large, i.e. $\Omega\left(\frac{1}{\text{poly}(n)}\right)$. This upper bound serves as a proxy for how large the *short jump* is. A second major component is determining the increased overlap with the optimal solution provided by the short-jump. To do so, we rely on the definition of Δ_P stability, which we restate below.

Definition 4.4.2 (Δ_P stability)**.** Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a Markov chain. We say that the cost Hamiltonian H is $\Delta_P(\eta)$ stable under $\mathcal{M}$ if

$$\mathbb{E}_{y \underset{P}{\sim} x}[h_\eta\left(H(y)\right)] \leq h_\eta\left(H(x) + \Delta_P(\eta)\right), \quad \forall x \in \mathcal{X}$$

where $h_\eta(x) := g_\eta(\frac{x}{|E^*|})$.

If the short jump can be accomplished efficiently, then Δ_P stability captures whether the short path approach provides a super-Grover runtime. This condition also has an intuitive interpretation. If we consider the optimization landscape defined by $\mathcal{M}$, H and a well (controlled by η) around the global minimum with energy E^*, then we do not want the energy to increase too much when moving within and around the well. Specifically, for a super-Grover runtime it should hold that $\Theta\left(\frac{|E^*|}{\ln(1/\pi(E^*))}\right)$, where π is the stationary distribution of $\mathcal{M}$. It is worth remarking that if $\eta = 0$, we recover the quantum unstructured search algorithm.

It turns out that any upper bound on $\Delta_P(\eta)$ suffices when bounding the runtime. For example, it is a simple consequence of Jensen's inequality that we can take $\Delta_P(\eta)$ to be $\sqrt{\|\psi\|_P}$ with $\psi = H$. A key technical contribution of this work is to reduce the conditions for determining whether a super-Grover runtime is possible to determining the log-Sobolev constant ω and the P-pseudo Lipschitz norm $\|H\|_P$ (or Δ_P) for cost Hamiltonian H.

We summarize our main result below:

Theorem 4.4.3. *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a reversible Markov chain, and let $H : \mathcal{X} \mapsto \mathbb{R}$ be a diagonal, Δ_P-stable Hamiltonian with ground state energy E^*, P-pseudo Lipschitz norm $\|H\|_P$. In addition, suppose $\mathcal{M}$ has a log-Sobolev constant ω. If b satisfies*

$$b < b^* := \frac{2}{3}\gamma\omega \ln\left(\frac{1}{\pi(E^*)}\right), \tag{4.9}$$

where

$$\gamma = \frac{\omega((1-\eta)E^* - \mathbb{E}_\pi[H])^2}{\|H\|_P \ln(1/\pi(E^*))}, \tag{4.10}$$

then there exists a short-path algorithm that determines the ground state of H over $\mathcal{X}$ with running time

$$\mathcal{O}\left(\text{poly}(n)\omega^{-1}[\pi(E^*)^{-1}]^{\left(\frac{1}{2}-\frac{\eta(1-\eta)|E^*|b}{2\ln(1/\pi(E^*))\Delta_P}\right)}\right). \tag{4.11}$$

Note that any upper bound on Δ_P suffices, for example one may use $\sqrt{\|H\|_P}$.

Proof. The proof is evident after combining the statements of Theorems 4.4.28, 4.4.15, Lemma 4.4.25, and Corollary 4.4.9 □

We also present a variant of the above result that only relies on a Poincaré inequality.

Theorem 4.4.4. *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a reversible Markov chain, and let $H : \mathcal{X} \mapsto \mathbb{R}$ be a diagonal, Δ_P-stable Hamiltonian with ground state energy E^*, P-pseudo Lipschitz norm $\|H\|_P$. In addition, suppose $\mathcal{M}$ has a Poincaré constant δ. If b satisfies*

$$b < b^* := \delta\frac{4\sqrt{6}-1}{10}, \tag{4.12}$$

then there exists a short-path algorithm that determines the ground state of H over $\mathcal{X}$ with running time

$$\mathcal{O}\left(\text{poly}(n)\delta^{-1}[\pi(E^*)^{-1}]^{\left(\frac{1}{2}-\frac{\eta(1-\eta)|E^*|b}{2\ln(1/\pi(E^*))\Delta_P}\right)}\right). \tag{4.13}$$

Note that any upper bound on Δ_P suffices, for example one may use $\sqrt{\|H\|_P}$.

Proof. The proof is evident after combining the statements of Theorems 4.4.28, 4.4.17, Lemma 4.4.25, and Corollary 4.4.10. □

In general, the log-Sobolev constant ω can be significantly smaller than the spectral gap of the chain δ, however, we argue that this is not the case when Theorem 4.4.3 provides a super-Grover runtime. Specifically, Theorem 4.4.3 requires that b^* is a constant, and by extension, implies we need $\omega^{-1} = \Theta(\ln(1/\pi(E^*)))$. For example, for a very hard problem, where Markov Chain search finds an optimal assignment with exponentially-small probability, i.e., $\Theta(\ln(1/\pi(E^*))) = \Theta(n)$, the condition on b^* will imply that ω

will be large, i.e., $\Omega\left(\frac{1}{\text{poly}(n)}\right)$. Thus, in cases where it provides a super-Grover runtime, Theorem 4.4.3 asserts that we do not get a slower runtime by using ω instead of δ.

We have the following evident corollary of the above results.

Corollary 4.4.5. *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a reversible Markov chain, and let $H : \mathcal{X} \mapsto \mathbb{R}$ be a diagonal Hamiltonian with ground state energy E^*. Suppose the pair $(\mathcal{M}, H)$ result in b^* and γ that are independent of n. If*

$$\frac{|E^*|}{\Delta_P} = \Theta\left(\ln\left(1/\pi(E^*)\right)\right), \tag{4.14}$$

then under Assumption 4.3.1 there exists a short-path algorithm that finds the minimizer of H over $\mathcal{X}$ with super-Grover running time. The advantage over Grover is:

$$\frac{\eta(1-\eta)|E^*|b^*}{\Delta_P}. \tag{4.15}$$

4.4.2 Constructing Short Path Algorithms from Markov Chains

This subsection details how the results from [DPCB23] can be generalized to the setting of reversible Markov Chains. One of the main conditions from the aformentioned paper is that there should be a small number of low-energy states, effectively capturing that the underlying problem is hard. This is made precise through the spectral density condition, which we restate below.

Definition 4.4.6 (γ Spectral Density)**.** The cost Hamiltonian H is said to satisfy the γ spectral density condition with respect to the stationary distribution π if:

$$\pi(E \leq (1-\eta)E^*) \leq \pi(E^*)^\gamma. \tag{4.16}$$

The tail bound given by the spectral density condition is implied by Pseudo Lipschitzness together with a functional inequality:

Theorem 4.4.7 (Herbst's Argument, adapted from Theorem 4.3 in [Lal13])**.** *Suppose $\mathcal{M} = (\mathcal{X}, P, \pi)$ is a Markov chain with log-Sobolev constant ω, and $f : \mathcal{X} \mapsto \mathbb{R}$ is $\|f\|_P$ pseudo-Lipschitz. Then,*

$$\mathbb{P}_\pi[f \geq \mathbb{E}_\pi[f] + t] \leq e^{-\frac{\omega}{\|f\|_P}t^2}. \tag{4.17}$$

Theorem 4.4.8 (Adapted from Theorem 3.5 in [Lal13]). *Suppose $\mathcal{M} = (\mathcal{X}, P, \pi)$ is a Markov chain with Poincaré constant δ, and $f : \mathcal{X} \mapsto \mathbb{R}$ is $\|f\|_P$ pseudo-Lipschitz. Then,*

$$\mathbb{P}_\pi[f \geq \mathbb{E}_\pi[f] + t] \leq e^{-\frac{\sqrt{\delta}}{\sqrt{\|f\|_P}}t}. \tag{4.18}$$

The following corollary is immediate, and reduces spectral density to pseudo Lipschitzness and a functional inequality.

Corollary 4.4.9. *Suppose $\mathcal{M} = (\mathcal{X}, P, \pi)$ is a Markov chain with log-Sobolev constant ω, and that the cost function H is $\|H\|_P$ pseudo Lipschitz. Then,*

$$\mathbb{P}_\pi[H \leq (1-\eta)E^*] \leq \pi(E^*)^{\frac{\omega((1-\eta)E^* - \mathbb{E}_\pi[H])^2}{\|H\|_P \ln(1/\pi(E^*))}}, \tag{4.19}$$

with

$$\gamma = \frac{\omega((1-\eta)E^* - \mathbb{E}_\pi[H])^2}{\|H\|_P \ln(1/\pi(E^*))}. \tag{4.20}$$

Corollary 4.4.10. *Suppose $\mathcal{M} = (\mathcal{X}, P, \pi)$ is a Markov chain with Poincaré constant δ, and that the cost function H is $\|H\|_P$ pseudo Lipschitz. Then,*

$$\mathbb{P}_\pi[H \leq (1-\eta)E^*] \leq \pi(E^*)^{\frac{\sqrt{\delta}((1-\eta)E^* - \mathbb{E}_\pi[H])}{\|H\|_P \ln(1/\pi(E^*))}}, \tag{4.21}$$

with

$$\gamma = \frac{\sqrt{\delta}((1-\eta)E^* - \mathbb{E}_\pi[H])}{\sqrt{\|H\|_P} \ln(1/\pi(E^*))}. \tag{4.22}$$

As noted in previous papers, the spectral density condition is relatively weak for hard problems. For example, suppose that for any constants γ and η it were not satisfied, but $\pi(E^*) = \mathcal{O}(2^{-cn})$. Then Markov Chain search can prepare an η relative error, for η arbitrarly close to 1, approximate minimizer in time subexponential in n.

4.4.2.1 The Short Jump

The short jump is defined as the preparation of $|\psi_b\rangle$ from $|\sqrt{\pi}\rangle$. The ability to find a good point to short-jump to (i.e., constant b where a jump takes poly(n) time to make) is where the inherent speedup over Grover comes from. If we just decided to do only the short-jump and sample until we found the ground state, the algorithm would only be quadratically slower due to amplitude amplification on the long-jump (assuming good gap costs $|\langle\psi_b|z^*\rangle|^{-1}$), than the full short-path algorithm.

The goal of this section is to determine conditions, using an initial Hamiltonian that is the discriminant of a reversible Markov chain, under which a short-jump can be done efficiently. However, it does not determine whether such a short-jump provides super-Grover runtime, which is the goal of the next subsection. The runtime of a short jump is captured by the short-path condition from [DPCB23, Has18b], where we present a natural generalization.

Definition 4.4.11 (θ Short-path condition)**.** Let $\Pi_\perp := \mathbb{I} - |\sqrt{\pi}\rangle\langle\sqrt{\pi}|$. Then, the θ short-path condition holds for some constant $b > 0$ if

$$\mathsf{GSE}(\Pi_\perp H_b \Pi_\perp) \geq -1 + \theta \tag{4.23}$$

where GSE denotes "ground-state energy".

The short-path condition was used in previous papers to prove a variety of other sufficient conditions for super-Grover runtime. For example, the short path condition implies a lower bound on the spectral gap of H_b.

Lemma 4.4.12 (θ Short-path $\implies$ θ Spectral Gap Bound, adapted from Proposition 5 of [DPCB23])**.** *If H_b satisfies the θ short-path condition, then the spectral gap of H_b is at least θ, i.e., all excited states have energy at least $-1 + \theta$.*

However, we stress that the short-path condition's main purpose is to show the existence of an efficient "short jump", i.e. $|\sqrt{\pi}\rangle \to |\psi_b\rangle$. In fact an inverse-poly(n) short-path condition is effectively equivalent to an inverse-poly(n) spectral gap at H_b and overlap between $|\psi_b\rangle$ and $|\sqrt{\pi}\rangle$. Thus one can view it as a convenient way of combining the two conditions. The other consequence derived from the short-path condition was of a more technical nature. We summarize the implications of the short-path condition on the short jump in the following result.

Theorem 4.4.13 (Sufficient conditions for Efficient Short Jump)**.** *Suppose $\mathcal{M} = (\mathcal{X}, P, \pi)$ is a reversible Markov chain with spectral gap that is at least inverse-polynomial in n, and H is a cost Hamiltonian satisfying the θ short-path condition at b independent of the problem size n. If $\theta = \Omega\left(\frac{1}{\text{poly}(n)}\right)$, then there exists quantum algorithm for preparing an ε-approximation to $|\psi_b\rangle$ starting with $|\sqrt{\pi}\rangle$, which makes* poly$(n, \log(1/\varepsilon))$ *queries to block-encodings of $D(P)$ and H_b.*

Proof. The result will follow if we can show that

$$\min(\text{Gap}(-D(P)), \text{Gap}(H_b))|\langle\sqrt{\pi}|\psi_b\rangle| \tag{4.24}$$

is inverse-polynomial in n. The assumption on the Markov chain gap implies that $\mathrm{Gap}(-D(P))$ is inverse-polynomial. Lemma 4.4.12 implies that $\mathrm{Gap}(H_b) = \Omega\left(\frac{1}{\mathrm{poly}(n)}\right)$.

The overlap is implied by the following argument. Let $|\psi_b^{\perp}\rangle$ be the component of ψ_b orthogonal to $|\sqrt{\pi}\rangle$. The short-path condition implies that

$$\langle\psi_b|\Pi_{\perp}H_b\Pi_{\perp}|\psi_b\rangle \geq -1 + \Omega\left(\frac{1}{\mathrm{poly}(n)}\right), \tag{4.25}$$

and since $\langle\psi_b|H_b|\psi_b\rangle \leq -1$, we have

$$|\langle\psi_b|H_b|\psi_b\rangle - \langle\psi_b^{\perp}|H_b|\psi_b^{\perp}\rangle| = \Omega\left(\frac{1}{\mathrm{poly}(n)}\right). \tag{4.26}$$

Suppose $|\langle\sqrt{\pi}|\psi_b\rangle| = o\left(\frac{1}{\mathrm{poly}(n)}\right)$, then since $\|H_b\|_2 \leq 2$

$$|\langle\psi_b|H_b|\psi_b\rangle - \langle\psi_b^{\perp}|H_b|\psi_b^{\perp}\rangle| = o\left(\frac{1}{\mathrm{poly}(n)}\right), \tag{4.27}$$

a contradiction. □

We will later show that an even stronger condition on the overlap can be established assuming the spectral density condition and in this case $\langle\psi_b|\sqrt{\pi}\rangle = 1 - o(1)$.

If the conditions of Theorem 4.4.13 are satisfied, then we only need show that $\|\Pi^*|\psi_b\rangle\|_2^{-1}$ is exponentially smaller (say by $\pi(E^*)^{c/2}$ for some constant c) than $\|\Pi^*|\psi_b\rangle\|_2^{-1}$ to achieve a super-quadratic speedup over Markov chain search. This is the task of bounding the runtime of the long jump.

In this work and [DPCB23], the long-jump runtime is upper bounded by lower bounding the easier-to-handle quantity $|\langle\sqrt{\pi}|\psi_b\rangle\langle\psi_b|z^*\rangle|$ by constructing an approximation to $|\psi_b\rangle\langle\psi_b|$. This makes use of a technical condition that ensures the ground state energy of H_b, E_b, does not decrease too much from the ground state energy of $-D(P)$. While it can be shown to hold via the short-path condition, we give intuition that it is a significantly weaker condition.

The following results provide generalized conditions under which the existence of a log-Sobolev and/or Poincaré inequality enable the spectral density condition (a very weak condition) to imply a constant b at which a θ short-path condition exists, where θ is either the log-Sobolev or spectral gap of the chain $\mathcal{M}$. For Poincaré case, the spectral density condition is not even needed, although the spectral gap must of $\mathcal{M}$ must be

constant.

From a bird's-eye view, the role of the functional inequality is to upper bound a metric or divergence between ψ_b^2, ℓ_2 distribution of $|\psi_b\rangle$ in the computational basis, and π. The variational definition of the corresponding metric or divergence plays the role of lower bounding. Together, these bounds are sufficient to derive a range of b's where short-path holds.

Lemma 4.4.14. *Suppose $\mathcal{M} = (\mathcal{X}, P, \pi)$ is a Markov chain that satisfies a log-Sobolev inequality with constant ω. Then, for all quantum states $|\psi\rangle$ one has*

$$\frac{1 - \langle\psi|D(P)|\psi\rangle}{\omega} \geq \mathrm{KL}(\psi^2\|\pi), \tag{4.28}$$

where $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence, and ψ^2 is the ℓ_2 distribution of $|\psi\rangle$ in the computational basis.

Proof. From a straightforward calculation, it follows

$$\begin{aligned}
\mathcal{D}(\psi,\psi) &= \langle\psi, (I-P)\psi\rangle_\pi && (4.29)\\
&= \mathbb{E}_\pi(\psi^\mathsf{T}\psi) - \mathbb{E}_\pi(\psi^\mathsf{T}P\psi) && (4.30)\\
&= \sum_{x\in\mathcal{X}} (\sqrt{\pi(x)}\psi(x))^2 - \sum_{x\in\mathcal{X}} \sqrt{\pi(x)}\psi(x)D(P)_{xy}\sqrt{\pi(y)}\psi(y) && (4.31)\\
&= \|\sqrt{\pi}\psi\|_2^2 - \langle\sqrt{\pi}\psi|D(P)|\sqrt{\pi}\psi\rangle. && (4.32)
\end{aligned}$$

Now,

$$\|\sqrt{\pi}\psi\|_2^2 - \langle\sqrt{\pi}\psi|D(P)|\sqrt{\pi}\psi\rangle \geq \omega(\mathbb{E}_\pi(\psi^2\ln\Big(\psi^2\Big)) - \mathbb{E}_\pi(\psi^2\ln\Big(\mathbb{E}\psi^2\Big))), \tag{4.33}$$

and

$$\mathbb{E}_\pi(\psi^2\ln\Big(\psi^2\Big)) - \mathbb{E}_\pi(\psi^2\ln\Big(\mathbb{E}\psi^2\Big)) = \sum_{x\in\mathcal{X}} \pi(x)\psi^2(x)\ln\Big(\psi^2(x)\Big) - \|\sqrt{\pi}\psi\|_2^2\ln\Big(\|\sqrt{\pi}\psi\|_2^2\Big). \tag{4.34}$$

Consider $\psi = \frac{\psi'}{\sqrt{\pi}}$ with $\|\psi'\|_2 = 1$. Then

$$1 - \langle\psi|D(P)|\psi\rangle \geq \omega\sum_{x\in\mathcal{X}} \psi^2(x)\ln\left(\frac{\psi(x)^2}{\pi(x)}\right) = \omega\mathrm{KL}(\psi^2||\pi). \tag{4.35}$$

□

Theorem 4.4.15 (Sufficient b for spectral density to imply short path)**.** *Suppose $\mathcal{M} = (\mathcal{X}, P, \pi)$ is a reversible Markov Chain that satisfies an ω log-Sobolev inequality. If*

$$b < \frac{2}{3}\gamma\omega \ln\left(\frac{1}{\pi(E^*)}\right) \tag{4.36}$$

then γ spectral density implies an $\frac{\omega}{2}$ short-path condition.

Proof. Let $F(E)$ be the cumulative distribution function for the cost function H. Define $F_\eta(E)$ to be the cumulative distribution function for $g_\eta\left(\frac{H}{|E^*|}\right)$. Then,

$$F_\eta(v) = \begin{cases} 0 & v < -1 \\ F(E^*(1-\eta-\eta v)) \leq (\pi(E^*))^\gamma & -1 \leq v < 0 \\ 1 & v \geq 0 \end{cases} \tag{4.37}$$

where we assume that the probability of low energy states under π is low, i.e.,

$$F((1-\eta)E^*) = \mathbb{P}_\pi(E \leq (1-\eta)E^*) \leq (\pi(E^*))^\gamma \tag{4.38}$$

for $0 < \gamma \leq 1$ where $\pi(E^*)$ is given by

$$\pi(E^*) := \mathbb{P}_\pi(E = E^*) = \sum_{\{x \in \mathcal{X} : E(x) = E^*\}} \pi(x). \tag{4.39}$$

If this bound does not hold, then for a very small η, there is a high probability mass for the states with energy closer to E^*. Therefore, we can find an approximate optimizer by randomly sampling from $\pi(x)$ in time sub-exponential in $\log(1/\pi(E^*))$. Note that when π is uniform and the ground state is non-degenerate, we recover the original condition of [DPCB23, see, Lemma 5] since

$$F((1-\eta)E^*) \leq 2^{-\gamma n}. \tag{4.40}$$

For a quantum state $|\psi\rangle$ let ψ^2 denote its ℓ_2 distribution in the computational basis. Applying Donsker and Varadhan's variational formula [DV83] for $\mathrm{KL}(\psi^2 \| \pi)$, we may write

$$\mathrm{KL}(\psi^2 \| \pi) = \sup_f \{\mathbb{E}_{\psi^2}[f(x)] - \ln(\mathbb{E}_\pi[\exp(f(x))])\}. \tag{4.41}$$

Choosing $f(x) = -\gamma \ln\left(\frac{1}{\pi^*}\right) g_\eta\left(\frac{H(x)}{E^*}\right)$ and defining $U_\psi = \mathbb{E}_{\psi^2}[g_\eta(\frac{H}{E^*})]$, it follows

$$\mathrm{KL}(\psi^2\|\pi) \geq -\gamma \ln\left(\frac{1}{\pi^*}\right) \mathbb{E}_{\psi^2}\left[g_\eta\left(\frac{H(x)}{E^*}\right)\right] - \ln\left(\mathbb{E}_\pi\left[e^{-\gamma \ln\left(\frac{1}{\pi^*}\right) g_\eta\left(\frac{H(x)}{E^*}\right)}\right]\right) \tag{4.42}$$

$$\geq -\gamma \ln\left(\frac{1}{\pi^*}\right) U_\psi - 1. \tag{4.43}$$

The final inequality follows from

$$\mathbb{E}_\pi\left[e^{-\gamma \ln\left(\frac{1}{\pi^*}\right) g_\eta\left(\frac{H(x)}{E^*}\right)}\right] = \sum_{g_\eta\left(\frac{H(x)}{E^*}\right))=0} \pi(x) + \sum_{g_\eta\left(\frac{H(x)}{E^*}\right)\neq 0} \pi(x) e^{-\gamma \ln\left(\frac{1}{\pi^*}\right) g_\eta\left(\frac{H(x)}{E^*}\right)} \tag{4.44}$$

$$\leq \pi(E(x)=0) + \pi(E(x)\neq 0) e^{\gamma \ln\left(\frac{1}{\pi^*}\right)} \tag{4.45}$$

$$= \pi(E(x)=0) + (\pi^*)^\gamma\, e^{\gamma \ln\left(\frac{1}{\pi^*}\right)} \tag{4.46}$$

$$\leq 2, \tag{4.47}$$

as we assume $\pi(g_\eta(\frac{H}{E^*}) \neq 0) \leq (\pi^*)^\gamma$ and $g_\eta(\frac{H}{E^*}) \geq -1$.

Thus we have the following lower bound on the KL divergence:

$$\mathrm{KL}(\psi^2\|\pi) \geq -\gamma \ln\left(\frac{1}{\pi^*}\right) U_\psi - 1. \tag{4.48}$$

We also have the following upper bound the KL divergence from Lemma 4.4.14:

$$\frac{1 - \langle\psi|D(P)|\psi\rangle}{\omega} \geq \mathrm{KL}(\psi^2\|\pi), \tag{4.49}$$

where ω is the LS constant of P.

The following argument attempts to find an upper bound b^* on the b, such that for all $b < b^*$ the two bounds above become contradicting if short-path is not satisfied.

Suppose for contradiction that the $\frac{\omega}{2}$ short-path condition is violated at b, i.e.

$$\mathsf{GSE}(\Pi_\perp H_b \Pi_\perp) < -1 + \frac{\omega}{2}, \tag{4.50}$$

where ω is the log-Sobolev constant of P. If $|\psi_b'\rangle$ is the ground state of $\Pi_\perp H_b \Pi_\perp$, then

$$-1 + \frac{\omega}{2} > \langle\psi_b'|H_b|\psi_b'\rangle = -\langle\psi_b'|D(P)|\psi_b'\rangle + b\langle\psi_b'|G_\eta|\psi_b'\rangle = -\langle\psi_b'|D(P)|\psi_b'\rangle + bU_{\psi_b'}, \tag{4.51}$$

which implies

$$1 - \langle \psi_b' | D(P) | \psi_b' \rangle < \frac{\omega - 2bU_{\psi_b'}}{2}. \tag{4.52}$$

Thus if short-path is violated, then the KL upper bound above reduces to

$$\frac{\omega - 2bU_{\psi_b'}}{2\omega} \geq \mathrm{KL}(\psi_b'^2 \| \pi). \tag{4.53}$$

We also have generally that a Poincaré inequality (LS inequality implies Poincaré) implies that for any $|\psi\rangle$

$$\langle \psi | - D(P) | \psi \rangle \geq -1 + \delta \geq -1 + \omega, \tag{4.54}$$

where δ denotes the Poincaré constant of P, so if short-path is violated combining the above with Equation (4.51) gives

$$\frac{\omega}{2b} < -U_{\psi_b'}, \tag{4.55}$$

where $U_{\psi_b'} < 0$ by construction. Also, by construction $-U_{\psi_b'} \leq 1$.

Hence, when the short path condition is violated, Equations (4.48) and (4.49) imply that

$$0 \geq -\gamma \ln \left(\frac{1}{\pi(E^*)} \right) U_{\psi_b'} - 1 - \frac{\omega - 2bU_{\psi_b'}}{2\omega}. \tag{4.56}$$

As mentioned earlier, we want to a range of b's that contradicts this inequality, and so we solve

$$0 < -\gamma \ln \left(\frac{1}{\pi(E^*)} \right) U_{\psi_b} - 1 - \frac{\omega - 2bU_{\psi_b'}}{2\omega}, \tag{4.57}$$

which yields

$$b < \frac{3\omega}{2U_{\psi_b'}} + 2\gamma\omega \ln(1/\pi(E^*)) \tag{4.58}$$

and using Equation (4.55) since we want to consider the smallest right hand side:

$$b < \frac{2}{3}\gamma\omega \ln \left(\frac{1}{\pi(E^*)} \right). \tag{4.59}$$

Thus, the short path condition must hold for the values of b given in theorem statement.

□

It is reasonable to question if a simpler Poincaré inequality for $\mathcal{M} = (\mathcal{X}, P, \pi)$ which only depends on the spectral gap δ of P would suffice to get a bound on b. Unfortunately, it appears that this does not provide a useful bound unless δ is constant.

Lemma 4.4.16. *Suppose $\mathcal{M} = (\mathcal{X}, P, \pi)$ is a Markov chain that satisfies a Poincaré inequality with constant δ. Then, for all quantum states $|\psi\rangle$ one has*

$$\frac{1 - \langle\psi|D(P)|\psi\rangle}{\delta} \geq [\mathrm{TV}(\psi^2, \pi)]^2, \tag{4.60}$$

where $\mathrm{TV}(\cdot,\cdot)$ denotes the total variation distance, and ψ^2 is the ℓ_2 distribution of $|\psi\rangle$ in the computational basis.

Proof. The proof roughly follows that of Lemma 4.4.14, i.e. using Equation (4.29), but instead uses π-Variance. From Poincaré:

$$\frac{1 - \langle\psi|D(P)|\psi\rangle}{\delta} = \frac{\mathcal{D}(\psi/\sqrt{\pi}, \psi/\sqrt{\pi})}{\delta} \geq \mathrm{Var}_\pi[\psi/\sqrt{\pi}] = \mathbb{E}_\pi[\psi^2/\pi] - (\mathbb{E}_\pi[\psi/\sqrt{\pi}])^2 \tag{4.61}$$

$$= 1 - |\langle\sqrt{\pi}|\psi\rangle|^2 \tag{4.62}$$

$$\geq [\mathrm{TV}(\pi, \psi^2)]^2. \tag{4.63}$$

□

Note the following does not make use of the spectral density condition, which is one reason for the weak upper bound on b. However, it does suffice for P with constant spectral gaps.

Theorem 4.4.17 (Sufficient b for short-path under Poincaré inequality)**.** *Suppose $\mathcal{M} = (\mathcal{X}, P, \pi)$ is a reversible Markov chain that satisfies a Poincaré inequality with constant δ. If*

$$b < \delta \frac{4\sqrt{6} - 1}{10}, \tag{4.64}$$

then, the $\frac{\delta}{2}$ short-path condition is satisfied.

Proof. The proof follows a similar structure as Theorem 4.4.15. We have the variational definition of TV:

$$\mathrm{TV}(\pi, \psi) = \sup_{f:\|f\|\leq 1} \frac{1}{2}(\mathbb{E}_{\psi^2}[f(x)] - \mathbb{E}_\pi[f(x)]). \tag{4.65}$$

We can choose $f(x) = -g_\eta(H(x)/E^*)$, which satisfies $\|f\| \leq 1$, so

$$\mathrm{TV}(\pi, \psi^2) \geq \frac{1}{2}(-U_\psi - 1), \tag{4.66}$$

since $\mathbb{E}_\pi[f(x)] \leq 1$. Note $U_\psi = \mathbb{E}_{\psi^2}[f(x)]$.

Thus, we have the following lower bound:

$$[\mathrm{TV}(\pi, \psi^2)]^2 \geq \frac{1}{4}(U_\psi + 1)^2. \tag{4.67}$$

We also have the following upper bound on the TV from Lemma 4.4.16:

$$\frac{1 - \langle\psi|D(P)|\psi\rangle}{\delta} \geq [\mathrm{TV}(\psi^2\|\pi)]^2, \tag{4.68}$$

where δ is the Poincaré constant of P.

Suppose for contradiction that the $\frac{\delta}{2}$ short-path condition is violated at b, i.e.

$$\mathsf{GSE}(\Pi_\perp H_b \Pi_\perp) < -1 + \frac{\delta}{2}, \tag{4.69}$$

where ω is the log-Sobolev constant of P. Thus, if $|\psi_b'\rangle$ is the ground state of $\Pi_\perp H_b \Pi_\perp$, then

$$-1 + \frac{\delta}{2} > \langle\psi_b'|H_b|\psi_b'\rangle = -\langle\psi_b'|D(P)|\psi_b'\rangle + b\langle\psi_b'|G_\eta|\psi_b'\rangle = -\langle\psi_b'|D(P)|\psi_b'\rangle + bU_{\psi_b'}, \tag{4.70}$$

which implies

$$1 - \langle\psi_b'|D(P)|\psi_b'\rangle < \frac{\delta - 2bU_{\psi_b'}}{2}. \tag{4.71}$$

Thus if short-path is violated, then the KL upper bound above reduces to

$$\frac{\delta - 2bU_{\psi_b'}}{2\delta} \geq [\mathrm{TV}(\psi_b'^2, \pi)]^2. \tag{4.72}$$

If P satisfies a δ Poincaré inequality, then we have that for any $|\psi\rangle$

$$\langle\psi| - D(P)|\psi\rangle \geq -1 + \delta, \tag{4.73}$$

so if short-path is violated combining the above with Equation (4.70) gives

$$\frac{\delta}{2b} < -U_{\psi_b'}, \tag{4.74}$$

where $U_{\psi'_b} < 0$ by construction. Also, by construction $-U_{\psi'_b} \leq 1$.

Hence, when the short path condition is violated, Equations (4.67) and (4.68) imply that

$$0 \geq \frac{1}{4}(U_{\psi'_b} + 1)^2 - \frac{\delta - 2bU_{\psi'_b}}{2\delta}. \tag{4.75}$$

As mentioned earlier, we want to a range of b's that contradicts this, so we solve

$$0 < \frac{1}{4}(U_{\psi'_b} + 1)^2 - \frac{\delta - 2bU_{\psi'_b}}{2\delta}, \tag{4.76}$$

so

$$0 \leq \delta U_{\psi'_b}^2 + (2\delta + 4b)U_{\psi'_b} - 3\delta \tag{4.77}$$

and using Equation (4.74) since we want to consider the smallest r.h.s.:

$$0 \leq \delta^2 - 4\delta b - 20b^2, \tag{4.78}$$

so using the positive root

$$b < \delta\frac{4\sqrt{6} - 1}{10}. \tag{4.79}$$

Thus, the short-path condition must hold for the values of b given in theorem statement. □

It is also natural to ask if a modified log-Sobolev inequality would work. The apparent issue is with relating the Dirichlet form to the energy with respect to $D(P)$. Although we could not prove a generic bound in terms of modified log-Sobolev constant, we can use it to derive a lower bound on standard Log Sobolev constant by using the following theorem.

Theorem 4.4.18 (Theorem 1 in [Sal21]). *Let ω_{LSI} and ω_{MLSI} respectively denote the log-Sobolev constant and the modified log-Sobolev constant of a Markov chain $\mathcal{M} = (\mathcal{X}, P, \pi)$. If $\mathcal{M}$ is reversible, then*

$$\omega_{\mathrm{LSI}} \geq \omega_{\mathrm{MLSI}}/\log(1/p) \tag{4.80}$$

where p is the smallest non-zero element in P.

Note that for general Markov chains p can be exponentially small. However, in

certain cases such as single site Glauber dynamics on bounded degree graphs, p is only polynomially small. Therefore, in these special cases, our main theorem implies a super quadratic speed up although the speedup term c might be falling with n.

4.4.2.1.1 Properties of the Short Jump We now show some properties of the ground state $|\psi_b\rangle$ that is obtained as a result of the short jump. In the next section, we will use an approximation ground state projector to bound the runtime of the long and short jumps together (as in [DPCB23]). It is instructive however, to bound the runtime of the short jump alone and confirm that it indeed takes only polynomial time. In the following, we show that if we select b such that a θ-short path condition is satisfied, the time taken for the short jump is in fact $\mathcal{O}(\delta^{-1})$ where δ is the spectral gap of P. This analysis supports the discussion of the algorithm in Section 4.3. We first establish a technical condition on the change in ground state energy. An analogous bound is used in [DPCB23] to bound the runtime and while we will do the same, it is convenient to introduce it here as we establish further properties of $|\psi_b\rangle$.

Lemma 4.4.19 (Ground-state energy shift bound, adapted from Proposition 6 of [DPCB23])**.** *Suppose that γ-spectral density holds and the $\frac{\theta}{2}$-short-path condition holds at b, then*

$$|E_b| < 1 + \frac{4(\pi(E^*))^\gamma}{\theta}. \tag{4.81}$$

The above lemma utilizes the following expression for E_b:

$$E_b = \langle\psi_b|H_b|\psi_b\rangle = -1 - b\langle\sqrt{\pi}|G_\eta|\sqrt{\pi}\rangle - b^2\langle\sqrt{\pi}|G_\eta W_b G_\eta|\sqrt{\pi}\rangle, \tag{4.82}$$

where

$$W_b := (\Pi_\perp(H_b - E_b)\Pi_\perp)^{-1}. \tag{4.83}$$

The main purpose of this lemma is for determining the cost of the long-jump, which we do in the next section. Specifically, we will want to bound the magnitude of the energy shift away from two. We provide intuition why doing so does not require a strong (θ being inverse-polynomial in n) short-path condition. In the above lemma, the short-path condition is used to bound the spectral norm of W_b, where a θ short-path condition implies $\|W_b\|_2 \leq \theta^{-1}$. However, the actual quantity of interest is $\langle\sqrt{\pi}|G_\eta W_b G_\eta|\sqrt{\pi}\rangle$, where $\|G_\eta|\sqrt{\pi}\rangle\|_2 \leq \pi(E^*)^\gamma$. For "hard" problems $\pi(E^*)^\gamma$ will be exponentially small in

the problem size, and so $\|W_b\|$ can be even exponentially large for a small ground-state energy shift. Thus, for such problems, the short-path condition we need for a good gap (θ being inverse-polynomial in n) is stronger than the short-path condition needed for a small-enough energy shift.

If we do have an inverse-polynomial short-path condition, then the above lemma places strong conditions on the quantum state obtained by the short jump, namely that its trace distance from the starting state is small whenever hard to solve exactly by sampling from π. Specifically, we have the following lemma

Lemma 4.4.20. *Suppose that the γ spectral density and $\frac{\delta}{2}$-short path condition hold for some constant b. It holds that $\left\||\sqrt{\pi}\rangle\langle\sqrt{\pi}| - |\psi_b\rangle\langle\psi_b|\right\|_{\mathrm{Tr}} = \mathcal{O}\left(\delta^{-1}(\pi(E^*))^{\gamma/2}\right)$ where δ is the spectral gap of P. Consequently, if $\delta = \Omega\left(\frac{1}{\mathrm{poly}(n)}\right)$, $\left\||\sqrt{\pi}\rangle\langle\sqrt{\pi}| - |\psi_b\rangle\langle\psi_b|\right\|_{\mathrm{Tr}} = \mathcal{O}^*\left((\pi(E^*))^{\gamma/2}\right)$.*

Proof. Note that the ground state energy of H_b is ≤ -1 since $H_b - (-D(P))$ is negative definite. It follows from Lemma 4.4.19 and the conditions of this lemma that $-1 - \mathcal{O}^*(\pi(E^*))^{\gamma}) \leq E_b \leq -1$. We observe that

$$0 \leq -\langle\sqrt{\pi}|D(P)|\sqrt{\pi}\rangle - \langle\psi_b|H_b|\psi_b\rangle = \mathcal{O}(\delta^{-1}\pi(E^*)^{\gamma}), \tag{4.84}$$

$$\implies \langle\psi_b|D(P)|\psi_b\rangle - \langle\sqrt{\pi}|D(P)|\sqrt{\pi}\rangle = \mathcal{O}(\delta^{-1}\pi(E^*)^{\gamma}) + b\langle\psi_b|G_\eta|\psi_b\rangle \tag{4.85}$$

$$\implies |\langle\psi_b|D(P)|\psi_b\rangle - \langle\sqrt{\pi}|D(P)|\sqrt{\pi}\rangle| = \mathcal{O}(\delta^{-1}(\pi(E^*)^{\gamma}), \tag{4.86}$$

where the last equality follows from the negative semi-definiteness of G_η.

We may write the state $|\psi_b\rangle$ as $\alpha|\sqrt{\pi}\rangle + \sqrt{1-\alpha^2}|\sqrt{\pi}^{\perp}\rangle$ where $|\pi^{\perp}\rangle$ is a quantum state such that $\langle\sqrt{\pi}^{\perp}|\sqrt{\pi}\rangle = 0$ (We may take α to be real and in the interval $[0,1]$ since $|\psi_b\rangle, |\sqrt{\pi}\rangle$ are the ground states of stoquastic Hermitian matrices). It follows from the definition of trace distance that $\left\||\sqrt{\pi}\rangle\langle\sqrt{\pi}| - |\psi_b\rangle\langle\psi_b|\right\|_{\mathrm{Tr}} = \sqrt{1-\alpha^2}$. From the definition of spectral gap, and observing that $\sqrt{\pi}$ is the ground state of $-D(P)$, we have

$$|\langle\psi_b|D(P)|\psi_b\rangle - \langle\sqrt{\pi}|D(P)|\sqrt{\pi}\rangle| \tag{4.87}$$

$$=|(1-\alpha^2)\left(\langle\sqrt{\pi}^{\perp}|D(P)|\sqrt{\pi}^{\perp}\rangle - \langle\sqrt{\pi}|D(P)|\sqrt{\pi}\rangle\right)| \geq \delta(1-\alpha^2). \tag{4.88}$$

From our two bounds on $|\langle\psi_b|D(P)|\psi_b\rangle - \langle\sqrt{\pi}|D(P)|\sqrt{\pi}\rangle|$, it follows that

$$\left\||\sqrt{\pi}\rangle\langle\sqrt{\pi}| - |\psi_b\rangle\langle\psi_b|\right\|_{\mathrm{Tr}} = \sqrt{1-\alpha^2} = \mathcal{O}\left(\delta^{-1}(\pi(E^*))^{\gamma/2}\right) = \mathcal{O}^*\left(\pi(E^*))^{\gamma/2}\right), \tag{4.89}$$

which completes the proof. □

Since $|\sqrt{\pi}\rangle$ and $|\psi_b\rangle$ are pure states, we obtain a bound on the overlap between the states as an immediate consequence of the above.

Corollary 4.4.21. *Suppose that the γ spectral density and $\frac{\delta}{2}$-short path condition hold for some constant b with $\delta = \Omega\left(\frac{1}{\mathrm{poly}(n)}\right)$, where δ is the spectral gap of P. Then, $|\langle\psi_b|\sqrt{\pi}\rangle| = 1 - \mathcal{O}^*((\pi(E^*)^\gamma)$.*

It is also clear from the above that if $\pi(E^*)^{-1}$ is super-polynomial in n (as is the case for the problems considered here), the trace distance between $|\psi\rangle$ and $|\psi_b\rangle$ decays super-polynomially in n. From the operational definition of trace distance, the difference in probability of obtaining a specific outcome when performing a fixed measurement on two states ρ_1, ρ_2 is upper bounded by their trace distance. Since the trace distance is monotonic under discarding of identical subsystems, it also holds for all $m \in \mathbb{N}$ that $\left\|\rho_1^{\otimes m} - \rho_2^{\otimes m}\right\|_{\mathrm{Tr}} \leq m\|\rho_1 - \rho_2\|_{\mathrm{Tr}}$. As a consequence of the above considerations, and the monotonicity of trace distance we have the following corollary, which illustrates that the short jump by itself does not offer any advantage if used in a polynomial time approximation algorithm.

Corollary 4.4.22. *Suppose that the conditions of Lemma 4.4.20 are satisfied, and let $\mathcal{A}$ be any non-adaptive quantum or classical algorithm applied to* $\mathrm{poly}(n)$ *copies of an input quantum state. Each outcome of $\mathcal{A}$ that is observed with probability p_1 when the algorithm is applied to $|\psi_b\rangle$, must be observed with probability $p_2 = p_1 \pm \mathcal{O}^*((\pi(E^*)^\gamma)$.*

4.4.2.2 The Long Jump

The long jump is the preparation of an optimal solution in Π^* from $|\psi_b\rangle$, where ideally $\|\Pi^*|\psi_b\rangle\|_2^2 \geq \pi(E^*)^{1-c}$. As discussed in the previous subsection, the only quantum speedup from this step is quadratic and is due to amplitude amplification. That is, the quantum short-jump plus classical sampling would cost $\mathcal{O}^*(\pi(E^*)^{-(1-c)})$, provided that the overlap condition just mentioned is satisfied. Accordingly, a quantum short-jump applied to the discriminant of the Markov Chain in Markov Chain search could still provide a nontrivial speedup. The goal of this subsection is to determine conditions on the cost Hamiltonian in terms of the Markov Chain $\mathcal{M}$ under which $\|\Pi^*|\psi_b\rangle\|_2^2 \geq \pi(E^*)^{1-c}$ holds for some constant c. If we combine with amplitude amplification we get the runtime stated for the generalized short-path algorithm.

In order to bound the terms in the runtime involving the ground state projector $|\psi_b\rangle\langle\psi_b|$, we follow [DPCB23] in approximating $|\psi_b\rangle\langle\psi_b|$ through the use of a simple

degree-ℓ polynomial $\mathcal{P}_\ell$. This quantity is related to the runtime via

$$|\langle\sqrt{\pi}|\psi_b\rangle|^{-1} + \|\Pi^*|\psi_b\rangle\|_2^{-1} \leq 2\|\langle\sqrt{\pi}|\psi_b\rangle\langle\psi_b|\Pi^*\|_2^{-1}. \tag{4.90}$$

Let E_b denote the ground state energy of H_b.

Lemma 4.4.23 (Runtime bound by approximate projector)**.** *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a Markov chain with $|\mathcal{X}| = V$. Define*

$$\mathcal{P}_\ell := \left(\frac{H_b}{|E_b|}\right)^\ell, \tag{4.91}$$

and let ω be the log-Sobolev constant of $D(P)$. For either all even or all odd ℓ we have

$$\left\langle\sqrt{\pi}\middle|\mathcal{P}_\ell\middle|z\right\rangle - V\left(1 - \frac{\omega}{2}\right)^\ell < \langle\sqrt{\pi}|\psi_b\rangle\,\langle\psi_b|z\rangle\,, \tag{4.92}$$

for any assignment z. Note the same also holds with ω replaced by the spectral gap δ.

We now recall the definition of α-subdepolarizing from [DPCB23] but generalized to arbitrary Markov Chains.

Definition 4.4.24 (α_P-subdepolarizing)**.** Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a Markov chain. The pair $(\mathcal{M}, H)$ satisfies the α-subdepolarizing property if for $f(x) = -g_\eta(-x)$ (as defined earlier), the following holds for any set of constants $0 < c_1, \ldots, c_T < 1, \forall T \in \mathbb{N}$:

$$\mathbb{E}_{\substack{y\sim x\\ P}} \prod_{t=1}^{T} f\left(\frac{c_t H(y)}{E^*}\right) \geq \prod_{t=1}^{T} f\left(\frac{c_t(1-\alpha_P)H(x)}{E^*}\right), \tag{4.93}$$

where E^* is the ground state energy of H.

As shown in the proof of Proposition 3 of [DPCB23], if $\Delta_P(\eta)$-stability holds, then α_P-subdepolarizing is satisfied. However, the converse also holds.

Lemma 4.4.25 ($\Delta_P(\eta)$ stable $\iff$ α_P subdepolarizing)**.** *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a Markov chain. The pair $(\mathcal{M}, H)$ satisfies the $\Delta_P(\eta)$-stable if and only if it satisfies α_P-subdepolarizing. They are related by the equation $\alpha_P = \frac{\Delta_P(\eta)}{|E^*|(1-\eta)}$.*

We now briefly remark on some useful upper bounds on $\Delta_P(\eta)$ that we alluded to earlier. However, using too loose of an upper bound may result in the runtime analysis not indicating a speedup. Note that choice of the bound on $\Delta_P(\eta)$ is not actually used by Algorithm 9.

Lemma 4.4.26 (Upper bounds on $\Delta(\eta)$)**.** *If H has P pseudo-Lipschitz norm $\|H\|_P$, then for $\eta \in [0,1)$,*

$$\sqrt{\|H\|_P} \geq \Delta_P(\eta). \tag{4.94}$$

Furthermore, if

$$\mathbb{E}_{y \underset{P}{\sim} x}[H(y)] \leq H(x) + \tilde{\Delta}_P, \tag{4.95}$$

then for $\eta \in [0,1)$

$$\tilde{\Delta}_P \geq \Delta_P(\eta). \tag{4.96}$$

Our next result generalizes [DPCB23, Lemma 3], which uses α_P subdepolarizing (or equivalently $\Delta_P(\eta)$) to lower bound $|\langle\sqrt{\pi}|\mathcal{P}_\ell|z^*\rangle|$.

Lemma 4.4.27 (Overlap with general Markov chain)**.** *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a Markov chain. Given positive parameters $\eta < 1$, $b < 1$, $\alpha < (1-b)/2$, and integer ℓ, suppose that (H, g_η) has the α-subdepolarizing property, $3/\alpha^2 \leq \ell = \mathcal{O}\left(\frac{\omega}{(\pi(E^*))^\gamma}\right)$, and that H_b satisfies the small-ground-energy shift condition. Define the function $F : [0,1] \mapsto [0,1]$ as $F(x) := 1 - x + x\ln(x)$. Let $z^* \in \{-1,+1\}^n$ be an optimal assignment, i.e. $H(z^*) = E^*$. Then,*

$$\langle\pi|\mathcal{P}_\ell|z^*\rangle \geq \pi(z^*)^{1/2} \exp\left(\frac{b}{\eta\alpha} F\left(1-\eta\right)\right)(e^{-1} - 2e^{-2}). \tag{4.97}$$

Proof. Define A, B, and f by the following equations:

$$A = D(P) \tag{4.98}$$

$$B = -bg_\eta\left(\frac{H}{E^*}\right) = bf\left(\frac{H}{E^*}\right), \tag{4.99}$$

so

$$\langle y|A|x\rangle = \pi^{1/2}(x)P(x,y)\pi^{-1/2}(y). \tag{4.100}$$

Then, the approximate projector can be written as

$$\mathcal{P}_\ell = \frac{(-A-B)^\ell}{|E_b|^\ell}. \tag{4.101}$$

By Lemma 4.81, if $\ell = \mathcal{O}\left(\frac{\omega}{(\pi(E^*))^\gamma}\right)$, then we can take $|E_b|^\ell = \Theta(1)$ and ignore the denominator. To compute the numerator, start with

$$\langle \pi | A | z^* \rangle = \langle \pi | z^* \rangle = \pi(z^*)^{1/2} \tag{4.102}$$

$$\langle \pi | B | z^* \rangle = b\pi(z^*)^{1/2}. \tag{4.103}$$

Next, we compute

$$\langle \pi | BA^k | z^* \rangle = b\langle \pi | BD^k | z \rangle = b \sum_{x_1 \dots x_k} \langle \pi | B | x_k \rangle \langle x_k | D | x_{k-1} \rangle \cdots \langle x_1 | D | z \rangle \tag{4.104}$$

$$= b \sum_{x_1 \dots x_k} \langle \pi | B | x_k \rangle \pi(x_k)^{-1/2} P(x_{k-1}, x_k) \pi(x_{k-1})^{1/2} \cdots \pi(x_1)^{-1/2} P(z^*, x_1) \pi(z^*)^{1/2} \tag{4.105}$$

$$= b \sum_{x_1 \dots x_k} f\left(\frac{H(x_k)}{E^*}\right) P(z^*, x_1) \cdots P(x_{k-1}, x_k) \pi(z^*)^{1/2} \tag{4.106}$$

$$= b\pi(z^*)^{1/2} \mathbb{E}_{x_1 \sim z} \cdots \mathbb{E}_{x_k \sim x_{k-1}} f\left(\frac{H(x_k)}{E^*}\right) \tag{4.107}$$

$$\geq b\pi(z^*)^{1/2} f\left((1-\alpha)^k\right). \tag{4.108}$$

In general, we can write any string of A's and B's as

$$\dots AB^{c_3} AB^{c_2} AB^{c_1} AB^{c_0}, \tag{4.109}$$

for $c \in \ell_1(\mathbb{N}^\infty)$, i.e., finite sequences of natural numbers of unbounded length. Let $\tilde{f}(x) := f(H(x)/E^*)$. Accordingly, we can compute

$$\langle \pi | \dots AB^{c_3} AB^{c_2} AB^{c_1} AB^{c_0} | z^* \rangle \tag{4.110}$$

$$= \sum_{x_1, \dots} \dots \langle x_4 | AB^{c_3} | x_3 \rangle \langle x_3 | AB^{c_2} | x_2 \rangle \langle x_2 | AB^{c_1} | x_1 \rangle \langle x_1 | AB^{c_0} | z^* \rangle \tag{4.111}$$

$$= \sum_{x_1, \dots} b^{\sum_j x_j} \dots \langle x_2 | D | x_1 \rangle f(H(x_1)/E^*)^{c_1} \langle x_1 | D | z^* \rangle \tag{4.112}$$

$$= \pi(z^*)^{1/2} b^{\sum_{j=0}^\infty c_j} \sum_{x_1, \dots} \dots P(x_1, x_2) f(H(x_1)/E^*)^{c_1} P(x_1, z^*) \tag{4.113}$$

$$= \pi(z^*)^{1/2} b^{\sum_{j=0}^\infty c_j} \mathbb{E}_{z^* \underset{P}{\sim} x_1} [(\tilde{f}(x_1))^{c_1} \mathbb{E}_{x_1 \underset{P}{\sim} x_2} [(\tilde{f}(x_2))^{c_2} \cdots]] \tag{4.114}$$

$$\geq \pi(z^*)^{1/2} b^{\sum_{j=0}^\infty c_j} \prod_{j=0}^\infty f((1-\alpha)^j)^{c_j}. \tag{4.115}$$

By assumption, $b^{\sum_{j=0}^\infty c_j}$ is finite.

We only pick up $\pi(z^*)^{1/2}$ term instead of $2^{-n/2}$ in front of the product. Therefore Propositions 15 and 16 of [DPCB23] hold. Hence, we obtain the stated result. □

The following uses the above results to bound the complexity of the short and long jumps. As mentioned earlier, the runtimes of the two jumps are bounded together due to the ease of analysis.

Theorem 4.4.28. *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a Markov chain, and suppose $|\mathcal{X}| = V$. If*

$$\ell \;\geq\; \max\left(\frac{3}{\alpha^2}, \max_{z^*} 4\frac{\ln\left(V/\sqrt{\pi(z^*)}\right)}{\omega}\right) \tag{4.116}$$

and $\ell = \mathcal{O}\left(\frac{\omega}{(\pi(E^*))^\gamma}\right)$, then

$$|\langle\sqrt{\pi}|\psi_b\rangle|^{-1} + \|\Pi^*|\psi_b\rangle\|_2^{-1} = \mathcal{O}\left([\pi(E^*)^{-1}]^{\left(\frac{1}{2} - \frac{b\eta}{2\ln(1/\pi(E^*))\alpha}\right)}\right). \tag{4.117}$$

A direct consequence is that the overall complexity in terms of queries to block-encodings of $D(P)$ and H is

$$\mathcal{O}\left([\min(\mathrm{Gap}(D(P)), \mathrm{Gap}(H_b))]^{-1}[\pi(E^*)^{-1}]^{\left(\frac{1}{2} - \frac{b\eta}{2\ln(1/\pi(E^*))\alpha}\right)}\right). \tag{4.118}$$

Proof. By Lemmas 4.4.23 and 4.4.19, for any optimal assignment z^* we have

$$\langle\sqrt{\pi}|\psi_b\rangle\,\langle\psi_b|z^*\rangle \geq \pi(z^*)^{1/2}\exp\left(\frac{b}{\eta\alpha}F\left(1-\eta\right)\right)(e^{-1} - 2e^{-2}) - Ve^{-\omega\ell/2}. \tag{4.119}$$

Consider $\ell \geq \max_{z^*} 4\frac{\ln\left(V/\sqrt{\pi(z^*}\right))}{\omega}$, where the max is over optimal assignments, then

$$\langle\sqrt{\pi}|\psi_b\rangle\,\langle\psi_b|z^*\rangle \geq \pi(z^*)^{1/2}\exp\left(\frac{b}{\eta\alpha}F\left(1-\eta\right)\right)(e^{-1} - 2e^{-2}) - \pi(z^*). \tag{4.120}$$

Note that $\exp\left(\frac{b}{\eta\alpha}F\left(1-\eta\right)\right) > 1$, so the first term on the right hand side will dominate asymptotically. In fact $\frac{F(1-\eta)}{\eta} \geq \eta/2$, so we have

$$\langle\sqrt{\pi}|\psi_b\rangle\,\langle\psi_b|z^*\rangle \geq \Omega\left(\sqrt{\pi(z^*)\exp\left(\frac{b\eta}{\alpha}\right)}\right). \tag{4.121}$$

This clearly gives that

$$\|\langle\sqrt{\pi}|\psi_b\rangle\langle\psi_b|\Pi^*\|_2^{-1} \geq \Omega\left(\sqrt{\pi(E^*)\exp\left(\frac{b\eta}{\alpha}\right)}\right). \tag{4.122}$$

The result follows by using Equation (4.90). □

It may not be immediately obvious that the conditions on ℓ are not contradicting. Here, we give intuition for why this is not the case for the typical applications of the algorithm. Note for an efficient algorithm, at the very least we will need $\omega^{-1} = \mathcal{O}(\text{poly}(n))$. For hard problems, $\pi(E^*)$ will be exponentially small in n, thus the $\mathcal{O}\left(\min_{z^*} \frac{\omega}{(\pi(z^*))^\gamma}\right)$ upper bound on ℓ will be significantly larger than the lower bound. Note $1/\alpha^2$ is significantly smaller than $1/\pi(E^*)$ for a hard problem, for showing super-Grover runtime we will want $\alpha = \Theta(\frac{1}{\ln(1/\pi(E^*))})$ anyways. Thus it is fine to assume we have the conditions on ℓ stated in Theorem 4.4.28 in settings where the algorithm can successfully be applied.

As should be apparent from Theorem 4.4.3, at a constant b the existence of a speedup over Grover is determined solely by $\frac{\Delta_P}{|E^*|}$. Clearly, we must have that $\frac{\Delta_P}{|E^*|} = \Omega\left(\frac{1}{\ln(1/\pi(E^*))}\right)$, at least for a problem with at least exponential runtime. If this does not hold, then the runtime goes to zero asymptotically, an absurdity. However, it has not been *directly* shown that the derived runtime cannot lead to this contradiction. To put one's mind at ease, we present the following result.

Lemma 4.4.29. *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a Markov chain. Suppose b^* and γ in Theorem 4.4.3 are constants, and $\|H\|_2 = |E^*|$. If*

$$\mathbb{E}_{y \underset{P}{\sim} x}[H(y)] \leq H(x) + \tilde{\Delta}_P, \quad \forall x \in \mathcal{X}, \tag{4.123}$$

then $\frac{\tilde{\Delta}_P}{|E^*|} = \Omega\left(\frac{1}{\ln(1/\pi(E^*))}\right)$.

Proof. Let n be a real parameter that parametrizes the space of feasible states $\mathcal{S}(n)$ with size $|\mathcal{S}(n)| = S(n)$. Let $M(n)$ denote the mixing time of a Markov chain P with transition density π on $\mathcal{S}(n)$ such that for any $t \geq M(n)$, $\text{TV}(P^t\delta_x, \pi) \leq \frac{1}{100}$ for any $x \in \mathcal{S}(n)$.

Assume that Δ is a global upper bound, such that for all $x \in \mathcal{S}(n)$, $\mathbb{E}_{y\sim_P x}[H(y)] \leq H(x) + \Delta$. It is easy to observe that for any random variable X taking values in $\mathcal{S}(n)$, it holds that $\mathbb{E}_{Y\sim_P X}[H(Y)] \leq \mathbb{E}_X[H(X)] + \Delta$. Now let x_* be a global minimum of H (with corresponding energy E^*) and consider taking $T = \lceil M(n) \rceil$ steps of P starting from x^*, with the random state after $t \in [1, T]$ steps being denoted X_t. By induction, it

is easy to see that $E_{X_T}[H(X_T)] \leq E^* + \Delta T$. From the definition of mixing time however, it follows that $\mathbb{E}_{X_T}[H(X_T)] \geq \mathbb{E}_\pi[H(x)] - \frac{|E^*|}{100}$. Denoting $\mathbb{E}_\pi[H(x)]$ by $\bar{E}$ it follows that $\Delta \geq \frac{\bar{E}-E^*}{T} - \frac{|E^*|}{100T} = \Omega\left(\frac{|E^*|}{M(n)}\right)$. So $\frac{\tilde{\Delta}_P}{|E^*|} = \Omega\left(\frac{1}{M(n)}\right)$.

If b^* and γ are constant, then the log-Sobolev constant must satisfy $\omega = \Omega\left(\frac{1}{\ln(1/\pi(E^*))}\right)$. It follows from standard results on Markov Chains that $\omega \leq \frac{1}{M(n)}$. □

Note that in [DPCB23], the authors state an additional technical condition that is of course easy to satisfy, which is $\mathbb{E}_\pi[H(x)] = 0$. The main reason for setting $\mathbb{E}_\pi[H(x)] = 0$ is to ensure that $E^* < 0$ and for the ease of proving the tail bounds. In our setting, for arbitrary π, the expectation may need to be estimated if used as a shift. However, the shift is not necessary to run the algorithm if the $E^* < 0$ condition is already satisfied. The mean just appears as component of the runtime. Also, Corollaries 4.4.9 and 4.4.10 for the tail bounds do not assume this shift.

4.5 Applications of Generalized Short-Path Framework

4.5.1 Optimization with Fixed Hamming Weight: Transposition Walk

The k-particle Bernoulli-Laplace diffusion or Transposition Walk on n sites is a random walk on the space of Hamming weight k bistrings. For our case we will work with ± 1 strings or "spin configurations" x and define the Hamming weight $|x|$ as the number of $+1$'s. Formally, $\mathcal{X} = \{x \in \{-1, 1\}^n : |x| = k\}$. A single step of BL consists of choosing, uniformly at random, a transposition that swaps some $x_j = 1$ with another $x_i = -1$.

There is a very natural quantum Hamiltonian on n-qubits that encodes the discriminant of the transposition walk:

$$D(P) = P = \frac{1}{k(n-k)} \sum_{i<j} \frac{X_i X_j + Y_i Y_j}{2}, \tag{4.124}$$

where X_j, Y_j denote the Pauli operators applied to qubit j. This is commonly called the complete-graph XY mixer. Note that there is equality between $D(P)$ and P because the walk is symmetric. The ground state of $-D(P)$ is the uniform superposition over Hamming-weight k computational basis states, and thus encodes the stationary distribution of the transposition walk. Note that $|\sqrt{\pi}\rangle$ is just the Hamming-weight k Dicke state, which can be prepared efficiently [BE19].

We have the following log-Sobolev inequality for the transposition walk.

Theorem 4.5.1 ([LY98,Sal21, Theorem 5], discrete-time)**.** *Let P be the transition matrix for k-particle Bernoulli-Laplace diffusion on n sites and π the stationary distribution. It then holds for any real valued function ψ that*

$$\mathcal{D}(\psi,\psi) \geq \frac{n}{k(n-k)\tau_{LS}}\mathrm{Ent}(\psi^2). \tag{4.125}$$

There is also a universal constant τ_0 such that

$$\tau_{LS} \leq \tau_0 \log\left(\frac{n}{\min(k,n-k)}\right). \tag{4.126}$$

This leads to the following bound on b^* using the formula in Theorem 4.4.3.

Lemma 4.5.2. *For all k, we have the following bound on b for the transposition mixer:*

$$b^* = \frac{2C\gamma}{3}, \tag{4.127}$$

for some constant C.

Proof. We have

$$\mathcal{D}(\psi,\psi) \geq \frac{n}{k(n-k)\tau_0 \log\left(\frac{n}{\min(k,n-k)}\right)}\mathrm{Ent}(\psi^2). \tag{4.128}$$

Thus

$$b < \frac{2n\log_2\binom{n}{k}\gamma}{3k(n-k)\tau_0\log_2\left(\frac{n}{\min(k,n-k)}\right)}. \tag{4.129}$$

Using that for $k = o(n)$, $\log_2\binom{n}{k} = \Theta(k\log_2(n/k))$, and $\frac{n}{\min(k,n-k)} = \Theta(\log_2(n/k))$, we get

$$b < \frac{C2\gamma}{3}, \tag{4.130}$$

for some constant C to be determined, so b is constant for $k = o(n)$.

For $k = \Theta(n)$, we have that $\log_2\binom{n}{k} = \Theta(n)$, $\frac{n}{\min(k,n-k)} = \Theta(1)$, so b^* is also constant. □

This leads to the following simple result that follows from applying Theorem 4.4.3.

Theorem 4.5.3. *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be the k-particle transposition walk on n sites. Let $H : \{-1,1\}^n \mapsto \mathbb{R}$ be a diagonal Hamiltonian with ground state energy E^*. If*

$\|H\|_P = \mathcal{O}(1)$, *and*

$$|E^*| = \Theta\left(\ln\binom{n}{k}\right), \tag{4.131}$$

then there exists a short-path algorithm with runtime

$$\mathcal{O}^*\left(\binom{n}{k}^{\left(\frac{1}{2}-\frac{\eta(1-\eta)|E^*|b}{2\ln\binom{n}{k}\Delta_P}\right)}\right). \tag{4.132}$$

Proof. For all k we have $\omega = \Omega\left(\frac{1}{k\ln(n/k)}\right)$, and $k(\ln(n/k))\ln(1/\pi(E^*)) \asymp (\ln\binom{n}{k})^2$. Thus if $\|H\|_P = \mathcal{O}(1)$ and $|E^*| = \Theta\left(\ln\binom{n}{k}\right)$, then γ is constant so then b^* is. We also have that $\frac{|E^*|}{\Delta_P} = \mathcal{O}\left(\ln\binom{n}{k}\right)$, leading to the runtime presented. □

We apply the above result to a Hamming-weight constrained version of MaxCut over Erdős-Rényi graphs, which we call MaxCut-Hamming. One well-known special case is Hamming-weight $\frac{n}{2}$ called MaxBisection.

4.5.1.1 Hamming-weight Constrained MaxCut

Consider a graph $G(\mathcal{N}, \mathcal{E})$ with vertex set $\mathcal{N} := [n]$ and edge set $\mathcal{E}$. We assume G is drawn from the Erdős-Rényi ensemble $\mathcal{G}\left(n, \frac{p}{n-1}\right)$ for a constant p, i.e., each edge $e_{ij} \in \mathcal{E}$ for $(i,j) \in [n] \times [n]$ is created with probability $\frac{p}{n-1}$ such that G has an average degree of p. We are interested in solving the *Maximum Bisection* problem:

$$\mathcal{C}^*_{\frac{n}{2}} := \min_{x\in\{-1,1\}^n}\left\{-\frac{1}{2}\sum_{i<j} e_{ij}(1-x_ix_j) : |x| = \frac{n}{2}\right\}, \tag{MaxBisection}$$

where e_{ij} is a $\frac{p}{n-1}$ Bernoulli indicating whether the (i,j) edge is present.

For generality, we strive to present the results for an arbitrary Hamming weight constraint of size k and specify $k = \frac{n}{2}$ where necessary. We call the case where k can be arbitrary the *MaxCut Hamming* problem:

$$\mathcal{C}^*_k := \min_{x\in\{-1,1\}^n}\left\{-\frac{1}{2}\sum_{i<j} e_{ij}(1-x_ix_j) : |x| = k\right\}. \tag{MaxCut-Hamming}$$

The following result shows that $\|H\|_P = \mathcal{O}(1)$.

Lemma 4.5.4. *For the MaxCut-Hamming Hamiltonian H, the pseudo Lipschitz constant $\|H\|_P$ under the transposition walk is $\mathcal{O}(1)$ with high probability over the graph.*

The proof of the above lemma is deferred to the appendix. Next we show the existence of a tail bound for MaxBisection.

Lemma 4.5.5. *Let $\mathcal{X} = \{x \in \{-1,1\}^n : |x| = \frac{n}{2}\}$ and $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a reversible Markov chain. For* (MaxBisection) *on a graph $G \sim \mathcal{G}\left(n, \frac{p}{n-1}\right)$ and $D(P)$ being the transposition mixer, we have that*

$$\frac{4((1-\eta)((1-\eta)+\frac{p}{2})^2}{\tau_0} \lesssim \gamma. \tag{4.133}$$

Proof. Recall the expression for γ in terms of the Herbst argument provided in Theorem 4.4.7:

$$\gamma = \frac{\omega((1-\eta)E^* - \mathbb{E}_\pi[H])^2}{\|\psi\|_P \ln(1/\pi(E^*))}. \tag{4.134}$$

From Lemma 4.5.4 we have that $\|H\|_P = \mathcal{O}(1)$. Applying Equation (4.128) for $k = \Theta(n)$, the log-Sobolev constant ω satisfies:

$$\omega \geq \frac{n}{k(n-k)\tau_0 \log\left(\frac{n}{\min(k,n-k)}\right)} = \frac{4}{\tau_0 n}. \tag{4.135}$$

From Lemma 4.8.4 and Lemma 4.8.1 we have

$$((1-\eta)E^* - \mathbb{E}_\pi(H_c))^2 \asymp \left[(1-\eta)n + \frac{np}{2}\right]^2. \tag{4.136}$$

The definition of π combined with the asymptotics of the binomial coefficient for $k = \Theta(n)$ gives

$$\ln(1/\pi(E^*)) \asymp n. \tag{4.137}$$

Putting everything together:

$$\frac{4((1-\eta)((1-\eta)+\frac{p}{2})^2}{\tau_0} \lesssim \gamma. \tag{4.138}$$

□

Lastly, $|E^*| = \Theta(n)$ with high probability from Lemma 4.8.4. Since this is $\Theta\left(\ln\binom{n}{k}\right)$ for $k = \Theta(n)$ all of the conditions of Theorem 4.5.3 are met.

Unfortunately, the current analysis is insufficient to show this for $k = o(n)$. For example, using Lemma 4.8.2 we can take an upper bound of

$$\Delta_P = \frac{\mathcal{C}_k^*(n-2)}{k(n-k)}, \tag{4.139}$$

which gives that $\frac{|E^*|}{\Delta_P} = \mathcal{O}(k)$. However, $\ln\binom{n}{k} = \Theta(k\ln(n/k))$ for $k = o(n)$. Assuming a tail bound, this leads to a speedup that is falling with n. Specifically, the speedup is falling with $\frac{1}{\ln(n)}$, which is reminiscent of the running time achieved by [Has18b]. We summarize the two cases in the following theorem

Theorem 4.5.6. *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be the k-particle transposition walk on n sites. Let $H : \{-1,1\}^n \mapsto \mathbb{R}$ be a diagonal Hamiltonian encoding the MaxCut-Hamming cost function for a graph $G \sim \mathcal{G}\left(n, \frac{p}{n-1}\right)$. Then either of the following runtimes hold depending on $k \leq n$.*

- *If $k = \Theta(n)$, then there exists a short-path algorithm with runtime*

$$\mathcal{O}^*\left(\binom{n}{k}^{\frac{1}{2}-c}\right), \tag{4.140}$$

  for some constant c, and

- *if $k = o(n)$, then there exists a short-path algorithm with runtime*

$$\mathcal{O}^*\left(\binom{n}{k}^{\frac{1}{2}-\frac{c}{\ln(n)}}\right), \tag{4.141}$$

  for some constant c.

4.5.2 Glauber Dynamics

Glauber dynamics is a Markov Chain algorithm designed to sample from the Gibbs distribution of a system, particularly in spin models like the Ising or hard-core model [Gla63]. The Gibbs measure π for a system with configuration space $\mathcal{X}$ and Hamiltonian $H(x)$ is defined as

$$\pi(x) = \frac{\exp(-\beta H(x))}{Z(\beta)} \tag{4.142}$$

where $x \in \mathcal{X}$ is a configuration, $\beta = \frac{1}{T}$ is the inverse temperature, and $Z(\beta)$ is the partition function

$$Z(\beta) = \sum_{x \in \mathcal{X}} \exp(-\beta H(x)). \tag{4.143}$$

Glauber dynamics generates a Markov chain with this measure as its stationary distribution by sequentially updating a single site (spin or vertex) according marginal distribution. Specifically, the update proceeds as follows: (i) a site (vertex) v is chosen uniformly at random; (ii) the states of all other sites $u \neq v$ remain unchanged; (iii) the new state of v is sampled from the marginal distribution of v conditioned on its neighbors.

The efficiency of Gibbs sampling in this case is related to how fast the Glauber dynamics mixes to its stationary distribution. In fact, approximate Gibbs sampling is tightly connected to partition function estimation in terms of computational complexity [ŠVV09], both of which are closely related to statistical phase transitions. These transitions correspond to the *uniqueness/non-uniqueness* threshold on an infinite d-regular tree which captures whether the root of the tree is affected by the leaves. In the uniqueness regime, correlations decay rapidly, allowing efficient approximation of the partition function. However, beyond the non-uniqueness threshold, long range correlations emerge and no polynomial-time algorithm can approximate the partition function.

In particular, for the hardcore model with fugacity parameter $\lambda = e^{\beta}$, [Wei06] presented a fully polynomial time approximation scheme (FPTAS) for computing the partition function a graphs with maximum degree d when $\lambda \leq (1-\delta)\lambda_c$ where $\lambda_c = \frac{(d-1)^{(d-1)}}{(d-2)^d}$ is the corresponding uniqueness threshold. On the other hand, [Sly10] proved that is no fully polynomial randomized approximation scheme (FPRAS) to approximate the partition function when $\lambda > \lambda_c$ unless NP = RP confirming the main conjecture of [MWW09]. Similarly, for Ising model the phase transition occurs at $\beta_c = \frac{d-2}{d}$ for the antiferromagnetic case and $\beta_c = \frac{d}{d-2}$ for ferromagnetic case.

We first present our main result for the Glauber dynamics and then show that the necessary conditions hold for hardcore-model (maximum independent set problem) and Ising model.

Theorem 4.5.7. *Let P be a Glauber dynamics chain on $\mathcal{X}$ that satisfies an ω log-Sobolev inequality with stationary distribution π, and let $H : \mathcal{X} \mapsto \mathbb{R}$ be a diagonal Hamiltonian with ground state energy*

$$E^* \coloneqq \min_{x \in \mathcal{X}} H(x). \tag{4.144}$$

If $\|H\|_P = \mathcal{O}(1)$ and $\omega^{-1}, |E^*|$, and $\ln(1/\pi(E^*))$ are all $\Theta(n)$, then the short-path algorithm applied to the Glauber chain has a super-quadratic speedup over Markov Chain

search with Glauber dynamics.

Proof. This directly follows from application of Theorem 4.4.3. □

For convenience, we prove the following lemma that is useful when establishing $\|H\|_P = \mathcal{O}(1)$.

Lemma 4.5.8. *Let P be a Glauber dynamics chain on $\mathcal{X}$ and let $H : \mathcal{X} \mapsto \mathbb{R}$ be a diagonal Hamiltonian with ground state energy*

$$E^* \coloneqq \min_{x \in \mathcal{X}} H(x). \tag{4.145}$$

If the following holds for all x, x' such that $P(x, x') > 0$

$$|H(x') - H(x)| = \mathcal{O}(1), \tag{4.146}$$

then $\|H\|_P = \mathcal{O}(1)$.

Proof. The proof simply follows from the definition of $\|H\|_P$ as

$$\|H\|_P = \max_x \sum_{x'} P(x, x')(H(x') - H(x))^2 \leq \max_{x,x'} |H(x') - H(x)|^2 = \mathcal{O}(1). \tag{4.147}$$

□

4.5.2.1 Maximum Independent Set Problem

Given a graph $G(\mathcal{N}, \mathcal{E})$, an *independent set* is a subset of vertices where no two vertices are connected by an edge. A *maximal independent set* (MIS) is the largest independent set of G. Namely, we solve the optimization problem

$$\mathcal{C}^*_{G=(\mathcal{N},\mathcal{E})} \coloneqq \min_{x \in \{0,1\}^{|\mathcal{N}|}} \left\{ -\sum_{i \in \mathcal{N}} x_i : x_i + x_j \leq 1 \ \forall (i,j) \in \mathcal{E} \right\}. \tag{MIS}$$

Equivalently, for $|\mathcal{N}| = n$, we can denote an independent set by a configuration $x \in \mathcal{X} \subseteq \{0,1\}^n$ such that if an edge $e = (i,j) \in \mathcal{E}$ then, $x_i x_j = 0$. We denote the size of the maximum independent set by $|x^*|$. Then, finding the maximum independent set is equivalent to finding a ground state of Hamiltonian $H : \mathcal{X} \to [1, n]$,

$$H(x) = -\sum_{i=1}^{n} x_i.$$

The Hamiltonian H is defined on constrained space $\mathcal{X}$ and therefore we need a constrained walk to explore the state space. For this purpose, we use the Glauber dynamics defined as

$$P_\lambda(x,x') = \begin{cases} 0 & \text{if } |x-x'|>1 \\ \frac{1}{n}\frac{\lambda}{\lambda+1} & \text{if } |x-x'|=1 \text{ and } x\subseteq x' \\ \frac{1}{n}\frac{1}{\lambda+1} & \text{if } |x-x'|=1 \text{ and } x'\subseteq x \\ 1-\sum_{x''\neq x} P_\lambda(x,x'') & \text{if } x=x'. \end{cases} \tag{4.148}$$

This model is also referred as hard-core model in statistical physics and we'll use the same terminology. Note that Glauber dynamics initialized at an independent set can only move between independent sets. Furthermore, it converges to its stationary distribution

$$\pi_\lambda(x) = \frac{\lambda^{|x|}}{Z}. \tag{4.149}$$

The parameter λ is also called *fugacity* and one can recover the Gibbs form in 4.142 by setting $e^\beta = \lambda$. In principle, one can find the maximum independent set by setting λ sufficiently high so that distribution π_λ concentrates around the global minimum of H. Unfortunately, this approach results in exponential mixing time due to uniqueness/non-uniqueness phase transitions. Alternatively, one can draw exponentially many samples from π_λ at $\lambda<\lambda_c$ by running the Glauber dynamics chain in polynomial time as Glauber dynamics mix efficiently below the critical threshold [CLV21]. As the current quantum techniques can only quadratically improve the run time of the first approach, we use the second approach. More specifically, we consider Glauber chain P_λ at $\lambda<\lambda_c$ so that we can prepare $\left|\sqrt{\pi_\lambda}\right\rangle$ efficiently. Next, we consider the short-path Hamiltonian $H_b : \mathcal{X}\mapsto\mathbb{R}$,

$$H_b = -D(P_\lambda) + bg_\eta\left(\frac{H}{|E^*|}\right), \tag{4.150}$$

where D is the discriminant matrix as usual and $E^* = \min_x H(x) = -|x^*|$. In accordance with the generalized short path framework, the algorithm starts from $\left|\sqrt{\pi_\lambda}\right\rangle$ and jumps to the ground state of H_b for $b>0$. We describe how we can prepare the block-encoding of D for Glauber dynamics and also prepare π_λ in Appendix 4.10.

The following lemma establishes the condition on $\|H\|_P$ given in theorem 4.5.7.

Lemma 4.5.9. *Let P be a Glauber dynamics chain with fugacity parameter λ on a graph $G(\mathcal{N},\mathcal{E})$. Then, $\|H\|_P = \mathcal{O}(1)$.*

Proof. As Glauber dynamics flips one spin at a time, for all $x,x'\in\mathcal{X}$ such that $P(x,x')>$

0 we have

$$|H(x) - H(x')| \leq 1. \tag{4.151}$$

Then, by Lemma 4.5.8, $\|H\|_P = \mathcal{O}(1)$. □

As required by Theorem 4.5.7, we need to characterize the log-Sobolev constant of Glauber dynamics for hard-core model on a graphs with a degree upper bounded by d. We start with the following fact from [CLV21],

Fact 4.5.10. Let V be a set of size n and μ be a distribution over $[q]^V$. If π satisfies the approximate tensorization of entropy with constant C_1 and π is u-marginally bounded, then the Glauber dynamics for π satisfies the standard log-Sobolev inequality with constant $\omega = \frac{1-2u}{\log(1/u-1)}\frac{1}{C_1 n}$ when $u < \frac{1}{2}$, or $\omega = \frac{1}{2C_1 n}$ when $u = \frac{1}{2}$.

Proof. Fix a configuration x and consider a Markov chain P_v that updates vertex v according to marginal probability distribution $\pi_v = \pi(v|x_{V-\{v\}})$. Then, LS constant of this Markov chain ρ_v is lower bounded by $\rho_v \geq \frac{1-2\pi_v^*}{\log(1/\pi_v^*-1)}$ when $\pi_v^* < \frac{1}{2}$ or $\rho_v = \frac{1}{2}$ when $\pi_v^* = \frac{1}{2}$ due to [DSC96, Theorem A.1]. By the definition of the log-Sobolev constant, we have

$$\rho_v \operatorname{Ent}_{\pi_v}[f] \leq \mathcal{E}_{P_v}(\sqrt{f}, \sqrt{f}) = \operatorname{Var}_{\pi_v}[\sqrt{f}]. \tag{4.152}$$

The transition matrix of Glauber dynamics can be written as $\frac{1}{n}\sum_{v\in V}\pi(v|\cdot)$. Therefore, the Dirichlet form for Glauber dynamics is

$$\mathcal{D}(\sqrt{f}, \sqrt{f}) = \frac{1}{n}\sum_{v\in V}\operatorname{Var}_{\pi_v}[\sqrt{f}]. \tag{4.153}$$

Using the tensorization of entropy,

$$\operatorname{Ent}_{\pi}[f] \leq C_1 \sum_{v\in V}\operatorname{Ent}_{\pi_v}[f] \leq C_1 \sum_{v\in V}\frac{1}{\rho_v}\operatorname{Var}_{\pi_v}[\sqrt{f}] \leq C_1 n \max_v\left(\frac{1}{\rho_v}\right)\mathcal{D}(\sqrt{f}, \sqrt{f}). \tag{4.154}$$

Combining the marginal boundedness property of π (i.e., $\pi_v \geq u$ for all $v \in V$), with the monotonicity of the function $\frac{1-2y}{\log(1/y-1)}$ for $y \in [0, 1/2]$, it follows that $\omega \geq \frac{1-2u}{\log(1/u-1)}\frac{1}{C_1 n}$ when $u < \frac{1}{2}$, and $\omega = \frac{1}{2C_1 n}$ when $u = \frac{1}{2}$.

□

Theorem 4.5.11 (Entropy factorization, Theorem 2.9 in [CLV21])**.** *Let $d \geq 3$ be an integer and $b, \eta > 0$ be reals. Suppose that $G = (\mathcal{N}, \mathcal{E})$ is an n-vertex graph of maximum*

degree at most d and μ is a totally connected Gibbs distribution of some spin system on G. If μ is both u-marginally bounded and η-spectrally independent and $n \geq \frac{24d}{u^2}(\frac{4\eta}{u^2}+1)$, then μ satisfies the approximate tensorization of entropy with constant

$$C_1 = \frac{18\log(1/u)}{u^4}\left(\frac{24d}{u^2}\right)^{\frac{4\eta}{u^2}+1}. \tag{4.155}$$

Spectral independence and marginal boundedness properties for graphs with constant maximum degree are proven in [CLV23, CLV21] respectively. This shows that the log-Sobolev constant of a graph with bounded degree is $\Omega(n^{-1})$. Having showed that Glauber dynamics chain satisfies desired Log-Sobolev constant and $\|H\|_P$, we present the final run time.

Theorem 4.5.12. *Let P be a Glauber dynamics chain on a graph $G(\mathcal{N},\mathcal{E})$ with a bounded maximum constant degree d at fugacity parameter $\lambda < \lambda_c$ and stationary distribution π. Then, there exists a short-path algorithm that finds the maximum independent set in G with running time*

$$\mathcal{O}^*\left([\pi(E^*)^{-1}]^{\frac{1}{2}-c}\right), \tag{4.156}$$

where $c > 0$ is a constant.

Proof. It is evident that for sparse graphs ($d = \mathcal{O}(1)$), the size of the maximum independent set is $\Theta(n)$. Since d is constant in n, the problem still can not be solved in poly(n) time and requires $2^{\mathcal{O}(n)}$ samples for hard instances. Hence, $\log(1/\pi_\lambda(E^*)) = \Theta(n)$. The log-Sobolev constant $\omega^{-1} = \mathcal{O}(n)$ for the Glauber dynamics on bounded degree graphs below the critical fugacity λ_c. Therefore, the conditions in Theorem 4.5.7 holds. Hence, we obtain a super-quadratic speedup over sampling from π_λ. □

A limitation of the result in Theorem 4.5.12 is that it is not clear how this run-time compares to the best classical algorithms for MIS problem. As we consider a Markov chain that mixes to Gibbs distribution with fugacity λ below the critical threshold λ_c, sampling from the Gibbs distribution may favor the smaller independent sets when $\lambda_c < 1$. However, even if this is the case, the stationary distributions only has support on the constrained space (i.e. on the valid independent sets) rather than on the entire 2^n possible configurations. Therefore, we might still expect to prove that the success probability is larger than brute force search for certain graph models. To this end, the following proposition lower bounds the probability of finding the optimum by sampling from Glauber dynamics for random regular graphs.

Proposition 4.5.13. *Let π_λ be the stationary distribution of Glauber dynamics on a random d-regular graph $G \sim \mathcal{G}(n,d)$. Choose x^* to be a particular maximum independent set in G. Then, for $\lambda = \lambda_c$ we have*

$$\pi_\lambda(x^*) \geq 2^{-\kappa n}, \tag{4.157}$$

with $\kappa = -\frac{2\log(\lambda)\log d}{d} + \frac{1}{2d} + \frac{\log(1+\lambda)}{2}$.

Proof. The probability of x^* in π_λ is given by

$$\pi_\lambda(x^*) = \frac{\lambda_c^{|x^*|}}{\sum_{x\in\mathcal{X}} \lambda^{|x|}}. \tag{4.158}$$

We first bound the denominator. To do that, we invoke [Zha09, Theorem 2], which asserts that

$$Z(\lambda) = \sum_{x\in\mathcal{X}} \lambda^{|x|} \leq (2(1+\lambda)^d - 1)^{\frac{n}{2d}}, \tag{4.159}$$

for any d-regular graph.

For the numerator, we need to bound $|x^*|$. A known upper bound for the size of the maximum independent set is given by $2\log(d)n/d$ [Bol81]. Combining these, we have

$$\pi(x^*) \geq \frac{\lambda^{(2\frac{n}{d}\log d)}}{(2(1+\lambda)^d - 1)^{\frac{n}{2d}}} \geq \frac{\lambda^{(2\frac{n}{d}\log d)}}{(2(1+\lambda)^d)^{\frac{n}{2d}}} = 2^{n(\frac{2\log(\lambda)\log d}{d} - \frac{1}{2d} - \frac{\log(1+\lambda)}{2})}. \tag{4.160}$$

□

For random regular graphs, the Glauber dynamics mixes up to $\lambda = \mathcal{O}(1/\sqrt{d})$ [CCC+25]. Note that this is beyond tree uniqueness threshold $\lambda_c = \mathcal{O}(1/d)$. For large degree $d > d_0$, κ in 4.5.13 is smaller than 1, hence the runtime is smaller than 2^n which is the runtime of brute force search over unconstrained space. Therefore, Markov Chain Search for this problem might be significantly faster than brute force search mainly due to fact that Glauber dynamics stays in the constrained space.

Remark 4.5.14. The parameter κ from Proposition 4.5.13 is a decreasing function of the degree. For sufficiently high degree, the runtime $2^{\kappa n}$ the is in fact faster than the best known generic algorithm for Maximum Independent Set (runtime of $\tilde{\mathcal{O}}(1.1996^n)$, due to Xiao and Nagamuchi [XN17]).

4.5.2.2 Ising Model

Consider the 2-spin Ising model on a graph $G(\mathcal{N}, \mathcal{E})$ defined via the Hamiltonian

$$H(x) = \sum_{(i,j)\in\mathcal{E}} x_i x_j + \sum_j h_j x_j, \tag{4.161}$$

where the entries of J are interaction coefficients and h defines an external field.

Assume $J_{ij} = 1$ for all $(i,j) \in \mathcal{E}$ and $h = \mathbf{0}$. This model corresponds to Gibbs sampling with weights $\pi(x) \propto \exp(-\beta H(x))$, and a Gibbs sample can be prepared by using Glauber dynamics similar to the hardcore model. The only change is the transition probabilities, which can be computed by the marginal distribution $\pi(x_i^{t+1}|x^t_{\mathcal{N}\setminus\{i\}})$. For simplicity we consider the anti-ferromagnetic model, where having two neighboring sites have the same spin results in lower probability than having the same spin.

Assuming the underlying graph is sparse ($d = \mathcal{O}(1)$), then Glauber dynamics mixes in poly(n) time for $\beta < \frac{d-2}{d} = \beta_c$ [CLV21] at which the uniqueness/non-uniqueness phase transition occurs. We consider the following short path Hamiltonian

$$H_b(x) = -D(P_\beta(x)) + bg_\eta\left(\frac{H}{|E^*|}\right), \tag{4.162}$$

where P_β is the Glauber dynamics transition matrix for $\beta < \beta_c$. Similar to the setting of the MIS problem, a block-encoding of $D(P_\beta)$ can be prepared efficiently.

Proposition 4.5.15. *The optimum energy of Ising Model Hamiltonian H on a random regular graph $G(\mathcal{N}, \mathcal{E})$ satisfies $|E^*| = \Theta(n)$.*

Proof. Let s denote the number of edges in the graph. The ground state of H can be related to minimum bisection width [ZB10] denoted by |BW| as follows

$$|\mathrm{BW}| = \frac{s + E_{\mathrm{gs}}}{2}. \tag{4.163}$$

Using this equality, $E_{\mathrm{gs}} = s - 2|BW|$. Next, we consider random regular graphs. For sparse random regular graphs $s = \Theta(n)$ and $|\mathrm{BW}| = \Theta(n)$ (See [DSW07, COLMS22]). □

Theorem 4.5.16. *Let P be a Glauber dynamics chain on a random regular graph $G(\mathcal{N}, \mathcal{E})$ with constant degree d at inverse temperature parameter $\beta < \beta_c$ and stationary distribution π_β. Then there exists a short-path algorithm that finds the optimum of Ising*

model Hamiltonian on a random regular graph with running time

$$\mathcal{O}^*\left([\pi_\beta(E^*)^{-1}]^{\frac{1}{2}-c}\right), \tag{4.164}$$

where $c > 0$ is a constant.

Proof. The spectral independence and marginal boundedness for Ising model on sparse graphs are proven in [CLV21]. Therefore, Glauber dynamics for Ising model on a regular graph has $\omega^{-1} = \mathcal{O}(n)$. Furthermore, $\log(1/\pi_\beta(E^*)) = \Theta(n)$ due to hardness of the problem. Similar to MIS problem, for all $x, x' \in \{-1, 1\}^n$ such that $P(x, x') > 0$, $|H(x) - H(x')| \leq 2d = \mathcal{O}(1)$. By Proposition 4.5.15, $|E^*| = \mathcal{O}(n)$. Therefore, the conditions in Theorem 4.5.7 are satisfied. Hence, we have the super-quadratic sampling over sampling from π_β. □

4.5.2.3 Sherrington-Kirkpatrick Model

Consider the (possibly diluted) Sherrington-Kirkpatrick Hamiltonian on a graph $G(\mathcal{N}, \mathcal{E})$,

$$H(x) = \frac{1}{\sqrt{n}} \sum_{(i,j)\in\mathcal{E}}^{n} g_{ij} x_i x_j, \tag{4.165}$$

where the interaction coefficients g_{ij} are i.i.d. standard Gaussian random variables. We can Gibbs sample from the following distribution,

$$\pi(x) \propto \exp(-\beta H(x)) \tag{4.166}$$

using Glauber dynamics.

Lemma 4.5.17. *Let P be a Glauber dynamics chain for the SK model on a graph $G(\mathcal{N}, \mathcal{E})$. Then,*

$$\Delta_P = \mathcal{O}(1). \tag{4.167}$$

Proof. We first consider hypercube walk. If we flip a spin at random, the sign of each term in H will flip with probability $2/n$. Therefore the energy of each term increases at most by a factor of $1 - \frac{4}{n}$ in expectation. Since the ground state energy $|E^*| = \Theta(n)$, the energy increases at most by constant in expectation. From the definition of Glauber dynamics a bit flip is proposed uniformly, and accepted with probability larger than $\frac{1}{2}$ if $\beta(H(x') - H(x)) < 0$. Therefore, if we are running the Glauber dynamics at some positive finite temperature, moves that increase energy are made with strictly lower

probability than the hypercube walk. Thus an upper bound on stability with respect to the hypercube walk is also a valid upper bound for Glauber dynamics at positive β. $\square$

Lemma 4.5.18. *Let P be a Glauber dynamics chain for SK model on a graph $G(\mathcal{N}, \mathcal{E})$ with constant maximum degree, at inverse temperature $\beta < \beta_c$. Then, the log-Sobolev constant satisfies $\omega \geq \Omega(1/(n \log n))$.*

Proof. By [AJK+22, Theorem 12, part (a)], the modified Log Sobolev constant for Glauber dynamics is $\Omega(1/n)$ when $\beta < \beta_c$ (See the discussion in Page 11). On a graph with constant bounded degree, the transition probability of Glauber dynamics $\Theta(n^{-1})$. Hence, by Theorem 4.4.18, the log-Sobolev constant scales as $\Omega(1/(n \log(n)))$. $\square$

Theorem 4.5.19. *Let P be a Glauber dynamics chain for the Sherrington-Kirkpatrick model on a regular graph $G(\mathcal{N}, \mathcal{E})$ with a bounded maximum degree d at inverse temperature parameter $\beta < \beta_c$ and stationary distribution π_β. Then there exists a short-path algorithm that finds the optimal solution of the Sherrington Kirkpatrick Hamiltonian with running time*

$$\mathcal{O}\left(\text{poly}(n)[\pi_\beta(E^*)^{-1}]^{\left(\frac{1}{2}-\frac{c}{\log(n)}\right)}\right), \tag{4.168}$$

where $c > 0$ is a constant.

Proof. We first show that the tail bound holds for SK model. By using proposition 4 in [DPCB23], we know that the number of low energy states with energy smaller than $E^*(1-\eta)$ is smaller than $2^{\gamma n}$ where γ is a constant. By assuming that $\log(1/\pi(E^*)) = \Theta(n)$, we can conclude that the generalized tail bound holds as well. Note that if this assumption fails, then it means that there exists a sub-exponential solver for SK model. Finally, since $|E^*| = \Theta(n)$ and $\log(1/\pi(E^*)) = \Theta(n)$ for the SK Model by using Lemma 4.5.18 and Theorem 4.4.9, we have $\gamma = \mathcal{O}(1)$. Thus, by Theorem 4.4.15, we have $b = \mathcal{O}(1/\log(n))$. Since Δ is constant by Lemma 4.5.17, the total runtime scales as $(\pi_\beta(E^*))^{-1\left(\frac{1}{2}-\frac{c}{\log(n)}\right)}$ due to Theorem 4.3.4. $\square$

We conclude this section by demonstrating that Markov Chain search using Glauber dynamics at a positive inverse-temperature is faster than unstructured search.

Lemma 4.5.20. *Let π be the Gibbs distribution corresponding to a cost function $H\colon \{0,1\}^n \mapsto \mathbb{R}$ at some positive inverse temperature $\beta > 0$. We also assume that the cost function is concentrated away from its optimum, i.e., the number of states with cost greater than $(1-\eta)E^*$ is lower than $2^{-\gamma n}$.*

Proof. We first observe due to the assumption on concentration that the partition function $Z(\beta) \leq 2^n[(1-2^{-\gamma n})\exp(-\beta(1-\eta)E^*) + 2^{-\gamma n}\exp(-\beta E^*)]$. As a consequence,

$$\frac{\pi(E^*)}{2^{-n}} \geq \frac{\exp(-\beta E^*)}{(1-2^{-\gamma n})\exp(-\beta(1-\eta)E^*) + 2^{-\gamma n}\exp(-\beta E^*)} \tag{4.169}$$

$$\geq \frac{1}{0.5\exp(-\beta\eta|E^*|) + 2^{-\gamma n}} \tag{4.170}$$

$$= \Omega\left(\min\left(2^{\gamma n}, \exp(\beta\eta|E^*|)\right)\right). \tag{4.171}$$

□

4.6 Numerical Results

We perform numerical evaluations to empirically verify our findings. We focus on the constrained problems studied in this work, including (MaxBisection), MaxCut with a Hamming weight constraint $k = o(n)$ (MaxCut-Hamming), and MIS with a penalized objective. For (MaxBisection), we take n to be even since $k = \frac{n}{2}$. For (MaxCut-Hamming), we take $k = \lfloor\sqrt{n}\rfloor$. For MIS, we take the penalty factor to be n, such that no energy reduction from constraint violation can justify the penalty. For all three problems, we generate 100 random unweighted graphs for each n from the Erdős–Rényi model with the probability of each edge existing to be $\frac{2\ln n}{n}$. The constant factor 2 is chosen to ensure a reasonable graph density at the scale we cover. We set $\eta = 0.5$ in all experiments.

To improve the scalability of our numerical experiments, we construct H_b as a sparse matrix in the compressed sparse row format and employ a GPU-accelerated iterative eigensolver to compute only the two smallest eigenvalues and the corresponding eigenvectors. For (MaxBisection) and (MaxCut-Hamming), which are explicitly constrained, we can scale up further by projecting H_b onto the space spanned by all feasible states. The dimension of the computational space then drops from 2^n to $\binom{n}{k}$. With these efforts, we obtained results with up to 30 qubits for (MaxCut-Hamming).

First, we want to identify what values of b are practically appropriate. In [DPCB23], the authors numerically show that the original short-path algorithm works well for the 3-spin problem for b up to around 0.8, which is much larger than the theoretical bound of $b \leq 1.02 \times 10^{-4}$. Here, we show that a similar observation can be made for the constrained and penalized cases. For (MaxCut-Hamming), Figure 4.1 A shows the quartiles of b values that minimize the effective runtime (Equation 4.7) of the algorithm. Note that the b values are hard-capped at 1.25 and may be higher. We see that as n increases,

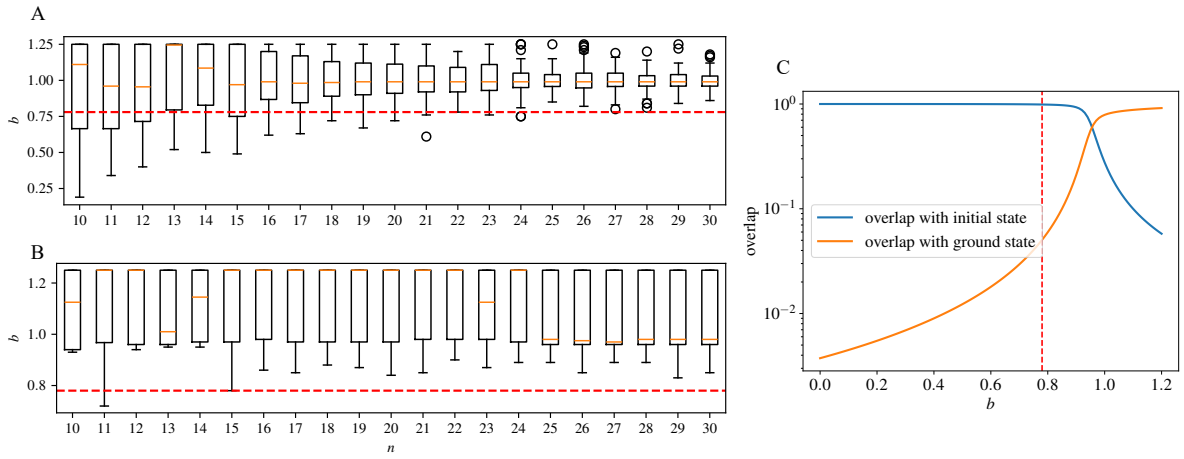

Figure 4.1. Empirical selection of b. **A** Quartiles of b values that minimize the effective runtime of the algorithm for (MaxCut-Hamming). As n increases, the runtime-optimal b converges to a range approximately between 0.8 and 1.2. The red dot line shows the converged value of $b \approx 0.78$ of phase transition where the overlap with the initial state crosses 0.99. **B** Quartiles of b values that minimize the spectral gap for (MaxCut-Hamming). For most instances tested, the spectral gap is minimized when b is larger than the phase transition value, rendering the phase transition b a safe choice. **C** The overlap values with the initial state and the ground state (optimal solution) for one $n = 30$ (MaxCut-Hamming) instance with varying b. The dotted vertical line denotes the phase transition b.

the optimal b converges to a range approximately between 0.8 and 1.2. However, when choosing the value of b that needs to work for all instances, we want a conservative value that avoids encountering the possibly super-exponentially small spectral gap. For this purpose, we identify the value of b at which the phase transition occurs. We numerically characterize the phase transition point by the overlap of $|\psi_b\rangle$ with the initial state dropping below 0.99. An example of the overlap with varying b is shown in Figure 4.1 C. Empirically, we observe that the phase transition b converges to around 0.78 as n increases. In Figure 4.1 B, we plot the quartiles of b values that minimize the spectral gap. For most instances, the spectral gap is minimized when b is greater than the phase transition value ≈ 0.78. Therefore, we expect a ubiquitous value of phase transition to work for a (MaxCut-Hamming) instance with high probability.

We then fit the worst-case runtime to empirically demonstrate the super-Grover speedup. Although the inverse of the spectral gap term in the runtime (Equation 4.7) has a $\mathcal{O}(\text{poly}(n))$ complexity, it may still affect the exponential fitting at the scale of numerical experiments. Thus, we use the inverse of ground state overlap $|\langle\psi_b|z\rangle|^{-1}$, the only exponential growth term in the runtime, to fit the asymptotic speedup. In

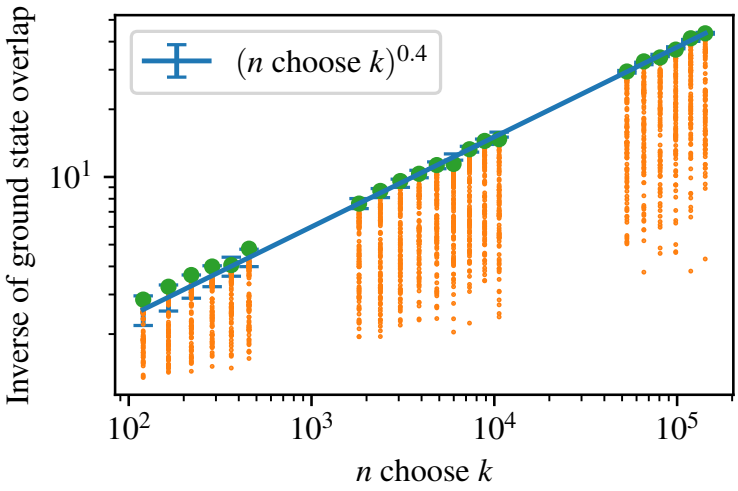

Figure 4.2. The inverse of the ground state overlap versus the feasible space size $\binom{n}{k}$ for (MaxCut-Hamming) with n varying from 10 to 30 and $b = 0.78$. The worst-case instances are fitted using an exponential function with base $\binom{n}{k}$ with an error bar denoting one standard deviation of the fitted exponent. The 95% confidence interval on the fitted exponent is $[0.391, 0.408]$.

Figure 4.2, we set b to be 0.78 and plot the inverse of ground state overlap $|\langle\psi_b|z\rangle|^{-1}$ of all (MaxCut-Hamming) instances with respect to $\binom{n}{k}$, the size of the feasible space. We fit the worst-case instances using an exponential function with base $\binom{n}{k}$, the exponent of which is equivalent to the factor a in 2^{an} for the unconstrained case. The error bar of the fitted line denotes one standard deviation of the fitted exponent. We see the empirical b values give a super-Grover speedup, which is much better than the theoretically guaranteed bounds.

In Figure 4.3, we show the empirical worst-case scaling for all three problems with different choices of b. A proper selection of b yields a super-Grover speedup across all examined problems. Conversely, when b is excessively high, the algorithm may encounter a small spectral gap in the worst case. To demonstrate this, we use the runtime (Equation 4.7, which includes the inverse of the spectral gap term) as the metric and observe that the quality of the fitting degrades. Our numeric lead to two interesting conceptual observations: firstly, as observed also by [DPCB23] the optimal choices of b are well beyond what is predicted by the theoretical analysis. Secondly, in the case of (MaxCut-Hamming) we numerically observe an advantage over quadratic speedup that does not decay with n which is beyond the current theoretical analysis and indicates that the runtime of the long jump can possibly be characterized through weaker conditions than Δ_p smoothness. Finally, we confirm in our setting that is indeed reasonable to make

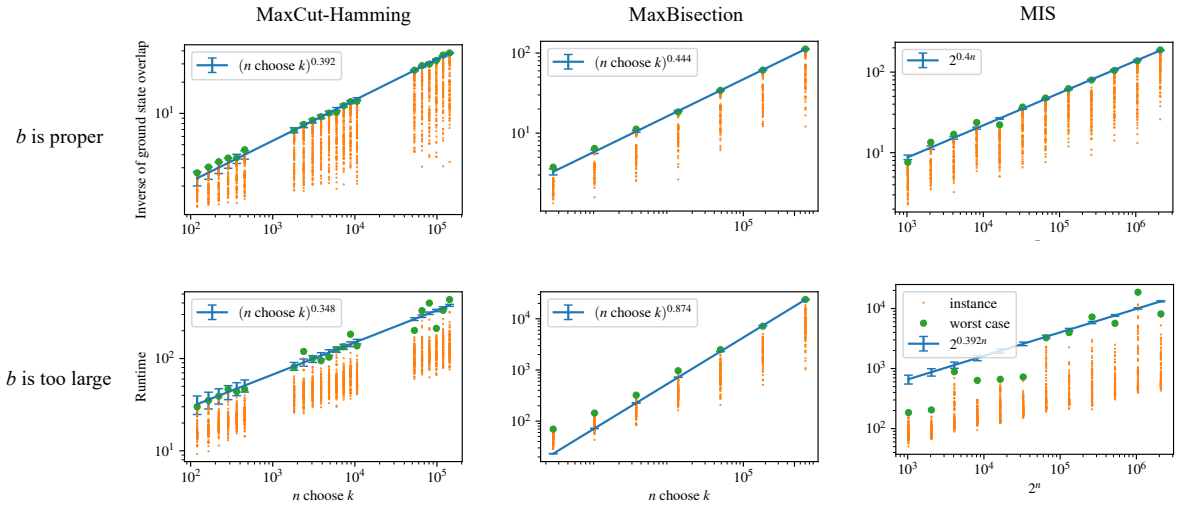

Figure 4.3. Empirical fitting of the inverse overlap of the ground state $|\langle\psi_b|z\rangle|^{-1}$ and the runtime 4.7 of (MaxCut-Hamming), (MaxBisection), and MIS with difference choices of b. For the left column (MaxCut-Hamming), we use data with n ranging from 10 to 30. Top: $b = 0.8$ the fitted exponent is 0.392 with 95% confidence interval $[0.383, 0.402]$, indicating there could exist b that is better than the phase transition one in Figure 4.2; Bottom: $b = 1$ the fitted exponent is 0.348 with 95% confidence interval $[0.271, 0.426]$. For the middle column (MaxBisection), we use data with n ranging from 16 to 22. Top: $b = 0.7$ the fitted exponent is 0.444 with 95% confidence interval $[0.436, 0.452]$; Bottom: $b = 1$ the fitted exponent is 0.873 with 95% confidence interval $[0.842, 0.905]$. For the right column (MIS), we use data with n ranging from 10 to 21. Top: $b = 0.6$ the fitted exponent is 0.400 with 95% confidence interval $[0.386, 0.415]$; Bottom: $b = 0.8$ the fitted exponent is 0.392 with 95% confidence interval $[0.122, 0.663]$.

the choice of b by choosing the largest such value that allows for large overlap with the ground state. If the value of this critical b asymptotes quickly as a function of n this suggests a numerical mechanism for the development of efficient short-path algorithms.

4.7 Technical Details for Generalized Short Path Framework

Lemma 4.7.1 (θ short-path $\implies$ θ Spectral Gap Bound, adapted from Proposition 5 of [DPCB23])**.** *If H_b satisfies the θ short-path condition, then the spectral gap of H_b is at least θ, i.e., all excited states have energy at least $-1 + \theta$.*

Proof. Recall by construction, the ground state energy of H_b is at most -1. Note that $\theta \leq 1$. Suppose, in order to arrive at a contradiction, that there are at least

two orthogonal eigenstates $|\psi_1\rangle$ and $|\psi_2\rangle$, with energy strictly below $-1+\theta$. Since the short-path condition is satisfied, at least one of $|\psi_1\rangle$ or $|\psi_2\rangle$ has nonzero overlap with $|\pi\rangle$. Without loss of generality, assume this is the case for $|\psi_2\rangle$.

Now consider the state

$$|\psi'\rangle = \left(1 + \frac{|\langle\sqrt{\pi}|\psi_1\rangle|^2}{|\langle\pi|\psi_2\rangle|^2}\right)^{-1/2} \left(|\psi_1\rangle - \frac{\langle\sqrt{\pi}|\psi_1\rangle}{\langle\sqrt{\pi}|\psi_2\rangle}|\psi_2\rangle\right), \tag{4.172}$$

which is orthogonal to $|\sqrt{\pi}\rangle$. Since $\mathsf{GSE}\left(\Pi_\perp H_b \Pi_\perp\right) \leq \langle\psi'|H_b|\psi'\rangle$, and

$$\langle\psi'|\Pi_\perp H_b \Pi_\perp|\psi'\rangle = \langle\psi'|H_b|\psi'\rangle \tag{4.173}$$

$$< \left(1 + \frac{|\langle\sqrt{\pi}|\psi_1\rangle|^2}{|\langle\sqrt{\pi}|\psi_2\rangle|^2}\right)^{-1} \left[(-1+\theta) + \frac{|\langle\sqrt{\pi}|\psi_1\rangle|^2}{|\langle\sqrt{\pi}|\psi_2\rangle|^2}(-1+\theta)\right] \leq -1+\theta, \tag{4.174}$$

we get a contradiction. □

Lemma 4.7.2 (Runtime bound by approximate projector)**.** *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a reversible Markov chain with $|\mathcal{X}| = V$. Define*

$$\mathcal{P}_\ell := \left(\frac{H_b}{|E_b|}\right)^\ell, \tag{4.175}$$

and let ω be the log-Sobolev constant of $D(P)$. For either all even or all odd ℓ we have

$$\left\langle\sqrt{\pi}\middle|\mathcal{P}_\ell\middle|z\right\rangle - V\left(1 - \frac{\omega}{2}\right)^\ell < \langle\sqrt{\pi}|\psi_b\rangle\langle\psi_b|z\rangle. \tag{4.176}$$

Proof. Note that $E_b \leq -1$. By Lemma 4.4.12 all excited states have energy at least $-1+\frac{\omega}{2}$.

Since ω is the log-Sobolev constant of $D(P)$ (lower bounding its spectral gap δ as defined in (1.24)) and $g_\eta(H)$ is negative definite, at most two eigenvectors of H_b have have magnitude $> 1-\frac{\omega}{2}$, the ground state and highest-energy state. Thus,

$$\langle\sqrt{\pi}|\mathcal{P}_\ell|z\rangle = \langle\sqrt{\pi}|\psi_b\rangle\langle\psi_b|z\rangle + \sum_{|E_b'|>1-\frac{\omega}{2}} \left(\frac{E_b'}{E_b}\right)^\ell \langle\sqrt{\pi}|\psi_b'\rangle\langle\psi_b'|z\rangle + \sum_{|E_b'|\leq 1-\frac{\omega}{2}} \left(\frac{E_b'}{E_b}\right)^\ell \langle\sqrt{\pi}|\psi_b'\rangle\langle\psi_b'|z\rangle \tag{4.177}$$

$$\leq \langle\sqrt{\pi}|\psi_b\rangle\langle\psi_b|z\rangle + \left(\frac{E_b'}{E_b}\right)^\ell \langle\sqrt{\pi}|\psi_b'\rangle\langle\psi_b'|z\rangle + (V-2)(1-\frac{\omega}{2})^\ell. \tag{4.178}$$

Supposing ℓ takes a value such that the middle term is non-positive, we obtain

$$\langle\sqrt{\pi}|\mathcal{P}_\ell|z\rangle - V(1-\frac{\omega}{2})^\ell < \langle\sqrt{\pi}|\psi_b\rangle\langle\psi_b|z\rangle. \tag{4.179}$$

□

Lemma 4.7.3 (ground-state energy shift bound, adapted from Proposition 6 of [DPCB23])**.** *Suppose that γ-spectral density and $\frac{\omega}{2}$-short-path condition hold, then*

$$|E_b| < 1 + \frac{4(\pi(E^*))^\gamma}{\omega}. \tag{4.180}$$

Proof. In the proof of Proposition 6 of [DPCB23], the authors showed that

$$E_b = \langle\psi_b|H_b|\psi_b\rangle = -1 - b\langle\sqrt{\pi}|G_\eta|\sqrt{\pi}\rangle - b^2\langle\sqrt{\pi}|G_\eta W_b G_\eta|\sqrt{\pi}\rangle, \tag{4.181}$$

where

$$W_b := (\Pi_\perp(H_b - E_b)\Pi_\perp)^{-1} = \Pi_\perp(H_b - E_b)^{-1}\Pi_\perp \tag{4.182}$$

$$\Pi_\perp := I - |\sqrt{\pi}\rangle\langle\sqrt{\pi}|. \tag{4.183}$$

Recall the short-path condition

$$\mathsf{GSE}(\Pi_\perp H_b \Pi_\perp) \geq -1 + \frac{\omega}{2}. \tag{4.184}$$

Thus, with $E_b \leq -1$,

$$\mathsf{GSE}(\Pi H_b \Pi) \geq -1 + \frac{\omega}{2} \geq E_b + \frac{\omega}{2} \tag{4.185}$$

$$\implies \mathsf{GSE}(\Pi_\perp(H_b - E_b)\Pi_\perp)) \geq \omega/2, \tag{4.186}$$

so $\|W_b\|_2 \leq \frac{2}{\omega}$ and as a consequence,

$$\langle\sqrt{\pi}|G_\eta(\Pi(H_b - E_b)\Pi)^{-1}G_\eta|\sqrt{\pi}\rangle \leq \frac{2}{\omega}\langle\sqrt{\pi}|G_\eta^2|\sqrt{\pi}\rangle. \tag{4.187}$$

From here, applying Equation (4.181) yields

$$E_b \geq -1 - b\langle\sqrt{\pi}|G_\eta|\sqrt{\pi}\rangle - \frac{2b^2}{\omega}\langle\sqrt{\pi}|G_\eta^2|\sqrt{\pi}\rangle. \tag{4.188}$$

By spectral density and definition of G_η:

$$-\langle\sqrt{\pi}|G_\eta|\sqrt{\pi}\rangle \leq (\pi^*)^\gamma \tag{4.189}$$

$$\langle\sqrt{\pi}|G_\eta^2|\sqrt{\pi}\rangle \leq (\pi^*)^\gamma. \tag{4.190}$$

Using $0 \leq b \leq 1$, $\pi^* < 1, \omega < 1$, we have

$$E_b \geq -1 - (\pi^*)^\gamma \left(b + b^2 \frac{2}{\omega}\right) > -1 - \frac{4(\pi^*)^\gamma}{\omega}. \tag{4.191}$$

□

Next we provide an alternative bound on the energy shift. At a high level, the goal is to show that the small ground state energy shift condition is satisfied for any point we can efficiently go to in the short jump. It removes the need for the short-path condition. Note, heuristically, under this large ground state overlap condition, the short-path condition reduces to the large gap condition:

$$\begin{aligned}&\mathsf{GSE}((I - |\sqrt{\pi}\rangle\langle\sqrt{\pi}|)H_b(I - |\sqrt{\pi}\rangle\langle\sqrt{\pi}|)) - E_b \\ &\approx \mathsf{GSE}((I - |\psi_b\rangle\langle\psi_b|)H_b(I - |\psi_b\rangle\langle\psi_b|)) - E_b = \mathrm{Gap}(H_b).\end{aligned} \tag{4.192}$$

Lemma 4.7.4 ($\Delta_P(\eta)$ stable $\iff$ α_P subdepolarizing)**.** *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be a reversible Markov chain. The pair $(\mathcal{M}, H)$ satisfies the $\Delta_P(\eta)$-stable if and only if it satisfies α_P-subdepolarizing. They are related by the equation $\alpha_P = \frac{\Delta_P(\eta)}{|E^*|(1-\eta)}$.*

Proof. The $\implies$ direction follows mostly from the proof of Proposition 3 in [DPCB23] By definition of g_η, f is monotonically non-decreasing. From [DPCB23, Proposition 12], $\prod_{t=1}^T f(c_t x)$ is also a convex function. Thus

$$\sum_y P(x,y) \prod_{t=1}^T f\left(\frac{c_t H(y)}{E^*}\right) \geq \prod_{t=1}^T f\left(\sum_y P(x,y) \frac{c_t H(y)}{E^*}\right) \tag{4.193}$$

$$\geq \prod_{t=1}^T f\left(\frac{c_t H(x)}{E^*}\left(1 + \frac{\Delta}{H(x)}\right)\right). \tag{4.194}$$

Recall that $g_\eta(z)$ is zero for all $z \geq (1-\eta)E^*$. If for some t, $\frac{c_t H(x)}{E^*} > (1-\eta)E^*$, then the whole product is zero, as in this case

$$\frac{c_t H(x)}{E^*}\left(1 + \frac{\Delta}{H(x)}\right) > (1-\eta)E^*. \tag{4.195}$$

Thus, the stated hypothesis is satisfied trivially. If

$$\frac{H(x)}{E^*} < \frac{(1-\eta)E^*}{c_t} \leq (1-\eta)E^* \tag{4.196}$$

for all t, then

$$\frac{1}{H(x)} > -\frac{1}{(1-\eta)|E^*|}. \tag{4.197}$$

By monotonicity of f

$$\prod_{t=1}^{T} f\left(\frac{c_t H(x)}{E^*}\left(1+\frac{\Delta}{H(x)}\right)\right) \geq \prod_{t=1}^{T} f\left(\frac{c_t H(x)}{E^*}\left(1-\frac{\Delta}{(1-\eta)|E^*|}\right)\right). \tag{4.198}$$

For the $\Leftarrow$ direction, consider taking $c_t \to 0, t \neq 0$ and $c_1 \to 1$. By continuity we have

$$\sum_y P(x,y) f\left(\frac{H(y)}{E^*}\right) \geq f\left(\frac{(1-\alpha_P)H(x)}{E^*}\right) \tag{4.199}$$

We can suppose $H(x) < -(1-\eta)|E^*|$, since otherwise the right-hand side is zero, and so our choice of Δ in terms of α_P clearly works. Thus,

$$\sum_y P(x,y) f\left(\frac{H(y)}{E^*}\right) \geq f\left(\frac{H(x)+\alpha_P(1-\eta)|E^*|}{E^*}\right). \tag{4.200}$$

so by definition of f

$$\sum_y P(x,y) g_\eta\left(|E^*|^{-1}H(y)\right) \leq g_\eta\left(|E^*|^{-1}(H(x)+\alpha_P(1-\eta)|E^*|)\right). \tag{4.201}$$

□

Lemma 4.7.5 (Upper bounds on $\Delta(\eta)$)**.** *Let $\eta \in [0,1)$. If H has P pseudo-Lipschitz norm $\|H\|_P$, then*

$$\sqrt{\|H\|_P} \geq \Delta_P(\eta). \tag{4.202}$$

Furthermore, if

$$\mathbb{E}_{y \underset{P}{\sim} x}[H(y)] \leq H_c(x) + \tilde{\Delta}_P, \tag{4.203}$$

then for $\eta \in [0, 1)$

$$\tilde{\Delta}_P \geq \Delta_P(\eta). \tag{4.204}$$

Proof. Note that if have the stronger condition, for some $\tilde{\Delta}_P > 0$,

$$\mathbb{E}_{y \underset{P}{\sim} x}[H(y)] \leq H_c(x) + \tilde{\Delta}_P, \tag{4.205}$$

which may actually be easier to show, then we also have $\Delta_P(\eta)$ stability with $\Delta_P(\eta) \leq \tilde{\Delta}_P$, since by concavity of $h_\eta = g_\eta(\frac{x}{|E^*|})$ and Jensen's inequality:

$$\mathbb{E}_{y \underset{P}{\sim} x}[h_\eta \left(|E^*|^{-1} H(y) \right)] \leq h_\eta \left(\sum_y P(x, y) H(y)) \right) \leq h_\eta(H(x) + \tilde{\Delta}_P). \tag{4.206}$$

Also from the above, it is a simple consequence of Jensen's inequality that we can take $\Delta_P(\eta)$ to be the $\sqrt{\|H\|_P}$:

$$\sqrt{\|\psi\|_P} = \sqrt{\max_{x \in \mathcal{X}} \sum_y P(y, x)(H(x) - H(y))^2} \tag{4.207}$$

$$\geq \max_{x \in \mathcal{X}} \sum_y P(y, x)|H(x) - H(y)| \tag{4.208}$$

$$\geq \tilde{\Delta}_P \tag{4.209}$$

$$\geq \Delta_P(\eta), \quad \forall \eta \in [0, 1). \tag{4.210}$$

□

4.8 Technical Details for MaxCut Hamming and MaxBisection

Lemma 4.8.1. *Let* $G(\mathcal{N}, \mathcal{E})$ *be drawn from the Erdős-Rényi ensemble* $\mathcal{G}\left(n, \frac{p}{n-1}\right)$ *for a constant* p*. Consider the objective of* (MaxCut-Hamming)*:*

$$H(x) := -\frac{1}{2} \sum_{i<j} e_{ij}(1 - x_i x_j). \tag{4.211}$$

Then, with probability at least $1-\delta$ over the graph, it follows:

$$\left|-\mathbb{E}_\pi H(x) - \frac{pk(n-k)}{n}\right| \leq C\log\left(\delta^{-1}\right)\frac{2\sqrt{p}k(n-k)}{n\sqrt{2(n-1)}}, \tag{4.212}$$

where $C>0$ is an arbitrary constant, and π is the uniform distribution of Hamming-weight k strings.

Proof. The shift required to ensure that the mean over in-constraint strings is zero is given by

$$-\mathbb{E}_\pi H(x) = \sum_{i<j} e_{ij}\mathbb{E}_\pi \frac{1}{2}(1-x_i x_j) = \sum_{i<j} e_{ij}\Pr[x_i \neq x_j]. \tag{4.213}$$

Note that there are $\binom{n}{k}$ bitstrings of Hamming weight k. If $x_i \neq x_j$, we have the freedom to place $k-1$ "-1"s in $n-2$ spots. Adding a factor of two since the same can be done for "$+1$"s, we get

$$\Pr[x_i \neq x_j] = \frac{2\binom{n-2}{k-1}}{\binom{n}{k}} = \frac{2k(n-k)}{n(n-1)}. \tag{4.214}$$

Suppose edge creation probability is $\frac{p}{n}$. Since $\sum_{i<j} e_{ij}$ a Binomial random variable $B\left(\binom{n}{2}, \frac{p}{n}\right)$, applying the Chernoff bound asserts that with probability at least $1-\delta$, we have:

$$\left|\sum_{i<j} e_{ij} - \binom{n}{2}\frac{p}{n}\right| \leq C\log\left(\delta^{-1}\right)\sqrt{\binom{n}{2}\frac{p}{n}\left(1-\frac{p}{n}\right)}, \tag{4.215}$$

where $C>0$ is an arbitrary constant.

Accordingly, with probability $1-\delta$, it follows:

$$\left|\mathbb{E}_\pi H(x) - \binom{n}{2}\frac{2k(n-k)}{n(n-1)}\frac{p}{n}\right| \leq C\log\left(\delta^{-1}\right)\frac{2k(n-k)}{n(n-1)}\sqrt{\binom{n}{2}\frac{p}{n}\left(1-\frac{p}{n}\right)} \tag{4.216}$$

$$\implies \left|\mathbb{E}_\pi H(x) - \frac{pk(n-k)}{n}\right| \leq C\log\left(\delta^{-1}\right)\frac{2\sqrt{p}k(n-k)}{n\sqrt{2(n-1)}}. \tag{4.217}$$

□

Thus, for $k = \frac{n}{2}$ one has

$$\left|\mathbb{E}_\pi H(x) - \frac{pn}{4}\right| \leq C\log\left(\delta^{-1}\right)\frac{\sqrt{p}n}{2\sqrt{2(n-1)}} \implies \left|\frac{\mathbb{E}_\pi H(x)}{n} - \frac{p}{4}\right| \leq C\log\left(\delta^{-1}\right)\frac{\sqrt{p}}{2\sqrt{2(n-1)}}, \tag{4.218}$$

from which one can conclude

$$-\frac{\mathbb{E}_\pi H(x)}{n} = \frac{p}{4}(1 + o(1)) \tag{4.219}$$

with high probability. Thus for $k = n/2$, $|E^*| = \Theta(n)$ so the cost function only gets shifted by constant to make it mean zero. More generally, for $k = o(n)$, with high probability

$$-\mathbb{E}_\pi H(x) = pk(1 + o(1)). \tag{4.220}$$

Lemma 4.8.2. *Let H be the cost function of* (MaxCut-Hamming)*:*

$$H(x) := -\sum_{i<j} e_{ij}\frac{(1 - x_i x_j)}{2}, \tag{4.221}$$

and P be the transition matrix for the transposition walk on the space Hamming-weight k bitstrings. Then, for an arbitrary graph G with $|\mathcal{E}|$ edges and Hamming-weight k MaxCut $\mathcal{C}_k^*$, it follows:

$$\frac{\mathcal{C}_k^*(n-2)}{k(n-k)} \geq \mathbb{E}_{y\sim x}[H(y)] - H(x) \geq \frac{\mathcal{C}_k^*(n-2)}{k(n-k)} - (|\mathcal{E}| - \mathcal{C}_k^*)\max\left\{\frac{2}{k}, \frac{2}{n-k}\right\}, \tag{4.222}$$

for every $x \in \{u \in \{-1,1\}^n : |u| = k\}$.

Proof. Fix a graph G, and consider an arbitrary $x \in \{-1,1\}^n$ such that $|x| = k$ corresponding to number of $+1$'s. Swaps only occur between x_i and x_j that are different. All one step transitions denoted by $\sim$ are implied to be under the transposition walk.

$$H(x) = -\sum_{i<j} e_{ij}\frac{(1 - x_i x_j)}{2} = -\sum_{i<j \mid x_i \neq x_j} e_{ij}, \tag{4.223}$$

$$\mathbb{E}_{y\sim x}[H(y)] = -\sum_{i<j} e_{ij}\mathbb{E}_{y\sim x}\left[\frac{1 - y_i y_j}{2}\right], \tag{4.224}$$

$$\mathbb{E}_{y\sim x}\left[\frac{1 - y_i y_j}{2}\right] = \mathbb{P}[y_i \neq y_j | x]. \tag{4.225}$$

For a given fixed $x \in \{-1,1\}^n$ satisfying $|x| = k$, we know there are $k(n-k)$ pairs of indices such that $x_i \neq x_j$, $\binom{n-k}{2}$ indices where $x_i = x_j = -1$, and $\binom{k}{2}$ indices where

$x_i = x_j = 1$. Thus we can decompose the expectation as follows:

$$\mathbb{E}_{y\sim x}[H] = -\sum_{i<j\ :\ x_i\neq x_j} e_{ij}\mathbb{P}_{y\sim x}[y_i \neq y_j | x_i \neq x_j] - \sum_{i<j\ :\ x_i=x_j=1} e_{ij}\mathbb{P}_{y\sim x}[y_i \neq y_j | x_i = x_j = 1] - \sum_{i<j\ :\ x_i=x_j=-1} e_{ij}\mathbb{P}_{y\sim x}[y_i \neq y_j | x_i = x_j = -1]. \tag{4.226}$$

Using the following facts:

$$\mathbb{P}_{y\sim x}[y_i \neq y_j | x_i \neq x_j] = \frac{(k-1)(n-k-1)+1}{k(n-k)} \tag{4.227}$$

$$\mathbb{P}_{y\sim x}[y_i \neq y_j | x_i = x_j = 1] = \frac{2}{k} \tag{4.228}$$

$$\mathbb{P}_{y\sim x}[y_i \neq y_j | x_i = x_j = -1] = \frac{2}{n-k}. \tag{4.229}$$

the expression (4.226) simplifies to

$$\mathbb{E}_{y\sim x}[H] = -\sum_{i<j\ :\ x_i\neq x_j} e_{ij}\frac{(k-1)(n-k-1)+1}{k(n-k)} - \sum_{i<j\ :\ x_i=x_j=1} e_{ij}\frac{2}{k} - \sum_{i<j\ :\ x_i=x_j=-1} e_{ij}\frac{2}{n-k}. \tag{4.230}$$

As a consequence, for all x with Hamming weight k,

$$\mathbb{E}_{y\sim x}[H] - H(x)$$

$$= -\sum_{i<j\ :\ x_i\neq x_j} e_{ij}\left(\frac{(k-1)(n-k-1)+1}{k(n-k)} - 1\right) - \sum_{i<j\ :\ x_i=x_j=1} e_{ij}\frac{2}{k} - \sum_{i<j\ :\ x_i=x_j=-1} e_{ij}\frac{2}{(n-k)} \tag{4.231}$$

$$= \sum_{i<j\ :\ x_i\neq x_j} e_{ij}\frac{n-2}{k(n-k)} - \sum_{i<j\ :\ x_i=x_j=1} e_{ij}\frac{2}{k} - \sum_{i<j\ :\ x_i=x_j=-1} e_{ij}\frac{2}{(n-k)}. \tag{4.232}$$

Thus for any $x \in \{-1,1\}^n$ and any graph G:

$$\mathbb{E}_{y\sim x}[H] \leq H(x) + \frac{-H(x)(n-2)}{k(n-k)}, \tag{4.233}$$

We have:

$$\forall x, \mathbb{E}_{y\sim x}[H] \leq H(x) + \frac{\mathcal{C}_k^*(n-2)}{k(n-k)}. \tag{4.234}$$

For the lower bound, for every $x \in \{u \in \{-1,1\}^n : |u| = k\}$, we have:

$$\mathbb{E}_{y\sim x}[H(y)] - H(x) \geq \frac{\mathcal{C}_k^*(n-2)}{k(n-k)} - (|\mathcal{E}| - \mathcal{C}_k^*) \max\left\{\frac{2}{k}, \frac{2}{n-k}\right\}, \tag{4.235}$$

where $|\mathcal{E}|$ is the number of edges. □

Lemma 4.8.3. *For the MaxCut-Hamming Hamiltonian H, the pseudo Lipschitz constant $\|H\|_P$ under the transposition walk is $\mathcal{O}(1)$ with high probability.*

Proof. Recall

$$\|H(x)\|_P := \mathbb{E}_{\substack{y\sim x \\ P}}[(H(y) - H(x))^2], \tag{4.236}$$

so we can consider

$$\mathbb{E}_{\substack{y\sim x \\ P}}[H(y)^2] - H(x)(2\mathbb{E}_{\substack{y\sim x \\ P}}[H(y)] - H(x)). \tag{4.237}$$

The only new term is $\mathbb{E}_{y\sim x}[H(y)^2]$, which requires us to look at terms like:

$$e_{ij}e_{rs}\mathbb{E}_{\substack{y\sim x \\ P}}\left[\frac{1-y_iy_j}{2}\frac{1-y_ry_s}{2}\right], \tag{4.238}$$

y is x after a single random transposition. We can put these terms in groups based on:

1. $x_j = x_i = x_r = x_s = \pm 1$, the term is always zero
2. $x_j = x_i = x_r \neq x_s = \pm 1$

$$-1 : \mathbb{E}_{y\sim x}\left[\frac{1-y_iy_j}{2}\frac{1-y_ry_s}{2}\right] = \frac{2(n-k-1)}{k(n-k)} \tag{4.239}$$

$$+1 : \mathbb{E}_{y\sim x}\left[\frac{1-y_iy_j}{2}\frac{1-y_ry_s}{2}\right] = \frac{2(k-1)}{k(n-k)} \tag{4.240}$$

3. $x_j = x_i \neq x_r = x_s = \pm 1$ the term is zero unless an element of $\{i,j\}$ is swapped with an element of $\{r,s\}$, giving

$$\pm 1 : \mathbb{E}_{y\sim x}\left[\frac{1-y_iy_j}{2}\frac{1-y_ry_s}{2}\right] = \frac{4}{(n-k)k} \tag{4.241}$$

4. $x_j \neq x_i = x_r \neq x_s = \pm 1$,

$$\mathbb{E}_{y\sim x}\left[\frac{1-y_iy_j}{2}\frac{1-y_ry_s}{2}\right] = \frac{(n-k-2)(k-2)+2}{k(n-k)} = \frac{k(n-k)-2(n-3)}{k(n-k)} \tag{4.242}$$

Let $\hat{y}_{ij} = \frac{1-y_i y_j}{2}$, then expanding:

$$H(y) = -\sum_{i<j \,:\, x_i \neq x_j} e_{ij}\hat{y}_{ij} - \sum_{i<j \,:\, x_i = x_j = 1} e_{ij}\hat{y}_{ij} - \sum_{i<j \,:\, x_i = x_j = -1} e_{ij}\hat{y}_{ij} \tag{4.243}$$

$$\begin{aligned}
\mathbb{E}_{y\sim x}[H(y)^2] = \mathbb{E}_{y\sim x}&\left\{\left(\sum_{i<j \,:\, x_i \neq x_j,} e_{ij}\hat{y}_{ij}\right)^2\right\} \\
&+ 2\mathbb{E}_{y\sim x}\left\{\left(\sum_{i<j \,:\, x_i = x_j = 1} e_{ij}\hat{y}_{ij}\right)\left(\sum_{r<s \,:\, x_r \neq x_s} e_{ij}\hat{y}_{ij}\right)\right\} \\
&+ 2\mathbb{E}_{y\sim x}\left\{\left(\sum_{i<j \,:\, x_i = x_j = -1} e_{ij}\hat{y}_{ij}\right)\left(\sum_{r<s \,:\, x_r \neq x_s} e_{ij}\hat{y}_{ij}\right)\right\} \\
&+ 2\mathbb{E}_{y\sim x}\left\{\left(\sum_{i<j \,:\, x_i = x_j = -1} e_{ij}\hat{y}_{ij}\right)\left(\sum_{r<s \,:\, x_r = x_s = 1} e_{ij}\hat{y}_{ij}\right)\right\},
\end{aligned} \tag{4.244}$$

where we have eliminated the square of the $x_i = x_j = \pm 1$ terms since they fall into group 1. Expanding further and passing the expectations through:

$$\begin{aligned}
\mathbb{E}_{y\sim x}[H(y)^2] = &\sum_{i<j \,:\, x_i \neq x_j} e_{ij}\mathbb{E}_{y\sim x}\hat{y}_{ij} \\
&+ 2\sum_{i<j, r<s, (i,j)\neq(r,s) \,:\, x_i \neq x_j,\ x_r \neq x_s} e_{ij}e_{rs}\mathbb{E}_{y\sim x}[\hat{y}_{ij}\hat{y}_{rs}] \\
&+ 2\sum_{i<j, r<s \,:\, x_i = x_j = 1, x_r \neq x_s} e_{ij}e_{rs}\mathbb{E}_{y\sim x}[\hat{y}_{ij}\hat{y}_{rs}] \\
&+ 2\sum_{i<j, r<s \,:\, x_i = x_j = -1, x_r \neq x_s} e_{ij}e_{rs}\mathbb{E}_{y\sim x}[\hat{y}_{ij}\hat{y}_{rs}] \\
&+ 2\sum_{i<j, r<s \,:\, x_i = x_j = 1, x_r = x_s = -1} e_{ij}e_{rs}\mathbb{E}_{y\sim x}[\hat{y}_{ij}\hat{y}_{rs}].
\end{aligned} \tag{4.245}$$

Next we compute the expectations by identifying which group they belong to:

$$\begin{aligned}
\mathbb{E}_{y\sim x}[H(y)^2] = &\frac{k(n-k)-(n-2)}{k(n-k)} \sum_{i<j \,:\, x_i \neq x_j} e_{ij} \\
&+ \frac{2k(n-k)-4(n-3)}{k(n-k)} \sum_{i<j, r<s, (i,j)\neq(r,s) \,:\, x_i \neq x_j,\ x_r \neq x_s} e_{ij}e_{rs} \\
&+ \frac{4(n-k-1)}{k(n-k)} \sum_{i<j, r<s \,:\, x_i = x_j = 1, x_r \neq x_s} e_{ij}e_{rs} \\
&+ \frac{4(k-1)}{k(n-k)} \sum_{i<j, r<s \,:\, x_i = x_j = -1, x_r \neq x_s} e_{ij}e_{rs}
\end{aligned}$$

$$+ \frac{8}{(n-k)k} \sum_{i<j, r<s \ : \ x_i = x_j = 1, x_r = x_s = -1} e_{ij} e_{rs}. \tag{4.246}$$

The only component to clarify is the first sum, which follows from a calculation in the previous lemma

$$\mathbb{E}_{y \underset{P}{\sim} x}[\hat{y} | x_i \neq x_j] = \frac{(k-1)(n-k-1)+1}{k(n-k)} = \frac{k(n-k)-(n-2)}{k(n-k)}. \tag{4.247}$$

For the $2\mathbb{E}_{y \sim x}[H] - H(x)$ part, we use the same computations from the previous lemma, where we computed $\mathbb{E}_{y \sim x}[H] - H(x)$:

$$2\mathbb{E}_{y \sim x}[H] - H(x)$$
$$= - \sum_{i<j \ : \ x_i \neq x_j} e_{ij} \left(\frac{2(k-1)(n-k-1)+2}{k(n-k)} - 1 \right) - \sum_{i<j \ : \ x_i = x_j = 1} e_{ij} \frac{4}{k} - \sum_{i<j \ : \ x_i = x_j = -1} e_{ij} \frac{4}{(n-k)} \tag{4.248}$$
$$= \sum_{i<j \ : \ x_i \neq x_j} e_{ij} \frac{-k(n-k)+2(n-2)}{k(n-k)} - \sum_{i<j \ : \ x_i = x_j = 1} e_{ij} \frac{4}{k} - \sum_{i<j \ : \ x_i = x_j = -1} e_{ij} \frac{4}{(n-k)}, \tag{4.249}$$

where recall that

$$\mathbb{E}_{y \underset{P}{\sim} x}[\hat{y} | x_i = x_j = 1] = \frac{2}{k} \tag{4.250}$$
$$\mathbb{E}_{y \underset{P}{\sim} x}[\hat{y} | x_i = x_j = -1] = \frac{2}{n-k}. \tag{4.251}$$

Then we compute $-H(x)[2\mathbb{E}_{y \sim x}[H] - H(x)]$:

$$\begin{aligned}
&- H(x)[2\mathbb{E}_{y \sim x}[H] - H(x)] \\
&= \sum_{i<j: \ x_i \neq x_j} e_{ij} \frac{-k(n-k)+2(n-2)}{k(n-k)} \\
&+ \sum_{i<j, r<s, (i,j) \neq (r,s): \ x_i \neq x_j, x_r \neq x_s} e_{rs} e_{ij} \frac{-2k(n-k)+4(n-2)}{k(n-k)} \\
&- \sum_{i<j, r<s \ : \ x_i = x_j = 1, x_r \neq x_s} e_{rs} e_{ij} \frac{4}{k} \\
&- \sum_{i<j, r<s: \ x_i = x_j = -1, x_r \neq x_s} e_{rs} e_{ij} \frac{4}{(n-k)}.
\end{aligned} \tag{4.252}$$

We can now put all the expressions together to get:

$$\begin{aligned}\mathbb{E}_{y\sim x}[H(y)^2] - H(x)(2\mathbb{E}_{y\sim x}[H(y)] - H(x)) = &\frac{(n-2)}{k(n-k)} \sum_{i<j \,:\, x_i \neq x_j} e_{ij} \\ &+ \frac{4}{k(n-k)} \sum_{i<j, r<s, (i,j)\neq(r,s) \,:\, x_i \neq x_j,\ x_r \neq x_s} e_{ij}e_{rs} \\ &- \frac{4}{k(n-k)} \sum_{i<j, r<s \,:\, x_i = x_j = 1, x_r \neq x_s} e_{ij}e_{rs} \\ &- \frac{4}{k(n-k)} \sum_{i<j, r<s \,:\, x_i = x_j = -1, x_r \neq x_s} e_{ij}e_{rs} \\ &+ \frac{8}{k(n-k)} \sum_{i<j, r<s \,:\, x_i = x_j = 1, x_r = x_s = -1} e_{ij}e_{rs}.\end{aligned} \tag{4.253}$$

Let X be a Binomial random variable $\mathcal{B}(M, q)$, then

$$\mathbb{P}[X^2 \geq (1+\delta)^2 \mathbb{E}[X^2]] \leq e^{-\frac{\delta^2 Mq}{2+\delta}}. \tag{4.254}$$

This follows simply from Jensen's inequality

$$\begin{aligned}\mathbb{P}[X^2 \geq (1+\delta)^2 \mathbb{E}[X^2]] &\leq \mathbb{P}[X^2 \geq (1+\delta)^2 (\mathbb{E}[X])^2] &(4.255)\\ &= \mathbb{P}[X \geq (1+\delta)\mathbb{E}[X]] &(4.256)\\ &\leq e^{-\frac{\delta^2 \mathbb{E}[X]}{2+\delta}}, &(4.257)\end{aligned}$$

where the last inequality is the multiplicative Chernoff bound. Also for X and Y being independent Binomials we have that:

$$\mathbb{P}[XY \geq (1+\delta)\mathbb{E}[XY]] \leq 2e^{-\frac{\delta^2 \min(\mathbb{E}[X], \mathbb{E}[Y])}{2+\delta}}, \tag{4.258}$$

which follows from

$$\mathbb{P}[XY \geq (1+\delta)\mathbb{E}[XY]] \leq \mathbb{P}[X \geq (1+\delta/2)\mathbb{E}[X] \vee Y \geq (1+\delta/2)\mathbb{E}[Y]], \tag{4.259}$$

which follows from

$$\begin{aligned}(X \geq (1+\delta/3)\mathbb{E}[X]) \wedge (Y \geq (1+\delta/3)\mathbb{E}[Y]) &\implies XY \leq (1+\delta)\mathbb{E}[X]\mathbb{E}[Y] &(4.260)\\ &= XY \leq (1+\delta)\mathbb{E}[XY], &(4.261)\end{aligned}$$

so union bound gives the desired result.

Thus, we can assume with high probability all of the sums of Bernoulli's in Equation (4.253) are constant factors from their means. Their means are

1. $\mathbb{E}[\sum_{i<j \,:\, x_i \neq x_j} e_{ij}] \frac{pk(n-k)}{n-1} \asymp pk$
2. $\mathbb{E}[\sum_{i<j, r<s, (i,j)\neq(r,s) \,:\, x_i \neq x_j,\ x_r \neq x_s} e_{ij} e_{rs}] = [\frac{p}{n-1} k(n-k)]^2 \asymp p^2 k^2$
3. $\mathbb{E}[\sum_{i<j, r<s \,:\, x_i = x_j = 1, x_r \neq x_s} e_{ij} e_{rs}] = [\frac{p}{n-1}]^2 \binom{k}{2} k(n-k) \asymp p^2 \frac{k^3}{n}$
4. $\mathbb{E}[\sum_{i<j, r<s \,:\, x_i = x_j = -1, x_r \neq x_s} e_{ij} e_{rs}] = [\frac{p}{n-1}]^2 \binom{n-k}{2} k(n-k) \asymp p^2 kn$
5. $\mathbb{E}[\sum_{i<j, r<s \,:\, x_i = x_j = 1, x_r = x_s = -1} e_{ij} e_{rs}] = [\frac{p}{n-1}]^2 \binom{k}{2} \binom{n-k}{2} \asymp p^2 k^2$.

Plugging the above asymptotics in for the Binomials in Equation (4.253) suffices to obtain $\|H\|_P = \mathcal{O}(1)$.

□

Lemma 4.8.4. *With high probability, the optimal objective value of* (MaxCut-Hamming) *satisfies:*

$$\mathcal{C}_k^* = \begin{cases} o(k \log(n)) & \text{if } k = o(n), \\ \mathcal{O}(n) & \text{if } k = \Theta(n), \end{cases} \tag{4.262}$$

where k is the Hamming weight.

Proof. We can use the indicator trick to try to bound the probability of a cut set of a given size. Let $z \in \{0,1\}^n$, $|z| = k$ and $I_{z,m}$ be a Bernoulli random variable indicating whether there are m edges cut with assignment z over the random choice of graph. Then

$$X_m := \sum_{z \in \{0,1\}^n, |z|=k} I_{z,m} \tag{4.263}$$

is the number of in-constraint cuts of size m. Note that $I_{z,m}$ are not independent. For any graph G drawn from $\mathcal{G}(n, p/n)$, we have

$$\mathbb{P}[I_{z,m} = 1] = \binom{k(n-k)}{m} (p/n)^m (1 - p/n)^{k(n-k)-m}. \tag{4.264}$$

The first moment method gives:

$$\mathbb{P}[X_m > 0] \leq \mathbb{E}[X_m] \leq \binom{n}{k} \binom{k(n-k)}{m} n^{-m}. \tag{4.265}$$

Suppose that $k = o(n) \cap \omega(1)$ and $m = \mathcal{O}(n)$. Then $k(n-k) = \mathcal{O}(nk)$, and we can apply the following asymptotics:

$$\mathbb{P}[X_m > 0] \leq \binom{n}{k}\binom{k(n-k)}{m} n^{-m} \asymp \left(\frac{n}{k}\right)^k \left(\frac{nk}{m}\right)^m n^{-m} = n^k k^{m-k} m^{-m}. \tag{4.266}$$

The goal is to try to identify the phase transition point at which the probability goes to zero asymptotically. We can look at

$$k\log(n) + m\log(k) - k\log(k) - m\log(m). \tag{4.267}$$

Upon taking $m = \Theta(k \cdot r)$, we obtain

$$k\log(n) + m\log(k) - k\log(k) - m\log(m) = k\log\left(\frac{n}{kr^r}\right) \tag{4.268}$$

Transition is at $\log(n/k) = r\log(r)$. For $r = \log(n)$, $\mathbb{P}[X_m > 0] \to 0$, thus $C_k^* = o(k\log(n))$.

Suppose $k = \Theta(n)$, then $\log\binom{n}{k} \asymp \mathcal{H}(\frac{k}{n})n$, where $\mathcal{H}$ is the binary entropy function. Thus

$$\mathbb{P}[X_m > 0] \leq \binom{n}{k}\binom{k(n-k)}{m} n^{-m} \asymp 2^{\mathcal{H}(n/k)n} \left(\frac{nk}{m}\right)^m n^{-m} = 2^{\mathcal{H}(n/k)n} k^m m^{-m}. \tag{4.269}$$

We can look at

$$\mathcal{H}(n/k)n - m\log(k/m), \tag{4.270}$$

so transition is at $m = 2^{\mathcal{H}(n/k)}k = \Theta(n)$. Thus for $k = \Theta(n)$, $\mathcal{C}_k^* = \mathcal{O}(n)$. □

4.9 Constrained Short Path via Penalized Objective

Suppose we want to solve the following constrained problem

$$\min_{x \in \mathcal{X} \subseteq \{-1,1\}^n} H(x).$$

Suppose we also have a CSP

$$\mathcal{C}(x) = \sum_{\ell=1}^{m} C_\ell(x), \tag{4.271}$$

with m constraints C_ℓ, indicating the in-constraint solutions. Suppose C_ℓ has k_ℓ literals and has s_ℓ satisfying assignments, then

$$\mathcal{C}_\ell(x) = \begin{cases} -\frac{1}{s_\ell} & \text{if x satisfies constraint } \ell \\ \frac{1}{2^{k_\ell} - s_\ell} & \text{otherwise.} \end{cases} \tag{4.272}$$

Consider the task of minimizing the penalized Hamiltonian

$$\tilde{H}(x) = \frac{H(x)}{\|H\|_2} + \mathcal{C}(x). \tag{4.273}$$

The Markov Chain $\mathcal{M} = (\mathcal{X}, P, \pi)$ is the random walk on the hypercube, and hence π is the uniform distribution over $\{-1, 1\}^n$. Thus, the short-path Hamiltonian is the same as [DPCB23] but with the penalized cost Hamiltonian.

The global minimum of Equation (4.273) is in fact the in-constraint minimum and $\mathbb{E}_\pi[\tilde{H}(x)] = \mathbb{E}_\pi[\frac{H(x)}{\|H\|_2}]$. Also note that $|E^*| = \Theta(m)$ for $\tilde{H}$. The below results will show that due to the normalization, the properties of the CSP are the only components that matter for determining the runtime. We denote $k = \max k_\ell$.

Lemma 4.9.1. *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be the random walk on the Hamming hypercube. The penalized Hamiltonian is Δ_P stable with*

$$\Delta_P(\eta) = \mathcal{O}\left(\frac{mk}{n}\right). \tag{4.274}$$

Proof.

$$\Delta_P(\eta) \leq \frac{\mathbb{E}_{y\sim}[H(y) - H(x)]}{\|H\|_2} + \frac{mk}{n} = \mathcal{O}\left(\frac{mk}{n}\right). \tag{4.275}$$

□

Lemma 4.9.2 (Tail bound condition for Penalized Hamiltonian)**.** *Suppose $\mathbb{E}_\pi[\tilde{H}_c] = 0$,*

$$\pi((1-\eta)E^*) \leq 2^{-\gamma}, \quad \gamma = \Omega\left[\frac{2(1-\eta)^2}{\ell^2}\right]. \tag{4.276}$$

Proof. Suppose flipping one variable x_i changes at most Δ_i in the value of H while holding other variables constant. Then

$$\Delta_i = \mathcal{O}\left(\frac{mk}{n}\right). \tag{4.277}$$

Assuming $\mathbb{E}_\pi[H(x)] = 0$. Let $C(E)$ denote the number of x with energy $\leq E$ under $\tilde{H}$. For $E < 0$, $C(E)/2^n$ is the probability for a random x, $H(x)$ deviates below $\mathbb{E}_\pi[H(x)] = 0$ by amount at least $|E|$. The P-pseudo Lipschitz norm of $\tilde{H}$ is bounded by

$$\|\tilde{H}\|_P = \sum_{i=1}^{m} \frac{\Delta_i^2}{n} = \mathcal{O}\left(\frac{m^2k^2}{n^2}\right). \tag{4.278}$$

Also, $\omega = \frac{2}{n}$ for the hypercube walk, and thus the Herbst argument implies

$$\pi(E) \leq e^{-\Omega\left(\frac{2nE^2}{(mk)^2}\right)}. \tag{4.279}$$

Taking $E = E^*(1-\eta)$, we have

$$\pi((1-\eta)E^*) \leq 2^{-n\gamma}, \quad \gamma = \Omega\left[\left(\frac{|E^*|}{m}\right)^2 \frac{2(1-\eta)^2}{k^2}\right]. \tag{4.280}$$

However $|E^*| = \Theta(m)$. □

The following result is then evident from Theorem 4.4.3, given that from [DPCB23] we have that b^* is constant for the random walk on the Hamming cube if γ is constant.

Theorem 4.9.3. *Let $\mathcal{M} = (\mathcal{X}, P, \pi)$ be random walk on the n-bit Hamming cube. Let $H : \{-1,1\}^n \mapsto \mathbb{R}$ be a diagonal Hamiltonian with ground state energy*

$$E^* := \min_{x \in \mathcal{X}} H(x).$$

If the number of constraints is $m = \Theta(n)$, then there exists a short-path algorithm with runtime

$$\mathcal{O}^*\left(2^{n\left(\frac{1}{2} - \frac{(1-\eta)|E^*|b}{n2\ln(2)\Delta_P}\right)}\right). \tag{4.281}$$

The penalty method ensures that our algorithm only outputs feasible solutions, but the framework only provides a speedup over unconstrained brute force search. Hence, the generalized short-path algorithm is significantly more effective.

4.10 Details about Mixer Implementation

4.10.1 Block-encoding the Glauber Mixer

Since the Glauber dynamics mixer is a symmetric sparse matrix, we can implement the block-encoding in polynomial time by assuming sparse oracle access to the non-zero entries of P. Let s be the maximum number of non-zero entries in any row of D. Then, we can implement the following oracle,

$$O_S \left|x\right\rangle \left|0\right\rangle \mapsto \frac{1}{\sqrt{s}} \sum_y \left|x\right\rangle \left|y\right\rangle \tag{4.282}$$

where y is an index of a non-zero entry in $P(x,\cdot)$. Since Glauber dynamics can only update one site at most, the transition matrix contains at most n entries. Hence, we can implement this oracle by computing P at most n times. Define the oracle for access to the elements of P in the following way,

$$O_A \left|x\right\rangle \left|y\right\rangle \left|0\right\rangle \mapsto \left|x\right\rangle \left|y\right\rangle \left(\sqrt{P(x,y)}\left|0\right\rangle + \sqrt{1-P(x,y)}\left|1\right\rangle\right). \tag{4.283}$$

Implementing this oracle takes at most $\mathcal{O}(n)$. Finally, we define SWAP operator,

$$\mathsf{SWAP} \left|a\right\rangle \left|x\right\rangle \left|y\right\rangle \left|b\right\rangle \mapsto \left|y\right\rangle \left|b\right\rangle \left|a\right\rangle \left|x\right\rangle. \tag{4.284}$$

Then, the circuit $O_S^\dagger O_A^\dagger \,\mathsf{SWAP}\, O_A O_S$ implements block-encoding of D/s. To see this, compute

$$\left\langle 0\right| \left\langle 0\right| \left\langle 0\right| \left\langle y\right| O_S^\dagger O_A^\dagger \,\mathsf{SWAP}\, O_A O_S \left|0\right\rangle \left|0\right\rangle \left|0\right\rangle \left|x\right\rangle. \tag{4.285}$$

One has

$$\left|0\right\rangle \left|0\right\rangle \left|0\right\rangle \left|x\right\rangle \xrightarrow{O_S} \frac{1}{\sqrt{s}} \sum_y \left|0\right\rangle \left|y\right\rangle \left|0\right\rangle \left|x\right\rangle \tag{4.286}$$

$$\xrightarrow{O_A} \frac{1}{\sqrt{s}} \sum_y \left|0\right\rangle \left|y\right\rangle \left(\sqrt{P(x,y)}\left|0\right\rangle + \sqrt{1-P(x,y)}\left|1\right\rangle\right) \left|x\right\rangle \tag{4.287}$$

$$\xrightarrow{\mathsf{SWAP}} \frac{1}{\sqrt{s}} \sum_y \left(\sqrt{P(x,y)}\left|0\right\rangle + \sqrt{1-P(x,y)}\left|1\right\rangle\right) \left|x\right\rangle \left|0\right\rangle \left|y\right\rangle. \tag{4.288}$$

On the other hand,

$$\left\langle 0\right| \left\langle 0\right| \left\langle 0\right| \left\langle y\right| O_S^\dagger O_A^\dagger = \frac{1}{\sqrt{s}} \sum_z \left\langle 0\right| \left\langle z\right| \left(\sqrt{P(y,z)}\left|0\right\rangle + \sqrt{1-P(y,z)}\left|1\right\rangle\right) \left|y\right\rangle. \tag{4.289}$$

Therefore,

$$\langle 0| \langle 0| \langle 0| \langle y| O_S^\dagger O_A^\dagger \,\mathsf{SWAP}\, O_A O_S |0\rangle |0\rangle |0\rangle |x\rangle = \frac{1}{s}\sqrt{P(x,y)P(y,x)}. \tag{4.290}$$

Hence, we can implement the block-encoding in at most polynomial time.

4.10.2 Ground State Preparation for Glauber Mixer

We discuss how to prepare the ground state of discriminant operator for Glauber dynamics for hard-core model and Ising model at $\lambda \leq \lambda_c$ and $\beta_c \leq \beta$ respectively. This is all we can afford to prepare since the spectral gap of D falls exponentially fast when $\lambda > \lambda_c$ ($\beta > \beta_c$) due to statistical phase transitions. For simplicity, we consider the hard-core model to explain the idea. However, the details for both models will be given as separate propositions below. Let D_λ be the discriminant matrix of Glauber dynamics at fugacity λ. We use the block-encoding of D_λ and amplitude amplification to prepare its ground state. Although, we can efficiently create block-encoding for D_λ and apply singular value transformations to build a projector to its ground state, amplitude amplification might need exponentially many calls to this projector when we do not have a warm initial state. Instead, we combine classical annealing with quantum singular value transformation to prepare the ground state of D_λ to create sequence of states where each state is warm with respect to the next one. Suppose that we prepare $\pi^{(1)}$, the coherent quantum state corresponding to the ground state of D_{λ_1} where $0 < \lambda_1 < \varepsilon$,

$$\left|\pi^{(1)}\right\rangle = \sum_{x\in\mathcal{X}} \sqrt{\pi_{\lambda_1}(x)}\,|x\rangle \tag{4.291}$$

where the sum is over all independent sets x in G. Next, we increase fugacity to $\lambda_2 = \lambda_1(1+\Delta)$ and prepare,

$$\left|\pi^{(2)}\right\rangle = \sum_{x\in\mathcal{X}} \sqrt{\pi_{\lambda_2}(x)}\,|x\rangle\,. \tag{4.292}$$

This quantum state can be prepared by applying ground state projector of D_{λ_1} to and D_{λ_2} to $\left|\pi^{(0)}\right\rangle$ through fixed point amplitude amplification. We repeat this process until we prepare $\left|\pi^{(k)}\right\rangle$,

$$\left|\pi^{(k)}\right\rangle = \sum_{x\in\mathcal{X}} \sqrt{\pi_{\lambda_k}(x)}\,|x\rangle \tag{4.293}$$

with $\lambda_k = \lambda_c$. We need to show that this process can be done in poly(n) time.

Proposition 4.10.1 (Preparation of Gibbs State for Hard-core Model)**.** *Consider Glauber dynamics chain for hard-core model on graph $G(\mathcal{N}, \mathcal{E})$ with transition matrix P_λ with stationary distribution,*

$$\pi_\lambda(x) = \frac{\lambda^{|x|}}{Z}. \tag{4.294}$$

Let $\delta(\lambda) > 0$ be the spectral gap of the discriminant matrix $D(P_\lambda)$ associated with P_λ. Then, there exists a quantum algorithm that prepares the ground state of $-D$ up to ε accuracy in L2 norm with run time $\widetilde{\mathcal{O}}(\delta_{\min}^{-1/2} \log(1/\varepsilon))$ where $\delta_{\min} = \inf_{0 \le \lambda' \le \lambda} \delta(\lambda')$.

Proof. We start with the following quantum state,

$$\left|\pi^{(0)}\right\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |x_i\rangle \tag{4.295}$$

where x_i is all 0 bit string except location i. This quantum state is ground state of D at $\lambda = 0$ as each x_i is an independent set and they are the only ones with non-zero probability. This quantum state is essentially superposition over all strings with Hemming weight 1. This state can be prepared by applying $\frac{1}{\sqrt{n}} \sum_{i=1}^{n} X_i$ where X_i is Pauli-X applied to the all 0 state. Therefore, this state can be prepared in polynomial time. However, we cannot implement the Glauber dynamics at $\lambda = 0$ as the transition density is not meaningful. Instead, we start with $\left|\pi^{(1)}\right\rangle$ which is the ground state of D_{λ_1} which can be prepared from $\left|\pi^{(0)}\right\rangle$ since $\left|\pi^{(0)}\right\rangle$ and $\left|\pi^{(1)}\right\rangle$ overlaps significantly

$$|\langle \pi^{(0)} | \pi^{(1)} \rangle| = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{\lambda_1^{1/2}}{\sqrt{Z_1}} \ge \frac{\sqrt{n\lambda_1}}{\sqrt{n\lambda_1 + \sum\limits_{|\mathcal{I}|>1} \lambda_1^{|\mathcal{I}|}}} = \Omega(1), \tag{4.296}$$

for $\lambda_1 \le \frac{3n}{2^n}$. Similarly, given $\left|\pi^{(k-1)}\right\rangle$, we can prepare $\left|\pi^{(k)}\right\rangle$ efficiently. The number of calls to the ground state projector by the fixed point amplitude amplification from $\left|\pi^{(k-1)}\right\rangle$ to $\left|\pi^{(k)}\right\rangle$ is $\widetilde{\mathcal{O}}(|\langle \pi^{(k-1)} | \pi^{(k)} \rangle|^{-1})$. The overlap can be calculated as

$$\left|\left\langle \pi^{(k-1)} | \pi^{(k)} \right\rangle\right| = \sum_{x \in \mathcal{X}} \frac{\lambda_{k-1}^{|x|/2}}{\sqrt{Z_{k-1}}} \frac{\lambda_k^{|x|/2}}{\sqrt{Z_k}} \tag{4.297}$$

$$\ge \sum_{x \in \mathcal{X}} \frac{\lambda_{k-1}^{|x|/2} \lambda_k^{|x|/2}}{Z_k} \tag{4.298}$$

$$= \sum_{x \in \mathcal{X}} \frac{\lambda_k^{|x|} \Delta^{-|x|/2}}{Z_k} \tag{4.299}$$

$$\geq (1+\Delta)^{\frac{-n}{2}} \tag{4.300}$$

$$\geq 1 - \frac{n\Delta}{2} \tag{4.301}$$

$$= \Omega(1), \tag{4.302}$$

if we set $\Delta \leq \frac{2}{n}$. Hence, starting from $\left|\pi^{(1)}\right\rangle$ we can prepare a schedule that maintains constant overlap with the subsequent state. Since we have constant overlap throughout the schedule, each amplitude amplification step only requires constant number of calls to ground state projectors. Also note that implementing the ground state projector requires $\widetilde{\mathcal{O}}(\delta^{-1})$ calls to block-encoding of D [GSLW19]. Finally, we only need to do $\mathcal{O}(\text{poly}(n))$ rounds of amplitude amplification since $\lambda_k = \lambda_1 \exp(2k/n)$ and for $k \geq \frac{n}{2}\log(\lambda_c/\lambda_1)$, $\lambda_k \geq \lambda_c$.

□

Proposition 4.10.2 (Preparation of Gibbs State for Ising Model Model)**.** *Consider Glauber dynamics chain for Ising model on graph $G(\mathcal{N}, \mathcal{E})$ with transition matrix P_β with stationary distribution,*

$$\pi_\beta(x) = \frac{\exp(-\beta H(x))}{Z}. \tag{4.303}$$

Let $\delta(\beta) > 0$ be the spectral gap of the discriminant matrix $D(P_\beta)$ associated with P_β. Then, there exists a quantum algorithm that prepares the ground state of $-D$ up to ε accuracy in ℓ_2-norm with run time $\widetilde{\mathcal{O}}(\delta_{\min}^{-1/2}\log(1/\varepsilon))$ where $\delta_{\min} = \inf_{0\leq\beta'\leq\beta}\delta(\beta')$.

Proof. The proof is similar to the hard-core model, and we start with the following quantum state,

$$\left|\pi^{(0)}\right\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \tag{4.304}$$

This quantum state is ground state of D at $\beta = 0$ since for $\beta = 0$, Glauber dynamics is equivalent to hypercube walk. Similar to MIS case, given $\left|\pi^{(k-1)}\right\rangle$, we can prepare $\left|\pi^{(k)}\right\rangle$ up to ε accuracy efficiently. The number of calls to the ground state projector by the fixed point amplitude amplification from $\left|\pi^{(k-1)}\right\rangle$ to $\left|\pi^{(k)}\right\rangle$ is $\widetilde{\mathcal{O}}(|\langle\pi^{(k-1)}|\pi^{(k)}\rangle|^{-1})$. The overlap can be calculated as

$$\left|\langle\pi^{(k-1)}|\pi^{(k)}\rangle\right| = \sum_{x\in G} \frac{\exp(-\beta_{k-1}H(x)/2)}{\sqrt{Z_{k-1}}} \frac{\exp(-\beta_k H(x)/2)}{\sqrt{Z_k}} \tag{4.305}$$

$$\geq \sum_{x\in G} \frac{\exp(-\beta_{k-1}H(x)/2)\exp(-\beta_k H(x)/2)}{Z_k} \tag{4.306}$$

$$= \sum_{x \in G} \frac{\exp(-\beta_k H(x)) \exp((\beta_k - \beta_{k-1}) H(x)/2)}{Z_k} \tag{4.307}$$

$$\geq \exp((\beta_k - \beta_{k-1}) \|H\|/2) \tag{4.308}$$

$$= \Omega(1), \tag{4.309}$$

if we set $(\beta_k - \beta_{k-1}) = \frac{1}{\|H\|}$. Hence, starting from $\left|\pi^{(0)}\right\rangle$ we can prepare a schedule that maintains constant overlap with the subsequent state.

Since we have constant overlap throughout the schedule, each amplitude amplification step only requires constant number of calls to ground state projectors. Also note that implementing the ground state projector requires $\widetilde{\mathcal{O}}(\delta^{-1})$ calls to block-encoding of D [GSLW19]. Finally, we only need to do $\text{poly}(n)$ rounds of amplitude amplification since $\|H\| = \text{poly}(n)$ and length of the schedule is polynomial. □

4.11 Conclusion

In this chapter, we generalized the short-path framework and identified conditions under which this generalized algorithm achieves a runtime of $\mathcal{O}((\pi^*)^{-(0.5-c)})$ for some constant $c > 0$. Our analysis simplifies much of the proof of speedup in [DPCB23], which corresponds to the special case of a random walk on the vertices of a hypercube. We also made explicit connections between the short-path framework and the classical theory of Markov chains, which may be fruitful for further applications and generalizations. Furthermore, we identified various settings where the conditions for speedup are satisfied. Our applications include sampling from the uniform distribution over constrained spaces for solving Max-Bisection and Max-Cut Hamming problems, as well as sampling from Gibbs distributions for optimizing spin Hamiltonians such as the Ising model, SK model, and hardcore model. We also showed that the generalized short-path algorithm is super-quadratically faster than any classical algorithm based on polynomial-time Gibbs sampling (whether or not that algorithm uses Glauber dynamics) for solving the MIS problem on random regular graphs. This provides evidence that it is not generally possible to construct classical algorithms that are at most quadratically slower than short-path algorithms.

Our work shares the limitation of [DPCB23] in that the quantitative improvement in the degree of the speedup that can be rigorously proved remains very small. Although extending the analysis to larger quantitative speedups is an important challenge, we focused mainly on extending the previous algorithms so they apply, in principle, to a larger

class of classical algorithms, and on providing theoretical evidence that super-quadratic speedups persist in this regime. In this sense, we regard the short-path algorithm as a framework for combinatorial optimization whose speedups are likely larger than what can currently be rigorously proved.

Bibliography

[AA14] Scott Aaronson and Andris Ambainis. The need for structure in quantum speedups, 2014.

[AAA+24] Amira Abbas, Andris Ambainis, Brandon Augustino, Andreas Bärtschi, Harry Buhrman, Carleton Coffrin, Giorgio Cortiana, Vedran Dunjko, Daniel J. Egger, Bruce G. Elmegreen, et al. Challenges and opportunities in quantum optimization. *Nature Reviews Physics*, 2024.

[AAKV01] Dorit Aharonov, Andris Ambainis, Julia Kempe, and Umesh Vazirani. Quantum walks on graphs. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, page 50–59, New York, NY, USA, 2001. Association for Computing Machinery.

[Aar22] Scott Aaronson. How much structure is needed for huge quantum speedups?, 2022.

[ABI+19] Andris Ambainis, Kaspars Balodis, Jānis Iraids, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. Quantum speedups for exponential-time dynamic programming algorithms. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1783–1793. SIAM, 2019.

[AHF+25] Brandon Augustino, Dylan Herman, Enrico Fontana, Junhyung Lyle Kim, Jacob Watkins, Shouvanik Chakrabarti, and Marco Pistoia. Fast convex optimization with quantum gradient methods, 2025.

[AHN+21] Srinivasan Arunachalam, Vojtech Havlicek, Giacomo Nannicini, Kristan Temme, and Pawel Wocjan. Simpler (classical) and faster (quantum) algorithms for Gibbs partition functions. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 112–122. IEEE, 2021.

[AJK+22] Nima Anari, Vishesh Jain, Frederic Koehler, Huy Tuan Pham, and Thuy-Duong Vuong. Entropic independence: optimal mixing of down-up random walks. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 1418–1430, New York, NY, USA, 2022. Association for Computing Machinery.

[AK17] Andris Ambainis and Martins Kokainis. Quantum algorithm for tree size estimation, with applications to backtracking and 2-player games. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, page 989–1002, New York, NY, USA, 2017. Association for Computing Machinery.

[AKR10] Boris Altshuler, Hari Krovi, and Jérémie Roland. Anderson localization makes adiabatic quantum optimization fail. *Proceedings of the National Academy of Sciences*, 107(28):12446–12450, 2010.

[AS19] Simon Apers and Alain Sarlette. Quantum fast-forwarding: Markov chains and graph property testing. *Quantum Info. Comput.*, 19(3–4):181–213, March 2019.

[ATS03] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC 2003)*, pages 20–29, 2003.

[BE19] Andreas Bärtschi and Stephan Eidenbenz. Deterministic preparation of dicke states. In L. Gąsieniec, J. Jansson, and C. Levcopoulos, editors, *Fundamentals of Computation Theory*, page 126–139. Springer International Publishing, 2019.

[BEL15] Sébastien Bubeck, Ronen Eldan, and Joseph Lehec. Sampling from a log-concave distribution with projected langevin monte carlo, 2015.

[BFFN19] Jack Baker, Paul Fearnhead, Emily B. Fox, and Christopher Nemeth. Control variates for stochastic gradient mcmc. *Statistics and Computing*, 29(3):599–615, May 2019.

[BG22] Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-order nonconvex stochastic optimization: Handling constraints, high dimensionality, and saddle points. *Found. Comput. Math.*, 22(1):35–76, February 2022.

[BGJM11] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, May 2011.

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation, 2002.

[BLNR15] Alexandre Belloni, Tengyuan Liang, Hariharan Narayanan, and Alexander Rakhlin. Escaping the local minima via simulated annealing: Optimization of approximately convex functions. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 240–265, Paris, France, 03–06 Jul 2015. PMLR.

[BMN+21] Ryan Babbush, Jarrod R. McClean, Michael Newman, Craig Gidney, Sergio Boixo, and Hartmut Neven. Focus beyond Quadratic Speedups for Error-Corrected Quantum Advantage. *PRX Quantum*, 2:010103, 2021.

[Bol81] Béla Bollobás. The independence ratio of regular graphs. *Proceedings of the American Mathematical Society*, 83(2):433–436, 1981.

[Bus82] Peter Buser. A note on the isoperimetric constant. *Annales scientifiques de l'École Normale Supérieure*, Ser. 4, 15(2):213–230, 1982.

[CCC+25] Xiaoyu Chen, Zejia Chen, Zongchen Chen, Yitong Yin, and Xinyuan Zhang. Rapid mixing on random regular graphs beyond uniqueness, 2025.

[CCD+03] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 59–68, New York, NY, USA, 2003. Association for Computing Machinery.

[CCH+23] Shouvanik Chakrabarti, Andrew M. Childs, Shih-Han Hung, Tongyang Li, Chunhao Wang, and Xiaodi Wu. Quantum algorithm for estimating volumes of convex bodies. *ACM Transactions on Quantum Computing*, 4(3), May 2023.

[CCLW20] Shouvanik Chakrabarti, Andrew M. Childs, Tongyang Li, and Xiaodi Wu. Quantum algorithms and lower bounds for convex optimization. *Quantum*, 4:221, January 2020.

[CDT20] Xi Chen, Simon S. Du, and Xin T. Tong. On stationary-point hitting time and ergodicity of stochastic gradient langevin dynamics. *Journal of Machine Learning Research*, 21(68):1–41, 2020.

[CEL+22] Sinho Chewi, Murat A Erdogdu, Mufan Li, Ruoqi Shen, and Shunshi Zhang. Analysis of langevin monte carlo from poincare to log-sobolev. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 1–2. PMLR, 02–05 Jul 2022.

[CFG14] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1683–1691, Bejing, China, 22–24 Jun 2014. PMLR.

[CFM+18] Niladri Chatterji, Nicolas Flammarion, Yian Ma, Peter Bartlett, and Michael Jordan. On the theory of variance reduction for stochastic gradient Monte Carlo. In Jennifer Dy and Andreas Krause, editors, *Proceedings*

of the 35th International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 764–773. PMLR, 10–15 Jul 2018.

[CGJ19] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132, pages 33:1–33:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[CH16] Elizabeth Crosson and Aram W. Harrow. Simulated quantum annealing can be exponentially faster than classical simulated annealing. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 714–723, 2016.

[CH23] Arjan Cornelissen and Yassine Hamoudi. A sublinear-time quantum algorithm for approximating partition functions. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2023)*, pages 1245–1264. Society for Industrial and Applied Mathematics, 2023.

[Cha87] David Chandler. Introduction to modern statistical. *Mechanics. Oxford University Press, Oxford, UK*, 5:449, 1987.

[Cha23] Sourav Chatterjee. Spectral gap of nonreversible markov chains, 2023.

[Che71] Jeff Cheeger. *A Lower Bound for the Smallest Eigenvalue of the Laplacian*, pages 195–200. Princeton University Press, Princeton, 1971.

[CHJ22] Arjan Cornelissen, Yassine Hamoudi, and Sofiene Jerbi. Near-optimal quantum algorithms for multivariate mean estimation. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC '22. ACM, June 2022.

[CHO+24] Shouvanik Chakrabarti, Dylan Herman, Guneykan Ozgul, Shuchen Zhu, Brandon Augustino, Tianyi Hao, Zichang He, Ruslan Shaydulin, and Marco Pistoia. Generalized short path algorithms: Towards super-quadratic speedup over markov chain search for combinatorial optimization, 2024.

[CHW+25] Shouvanik Chakrabarti, Dylan Herman, Jacob Watkins, Enrico Fontana, Brandon Augustino, Junhyung Lyle Kim, and Marco Pistoia. On speedups for convex optimization via quantum dynamics, 2025.

[CKS17] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46(6):1920–1950, 2017.

[CLL+22] Andrew M. Childs, Tongyang Li, Jin-Peng Liu, Chunhao Wang, and Ruizhe Zhang. Quantum algorithms for sampling log-concave distributions and estimating normalizing constants. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23205–23217. Curran Associates, Inc., 2022.

[CLV21] Zongchen Chen, Kuikui Liu, and Eric Vigoda. Optimal mixing of Glauber dynamics: entropy factorization via high-dimensional expansion. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 1537–1550, New York, NY, USA, 2021. Association for Computing Machinery.

[CLV23] Zongchen Chen, Kuikui Liu, and Eric Vigoda. Rapid mixing of glauber dynamics up to uniqueness via contraction. *SIAM Journal on Computing*, 52(1):196–237, 2023.

[CMYP22] Shouvanik Chakrabarti, Pierre Minssen, Romina Yalovetzky, and Marco Pistoia. Universal quantum speedup for branch-and-bound, branch-and-cut, and tree-search algorithms, 2022.

[COLMS22] Amin Coja-Oghlan, Philipp Loick, Balázs F. Mezei, and Gregory B. Sorkin. The Ising Antiferromagnet and Max Cut on Random Regular Graphs. *SIAM Journal on Discrete Mathematics*, 36(2):1306–1342, 2022.

[CPM25] Baptiste Claudon, Jean-Philip Piquemal, and Pierre Monmarché. Quantum speedup for nonreversible markov chains, 2025.

[CS24] Sinho Chewi and Austin J. Stromme. The ballistic limit of the log-sobolev constant equals the polyak-łojasiewicz constant, 2024.

[CV18] Ben Cousins and Santosh Vempala. Gaussian cooling and $o^*(n^3)$ algorithms for volume and gaussian volume. *SIAM Journal on Computing*, 47(3):1237–1273, 2018.

[Dal17a] Arnak S. Dalalyan. Further and stronger analogy between sampling and optimization: Langevin monte carlo and gradient descent. In *Conference on Learning Theory*, pages 678–689. PMLR, 2017.

[Dal17b] Arnak S. Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(3):651–676, 2017.

[DBLJ14] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[DJRW+16] Kumar Avinava Dubey, Sashank J. Reddi, Sinead A Williamson, Barnabas Poczos, Alexander J Smola, and Eric P Xing. Variance reduction in stochastic gradient langevin dynamics. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[DJWW15] John C. Duchi, Michael I. Jordan, Martin J. Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Trans. Inf. Theor.*, 61(5):2788–2806, May 2015.

[DK19] Arnak S. Dalalyan and Avetik Karagulyan. User-friendly guarantees for the langevin monte carlo with inaccurate gradient. *Stochastic Processes and their Applications*, 129(12):5278–5311, 2019.

[DM18] Alain Durmus and Eric Moulines. High-dimensional bayesian inference via the unadjusted langevin algorithm, 2018.

[DMB+23] Alexander M. Dalzell, Sam McArdle, Mario Berta, Przemyslaw Bienias, Chi-Fang Chen, András Gilyén, Connor T. Hann, Michael J. Kastoryano, Emil T. Khabiboulline, Aleksander Kubica, Grant Salton, Samson Wang, and Fernando G. S. L. Brandão. Quantum algorithms: A survey of applications and end-to-end complexities, 2023.

[DMM19] Alain Durmus, Szymon Majewski, and Błażej Miasojedow. Analysis of langevin monte carlo via convex optimization. *Journal of Machine Learning Research*, 20(73):1–46, 2019.

[DMS17] Amir Dembo, Andrea Montanari, and Subhabrata Sen. Extremal cuts of sparse random graphs. *The Annals of Probability*, 45(2):1190–1217, 2017.

[DNR23] Aniket Das, Dheeraj M. Nagaraj, and Anant Raj. Utilising the clt structure in stochastic gradient based sampling : Improved analysis and faster algorithms. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 4072–4129. PMLR, 12–15 Jul 2023.

[DPCB23] Alexander M Dalzell, Nicola Pancotti, Earl T Campbell, and Fernando GSL Brandão. Mind the gap: Achieving a super-grover quantum speedup by jumping to the end. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1131–1144, 2023.

[DSC96] P. Diaconis and L. Saloff-Coste. Logarithmic Sobolev inequalities for finite Markov chains. *The Annals of Applied Probability*, 6(3):695 – 750, 1996.

[DSW07] J. Díaz, M.J. Serna, and N.C. Wormald. Bounds on the bisection width for random d -regular graphs. *Theoretical Computer Science*, 382(2):120–130, 2007. Latin American Theoretical Informatics.

[DV83] Monroe D. Donsker and S.R. Srinivasa Varadhan. Asymptotic evaluation of certain Markov process expectations for large time. IV. *Communications on pure and applied mathematics*, 36(2):183–212, 1983.

[Fey82] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6/7):467–488, 1982.

[FGG02] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. Quantum adiabatic evolution algorithms versus simulated annealing, 2002.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

[FGGS00] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution, 2000. arXiv:quant-ph/0001106.

[FGS06] Fedor V Fomin, Serge Gaspers, and Saket Saurabh. Branching and treewidth based exact algorithms. In *Algorithms and Computation: 17th International Symposium, ISAAC 2006, Kolkata, India, December 18-20, 2006. Proceedings 17*, pages 16–25. Springer, 2006.

[FJ97] Alan Frieze and Mark Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18(1):67–81, 1997.

[FS02] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*, volume 1 of *Computational Science Series*. Academic Press, San Diego, second edition, 2002.

[FYC23] Jiaojiao Fan, Bo Yuan, and Yongxin Chen. Improved dimension dependence of a proximal algorithm for sampling. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 1473–1521. PMLR, 2023.

[GAW19] András Gilyén, Srinivasan Arunachalam, and Nathan Wiebe. Optimizing quantum optimization algorithms via faster quantum gradient computation. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*, pages 1425–1444. SIAM, Philadelphia, PA, 2019.

[Gla63] Roy J. Glauber. Time-dependent statistics of the Ising model. *Journal of Mathematical Physics*, 4(2):294–307, 1963.

[GLL20] Rong Ge, Holden Lee, and Jianfeng Lu. Estimating normalizing constants for log-concave distributions: Algorithms and lower bounds. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC 2020)*, pages 579–586, 2020.

[GLR21] Leslie Ann Goldberg, John Lapinskas, and David Richerby. Faster exponential-time algorithms for approximately counting independent sets. *Theoretical Computer Science*, 892:48–84, 2021.

[Goe04] Sharad Goel. Modified logarithmic sobolev inequalities for some models of random walk. *Stochastic Processes and their Applications*, 114(1):51–79, 2004.

[Gro96a] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery.

[Gro96b] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 212–219, 1996.

[Gro05] Lov K. Grover. Fixed-point quantum search. *Physical Review Letters*, 95(15), 2005.

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, page 193–204, New York, NY, USA, 2019. Association for Computing Machinery.

[Hal07] Sean Hallgren. Polynomial-time quantum algorithms for pell's equation and the principal ideal problem. *J. ACM*, 54(1), March 2007.

[Has18a] M. B. Hastings. Weaker assumptions for the short path optimization algorithm, 2018.

[Has18b] Matthew B. Hastings. A short path quantum algorithm for exact optimization. *Quantum*, 2:78, 2018.

[Has18c] Matthew B. Hastings. Talk on "The short-path algorithm for combinatorial optimization" at Simons Institute, 2018. `https://simons.berkeley.edu/talks/matthew-hastings-06-14-18`.

[Has19] Matthew B. Hastings. The short path algorithm applied to a toy model. *Quantum*, 3:145, 2019.

[HGL+23] Dylan Herman, Cody Googin, Xiaoyuan Liu, Yue Sun, Alexey Galda, Ilya Safro, Marco Pistoia, and Yuri Alexeev. Quantum computing for finance. *Nature Reviews Physics*, 5(8):450–465, 2023.

[HHL09a] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009.

[HHL09b] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009.

[HS87] Richard Holley and Daniel W. Stroock. Logarithmic sobolev inequalities and stochastic ising models. *Journal of Statistical Physics*, 46:1159–1194, 1987.

[HW20] Aram W. Harrow and Annie Y. Wei. Adaptive quantum simulated annealing for bayesian inference and estimating partition functions. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 193–212. SIAM, 2020.

[HZD+24] Xunpeng Huang, Difan Zou, Hanze Dong, Yian Ma, and Tong Zhang. Faster sampling via stochastic gradient proximal sampler. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20559–20596. PMLR, 21–27 Jul 2024.

[Jor05] Stephen P. Jordan. Fast quantum algorithm for numerical gradient estimation. *Physical Review Letters*, 95(5), jul 2005.

[Joz03] Richard Jozsa. Quantum computation in algebraic number theory: Hallgren's efficient quantum algorithm for solving pell's equation. *Annals of Physics*, 306(2):241–279, 2003.

[JSV04] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51:671–697, 2004.

[JSW+24] Stephen P. Jordan, Noah Shutty, Mary Wootters, Adam Zalcman, Alexander Schmidhuber, Robbie King, Sergei V. Isakov, and Ryan Babbush. Optimization by decoded quantum interferometry. *arXiv preprint arXiv:2408.08292*, 2024.

[JZ13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[KP17] Iordanis Kerenidis and Anupam Prakash. Quantum Recommendation Systems. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:21, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[KS22] Yuri Kinoshita and Taiji Suzuki. Improved convergence rate of stochastic gradient langevin dynamics with variance reduction and its application to optimization. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 19022–19034. Curran Associates, Inc., 2022.

[Lal13] Steven P. Lalley. Concentration inequalities. *Lecture notes, University of Chicago*, 2013.

[LGHL24] Chengchang Liu, Chaowen Guan, Jianhao He, and John C.S. Lui. Quantum algorithms for non-smooth non-convex optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[LRG18] Holden Lee, Andrej Risteski, and Rong Ge. Beyond log-concavity: Provable guarantees for sampling multi-modal distributions using simulated tempering langevin monte carlo. *Advances in neural information processing systems*, 31, 2018.

[LV06] László Lovász and Santosh Vempala. Simulated annealing in convex bodies and an o*(n4) volume algorithm. *Journal of Computer and System Sciences*, 72(2):392–417, 2006. JCSS FOCS 2003 Special Issue.

[LY98] Tzong-Yow Lee and Horng-Tzer Yau. Logarithmic Sobolev inequality for some models of random walks. *The Annals of Probability*, 26(4):1855–1873, 1998.

[LZ24] Tongyang Li and Ruizhe Zhang. Quantum speedups of optimizing approximately convex functions with applications to logarithmic regret stochastic convex bandits. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.

[LZJ22] Tianyi Lin, Zeyu Zheng, and Michael Jordan. Gradient-free methods for deterministic and stochastic nonsmooth nonconvex optimization. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 26160–26175. Curran Associates, Inc., 2022.

[LZT22] Ruilin Li, Hongyuan Zha, and Molei Tao. Sqrt(d) dimension dependence of langevin monte carlo. In *The International Conference on Learning Representations*, 2022.

[MCC+19] Yi-An Ma, Niladri Chatterji, Xiang Cheng, Nicolas Flammarion, Peter Bartlett, and Michael I. Jordan. Is there an analog of nesterov acceleration for mcmc?, 2019.

[MCJ+19] Yi-An Ma, Yuansi Chen, Chi Jin, Nicolas Flammarion, and Michael I. Jordan. Sampling can be faster than optimization. *Proceedings of the National Academy of Sciences*, 116(42):20881–20885, September 2019.

[MNRS07] Frederic Magniez, Ashwin Nayak, Jeremie Roland, and Miklos Santha. Search via quantum walk. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, page 575–584, New York, NY, USA, 2007. Association for Computing Machinery.

[Mon15] Ashley Montanaro. Quantum speedup of monte carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):20150301, sep 2015.

[Mon18] Ashley Montanaro. Quantum-walk speedup of backtracking algorithms. *Theory of Computing*, 14:1–24, 2018.

[Mon20] Ashley Montanaro. Quantum speedup of branch-and-bound algorithms. *Phys. Rev. Research*, 2:013056, 2020.

[MWW09] Elchanan Mossel, Dror Weitz, and Nicholas Wormald. On the hardness of sampling independent sets beyond the tree threshold. *Probability Theory and Related Fields*, 143(3):401–439, 2009.

[Nes18] Y. Nesterov. *Lectures on Convex Optimization*. Springer Optimization and Its Applications. Springer International Publishing, 2018.

[NLST17] Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2613–2621. PMLR, 06–11 Aug 2017.

[NS17] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Found. Comput. Math.*, 17(2):527–566, April 2017.

[OLMW24] Guneykan Ozgul, Xiantao Li, Mehrdad Mahdavi, and Chunhao Wang. Stochastic quantum sampling for non-logconcave distributions and estimating partition functions. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix

Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 38953–38982. PMLR, 21–27 Jul 2024.

[OLMW25] Guneykan Ozgul, Xiantao Li, Mehrdad Mahdavi, and Chunhao Wang. Quantum speedups for markov chain monte carlo methods with application to optimization, 2025.

[OV00] F. Otto and C. Villani. Generalization of an inequality by talagrand and links with the logarithmic sobolev inequality. *Journal of Functional Analysis*, 173(2):361–400, 2000.

[Par18] Leandro Pardo. *Statistical inference based on divergence measures*. CRC press, 2018.

[Rei04] Ben W. Reichardt. The quantum adiabatic optimization algorithm and local minima. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, page 502–510, New York, NY, USA, 2004. Association for Computing Machinery.

[Ric07] Peter C. Richter. Quantum speedup of classical mixing processes. *Physical Review A*, 76(4):042306, 2007.

[RRT17] Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1674–1703. PMLR, 07–10 Jul 2017.

[RSBG21] Abhishek Roy, Lingqing Shen, Krishnakumar Balasubramanian, and Saeed Ghadimi. Stochastic zeroth-order discretizations of langevin diffusions for bayesian inference, 2021.

[Sal21] Justin Salez. A sharp log-Sobolev inequality for the multislice. *Annales Henri Lebesgue*, 4:1143–1161, 2021.

[SBB07] R. Somma, S. Boixo, and H. Barnum. Quantum simulated annealing, 2007.

[SBBK08] R. D. Somma, S. Boixo, H. Barnum, and E. Knill. Quantum simulations of classical annealing processes. *Physical Review Letters*, 101(13), September 2008.

[Sch02] Uwe Schöning. A probabilistic algorithm for k-SAT based on limited local search and restart. *Algorithmica*, 32:615–623, 2002.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[SL19] Ruoqi Shen and Yin Tat Lee. The randomized midpoint method for log-concave sampling. *Advances in Neural Information Processing Systems*, 32, 2019.

[Sly10] Allan Sly. Computational transition at the uniqueness threshold. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 287–296, 2010.

[SOKB24] Alexander Schmidhuber, Ryan O'Donnell, Robin Kothari, and Ryan Babbush. Quartic quantum speedups for planted inference. *arXiv preprint arXiv:2406.19378*, 2024.

[SS14] Allan Sly and Nike Sun. Counting in two-spin models on d-regular graphs. *The Annals of Probability*, 42(6):2383 – 2416, 2014.

[ST13] Uwe Schöning and Jacobo Torán. *The Satisfiability Problem: Algorithms and Analyses*, volume 3. Lehmanns media, 2013.

[ŠVV09] Daniel Štefankovič, Santosh Vempala, and Eric Vigoda. Adaptive simulated annealing: A near-optimal connection between sampling and counting. *Journal of the ACM (JACM)*, 56(3):1–36, 2009.

[SZ23] Aaron Sidford and Chenyi Zhang. Quantum speedups for stochastic optimization. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 35300–35330. Curran Associates, Inc., 2023.

[Sze04] M. Szegedy. Quantum speed-up of markov chain based algorithms. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 32–41, 2004.

[Tan19] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC '19, page 217–228. ACM, June 2019.

[Tsy09] Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer New York, NY, 1 edition, 2009.

[vAGGdW20] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Convex optimization using quantum oracles. *Quantum*, 4:220, January 2020.

[VW19] Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[WA08] Pawel Wocjan and Anura Abeyesinghe. Speedup via quantum sampling. *Phys. Rev. A*, 78:042336, Oct 2008.

[WCNA09] Pawel Wocjan, Chen-Fu Chiang, Daniel Nagaj, and Anura Abeyesinghe. Quantum algorithm for approximating partition functions. *Physical Review A*, 80(2):022340, 2009. arXiv:0811.0596.

[Wei06] Dror Weitz. Counting independent sets up to the tree threshold. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, page 140–149, New York, NY, USA, 2006. Association for Computing Machinery.

[WFS15] Yu-Xiang Wang, Stephen Fienberg, and Alex Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2493–2502, Lille, France, 07–09 Jul 2015. PMLR.

[WT11] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress.

[XCZG18] Pan Xu, Jinghui Chen, Difan Zou, and Quanquan Gu. Global convergence of langevin dynamics based algorithms for nonconvex optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[XN17] Mingyu Xiao and Hiroshi Nagamochi. Exact algorithms for maximum independent set. *Information and Computation*, 255:126–146, 2017.

[YLC14] Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang. Fixed-point quantum search with an optimal number of queries. *Physical review letters*, 113(21):210501, 2014.

[YW23] Kaylee Yingxi Yang and Andre Wibisono. Convergence of the inexact langevin algorithm and score-based generative models in kl divergence, 2023.

[ZB10] Lenka Zdeborová and Stefan Boettcher. A conjecture on the maximum cut and bisection width in random regular graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2010(02):P02020, February 2010.

[ZG21] Difan Zou and Quanquan Gu. On the convergence of hamiltonian monte carlo with stochastic gradients. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 13012–13022. PMLR, 18–24 Jul 2021.

[Zha09] Yufei Zhao. The number of independent sets in a regular graph. *Combinatorics, Probability and Computing*, 19(2):315–320, November 2009.

[ZLC17] Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient langevin dynamics. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1980–2022. PMLR, 07–10 Jul 2017.

[ZXG19] Difan Zou, Pan Xu, and Quanquan Gu. Stochastic gradient hamiltonian monte carlo methods with recursive variance reduction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[ZXG21] Difan Zou, Pan Xu, and Quanquan Gu. Faster convergence of stochastic gradient langevin dynamics for non-log-concave sampling. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1152–1162. PMLR, 27–30 Jul 2021.

[ZZF+24] Yexin Zhang, Chenyi Zhang, Cong Fang, Liwei Wang, and Tongyang Li. Quantum algorithms and lower bounds for finite-sum optimization. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 60244–60270. PMLR, 21–27 Jul 2024.

GUNEYKAN OZGUL

guneykanozgul@gmail.com

EDUCATION

Pennsylvania State University, PA *2021-2025*
Ph.D. in Computer Science and Engineering
Advisors: Prof. Mehrdad Mahdavi, Prof. Chunhao Wang

Bogazici University, Istanbul *2012-2018*
B.S in Computer Engineering, High honor, 2nd place in class

Bogazici University, Istanbul *2012-2018*
B.S in Physics (Double Major), High honor

PUBLICATIONS AND PREPRINTS

- **Ozgul, G.**, Li, X., Mahdavi, M., and Wang, C. Quantum speedups for markov chain monte carlo methods with application to optimization, 2025. Preprint.
- Chakrabarti, S., Herman, D., **Ozgul, G.**, Zhu, S., Augustino, B., Hao, T., He, Z., Shaydulin, R., and Pistoia, M. Generalized short path algorithms: Towards super-quadratic speedup over markov chain search for combinatorial optimization, 2024. TQC 2025.
- **Ozgul, G.**, Li, X., Mahdavi, M., and Wang, C. Stochastic quantum sampling for non-logconcave distributions and estimating partition functions. Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pp. 38953–38982. PMLR, 21–27 Jul 2024.

WORK EXPERIENCE

JPMorgan & Chase, NY *Summer 2024*
Quantum Computing Intern

Pennsylvania State University, PA *2021-2025*
Graduate Researcher

Pointr, Istanbul *2019-2021*
Software Engineer

TEACHING EXPERIENCE

Pennsylvania State University 2021-2023
Teaching Assistant *State College, PA*

Fall 2021 - CMPSC 497 (Introduction to Quantum Computing)
Spring 2022 - CMPSC 448 (Machine Learning and Algorithmic AI)
Fall 2022 - CMPSC 497 (Introduction to Quantum Computing)
Spring 2023 - CMPSC 448 (Machine Learning and Algorithmic AI)