Quantum Algorithms for Escaping from Saddle Points

Chenyi Zhang*[1], Jiaqi Leng*[2], and Tongyang Li†[3,4,5]

[1]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
[2]Department of Mathematics and Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, MD, USA
[3]Center on Frontiers of Computing Studies, Peking University, Beijing, China
[4]Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA, USA
[5]Department of Computer Science and Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, MD, USA

We initiate the study of quantum algorithms for escaping from saddle points with provable guarantee. Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$, our quantum algorithm outputs an ϵ-approximate second-order stationary point using $\tilde{O}(\log^2(n)/\epsilon^{1.75})$[1] queries to the quantum evaluation oracle (i.e., the zeroth-order oracle). Compared to the classical state-of-the-art algorithm by Jin et al. with $\tilde{O}(\log^6(n)/\epsilon^{1.75})$ queries to the gradient oracle (i.e., the first-order oracle), our quantum algorithm is polynomially better in terms of $\log n$ and matches its complexity in terms of $1/\epsilon$. Technically, our main contribution is the idea of replacing the classical perturbations in gradient descent methods by simulating quantum wave equations, which constitutes the improvement in the quantum query complexity with $\log n$ factors for escaping from saddle points. We also show how to use a quantum gradient computation algorithm due to Jordan to replace the classical gradient queries by quantum evaluation queries with the same complexity. Finally, we also perform numerical experiments that support our theoretical findings.

1 Introduction

Nonconvex optimization is a central research topic in optimization theory, mainly because the loss functions in many machine learning models (including neural networks) are typically nonconvex. However, finding a global optimum of a nonconvex function is NP-hard in general. Instead, many theoretical works focus on finding local optima, since there are landscape results suggesting that local optima are nearly as good as the global optima for many learning problems [11, 35–38, 43]. On the other hand, it is known that saddle points (and local maxima) can give highly suboptimal solutions in many problems [45, 65]. Furthermore, saddle points are ubiquitous in high-dimensional nonconvex optimization problems [16, 29, 33].

*Equal contribution.

†Corresponding author. Email: tongyangli@pku.edu.cn

[1]The $\tilde{O}$ notation omits poly-logarithmic terms, i.e., $\tilde{O}(g) = O(g\,\mathrm{poly}(\log g))$.

Therefore, one of the most important problems in nonconvex optimization is to *escape from saddle points*. Suppose we have a twice-differentiable function $f\colon \mathbb{R}^n \to \mathbb{R}$ such that

- f is ℓ-smooth: $\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \le \ell \|\mathbf{x}_1 - \mathbf{x}_2\| \quad \forall\, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$,
- f is ρ-Hessian Lipschitz: $\|\nabla^2 f(\mathbf{x}_1) - \nabla^2 f(\mathbf{x}_2)\| \le \rho \|\mathbf{x}_1 - \mathbf{x}_2\| \quad \forall\, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$;

the goal is to find an ϵ-approximate local minimum $\mathbf{x}_\epsilon$ (also known as an ϵ-approximate second-order stationary point) such that[2]

$$\|\nabla f(\mathbf{x}_\epsilon)\| \le \epsilon, \quad \lambda_{\min}(\nabla^2 f(\mathbf{x}_\epsilon)) \ge -\sqrt{\rho\epsilon}. \tag{1}$$

Intuitively, this means that at $\mathbf{x}_\epsilon$, the gradient is small with norm being at most ϵ, and the Hessian is close to be positive semi-definite with the smallest eigenvalue being at least $-\sqrt{\rho\epsilon}$.

There have been two main focuses on designing algorithms for escaping from saddle points. First, algorithms with good performance in practice are typically dimension-free or almost dimension-free (i.e., having $\mathrm{poly}(\log n)$ dependence), especially considering that most machine learning models in the real world have enormous dimensions. Second, practical algorithms prefer simple oracle access to the nonconvex function. If we are given a Hessian oracle of f, which takes $\mathbf{x}$ as the input and outputs $\nabla^2 f(\mathbf{x})$, we can find an ϵ-approximate local minimum by second-order methods; for instance, Ref. [61] took $O(1/\epsilon^{1.5})$ queries. However, because the Hessian is an $n \times n$ matrix, its construction takes $\Omega(n^2)$ cost in general. Therefore, it has become a notable interest to escape from saddle points using simpler oracles.

A seminal work along this line was by Ge et al. [35], which can find an ϵ-approximate local minimum satisfying (1) only using the first-order oracle, i.e., gradients. Although this paper has a $\mathrm{poly}(n)$ dependence in the query complexity of the oracle, the follow-up work by [46] achieved to be almost dimension-free with complexity $\tilde{O}(\log^4(n)/\epsilon^2)$, and the state-of-the-art result takes $\tilde{O}(\log^6(n)/\epsilon^{1.75})$ queries [48]. However, these results suffer from a significant overhead in terms of $\log n$, and it has been an open question to keep both the merits of using only the first-order oracle as well as being close to dimension-free [49].

On the other hand, quantum computing is a rapidly advancing technology. In particular, the capability of quantum computers is dramatically increasing and recently reached "quantum supremacy" [63] by Google [7]. However, at the moment the noise of quantum gates prevents current quantum computers from being directly useful in practice; consequently, it is also of significant interest to understand quantum algorithms from a theoretical perspective for paving its way to future applications.

In this paper, we explore quantum algorithms for escaping from saddle points. This is a mutual generalization of both classical and quantum algorithms for optimization:

- For classical optimization theory, since many classical optimization methods are physics-motivated, including Nesterov's momentum-based methods [62], Hamiltonian Monte Carlo [34] or stochastic gradient Langevin dynamics [76], etc., the elevation from classical mechanics to quantum mechanics can potentially bring more observations on designing fast *quantum-inspired classical algorithms*. In fact, quantum-inspired classical machine

[2] In general, we can ask for an (ϵ_1, ϵ_2)-approximate local minimum $\mathbf{x}$ such that $\|\nabla f(\mathbf{x})\| \le \epsilon_1$ and $\lambda_{\min}(\nabla^2 f(\mathbf{x})) \ge -\epsilon_2$. The scaling in (1) was first adopted by [61] and is taken as a standard by subsequent works [1, 18, 31, 46–48, 68, 71, 72].

learning algorithms have been an emerging topic in theoretical computer science [20–22, 40, 64, 66, 67], and it is worthwhile to explore relevant classical algorithms for optimization.

- For quantum computing, the vast majority of previous quantum optimization algorithms had been devoted to convex optimization with the focuses on semidefinite programs [4, 5, 14, 15, 53] and general convex optimization [6, 19]; these results have at least a $\sqrt{n}$ dependence in their complexities, and their quantum algorithms are far from dimension-free methods. Up to now, little is known about quantum algorithms for nonconvex optimization.

  However, there are inspirations that quantum speedups in nonconvex scenarios can potentially be more significant than convex scenarios. In particular, *quantum tunneling* is a phenomenon in quantum mechanics where the wave function of a quantum particle can tunnel through a potential barrier and appear on the other side with significant probability. This very much resembles escaping from poor landscapes in nonconvex optimization. Moreover, quantum algorithms motivated by quantum tunneling will be essentially different from those motivated by the Grover search [42], and will demonstrate significant novelty if the quantum speedup compared to the classical counterparts is more than quadratic.

1.1 Contributions

Our main contribution is a quantum algorithm that can find an ϵ-approximate local minimum of a function $f\colon \mathbb{R}^n \to \mathbb{R}$ that is smooth and Hessian Lipschitz. Compared to the classical state-of-the-art algorithm by [48] using $\tilde{O}(\log^6(n)/\epsilon^{1.75})$ queries to the gradient oracle (i.e., the first-order oracle), our quantum algorithm achieves an improvement in query complexity with $\log n$ factors. Furthermore, our quantum algorithm only takes queries to the *quantum evaluation oracle* (i.e., the zeroth-order oracle), which is defined as a unitary map U_f on $\mathbb{R}^n \otimes \mathbb{R}$ such that for any $|\mathbf{x}\rangle \in \mathbb{R}^n$,

$$U_f(|\mathbf{x}\rangle \otimes |0\rangle) = |\mathbf{x}\rangle \otimes |f(\mathbf{x})\rangle\,. \tag{2}$$

Furthermore, for any $m \in \mathbb{N}$, $|\mathbf{x}_1\rangle, \ldots, |\mathbf{x}_m\rangle \in \mathbb{R}^n$, and $\mathbf{c} \in \mathbb{C}^m$ such that $\sum_{i=1}^m |\mathbf{c}_i|^2 = 1$,

$$U_f\Big(\sum_{i=1}^m \mathbf{c}_i\,|\mathbf{x}_i\rangle \otimes |0\rangle\Big) = \sum_{i=1}^m \mathbf{c}_i\,|\mathbf{x}_i\rangle \otimes |f(\mathbf{x}_i)\rangle\,. \tag{3}$$

If we measure this quantum state, we get $f(\mathbf{x}_i)$ with probability $|\mathbf{c}_i|^2$. Compared to the classical evaluation oracle (i.e., $m = 1$), the quantum evaluation oracle allows the ability to query different locations in *superposition*, which is the essence of speedups from quantum algorithms. In addition, if the classical evaluation oracle can be implemented by explicit arithmetic circuits, the quantum evaluation oracle in (2) can be implemented by quantum arithmetic circuits of about the same size. As a result, it is the standard assumption in previous literature on quantum algorithms for various optimization problems, including quadratic forms [51], basin hopper [17], and general convex optimization [6, 19]. Subsequently, we adopt it here for general nonconvex optimization.

Theorem 1 (Main result, informal)**.** *Our quantum algorithm finds an ϵ-approximate local minimum using $\tilde{O}(\log^2(n)/\epsilon^{1.75})$ queries to the quantum evaluation oracle (2).*

Technically, our work is inspired by both the perturbed gradient descent (PGD) algorithm in [46, 47] and the perturbed accelerated gradient descent (PAGD) algorithm in [48]. To be more specific, PGD applies gradient descent iteratively until it reaches a point with small gradient. It can potentially be a saddle point, so PGD applies uniform perturbation in a small ball centered at that point and then continues the GD iterations. It can be shown that with an appropriate choice of the radius, PGD can shake the point off from the saddle and converge to a local minimum with high probability. The PAGD in [48] adopts the similar perturbation idea, but the GD is replaced by Nesterov's AGD [62].

Our quantum algorithm is built upon PGD and PAGD and shares their simplicity of being single-loop, but we propose two main modifications. On the one hand, for the perturbation steps for escaping from saddle points, we replace the uniform perturbation by evolving a quantum wave function governed by the Schrödinger equation and using the measurement outcome as the perturbed result. Intuitively, the Schrödinger equation screens the local geometry of a saddle point through wave interference, which results in a phenomenon that the wave packet disperses rapidly along the directions with significant function value decrease. Specifically, quantum mechanics finds the negative curvature directions more efficiently than the classical counterpart: for a constant ϵ, the classical PGD and PAGD take $O(\log n)$ steps to decrease the function value by $\Omega(1/\log^3 n)$ and $\Omega(1/\log^5 n)$ with high probability, respectively. Quantumly, the simulation of the Schrödinger equation for time t takes $\tilde{O}(t \log n)$ evaluation queries,[3] but simulation for time $t = O(\log n)$ suffices to decrease the function value by $\Omega(1)$ with high probability. See Proposition 1 and Theorem 4.

In addition, we replace the gradient descent steps by a quantum algorithm for computing gradients using also quantum evaluation queries. The idea was initiated by Jordan in Ref. [50] which computed the gradient at a point by applying the quantum Fourier transform on a mesh near the point. Prior work has applied Jordan's algorithm to general convex optimization [6, 19]; we follow the same path by conducting a detailed analysis (see Theorem 5) showing how we replace classical gradient queries by the same number of quantum evaluation queries in nonconvex optimization.

It is worth highlighting that our quantum algorithm enjoys the following two nice features:

- *Classical-quantum hybrid:* In Algorithm 3 and Algorithm 4, the transition between consecutive iterations is still classical, while the only quantum computing part happens inside each iteration for replacing the classical uniform perturbation. Such feature is friendly for the implementation on near-term quantum computers.

- *Robustness:* Our quantum algorithm is robust from two aspects. On the one hand, we can even escape from an approximate saddle point by evolving the Schrödinger equation (see Proposition 1). On the other hand, Theorem 5 essentially shows the robustness of es-

[3] In general, the query complexity of quantum simulation depends on the properties of the Hamiltonian, i.e., norm, sparsity, etc. In our case, the Hamiltonian takes the form $H = A + B$, where A is of norm $\alpha_A = \text{poly}(n)$ but is independent of f, and B is a diagonal matrix (so its sparsity is 1) that encodes the evaluations of f. It turns out that the interaction picture simulation technique [60] is particularly suitable for this circumstance, and we only need $\tilde{O}(t \log n)$ queries to f. For details, see Section 2.1.1.

caping from saddle points by even noisy gradient descents, which may be of independent interest.

Finally, we perform numerical experiments that support our theoretical findings. Specifically, we observe the dispersion of quantum wave packets along the negative curvature direction in various landscapes. In a comparative study, our PGD with quantum simulation outperforms the classical PGD with a higher probability of escaping from saddle points and fewer iteration steps. We also compare the dimension dependence of classical and quantum algorithms in a model question with dimensions varying from 10 to 1000, and our quantum algorithm achieves a better dimension scaling overall.

Reference	Queries	Oracle
[28, 61]	$O(1/\epsilon^{1.5})$	Hessian
[1, 18]	$\tilde{O}(\log(n)/\epsilon^{1.75})$	Hessian-vector product
[46, 47]	$\tilde{O}(\log^4(n)/\epsilon^2)$	Gradient
[48]	$\tilde{O}(\log^6(n)/\epsilon^{1.75})$	Gradient
this work	$\tilde{O}(\log^2(n)/\epsilon^{1.75})$	Quantum evaluation

Table 1: A summary of the state-of-the-art results on finding approximate second-order stationary points. The query complexities are highlighted in terms of the dimension n and the precision ϵ.

1.2 Related Work

Escaping from saddle points by gradients was initiated by [35] with complexity $O(\text{poly}(n/\epsilon))$. The follow-up work by [55] improved it to $O(n^3\,\text{poly}(1/\epsilon))$, but it is still polynomial in dimension n. The breakthrough result by [46, 47] achieves iteration complexity $\tilde{O}(\log^4(n)/\epsilon^2)$ which is poly-logarithmic in n. The best-known result has complexity $\tilde{O}(\log^6(n)/\epsilon^{1.75})$ by [48] (the same result in terms of ϵ was independently obtained by [3, 71]). Besides the gradient oracle, escaping from saddle points can also be achieved using the Hessian-vector product oracle with $\tilde{O}(\log(n)/\epsilon^{1.75})$ queries [1, 18].

There has also been a rich literature on stochastic optimization algorithms for finding second-order stationary points only using the first-order oracle. The seminal work [35] showed that noisy stochastic gradient descent (SGD) finds approximate second-order stationary points in $O(\text{poly}(n)/\epsilon^4)$ iterations. This was later improved to $\tilde{O}(\text{poly}(\log n)/\epsilon^{3.5})$ [2, 3, 31, 68, 72], and the current state-of-the-art iteration complexity of stochastic algorithms is $\tilde{O}(\text{poly}(\log n)/\epsilon^3)$ due to [30, 77].

Quantum algorithms for nonconvex optimization with provable guarantee is a widely open topic. As far as we know, the only work along this direction is by [75], which gives a quantum algorithm for finding the negative curvature of a point in time $\tilde{O}(\text{poly}(r, 1/\epsilon))$, where r is the rank of the Hessian at that point. However, the algorithm has a few drawbacks: 1) The cost is expensive when $r = \Theta(n)$; 2) It relies on a quantum data structure [52] which can actually be dequantized to classical algorithms with comparable cost [20, 66, 67]; 3) It can only find the negative curvature for a fixed Hessian. In all, it is unclear whether this quantum algorithm achieves speedup for escaping from saddle points.

1.3 Open Questions

Our paper leaves several natural open questions for future investigation:

- Can we give quantum-inspired classical algorithms for escaping from saddle points? Our work suggests that compared to uniform perturbation, there exist physics-motivated methods to better exploit the randomness in gradient descent. A natural question is to understand the potential speedup of using (classical) mechanical waves.
- Can quantum algorithms achieve speedup in terms of $1/\epsilon$? The current speedup due to quantum simulation can only improve the dependence in terms of $\log n$.
- Beyond local minima, does quantum provide advantage for approaching global minima? Potentially, simulating quantum wave equations can not only escape from saddle points, but also escape from some poor local minima.

1.4 Organization

We introduce quantum simulation of the Schrödinger equation in Section 2.1, and present how it provides quantum speedup for perturbed gradient descent and perturbed accelerated gradient descent in Section 2.2 and Section 2.3, respectively. We introduce how to replace classical gradient descents by quantum evaluations in Section 3. We present numerical experiments in Section 4. Necessary tools for our proofs are given in Appendix A.

2 Escape from Saddle Points by Quantum Simulation

The main contribution of this section is to show how to escape from a saddle point by replacing the uniform perturbation in the perturbed gradient descent (PGD) algorithm [47, Algorithm 4] and the perturbed accelerated gradient descent (PAGD) algorithm [48, Algorithm 2] with a distribution adaptive to the saddle point geometry. The intuition behind the classical algorithms is that without a second-order oracle, we do not know in which direction a perturbation should be added, thus a uniform perturbation is appropriate. However, quantum mechanics allows us to find the negative curvature direction without explicit Hessian information.

2.1 Quantum Simulation of the Schrödinger Equation

We consider the most standard evolution in quantum mechanics, the Schrödinger equation:

$$i\frac{\partial}{\partial t}\Phi = \Big[-\frac{1}{2}\Delta + f(\mathbf{x})\Big]\Phi, \tag{4}$$

where Φ is a wave function in $\mathbb{R}^n$, Δ is the Laplacian operator, and f can be regarded as the potential of the evolution. In the one-dimensional case, we can prove that Φ enjoys an explicit form below if f is a quadratic function:

Lemma 1. *Suppose a quantum particle is in a one-dimensional potential field $f(x) = \frac{\lambda}{2}x^2$ with initial state $\Phi(0,x) = (\frac{1}{2\pi})^{1/4}\exp(-x^2/4)$; in other words, the initial position of this quantum particle follows the standard normal distribution $\mathcal{N}(0,1)$. The time evolution of this*

particle is governed by (4). Then, at any time $t \geq 0$, the position of the quantum particle still follows normal distribution $\mathcal{N}\left(0, \sigma^2(t;\lambda)\right)$, where the variance $\sigma^2(t;\lambda)$ is given by

$$\sigma^2(t;\lambda) = \begin{cases} 1 + \frac{t^2}{4} & (\lambda = 0), \\ \frac{(1+4\alpha^2)-(1-4\alpha^2)\cos 2\alpha t}{8\alpha^2} & (\lambda > 0, \alpha = \sqrt{\lambda}), \\ \frac{(1-e^{2\alpha t})^2+4\alpha^2(1+e^{2\alpha t})^2}{16\alpha^2 e^{2\alpha t}} & (\lambda < 0, \alpha = \sqrt{-\lambda}). \end{cases} \tag{5}$$

Lemma 1 shows that the wave function will disperse when the potential field is of negative curvature, i.e., $\lambda < 0$, and the dispersion speed is exponentially fast. Furthermore, we prove in Appendix A.1 that this "escaping-at-negative-curvature" behavior of the wave function still emerges given a quadratic potential field $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathcal{H}\mathbf{x}$ in any finite dimension.

To turn this idea into a quantum algorithm, we need to use quantum simulation. In fact, quantum simulation in real spaces is a classical problem and has been studied back in the 1990s [70, 73, 74]. There is a rich literature on the cost of quantum simulation [9, 10, 23, 57–59]; it is typically linear in the evolution time, which is formally known as the "no—fast—forwarding theorem", see Theorem 3 of [9], and Theorem 3 of [24]. In Section 2.1.1, we prove the following lemma about the cost of simulating the the Schrödinger equation using the quantum evaluation oracle in (2):

Lemma 2. *Let $f(x)\colon \mathbb{R}^n \to \mathbb{R}$ be a real-valued function with a saddle point at $x = 0$ and $f(0) = 0$. Consider the (scaled) Schrödinger equation*

$$i\frac{\partial}{\partial t}\Phi = \Big[-\frac{r_0^2}{2}\Delta + \frac{1}{r_0^2}f(\mathbf{x})\Big]\Phi \tag{6}$$

defined on the domain $\Omega = \{x \in \mathbb{R}^n : \|x\| \leq M\}$ (where $M > 0$ is the diameter that will be specified later) with periodic boundary condition.[4] Given the quantum evaluation oracle $U_f(|\mathbf{x}\rangle \otimes |0\rangle) = |\mathbf{x}\rangle \otimes |f(\mathbf{x})\rangle$ in (2) and an arbitrary initial state at time $t = 0$, the evolution of (6) for time $t > 0$ can be simulated using $\tilde{O}\big(t \log n \log^2(\frac{t}{\epsilon})\big)$ queries to U_f, where ϵ is the precision.

Because we have assumed that f is Hessian-Lipschitz, we can use the second-order Taylor expansion to approximate the function value of f near a saddle point $\tilde{x}$. Such an approximation is more accurate on a ball centered at $\tilde{x}$ with radius r_0 small enough. Regarding this, we scale the initial distribution as well as the Schrödinger equation to be more localized in terms of r_0, which results in Algorithm 1.

Algorithm 1 is the main building block of our quantum algorithms for escaping from saddle points, and also the main resource of our quantum speedup.

2.1.1 Quantum Query Complexity of Simulating the Schrödinger Equation

We prove Lemma 2 in this subsection. Before doing that, we want to briefly discuss the reason why we simulate the scaled Schrödinger equation (6) instead of the common version of

[4] Actually, we need to put this Ω in a flat n-torus $\mathbb{T}$, i.e., an n-dimensional hybercube with periodic boundary condition, because the flat torus is readily dealt with the finite difference method (FDM). Given the truncation of the function $f(x)$ on Ω, we may slightly "mollify" the edge of $f|_\Omega$ to observe the periodicity. This mollification will not have a significant impact for optimization because our simulation time is quite short and the wave function rarely has a chance to hit the boundary $\partial\Omega$.

Algorithm 1: QuantumSimulation($\tilde{\mathbf{x}}, r_0, t_e, f(\cdot)$).

1 Put a Gaussian wave packet into the potential field f, with its initial state being:

$$\Phi_0(\mathbf{x}) = \Big(\frac{1}{2\pi}\Big)^{n/4} \frac{1}{r_0^{n/2}} \exp\Big(-(\mathbf{x}-\tilde{\mathbf{x}})^2/4r_0^2\Big); \tag{7}$$

Simulate its evolution in potential field f with the Schrödinger equation for time t_e;

2 Measure the position of the wave packet and output the measurement outcome.

non-relativistic Schrödinger equation in (4), rewritten below:

$$i\frac{\partial}{\partial t}\Phi = \Big[-\frac{1}{2}\Delta + f(\mathbf{x})\Big]\Phi. \tag{8}$$

In real-world problems, we are likely to encounter an objective function $f(x)$ with a saddle point at x_0 but is not a quadratic form. In this situation, a quadratic approximation is only valid within a small neighborhood of the first-order stationary point x_0, say Ω defined in Lemma 2. Regarding this issue, it is necessary to scale the spatial variable in order to make the wave packet more localized. However, the scaling in the spatial variable will simultaneously cause a scaling in the time variable under Eq. (4). This is not preferable because the scaling in time can dramatically change the variance $\sigma(t;\lambda)$ in (5), which can cause troubles when bounding the time complexity in the analysis of algorithms. To leave the time scale invariant, we introduce a modified Schrödinger equation (6), in which the quantum simulation is restricted on a domain of diameter $O(r_0)$: this localization guarantees that the quantum wave packet captures the saddle point geometry while not to be significantly affected by other features on the landscape of $f(x)$, thus simplifying our further analysis. We may justify our construction of (6) in three aspects:

- **Geometric aspect:** Eq. (6) is obtained by considering a spatial dilation in the wave function $\Phi(t,x) \longmapsto \Phi(t,x/r)$ without changing the time scale. This property guarantees the variance of the Gaussian distribution corresponding to $\Phi(t,x/r)$ is just r^2 times the original variance $\sigma^2(t;\lambda)$ (we will prove this in Proposition 2). Mathematically, this time-invariant property means the dispersion speed is now an intrinsic quantity as it is mostly determined by the saddle point geometry.

- **Physical aspect:** When the wave function is too localized in the position space, due to the uncertainty principle, the momentum variable will spread on a large domain in the frequency space. To reconcile this imparity, we want to introduce a small r^2 factor for the kinetic energy operator $-\frac{1}{2}\Delta$ in order to balance between position and momentum.

- **Complexity aspect:** The circuit complexity of simulating Schrödinger equation is linear in the operator norm of the Hamiltonian. Our scaling in (6) drags down the operator norm of the Laplacian (we will discretize it when doing simulation) while leaves the operator norm of the potential field remain $O(\|\mathcal{H}\|)$ in a $O(r_0)$-ball. This normalization effect will help reducing the circuit complexity.

Complexity bounds of quantum simulation is a well-established research topic; see e.g. [9, 10, 23, 57–59] for detailed results and proofs. In this paper, we apply quantum simulation under the interaction picture [60]. In particular, we use the following result:

Theorem 2 ([60, Lemma 6])**.** *Let $A, B \in \mathbb{C}^{d\times d}$ be time-independent Hamiltonians that are promised to obey $\|A\| \leq \alpha_A$ and $\|B\| \leq \alpha_B$, where $\|\cdot\|$ represents the spectral norm. Then the time-evolution operator $e^{-i(A+B)t}$ can be simulated up to error ϵ by using*

$$O\Big(\alpha_B t \frac{\log(\alpha_B t/\epsilon)}{\log\log(\alpha_B t/\epsilon)}\Big)$$

queries to the unitary oracle O_B.[5]

Our Lemma 2 is inspired by [27] which gives a quantum algorithm for simulating the Schrödinger equation but without the potential function f. It discretizes the space into grids with side-length a; in this case, $-\frac{1}{2}\Delta$ reduces to $-\frac{1}{2a^2}L$ where L is the Laplacian matrix of the graph of the grids (whose off-diagonal entries are -1 for connected grids and zero otherwise; the diagonal entries are the degree of the grids). For instance, in the one-dimensional case,

$$-\frac{1}{a^2}[L\phi]_j = \frac{\phi_{j-1} - 2\phi_j + \phi_{j+1}}{a^2}, \tag{9}$$

where ϕ_j is the value on the j^{th} grid. When $a \to 0$, this becomes the second derivative of ϕ; in practice, as mentioned above, it suffices to take $1/a = \text{poly}(\log(1/\epsilon))$ such that the overall precision is bounded by ϵ.

The discretization method used in [27] is just a special example of the finite difference method (FDM), which is a common method in applied mathematics to discretize the space of ODE or PDE problems such that their solution is tractable numerically. To be more specific, the continuous space is approximated by discrete grids, and the partial derivatives are approximated by finite differences in each direction. There are higher-order approximation methods for estimating the derivatives by finite difference formulas [56], and it is known that the number of grids in each coordinate can be as small as $\text{poly}(\log(1/\epsilon))$ by applying the high-order approximations to the FDM adaptively [8]. See also Section 3 of [25] which gave quantum algorithms for solving PDEs that applied FDM with this $\text{poly}(\log(1/\epsilon))$ complexity for the grids.

We are now ready to prove Lemma 2.

Proof. There are two steps in the quantum simulation of (6): (1) discretizing the spatial domain using (9) so that the Schrödinger equation (6) is reduced to an ordinary differential equation (10); (2) simulating (10) under the interaction picture. In each step, we fix the error tolerance as $\epsilon/2$. By the triangle inequality, the overall error is ϵ.

First, we consider the k-th order finite difference method in Section 3 of [25] (the discrete Laplacian will be denoted as L_k). With the spacing between grid points being a, if we choose

[5] In fact, Lemma 6 in [60] gives an upper bound for the number of queries to the unitary oracle HAM-T. Note that the construction of HAM-T only needs 1 query to O_B (see Theorem 7), we directly give the query complexity in terms of O_B.

the mesh number along each direction as $1/a = \mathrm{poly}(n)\,\mathrm{poly}(\log(2/\epsilon))$, the finite difference error will be of order $\epsilon/2$. Then the Schrödinger equation in (6) becomes

$$i\frac{\partial}{\partial t}\Phi = \Big(-\frac{r_0^2}{2a^2}L_k + B\Big)\Phi, \tag{10}$$

where L_k is the Laplacian matrix associated to the k-th order finite difference method (discretization of the hypercube Ω) and B is a diagonal matrix such that the entry for the grid at $\mathbf{x}$ is $\frac{1}{r_0^2}f(\mathbf{x})$. Here, the function evaluation oracle U_f is trivially encoded in the matrix evaluation oracle O_B. By [25], the spectral norm of L_k is of order $O(n/a^2) = \mathrm{poly}(n)\,\mathrm{poly}(\log(2/\epsilon))$, where n is the spatial dimension of the Schödinger equation.

We simulate the evolution of (10) by Theorem 2 and taking $A = -\frac{r_0^2}{2a^2}L_k$ therein. Recall that $\|L_k\| \leq \mathrm{poly}(n)\,\mathrm{poly}(\log(2/\epsilon))$. By the ℓ-smooth condition, we have $\|\nabla f(\mathbf{x})\| \leq \ell M$ for $\mathbf{x} \in \Omega$ so that the maximal absolute value of function $f(x)$ on Ω is bounded by ℓM^2 by the Poincaré inequality. Therefore, we have $\alpha_A \leq Cr_0^2\,\mathrm{poly}(n)\,\mathrm{poly}(\log(2/\epsilon))$ where $C > 0$ is an absolute constant, and $\alpha_B \leq \ell(M/r_0)^2$. It follows from Theorem 2 that, to simulate the time evolution operator $e^{-i(A+B)t}$ for time $t > 0$, the total quantum queries to O_B (or equivalently, to U_f) is

$$O\bigg(\ell(M/r_0)^2 t\big(\log\big(t(Cr_0^2\,\mathrm{poly}(n)\,\mathrm{poly}(\log(2/\epsilon))) + \ell(M/r_0)^2\big)/\epsilon\big)\frac{\log\big(\ell(M/r_0)^2\|t/\epsilon\big)}{\log\log(\ell(M/r_0)^2 t/\epsilon)}\bigg).$$

The radius M of the simulation region is chosen large enough such that the wavepacket does not hit the boundary during simulation. Intuitively, the value of M should be proportional to the initial variance r_0. Quantitatively, it is shown in Section 2.2 such that under our choice of parameters, M/r_0 equals some constant $1/C_r$. Absorbing all poly-logarithmic constants in the big $\tilde{O}$ notation, the total quantum queries to f reduces to $\tilde{O}(t\log n\log^2(\frac{t}{\epsilon}))$ as claimed in Lemma 2. □

Remark 1. *In our scenario of escaping from saddle points, the initial state is a Gaussian wave packet* $(\frac{1}{2\pi})^{n/4}\frac{1}{r_0^{n/2}}\exp(-(\mathbf{x}-\tilde{\mathbf{x}})^2/4r_0^2)$ *as in Algorithm 1. It is well-known that a Gaussian state can be efficiently prepared on quantum computers [54]; Gaussian states are also ubiquitous in the literature of continuous-variable quantum information [69]. However, although when f is quadratic the evolution of the Schrödinger equation keeps the state being a Gaussian wave packet by Lemma 8, it intrinsically has dependence on f and it is not totally clear how to prepare the Gaussian wave packet at time t directly by continuous-variable quantum information. It seems that the quantum simulation above using the quantum evaluation oracle U_f in (2) is necessary for our purpose.*

2.2 Perturbed Gradient Descent with Quantum Simulation

We now introduce a modified version of perturbed gradient descent. We start with gradient descents until the gradient becomes small. Then, we perturb the point by applying Algorithm 1 for a time period $t_e = \mathscr{T}'$, perform a measurement on all the coordinates (which gives an output $\mathbf{x}_0$), and continue with gradient descent until the algorithm runs for T iterations. This is summarized as Algorithm 2.

Algorithm 2: Perturbed Gradient Descent with Quantum Simulation.

1 **for** $t = 0, 1, ..., T$ **do**
2 **if** $\|\nabla f(\mathbf{x}_t)\| \leq \epsilon$ **then**
3 $\xi \sim$QuantumSimulation$(\mathbf{x}_t, r_0, \mathscr{T}', f(\mathbf{x}) - \langle \nabla f(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t \rangle)$;
4 $\Delta_t \leftarrow \frac{2\xi}{3\|\xi\|}\sqrt{\frac{\rho}{\epsilon}}$;
5 $\mathbf{x}_t \leftarrow \arg\min_{\zeta \in \{\mathbf{x}_t + \Delta_t, \mathbf{x}_t - \Delta_t\}} f(\zeta)$;
6 $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)$;

Intuitively, in Algorithm 2 QuantumSimulation is applied to find negative curvature of saddle points. Hence in Line 3 we simulate the wavepacket under the potential $f(\mathbf{x}) - \langle \nabla f(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t \rangle$ instead of f itself, since the first order term in the Taylor expansion of f at $\mathbf{x}_t$ is not relevant to the negative curvature, which is characterized by the second-order Hessian matrix. After negative curvature is specified, we can add a perturbation Δ_t in that direction to decrease the function value and escape from saddle points.

2.2.1 Effectiveness of the Perturbation by Quantum Simulation

We show that our method of quantum wave packet simulation is significantly better than the classical method of uniform perturbation in a ball. To be more specific, we focus on the scenarios with $\epsilon \leq l^2/\rho$ (this is the standard assumption adopted in [48]); intuitively, this is the case when the local landscape is "flat" and the Hessian has a small spectral radius. Under this circumstance, the classical gradient descent may move slowly, but the quantum Gaussian wavepacket still disperses fast, i.e., the variance of the probability distribution corresponding to the wavepacket still has a large increasing rate. Hence, if we let this wavepacket evolve for a long enough time period, it is drastically stretched in the directions with negative curvature. As a result, if we measure its position at this time, with high probability the output vector indicates a negative curvature direction, or equivalently, a direction along which we can decrease the function value. We can thus add a large perturbation along that direction to escape from the saddle point. Formally, we prove:

Proposition 1. *For any constant $\delta_0 > 0$, we specify our choices for the parameters and constants that we use:*

$$\mathscr{T}' := \frac{8}{(\rho\epsilon)^{1/4}} \log\Big(\frac{\ell}{\delta_0\sqrt{\rho\epsilon}}(n + 2\log(3/\delta_0))\Big) \qquad \mathscr{F}' := \frac{2}{81}\sqrt{\frac{\epsilon^3}{\rho}} \tag{11}$$

$$r_0 := \frac{4C_r^3}{9\mathscr{T}'^4}\Big(\frac{\delta_0}{3} \cdot \frac{1}{n^{3/2} + 2C_0 n\ell(\log \mathscr{T}')^\alpha}\Big)^2 \qquad \eta := \frac{1}{\ell} \tag{12}$$

where C_r, C_0, α are absolute constants, and the value of α is specified in Lemma 3. Let $f : \mathbb{R}^n \to \mathbb{R}$ be an ℓ-smooth, ρ-Hessian Lipschitz function. For an approximate saddle point $\tilde{\mathbf{x}}$ satisfying $\|\nabla f(\tilde{\mathbf{x}})\| \leq \epsilon$ and $\lambda_{\min}(\nabla^2 f(\tilde{\mathbf{x}})) \leq -\sqrt{\rho\epsilon}$, Algorithm 2 adds a perturbation by QuantumSimulation with the radius M of the simulation region being set to $M = r_0/C_r$, and decreases the function value for at least $\mathscr{F}'$, with probability at least $1 - \delta_0$.

Compared to the provable guarantee from classical perturbation [47, Lemma 22], speaking only in terms of n, classically it takes $\mathscr{T} = O(\log n)$ steps to decrease the function value by

$\mathscr{F} = \Omega(1/\log^3 n)$, whereas our quantum simulation with time $\mathscr{T}' = O(\log n)$ together with also $\mathscr{T}'$ subsequent GD iterations decrease the function value by $\mathscr{F}' = \Omega(1)$ with high success probability.

Intuitively, the proof of Proposition 1 is composed of two parts. If the potential f is quadratic, we can use Lemma 1 to prove Proposition 2 (both proof details are given in Appendix A.1), which demonstrates the exponential rate for quantum simulation to escape along the negative eigen-directions of the Hessian of f. However, the objective function f is rarely a standard quadratic form in reality, and we cannot expect the quantum wave packet to preserve its shape as a Gaussian distribution. Nevertheless, we are able to show that the quantum wave packets do not differ significantly from a perfect Gaussian distribution in the course of quantum simulation, which preserves our quantum speedup in the general case.

Formally, we introduce the following lemma to bound the deviation from perfect Gaussian in quantum evolution. Before proceeding to its details, we first specify our choice for the constant C_r. As shown in the statement of Proposition 1, C_r stands for the ratio between the initial wavepacket variance and the radius of the simulation region. We choose a small enough constant C_r, such that the simulation region would be much larger than the range of the wavepacket, during the entire simulation process. Since the function f is ℓ-smooth, the spectral norm of its Hessian matrix at any point is upper bounded by constant ℓ. Hence, the small enough constant C_r is independent of f. Then, the radius M of the simulation region satisfies

$$M = r_0/C_r = \frac{4C_r^2}{9\mathscr{T}'^4}\Big(\frac{\delta_0}{3}\cdot\frac{1}{n^{3/2}+2C_0 n\ell(\log\mathscr{T}')^\alpha}\Big)^2 \le 1. \tag{13}$$

Lemma 3. *Let $\mathcal{H}$ be the Hessian matrix of f at a saddle point $\tilde{\mathbf{x}}$, and define $f_q(\mathbf{x}) := f(\tilde{\mathbf{x}}) + \frac{1}{2}(\mathbf{x}-\tilde{\mathbf{x}})^T\mathcal{H}(\mathbf{x}-\tilde{\mathbf{x}})$ to be the quadratic approximation of the function f near $\tilde{\mathbf{x}}$. Denote the measurement outcome from the quantum simulation (see Algorithm 1) with potential field f and evolution time t_e as random variable ξ, and the measurement outcome from the quantum simulation with potential field f_q and the same evolution time t_e as another random variable ξ'. Let the law of ξ (or ξ', resp.) be $\mathbb{P}_\xi$ (or $\mathbb{P}_{\xi'}$, resp.). If the quantum wave packet is confined to a hypercube with edge length M, then*

$$TV(\mathbb{P}_\xi, \mathbb{P}_{\xi'}) \le \left(\frac{\sqrt{n}\rho}{2} + \frac{2C_f\ell}{\sqrt{r_0}}(\log t_e)^\alpha\right)\frac{nMt_e^2}{2}, \tag{14}$$

where $TV(\cdot,\cdot)$ is the total variation distance between measures, α is an absolute constant, and C_f is an f-related constant.

The proof of Lemma 3 is deferred to Appendix A.2. This lemma shows that the true perturbation given by quantum simulation $\xi \sim \mathbb{P}_\xi$ only deviates from the Gaussian random vector $\xi' \sim \mathbb{P}_{\xi'}$ at a magnitude of $\tilde{O}(Mn^{3/2}t_e^2)$ when $t_e = \mathscr{T}' = O(\log n)$ in Algorithm 2. Such a deviation is non-material compared to our choice of M in (13). Therefore, we may estimate the performance of our quantum simulation subroutine using a quadratic approximation function and then bound the error caused by the non-quadratic part, as in the following lemma:

Lemma 4. *Under the setting of Proposition 1, let $\tilde{\mathcal{H}}$ be the Hessian matrix of f at point $\tilde{\mathbf{x}}$. Then, the output of QuantumSimulation$(\tilde{\mathbf{x}}, r_0, \mathscr{T}')$ by applying Algorithm 1, denoted as ξ,*

satisfies

$$\frac{\xi^T \tilde{\mathcal{H}} \xi}{\|\xi\|^2} \leq -\frac{\sqrt{\rho\epsilon}}{3}, \tag{15}$$

with probability at least $1-\delta_0$.

Proof. Without loss of generality, assume $\nabla f(\mathbf{x}_t) = \mathbf{0}$. First consider the case where the potential f is purely quadratic, and add the estimate the error term caused by the non-quadratic deflation afterwards.

First note that the Hessian matrix $\tilde{\mathcal{H}}$ admits the following eigen-decomposition:

$$\tilde{\mathcal{H}} = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^T, \tag{16}$$

where the set $\{\mathbf{u}_i\}_{i=1}^n$ forms an orthonormal basis of $\mathbb{R}^n$. Without loss of generality, we assume that the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ corresponding to $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ satisfy

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n, \tag{17}$$

in which $\lambda_1 \leq -\sqrt{\rho\epsilon}$. If $\lambda_n \leq -\sqrt{\rho\epsilon}/2$, Lemma 4 holds directly. Hence, we only need to prove the case where $\lambda_n > -\sqrt{\rho\epsilon}/2$, in which there exists some p, p' with

$$\lambda_p \leq -\sqrt{\rho\epsilon} < \lambda_{p+1}, \quad \lambda_{p'} \leq -\sqrt{\rho\epsilon}/2 < \lambda_{p'+1}. \tag{18}$$

We use $\mathfrak{S}_{\|}$, $\mathfrak{S}_{\perp}$ to respectively denote the subspace of $\mathbb{R}^n$ spanned by

$$\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p\}, \quad \{\mathbf{u}_{p+1}, \mathbf{u}_{p+2}, \ldots, \mathbf{u}_n\}, \tag{19}$$

and use $\mathfrak{S}'_{\|}$, $\mathfrak{S}'_{\perp}$ to respectively denote the subspace of $\mathbb{R}^n$ spanned by

$$\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{p'}\}, \quad \{\mathbf{u}_{p'+1}, \mathbf{u}_{p+2}, \ldots, \mathbf{u}_n\}. \tag{20}$$

Furthermore, we define $\xi_{\|} := \sum_{i=1}^{p} \langle \mathbf{u}_i, \xi \rangle \mathbf{u}_i$, $\xi_{\perp} := \sum_{i=p}^{n} \langle \mathbf{u}_i, \xi \rangle \mathbf{u}_i$, $\xi_{\|'} := \sum_{i=1}^{p'} \langle \mathbf{u}_i, \xi \rangle \mathbf{u}_i$, $\xi_{\perp'} := \sum_{i=p'}^{n} \langle \mathbf{u}_i, \xi \rangle \mathbf{u}_i$ respectively to denote the component of ξ in the subspaces $\mathfrak{S}_{\|}$, $\mathfrak{S}_{\perp}$, $\mathfrak{S}'_{\|}$, $\mathfrak{S}'_{\perp}$. Also, we define $\xi_1 := \langle \mathbf{u}_1, \xi \rangle \mathbf{u}_1$ to be the component of ξ along $\mathbf{u}_1$, the most negative eigen-direction.

Under the basis $\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$, by Proposition 2, the time evolution of the initial wave function is governed by (6). Then, at $t_e = \mathscr{T}'$, the wave function still follows multivariate Gaussian distribution $\mathcal{N}(0, r_0^2 \Sigma)$, with the covariance matrix being

$$\Sigma = \mathrm{diag}(\sigma^2(\mathscr{T}'; \lambda_1), ..., \sigma^2(\mathscr{T}'; \lambda_n)), \tag{21}$$

where the variance $\sigma(\mathscr{T}'; \lambda_i)$ is defined in (5). Denote $\sigma_i := \sigma(\mathscr{T}'; \lambda_i)$ for each $i \in [n]$. Then, for any $i \in [n]$ with $\mathbf{u}_i \in \mathfrak{S}'_{\perp}$, since $\lambda_i \geq -\sqrt{\rho\epsilon}/2$, we have

$$\sigma_i^2 \leq \frac{(1 - e^{(4\rho\epsilon)^{1/4}\mathscr{T}'})^2 + \rho\epsilon(1 + e^{(4\rho\epsilon)^{1/4}\mathscr{T}'})^2}{4\rho\epsilon \cdot e^{(4\rho\epsilon)^{1/4}\mathscr{T}'}}. \tag{22}$$

Due to our choice of the parameter $\mathscr{T}'$, we can further derive that

$$\sigma_i^2 \le \frac{(1+2\rho\epsilon)e^{2(4\rho\epsilon)^{1/4}\mathscr{T}'}}{4\rho\epsilon\cdot e^{(4\rho\epsilon)^{1/4}\mathscr{T}'}} \le \frac{e^{(4\rho\epsilon)^{1/4}\mathscr{T}'}}{2\rho\epsilon}. \tag{23}$$

Denote $\sigma'_\perp := \frac{e^{(4\rho\epsilon)^{1/4}\mathscr{T}'}}{2\rho\epsilon}$. We define an $(n-p')$-dimensional Gaussian distribution $\tilde{p}(\cdot)$ in $\mathfrak{S}'_\perp$:

$$\tilde{p}(\mathbf{y}) = \Big(\frac{1}{2\pi}\Big)^{(n-p')/2}\Big(\frac{\sqrt{n-p'}}{{\sigma'_\perp}r_0}\Big)\exp\Big(-\frac{(n-p')\|\mathbf{y}\|^2}{2{\sigma'_\perp}^2 r_0^2}\Big), \tag{24}$$

then the actual distribution of $\|\xi_{\perp'}\|$ is upper bounded by the distribution of $\|\mathbf{y}\|$ under the probability density function $\tilde{p}(\mathbf{y})$. Furthermore, by Lemma 12 in Appendix A.3, with probability at least $1-\delta_0/3$ we have

$$\|\xi_{\perp'}\|^2/r_0^2 \le \sum_{i=p'+1}^{n}\sigma_i^2 + 2\sqrt{\log(3/\delta_0)\sum_{i=p'+1}^{n}\sigma_i^4} + 2\max_{p'+1\le i\le n}\sigma_i^2\log(3/\delta_0) \tag{25}$$

$$\le (n-p'){\sigma'_\perp}^2 + 2\big(\sqrt{(n-p')\log(3/\delta_0)} + \log(3/\delta_0)\big){\sigma'_\perp}^2 \tag{26}$$

$$\le 2(n+2\log(3/\delta_0)){\sigma'_\perp}^2. \tag{27}$$

On the other hand, on the most negative direction $i=1$, by $\lambda_1 \le -\sqrt{\rho\epsilon}$, we can derive that

$$\sigma_1^2 \ge \frac{(1-e^{2(\rho\epsilon)^{1/4}\mathscr{T}'})^2 + 4\rho\epsilon(1+e^{2(\rho\epsilon)^{1/4}\mathscr{T}'})^2}{16\rho\epsilon e^{2(\rho\epsilon)^{1/4}\mathscr{T}'}} \tag{28}$$

$$\ge \frac{e^{4(\rho\epsilon)^{1/4}\mathscr{T}'}/2 + 4\rho\epsilon e^{4(\rho\epsilon)^{1/4}\mathscr{T}'}}{16\rho\epsilon e^{2(\rho\epsilon)^{1/4}\mathscr{T}'}} \tag{29}$$

$$\ge \frac{e^{2(\rho\epsilon)^{1/4}\mathscr{T}'}}{32\rho\epsilon}. \tag{30}$$

Hence, after we measure the wavepacket, ξ_1 satisfies

$$\Pr\Big\{|\xi_1| \ge \frac{\delta_0\sigma_1 r_0}{2}\Big\} = \int_{-\delta_0\sigma_1 r_0/2}^{\delta_0\sigma_1 r_0/2}\Big(\frac{1}{2\pi}\Big)^{1/2}\cdot\frac{1}{r_0\sigma_1}\exp\Big(-\frac{\theta^2}{2r_0^2\sigma_1^2}\Big)\mathrm{d}\theta \tag{31}$$

$$\ge \Big(\frac{1}{2\pi}\Big)^{1/2}\cdot\frac{2}{r_0\sigma_1}\cdot\frac{\delta_0\sigma_1 r_0}{2} \ge \frac{\delta_0}{3}. \tag{32}$$

By the union bound, with probability at least $1-2\delta_0/3$, the output ξ would satisfy:

$$\frac{\|\xi_{\perp'}\|}{\|\xi_{\|'}\|} \le \frac{\|\xi_\perp\|}{|\xi_1|} \le \frac{\sqrt{2(n+2\log(3/\delta_0))}}{\delta_0/2}\cdot\frac{\sigma'_{\perp'}}{\sigma_1} \tag{33}$$

$$\le \frac{3\sqrt{(n+2\log(3/\delta_0))}}{\delta_0}\cdot\frac{e^{(4\rho\epsilon)^{1/4}\mathscr{T}'/2}}{\sqrt{2\rho\epsilon}}\cdot\frac{\sqrt{32\rho\epsilon}}{e^{(\rho\epsilon)^{1/4}\mathscr{T}'}} \tag{34}$$

$$\le \frac{12\sqrt{(n+2\log(3/\delta_0))}}{\delta_0}\cdot\exp\big(-(1-\sqrt{2}/2)(\rho\epsilon)^{1/4}\mathscr{T}'\big) \tag{35}$$

$$\le \frac{\sqrt{\rho\epsilon}}{12\ell}. \tag{36}$$

Considering the fact that the function f is not purely quadratic, by Lemma 3 the inequality above may be violated with probability at most

$$\frac{2}{3}\delta_0 + TV(\mathbb{P}_\xi, \mathbb{P}_{\xi'}) \leq \frac{2}{3}\delta_0 + \left(\sqrt{n}\rho + \frac{2C\ell}{\sqrt{r_0}}(\log \mathscr{T}')^\alpha\right)\frac{nM\mathscr{T}'^2}{2}, \tag{37}$$

in which $M = r_0/C_r$ due to our parameter choice. Choose the constant C_0 in r_0 large enough to satisfy $C_0 \geq C$. Then with probability at least $1 - \delta_0$, we can still have

$$\frac{\|\xi_{\perp'}\|}{\|\xi_{\|'}\|} \leq \frac{\sqrt{\rho\epsilon}}{12\ell}, \tag{38}$$

after counting in the deviation from pure quadratic field. Under this circumstance, use $\hat{\xi}$ to denote $\xi/\|\xi\|$. Observe that

$$\hat{\xi}^T\tilde{\mathcal{H}}\hat{\xi} = (\hat{\xi}_{\perp'} + \hat{\xi}_{\|'})^T\tilde{\mathcal{H}}(\hat{\xi}_{\perp'} + \hat{\xi}_{\|'}) = \hat{\xi}_{\perp'}^T\tilde{\mathcal{H}}\hat{\xi}_{\perp'} + \hat{\xi}_{\|'}^T\tilde{\mathcal{H}}\hat{\xi}_{\|'} \tag{39}$$

since $\tilde{\mathcal{H}}\hat{\xi}_{\perp'} \in \mathfrak{S}'_\perp$ and $\tilde{\mathcal{H}}\hat{\xi}_{\|'} \in \mathfrak{S}'_\|$. Due to the ℓ-smoothness of the function, all eigenvalue of the Hessian matrix has its absolute value upper bounded by ℓ. Thus we have,

$$\hat{\xi}_{\perp'}^T\tilde{\mathcal{H}}\hat{\xi}_{\perp'} \leq \ell\|\hat{\xi}_{\perp'}^T\|_2^2 = \rho\epsilon/(144\ell^2). \tag{40}$$

Further according to the definition of $\mathfrak{S}_\|$, we have

$$\hat{\xi}_{\|'}^T\tilde{\mathcal{H}}\hat{\xi}_{\|'} \leq -\sqrt{\rho\epsilon}\|\hat{\xi}_{\|'}\|^2/2. \tag{41}$$

Combining these two inequalities together, we can obtain

$$\hat{\xi}^T\tilde{\mathcal{H}}\hat{\xi} = \hat{\xi}_{\perp'}^T\tilde{\mathcal{H}}\hat{\xi}_{\perp'} + \hat{\xi}_{\|'}^T\tilde{\mathcal{H}}\hat{\xi}_{\|'} \leq -\sqrt{\rho\epsilon}\|\hat{\xi}_{\|'}\|^2/2 + \rho\epsilon/(144\ell^2) \leq -\sqrt{\rho\epsilon}/3. \tag{42}$$

□

Now we are ready to prove Proposition 1.

Proof. Without loss of generality, we assume $\tilde{\mathbf{x}} = \mathbf{0}$. By Lemma 4, with probability at least $1 - \delta_0$, the output ξ of QuantumSimulation would be in a negative curvature direction, or quantitatively,

$$\frac{\xi^T\tilde{\mathcal{H}}\xi}{\|\xi\|^2} \leq -\sqrt{\rho\epsilon}/3. \tag{43}$$

Since we choose the one with smaller function value from $\{\Delta_t, -\Delta_t\}$ to be the perturbation result, without loss of generality we can assume $\langle\nabla f(\mathbf{0}), \Delta_t\rangle \leq 0$. Then,

$$f(\Delta_t) - f(\mathbf{0}) = \int_0^1 \langle\nabla f(\theta\Delta_t), \Delta_t\rangle \mathrm{d}\theta, \tag{44}$$

where the gradient $\nabla f(\theta\Delta_t)$ can be expressed as

$$\nabla f(\theta\Delta_t) = \nabla f(\mathbf{0}) + \int_0^\theta \mathcal{H}(\nu\Delta_t)\Delta_t \mathrm{d}\nu, \tag{45}$$

which leads to

$$f(\Delta_t) - f(\mathbf{0}) = \langle \nabla f(\mathbf{0}), \Delta_t \rangle + \int_0^1 \mathrm{d}\theta \Big\langle \int_0^\theta \mathcal{H}(\nu\Delta_t)\Delta_t \mathrm{d}\nu, \Delta_t \Big\rangle \tag{46}$$

$$\leq \int_0^1 \mathrm{d}\theta \int_0^\theta \langle \mathcal{H}(\nu\Delta_t)\Delta_t, \Delta_t \rangle \mathrm{d}\nu. \tag{47}$$

Here, $\mathcal{H}(\nu, \Delta_t)$ satisfies

$$\|\mathcal{H}(\nu\Delta_t) - \tilde{\mathcal{H}}\| \leq \rho \|\nu\Delta_t\| \tag{48}$$

due to the ρ-Hessian Lipschitz property of f, which indicates

$$\langle \mathcal{H}(\nu\Delta_t)\Delta_t, \Delta_t \rangle = \langle \tilde{\mathcal{H}}\Delta_t, \Delta_t \rangle + \langle (\mathcal{H}(\nu\Delta_t) - \tilde{\mathcal{H}})\Delta_t, \Delta_t \rangle \tag{49}$$

$$\leq \langle \tilde{\mathcal{H}}\Delta_t, \Delta_t \rangle + \|\mathcal{H}(\nu\Delta_t) - \tilde{\mathcal{H}}\| \cdot \|\Delta_t\|^2 \tag{50}$$

$$\leq \langle \tilde{\mathcal{H}}\Delta_t, \Delta_t \rangle + \rho \|\Delta_t\|^3 \nu, \quad \forall \nu > 0. \tag{51}$$

Hence,

$$f(\Delta_t) - f(\mathbf{0}) \leq \int_0^1 \mathrm{d}\theta \int_0^\theta \langle \mathcal{H}(\nu\Delta_t)\Delta_t, \Delta_t \rangle \mathrm{d}\nu \tag{52}$$

$$\leq \int_0^1 \mathrm{d}\theta \int_0^\theta \langle \tilde{\mathcal{H}}\Delta_t, \Delta_t \rangle \mathrm{d}\nu + \int_0^1 \mathrm{d}\theta \int_0^\theta \rho \|\Delta_t\|^3 \nu \mathrm{d}\nu \tag{53}$$

$$\leq -\frac{\sqrt{\rho\epsilon}}{6} \cdot \|\Delta_t\|^2 + \frac{\rho}{6} \cdot \|\Delta_t\|^3 \tag{54}$$

$$= -\frac{\sqrt{\rho\epsilon}}{6} \cdot \frac{4\epsilon}{9\rho} + \frac{\rho}{6} \cdot \frac{8\epsilon^{3/2}}{27\rho^{3/2}} = -\mathscr{F}'. \tag{55}$$

□

2.2.2 Proof of Our Quantum Speedup

We now prove the following theorem using Proposition 1:

Theorem 3. *For any ϵ, $\delta > 0$, Algorithm 2 with parameters chosen in Proposition 1 satisfies that at least one half of its iterations of will be ϵ-approximate local minima, using*

$$\tilde{O}\Big(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \cdot \log^2 n\Big)$$

queries to U_f in (2) and gradients with probability $\geq 1 - \delta$, where f^* is the global minimum of f.

Proof. Set $\delta_0 = \frac{2}{81(f(\mathbf{x}_0) - f^*)} \sqrt{\frac{\epsilon^3}{\rho}}$, let the parameters be chosen according to (11) and (12), and set the total iteration steps T to be:

$$T = 4 \max\Big\{ \frac{(f(\mathbf{x_0}) - f^*)}{\mathscr{F}'}, \frac{(f(\mathbf{x_0}) - f^*)}{\eta\epsilon^2} \Big\} = \tilde{O}\Big(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \cdot \log n\Big), \tag{56}$$

similar to the classical GD algorithm. We first assume that for each $\mathbf{x}_t$ we apply QuantumSimulation (Algorithm 1), we can successfully obtain an output ξ with $\xi^T\mathcal{H}\xi/\|\xi\|^2 \leq -\sqrt{\rho\epsilon}/3$, as long as $\lambda_{\min}(\mathcal{H}(\mathbf{x}_t)) \leq -\sqrt{\rho\epsilon}$. The error probability of this assumption is provided later.

Under this assumption, Algorithm 1 can be called for at most $\frac{81(f(\mathbf{x_0})-f^*)}{2}\sqrt{\frac{\rho}{\epsilon^3}} \leq \frac{T}{4}$ times, for otherwise the function value decrease will be greater than $f(\mathbf{x_0})-f^*$, which is not possible. Then, the error probability that some calls to Algorithm 1 fail to indicate a negative curvature is upper bounded by

$$\frac{81(f(\mathbf{x_0})-f^*)}{2}\sqrt{\frac{\rho}{\epsilon^3}}\cdot\delta_0 = \delta. \tag{57}$$

Excluding those iterations that QuantumSimulation is applied, we still have at least $3T/4$ steps left. They are either large gradient steps, $\|\nabla f(\mathbf{x}_t)\| \geq \epsilon$, or ϵ-approximate second-order stationary points. Within them, we know that the number of large gradient steps cannot be more than $T/4$ because otherwise, by Lemma 13 in Appendix A.4:

$$f(\mathbf{x}_T) \leq f(\mathbf{x}_0) - T\eta\epsilon^2/4 < f^*, \tag{58}$$

a contradiction. Therefore, we conclude that at least $T/2$ of the iterations must be ϵ-approximate second-order stationary points with probability at least $1-\delta$.

The number of queries can be viewed as the sum of two parts, the number of queries needed for gradient descent, denoted by T_1, and the number of queries needed for quantum simulation, denoted by T_2. Then with probability at least $1-\delta$,

$$T_1 = T = \tilde{O}\Big(\frac{(f(\mathbf{x}_0)-f^*)}{\epsilon^2}\cdot\log n\Big). \tag{59}$$

As for T_2, with probability at least $1-\delta$ quantum simulation is called for at most $\frac{4(f(\mathbf{x_0})-f^*)}{\mathscr{F}'}$ times, and by Lemma 2 it takes $\tilde{O}(\mathscr{T}'\log n\log^2(\mathscr{T}'^2/\epsilon))$ queries to carry out each simulation. Therefore,

$$T_2 = \frac{4(f(\mathbf{x_0})-f^*)}{\mathscr{F}'}\cdot\tilde{O}(\mathscr{T}'\log n\log^2(\mathscr{T}'^2/\epsilon)) = \tilde{O}\Big(\frac{(f(\mathbf{x}_0)-f^*)}{\epsilon^{1.75}}\cdot\log^2 n\Big). \tag{60}$$

As a result, the total query complexity T_1+T_2 is

$$\tilde{O}\Big(\frac{(f(\mathbf{x}_0)-f^*)}{\epsilon^2}\cdot\log^2 n\Big). \tag{61}$$

□

2.3 Perturbed Accelerated Gradient Descent with Quantum Simulation

In Theorem 3, the $1/\epsilon^2$ term is a bottleneck of the whole algorithm, but [48] improved it to $1/\epsilon^{1.75}$ by replacing the GD with the accelerated GD by [62]. We next introduce a hybrid quantum-classical algorithm (Algorithm 3) that reflects this intuition. We make the following comparisons to [48]:

- **Same:** When the gradient is large, we both apply AGD iteratively until we reach a point with small gradient. If the function becomes "too nonconvex" in the AGD, we both reset the momentum and decide whether to exploit the negative curvature at that point.

- **Difference:** At a point with small gradient, we apply quantum simulation instead of the classical uniform perturbation. Speaking only in terms of n, [48] takes $O(\log n)$ steps to decrease the Hamiltonian $f(\mathbf{x}) + \frac{1}{2\eta}\|\mathbf{v}\|^2$ by $\Omega(1/\log^5 n)$ with high probability, whereas our quantum simulation for time $\mathscr{T}' = O(\log n)$ decreases the Hamiltonian by $\Omega(1)$ with high probability.

Algorithm 3: Perturbed Accelerated Gradient Descent with Quantum Simulation.

$\mathbf{v}_0 \leftarrow 0$;
for $t = 0, 1, \ldots, T$ **do**
  if $\|\nabla f(\mathbf{x}_t)\| \le \epsilon$ **then**
    $\xi \sim$QuantumSimulation$(\mathbf{x}_t, r_0, \mathscr{T}', f(\mathbf{x}) - \langle \nabla f(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t\rangle)$;
    $\Delta_t \leftarrow \frac{2\xi}{3\|\xi\|}\sqrt{\frac{\rho}{\epsilon}}$;
    $\mathbf{x}_t \leftarrow \arg\min_{\zeta \in \{\mathbf{x}_t + \Delta_t, \mathbf{x}_t - \Delta_t\}} f(\zeta)$;
    $\mathbf{v}_t \leftarrow \mathbf{0}$;
  else
    $\mathbf{y}_t \leftarrow \mathbf{x}_t + (1-\theta)\mathbf{v}_t$, $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t - \eta' f(\mathbf{y}_t)$, and $\mathbf{v}_{t+1} \leftarrow \mathbf{x}_{t+1} - \mathbf{x}_t$;
    if $f(\mathbf{x}_t) \le f(\mathbf{y}_t) + \langle \nabla f(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t\rangle - \frac{\gamma}{2}\|\mathbf{x}_t - \mathbf{y}_t\|$ **then**
      $(\mathbf{x}_{t+1}, \mathbf{v}_{t+1}) \leftarrow$Negative-Curvature-Exploitation$(\mathbf{x}_t, \mathbf{v}_t, s)$;

The following theorem provides the complexity of this algorithm:

Theorem 4. *Suppose that the function f is ℓ-smooth and ρ-Hessian Lipschitz. We choose the parameters appearing in Algorithm 3 as follows:*

$$\delta_0 := \frac{2}{81(f(\mathbf{x}_0) - f^*)}\sqrt{\frac{\epsilon^3}{\rho}} \qquad \chi := 1 \qquad \eta := \frac{1}{\ell} \tag{62}$$

$$\mathscr{T}' := \frac{8}{(\rho\epsilon)^{1/4}}\log\Big(\frac{\ell}{\delta_0\sqrt{\rho\epsilon}}(n + 2\log(3/\delta_0))\Big) \qquad \eta' := \frac{1}{4\ell} \qquad \mathscr{F}' := \frac{2}{81}\sqrt{\frac{\epsilon^3}{\rho}} \tag{63}$$

$$r_0 := \frac{4C_r^3}{9\mathscr{T}'^4}\Big(\frac{\delta_0}{3}\cdot\frac{1}{n^{3/2} + 2C_0 n\ell(\log\mathscr{T}')^\alpha}\Big)^2 \qquad \kappa := \frac{\ell}{\sqrt{\rho\epsilon}} \qquad \theta := \frac{1}{4\sqrt{\kappa}} \tag{64}$$

$$\gamma := \frac{\theta^2}{\eta} \qquad s := \frac{\gamma}{4\rho} \qquad \mathscr{T} := \sqrt{\kappa}\cdot c_A \tag{65}$$

where c_A is chosen large enough to satisfy the condition in Lemma 14, C_0 and C_r are constants specified in Proposition 1. Then, for any $\delta > 0$, $\epsilon \le \frac{\ell^2}{\rho}$, if we run Algorithm 3 with choice of parameters specified above, then with probability at least $1 - \delta$ one of the iterations $\mathbf{x}_t$ will be an ϵ-approximate second-order stationary point, using the following number of queries to U_f in (2) and classical gradients:

$$\tilde{O}\Big(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^{1.75}}\cdot\log^2 n\Big). \tag{66}$$

Proof. We use T to denote total number of iterations and specify our choice for T as:

$$T = 3\max\left\{\frac{(f(\mathbf{x_0}) - f^*)}{\mathscr{F}'}, \frac{(f(\mathbf{x_0}) - f^*)\mathscr{T}}{\mathscr{E}}\right\}, \tag{67}$$

where $\mathscr{E} = \sqrt{\frac{\epsilon^3}{\rho}} \cdot c_A^{-7}$, the same as our choice for $\mathscr{E}$ in Lemma 14. Similar to Proposition 1, we set the radius M of the simulation region to be r_0/C_r. We assume the contrary, i.e., the outputs of all of the iterations are not ϵ-approximate second-order stationary points.

Similar to our analysis in the proof of Theorem 3, we first assume that for each $\mathbf{x}_t$ we apply QuantumSimulation (Algorithm 1), we can successfully obtain an output ξ with $\xi^T \mathcal{H} \xi / \|\xi\|^2 \leq -\sqrt{\rho\epsilon}/3$, as long as $\lambda_{\min}(\mathcal{H}(\mathbf{x}_t)) \leq -\sqrt{\rho\epsilon}$. The error probability of this assumption is provided later.

Under this assumption, Algorithm 1 can be called for at most $\frac{81(f(\mathbf{x_0})-f^*)}{2}\sqrt{\frac{\rho}{\epsilon^3}} \leq \frac{T}{3}$ times, for otherwise the function value decrease will be greater than $f(\mathbf{x_0}) - f^*$, which is not possible. Then, the error probability that some calls to Algorithm 1 fails to indicate a negative curvature is upper bounded by

$$\frac{81(f(\mathbf{x_0}) - f^*)}{2}\sqrt{\frac{\rho}{\epsilon^3}} \cdot \delta_0 = \delta. \tag{68}$$

Excluding those iterations that QuantumSimulation is applied, we still have at least $2T/3$ steps left, which are all accelerated gradient descent steps.

Since from $\epsilon \leq \ell^2/\rho$ we have $\mathscr{T}' \geq \mathscr{T}$, then we can found at least $\frac{T}{3\mathscr{T}}$ disjoint time periods, each of time interval $\mathscr{T}$. From Lemma 14, during these time intervals the Hamiltonian will decrease in total at least:

$$\frac{T}{3\mathscr{T}} \times \mathscr{E} = f(\mathbf{x_0}) - f^*, \tag{69}$$

which is impossible due to Lemma 15, the Hamiltonian decreases monotonically for every step where quantum simulation is not called, and the overall decrease cannot be greater than $f(\mathbf{x_0}) - f^*$.

Note that the iteration numbers T satisfies:

$$T = 3\max\left\{\frac{(f(\mathbf{x_0}) - f^*)}{\mathscr{F}'}, \frac{(f(\mathbf{x_0}) - f^*)\mathscr{T}}{\mathscr{E}}\right\} = \tilde{O}\Big(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^{1.75}} \cdot \log n\Big). \tag{70}$$

As for the number of queries, it can be viewed as the sum of two parts, the number of queries needed for accelerated gradient descent, denoted by T_1, and the number of queries needed for quantum simulation, denoted by T_2. Then with probability at least $1 - \delta$,

$$T_1 = T = \tilde{O}\Big(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^{1.75}} \cdot \log n\Big). \tag{71}$$

For T_2, with probability at least $1 - \delta$ quantum simulation is called for at most $\frac{(f(\mathbf{x_0})-f^*)}{\mathscr{F}'}$ times, and by Lemma 2 it takes $\tilde{O}(\mathscr{T}' \log n \log^2(\mathscr{T}'^2/\epsilon))$ queries to carry out each simulation. Therefore,

$$T_2 = \frac{3(f(\mathbf{x_0}) - f^*)}{\mathscr{F}'} \cdot \tilde{O}(\mathscr{T}' \log n \log^2(\mathscr{T}'^2/\epsilon)) = \tilde{O}\Big(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^{1.75}} \cdot \log^2 n\Big). \tag{72}$$

As a result, the total query complexity $T_1 + T_2$ is

$$\tilde{O}\Big(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^{1.75}} \cdot \log^2 n\Big). \tag{73}$$

□

Remark 2. *Although the theorem above only guarantees that one of the iterations is an ϵ-approximate second-order stationary point, it can be easily accessed by adding a proper termination condition: once the quantum simulation is called, we keep track of the point $\tilde{\mathbf{x}}$ prior to quantum simulation, and compare the function value at $\tilde{\mathbf{x}}$ with that of $\mathbf{x}_t$ after the perturbation. If the function value decreases by at least $\mathscr{F}'$, then the algorithm has made progress, otherwise with probability at least $1-\delta$, $\tilde{\mathbf{x}}$ is an ϵ-approximate second-order stationary point. Doing so will add an extra register for saving the point but does not increase the asymptotic complexity.*

3 Gradient Descent by the Quantum Evaluation Oracle

Another important contribution of this paper is to show how to replace the classical gradient queries by quantum evaluation queries. This is shown in the case of convex optimization [6, 19], and we generalize the same result to nonconvex optimization.

The idea was initiated by [50]. Classically, with only an evaluation oracle, the best way to construct a gradient oracle is probably to walk along each direction a little bit and compute the finite difference in each coordinate. Quantumly, a clever approach is to take the uniform superposition on a mesh around the point, query the quantum evaluation oracle (in superposition) in phase,[6] and apply the quantum Fourier transform (QFT). Due to Taylor expansion,

$$\sum_{\mathbf{x}} e^{if(\mathbf{x})}\mathbf{x} \approx \sum_{\mathbf{x}} \bigotimes_{k=1}^{n} e^{i\frac{\partial f}{\partial x_k}\mathbf{x}_k}\mathbf{x}_k, \tag{74}$$

the QFT can recover all the partial derivatives simultaneously. In this paper, we refer to Lemma 2.2 of [19] for a precise version of Jordan's algorithm:

Lemma 5. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be an ℓ-smooth function specified by the evaluation oracle in (2) with accuracy δ_q, i.e., it returns a value $\tilde{f}(x)$ such that $|\tilde{f}(x) - f(x)| \le \delta_q$. For any $\mathbf{x} \in \mathbb{R}^n$, there is a quantum algorithm that uses one query to (2) and returns a vector $\tilde{\nabla} f(\mathbf{x})$ s.t.*

$$\mathbb{P}\left[\|\tilde{\nabla} f(\mathbf{x}) - \nabla f(\mathbf{x})\|_2 > 400\omega n\sqrt{\delta_q \ell}\right] < \min\left\{\frac{n}{\omega - 1}, 1\right\}, \qquad \forall \omega > 1. \tag{75}$$

The main technical contribution of this section is to replace the gradient descent steps in Section 2 by Lemma 5. We give error bounds of gradient computation steps in Section 3.1, and give the proof details of escaping from saddle points in Section 3.2.

3.1 Error Bounds of Gradient Computation Steps

We first give the following bound on gradient descent using Lemma 5:

Lemma 6. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be an ℓ-smooth, ρ-Hessian Lipschitz function, and let $\eta \le 1/\ell$. Then the gradient outputted by Lemma 5 satisfies that for any fixed constant c, with probability*

[6] This can be achieved by a standard technique called phase kickback. See more details at [39] and [19].

at least $1-\frac{n}{\frac{1}{A_q}\sqrt{\frac{2c}{\eta}-1}}$*, any specific step of the gradient descent sequence* $\{\mathbf{x}_t : \mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta\tilde{\nabla}\mathbf{x}_t\}$ *satisfies:*

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq -\eta\|\nabla f(\mathbf{x}_t)\|^2/2 + c, \tag{76}$$

where $A_q = 400n\sqrt{\delta_q \ell}$ *in the formula stands for a constant of the accuracy of the quantum algorithm.*

Ideally speaking, A_q can be arbitrarily small given a quantum computer that is accurate enough using more qubits for the precision δ_q.

Proof. Considering our condition of f being ℓ-smooth, we have

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t) \cdot (\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{\ell}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2. \tag{77}$$

we use $\mathbf{g}(\mathbf{x})$ to denote the outcome of the quantum algorithm. Then by the definition of gradient descent, $\mathbf{x}_{t+1} - \mathbf{x}_t = \eta\mathbf{g}(\mathbf{x}_t)$. Let $\delta[\mathbf{g}(\mathbf{x})] := \mathbf{g}(\mathbf{x}) - \nabla f(\mathbf{x})$. Then we have

$$\begin{aligned} f(\mathbf{x}_{t+1}) &\leq f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t) \cdot (\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{\ell}{2}\|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 && (78)\\ &\leq f(\mathbf{x}_t) - \eta\nabla f(\mathbf{x}_t) \cdot (\nabla f(\mathbf{x}_t) + \delta[\mathbf{g}(\mathbf{x}_t)]) + \frac{\eta}{2}\|\nabla f(\mathbf{x}_t) + \delta[\mathbf{g}(\mathbf{x}_t)]\|^2 && (79)\\ &= f(\mathbf{x}_t) - \frac{\eta}{2}\|\nabla f(\mathbf{x}_t)\|^2 + \frac{\eta}{2}\|\delta[\mathbf{g}(\mathbf{x}_t)]\|^2. && (80) \end{aligned}$$

By Lemma 5, for a fixed constant c, the value of $\frac{\eta}{2}\|\delta[\mathbf{g}(\mathbf{x}_t)]\|^2$ is smaller than c with probability at least $1 - \frac{n}{\frac{1}{A_q}\sqrt{\frac{2c}{\eta}-1}}$, completing the proof. □

Now, we replace all the gradient queries in Algorithm 2 by quantum evaluation queries, which results in Algorithm 4. We aim to show that if it starts at $\mathbf{x}_0$ and the value of the objective function does not decrease too much over iterations, then its whole iteration sequence $\{\mathbf{x}_\tau\}_{\tau=0}^t$ will be located in a small neighborhood of $\mathbf{x}_0$. Intuitively, this is a robust version of the "improve or localize" phenomenon presented in [47].

Algorithm 4: Perturbed Gradient Descent with Quantum Simulation and Gradient Computation.

for $t = 0, 1, \ldots, T$ **do**
  Apply Lemma 5 to compute an estimate $\tilde{\nabla} f(\mathbf{x})$ of $\nabla f(\mathbf{x})$;
  if $\|\tilde{\nabla} f(\mathbf{x}_t)\| \leq \epsilon$ **then**
    $\xi \sim$QuantumSimulation$(\mathbf{x}_t, r_0, \mathscr{T}')$;
    $\Delta_t \leftarrow \frac{2\xi}{3\|\xi\|}\sqrt{\frac{\rho}{\epsilon}}$;
    $\mathbf{x}_t \leftarrow \arg\min_{\zeta \in \{\mathbf{x}_t + \Delta_t, \mathbf{x}_t - \Delta_t\}} f(\zeta)$;
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta\tilde{\nabla} f(\mathbf{x}_t)$;

Lemma 7. *Under the setting of Lemma 6, for arbitrary $t > \tau > 0$ and arbitrary constant c, with probability at least $1 - \frac{nt}{\frac{1}{A_q}\sqrt{\frac{2c}{\eta}}-1}$ we have*

$$\|\mathbf{x}_\tau - \mathbf{x}_0\| \leq 2\sqrt{\eta t|f(\mathbf{x}_0) - f(\mathbf{x}_t)|} + 2\eta t\sqrt{c}, \tag{81}$$

if quantum simulation is not called during $[0, t]$.

Proof. Observe that

$$\|\mathbf{x}_\tau - \mathbf{x}_0\| \leq \sum_{\tau=1}^{t} \|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\|. \tag{82}$$

Using the Cauchy-Schwartz inequality, the formula above can be converted to:

$$\|\mathbf{x}_\tau - \mathbf{x}_0\| \leq \sum_{\tau=1}^{t} \|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\| \leq \Big[t \sum_{\tau=1}^{t} \|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\|^2\Big]^{\frac{1}{2}}, \tag{83}$$

in which

$$\mathbf{x}_\tau - \mathbf{x}_{\tau-1} = \eta \mathbf{g}(\mathbf{x}_{\tau-1}) = \eta \nabla f(\mathbf{x}_{\tau-1}) + \eta \delta[\mathbf{g}(\mathbf{x}_{\tau-1})], \tag{84}$$

which results in

$$\begin{aligned} \|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\|^2 &\leq \eta^2 \|\nabla f(\mathbf{x}_{\tau-1})\|^2 + 2\eta^2 \nabla f(\mathbf{x}_{\tau-1}) \cdot \delta[\mathbf{g}(\mathbf{x}_{\tau-1})] + \eta^2 \|\delta[\mathbf{g}(\mathbf{x}_{\tau-1})]\|^2 && (85) \\ &\leq 2\eta^2 \|\nabla f(\mathbf{x}_{\tau-1})\|^2 + 2\eta^2 \|\delta[\mathbf{g}(\mathbf{x}_{\tau-1})]\|^2. && (86) \end{aligned}$$

Go back to the first inequality,

$$\|\mathbf{x}_\tau - \mathbf{x}_0\| \leq \Big[t \sum_{\tau=1}^{t} \|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\|^2\Big]^{\frac{1}{2}} \leq \Big[2\eta^2 t \sum_{\tau=1}^{t} (\|\nabla f(\mathbf{x}_{\tau-1})\|^2 + \|\delta[\mathbf{g}(\mathbf{x}_{\tau-1})]\|^2)\Big]^{\frac{1}{2}}. \tag{87}$$

Suppose during each step from 1 to t, the value of $\|\delta[\mathbf{g}(\mathbf{x}_{\tau-1})]\|^2$ is smaller than the fixed constant c. From Lemma 5, this condition can be satisfied with probability at least $1 - \frac{nt}{\frac{1}{A_q}\sqrt{\frac{2c}{\eta}}-1}$. Then,

$$\begin{aligned} \|\mathbf{x}_\tau - \mathbf{x}_0\| &\leq \Big[2\eta^2 t \sum_{\tau=1}^{t} \Big(\|\nabla f(\mathbf{x}_{\tau-1})\|^2 + \|\delta[\mathbf{g}(\mathbf{x}_{\tau-1})]\|^2\Big)\Big]^{\frac{1}{2}} && (88) \\ &\leq \Big[2\eta^2 t \Big(\frac{2f(\mathbf{x}_0) - 2f(\mathbf{x}_t)}{\eta} + 2t\|\delta[\mathbf{g}(\mathbf{x}_{\tau-1})]\|^2\Big)\Big]^{\frac{1}{2}} && (89) \\ &\leq [4\eta t(f(\mathbf{x}_0) - f(\mathbf{x}_t) + \eta t c)]^{\frac{1}{2}} && (90) \\ &\leq 2\sqrt{\eta t|f(\mathbf{x}_0) - f(\mathbf{x}_t)|} + 2\eta t\sqrt{c}. && (91) \end{aligned}$$

□

3.2 Escaping from Saddle Points with Quantum Simulation and Gradient Computation

In this subsection, we prove the result below for escaping from saddle points with both quantum simulation and gradient computation. Compared to Theorem 3, it reduces classical gradient queries to the same number of quantum evaluation queries.

Theorem 5. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be an ℓ-smooth, ρ-Hessian Lipschitz function. Suppose that we have the quantum evaluation oracle U_f in (2) with accuracy $\delta_q \leq O\Big(\frac{\delta^2\epsilon^2}{\ell n^4}\Big)$. Then Algorithm 4 finds an ϵ-approximate local minimum satisfying (1), using*

$$\tilde{O}\Big(\frac{(f(\mathbf{x}_0)-f^*)}{\epsilon^2}\cdot\log^2 n\Big)$$

queries to U_f with probability at least $1-\delta$, under the following parameter choices:

$$\mathscr{T}' := \frac{8}{(\rho\epsilon)^{1/4}}\log\Big(\frac{\ell}{\delta_0\sqrt{\rho\epsilon}}(n+2\log(3/\delta_0))\Big) \qquad \mathscr{F}' := \frac{2}{81}\sqrt{\frac{\epsilon^3}{\rho}} \tag{92}$$

$$r_0 := \frac{4C_r^3}{9\mathscr{T}'^4}\Big(\frac{\delta_0}{3}\cdot\frac{1}{n^{3/2}+2C_0 n\ell(\log\mathscr{T}')^\alpha}\Big)^2 \qquad \eta := \frac{1}{\ell} \tag{93}$$

where C_0 and C_r are constants specified in Proposition 1, $\mathbf{x}_0$ is the start point, and f^* is the global minimum of f.

Note that Theorem 5 essentially shows that the perturbed gradient descent method still converges with the same asymptotic bound if there is a small error in gradient queries. This robustness of escaping from saddle points may be of independent interest.

Proof. Set $\delta_0 = \frac{1}{81(f(\mathbf{x}_0)-f^*)}\sqrt{\frac{\epsilon^3}{\rho}}$ and set the quantum accuracy $\delta_q \leq \frac{1}{2\ell}\Big(\frac{\delta\epsilon}{1000n^2}\Big)^2$. Let total iteration steps T to be:

$$T = 4\max\Big\{\frac{(f(\mathbf{x_0})-f^*)}{\mathscr{F}'}, \frac{2(f(\mathbf{x_0})-f^*)}{\eta\epsilon^2}\Big\} = \tilde{O}\Big(\frac{(f(\mathbf{x}_0)-f^*)}{\epsilon^2}\cdot\log n\Big), \tag{94}$$

similar to the classical GD algorithm. The same to Proposition 1, we set the radius M of the simulation range to be r_0/C_r. First assume that for each $\mathbf{x}_t$ we apply QuantumSimulation (Algorithm 1), we can successfully obtain an output ξ with $\xi^T\mathcal{H}\xi/\|\xi\|^2 \leq -\sqrt{\rho\epsilon}/3$, as long as $\lambda_{\min}(\mathcal{H}(\mathbf{x}_t)) \leq -\sqrt{\rho\epsilon}$. The error probability of this assumption is provided later.

Under this assumption, Algorithm 1 can be called for at most $\frac{81(f(\mathbf{x_0})-f^*)}{2}\sqrt{\frac{\rho}{\epsilon^3}} \leq \frac{T}{4}$ times, for otherwise the function value decrease will be greater than $f(\mathbf{x_0})-f^*$, which is not possible. Then, the error probability that some calls to Algorithm 1 fails to indicate a negative curvature is upper bounded by

$$\frac{81(f(\mathbf{x_0})-f^*)}{2}\sqrt{\frac{\rho}{\epsilon^3}}\cdot\delta_0 = \delta/2. \tag{95}$$

Excluding those iterations that QuantumSimulation is applied, we still have $T/2$ steps left. They are either large gradient steps, $\|\nabla f(\mathbf{x}_t)\| \geq \epsilon$, or ϵ-approximate second-order stationary points. Within them, for each large gradient steps, by Lemma 6, with probability at least

$$1-\frac{n}{\frac{1}{400n}\sqrt{\frac{2}{\delta_q}\cdot\frac{\eta\epsilon^2}{4}}-1} = 1-\frac{n}{\frac{\epsilon}{400n}\sqrt{\frac{1}{2\delta_q\ell}}-1} \leq 1-\delta/2, \tag{96}$$

the function value decrease is greater than $\eta\epsilon^2/4$, there can be at most $T/4$ steps with large gradients—otherwise the function value decrease will be greater than $f(\mathbf{x}_0) - f^*$, which is impossible.

In summary, by the union bound we can deduce that with probability at least $1-\delta$, there are at most $T/2$ steps within $\mathscr{T}'$ steps after calling quantum simulation, and at most $T/4$ steps have a gradient greater than ϵ. As a result, the rest $T/4$ steps must all be ϵ-approximate second-order stationary points.

The number of queries can be viewed as the sum of two parts, the number of queries needed for gradient descent, denoted by T_1, and the number of queries needed for quantum simulation, denoted by T_2. Then with probability at least $1-\delta$,

$$T_1 = T = \tilde{O}\Big(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \cdot \log n\Big). \tag{97}$$

As for T_2, with probability at least $1-\delta$ quantum simulation is called for at most $\frac{4(f(\mathbf{x_0})-f^*)}{\mathscr{F}'}$ times, and by Lemma 2 it takes $\tilde{O}(\mathscr{T}' \log n \log^2(\mathscr{T}'^2/\epsilon))$ queries to carry out each simulation. Therefore,

$$T_2 = \frac{4(f(\mathbf{x_0}) - f^*)}{\mathscr{F}'} \cdot \tilde{O}(\mathscr{T}' \log n \log^2(\mathscr{T}'^2/\epsilon)) = \tilde{O}\Big(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon} \cdot \log^2 n\Big). \tag{98}$$

As a result, the total query complexity $T_1 + T_2$ is

$$\tilde{O}\Big(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \cdot \log^2 n\Big). \tag{99}$$

□

Theorem 4 and Theorem 5 together imply the main result Theorem 1 of this paper.

Remark 3. *One may notice that in Section 3, we only demonstrated the robustness of Algorithm 2 where the classical gradient oracle is replaced by the quantum evaluation oracle. We argue that the same argument holds for Algorithm 3 because the difference between Algorithm 2 and Algorithm 3 only exists in large gradient steps, while the relative error caused by Jordan's algorithm is small since the absolute error remains to be a constant. Hence, in principle Algorithm 3 satisfies the similar robustness property compared to Algorithm 2 under the change of gradient oracles.*

4 Numerical Experiments

In this section, we provide numerical results that demonstrate the power of quantum simulation for escaping from saddle points. Due to the limitation of current quantum computers, we simulate all quantum algorithms numerically on a classical computer (with Dual-Core Intel Core i5 Processor, 8GB memory). Nevertheless, our numerical results strongly assert the quantum speedup in small to intermediate scales. All the numerical results and plots are obtained by MATLAB 2019a.

In the first two experiments, we look at the wave packet evolution on both quadratic and non-quadratic potential fields. Before bringing out numerical results and related discussions, we want to briefly discuss the leapfrog scheme [41], which is the technique we employed for

numerical integration of the Schrödinger equation. We discretize the Schrödinger equation as a linear system of an ordinary differential equation (for details, see Section 2.1.1):

$$i\frac{\mathrm{d}\Psi}{\mathrm{d}t} = H\Psi, \tag{100}$$

where $\Psi\colon [0,T] \to \mathbb{C}^N$ is a vector-valued function in time. We may have a decomposition $\Psi(t) = Q(t) + iP(t)$ for $Q, P\colon [0,T] \to \mathbb{R}^N$ being the real and imaginary part of Ψ, respectively. Then plugging the decomposition into the ODE (100), we have a separable N-body Hamiltonian system

$$\begin{cases} \dot{Q} = HP; \\ \dot{P} = -HQ. \end{cases} \tag{101}$$

The optimal integration scheme for solving this Hamiltonian system is the symplectic integrator [41], and we use a second-order leapfrog integrator for separable canonical Hamiltonian systems [32] in this section. In all of our PDE simulations, we fix the spatial domain to be $\Omega = \{(x,y) : |x| \le 3, |y| \le 3\}$ and the mesh number to be 512 on each edge.

4.1 Dispersion of the Wave Packet

In Proposition 2, we showed that a centered Gaussian wave packet will disperse along the negative curvature direction of the saddle point. In the numerical simulation presented in Figure 1, we have a potential function $f_1(x,y) = -x^2/2 + 3y^2/2$ and the initial wave function as described in Proposition 2 ($r = 0.5$). In each subplot, the Gaussian wave packet (i.e., modulus square of the wave function $\Phi(t,x)$) at a specific time is shown. The quantum evolution "squeezes" the wave packet along the x-axis: the variance of the marginal distribution on the x-axis is 0.25, 0.33, 0.68 at time $t = 0, 0.5, 1$, respectively.

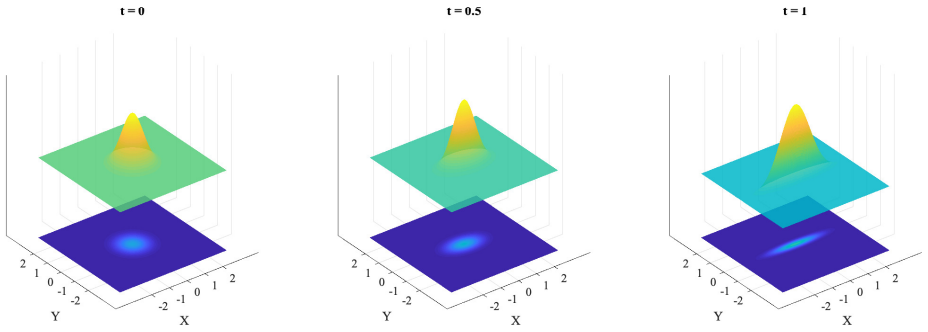

Figure 1: Dispersion of wave packet over the potential field $f_1(x,y)$. We use the finite difference method (5-point stencil) and the Leapfrog integration to simulate the Schrödinger equation (6) on a square domain (center $= (0,0)$, edge $= 6$), up to $T = 1$. The mesh number is 512 on each edge. The average runtime for this simulation is 43.7 seconds.

In the preceding experiment, we have provided a numerical simulation of the dispersion of the Gaussian wave packet on a quadratic potential field. Next, we only require that the function is Hessian-Lipschitz near the saddle point. This is enough to promise that the second-order Taylor series is a good approximation near a small neighborhood of the saddle point.

4.2 Quantum Simulation on Non-quadratic Potential Fields

Now, we explore the behavior of the wave packet on non-quadratic potential fields. It is worth noting that: (1) the wave packet is not necessarily Gaussian during the time evolution; (2) for practical reason, we will truncate the unbounded spatial domain $\mathbb{R}^2$ to be a bounded square Ω and assume Dirichlet boundary conditions ($\Phi(t,x) = 0$ on $\partial\Omega$ for all $t \in [0,T]$). Nevertheless, it is still observed that the wave packet will be mainly confined to the "valley" on the landscape which corresponds to the direction of the negative curvature.

We will run quantum simulation (Algorithm 1) near the saddle point of two non-quadratic potential landscapes. The first one is $f(x,y) = \frac{1}{12}x^4 - \frac{1}{2}x^2 + \frac{1}{2}y^2$. The Hessian matrix of $f(x,y)$ is

$$\nabla^2 f(x,y) = \begin{pmatrix} x^2 - 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{102}$$

It has a saddle point at $(0,0)$ and two global minima $(\pm\sqrt{3},0)$. The minimal function value is $-3/4$. This is the landscape used in the next experiment in which a comparison study between quantum and classical is conducted. We claimed that the wave packet will remain (almost) Gaussian at $t_e = 1.5$. This claim is confirmed by the numerical result illustrated in Figure 2. The wave packet has been "squeezed" along the x-axis, the negative curvature direction. Compared to the uniform distribution in a ball used in PGD, this "squeezed" bivariant Gaussian distribution assigns more probability mass along the x-axis, thus allowing escaping from the saddle point more efficiently.

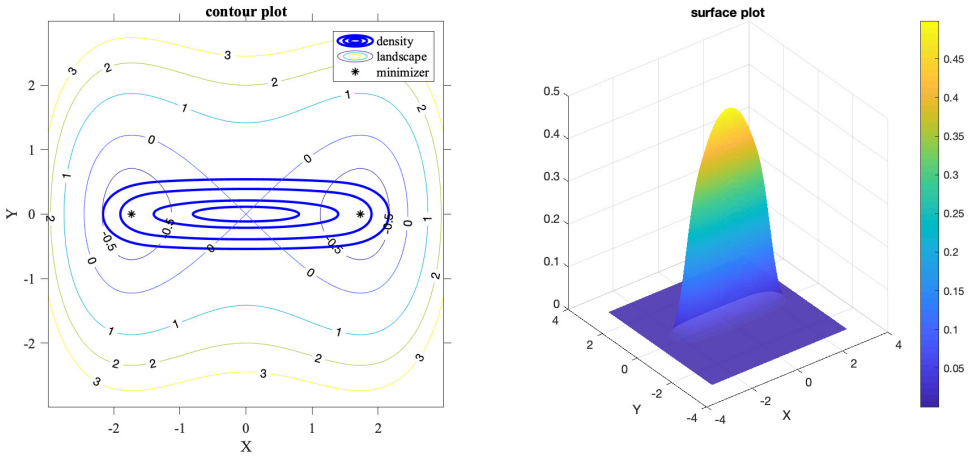

Figure 2: Quantum simulation on landscape 1: $f(x,y) = \frac{1}{12}x^4 - \frac{1}{2}x^2 + \frac{1}{2}y^2$. Parameters: $r_0 = 0.5$, $t_e = 1.5$. Left: The contour of the landscape is placed on the background with labels being function values; the thick blue contours illustrate the wave packet at $t_e = 1.5$ (i.e., modulus square of the wave function $\Phi(t_e, x, y)$).
Right: A surface plot of the same wave packet at $t_e = 1.5$. The average runtime for this simulation is 60.70 seconds.

The second landscape we explore is $g(x,y) = x^3 - y^3 - 2xy + 6$. Its Hessian matrix is

$$\nabla^2 g(x,y) = \begin{pmatrix} 6x & -2 \\ -2 & -6y \end{pmatrix}. \tag{103}$$

It has a saddle point at $(0,0)$ with no global minimum. This objective function has a circular "valley" along the negative curvature direction $(1,1)$, and a "ridge" along the positive curvature direction $(1,-1)$. We aim to study the long-term evolution of the Gaussian wave packet on the landscape restricted on a square region. The evolution of the wave packet is illustrated in Figure 3. In a small time scale (e.g., $t=1$), the wave packet disperses down the valley on the landscape, and it preserves a bell shape; waves are reflected from the boundary and an interference pattern can be observed near the upper and left edges of the square. Dispersion and interference coexist in the plot at $t=2$, in which the wave packet splits into two symmetric components, each locates in a lowland. Since the total energy is conserved in the quantum-mechanical system, the wave packet bounces back at $t=5$, but is blurred due to wave interference. In the whole evolution in $t\in[0,5]$, the wave packet is confined to the valley area of the landscape (even after bouncing back from the boundary). This evidence suggests that Gaussian wave packet is able to adapt to more complicated saddle point geometries.

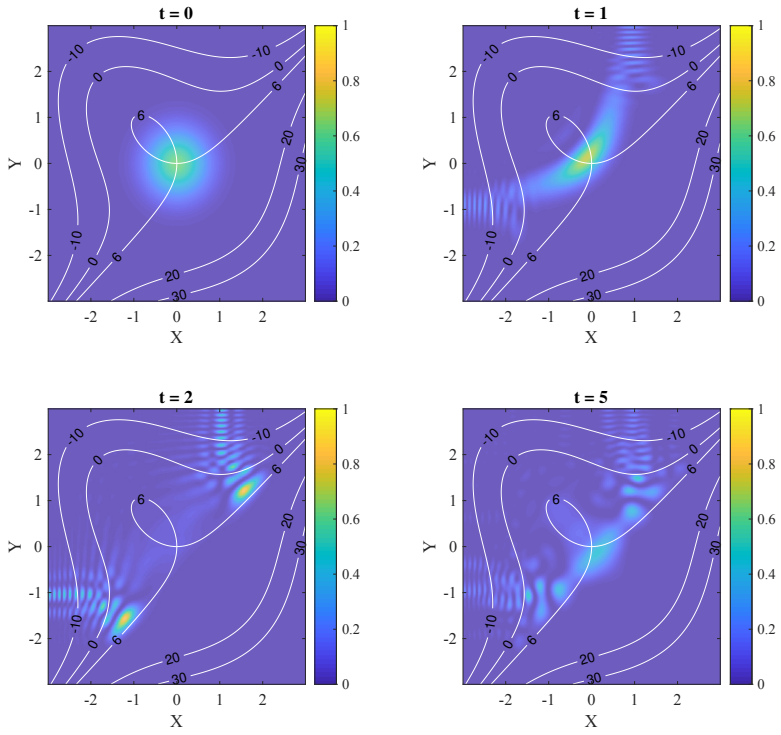

Figure 3: Quantum simulation on landscape 2: $g(x,y)=x^3-y^3-2xy+6$. Parameters: $r_0=0.5$, $t_e=5$. In each subplot, a colored contour plot of the wave packet at a specific time is shown, and the landscape contour is placed on top of the wave packet for quick reference. The average runtime for this simulation is 209.95 seconds.

4.3 Comparison Between PGD and Algorithm 2

In addition to the numerical study of the evolution of wave packets, we compare the performance of the PGD algorithm [46] with Algorithm 2 on a test function $f_2(x,y)=\frac{1}{12}x^4-\frac{1}{2}x^2+$

$\frac{1}{2}y^2$.

In this experiment and the last one in this section, we only implement a mini-batch from the whole algorithm (for both classical PGD and PGD with quantum simulation). In fact, a mini-batch is good enough for us to demonstrate the power of quantum simulation as well as the dimension dependence in both algorithms. A *mini-batch* in the experiment is defined as follows:

- Classical algorithm (PGD) mini-batch [following Algorithm 4 of 47]: x_0 is uniformly sampled from the ball $B_0(r)$ (saddle point at the origin), and then run $\mathscr{T}_c$ gradient descent steps to obtain $x_{\mathscr{T}_c}$. Record the function value $f(x_{\mathscr{T}_c})$. Repeat this process M times. The resulting function values are presented in a histogram.

- Quantum algorithm mini-batch (following Algorithm 2): Run the quantum simulation with evolution time t_e to generate a multivariate Gaussian distribution centered at 0. x_0 is sampled from this multivariate Gaussian distribution. Run $\mathscr{T}_q$ gradient descent steps and record the function value $f(x_{\mathscr{T}_q})$. Repeat this process M times. The resulting function values are also presented in a histogram, superposed to the results given by the classical algorithm.

The experimental results from 1000 samples are illustrated in Figure 4. Although the test function is non-quadratic, the quantum speedup is apparent.

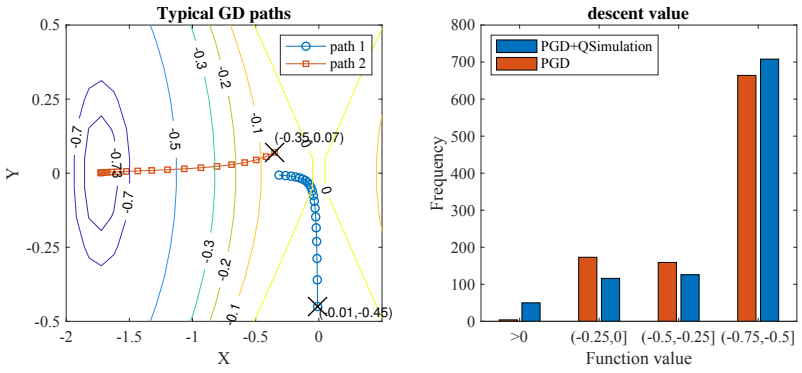

Figure 4: Left: Two typical gradient descent paths on the landscape of f_2 illustrated as a contour plot. Path 1 (resp. 2) starts from $(-0.01, 0.45)$ (resp. $(-0.35, 0.07)$); both have step length $\eta = 0.2$ and $T = 20$ iterations. Note that path 2 approaches the local minimum $(-\sqrt{3}, 0)$, while path 1 is still far away. In PGD, path 1 and 2 will be sampled with equal probability by the uniform perturbation, whereas in Algorithm 2, the dispersion of the wave packet along the x-axis enables a much higher probability of sampling a path like path 2 (that approaches the local minimum).
Right: A histogram of function values $f_2(x_{\mathscr{T}_c})$ (PGD) and $f_2(x_{\mathscr{T}_q})$ (Algorithm 2). We set step length $\eta = 0.05$, $r = 0.5$ (ball radius in PGD and r_0 in Algorithm 1), $M = 1000$, $\mathscr{T}_c = 50$, $\mathscr{T}_q = 10$, $t_e = 1.5$. Although we run five more times of iterations in PGD, there are still over 70% of gradient descent paths arriving the neighborhood of the local minimum, while there are less than 70% paths in Algorithm 2 approaching the local minimum. The average runtime of this experiment is 0.02 seconds.

4.4 Dimension Dependence

Recall that n is the dimension of the problem. Classically, it has been shown in [47] that the PGD algorithm requires $O(\log^4 n)$ iterations to escape from saddle points; however, quantum simulation for time $O(\log n)$ suffices in our Algorithm 2 by Theorem 3. The following experiment is designed to compare this dimension dependence of PGD and Algorithm 2. We choose a test function $h(x) = \frac{1}{2}x^T\mathcal{H}x$ where $\mathcal{H}$ is an n-by-n diagonal matrix: $\mathcal{H} = \text{diag}(-\epsilon, 1, 1, ..., 1)$. The function $h(x)$ has a saddle point at the origin, and only one negative curvature direction. Throughout the experiment, we set $\epsilon = 0.01$. Other hyperparameters are: dimension $n \in \mathbb{N}$, radius of perturbation $r > 0$, classical number of iterations $\mathscr{T}_c$, quantum number of iterations $\mathscr{T}_q$, quantum evolution time t_e, number of samples $M \in \mathbb{N}$, and GD step size (learning rate) η. For the sake of comparison, the iteration numbers $\mathscr{T}_c$ and $\mathscr{T}_q$ are chosen in a manner such that the statistics of the classical and quantum algorithms in each category of the histogram in Figure 5 are of similar magnitude.

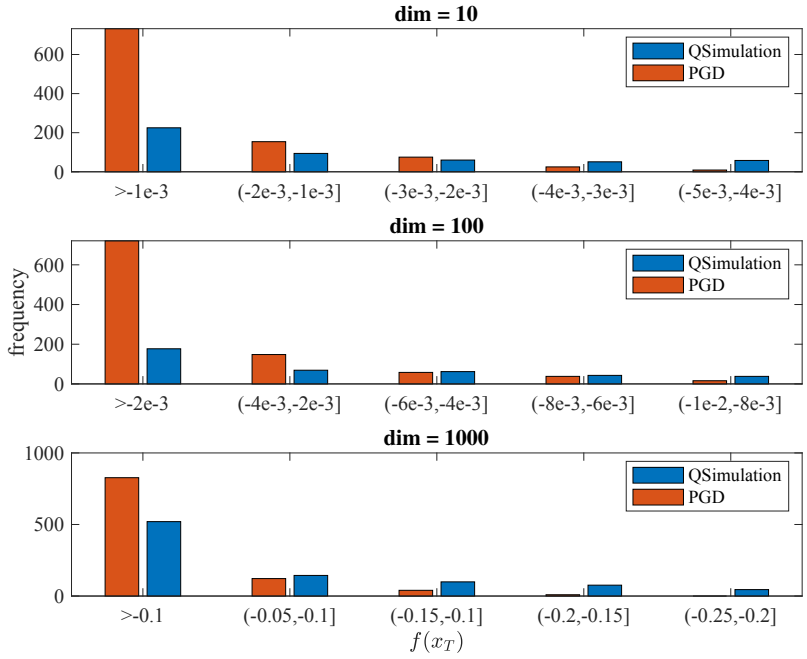

Figure 5: Dimension dependence of classical and quantum algorithms. We set $\epsilon = 0.01$, $r = 0.1$, $n = 10^p$ for $p = 1, 2, 3$. Quantum evolution time $t_e = p$, classical iteration number $\mathscr{T}_c = 50p^2 + 50$, quantum iteration number $\mathscr{T}_q = 30p$, and sample size $M = 1000$. The average runtime for this simulation is 90.92 seconds.

The numerical results are illustrated in Figure 5. The number of dimensions varies drastically from 10 to 1000, while the distribution patterns in all three subplots are similar: setting $\mathscr{T}_c = \Theta(\log^2 n)$ and $\mathscr{T}_q = \Theta(\log n)$, the PGD with quantum simulation outperforms the classical PGD in the sense that more samples can escape from the saddle point (as they have lower function values). At the same time, under this choice of parameters, the performance of the classical PGD is still comparable to that of the PGD with quantum simulation, i.e., the statistics in each category are of similar magnitude. This numerical evidence might suggest

that for a generic problem, the classical PGD method in [47] has better dimension dependence than $O(\log^4 n)$.

Acknowledgement

We thank Andrew M. Childs, András Gilyén, Aram W. Harrow, Jin-Peng Liu, Ronald de Wolf, and Xiaodi Wu for helpful discussions. We also thank anonymous reviewers for helpful suggestions on earlier versions of this paper. JL was supported by the National Science Foundation (grant CCF-1816695). TL was supported by an IBM PhD Fellowship, an QISE-NET Triplet Award (NSF grant DMR-1747426), the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Algorithms Teams program, NSF grant PHY-1818914, and a Samsung Advanced Institute of Technology Global Research Partnership.

References

[1] Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma, *Finding approximate local minima faster than gradient descent*, Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, pp. 1195–1199, 2017, arXiv:1611.01146. https://doi.org/10.1145/3055399.3055464

[2] Zeyuan Allen-Zhu, *Natasha 2: Faster non-convex optimization than SGD*, Advances in Neural Information Processing Systems, pp. 2675–2686, 2018, arXiv:1708.08694.

[3] Zeyuan Allen-Zhu and Yuanzhi Li, *Neon2: Finding local minima via first-order oracles*, Advances in Neural Information Processing Systems, pp. 3716–3726, 2018, arXiv:1711.06673.

[4] Joran van Apeldoorn and András Gilyén, *Improvements in quantum SDP-solving with applications*, Proceedings of the 46th International Colloquium on Automata, Languages, and Programming, Leibniz International Proceedings in Informatics (LIPIcs), vol. 132, pp. 99:1–99:15, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, arXiv:1804.05058. https://doi.org/10.4230/LIPIcs.ICALP.2019.99

[5] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf, *Quantum SDP-solvers: Better upper and lower bounds*, 58th Annual Symposium on Foundations of Computer Science, IEEE, 2017, arXiv:1705.01843. https://doi.org/10.22331/q-2020-02-14-230

[6] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf, *Convex optimization using quantum oracles*, Quantum **4** (2020), 220, arXiv:1809.00643. https://doi.org/10.22331/q-2020-01-13-220

[7] Frank Arute et al., *Quantum supremacy using a programmable superconducting processor*, Nature **574** (2019), no. 7779, 505–510, arXiv:1910.11333. https://doi.org/10.1038/s41586-019-1666-5

[8] Ivo Babuška and Manil Suri, *The h-p version of the finite element method with quasiuniform meshes*, ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique **21** (1987), no. 2, 199–238.

[9] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders, *Efficient quantum algorithms for simulating sparse Hamiltonians*, Communications in Mathemati-

cal Physics **270** (2007), no. 2, 359–371, arXiv:quant-ph/0508139. https://doi.org/10.1007/s00220-006-0150-x

[10] Dominic W. Berry, Andrew M. Childs, and Robin Kothari, *Hamiltonian simulation with nearly optimal dependence on all parameters*, Proceedings of the 56th Annual Symposium on Foundations of Computer Science, pp. 792–809, IEEE, 2015, arXiv:1501.01715. https://doi.org/10.1109/FOCS.2015.54

[11] Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro, *Global optimality of local search for low rank matrix recovery*, Advances in Neural Information Processing Systems, pp. 3880–3888, 2016, arXiv:1605.07221.

[12] Jean Bourgain, *Growth of Sobolev norms in linear Schrödinger equations with quasi-periodic potential*, Communications in Mathematical Physics **204** (1999), no. 1, 207–247. https://doi.org/10.1007/s002200050644

[13] Jean Bourgain, *On growth of Sobolev norms in linear Schrödinger equations with smooth time dependent potential*, Journal d'Analyse Mathématique **77** (1999), no. 1, 315–348. https://doi.org/10.1007/BF02791265

[14] Fernando G.S.L. Brandão, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu, *Quantum SDP solvers: Large speed-ups, optimality, and applications to quantum learning*, Proceedings of the 46th International Colloquium on Automata, Languages, and Programming, Leibniz International Proceedings in Informatics (LIPIcs), vol. 132, pp. 27:1–27:14, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, arXiv:1710.02581. https://doi.org/10.4230/LIPIcs.ICALP.2019.27

[15] Fernando G.S.L. Brandão and Krysta Svore, *Quantum speed-ups for semidefinite programming*, Proceedings of the 58th Annual Symposium on Foundations of Computer Science, pp. 415–426, 2017, arXiv:1609.05537. https://doi.org/10.1109/FOCS.2017.45

[16] Alan J. Bray and David S. Dean, *Statistics of critical points of Gaussian fields on large-dimensional spaces*, Physical Review Letters **98** (2007), no. 15, 150201, arXiv:cond-mat/0611023. https://doi.org/10.1103/PhysRevLett.98.150201

[17] David Bulger, *Quantum basin hopping with gradient-based local optimisation*, 2005, arXiv:quant-ph/0507193.

[18] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford, *Accelerated methods for nonconvex optimization*, SIAM Journal on Optimization **28** (2018), no. 2, 1751–1772, arXiv:1611.00756. https://doi.org/10.1137/17M1114296

[19] Shouvanik Chakrabarti, Andrew M. Childs, Tongyang Li, and Xiaodi Wu, *Quantum algorithms and lower bounds for convex optimization*, Quantum **4** (2020), 221, arXiv:1809.01731. https://doi.org/10.22331/q-2020-01-13-221

[20] Nai-Hui Chia, András Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang, *Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning*, Proceedings of the 52nd Annual ACM Symposium on Theory of Computing, pp. 387–400, ACM, 2020, arXiv:1910.06151. https://doi.org/10.1145/3357713.3384314

[21] Nai-Hui Chia, András Gilyén, Han-Hsuan Lin, Seth Lloyd, Ewin Tang, and Chunhao Wang, *Quantum-inspired algorithms for solving low-rank linear equation systems with logarithmic dependence on the dimension*, Proceedings of the 31st International Symposium on Algorithms and Computation, vol. 181, p. 47, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. https://doi.org/10.4230/LIPIcs.ISAAC.2020.47

[22] Nai-Hui Chia, Tongyang Li, Han-Hsuan Lin, and Chunhao Wang, *Quantum-inspired sublinear algorithm for solving low-rank semidefinite programming*, 45th International Symposium on Mathematical Foundations of Computer Science, 2020, arXiv:1901.03254. https://doi.org/10.4230/LIPIcs.MFCS.2020.23

[23] Andrew M. Childs, *Lecture notes on quantum algorithms*, https://www.cs.umd.edu/%7Eamchilds/qa/qa.pdf, 2017.

[24] Andrew M. Childs and Robin Kothari, *Limitations on the simulation of non-sparse Hamiltonians*, Quantum Information & Computation **10** (2010), no. 7, 669–684, arXiv:0908.4398.

[25] Andrew M. Childs, Jin-Peng Liu, and Aaron Ostrander, *High-precision quantum algorithms for partial differential equations*, 2020, arXiv:2002.07868.

[26] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu, *Theory of Trotter error with commutator scaling*, Physical Review X **11** (2021), no. 1, 011020, arXiv:1912.08854. https://doi.org/10.1103/PhysRevX.11.011020

[27] Pedro C.S. Costa, Stephen Jordan, and Aaron Ostrander, *Quantum algorithm for simulating the wave equation*, Physical Review A **99** (2019), no. 1, 012323, arXiv:1711.05394. https://doi.org/10.1103/PhysRevA.99.012323

[28] Frank E. Curtis, Daniel P. Robinson, and Mohammadreza Samadi, *A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization*, Mathematical Programming **162** (2017), no. 1-2, 1–32. https://doi.org/10.1007/s10107-016-1026-2

[29] Yann N. Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio, *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization*, Advances in Neural Information Processing Systems, pp. 2933–2941, 2014, arXiv:1406.2572.

[30] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang, *Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator*, Advances in Neural Information Processing Systems, pp. 689–699, 2018, arXiv:1807.01695.

[31] Cong Fang, Zhouchen Lin, and Tong Zhang, *Sharp analysis for nonconvex SGD escaping from saddle points*, Conference on Learning Theory, pp. 1192–1234, 2019, arXiv:1902.00247.

[32] Mauger François, *Symplectic leap frog scheme*, https://www.mathworks.com/matlabcentral/fileexchange/38652-symplectic-leap-frog-scheme, 2020.

[33] Yan V. Fyodorov and Ian Williams, *Replica symmetry breaking condition exposed by random matrix calculation of landscape complexity*, Journal of Statistical Physics **129** (2007), no. 5-6, 1081–1116, arXiv:cond-mat/0702601. https://doi.org/10.1007/s10955-007-9386-x

[34] Xuefeng Gao, Mert Gürbüzbalaban, and Lingjiong Zhu, *Global convergence of stochastic gradient Hamiltonian monte carlo for non-convex stochastic optimization: Non-asymptotic performance bounds and momentum-based acceleration*, 2018, arXiv:1809.04618.

[35] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan, *Escaping from saddle points – online stochastic gradient for tensor decomposition*, Conference on Learning Theory, pp. 797–842, 2015, arXiv:1503.02101.

[36] Rong Ge, Jason D. Lee, and Tengyu Ma, *Matrix completion has no spurious local*

minimum, Advances in Neural Information Processing Systems, pp. 2981–2989, 2016, arXiv:1605.07272.

[37] Rong Ge, Jason D. Lee, and Tengyu Ma, *Learning one-hidden-layer neural networks with landscape design*, International Conference on Learning Representations, 2018, arXiv:1711.00501.

[38] Rong Ge and Tengyu Ma, *On the optimization landscape of tensor decompositions*, Advances in Neural Information Processing Systems, pp. 3656–3666, Curran Associates Inc., 2017, arXiv:1706.05598.

[39] András Gilyén, Srinivasan Arunachalam, and Nathan Wiebe, *Optimizing quantum optimization algorithms via faster quantum gradient computation*, Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1425–1444, Society for Industrial and Applied Mathematics, 2019, arXiv:1711.00465. https://doi.org/10.1137/1.9781611975482.87

[40] András Gilyén, Zhao Song, and Ewin Tang, *An improved quantum-inspired algorithm for linear regression*, 2020, arXiv:2009.07268.

[41] Stephen K. Gray and David E. Manolopoulos, *Symplectic integrators tailored to the time-dependent Schrödinger equation*, The Journal of chemical physics **104** (1996), no. 18, 7099–7112. https://doi.org/10.1063/1.471428

[42] Lov K. Grover, *A fast quantum mechanical algorithm for database search*, Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, pp. 212–219, ACM, 1996, arXiv:quant-ph/9605043. https://doi.org/10.1145/237814.237866

[43] Moritz Hardt, Tengyu Ma, and Benjamin Recht, *Gradient descent learns linear dynamical systems*, Journal of Machine Learning Research **19** (2018), no. 29, 1–44, arXiv:1609.05191.

[44] Daniel Hsu, Sham Kakade, and Tong Zhang, *A tail inequality for quadratic forms of subgaussian random vectors*, Electronic Communications in Probability **17** (2012), 1–6, arXiv:1110.2842. https://doi.org/10.1214/ECP.v17-2079

[45] Prateek Jain, Chi Jin, Sham Kakade, and Praneeth Netrapalli, *Global convergence of non-convex gradient descent for computing matrix squareroot*, Artificial Intelligence and Statistics, pp. 479–488, 2017, arXiv:1507.05854.

[46] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan, *How to escape saddle points efficiently*, Conference on Learning Theory, pp. 1724–1732, 2017, arXiv:1703.00887.

[47] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M. Kakade, and Michael I. Jordan, *On Nonconvex Optimization for Machine Learning: Gradients, Stochasticity, and Saddle Points*, Journal of the ACM **68.2** (2021), 1–29. arXiv:1902.04811. https://doi.org/10.1145/3418526

[48] Chi Jin, Praneeth Netrapalli, and Michael I. Jordan, *Accelerated gradient descent escapes saddle points faster than gradient descent*, Conference on Learning Theory, pp. 1042–1085, 2018, arXiv:1711.10456.

[49] Michael I. Jordan, *On gradient-based optimization: Accelerated, distributed, asynchronous and stochastic optimization*, https://www.youtube.com/watch?v=VE2ITg%5FhGnI, 2017.

[50] Stephen P. Jordan, *Fast quantum algorithm for numerical gradient estimation*, Physical Review Letters **95** (2005), no. 5, 050501, arXiv:quant-ph/0405146. https://doi.org/10.1103/PhysRevLett.95.050501

[51] Stephen P. Jordan, *Quantum computation beyond the circuit model*, Ph.D. thesis, Massachusetts Institute of Technology, 2008, arXiv:0809.2307.

[52] Iordanis Kerenidis and Anupam Prakash, *Quantum recommendation systems*, Proceedings of the 8th Innovations in Theoretical Computer Science Conference, pp. 49:1–49:21, 2017, arXiv:1603.08675. https://doi.org/10.4230/LIPIcs.ITCS.2017.49

[53] Iordanis Kerenidis and Anupam Prakash, *A quantum interior point method for LPs and SDPs*, ACM Transactions on Quantum Computing, pp. 1–32, ACM, 2020, arXiv:1808.09266. https://doi.org/10.1145/3406306

[54] Alexei Kitaev and William A. Webb, *Wavefunction preparation and resampling using a quantum computer*, 2008, arXiv:0801.0342.

[55] Kfir Y. Levy, *The power of normalization: Faster evasion of saddle points*, 2016, arXiv:1611.04831.

[56] Jianping Li, *General explicit difference formulas for numerical differentiation*, Journal of Computational and Applied Mathematics **183** (2005), no. 1, 29–52. https://doi.org/10.1016/j.cam.2004.12.026

[57] Seth Lloyd, *Universal quantum simulators*, Science **273** (1996), no. 5278, 1073. https://doi.org/10.1126/science.273.5278.1073

[58] Guang Hao Low and Isaac L. Chuang, *Optimal Hamiltonian simulation by quantum signal processing*, Physical Review Letters **118** (2017), no. 1, 010501, arXiv:1606.02685. https://doi.org/10.1103/PhysRevLett.118.010501

[59] Guang Hao Low and Isaac L. Chuang, *Hamiltonian simulation by qubitization*, Quantum **3** (2019), 163, arXiv:1610.06546. https://doi.org/10.22331/q-2019-07-12-163

[60] Guang Hao Low and Nathan Wiebe, *Hamiltonian simulation in the interaction picture*, 2018, arXiv:1805.00675.

[61] Yurii Nesterov and Boris T. Polyak, *Cubic regularization of Newton method and its global performance*, Mathematical Programming **108** (2006), no. 1, 177–205. https://doi.org/10.1007/s10107-006-0706-8

[62] Yurii E. Nesterov, *A method for solving the convex programming problem with convergence rate* $O(1/k^2)$, Soviet Mathematics Doklady, vol. 27, pp. 372–376, 1983.

[63] John Preskill, *Quantum computing in the NISQ era and beyond*, Quantum **2** (2018), 79, arXiv:1801.00862. https://doi.org/10.22331/q-2018-08-06-79

[64] Changpeng Shao and Ashley Montanaro, *Faster quantum-inspired algorithms for solving linear systems*, 2021, arXiv:2103.10309.

[65] Ju Sun, Qing Qu, and John Wright, *A geometric analysis of phase retrieval*, Foundations of Computational Mathematics **18** (2018), no. 5, 1131–1198, arXiv:1602.06664. https://doi.org/10.1007/s10208-017-9365-9

[66] Ewin Tang, *Quantum-inspired classical algorithms for principal component analysis and supervised clustering*, 2018, arXiv:1811.00414.

[67] Ewin Tang, *A quantum-inspired classical algorithm for recommendation systems*, Proceedings of the 51st Annual ACM Symposium on Theory of Computing, pp. 217–228, ACM, 2019, arXiv:1807.04271. https://doi.org/10.1145/3313276.3316310

[68] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I. Jordan, *Stochastic cubic regularization for fast nonconvex optimization*, Advances in Neural Information Processing Systems, pp. 2899–2908, 2018, arXiv:1711.02838.

[69] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J. Cerf, Timothy C. Ralph, Jeffrey H. Shapiro, and Seth Lloyd, *Gaussian quantum information*, Re-

views of Modern Physics **84** (2012), no. 2, 621, arXiv:1110.3234. https://doi.org/10.1103/RevModPhys.84.621

[70] Stephen Wiesner, *Simulations of many-body quantum systems by a quantum computer*, 1996, arXiv:quant-ph/9603028.

[71] Yi Xu, Rong Jin, and Tianbao Yang, *NEON+: Accelerated gradient methods for extracting negative curvature for non-convex optimization*, 2017, arXiv:1712.01033.

[72] Yi Xu, Rong Jin, and Tianbao Yang, *First-order stochastic algorithms for escaping from saddle points in almost linear time*, Advances in Neural Information Processing Systems, pp. 5530–5540, 2018, arXiv:1711.01944.

[73] Christof Zalka, *Efficient simulation of quantum systems by quantum computers*, Fortschritte der Physik: Progress of Physics **46** (1998), no. 6-8, 877–879. https://doi.org/10.1002/(SICI)1521-3978(199811)46:6/8<877::AID-PROP877>3.0.CO;2-A

[74] Christof Zalka, *Simulating quantum systems on a quantum computer*, Proceedings of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences **454** (1998), no. 1969, 313–322, arXiv:quant-ph/9603026. https://doi.org/10.1098/rspa.1998.0162

[75] Kaining Zhang, Min-Hsiu Hsieh, Liu Liu, and Dacheng Tao, *Quantum algorithm for finding the negative curvature direction in non-convex optimization*, 2019, arXiv:1909.07622.

[76] Yuchen Zhang, Percy Liang, and Moses Charikar, *A hitting time analysis of stochastic gradient Langevin dynamics*, Conference on Learning Theory, pp. 1980–2022, 2017, arXiv:1702.05575.

[77] Dongruo Zhou and Quanquan Gu, *Stochastic recursive variance-reduced cubic regularization methods*, International Conference on Artificial Intelligence and Statistics, pp. 3980–3990, 2020, arXiv:1901.11518.

A Auxiliary Lemmas

In this appendix, we collect all auxiliary lemmas that we use in the proofs.

A.1 Schrödinger Equation with a Quadratic Potential

In this subsection, we prove several results that lay the foundation of the quantum algorithm described in Section 2.

Lemma 1. *Suppose a quantum particle is in a one-dimensional potential field* $f(x) = \frac{\lambda}{2}x^2$ *with initial state* $\Phi(0,x) = (\frac{1}{2\pi})^{1/4}\exp(-x^2/4)$*; in other words, the initial position of this quantum particle follows the standard normal distribution* $\mathcal{N}(0,1)$*. The time evolution of this particle is governed by* (4)*. Then, at any time* $t \geq 0$*, the position of the quantum particle still follows normal distribution* $\mathcal{N}\left(0, \sigma^2(t;\lambda)\right)$*, where the variance* $\sigma^2(t;\lambda)$ *is given by*

$$\sigma^2(t;\lambda) = \begin{cases} 1 + \frac{t^2}{4} & (\lambda = 0), \\ \frac{(1+4\alpha^2)-(1-4\alpha^2)\cos 2\alpha t}{8\alpha^2} & (\lambda > 0, \alpha = \sqrt{\lambda}), \\ \frac{(1-e^{2\alpha t})^2+4\alpha^2(1+e^{2\alpha t})^2}{16\alpha^2 e^{2\alpha t}} & (\lambda < 0, \alpha = \sqrt{-\lambda}). \end{cases} \tag{5}$$

Proof. Due to the well-posedness of the Schrödinger equation, if we find a solution to the initial value problem (4), this solution is unique. We take the following ansatz

$$\Phi(t,x) = \left(\frac{1}{\pi}\right)^{1/4} \frac{1}{\sqrt{\delta(t)}} \exp(-i\theta(t)) \exp\left(\frac{-x^2}{2\delta(t)^2}\right), \tag{104}$$

with $\theta(0) = 0$, $\delta(0) = \sqrt{2}$.

In this Ansatz, the probability density $p_\lambda(t,x)$, i.e., the modulus square of the wave function, is given by

$$p_\lambda(t,x) := |\Phi(t,x)|^2 = \frac{1}{\sqrt{\pi}} \frac{1}{|\delta(t)|} \exp\Big(2\,\mathrm{Im}(\theta(t))\Big) \exp\Big(-x^2\,\mathrm{Re}(1/y(t))\Big), \tag{105}$$

where $y(t) = \delta^2(t)$.

If the ansatz (104) solves the Schrödinger equation, we will have the conservation of probability, i.e., $\|\Phi(t,x)\|^2 = 1$ for all $t \geq 0$; in other words, the $\int_{\mathbb{R}} p_\lambda(t,x)\mathrm{d}x = 1$ for all $t \geq 0$. It is now clear that (105) is the density of a Gaussian random variable with zero mean and variance

$$\sigma^2(t;\lambda) = \frac{1}{2\,\mathrm{Re}(1/y(t))}. \tag{106}$$

Therefore, it is sufficient to compute $y(t)$ in order to obtain the distribution of the quantum particle at time $t \geq 0$. For simplicity, we will not compute the global phase $\theta(t)$ as it is not important in the the variance.

Substituting the ansatz (104) to (4) with potential function $f(x) = \frac{\lambda}{2}x^2$, and introducing change of variables $y(t) = \delta^2(t)$, we attain the following system of ordinary differential equations

$$\begin{cases} y' + i\lambda y^2 - i = 0, \\ \theta' = \frac{i}{4}\frac{y'}{y} + \frac{1}{2}\frac{1}{y}, \\ \theta(0) = 0, y(0) = 2. \end{cases} \tag{107}$$

Case 1: $\lambda = 0$**.** The system (107) is linear with solutions

$$y(t) = 2 + it. \tag{108}$$

It follows that

$$\frac{1}{y(t)} = \frac{2}{4+t^2} - i\frac{t}{4+t^2}, \tag{109}$$

And by Equation (106), the variance is

$$\sigma^2(t;0) = 1 + \frac{t^2}{4}. \tag{110}$$

Case 2: $\lambda \neq 0$**.** The equation $y' + i\lambda y^2 - i = 0$ in (107) is a Riccati equation. Using the standard change of variable $y = \frac{-i}{\lambda}\frac{u'}{u}$, we transfer the Riccati equation into a second-order linear equation

$$u'' + \lambda u = 0. \tag{111}$$

Clearly, the sign of λ matters.

Case 2.1: $\lambda > 0$**.** Let $\alpha = \sqrt{\lambda}$, the solution to (111) is $u(t) = c_1 e^{i\alpha t} + c_2 e^{-i\alpha t}$ (c_1, c_2 are constants), and

$$y(t) = \frac{-i}{\lambda}\frac{u'}{u} = \frac{1}{\alpha}\frac{c_1 e^{i\alpha t} - c_2 e^{-i\alpha t}}{c_1 e^{i\alpha t} + c_2 e^{-i\alpha t}}. \tag{112}$$

Provided the initial condition $y(0) = 2$, we choose $c_1 = 1$, $\beta := c_2 = (1 - 2\alpha)/(1 + 2\alpha)$, and it turns out that

$$y(t) = \frac{1}{\alpha}\Big(\frac{e^{2i\alpha t} - \beta}{e^{2i\alpha t} + \beta}\Big). \tag{113}$$

By (106) and (113), we attain the variance when $\lambda > 0$.

Case 2.2: $\lambda < 0$**.** Let $\alpha = \sqrt{-\lambda} > 0$, similar as Case 2.1, we have

$$y(t) = \frac{i}{\alpha}\frac{e^{2\alpha t} - \beta}{e^{2\alpha t} + \beta}, \tag{114}$$

where $\beta = \frac{1+2i\alpha}{1-2i\alpha}$. And the variance $\sigma(t;\lambda)$ for $\lambda < 0$ can be obtained from (106) and (114). □

Remark 4. *Essentially, the three cases $\lambda = 0$, $\lambda > 0$, and $\lambda < 0$ in Eq. (5) can be written as a simple expression following (113) and (114). Here we present these cases separately to explicitly demonstrate that when $\lambda < 0$, the variance $\sigma^2(t;\lambda)$ grows exponentially fast in t.*

Furthermore, we prove that the argument applies to n-dimensional cases in general:

Lemma 8 (n-dimensional evolution)**.** *Let $\mathcal{H}$ be an n-by-n symmetric matrix with diagonalization $\mathcal{H} = U^T \Lambda U$, with $\Lambda = \mathrm{diag}(\lambda_1, ..., \lambda_n)$ and U an orthogonal matrix. Suppose a quantum particle is in an n-dimensional potential field $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathcal{H}\mathbf{x}$ with initial state $\Phi(0, x) = (\frac{1}{2\pi})^{n/4}\exp(-\|\mathbf{x}\|^2/4)$; in other words, the initial position of this quantum particle follows multivariate Gaussian distribution $\mathcal{N}(0, I)$. Then, at any time $t \geq 0$, the position of the quantum particle still follows multivariate Gaussian distribution $\mathcal{N}(0, \Sigma(t))$, with the covariance matrix*

$$\Sigma(t) = U^T \mathrm{diag}(\sigma^2(t;\lambda_1), ..., \sigma^2(t;\lambda_n))U. \tag{115}$$

The function $\sigma(t;\lambda)$ is defined in (5).

Proof. The proof follows the same idea in Lemma 1. We take the following ansatz

$$\Phi(t, \mathbf{x}) = \left(\frac{1}{\pi}\right)^{n/4} (\det D(t))^{-1/4} \exp(-i\theta(t)) \exp\left[-\frac{1}{2}\mathbf{x}^T (D(t))^{-1}\mathbf{x}\right], \tag{116}$$

with $\theta(0) = 0$, $D(0) = \sqrt{2}I$, and $D(t) = U^T \mathrm{diag}(\delta_1^2(t), ..., \delta_n^2(t))U$.

The global phase parameter $\theta(t)$, together with the factor $\left(\frac{1}{\pi}\right)^{n/4} (\det D(t))^{-1/4}$, will contribute to a scalar factor in the probability density function such that the L^2-norm of the wave function (116) will remain unit 1. It is the matrix $D(t)$ that controls the covariance matrix (see Eqn. 121). Regarding this, we do not delve into the derivation of $\theta(t)$ in this proof.

Substituting the ansatz (116) to the Schrödinger equation (4), we have the following system of ordinary differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(D(t)^{-1}\right) + iD(t)^{-2} - i\mathcal{H} = 0, \tag{117}$$

$$\dot{\theta} = \frac{i}{4}\left(\det D(t)\right)^{-1}\frac{\mathrm{d}}{\mathrm{d}t}\left(D(t)\right) + \frac{1}{2}\operatorname{Tr}[D(t)^{-1}]. \tag{118}$$

We immediately observe that Eq. (117) is a decoupled system

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\delta_j(t)^2}\right) + i\frac{1}{(\delta_j(t))^4} - i\lambda_j = 0, \text{ for } j = 1, ..., n. \tag{119}$$

Again, introduce change of variables $y_j(t) = \delta_j^2(t)$, we have

$$\dot{y}_j + i\lambda_j y^2 - i = 0, \text{ for } j = 1, ..., n. \tag{120}$$

They are precisely the same as the first equation in (107), thus the calculation of one-dimensional case in Lemma 1 applies directly to (120).

Given the ansatz (116), it is clear that the probability density of the quantum particle in $\mathbb{R}^n$ is an n-dimensional Gaussian with mean 0 and covariance matrix

$$\Sigma(t) = \left(2\operatorname{Re} D^{-1}(t)\right)^{-1} = U^T\left(\frac{1}{2\operatorname{Re}(1/y_1(t))}, ..., \frac{1}{2\operatorname{Re}(1/y_n(t))}\right)U. \tag{121}$$

It follows from (106) and (5) that the covariance matrix is given as (115). □

Finally, we state the following proposition with different scales:

Proposition 2. *Let $\mathcal{H}$ be an n-by-n symmetric matrix with diagonalization $\mathcal{H} = U^T\Lambda U$, with $\Lambda = \operatorname{diag}(\lambda_1, ..., \lambda_n)$ and U an orthogonal matrix. Suppose a quantum particle is in an n-dimensional potential field $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathcal{H}\mathbf{x}$ with the initial state being*

$$\Phi(0, \mathbf{x}) = \left(\frac{1}{2\pi}\right)^{n/4} r^{-n/2}\exp\left(-\|\mathbf{x}\|^2/4r^2\right); \tag{122}$$

in other words, the initial position of the particle follows multivariate Gaussian distribution $\mathcal{N}(0, r^2 I)$. The time evolution of this particle is governed by (6). Then, at any time $t \geq 0$, the position of the quantum particle still follows multivariate Gaussian distribution $\mathcal{N}(0, r^2\Sigma(t))$, with the covariance matrix

$$\Sigma(t) = U^T\operatorname{diag}(\sigma^2(t; \lambda_1), ..., \sigma^2(t; \lambda_n))U. \tag{123}$$

The function $\sigma(t; \lambda)$ is the same as in (5).

Proof. Here, we only prove the one-dimensional case, as the n-dimensional case follows almost the same manner, together with a similar argument from the proof of Lemma 8. Let $\Phi(t, x)$ be the wave function as in Lemma 1, namely, it satisfies the standard Schrödinger equation (4). Define $\Psi(t, x) = \frac{1}{\sqrt{r}}\Phi(t, \frac{x}{r})$. Since $\|\Phi(t, \cdot)\|^2 = 1$ for all $t \geq 0$, the factor $\frac{1}{\sqrt{r}}$ ensures the L^2-norm of $\Psi(t, x)$ is always 1.

We claim that $\Psi(t, x)$ satisfies the modified Schrödinger equation (6). To do so, we substitute $\Psi(t, x)$ back to (6). Its LHS is just $i\frac{\partial}{\partial t}\frac{1}{\sqrt{r}}\Phi(t, x/r)$, whereas the RHS is

$$\Big[-\frac{r^2}{2}\Delta + \frac{1}{r^2}f(\mathbf{x})\Big]\Psi(t, x) = \frac{1}{\sqrt{r}}\Big[-\frac{1}{2}\Delta + \frac{1}{2}\left(\frac{\mathbf{x}}{r}\right)^T\mathcal{H}\left(\frac{\mathbf{x}}{r}\right)\Big]\Phi\left(t, \frac{x}{r}\right). \tag{124}$$

Since $\Phi(t, x)$ satisfies (4), it turns out that the LHS equals to the RHS. Furthermore, the variance of $\Phi(t, x)$ is $\sigma^2(t; \lambda)$, and that of $\Psi(t, x) = \frac{1}{\sqrt{r}}\Phi(t, x/r)$ is simply $r^2\sigma^2(t; \lambda)$. □

Throughout the discussion, we only concern the evolution of the wave packet when it happens to center on the saddle point. However, in reality, the exact location of the saddle point is rarely known and the initial Gaussian wave may be slightly off the saddle point. In the following proposition, we investigate this more general situation in which the potential function is shifted by a distance of d. It turns out that the wave packet remains Gaussian with exactly the same rate of dispersion in its variance, while the mean of the Gaussian wave behaves like the trajectory of a classical particle, i.e., governed by the Hamiltonian mechanics $\ddot{X} = -\nabla f(X)$. Thus, we believe the source of quantum speedup in our algorithm is the variance dispersion along the negative curvature direction.

Proposition 3. *Suppose a quantum particle is in a one-dimensional potential field $f(x) = \frac{\lambda}{2}(x-d)^2$ with initial state $\Phi(0,x) = (\frac{1}{2\pi})^{1/4}\exp(-x^2/4)$; in other words, the initial position of this quantum particle follows the standard normal distribution $\mathcal{N}(0,1)$. The time evolution of this particle is governed by (4). Then, at any time $t \geq 0$, the position of the quantum particle still follows normal distribution $\mathcal{N}\left(\mu(t;\lambda), \sigma^2(t;\lambda)\right)$, where the mean $\mu(t;\lambda)$ is given by*

$$\mu(t;\lambda) = \begin{cases} 0 & (\lambda = 0), \\ d(1-\cos(\alpha t)) & (\lambda > 0, \alpha = \sqrt{\lambda}), \\ d(1-\cosh(\alpha t)) & (\lambda < 0, \alpha = \sqrt{-\lambda}), \end{cases} \tag{125}$$

while the variance $\sigma^2(t;\lambda)$ is exactly the same as in (5).

Proof. The main idea of the proof is to use the undetermined coefficient method similar to the proof of Lemma 1, though we will use a different ansatz with more parameters:

$$\Phi(t,x) = \exp\left(-a(t)x^2 + b(t)x + c(t)\right), \tag{126}$$

where $a(t)$, $b(t)$, and $c(t)$ are complex-valued functions. For simplicity, the normalization constant is absorbed in the $c(t)$ term. The probability density $p_\lambda(t,x)$, i.e., the modulus square of the wave function, is then given by

$$p_\lambda(t,x) := |\Phi(t,x)|^2 = \exp\left(-\frac{\left(x - \mathscr{B}(t)/\mathscr{A}(t)\right)^2}{1/2\mathscr{A}(t)} + \left(\mathscr{B}(t)^2/2\mathscr{A}(t) + 2\mathscr{C}(t)\right)\right), \tag{127}$$

where $\mathscr{A}(t)$, $\mathscr{B}(t)$, and $\mathscr{C}(t)$ are the real parts of the functions $a(t)$, $b(t)$, and $c(t)$, respectively. One can readily observe that $p_\lambda(t,x)$ is a Gaussian density function with mean and variance being

$$\begin{cases} \mu(t;\lambda) = \frac{\mathscr{B}(t)}{2\mathscr{A}(t)}, \\ \sigma^2(t;\lambda) = \frac{1}{4\mathscr{A}(t)}. \end{cases} \tag{128}$$

It turns out that the distribution of the quantum particle is completely determined by the mean $\mu(t)$ and variance $\sigma^2(t)$ if we can show that the ansatz function (126) indeed solves the Schrödinger equation (4) with a potential field $f(x) = \frac{\lambda}{2}(x-d)^2$.

Substituting the ansatz (126) to the Schrödinger equation (4), we obtain the following system of ordinary differential equations:

$$\begin{cases} -i\dot{a} = -2a^2 + \frac{\lambda}{2}, \\ i\dot{b} = 2ab - \lambda d, \\ i\dot{c} = a - \frac{1}{2}b^2 + \frac{\lambda}{2}d^2, \end{cases} \tag{129}$$

subject to the initial condition $a(0) = 1/4$, $b(0) = 0$, and $c(0) = -\log(2\pi)/4$. The last equation says $c(t)$ can be directly integrated as long as $a(t)$ and $b(t)$ are known. In other words, $c(t)$ exists given that $a(t)$ and $b(t)$ are determined, and we do not care about the exact value of $c(t)$ because it sheds no light on either the mean $\mu(t;\lambda)$ nor the variance $\sigma^2(t;\lambda)$. To prove the lemma, it suffices to calculate $a(t)$ and $b(t)$.

The first equation in the system (129) is a Riccati equation; by the change of variable $a = -\frac{i}{2}\frac{\dot{u}}{u}$, the Riccati equation is transformed into a second-order linear equation $\ddot{u} + \lambda u = 0$. Then, similarly, we shall discuss three cases $\lambda = 0$, $\lambda > 0$, and $\lambda < 0$. Here, we only do the $\lambda > 0$ case, as the other two cases are solved following essentially the same procedures.

Before we proceed with the calculation of $a(t)$, we discuss how the change of variable $a = -\frac{i}{2}\frac{\dot{u}}{u}$ simplifies the second equation in the system (129). With the change of variable into $i\dot{b} = 2ab - \lambda d$ and proper algebraic manipulation, we end up with the nice form

$$\dot{u}b + u\dot{b} = i\lambda d u, \tag{130}$$

Note that the left hand side is simply $\frac{\mathrm{d}}{\mathrm{d}t}(ub)$, and hence the function $b(t)$ can be expressed in terms of $u(t)$:

$$b(t) = i\lambda d \cdot \frac{\int_0^t u(s)\mathrm{d}s + C}{u(t)}, \tag{131}$$

where C is a constant.

Now, we are ready to compute both the mean and variance for the case $\lambda > 0$. Suppose $\alpha = \sqrt{\lambda}$, we have

$$u(t) = e^{i\alpha t} + ce^{-i\alpha t}, \text{ with } c = (1-2\alpha)/(1+2\alpha). \tag{132}$$

This particular choice of c will give rise to the function $a(t)$ satisfying the initial condition $a(0) = 1/4$, which reads

$$a(t) = \frac{\alpha}{2}\frac{e^{2i\alpha t} - c}{e^{2i\alpha t} + c}, \text{ with } c = (1-2\alpha)/(1+2\alpha). \tag{133}$$

Similarly, we substitute the solution of $u(t)$ (132) back into the formula for $b(t)$ (131), together with the initial condition $b(0) = 0$, we can write down the closed form of $b(t)$:

$$b(t) = \alpha r \frac{e^{2i\alpha t} - c + (c-1)e^{i\alpha t}}{e^{2i\alpha t} + c}, \text{ with } c = (1-2\alpha)/(1+2\alpha). \tag{134}$$

The real parts of $a(t)$ and $b(t)$ can then be computed as follows

$$\begin{cases} \mathscr{A}(t) = \mathrm{Re}\,(a(t)) = \frac{(1-c^2)\alpha}{2(1+c^2+2\cos(2\alpha t))}, \\ \mathscr{B}(t) = \mathrm{Re}\,(b(t)) = \alpha d \frac{(1-c^2)\big(1-\cos(\alpha t)\big)}{1+c^2+2\cos(2\alpha t)}, \end{cases} \tag{135}$$

and the mean $\mu(t;\lambda)$ and variance $\sigma^2(t;\lambda)$ follows from (128). □

A.2 Bounding the deviation from perfect Gaussian in quantum evolution

In what follows, we will use $\|\cdot\|_p$ to denote the L^p-norm of an integrable function $g\colon \Omega \to \mathbb{R}$:

$$\|g\|_p := \left(\int_\Omega |g|^p \,\mathrm{d}x\right)^{1/p}, \tag{136}$$

where $1 \le p < \infty$. For a continuous function $g\colon \Omega \to \mathbb{R}$, the L^∞ norm is $\|g\|_\infty = \sup_{x\in\Omega} |g(x)|$. For a finite-dimensional vector $\vec{v}$, we simply use $\|\vec{v}\|$ to denote its ℓ^2-norm (or the Euclidean norm):

$$\|\vec{v}\| := \left(\sum_j |v_j|^2\right)^{1/2}. \tag{137}$$

For a vector-valued function $G\colon \Omega \to \mathbb{R}^n$, we also define its L^p-norm for $1 \le p < \infty$:

$$\|G\|_p := \left(\int_\Omega \sum_{j=1}^n |G_j(x)|^p \, \mathrm{d}x\right)^{1/p}, \tag{138}$$

where $G_j(x)$ is the j-th component of the function $G(x)$. The L^∞-norm is defined in the same manner: $\|G\|_\infty = \max_{1\le j\le n} \|G_j\|_\infty$.

First, we prove the following vector norm error bound of quantum simulation:

Lemma 9 (Vector norm error bound)**.** *Let H_1, H_2 be two Hermitian operators and $H = H_1 + H_2$. Then, for any $t > 0$ and an arbitrary vector $|\varphi\rangle$, we have*

$$\left\| e^{-iH_1 t} e^{-iH_2 t} |\varphi\rangle - e^{-iHt} |\varphi\rangle \right\| \le \frac{t^2}{2} \sup_{\tau_1,\tau_2\in[0,t]} \left\| [H_1, H_2] e^{-iH_2\tau_2} e^{-iH_1\tau_1} |\varphi\rangle \right\|. \tag{139}$$

Proof. By [26, Proposition 15], we have the variation-of-parameter formula

$$e^{-iH_1 t} e^{-iH_2 t} = e^{-iHt} + \int_0^t \mathrm{d}\tau_1 \int_0^{\tau_1} \mathrm{d}\tau_2 \; e^{-iH(t-\tau_1)} e^{-iH_2\tau_1} e^{-iH_2\tau_2} [H_1, H_2] e^{-iH_2\tau_2} e^{-iH_1\tau_1}. \tag{140}$$

Thus, for an arbitrary vector $|\varphi\rangle$, we have

$$\begin{aligned} &\left(e^{-iH_1 t} e^{-iH_2 t} - e^{-iHt}\right) |\varphi\rangle \\ &\quad = \int_0^t \mathrm{d}\tau_1 \int_0^{\tau_1} \mathrm{d}\tau_2 \; e^{-iH(t-\tau_1)} e^{-iH_2\tau_1} e^{-iH_2\tau_2} [H_1, H_2] e^{-iH_2\tau_2} e^{-iH_1\tau_1} |\varphi\rangle. \end{aligned} \tag{141}$$

Since the spectral norm of the vector in the integrand is upper bounded by

$$\sup_{\tau_1,\tau_2\in[0,t]} \left\| [H_1, H_2] e^{-iH_2\tau_2} e^{-iH_1\tau_1} |\varphi\rangle \right\|, \tag{142}$$

and $\int_0^t \mathrm{d}\tau_1 \int_0^{\tau_1} \mathrm{d}\tau_2 = \frac{t^2}{2}$, we obtain the desired vector norm error bound (139). □

Second, we observe the following fact:

Theorem 6 ([12, Theorem 2, informal])**.** *For Schrödinger equations of the form*

$$i\frac{\partial}{\partial t} u + \Delta u + V(x,t) u = 0, \tag{143}$$

defined over an arbitrary finite-dimensional space with periodic boundary condition, let $u(x,t)$ be the solution at time t. If $V(x,t)$ is smooth in space and periodic in time, and the initial condition $u(x,0)$ is smooth, then we have

$$\|\nabla u(t)\|_2 \le C(\log t)^\alpha \|\nabla u(0)\|_2, \tag{144}$$

where C and α are absolute constants.

Remark 5. *The original Theorem 2 in [12] actually proved the logarithmic growth in Sobolev norm $\|u(t)\|_{H^s}$ for all $s > 0$, while we only cite the special case $s = 1$. The $\|\nabla u(0)\|_2$ term was absorbed in the constant factor in the original statement, while we feel necessary to expand it out because it may introduce dependence on n and r_0. It is worth noting that the theorem was proven for two-dimensional Schrödinger equations with quasi-periodic potential field $V(x,t)$, while it has been made clear in the context that this result holds for arbitrary-dimensional cases if V is periodic. Bourgain also explicitly discussed the periodic-V case in [13].*

Corollary 1. *For a quadratic function of the form $f_q = \frac{1}{2}(\mathbf{x}-\tilde{\mathbf{x}})^T\mathcal{H}(\mathbf{x}-\tilde{\mathbf{x}}) + F$ where $\mathcal{H}$ is a Hermitian matrix and F is a constant, consider the Schrödinger equation of the form*

$$i\frac{\partial}{\partial t}\Phi = \left[-\frac{r_0^2}{2}\Delta + \frac{1}{r_0^2}f_q\right]\Phi, \tag{145}$$

with periodic boundary conditions and initial condition $\Phi_0(x)$ defined in (7) (i.e., the initial state of the quantum simulation Algorithm 1), then we have

$$\|\nabla\Phi(t)\|_2 \leq C\sqrt{\frac{n}{r_0}}(\log t)^\alpha, \tag{146}$$

where C and α are absolute constants.

Proof. Note that the constant F just adds a global phase to the solution which does not influence either $\|\Phi(t)\|_2$ or $\|\nabla\Phi(t)\|$, and the Schrödinger equation is translation-invariant under $\mathbf{x} \to \mathbf{x} - \tilde{\mathbf{x}}$, we may assume without loss of generality that $f_q = \frac{1}{2}\mathbf{x}^T\mathcal{H}\mathbf{x}$.

Define a new function $u(\mathbf{x},t) = \Phi\left(\frac{r_0\mathbf{x}}{\sqrt{2}},t\right)$, and it is straightforward to verify that

$$i\frac{\partial}{\partial t}u + \Delta u - \frac{1}{r_0^2}f_q\left(\frac{r_0\mathbf{x}}{\sqrt{2}}\right)u = 0. \tag{147}$$

Note that the function $f_q(\mathbf{x})$ is quadratic, so $\frac{1}{r_0^2}f_q\left(\frac{r_0\mathbf{x}}{\sqrt{2}}\right) = \frac{1}{2}f_q(\mathbf{x})$, which is a constant multiple of f_q. Thus, we may directly invoke Theorem 6 to yield

$$\|\nabla u(t)\|_2 \leq C\,(\log t)^\alpha\,\|\nabla u(0)\|_2, \tag{148}$$

where the $\|\nabla u(0)\|_2$ can be directly calculated as follows:

$$\|\nabla u(0)\|_2 \leq \left(\sum_{j=1}^n \int_{\mathbb{R}^n} |u_{x_j}(\mathbf{x},0)|^2\,\mathrm{d}\mathbf{x}\right)^{1/2} = \frac{r_0}{\sqrt{2}}\left(\sum_{j=1}^n \int_{\mathbb{R}^n} |(\Phi_0)_{x_j}(r_0\mathbf{x}/\sqrt{2},0)|^2\,\mathrm{d}\mathbf{x}\right)^{1/2} \tag{149}$$

$$= \frac{\sqrt{r_0}}{2^{1/4}}\left(\sum_{j=1}^n \int_{\mathbb{R}^n} |(\Phi_0)_{x_j}(\mathbf{x},0)|^2\,\mathrm{d}\mathbf{x}\right)^{1/2} \tag{150}$$

$$= \frac{\sqrt{r_0}}{2^{1/4}}\frac{1}{2r_0^2}\left(\sum_{j=1}^n \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}r_0}e^{-(x_j-\tilde{x}_j)^2/2r_0^2}(x_j-\tilde{x}_j)^2\,\mathrm{d}x_j\right)^{1/2} = \frac{1}{2^{5/4}}\sqrt{\frac{n}{r_0}}. \tag{151}$$

Absorbing the $2^{-5/4}$ factor into the absolute constant C, we complete the proof. □

Now, we are ready to prove Lemma 3, our result of bounding the deviation from perfect Gaussian in quantum evolution.

Lemma 3. *Let $\mathcal{H}$ be the Hessian matrix of f at a saddle point $\tilde{\mathbf{x}}$, and define $f_q(\mathbf{x}) := f(\tilde{\mathbf{x}}) + \frac{1}{2}(\mathbf{x}-\tilde{\mathbf{x}})^T\mathcal{H}(\mathbf{x}-\tilde{\mathbf{x}})$ to be the quadratic approximation of the function f near $\tilde{\mathbf{x}}$. Denote the measurement outcome from the quantum simulation (see Algorithm 1) with potential field f and evolution time t_e as random variable ξ, and the measurement outcome from the quantum simulation with potential field f_q and the same evolution time t_e as another random variable ξ'. Let the law of ξ (or ξ', resp.) be $\mathbb{P}_\xi$ (or $\mathbb{P}_{\xi'}$, resp.). If the quantum wave packet is confined to a hypercube with edge length M, then*

$$TV(\mathbb{P}_\xi, \mathbb{P}_{\xi'}) \leq \left(\frac{\sqrt{n}\rho}{2} + \frac{2C_f\ell}{\sqrt{r_0}}(\log t_e)^\alpha\right)\frac{nMt_e^2}{2}, \tag{14}$$

where $TV(\cdot,\cdot)$ is the total variation distance between measures, α is an absolute constant, and C_f is an f-related constant.

Proof. Define the following (Hermitian) operators:

$$A = -\frac{r_0^2}{2}\Delta, \quad B = \frac{1}{r_0^2}f, \quad B' = \frac{1}{r_0^2}f_q, \tag{152}$$

$$H = A + B, \; H' = A + B', \; E = H - H' = \frac{1}{r_0^2}(f - f_q). \tag{153}$$

Let $|\Phi(t)\rangle = e^{-iHt}|\Phi_0\rangle$ be the wave function generated by the quantum simulation with potential field f and evolution time t, and similarly, $|\Phi'(t)\rangle := e^{-iH't}|\Phi_0\rangle$ as the wave function generated by the quantum simulation with potential field f_q and the evolution time t.

By Lemma 9, and notice that E is a scalar-valued function, we have

$$\left\|e^{-iEt_e}|\Phi'(t_e)\rangle - |\Phi(t_e)\rangle\right\|_2 \leq \frac{t_e^2}{2}\sup_{\tau_1,\tau_2\in[0,t]}\left\|[H',E]e^{-iE\tau_2}e^{-iH'\tau_1}|\Phi_0\rangle\right\|_2 \tag{154}$$

$$= \frac{t_e^2}{2}\sup_{\tau_1\in[0,t]}\left\|[H',E]e^{-iH'\tau_1}|\Phi_0\rangle\right\|_2. \tag{155}$$

Denote $|\Psi(\tau_1)\rangle := e^{-iH'\tau_1}|\Phi_0\rangle$. Note that $[H',E] = [A+B',E]$ and B' commutes with E, we have

$$\sup_{\tau_1\in[0,t]}\|[H',E]\Psi(\tau_1)\|_2 = \frac{1}{2}\sup_{\tau_1\in[0,t]}\|[-\Delta, f-f_q]\Psi(\tau_1)\|_2 \tag{156}$$

$$= \frac{1}{2}\sup_{\tau_1\in[0,t]}\|-\Delta(f-f_q)\Psi(\tau_1) - 2\nabla(f-f_q)\cdot\nabla\Psi(\tau_1)\|_2 \tag{157}$$

$$\leq \frac{1}{2}\|\Delta(f-f_q)\|_\infty + \|\nabla(f-f_q)\|_\infty\|\nabla\Psi(\tau_1)\|_2. \tag{158}$$

The second equality follows from the fact that $[-\Delta, g]\varphi = -(\Delta g)\varphi - 2\nabla g\cdot\nabla\varphi$ for smooth functions g and φ. The last step follows from the triangle inequality (and the fact that $\|\Psi(\tau_1)\| = 1$). By the ρ-Hessian Lipschitz condition of f, we have

$$|\Delta(f(\mathbf{x}) - f_q(\mathbf{x}))| = \left|\mathrm{tr}\Big(\nabla^2 f(\mathbf{x}) - \nabla^2 f_q(\mathbf{x})\Big)\right| = \left|\mathrm{tr}\Big(\nabla^2 f(\mathbf{x}) - \nabla^2 f(\tilde{\mathbf{x}})\Big)\right| \tag{159}$$

$$\leq n\|\nabla^2 f - \nabla^2 f(\tilde{\mathbf{x}})\| \leq n^{3/2}\rho M, \tag{160}$$

where the second equality holds because f_q is a quadratic form and $\nabla^2 f_q(\mathbf{x}) = \mathcal{H} = \nabla^2 f(\tilde{\mathbf{x}})$. Note that the diameter of the hypercube domain is $n^{1/2}M$, and the last step follows from the ρ-Hessian Lipschitz condition. It turns out that

$$\|\Delta(f - f_q)\|_\infty \leq n^{3/2}\rho M. \tag{161}$$

Next, we bound the L^∞-norm of the gradient of $f - f_q$:

$$\|\nabla f - \nabla f_q\|_\infty \leq \sup_{\mathbf{x}} \|\nabla f(\mathbf{x}) - \nabla f_q(\mathbf{x})\| = \sup_{\mathbf{x}} \|\nabla f(\mathbf{x}) - \mathcal{H}(\mathbf{x} - \tilde{\mathbf{x}})\| \tag{162}$$

$$\leq \sup_{\mathbf{x}} \|\nabla f(\mathbf{x})\| + \sup_{\mathbf{x}} \|\mathcal{H}(\mathbf{x} - \tilde{\mathbf{x}})\|, \tag{163}$$

where the last step uses the triangle inequality. Note that $\tilde{\mathbf{x}}$ is a stationary point of f, so $\nabla f(\tilde{\mathbf{x}}) = 0$. By the ℓ-smoothness condition of f, we obtain

$$\sup_{\mathbf{x}} \|\nabla f(\mathbf{x})\| = \sup_{\mathbf{x}} \|\nabla f(\mathbf{x}) - \nabla f(\tilde{\mathbf{x}})\| \leq \ell \sup_{\mathbf{x}} \|\mathbf{x} - \tilde{\mathbf{x}}\| \leq \ell n^{1/2} M. \tag{164}$$

Meanwhile, the ℓ-smoothness of f implies that $\|\nabla^2 f(\mathbf{x})\| \leq \ell$ for all $\mathbf{x} \in \mathbb{R}^n$, therefore $\|\mathcal{H}\| \leq \ell$ and

$$\sup_{\mathbf{x}} \|\mathcal{H}(\mathbf{x} - \tilde{\mathbf{x}})\| \leq \ell n^{1/2} M. \tag{165}$$

Plugging (164) and (165) to (163), we end up with

$$\|\nabla(f - f_q)\|_\infty \leq 2\ell n^{1/2} M. \tag{166}$$

The upper bound for $\sup_{\tau_1} \|\nabla\Psi(\tau_1)\|_2$ is given by Corollary 1. Combining all three bounds, we end up with

$$\left\| e^{-iEt_e} |\Phi'(t_e)\rangle - |\Phi(t_e)\rangle \right\|_2 \leq \left(\frac{\sqrt{n}\rho}{2} + \frac{2C\ell}{\sqrt{r_0}} (\log t_e)^\alpha \right) \frac{nMt_e^2}{2}. \tag{167}$$

In what follows, we will simply write Ψ' for $\Psi'(t_e)$. We also denote $|\Psi''\rangle := e^{-iEt_e} |\Psi'\rangle$. Note that e^{-iEt_e} is actually a scalar function with modulus 1, hence the two wave functions $|\Psi'\rangle$ and $|\Psi''\rangle$ yield the same probability density, i.e., $|\Psi'|^2 = |\Psi''|^2$. By the definition of total variation distance,

$$TV(\mathbb{P}_\xi, \mathbb{P}_{\xi'}) = TV(|\Psi|^2, |\Psi''|^2) \tag{168}$$

$$= \frac{1}{2} \int_{x \in \mathbb{R}^n} \left| \Psi\overline{\Psi} - \Psi''\overline{\Psi''} \right| \mathrm{d}x \tag{169}$$

$$\leq \frac{1}{2} \int_{x \in \mathbb{R}^n} \left| (\Psi - \Psi'')\overline{\Psi} \right| \mathrm{d}x + \frac{1}{2} \int_{x \in \mathbb{R}^n} \left| \Psi''(\overline{\Psi} - \overline{\Psi''}) \right| \mathrm{d}x \tag{170}$$

$$\leq \left(\int_{x \in \mathbb{R}^n} |\Psi - \Psi''|^2 \mathrm{d}x \right)^{1/2} \leq \left(\frac{\sqrt{n}\rho}{2} + \frac{2C\ell}{\sqrt{r_0}} (\log t_e)^\alpha \right) \frac{nMt_e^2}{2}. \tag{171}$$

□

A.3 Variance of Gaussian Wave Packets

Although the variance of the Gaussian wave packet $\sigma(\lambda;t)$ is explicitly given in (5), it is a bit heavy to use in analysis. In this subsection, we prove several lemmas that can be utilized to estimate the variance $\sigma(\lambda;t)$. Based on these lemmas, it is then possible to quantify the performance of Algorithm 2.

Lemma 10. *When $\lambda > 0$,*

$$\min\left\{1, \frac{1}{2\alpha}\right\} \leq \sigma(t;\lambda) \leq \max\left\{1, \frac{1}{2\alpha}\right\}. \tag{172}$$

When $\lambda < 0$, let $\alpha = \sqrt{-\lambda}$,

$$\frac{1}{\sqrt{2}}\varphi(t;\alpha) \leq \sigma(t;\lambda) \leq \varphi(t;\alpha), \tag{173}$$

with $\varphi(t;\alpha) = \frac{1}{2\alpha}\sinh(\alpha t) + \cosh(\alpha t)$.

Proof. The first estimate follows from $\cos 2\alpha t \in [0,1]$, while the second estimate follows from the inequality

$$\frac{a+b}{2} \leq \sqrt{\frac{a^2+b^2}{2}} \leq \frac{a+b}{\sqrt{2}}. \tag{174}$$

□

Lemma 11. *When $\lambda < 0$,*

$$\sigma^2(t;\lambda) \geq 1 + \frac{t^2}{4}. \tag{175}$$

Proof. Recall (5) that $\sigma(t;\lambda)$ equals to:

$$\sigma^2(t;\lambda) = \frac{(1-e^{2\alpha t})^2 + 4\alpha^2(1+e^{2\alpha t})^2}{16\alpha^2 e^{2\alpha t}}, \tag{176}$$

in which $\alpha = \sqrt{-\lambda}$. The equation above can be converted to:

$$\sigma^2(t;\lambda) = \frac{(1+4\alpha^2)e^{4\alpha t} + (1+4\alpha^2) - 2(1-4\alpha^2)e^{2\alpha t}}{16\alpha^2 e^{2\alpha t}} \tag{177}$$

$$= \frac{(1+4\alpha^2)e^{2\alpha t} + (1+4\alpha^2)e^{-2\alpha t} - 2(1-4\alpha^2)}{16\alpha^2}. \tag{178}$$

We denote $\mu := 2\alpha t$. Note that $\mu > 0$. By the Taylor expansion of e^{μ} with Lagrange form of remainder, there exists real numbers $\zeta, \xi \in (0,\mu)$ such that

$$e^{\mu} = 1 + \mu + \frac{\mu^2}{2} + \frac{\mu^3}{6} + \frac{e^{\zeta}}{24}\mu^4; \tag{179}$$

$$e^{-\mu} = 1 - \mu + \frac{\mu^2}{2} - \frac{\mu^3}{6} + \frac{e^{-\xi}}{24}\mu^4. \tag{180}$$

Adding these two equations, we have

$$e^{\mu} + e^{-\mu} \geq 2 + \mu^2 + \frac{\mu^4}{24}(e^{\zeta} + e^{-\xi}) \geq 2 + \mu^2 + \frac{\mu^4}{24}(1 + e^{-\mu}) \geq 2 + \mu^2. \tag{181}$$

In other words,

$$e^{2\alpha t} + e^{-2\alpha t} \geq 2 + (2\alpha t)^2, \tag{182}$$

which results in

$$\frac{(1+4\alpha^2)e^{2\alpha t} + (1+4\alpha^2)e^{-2\alpha t} - 2(1-4\alpha^2)}{16\alpha^2} \geq \frac{(1+4\alpha^2)(2+4\alpha^2 t^2) - 2(1-4\alpha^2)}{16\alpha^2} \tag{183}$$

$$\geq \frac{16\alpha^2 + 4\alpha^2 t^2}{16\alpha^2} \tag{184}$$

$$= 1 + \frac{t^2}{4}; \tag{185}$$

or equivalently,

$$\sigma^2(t;\lambda) \geq 1 + \frac{t^2}{4}. \tag{186}$$

□

In the proof of Proposition 1, we will also use the following fact about multivariate Gaussian distributions:

Lemma 12 ([44, Proposition 1])**.** *Let $A \in \mathbb{R}^{m\times n}$ be a matrix, and let $\Sigma := A^T A$. Let $\mathbf{x} = (x_1, \cdots, x_n)$ be an isotropic multivariate Gaussian random vector with mean zero. For all $t > 0$:*

$$\mathbb{P}\Big(\|A\mathbf{x}\|^2 > \mathrm{tr}(\Sigma) + 2\sqrt{\mathrm{tr}(\Sigma^2)t} + 2\|\Sigma\|t\Big) \leq e^{-t}. \tag{187}$$

A.4 Existing Lemmas

In this subsection, we list existing lemmas from [47, 48] that we use in our proof.

First, we use the following lemma for the large gradient scenario of gradient descent method:

Lemma 13 ([47, Lemma 19])**.** *If $f(\cdot)$ is ℓ-smooth and ρ-Hessian Lipschitz, $\eta = 1/\ell$, then the gradient descent sequence $\{\mathbf{x}_t\}$ satisfies:*

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq \eta\|\nabla f(\mathbf{x})\|^2, \tag{188}$$

for any step t in which quantum simulation is not called.

The next lemmas are frequently used in the large gradient scenario of the accelerated gradient descent method:

Lemma 14 ([48, Lemma 7])**.** *Consider the setting of Theorem 4. If we have* $\|\nabla f(\mathbf{x}_\tau)\| \geq \epsilon$ *for all* $\tau \in [0, \mathscr{T}]$*, then there exists a large enough positive constant* c_{A0}*, such that if we choose* $c_A \geq c_{A0}$*, by running Algorithm 3 we have* $E_{\mathscr{T}} - E_0 \leq -\mathscr{E}$*, in which* $\mathscr{E} = \sqrt{\frac{\epsilon^3}{\rho}} \cdot c_A^{-7}$*, and* E_τ *is defined as:*

$$E_\tau := f(\mathbf{x}_\tau) + \frac{1}{2\eta'}\|\mathbf{v}_\tau\|^2 \tag{189}$$

where $\eta' = \frac{1}{4\ell}$ *as in Theorem 4.*

Note that this lemma is not exactly the same as Lemma 7 of [48]: to be more specific, they have an extra ι^{-5} term appearing in the $\mathscr{E}$. However, this term actually only appears when we need to escape from a saddle point using the original AGD algorithm. In large gradient scenarios where the gradient is greater than ϵ, it does not make a difference if we ignore this ι^{-5} term.

Lemma 15 ([48, Lemma 4 and Lemma 5])**.** *Assume that the function* f *is* ℓ*-smooth. Consider the setting of Theorem 4, for every iteration* τ *where quantum simulation was not called, we have*

$$E_{\tau+1} \leq E_\tau, \tag{190}$$

where E_τ *is defined in (189) in Lemma 14.*

The correctness of these two lemmas above is guaranteed by two mechanisms. If the function does not have a large negative curvature between $\mathbf{x}_t$ and $\mathbf{y}_t$ in the current iteration, the AGD will simply make the Hamiltonian decrease efficiently. Otherwise, the Negative-Curvature-Exploitation procedure in Line 11 of Algorithm 3 will be triggered (same as in [48]) and decrease the Hamiltonian by either finding the minimum function value in the nearby region of $\mathbf{x}_t$ if $\mathbf{v}_t$ is small, or directly resetting $\mathbf{v}_t = 0$ if it is large.