

# Quantum Algorithms for Escaping from Saddle Points

Chenyi Zhang<sup>\*1</sup>, Jiaqi Leng<sup>\*2,3</sup>, and Tongyang Li<sup>†3,4,5</sup>

<sup>1</sup>Institute for Interdisciplinary Information Sciences, Tsinghua University

<sup>2</sup>Department of Mathematics, University of Maryland

<sup>3</sup>Joint Center for Quantum Information and Computer Science, University of Maryland

<sup>4</sup>Department of Computer Science and Institute for Advanced Computer Studies,  
University of Maryland

<sup>5</sup>Center for Theoretical Physics, Massachusetts Institute of Technology

## Abstract

We initiate the study of quantum algorithms for escaping from saddle points with provable guarantee. Given a function  $f: \mathbb{R}^n \rightarrow \mathbb{R}$ , our quantum algorithm outputs an  $\epsilon$ -approximate second-order stationary point using  $\tilde{O}(\log^2 n / \epsilon^{1.75})$ <sup>1</sup> 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  $n$  and matches its complexity in terms of  $1/\epsilon$ . Our quantum algorithm is built upon two techniques: First, we replace the classical perturbations in gradient descent methods by simulating quantum wave equations, which constitutes the polynomial speedup in  $n$  for escaping from saddle points. Second, we 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 quantum speedup.

## 1 Introduction

**Motivations.** Nonconvex optimization has been 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 the global optima 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 (see e.g. [11, 31–34, 39]). On the other hand, it is known that saddle points (and local maxima) can give highly suboptimal solutions in many problems; see e.g. [41, 59]. Furthermore, saddle points are ubiquitous in high-dimensional nonconvex optimization problems [14, 25, 29].

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

- $f$  is  $\ell$ -smooth:  $\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \leq \ell \|\mathbf{x}_1 - \mathbf{x}_2\| \quad \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ ,

---

<sup>\*</sup>Equal contribution.

<sup>†</sup>Corresponding author. Email: tongyang@mit.edu

<sup>1</sup>The  $\tilde{O}$  notation omits poly-logarithmic terms, i.e.,  $\tilde{O}(g) = O(g \text{ poly}(\log g))$ .

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

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

$$\|\nabla f(\mathbf{x}_\epsilon)\| \leq \epsilon, \quad \lambda_{\min}(\nabla^2 f(\mathbf{x}_\epsilon)) \geq -\sqrt{\rho\epsilon}. \quad (1.1)$$

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  $\text{poly}(\log n)$  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  $\epsilon$ -approximate local minimum by second-order methods; for instance, Ref. [56] takes  $O(1/\epsilon^{1.5})$  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. [31], which can find an  $\epsilon$ -approximate local minimum satisfying (1.1) only using the first-order oracle, i.e., gradients. Although this paper has a  $\text{poly}(n)$  dependence in the query complexity of the oracle, the follow-up work [42] can 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 [44]. 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 [45].

On the other hand, quantum computing has been a rapidly advancing technology. In particular, the capability of quantum computers has been dramatically increasing and recently reached “quantum supremacy” [58] 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 [57], stochastic gradient Langevin dynamics [70] or Hamiltonian Monte Carlo [30], 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 learning algorithms have been an emerging topic in theoretical computer science [17–19, 36, 60, 61], 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, 12, 13, 48] and general convex optimization [6, 16]; 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

---

<sup>2</sup>In general, we can ask for an  $(\epsilon_1, \epsilon_2)$ -approximate local minimum  $\mathbf{x}$  such that  $\|\nabla f(\mathbf{x})\| \leq \epsilon_1$  and  $\lambda_{\min}(\nabla^2 f(\mathbf{x})) \geq -\epsilon_2$ . The scaling in (1.1) was first adopted by [56] and has been taken as a standard by subsequent works (e.g. [1, 15, 27, 42–44, 62, 65, 66]).

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 [38], and will demonstrate significant novelty if the quantum speedup compared to the classical counterparts is more than quadratic.

**Contributions.** Our main contribution is a quantum algorithm that finds an  $\epsilon$ -approximate local minimum with polynomial quantum speedup in  $n$  compared to the classical state-of-the-art algorithm [44] using the gradient oracle (i.e., the first-order oracle). 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  $\mathbf{x} \in \mathbb{R}^n$ ,

$$U_f|\mathbf{x}\rangle|0\rangle = |\mathbf{x}\rangle|f(\mathbf{x})\rangle. \quad (1.2)$$

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

$$U_f \sum_{i=1}^m \mathbf{c}_i |\mathbf{x}_i\rangle |0\rangle = \sum_{i=1}^m \mathbf{c}_i |\mathbf{x}_i\rangle |f(\mathbf{x}_i)\rangle. \quad (1.3)$$

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 (1.2) can be implemented by quantum arithmetic circuits of about the same size; as a result, it has been the standard assumption in previous literature on quantum algorithms for convex optimization [6, 16], and we subsequently adopt it here for nonconvex optimization.

**Theorem 1.1** (Main result, informal). *There is a quantum algorithm that finds an  $\epsilon$ -approximate local minimum using  $\tilde{O}(\log^2 n / \epsilon^{1.75})$  queries to the quantum evaluation oracle (1.2).*

Technically, our work is inspired by both the perturbed gradient descent (PGD) algorithm in [42, 43] and the perturbed accelerated gradient descent (PAGD) algorithm in [44]. 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 [44] adopts the similar perturbation idea, but the GD is replaced by Nesterov’s AGD [57].

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 is able to screen 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  $\epsilon$ , the classical PGD takes  $O(\log n)$  steps to decrease the function value by  $\Omega(1/\log^3 n)$  with high probability, and the PAGD takes  $O(\log n)$  steps to decrease the function value by  $\Omega(1/\log^5 n)$ .

with high probability. Quantumly, the simulation of the Schrödinger equation for time  $t$  takes  $\tilde{O}(t \log n)$  evaluation queries, but simulation for time  $t = O(\log n)$  and  $O(\log n)$  subsequent GD iterations suffice to decrease the function value by  $\Omega(1)$  with high probability. In addition, our quantum algorithm is *classical-quantum hybrid*: the transition between consecutive iterations is still classical, while the only quantum computing part happens in each iteration for replacing the classical uniform perturbation.

On the other hand, for the gradient descent steps, we replace them by a quantum algorithm for computing gradients using also quantum evaluation queries. The idea was initiated by Jordan [46] who computed the gradient at a point by applying the quantum Fourier transform on a mesh near the point. Prior arts have shown how to apply Jordan’s algorithm for convex optimization [6, 16], and we conduct a detailed analysis showing how in nonconvex optimization we replace classical gradient queries by the same number of quantum evaluation queries. Technically, we essentially show the robustness of escaping from saddle points by PGD, which may be of independent interest.

Finally, we perform numerical experiments that support our quantum speedup. 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
[24, 56]	$O(1/\epsilon^{1.5})$	Hessian
[1, 15]	$\tilde{O}(\log n/\epsilon^{1.75})$	Hessian-vector product
[42, 43]	$\tilde{O}(\log^4 n/\epsilon^2)$	Gradient
[44]	$\tilde{O}(\log^6 n/\epsilon^{1.75})$	Gradient
<b>this work</b>	$\tilde{O}(\log^2 n/\epsilon^{1.75})$	Quantum evaluation

Table 1: A summary of the state-of-the-art works on finding approximate second-order stationary points using different oracles. The query complexities are highlighted in terms of the dimension  $n$  and the error  $\epsilon$ .

**Related works.** Escaping from saddle points by gradients was initiated by Ge et al. [31] with complexity  $O(\text{poly}(n/\epsilon))$ . The follow-up work [50] improved it to  $O(n^3 \text{poly}(1/\epsilon))$ , but it is still polynomial in dimension  $n$ . The breakthrough result by Jin et al. [42, 43] 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 [44] (the same result in terms of  $\epsilon$  was independently obtained by [3, 65]). 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, 15].

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 [31] showed that noisy stochastic gradient descent (SGD) finds second-order stationary points in  $O(\text{poly}(n)/\epsilon^4)$  iterations. This is later improved to  $\tilde{O}(\text{poly}(\log n)/\epsilon^{3.5})$  [2, 3, 27, 62, 66], and the current state-of-the-art iteration complexity of stochastic algorithms is  $\tilde{O}(\text{poly}(\log n)/\epsilon^3)$  due to [26, 71].

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 [69], 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 [47] which can actually be dequantized to classical algorithms with comparable cost [17, 60, 61]; 3) It can only apply negative curvature finding for one iteration. In all, it is unclear whether this quantum algorithm achieves speedup for escaping from saddle points.

**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 for future exploration is to understand the potential speedup of using (classical) mechanical waves.
- Can quantum algorithms achieve speedup in terms of  $1/\epsilon$ ? The current quantum 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.

**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) [43, Algorithm 4] and the perturbed accelerated gradient descent (PAGD) [44, 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:

$$i \frac{\partial}{\partial t} \Phi = \left[ -\frac{1}{2} \nabla^2 + f(\mathbf{x}) \right] \Phi, \quad (2.1)$$

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

**Lemma 2.1.** *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 (2.1). Then, at any time  $t \geq 0$ , the position of the quantum particle still follows normal distribution  $\mathcal{N}(0, \sigma^2(t; \lambda))$ , where the variance  $\sigma^2(t; \lambda)$  is given by

$$\sigma^2(t; \lambda) = \begin{cases} 1 + \frac{t^2}{4} & (\lambda = 0), \\ \frac{(1+4\alpha^2) - (1-4\alpha^2)\cos 2\alpha t}{8\alpha^2} & (\lambda > 0, \alpha = \sqrt{\lambda}), \\ \frac{(1-e^{2\alpha t})^2 + 4\alpha^2(1+e^{2\alpha t})^2}{16\alpha^2 e^{2\alpha t}} & (\lambda < 0, \alpha = \sqrt{-\lambda}). \end{cases} \quad (2.2)$$

**Lemma 2.1** 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 [64, 67, 68]. There has been a rich literature on the cost of quantum simulation, see for example [9, 10, 20, 52–54]; it is typically linear in the evolution time, which is formally known as the “no-fast-forwarding theorem” (see [9, Theorem 3] and [21, Theorem 3]). In **Section 2.1.1**, we prove the following lemma about the cost of simulating the Schrödinger equation using the quantum evaluation oracle in (1.2):

**Lemma 2.2.** *Let  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  be a real-valued function with a saddle point at  $x = 0$  and  $f(0) = 0$ . Suppose  $f(x) \approx \frac{1}{2}x^T \mathcal{H} x$  in a hypercube region  $\Omega = \{x \in \mathbb{R}^n : \|x\|_\infty \leq Mr_0\}$  for some universal upper bound  $M > 0$ . Consider the (scaled) Schrödinger equation*

$$i \frac{\partial}{\partial t} \Phi = \left[ -\frac{r_0^2}{2} \nabla^2 + \frac{1}{r_0^2} f(\mathbf{x}) \right] \Phi \quad (2.3)$$

*defined on the compact domain  $\Omega$  with Dirichlet boundary condition. Given the quantum evaluation oracle  $U_f|\mathbf{x}\rangle|0\rangle = |\mathbf{x}\rangle|f(\mathbf{x})\rangle$  in (1.2) and an arbitrary initial state at time  $t = 0$ , the evolution of (2.3) for time  $t > 0$  can be simulated using  $\tilde{O}(t \log n \log^2(\frac{t}{\epsilon}))$  queries to  $U_f$ , where  $\epsilon$  is the precision.*

In practice, nevertheless, the objective function  $f(\mathbf{x})$  is not necessarily quadratic, and the second order approximation is only valid near a small neighborhood of the saddle point  $\mathbf{x}^*$ :  $f(\mathbf{x}) \approx f(\mathbf{x}^*) + \frac{1}{2}\mathbf{x}^T \mathcal{H} \mathbf{x}$ , where  $\mathcal{H} = D^2 f(\mathbf{x}^*)$  is the Hessian matrix of  $f$  at the saddle point  $\mathbf{x}^*$ , and the gradient vanishes at the saddle point  $\mathbf{x}^*$ . Without loss of generality, we translate the saddle point to the origin and set  $f(\mathbf{x}^*) = 0$ ; the potential field is now

$$f(\mathbf{x}) \approx \frac{1}{2}\mathbf{x}^T \mathcal{H} \mathbf{x}. \quad (2.4)$$

Because we have assumed that  $f$  is Hessian-Lipschitz, such approximation is legitimate if we take in a ball 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 2.2** in this subsection. Before doing that, we want to briefly discuss the reason why we simulate the scaled Schrödinger equation (2.3) instead of the common version of non-relativistic Schrödinger equation

$$i \frac{\partial}{\partial t} \Phi = \left[ -\frac{1}{2} \nabla^2 + f(\mathbf{x}) \right] \Phi. \quad (2.6)$$



---

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

---

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

$$\Phi_0(\mathbf{x}) = \left(\frac{1}{2\pi}\right)^{n/4} \frac{1}{r_0^{n/2}} \exp(-(\mathbf{x} - \tilde{\mathbf{x}})^2/4r_0^2); \quad (2.5)$$

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.

---

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  $\Omega$  defined in [Lemma 2.2](#). Regarding this issue, it is necessary to scale the spatial variable in order to make the wave packet more localized. However, the scaling in spatial variable will simultaneously cause a scaling in the time variable under Eq. (2.6). This is not preferable because the scaling in time can dramatically change the variance  $\sigma(t; \lambda)$  in (2.2), 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 (2.3), in which  $r_0$  is a small number such that  $f(x) \approx f(x_0) + \frac{1}{2}x^T \mathcal{H}x$  within a range  $O(r_0)$ . We justify our choice of (2.3) in three aspects:

- **Geometric aspect:** Eq. (2.3) is obtained by considering a spatial dilation in the wave function  $\Phi(t, x) \mapsto \Phi(t, x/r)$  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 A.1](#)). 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 disparity, we want to introduce a small  $r^2$  factor for the kinetic energy operator  $-\frac{1}{2}\nabla^2$  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 (2.3) 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 has been a well-established research topic; see e.g. [\[9, 10, 20, 52–54\]](#) for detailed results and proofs. In this paper, we apply quantum simulation under the interaction picture [\[55\]](#). In particular, we use the following result:

**Theorem 2.1** ([\[55, Theorem 7\]](#)). *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 by using*

$$O\left(\alpha_B t \left(\log(t(\alpha_A + \alpha_B)/\epsilon)\right) \frac{\log(\alpha_B t/\epsilon)}{\log \log(\alpha_B t/\epsilon)}\right) \quad (2.7)$$

*queries to the unitary oracle  $O_B$ .*

Our [Lemma 2.2](#) is inspired by Costa et al. [23] 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,  $-\frac{1}{2}\nabla^2$  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,

$$-\frac{1}{a^2}[L\phi]_j = \frac{\phi_{j-1} - 2\phi_j + \phi_{j+1}}{a^2}, \quad (2.8)$$

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

The discretization method used in [23] 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 [51], 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 [22, Section 3] 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 2.2](#).

*Proof.* There are two steps in the quantum simulation of (2.3): (1) discretizing the spatial domain using (2.8) so that the Schrödinger equation (2.3) is reduced to an ordinary differential equation (2.9); (2) simulating (2.9) under the interaction picture. In each step, we fix the error tolerance as  $\epsilon/2$ . By the triangle inequality, the overall error is  $\epsilon$ .

First, we consider the  $k$ -th order finite difference method in [22, Section 3] (the discrete Laplacian will be denoted as  $L_k$ ). With the spacing between grid points being  $a$ , if we choose the mesh number along each direction as  $1/a = \text{poly}(n) \text{poly}(\log(2/\epsilon))$ , the finite difference error will be of order  $\epsilon/2$ . Then the Schrödinger equation in (2.3) becomes

$$i\frac{\partial}{\partial t}\Phi = \left(-\frac{r_0^2}{2a^2}L_k + B\right)\Phi, \quad (2.9)$$

where  $L_k$  is the Laplacian matrix associated to the  $k$ -th order finite difference method (discretization of the hypercube  $\Omega$ ) 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 [22], the spectral norm of  $L_k$  is of order  $O(n/a^2) = \text{poly}(n) \text{poly}(\log(2/\epsilon))$ , where  $n$  is the spatial dimension of the Schrödinger equation.

We simulate the evolution of (2.9) by [Theorem 2.1](#) and taking  $A = -\frac{r_0^2}{2a^2}L_k$  therein. Recall that  $\|L_k\| \leq \text{poly}(n) \text{poly}(\log(2/\epsilon))$ . The maximal absolute value of function  $f(x)$  on  $\Omega$  is approximately  $M^2\|\mathcal{H}\|$ . Therefore, we have  $\alpha_A \leq Cr_0^2 \text{poly}(n) \text{poly}(\log(2/\epsilon))$  where  $C > 0$  is an absolute constant, and  $\alpha_B = M^2\|\mathcal{H}\|$ . It follows from [Theorem 2.1](#) 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

$$O\left(M^2\|\mathcal{H}\|t\left(\log(t(Cr_0^2 \text{poly}(n) \text{poly}(\log(2/\epsilon))) + M^2\|\mathcal{H}\|)/\epsilon\right)\frac{\log(M^2\|\mathcal{H}\|t/\epsilon)}{\log \log(M^2\|\mathcal{H}\|t/\epsilon)}\right). \quad (2.10)$$

Absorbing all absolute constants in the big  $O$  notation, the total quantum queries to  $f$  reduces to  $\tilde{O}(t \log n \log^2(\frac{t}{\epsilon}))$  as claimed in [Lemma 2.2](#).  $\square$



**Remark 2.1.** 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 [\[49\]](#); Gaussian states are also ubiquitous in the literature of continuous-variable quantum information, for instance [\[63\]](#). However, although when  $f$  is quadratic the evolution of the Schrödinger equation keeps the state being a Gaussian wave packet by [Lemma A.1](#), 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 [\(1.2\)](#) 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  $t_e = \mathcal{T}'$ , 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  $t_{\text{perturb}} = 0;$ 
2 for  $t = 0, 1, \dots, T$  do
3   if  $\|\nabla f(\mathbf{x}_t)\| \leq \epsilon$  and  $t - t_{\text{perturb}} > \mathcal{T}'$  then
4      $\mathbf{x}_t \leftarrow \mathbf{x}_t - \eta \xi_t$ , where  $\xi_t \sim \text{QuantumSimulation}(\mathbf{x}_t, r_0, \mathcal{T}')$ ;
5      $t_{\text{perturb}} = t;$ 
6    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t);$ 

```

---

### 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 has been the standard assumption as in [\[44\]](#)); 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 wave packet still disperses fast. Formally, we prove the following probability bound:

**Proposition 2.1.** We specify our choices for the parameters and constants that we use:

$$\iota := k \cdot \log(n\ell(f(\mathbf{x}_0) - f^*)/(\rho\epsilon\delta)) \quad \eta := \frac{1}{\ell} \quad r_0 := \frac{\epsilon}{\iota^3} \cdot \frac{1}{\sqrt{n}} \quad (2.11)$$

$$\mathcal{T}' := \frac{\ell}{\sqrt{\rho\epsilon}} \cdot \iota \quad \mathcal{T} := \sqrt{\frac{\epsilon}{\rho^3}} \quad \alpha'_0 := \frac{\sqrt{2}}{(1 + \eta\sqrt{\rho\epsilon})^{\mathcal{T}'}} \cdot \frac{1}{\rho} \quad (2.12)$$

where  $k$  is an absolute constant. Let  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  be an  $\ell$ -smooth,  $\rho$ -Hessian Lipschitz function. For a saddle point  $\tilde{\mathbf{x}}$  satisfying  $\|\nabla f(\tilde{\mathbf{x}})\| = 0$  and  $\lambda_{\min}(\nabla^2 f(\tilde{\mathbf{x}})) \leq -\sqrt{\rho\epsilon}$ , [Algorithm 2](#) satisfies:

$$\mathbb{P}\left(f(\mathbf{x}_{\mathcal{T}'}) - f(\tilde{\mathbf{x}}) \leq -\frac{\mathcal{T}'}{2}\right) \geq 1 - \frac{2}{\ell} \sqrt{\frac{n\ell^4}{\pi\rho\epsilon}} \cdot 2^{-\iota} - \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1 + \ell^2)}\right), \quad (2.13)$$

where  $\mathbf{x}_{\mathcal{T}'}$  is the  $\mathcal{T}'$ <sup>th</sup> GD iteration starting from  $\mathbf{x}_0$ , if [Algorithm 1](#) was called at  $t = 0$  in [Line 4](#).

Compared to the provable guarantee from classical perturbation [[43](#), Lemma 22], speaking only in terms of  $n$ , classically it takes  $\mathcal{T} = O(\log n)$  steps to decrease the function value by  $\mathcal{F} = \Omega(1/\log^3 n)$ , whereas our quantum simulation with time  $\mathcal{T}' = O(\log n)$  together with also  $\mathcal{T}'$  subsequent GD iterations decrease the function value by  $\mathcal{F}' = \Omega(1)$  with high success probability.

To prove [Proposition 2.1](#), we first prove the following lemma:

**Lemma 2.3.** *Under the setting of [Proposition 2.1](#), we have:*

$$\mathbb{P}\left(\Delta f_{\perp} \geq \frac{\mathcal{F}'}{2}\right) \leq \exp\left(-\frac{\ell^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\right), \quad (2.14)$$

where  $\Delta f_{\perp}$  stands for the function value increase in the eigen-directions other than the most negative one due to the perturbation from the quantum simulation for time  $\mathcal{T}'$ . Specifically,

$$\Delta f_{\perp} = f(\mathbf{x}_0 - \Delta \mathbf{x}_{\parallel}) - f(\tilde{\mathbf{x}}), \quad (2.15)$$

where  $\Delta \mathbf{x}_{\parallel}$  stands for the component of  $\mathbf{x}_0 - \tilde{\mathbf{x}}$  along the most negative eigen-direction.

*Proof.* For every direction  $x_i$  with positive eigenvalue  $\lambda_i$ , from [Proposition A.1](#), the probability wave packet on this direction can be written as:

$$p_i(t, x_i) = \Phi_i(t, x_i)^2 = \left(\frac{1}{2\pi}\right)^{1/2} \cdot \frac{1}{r_0\sigma(t; \lambda_i)} \exp\left(-\frac{x_i^2}{2r_0^2\sigma^2(t; \lambda_i)}\right), \quad (2.16)$$

where

$$\sigma^2(t; \lambda_i) = \frac{(1 + 4\alpha^2) - (1 - 4\alpha^2) \cos 2\alpha t}{8\alpha^2}. \quad (2.17)$$

Since we have set  $t = t_e = \mathcal{T}'$ ,

$$\sigma^2(\mathcal{T}'; \lambda_i) = \frac{(1 + 4\lambda_i) - (1 - 4\lambda_i) \cos 2\sqrt{\lambda_i}\mathcal{T}'}{8\lambda_i} \leq \max\left(1, \frac{1}{4\lambda_i}\right) \quad (2.18)$$

by [Lemma A.2](#). We define a new variable  $y_i = \sqrt{\lambda_i}x_i$ . We consider all the positive eigen-directions, the function value increase can be expressed as:

$$\Delta f_{\perp} \leq \sum_i y_i^2, \quad (2.19)$$

which means summing over all eigen-directions  $x_i$  with positive eigenvalue  $\lambda_i$ . As we can see,  $y_i$  follows the Gaussian distribution  $\mathcal{N}(0, \lambda_i r_0^2 \sigma^2(\mathcal{T}'; \lambda_i))$  where  $\lambda_i$  can be upper bounded by  $\ell$  due to the  $\ell$ -smooth condition of  $f$ . Hence, the variance cannot be greater than  $(1 + \ell^2)r_0^2$ . We let  $\delta' = \frac{\epsilon}{3}\sqrt{1 + \ell^2}$  and consider the following  $n$ -dimensional Gaussian distribution:

$$p'(\mathbf{y}') = \left(\frac{1}{2\pi}\right)^{n/2} \left(\frac{\sqrt{n}}{\delta'}\right)^n \exp(-n\mathbf{y}'^2/2\delta'^2). \quad (2.20)$$

Then  $\mathbf{y}'$  will always has a larger dispersion than for  $\mathbf{y}$ . To be more specific, we have the following relation:

$$\mathbb{P}\left(\Delta f_{\perp} \leq \frac{\mathcal{F}'}{2}\right) \leq \mathbb{P}\left(\|\mathbf{y}\|^2 \leq \frac{\mathcal{F}'}{2}\right) \leq \mathbb{P}\left(\|\mathbf{y}'\|^2 \leq \frac{\mathcal{F}'}{2}\right), \quad (2.21)$$

in which  $\mathbf{y} \sim p'(\mathbf{y})$ . We can use [Lemma A.4](#) to handle this probability. To be specific, we choose  $A$  therein to be an  $n \times n$  diagonal matrix with each nonzero element being  $1/\sqrt{n}$ , and let  $t = \frac{\mathcal{F}'}{2\delta'^2}$ . From our choice of parameters, we then know that:

$$\mathbb{P}\left[\frac{\|\mathbf{y}\|^2}{\delta'^2} \geq 1 + 2\sqrt{\frac{\mathcal{F}'}{2\delta'^2 n}} + \frac{\mathcal{F}'}{2\delta'^2 n}\right] \leq \exp\left(-\frac{\mathcal{F}'}{2\delta'^2}\right). \quad (2.22)$$

from our choice of parameters, we know that:

$$\mathcal{F}' = \sqrt{\frac{\epsilon}{\rho^3}} \quad \delta' = \frac{\epsilon}{\iota^3} \sqrt{1 + \ell^2}, \quad (2.23)$$

which leads to

$$\frac{\mathcal{F}'}{2\delta'^2} = \sqrt{\frac{\epsilon}{\rho^3}} \cdot \frac{\iota^6}{\epsilon^2(1 + \ell^2)} = \frac{\iota^6}{(\rho\epsilon)^{3/2}(1 + \ell^2)}, \quad (2.24)$$

under our choice of parameters. When  $n \geq 2$ , we have:

$$\frac{\mathcal{F}'}{2\delta'^2} \geq 1 + 2\sqrt{\frac{\mathcal{F}'}{2\delta'^2 n}} + \frac{\mathcal{F}'}{2\delta'^2 n}. \quad (2.25)$$

If  $n = 1$ , there will be no positive eigen-direction of a saddle point, thus we do not have to consider it at this part. In all, we can deduce that

$$\mathbb{P}\left[\frac{\|\mathbf{y}\|^2}{\delta'^2} \geq \frac{\mathcal{F}'}{2\delta'^2}\right] \leq \mathbb{P}\left[\|\mathbf{y}\|^2 \geq 1 + 2\sqrt{\frac{\mathcal{F}'}{2\delta'^2 n}} + \frac{\mathcal{F}'}{2\delta'^2 n}\right] \leq \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1 + \ell^2)}\right), \quad (2.26)$$

which leads to

$$\mathbb{P}\left(\Delta f_{\perp} \geq \frac{\mathcal{F}'}{2}\right) \leq \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1 + \ell^2)}\right). \quad (2.27)$$

□

Now we are ready to prove [Proposition 2.1](#).

*Proof.* Following the discussion in [Section 2](#), without loss of generality we assume that  $f$  is quadratic. We can then view the gradient descent process as the combination of  $n$  gradient descent processes, each carried out in one single direction. Without loss of generality, we assume that  $\tilde{\mathbf{x}} = 0$ , otherwise we shift  $\tilde{\mathbf{x}}$  to the original point.

We first consider the gradient descent in the  $\mathbf{e}_1$  direction, where  $\mathbf{e}_1$  stands for the eigenvector of  $\lambda_{\min}$ , or equivalently, the direction with the most negative eigenvalue. We use  $x_{\parallel}$  to denote the corresponding coordinate and use  $\Phi_m$  to denote the wave function component. By [Proposition A.1](#),  $\Phi_m$  satisfies:

$$\Phi_m(\mathcal{T}', x_{\parallel}) = \left(\frac{1}{2\pi}\right)^{1/4} \cdot \frac{1}{\sqrt{r_0\sigma(\mathcal{T}'; \lambda_{\min})}} \exp\left(-\frac{x_{\parallel}^2}{2r_0^2\sigma^2(\mathcal{T}'; \lambda_{\min})}\right). \quad (2.28)$$

Therefore, the probability distribution of the wave packet is:

$$p_m(\mathcal{T}', x_{\parallel}) = \Phi_m(\mathcal{T}', x_{\parallel})^2 = \frac{1}{\sqrt{2\pi} \cdot r_0\sigma(\mathcal{T}'; \lambda_{\min})} \exp\left(-\frac{x_{\parallel}^2}{r_0^2\sigma^2(\mathcal{T}'; \lambda_{\min})}\right). \quad (2.29)$$

After  $x'_0$  is chosen, as long as the gradient descent sequence is not far from the saddle point, the landscape near the current position can still be viewed as approximately quadratic, which indicates that its component  $x'$  in the direction with the most negative eigenvalue satisfies:

$$x_{t+1\parallel} = x_{t\parallel} + \eta(-\lambda_{\min})x_{t\parallel} = (1 - \eta\lambda_{\min})x_{t\parallel} \quad (2.30)$$

which results in

$$x_{t\parallel} = (1 - \eta\lambda_{\min})^t x_{0\parallel}. \quad (2.31)$$

Therefore, we can deduce that:

$$\mathbb{P}\left(\exists t \in [1, \mathcal{T}'] \mid -\frac{1}{2}\lambda_{\min}x_{t\parallel}^2 \geq \mathcal{F}'\right) = \mathbb{P}\left(\exists t \in [1, \mathcal{T}'] \mid x_{t\parallel} \geq \sqrt{-\frac{2\mathcal{F}'}{\lambda_{\min}}}\right) \quad (2.32)$$

$$= \mathbb{P}\left(\exists t \in [1, \mathcal{T}'] \mid x_{0\parallel} \geq \frac{1}{(1 - \eta\lambda_{\min})^t} \sqrt{-\frac{2\mathcal{F}'}{\lambda_{\min}}}\right) \quad (2.33)$$

$$\geq \mathbb{P}\left(x_{0\parallel} \geq \frac{1}{(1 - \eta\lambda_{\min})^{\mathcal{T}'}} \sqrt{-\frac{2\mathcal{F}'}{\lambda_{\min}}}\right). \quad (2.34)$$

Let  $\alpha := \frac{1}{(1 - \eta\lambda_{\min})^{\mathcal{T}'}} \sqrt{-\frac{2\mathcal{F}'}{\lambda_{\min}}}$ . Since  $\lambda_{\min} \leq -\sqrt{\rho\epsilon}$ ,  $\alpha'_0$  is the upper bound for all possible  $\alpha$ . Thus we have:

$$\mathbb{P}\left(x_{0\parallel} \leq \frac{1}{(1 - \eta\lambda_{\min})^{\mathcal{T}'}} \sqrt{-\frac{2\mathcal{F}'}{\lambda_{\min}}}\right) = \int_{-\alpha}^{\alpha} p_m(t, x_{\parallel}) dx_{\parallel} \quad (2.35)$$

$$\leq \int_{-\alpha_0}^{\alpha_0} \frac{1}{\sqrt{2\pi} \cdot r_0 \sigma(\mathcal{T}'; \lambda_{\min})} \exp\left(-\frac{x_{\parallel}^2}{r_0^2 \sigma^2(\mathcal{T}'; \lambda_{\min})}\right) dx_{\parallel} \quad (2.36)$$

$$\leq \int_{-\alpha_0}^{\alpha_0} \frac{1}{\sqrt{2\pi} \cdot r_0 \sigma(\mathcal{T}'; -\sqrt{\rho\epsilon})} \exp\left(-\frac{x_{\parallel}^2}{r_0^2 \sigma^2(\mathcal{T}'; -\sqrt{\rho\epsilon})}\right) dx_{\parallel}. \quad (2.37)$$

Note that

$$\int_{-\alpha_0}^{\alpha_0} \frac{1}{\sqrt{2\pi} \cdot r_0 \sigma(\mathcal{T}'; -\sqrt{\rho\epsilon})} \exp\left(-\frac{x_{\parallel}^2}{r_0^2 \sigma^2(\mathcal{T}'; -\sqrt{\rho\epsilon})}\right) dx_{\parallel} \leq \frac{2\alpha'_0}{\sqrt{2\pi} \cdot r_0 \sigma(\mathcal{T}'; -\sqrt{\rho\epsilon})}. \quad (2.38)$$

By [Lemma A.3](#) in [Appendix A.2](#), we have:

$$\frac{2\alpha'_0}{\sqrt{2\pi} \cdot r_0 \sigma(\mathcal{T}'; -\sqrt{\rho\epsilon})} \leq \frac{2\alpha'_0}{r_0 \sqrt{2\pi} (1 + \mathcal{T}'^2/4)} \leq \frac{2\sqrt{2}\alpha'_0}{r_0 \mathcal{T}' \sqrt{\pi}}. \quad (2.39)$$

Note that [Lemma A.3](#) contains strong physics intuition:  $1 + \mathcal{T}'^2/4$  is the dispersion (the variance of the probability distribution function) of a free quantum particle, and with a negative potential field this dispersion can only be larger. Further we can deduce that:

$$\frac{2\sqrt{2}\alpha'_0}{r_0 \mathcal{T}' \sqrt{\pi}} = \frac{2\sqrt{2}}{\sqrt{\pi}} \cdot \frac{\sqrt{2}}{\rho(1 + \eta\sqrt{\rho\epsilon})^{\mathcal{T}'}} \cdot \frac{\iota^3 \sqrt{n}}{\epsilon} \cdot \frac{\sqrt{\rho\epsilon}}{\iota \ell} = \frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot (1 + \eta\sqrt{\rho\epsilon})^{-1/\eta\sqrt{\rho\epsilon}}. \quad (2.40)$$

Denote  $\mu := \eta\sqrt{\rho\epsilon}$ . Note that the function  $h(\mu) := \ln((1 + \mu)^{1/\mu}) = \ln(1 + \mu)/\mu$  satisfies

$$h'(\mu) = \frac{\frac{\mu}{1+\mu} - \ln(1 + \mu)}{\mu^2} \leq 0 \quad \forall \mu \in (0, 1]; \quad (2.41)$$

therefore,  $h$  is a decreasing function on  $(0, 1]$ , which implies that  $(1 + \mu)^{1/\mu} \geq (1 + 1)^1 = 2$  for any  $\mu \in (0, 1]$ . As a result, as long as  $\eta\sqrt{\rho\epsilon} \leq 1$ , or equivalently  $\epsilon \leq l^2/\rho$ , we have

$$(1 + \eta\sqrt{\rho\epsilon})^{1/(\eta\sqrt{\rho\epsilon})} \geq 2, \quad (2.42)$$

which then implies

$$\mathbb{P}\left(x_{0\parallel} \leq \frac{1}{(1 - \eta\lambda_{\min})^{\mathcal{T}'}} \sqrt{-\frac{2\mathcal{F}'}{\lambda_{\min}}}\right) \leq \frac{2\sqrt{2}\alpha'_0}{r_0\mathcal{T}'\sqrt{\pi}} \leq \frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot 2^{-\iota}. \quad (2.43)$$

Equipped with this formula, we are ready to prove (2.13). First note that:

$$\mathbb{P}\left(f(\mathbf{x}_{\mathcal{T}}) - f(\tilde{\mathbf{x}}) \leq -\frac{\mathcal{F}'}{2}\right) = \mathbb{P}\left(\exists t \in [1, \mathcal{T}'] \mid f(\mathbf{x}_t) - f(\tilde{\mathbf{x}}) \leq -\frac{\mathcal{F}'}{2}\right), \quad (2.44)$$

which is due to [Lemma A.5](#) that the function value decreases monotonically in the gradient descent sequence. Since  $\mathcal{F}'$  is small when  $\epsilon$  is small, among all possible value of  $t$  satisfying the condition above, we must can find one  $t_0$ , for which  $\mathbf{x}_{t_0}$  is near the saddle point. Due to the  $\rho$ -Hessian Lipschitz condition, the function value near  $\mathbf{x}_{t_0}$  can also be approximately viewed as a product form of quadratic functions of each eigen-direction, which then leads to:

$$\mathbb{P}\left(f(\mathbf{x}_{t_0}) - f(\tilde{\mathbf{x}}) \leq -\frac{\mathcal{F}'}{2}\right) \geq \mathbb{P}\left(\Delta f_{\perp} \geq \frac{\mathcal{F}'}{2}\right) \cdot \mathbb{P}\left(-\frac{1}{2}\lambda_{\min}x_{t_0\parallel}^2 \geq \mathcal{F}'\right), \quad (2.45)$$

where  $\Delta f_{\perp}$  stands for the function value increase in the eigen-directions other than  $x_{\parallel}$  due to perturbation, specific definition referring to [Lemma 2.3](#), by which we have:

$$\mathbb{P}\left(\Delta f_{\perp} \leq \frac{\mathcal{F}'}{2}\right) \geq 1 - \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1 + \ell^2)}\right), \quad (2.46)$$

and hence we further have

$$\begin{aligned} & \mathbb{P}\left(f(\mathbf{x}_{\mathcal{T}}) - f(\tilde{\mathbf{x}}) \leq -\frac{\mathcal{F}'}{2}\right) \\ &= \mathbb{P}\left(\exists t \in [1, \mathcal{T}'] \mid f(\mathbf{x}_t) - f(\tilde{\mathbf{x}}) \leq -\frac{\mathcal{F}'}{2}\right) \end{aligned} \quad (2.47)$$

$$\geq \mathbb{P}\left(\Delta f_{\perp} \leq \frac{\mathcal{F}'}{2}\right) \cdot \mathbb{P}\left(\exists t \in [1, \mathcal{T}'] \mid -\frac{1}{2}\lambda_{\min}x_{t\parallel}^2 \geq \mathcal{F}'\right). \quad (2.48)$$

$$\geq \left(1 - \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1 + \ell^2)}\right)\right) \cdot \left(1 - \mathbb{P}\left(x_{0\parallel} \leq \frac{1}{(1 - \eta\lambda_{\min})^{\mathcal{T}'}} \sqrt{-\frac{2\mathcal{F}'}{\lambda_{\min}}}\right)\right) \quad (2.49)$$

$$\geq 1 - \frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot 2^{-\iota} - \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1 + \ell^2)}\right). \quad (2.50)$$

□

Let us compare the quantum error bound

$$\frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot 2^{-\iota} + \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\right) \quad (2.51)$$

to the classical one

$$\frac{\ell\sqrt{n}}{\sqrt{\rho\epsilon}} \cdot \iota^2 \cdot 2^{8-\iota}. \quad (2.52)$$

As we can see, the error bound of our quantum wave packet simulation is of the same order as the classical uniform perturbation in a ball, while the total query/time complexity is significantly reduced with respect to  $\log n$ .

### 2.2.2 Proof of our quantum speedup

We now prove the following theorem using [Proposition 2.1](#):

**Theorem 2.2.** *For any  $\epsilon, \delta > 0$ , [Algorithm 2](#) with parameters chosen in [Proposition 2.1](#) satisfies that at least one half of its iterations of will be  $\epsilon$ -approximate local minima, using*

$$\tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \cdot \iota^2\right) \quad (2.53)$$

queries to  $U_f$  in [\(1.2\)](#) and gradients with probability  $\geq 1 - \delta$ , where  $f^*$  is the global minimum of  $f$ .

*Proof.* Let the parameters be chosen according to [\(2.11\)](#) and [\(2.12\)](#), and set the total iteration steps  $T$  to be:

$$T = 8 \max \left\{ \frac{(f(\mathbf{x}_0) - f^*) \mathcal{T}'}{\mathcal{F}'}, \frac{2(f(\mathbf{x}_0) - f^*)}{\eta\epsilon^2} \right\} = \tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \cdot \iota\right), \quad (2.54)$$

similar to the classical GD algorithm. Since we perturb at most once in  $\mathcal{T}'$  consecutive steps, we only need to apply the quantum simulation for at most  $T/\mathcal{T}'$  times. Then by [Proposition 2.1](#), with probability at least

$$1 - \frac{T}{\mathcal{T}'} \left[ \frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot 2^{-\iota} + \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\right) \right], \quad (2.55)$$

one can find that after each time a perturbation was added, the function value decreases at least  $\mathcal{F}'$  in the following  $\mathcal{T}'$  steps. We choose the constant  $k$  in the formula of  $\iota$  large enough to satisfy:

$$\frac{T}{\mathcal{T}'} \left[ \frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot 2^{-\iota} + \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\right) \right] \leq \delta; \quad (2.56)$$

then with probability at least  $1 - \delta$ , after each time a perturbation was added, the function value decreases at least  $\mathcal{F}'/2$  in the following  $\mathcal{T}'$  steps. As a consequence, there can at most be  $\frac{2(f(\mathbf{x}_0) - f^*)}{\mathcal{F}'}$  perturbations.

Excluding those iterations that are within  $\mathcal{T}'$  steps after adding perturbations, we still have  $3T/4$  steps left. They are either large gradient steps,  $\|\nabla f(\mathbf{x}_t)\| \geq \epsilon$ , or  $\epsilon$ -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 A.5](#) in [Appendix A.3](#):

$$f(\mathbf{x}_T) \leq f(\mathbf{x}_0) - T\eta\epsilon^2/4 < f^*, \quad (2.57)$$



a contradiction. Therefore, we conclude that at least  $T/2$  of the iterations must be  $\epsilon$ -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  $1 - \delta$ ,

$$T_1 = T = \tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \cdot \iota\right). \quad (2.58)$$

As for  $T_2$ , with probability at least  $1 - \delta$  quantum simulation is called for at most  $\frac{2(f(\mathbf{x}_0) - f^*)}{\mathcal{T}'}$  times, and by [Lemma 2.2](#) it takes  $\tilde{O}(\mathcal{T}' \log n \log^2(\mathcal{T}'^2/\epsilon))$  queries to carry out each simulation. Therefore,

$$T_2 = \frac{2(f(\mathbf{x}_0) - f^*)}{\mathcal{T}'} \cdot \tilde{O}(\mathcal{T}' \log n \log^2(\mathcal{T}'^2/\epsilon)) = \tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon} \cdot \iota^2\right). \quad (2.59)$$

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

$$\tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \cdot \iota^2\right). \quad (2.60)$$

□

### 2.3 Perturbed accelerated gradient descent with quantum simulation

In [Theorem 2.2](#), the  $1/\epsilon^2$  term is a bottleneck of the whole algorithm, but Jin et al. [\[44\]](#) improved it to  $1/\epsilon^{1.75}$  by replacing the GD with the accelerated GD by Nesterov [\[57\]](#). We next introduce a hybrid quantum-classical algorithm ([Algorithm 3](#)) that reflects this intuition. We make the following comparisons to [\[44\]](#):

- **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. After that, we apply GD for  $\mathcal{T}'$  steps and then return to AGD with  $\mathbf{v}$  being reset to 0. Speaking only in terms of  $n$ , Jin et al. [\[44\]](#) 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  $\mathcal{T}' = O(\log n)$  and also  $\mathcal{T}'$  subsequent GD iterations decrease the Hamiltonian by  $\Omega(1)$  with high probability.

The following theorem provides the complexity of this algorithm:

**Theorem 2.3.** *Suppose that the function  $f$  is  $\ell$ -smooth and  $\rho$ -Hessian Lipschitz. We choose the parameters appearing in [Algorithm 3](#) as follows:*

$$\iota := k \cdot \log(n\ell(f(\mathbf{x}_0) - f^*)/(\rho\epsilon\delta)) \quad \eta := \frac{1}{\ell} \quad r_0 := \frac{\epsilon}{\iota^3} \cdot \frac{1}{\sqrt{n}} \quad (2.61)$$

$$\mathcal{T}' := \frac{\ell}{\sqrt{\rho\epsilon}} \cdot \iota \cdot c_A \quad \mathcal{T}' := \sqrt{\frac{\epsilon}{\rho^3}} \quad \chi := 1 \quad (2.62)$$

$$\eta' := \frac{1}{4\ell} \quad \kappa := \frac{\ell}{\sqrt{\rho\epsilon}} \quad \theta := \frac{1}{4\sqrt{\kappa}} \quad (2.63)$$

$$\gamma := \frac{\theta^2}{\eta} \quad s := \frac{\gamma}{4\rho} \quad \mathcal{T} := \sqrt{\kappa} \cdot c_A \quad (2.64)$$

---

**Algorithm 3:** Perturbed Accelerated Gradient Descent with Quantum Simulation.

---

```

1  $\mathbf{v}_0 \leftarrow 0$ ;
2 for  $t = 0, 1, \dots, T$  do
3   if  $\|\nabla f(\mathbf{x}_t)\| \leq \epsilon$  and no perturbation in last  $\mathcal{T}'$  steps then
4      $\mathbf{x}_t \leftarrow \mathbf{x}_t + \xi_t$      $\xi_t \sim \text{QuantumSimulation}(\mathbf{x}_t, r_0, \mathcal{T}')$ ;
5   if a perturbation was added in last  $\mathcal{T}'$  steps then
6      $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)$  and  $\mathbf{v}_{t+1} = 0$ ;
7   else
8      $\mathbf{y}_t \leftarrow \mathbf{x}_t + (1 - \theta)\mathbf{v}_t$ ,  $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t - \eta' f(\mathbf{y}_t)$ , and  $\mathbf{v}_{t+1} \leftarrow \mathbf{x}_{t+1} - \mathbf{x}_t$ ;
9     if  $f(\mathbf{x}_t) \leq f(\mathbf{y}_t) + \langle \nabla f(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t \rangle - \frac{\gamma}{2} \|\mathbf{x}_t - \mathbf{y}_t\|$  then
10       $(\mathbf{x}_{t+1}, \mathbf{v}_{t+1}) \leftarrow \text{Negative-Curvature-Exploitation}(\mathbf{x}_t, \mathbf{v}_t, s)$ ;

```

---

in which  $c_A$  is chosen large enough to satisfy the condition in [Lemma A.6](#). Then there exists a absolute constant  $k_{\min}$  such that for any  $\delta > 0$ ,  $\epsilon \leq \frac{\ell^2}{\rho}$ ,  $k \geq k_{\min}$  and such that 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  $\epsilon$ -approximate second-order stationary point, using the following number of queries to  $U_f$  in (1.2) and classical gradients:

$$\tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^{1.75}} \cdot \iota^2\right). \quad (2.65)$$

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

$$T = 3 \max \left\{ \frac{2(f(\mathbf{x}_0) - f^*)\mathcal{T}'}{\mathcal{T}'}, \frac{(f(\mathbf{x}_0) - f^*)\mathcal{T}}{\mathcal{E}} \right\}, \quad (2.66)$$

in which  $\mathcal{E} = \sqrt{\frac{\epsilon^3}{\rho}} \cdot c_A^{-7}$ , the same as our choice for  $\mathcal{E}$  in [Lemma A.6](#). We assume the contrary, i.e., the outputs of all of the iterations are not  $\epsilon$ -approximate second-order stationary points. Similar to our analysis in the proof of [Theorem 2.2](#), since we perturb at most once in  $\mathcal{T}'$  consecutive steps, we only need to apply the quantum simulation for at most  $T/\mathcal{T}'$  times. Then by [Proposition 2.1](#), with probability at least

$$1 - \frac{T}{\mathcal{T}'} \left[ \frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot 2^{-\iota} + \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\right) \right], \quad (2.67)$$

one can find that after each time a perturbation was added, the function value decreases at least  $\mathcal{T}'/2$  in the following  $\mathcal{T}'$  steps. We choose the constant  $k$  in the formula of  $\iota$  large enough to satisfy:

$$\frac{T}{\mathcal{T}'} \left[ \frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot 2^{-\iota} + \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\right) \right] \leq \delta; \quad (2.68)$$

then with probability at least  $1 - \delta$ , after each time a perturbation was added, the function value decreases at least  $\mathcal{T}'/2$  in the following  $\mathcal{T}'$  steps. Then, there are at most  $T/3$  steps of gradient descent used to escape from saddle points, otherwise the function value decrease will be greater than  $f(\mathbf{x}_0) - f^*$ , which is impossible. As a result, the rest  $2T/3$  steps are all accelerated gradient descent steps.

Since from  $\epsilon \leq \ell^2/\rho$  we have  $\mathcal{T}' \geq \mathcal{T}$ , then we can find at least  $\frac{T}{3\mathcal{T}}$  disjoint time periods, each of time interval  $\mathcal{T}$ . From [Lemma A.6](#), during these time intervals the Hamiltonian will decrease in total at least:

$$\frac{T}{3\mathcal{T}} \times \mathcal{E} = f(\mathbf{x}_0) - f^*, \quad (2.69)$$

which is impossible due to [Lemma A.7](#), 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:

$$T = 3 \max \left\{ \frac{2(f(\mathbf{x}_0) - f^*)\mathcal{T}'}{\mathcal{T}'}, \frac{(f(\mathbf{x}_0) - f^*)\mathcal{T}}{\mathcal{E}} \right\} = \tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^{1.75}} \cdot \iota\right). \quad (2.70)$$

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  $1 - \delta$ ,

$$T_1 = T = \tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^{1.75}} \cdot \iota\right). \quad (2.71)$$

For  $T_2$ , with probability at least  $1 - \delta$  quantum simulation is called for at most  $\frac{2(f(\mathbf{x}_0) - f^*)}{\mathcal{T}'}$  times, and by [Lemma 2.2](#) it takes  $\tilde{O}(\mathcal{T}' \log n \log^2(\mathcal{T}'^2/\epsilon))$  queries to carry out each simulation. Therefore,

$$T_2 = \frac{2(f(\mathbf{x}_0) - f^*)}{\mathcal{T}'} \cdot \tilde{O}(\mathcal{T}' \log n \log^2(\mathcal{T}'^2/\epsilon)) = \tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon} \cdot \iota^2\right). \quad (2.72)$$

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

$$\tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^{1.75}} \cdot \iota^2\right). \quad (2.73)$$

□

**Remark 2.2.** Although the theorem above only guarantees that one of the iterations is an  $\epsilon$ -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  $\mathcal{T}'$  steps after. If the function value decreases by at least  $\mathcal{T}'/2$ , then the algorithm has made progress, otherwise with probability at least  $1 - \delta$ ,  $\tilde{\mathbf{x}}$  is an  $\epsilon$ -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 has been shown in the case of convex optimization [[6](#), [16](#)], and we generalize the same phenomenon to nonconvex optimization.

The idea was initiated by Jordan [[46](#)]. 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<sup>3</sup>, and apply the quantum Fourier transform (QFT). Due to Taylor expansion,

$$\sum_{\mathbf{x}} e^{if(\mathbf{x})} \mathbf{x} \approx \sum_{\mathbf{x}} \bigotimes_{k=1}^n e^{i \frac{\partial f}{\partial x_k} \mathbf{x}_k} \mathbf{x}_k, \quad (3.1)$$

the QFT can recover all the partial derivatives simultaneously. In this paper, we refer to [16, Lemma 2.2] for a precise version of Jordan's algorithm:

**Lemma 3.1.** *Let  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  be an  $\ell$ -smooth function specified by the evaluation oracle in (1.2) with accuracy  $\delta_q$ , i.e., it returns a value  $\tilde{f}(x)$  such that  $|\tilde{f}(x) - f(x)| \leq \delta_q$ . For any  $\mathbf{x} \in \mathbb{R}^n$ , there is a quantum algorithm that uses one query to (1.2) and returns a vector  $\tilde{\nabla} f(\mathbf{x})$  s.t.*

$$\mathbb{P} \left[ \|\tilde{\nabla} f(\mathbf{x}) - \nabla f(\mathbf{x})\|_2 > 400\omega n \sqrt{\delta_q \ell} \right] < \min \left\{ \frac{n}{\omega - 1}, 1 \right\} \quad \forall \omega > 1. \quad (3.2)$$

The main technical contribution of this section is to replace the gradient descent steps in Section 2 by Lemma 3.1. 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 3.1:

**Lemma 3.2.** *Let  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  be an  $\ell$ -smooth,  $\rho$ -Hessian Lipschitz function, and let  $\eta \leq 1/\ell$ . Then the gradient outputted by Lemma 3.1 satisfies that for any fixed constant  $c$ , with probability 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:*

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq -\eta \|\nabla f(\mathbf{x}_t)\|^2 / 2 + c, \quad (3.3)$$

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  $\delta_q$ .

*Proof.* Considering our condition of  $f$  being  $\ell$ -smooth, we have:

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t) \cdot (\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{\ell}{2} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2. \quad (3.4)$$

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:

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t) \cdot (\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{\ell}{2} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 \quad (3.5)$$

$$\leq f(\mathbf{x}_t) - \eta \nabla f(\mathbf{x}_t) \cdot (\nabla f(\mathbf{x}_t) + \delta[\mathbf{g}(\mathbf{x}_t)]) + \frac{\eta}{2} \|\nabla f(\mathbf{x}_t) + \delta[\mathbf{g}(\mathbf{x}_t)]\|^2 \quad (3.6)$$

$$= f(\mathbf{x}_t) - \frac{\eta}{2} \|\nabla f(\mathbf{x}_t)\|^2 + \frac{\eta}{2} \|\delta[\mathbf{g}(\mathbf{x}_t)]\|^2. \quad (3.7)$$

By Lemma 3.1, 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.  $\square$

---

<sup>3</sup>This can be achieved by a standard technique called phase kickback. See more details at [35] and [16].

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 [\[43\]](#).

---

**Algorithm 4:** Perturbed GD with Quantum Simulation and Gradient Computation.

---

```

1  $t_{\text{perturb}} = 0;$ 
2 for  $t = 0, 1, \dots, T$  do
3   Apply Lemma 3.1 to compute an estimate  $\tilde{\nabla}f(\mathbf{x})$  of  $\nabla f(\mathbf{x})$ ;
4   if  $\|\tilde{\nabla}f(\mathbf{x}_t)\| \leq \epsilon$  and no perturbation in last  $\mathcal{T}'$  steps then
5      $\mathbf{x}_t \leftarrow \mathbf{x}_t + \xi_t \quad \xi_t \sim \text{QuantumSimulation}(\mathbf{x}_t, r_0, \mathcal{T}')$ ;
6      $t_{\text{perturb}} \leftarrow t;$ 
7    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta \tilde{\nabla}f(\mathbf{x}_t);$ 

```

---

**Lemma 3.3.** *Under the setting of [Lemma 3.2](#), 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:*

$$\|\mathbf{x}_\tau - \mathbf{x}_0\| \leq 2\sqrt{\eta t |f(\mathbf{x}_0) - f(\mathbf{x}_t)|} + 2\eta t \sqrt{c}, \quad (3.8)$$

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

*Proof.* Observe that

$$\|\mathbf{x}_\tau - \mathbf{x}_0\| \leq \sum_{\tau=1}^t \|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\|. \quad (3.9)$$

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

$$\|\mathbf{x}_\tau - \mathbf{x}_0\| \leq \sum_{\tau=1}^t \|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\| \leq \left[ t \sum_{\tau=1}^t \|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\|^2 \right]^{\frac{1}{2}}, \quad (3.10)$$

in which

$$\mathbf{x}_\tau - \mathbf{x}_{\tau-1} = \eta \mathbf{g}(\mathbf{x}_{\tau-1}) = \eta \nabla f(\mathbf{x}_{\tau-1}) + \eta \delta[\mathbf{g}(\mathbf{x}_{\tau-1})], \quad (3.11)$$

which results in

$$\|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\|^2 \leq \eta^2 \|\nabla f(\mathbf{x}_{\tau-1})\|^2 + 2\eta^2 \nabla f(\mathbf{x}_{\tau-1}) \cdot \delta[\mathbf{g}(\mathbf{x}_{\tau-1})] + \eta^2 \|\delta[\mathbf{g}(\mathbf{x}_{\tau-1})]\|^2 \quad (3.12)$$

$$\leq 2\eta^2 \|\nabla f(\mathbf{x}_{\tau-1})\|^2 + 2\eta^2 \|\delta[\mathbf{g}(\mathbf{x}_{\tau-1})]\|^2. \quad (3.13)$$

Go back to the first inequality,

$$\|\mathbf{x}_\tau - \mathbf{x}_0\| \leq \left[ t \sum_{\tau=1}^t \|\mathbf{x}_\tau - \mathbf{x}_{\tau-1}\|^2 \right]^{\frac{1}{2}} \leq \left[ 2\eta^2 t \sum_{\tau=1}^t (\|\nabla f(\mathbf{x}_{\tau-1})\|^2 + \|\delta[\mathbf{g}(\mathbf{x}_{\tau-1})]\|^2) \right]^{\frac{1}{2}}. \quad (3.14)$$

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 3.1](#), this condition can be satisfied with probability at least  $1 - \frac{nt}{\frac{1}{A_q}\sqrt{\frac{2c}{\eta}} - 1}$ . Then,

$$\|\mathbf{x}_\tau - \mathbf{x}_0\| \leq \left[ 2\eta^2 t \sum_{\tau=1}^t \left( \|\nabla f(\mathbf{x}_{\tau-1})\|^2 + \|\delta[\mathbf{g}(\mathbf{x}_{\tau-1})]\|^2 \right) \right]^{\frac{1}{2}} \quad (3.15)$$

$$\leq \left[ 2\eta^2 t \left( \frac{2f(\mathbf{x}_0) - 2f(\mathbf{x}_t)}{\eta} + 2t\|\delta[\mathbf{g}(\mathbf{x}_{\tau-1})]\|^2 \right) \right]^{\frac{1}{2}} \quad (3.16)$$

$$\leq [4\eta t(f(\mathbf{x}_0) - f(\mathbf{x}_t) + \eta tc)]^{\frac{1}{2}} \quad (3.17)$$

$$\leq 2\sqrt{\eta t|f(\mathbf{x}_0) - f(\mathbf{x}_t)|} + 2\eta t\sqrt{c}. \quad (3.18)$$

□

In [Algorithm 4](#), a perturbation is added in order to circumvent the case that the algorithm gets stuck at a saddle point. If the gradient can be accessed accurately with no error, this method is effective due to [\[43\]](#). Here we show its robustness such that the part of [Algorithm 4](#) aimed at escaping from saddle points still works given a gradient oracle of  $f$  with certain error. Furthermore, this implies that given a quantum evaluation oracle, we can give a quantum algorithm for escaping from saddle points by [Lemma 3.1](#). This claim is rigorously presented as the following lemma:

**Lemma 3.4.** *We first specify our choices for some parameters and constants that appear in [Algorithm 4](#) or are frequently used:*

$$\iota := k \cdot \log(n\ell(f(\mathbf{x}_0) - f^*)/(\rho\epsilon\delta)) \quad \eta := \frac{1}{\ell} \quad r_0 := \frac{\epsilon}{\iota^3} \cdot \frac{1}{\sqrt{n}} \quad (3.19)$$

$$\mathcal{T}' := \frac{\ell}{\sqrt{\rho\epsilon}} \cdot \iota \quad \mathcal{F}' := \sqrt{\frac{\epsilon}{\rho^3}} \quad \alpha_0'' := \frac{\sqrt{2}}{(1 + \eta\sqrt{\rho\epsilon}/2)^{\mathcal{T}'}} \cdot \frac{1}{\rho} \quad (3.20)$$

where  $k$  is an absolute constant. Let  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  be an  $\ell$ -smooth,  $\rho$ -Hessian Lipschitz function. For a saddle point  $\tilde{\mathbf{x}}$  satisfying  $\|\nabla f(\tilde{\mathbf{x}})\| = 0$  and  $\lambda_{\min}(\nabla^2 f(\tilde{\mathbf{x}})) \leq -\sqrt{\rho\epsilon}$ , if quantum simulation was called at  $t = 0$ , [Algorithm 4](#) satisfies:

$$\mathbb{P}\left(f(\mathbf{x}_{\mathcal{T}'}) - f(\tilde{\mathbf{x}}) \leq -\frac{\mathcal{F}'}{4}\right) \geq 1 - p_{\text{error}}, \quad (3.21)$$

where

$$p_{\text{error}} = \frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot 2^{-\iota/2} + \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\right) \quad (3.22)$$

$$+ \frac{\mathcal{T}'}{\frac{1}{800} \cdot \sqrt{\frac{\rho\epsilon}{n\delta_q\ell}} \cdot \alpha_0'' - 1} + \frac{2n\mathcal{T}'}{\frac{1}{800n\sqrt{\delta_q}} \sqrt{\frac{\mathcal{T}'}{\mathcal{T}'}} - 1} \quad (3.23)$$

in which  $\mathbf{x}_{\mathcal{T}'}$  stands for the  $\mathcal{T}'$ th gradient descent iteration starting from  $\mathbf{x}_0$ .

We first prove the following lemma, which can be viewed as a robust version of [Lemma 2.3](#):

**Lemma 3.5.** *Under the setting of [Lemma 3.4](#), we have:*

$$\mathbb{P}\left(\Delta f'_\perp \geq \frac{5\mathcal{F}'}{8}\right) \leq \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\right) + \mathcal{T}' \cdot \frac{n}{\frac{1}{800n\sqrt{\delta_q}} \sqrt{\frac{\mathcal{T}'}{\mathcal{T}'}} - 1}, \quad (3.24)$$



where  $\Delta f'_\perp$  stands for the function value increase in the eigen-directions other than the most negative one due to quantum simulation and the following  $\mathcal{T}'$  steps of gradient descent. Specifically,

$$\Delta f'_\perp = f(\mathbf{x}_{\mathcal{T}'} - \Delta \mathbf{x}_\parallel) - f(\tilde{\mathbf{x}}), \quad (3.25)$$

where  $\Delta \mathbf{x}_\parallel$  stands for the component of  $\mathbf{x}_{\mathcal{T}'} - \tilde{\mathbf{x}}$  along the most negative eigen-direction.

*Proof.* On the one hand, by [Lemma 2.3](#), the function value increase  $\Delta f'$  in the eigen-directions other than the most negative one due to quantum simulation is at least  $\mathcal{T}'/2$  with probability at most  $\exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\right)$ .

On the other hand, if the gradients can be queried accurately, from [Lemma A.5](#) we know that the function value change caused by gradient descent on each eigen-direction will be negative or zero for each iteration step; however, this is not the case when the gradient is obtained from [Lemma 3.1](#). Nevertheless, from [Lemma 3.2](#), we know that when the accuracy parameter  $\delta_q$  of the quantum evaluation oracle (1.2) is small, with high probability the function value increase in gradient descent steps will be small. Specifically, in each step, with probability at least

$$1 - \frac{n}{\frac{1}{400n\sqrt{\delta_q\ell}}\sqrt{\frac{2c}{\eta}} - 1}, \quad (3.26)$$

the total function value increase caused by the movement along all the eigen-directions except the most negative one can be bounded by  $c$  in each gradient descent step. If we take  $c = \mathcal{T}'/8\mathcal{T}'$ , the function value increase along these directions during the entire  $\mathcal{T}'$  gradient descent steps will be no more than  $\mathcal{T}'/8$ , with probability at least

$$1 - \mathcal{T}' \cdot \frac{n}{\frac{1}{800n\sqrt{\delta_q\ell}}\sqrt{\frac{\mathcal{T}'}{\eta\mathcal{T}'}} - 1} \quad (3.27)$$

by the union bound. In all, we have

$$\mathbb{P}\left(\Delta f'_\perp \geq \frac{5\mathcal{T}'}{8}\right) \leq \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\right) + \mathcal{T}' \cdot \frac{n}{\frac{1}{800n\sqrt{\delta_q\ell}}\sqrt{\frac{\mathcal{T}'}{\eta\mathcal{T}'}} - 1}. \quad (3.28)$$

□

Now we are ready to prove [Lemma 3.4](#).

*Proof.* Following the discussion in [Section 2](#), without loss of generality we assume that  $f$  is quadratic. We can then view the gradient descent process as the combination of  $n$  gradient descent processes, each carried out in one single direction. Without loss of generality, we assume that  $\tilde{\mathbf{x}} = 0$ , otherwise we shift  $\tilde{\mathbf{x}}$  to the original point.

We first consider the gradient descent in the  $\mathbf{e}_1$  direction, where  $\mathbf{e}_1$  stands for the eigenvector of  $\lambda_{\min}$ , or equivalently, the direction with the most negative eigenvalue. We use  $x_\parallel$  to denote the corresponding coordinate and use  $\Phi_m$  to denote the wave function component. By [Proposition A.1](#),  $\Phi_m$  satisfies:

$$\Phi_m(\mathcal{T}', x_\parallel) = \left(\frac{1}{2\pi}\right)^{1/4} \cdot \frac{1}{\sqrt{r_0\sigma(\mathcal{T}'; \lambda_{\min})}} \exp\left(-\frac{x_\parallel^2}{2r_0^2\sigma^2(\mathcal{T}'; \lambda_{\min})}\right). \quad (3.29)$$

Therefore, the probability distribution of the wave packet is:

$$p_m(\mathcal{T}', x_{\parallel}) = \Phi_m(\mathcal{T}', x_{\parallel})^2 = \frac{1}{\sqrt{2\pi} \cdot r_0 \sigma(\mathcal{T}'; \lambda_{\min})} \exp\left(-\frac{x_{\parallel}^2}{r_0^2 \sigma^2(\mathcal{T}'; \lambda_{\min})}\right). \quad (3.30)$$

After  $x_{0\parallel}$  is chosen, as long as the gradient descent sequence is not far from the saddle point, its component  $x_{\parallel}$  in the direction with the most negative eigenvalue satisfies:

$$x_{t+1\parallel} = x'_{t\parallel} + \eta \tilde{\nabla} f(\mathbf{x}_t)_{\parallel}, \quad (3.31)$$

in which the value  $\tilde{\nabla} f(\mathbf{x}_t)_{\parallel}$  stands for the  $x_{\parallel}$  direction component of  $\tilde{\nabla} f(\mathbf{x}_t)$ . Here we assume that in each step we consider, we have

$$|\lambda_{\min} x_{t\parallel} - \tilde{\nabla} f(\mathbf{x}_t)_{\parallel}| \leq -\lambda_{\min} x_{t\parallel}/2; \quad (3.32)$$

we will use this assumption first and calculate the probability of it being not true later. Then we can deduce that

$$x_{t+1\parallel} = x_{t\parallel} + \eta \tilde{\nabla} f(\mathbf{x}_t)_{\parallel} \geq \left(1 - \frac{\eta \lambda_{\min}}{2}\right) x_{t\parallel}, \quad (3.33)$$

and hence we have

$$\mathbb{P}\left(\exists t \in [1, \mathcal{T}'] \mid -\frac{1}{2} \lambda_{\min} x_{t\parallel}^2 \geq \mathcal{T}'\right) = \mathbb{P}\left(\exists t \in [1, \mathcal{T}'] \mid x_{t\parallel} \geq \sqrt{-\frac{2\mathcal{T}'}{\lambda_{\min}}}\right) \quad (3.34)$$

$$= \mathbb{P}\left(\exists t \in [1, \mathcal{T}'] \mid x_{0\parallel} \geq \frac{1}{(1 - \eta \lambda_{\min}/2)^t} \sqrt{-\frac{2\mathcal{T}'}{\lambda_{\min}}}\right) \quad (3.35)$$

$$\geq \mathbb{P}\left(x_{0\parallel} \geq \frac{1}{(1 - \eta \lambda_{\min}/2)^{\mathcal{T}'}} \sqrt{-\frac{2\mathcal{T}'}{\lambda_{\min}}}\right). \quad (3.36)$$

Let  $\alpha := \frac{1}{(1 - \eta \lambda_{\min}/2)^{\mathcal{T}'}} \sqrt{-\frac{2\mathcal{T}'}{\lambda_{\min}}}$ . Since  $\lambda_{\min} \leq -\sqrt{\rho\epsilon}$ ,  $\alpha''_0$  is the upper bound for all possible  $\alpha$ . Thus we have:

$$\mathbb{P}\left(x_{0\parallel} \leq \frac{1}{(1 - \eta \lambda_{\min}/2)^{\mathcal{T}'}} \sqrt{-\frac{2\mathcal{T}'}{\lambda_{\min}}}\right) = \int_{-\alpha}^{\alpha} p_m(t, x_{\parallel}) dx_{\parallel} \quad (3.37)$$

$$\leq \int_{-\alpha''_0}^{\alpha''_0} \frac{1}{\sqrt{2\pi} \cdot r_0 \sigma(\mathcal{T}'; \lambda_{\min})} \exp\left(-\frac{x_{\parallel}^2}{r_0^2 \sigma^2(\mathcal{T}'; \lambda_{\min})}\right) dx_{\parallel} \quad (3.38)$$

$$\leq \int_{-\alpha''_0}^{\alpha''_0} \frac{1}{\sqrt{2\pi} \cdot r_0 \sigma(\mathcal{T}'; -\sqrt{\rho\epsilon})} \exp\left(-\frac{x_{\parallel}^2}{r_0^2 \sigma^2(\mathcal{T}'; -\sqrt{\rho\epsilon})}\right) dx_{\parallel}. \quad (3.39)$$

Note that

$$\int_{-\alpha''_0}^{\alpha''_0} \frac{1}{\sqrt{2\pi} \cdot r_0 \sigma(\mathcal{T}'; -\sqrt{\rho\epsilon})} \exp\left(-\frac{x_{\parallel}^2}{r_0^2 \sigma^2(\mathcal{T}'; -\sqrt{\rho\epsilon})}\right) dx_{\parallel} \leq \frac{2\alpha''_0}{\sqrt{2\pi} \cdot r_0 \sigma(\mathcal{T}'; -\sqrt{\rho\epsilon})}. \quad (3.40)$$

By [Lemma A.3](#) in [Appendix A.2](#), we have:

$$\frac{2\alpha''_0}{\sqrt{2\pi} \cdot r_0 \sigma(\mathcal{T}'; -\sqrt{\rho\epsilon})} \leq \frac{2\alpha''_0}{r_0 \sqrt{2\pi(1 + \mathcal{T}'^2/4)}} \leq \frac{2\sqrt{2}\alpha''_0}{r_0 \mathcal{T}' \sqrt{\pi}}. \quad (3.41)$$

Note that [Lemma A.3](#) contains strong physics intuition:  $1 + \mathcal{T}'^2/4$  is the dispersion (the variance of the probability distribution function) of a free quantum particle, and with a negative potential field this dispersion can only be larger. Furthermore, we can deduce that:

$$\frac{2\sqrt{2}\alpha_0''}{r_0\mathcal{T}'\sqrt{\pi}} = \frac{2\sqrt{2}}{\sqrt{\pi}} \cdot \frac{\sqrt{2}}{\rho(1 + \eta\sqrt{\rho\epsilon}/2)^{\mathcal{T}'}} \cdot \frac{\iota^3\sqrt{n}}{\epsilon} \cdot \frac{\sqrt{\rho\epsilon}}{\iota\ell} \quad (3.42)$$

$$= \frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot [(1 + \eta\sqrt{\rho\epsilon}/2)^{-2/\eta\sqrt{\rho\epsilon}}]^{1/2}. \quad (3.43)$$

Denote  $\mu := \eta\sqrt{\rho\epsilon}$ . Note that the function  $h(\mu) := \ln((1 + \mu)^{1/\mu}) = \ln(1 + \mu)/\mu$  satisfies

$$h'(\mu) = \frac{\frac{\mu}{1+\mu} - \ln(1 + \mu)}{\mu^2} \leq 0 \quad \forall \mu \in (0, 1]; \quad (3.44)$$

therefore,  $h$  is a decreasing function on  $(0, 1]$ , which implies that  $(1 + \mu)^{1/\mu} \geq (1 + 1)^1 = 2$  for any  $\mu \in (0, 1]$ . As a result, as long as  $\eta\sqrt{\rho\epsilon} \leq 1$ , or equivalently  $\epsilon \leq l^2/\rho$ , we have

$$(1 + \eta\sqrt{\rho\epsilon}/2)^{2/(\eta\sqrt{\rho\epsilon})} \geq 2, \quad (3.45)$$

which then implies

$$\mathbb{P}(x_{0\parallel} \leq \frac{1}{(1 - \eta\lambda_{\min}/2)^{\mathcal{T}'}} \sqrt{-\frac{2\mathcal{T}'}{\lambda_{\min}}}) \leq \frac{2\sqrt{2}\alpha_0''}{r_0\mathcal{T}'\sqrt{\pi}} \leq \frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot 2^{-\iota/2}. \quad (3.46)$$

Suppose we already have

$$x_{0\parallel} \geq \alpha_0'', \quad (3.47)$$

then from [Lemma 3.1](#), with probability at least

$$1 - \frac{\mathcal{T}'}{\frac{1}{800} \cdot \sqrt{\frac{\rho\epsilon}{n\delta_q\ell}} \cdot \alpha_0'' - 1}, \quad (3.48)$$

Eq. (3.32) will be true in each gradient descent step.

Now we are ready to prove (3.21). It follows from:

$$\begin{aligned} & \mathbb{P}\left(f(\mathbf{x}_{\mathcal{T}}) - f(\tilde{\mathbf{x}}) \leq -\frac{\mathcal{F}'}{4}\right) \\ & \geq \mathbb{P}(x_{0\parallel} \geq \alpha_0'') \cdot \left(1 - \frac{\mathcal{T}'}{\frac{1}{800} \cdot \sqrt{\frac{\rho\epsilon}{n\delta_q\ell}} \cdot \alpha_0'' - 1}\right) \cdot \mathbb{P}\left(\Delta f_{\perp} \leq \frac{5\mathcal{F}'}{8}\right) \cdot \mathbb{P}\left(\Delta f_{\text{abs}} \leq \mathcal{F}'/8\right), \end{aligned} \quad (3.49)$$

where

$$\Delta f_{\text{abs}} := \frac{1}{2} \sum_{t=1}^{\mathcal{T}'} (f(\mathbf{x}_t) - f(\mathbf{x}_{t-1}) + |f(\mathbf{x}_t) - f(\mathbf{x}_{t-1})|) \quad (3.50)$$

is the upper bound for function value increase caused by the inaccuracy of quantum evaluation oracle. Note that in each step, with probability at least

$$1 - \mathcal{T}' \cdot \frac{n}{\frac{1}{800n\sqrt{\delta_q\ell}} \sqrt{\frac{\mathcal{F}'}{\eta\mathcal{T}'} - 1}}, \quad (3.51)$$

the total function value increase can be bounded by  $\mathcal{F}'/8$ . Then by the union bound,

$$\begin{aligned} \mathbb{P}\left(f(\mathbf{x}_{\mathcal{T}}) - f(\tilde{\mathbf{x}}) \leq -\frac{\mathcal{F}'}{4}\right) &\geq 1 - \frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot 2^{-\iota/2} - \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\right) \\ &\quad - \frac{\mathcal{F}'}{\frac{1}{800} \cdot \sqrt{\frac{\rho\epsilon}{n\delta_q\ell}} \cdot \alpha_0'' - 1} - \frac{2n\mathcal{F}'}{\frac{1}{800n\sqrt{\delta_q}} \sqrt{\frac{\mathcal{F}'}{\mathcal{T}'}} - 1}. \end{aligned} \quad (3.52)$$

□

### 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 2.2](#), it reduces classical gradient queries to the same number of quantum evaluation queries.

**Theorem 3.1.** *Let  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  be an  $\ell$ -smooth,  $\rho$ -Hessian Lipschitz function. Suppose that we have the quantum evaluation oracle  $U_f$  in [\(1.2\)](#) with accuracy  $\delta_q \leq O\left(\frac{\epsilon^7\delta^2}{n^4\iota^5} \cdot 2^{-\iota}\right)$ . Then [Algorithm 4](#) finds an  $\epsilon$ -approximate local minimum satisfying [\(1.1\)](#), using*

$$\tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \cdot \iota^2\right) \quad (3.53)$$

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

$$\iota := k \cdot \log(n\ell(f(\mathbf{x}_0) - f^*)/(\rho\epsilon\delta)) \quad \eta := \frac{1}{\ell} \quad (3.54)$$

$$\mathcal{T}' := \frac{\ell}{\sqrt{\rho\epsilon}} \cdot \iota \quad r_0 := \frac{\epsilon}{\iota^3} \cdot \frac{1}{\sqrt{n}} \quad (3.55)$$

where  $k$  is a large absolute constant,  $\mathbf{x}_0$  is the start point, and  $f^*$  is the global minimum of  $f$ .

Note that [Theorem 3.1](#) 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.* We first specify our choice for iteration times:

$$T = 8 \max\left\{\frac{(f(\mathbf{x}_0) - f^*)\mathcal{T}'}{\mathcal{F}'}, \frac{2(f(\mathbf{x}_0) - f^*)}{\eta\epsilon^2}\right\} = \tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \cdot \iota\right), \quad (3.56)$$

similar to our choice in [Theorem 2.2](#). We choose  $k$  to be large enough to satisfy:

$$T \cdot \left(\frac{4\iota^2}{\ell} \cdot \sqrt{\frac{n}{\pi\rho\epsilon}} \cdot 2^{-\iota/2} + \exp\left(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\right)\right) \leq \frac{\delta}{3}. \quad (3.57)$$

As for the accuracy  $\delta_q$  of the quantum evaluation oracle, we first let it small enough to promise the following simplifications:

$$\begin{cases} \frac{1}{800} \cdot \sqrt{\frac{\rho\epsilon}{n\delta_q\ell}} \cdot \alpha_0'' \geq 2 \\ \frac{1}{800n\sqrt{\delta_q}} \sqrt{\frac{\mathcal{F}'}{\mathcal{T}'}} \geq 2, \end{cases} \quad (3.58)$$

which are equivalent to

$$\begin{cases} \delta_q \leq O\left(\frac{\rho\epsilon}{n\ell'} \cdot \alpha_0''^2\right) = O\left(\frac{\epsilon}{n} \cdot 2^{-\iota}\right) \\ \delta_q \leq O\left(\frac{\mathcal{F}'}{n^2\mathcal{T}'}\right) = O\left(\frac{\epsilon}{n^2\iota}\right) \end{cases} \quad (3.59)$$

and are satisfied as long as

$$\delta_q \leq \left(\frac{\epsilon}{n^2\iota} 2^{-\iota}\right). \quad (3.60)$$

Under this setting, we can find that

$$\frac{\mathcal{T}'}{\frac{1}{800} \cdot \sqrt{\frac{\rho\epsilon}{n\delta_q\ell}} \cdot \alpha_0'' - 1} + \frac{2n\mathcal{T}'}{\frac{1}{800n\sqrt{\delta_q}} \sqrt{\frac{\mathcal{F}'}{\mathcal{T}'}} - 1} \leq 2\mathcal{T}'\sqrt{\delta_q} \left( \frac{1}{\frac{1}{800} \cdot \sqrt{\frac{\rho\epsilon}{n\delta_q\ell}} \cdot \alpha_0''} + \frac{2n}{\frac{1}{800n\sqrt{\delta_q}} \sqrt{\frac{\mathcal{F}'}{\mathcal{T}'}}} \right). \quad (3.61)$$

Furthermore, if  $\delta_q$  is small enough that satisfies

$$T \cdot 2\mathcal{T}'\sqrt{\delta_q} \cdot \left( \frac{1}{\frac{1}{800} \cdot \sqrt{\frac{\rho\epsilon}{n\ell}} \cdot \alpha_0''} + \frac{2n}{\frac{1}{800n} \sqrt{\frac{\mathcal{F}'}{2\mathcal{T}'}}} \right) \leq \frac{\delta}{3}, \quad (3.62)$$

together with (3.57) and (3.58), it can then be guaranteed that with probability at least  $1 - 2\delta/3$ , after each time quantum simulation is called, function value will decrease at least  $\mathcal{F}'/4$  in the following  $\mathcal{T}'$  steps. Eq. (3.62) can be satisfied if we take

$$\delta_q \leq \frac{\delta^2}{36\mathcal{T}'^2T^2} \cdot \left( \frac{1}{\frac{1}{800} \cdot \sqrt{\frac{\rho\epsilon}{n\ell}} \cdot \alpha_0''} + \frac{2n}{\frac{1}{800n} \sqrt{\frac{\mathcal{F}'}{\mathcal{T}'}}} \right)^{-2} \quad (3.63)$$

$$= \frac{\delta^2}{36\mathcal{T}'^2T^2} \left( O\left(\frac{\sqrt{n}}{\epsilon} \cdot 2^{\iota/2}\right) + O\left(n^2\sqrt{\frac{\iota}{\epsilon}}\right) \right)^{-2} \quad (3.64)$$

$$= O\left(\delta^2 \cdot \frac{\epsilon^4}{\iota^2} \cdot \frac{\epsilon}{\iota^2} \cdot \frac{\epsilon^2}{n^4\iota} \cdot 2^{-\iota}\right) \quad (3.65)$$

$$= O\left(\frac{\epsilon^7\delta^2}{n^4\iota^5} \cdot 2^{-\iota}\right). \quad (3.66)$$

Since we perturb at most once in  $\mathcal{T}'$  consecutive steps, we only need to apply the quantum simulation for at most  $T/\mathcal{T}'$  times. Then by Lemma 3.5, with probability at least  $1 - \delta$ , after each time a perturbation was added, the function value decreases at least  $\mathcal{F}'/4$  in the following  $\mathcal{T}'$  steps. As a consequence, there can at most be  $\frac{4(f(\mathbf{x}_0) - f^*)}{\mathcal{F}'}$  perturbations.

Excluding those iterations that are within  $\mathcal{T}'$  steps after adding perturbations, we still have  $T/2$  steps left. They are either large gradient steps,  $\|\nabla f(\mathbf{x}_t)\| \geq \epsilon$ , or  $\epsilon$ -approximate second-order stationary points. Within them, for each large gradient steps, by Lemma 3.2, with probability at least

$$1 - \frac{n}{\frac{1}{400n} \sqrt{\frac{2}{\delta_q} \cdot \frac{\eta\epsilon^2}{4}} - 1} = 1 - \frac{n}{\frac{\epsilon}{400n} \sqrt{\frac{1}{2\delta_q\ell}} - 1}, \quad (3.67)$$

the function value decrease is greater than  $\eta\epsilon^2/4$ . We consider two additional constraints on  $\delta_q$ :

$$\begin{cases} \frac{\epsilon}{400n} \sqrt{\frac{1}{2\delta_q\ell}} \geq 2 \\ T \cdot \frac{2n}{\frac{\epsilon}{400n} \sqrt{\frac{1}{2\delta_q\ell}}} \leq \frac{\delta}{3}, \end{cases} \quad (3.68)$$

which have been satisfied because of

$$\delta_q \leq O\left(\frac{\epsilon^7 \delta^2}{n^4 \iota^5} \cdot 2^{-\iota}\right) \leq O\left(\frac{\delta^2 \epsilon^2}{n^4 T^2}\right). \quad (3.69)$$

As a result, with probability at least  $1 - \delta/3$ , each large gradient step will have function value decrease at most  $\eta\epsilon^2/4$ , under which circumstance 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  $\mathcal{T}'$  steps after calling quantum simulation, and at most  $T/4$  steps have a gradient greater than  $\epsilon$ . As a result, the rest  $T/4$  steps must all be  $\epsilon$ -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  $1 - \delta$ ,

$$T_1 = T = \tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \cdot \iota\right). \quad (3.70)$$

As for  $T_2$ , with probability at least  $1 - \delta$  quantum simulation is called for at most  $\frac{4(f(\mathbf{x}_0) - f^*)}{\mathcal{T}'}$  times, and by [Lemma 2.2](#) it takes  $\tilde{O}(\mathcal{T}' \log n \log^2(\mathcal{T}'^2/\epsilon))$  queries to carry out each simulation. Therefore,

$$T_2 = \frac{4(f(\mathbf{x}_0) - f^*)}{\mathcal{T}'} \cdot \tilde{O}(\mathcal{T}' \log n \log^2(\mathcal{T}'^2/\epsilon)) = \tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon} \cdot \iota^2\right). \quad (3.71)$$

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

$$\tilde{O}\left(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^2} \cdot \iota^2\right). \quad (3.72)$$

□

[Theorem 2.3](#) and [Theorem 3.1](#) together imply the main result [Theorem 1.1](#) of this paper.

## 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 [\[37\]](#), 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](#)):

$$i \frac{d\Psi}{dt} = H\Psi, \quad (4.1)$$



where  $\Psi: [0, T] \rightarrow \mathbb{C}^N$  is a vector-valued function in time. We may have a decomposition  $\Psi(t) = Q(t) + iP(t)$  for  $Q, P: [0, T] \rightarrow \mathbb{R}^N$  being the real and imaginary part of  $\Psi$ , respectively. Then plugging the decomposition into the ODE (4.1), we have a separable  $N$ -body Hamiltonian system

$$\begin{cases} \dot{Q} = HP; \\ \dot{P} = -HQ. \end{cases} \quad (4.2)$$

The optimal integration scheme for solving this Hamiltonian system is the symplectic integrator [37], and we use a second-order leapfrog integrator for separable canonical Hamiltonian systems [28] in this section. In all of our PDE simulations, we fix the spatial domain to be  $\Omega = \{(x, y) : |x| \leq 3, |y| \leq 3\}$  and the mesh number to be 512 on each edge.

**Dispersion of the wave packet.** In Proposition A.1, 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 A.1 ( $r = 0.5$ ). 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.

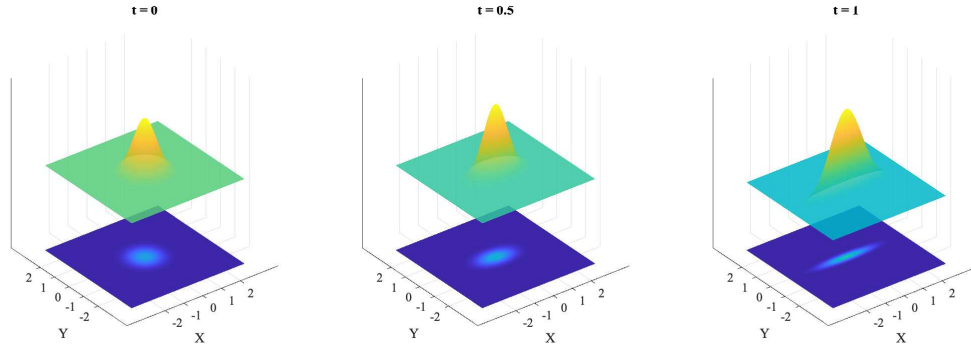


Figure 1: Dispersion of wave packet over the potential field  $f_1(x, y)$ . We use the finite difference method (5-point stencil) and the Leapfrog integration to simulate the Schrödinger equation (2.3) on a square domain (center =  $(0, 0)$ , edge = 6), up to  $T = 1$ . The mesh number is 512 on each 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.

**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  $\Omega$  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

$$D^2f(x, y) = \begin{pmatrix} x^2 - 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (4.3)$$

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” bivariate Gaussian distribution assigns more probability mass along the  $x$ -axis, thus allowing escaping from the saddle point more efficiently.

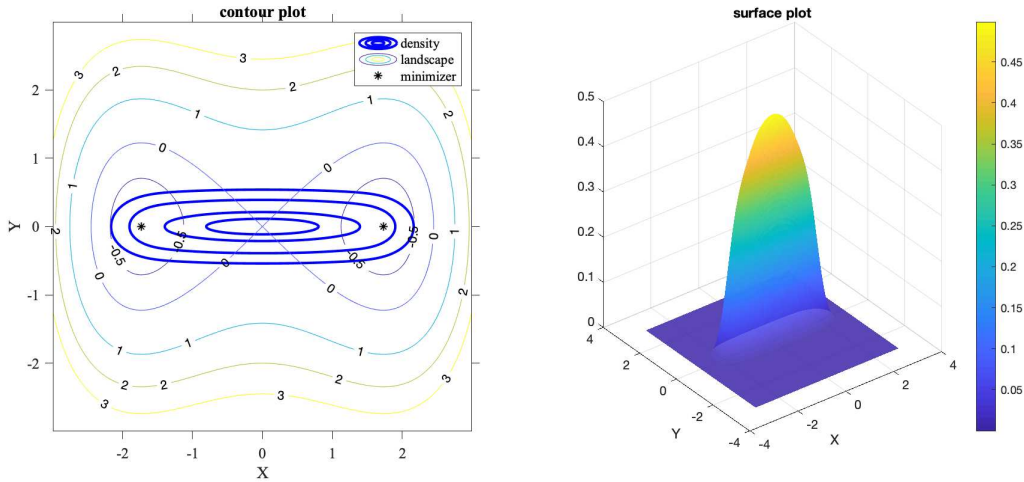


Figure 2: Quantum simulation on landscape 1:  $f(x, y) = \frac{1}{12}x^4 - \frac{1}{2}x^2 + \frac{1}{2}y^2$ . Parameters:  $r_0 = 0.5$ ,  $t_e = 1.5$ . **Left:** The contour of the landscape is placed on the background with labels being function values; the thick blue contours illustrate the wave packet 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

$$D^2g(x, y) = \begin{pmatrix} 6x & -2 \\ -2 & -6y \end{pmatrix}. \quad (4.4)$$

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.

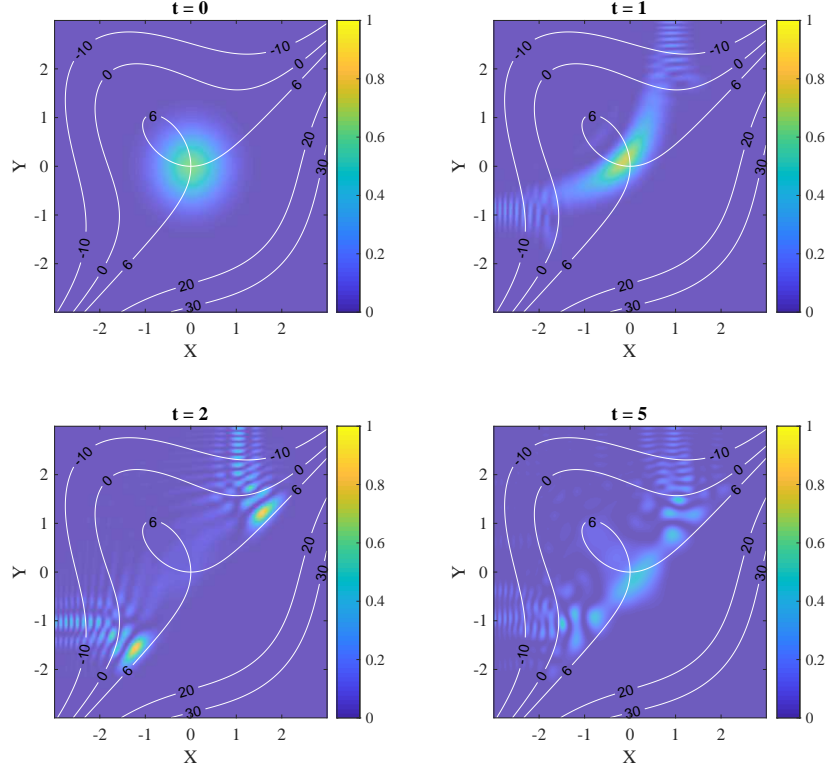


Figure 3: Quantum simulation on landscape 2:  $g(x, y) = x^3 - y^3 - 2xy + 6$ . Parameters:  $r_0 = 0.5$ ,  $t_e = 5$ . In each subplot, a colored contour plot of the wave packet at a specific time is shown, and the landscape contour is placed on top of the wave packet for quick reference. The average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

**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 [42] 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 [43, Algorithm 4]):  $x_0$  is uniformly sampled from the ball  $B_0(r)$  (saddle point at the origin), and then run  $\mathcal{T}_c$  gradient descent steps to obtain  $x_{\mathcal{T}_c}$ . Record the function value  $f(x_{\mathcal{T}_c})$ . 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  $\mathcal{T}_q$  gradient descent steps and record the function value  $f(x_{\mathcal{T}_q})$ . 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.

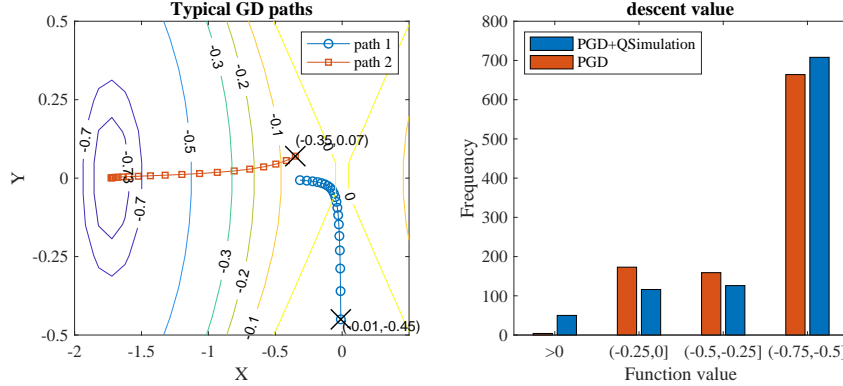


Figure 4: **Left:** Two typical gradient descent paths on the landscape of  $f_2$  illustrated as a contour plot. Path 1 (resp. 2) starts from  $(-0.01, 0.45)$  (resp.  $(-0.35, 0.07)$ ); both have step length  $\eta = 0.2$  and  $T = 20$  iterations. Note that path 2 approaches the local minimum  $(-\sqrt{3}, 0)$ , 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  $f_2(x_{\mathcal{T}_c})$  (PGD) and  $f_2(x_{\mathcal{T}_q})$  (Algorithm 2). We set step length  $\eta = 0.05$ ,  $r = 0.5$  (ball radius in PGD and  $r_0$  in Algorithm 1),  $M = 1000$ ,  $\mathcal{T}_c = 50$ ,  $\mathcal{T}_q = 10$ ,  $t_e = 1.5$ . 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.

**Dimension dependence.** Recall that  $n$  is the dimension of the problem. Classically, it has been shown in [43] 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 2.2. 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:  $\mathcal{H} = \text{diag}(-\epsilon, 1, 1, \dots, 1)$ . 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  $\mathcal{T}_c$ , quantum number of iterations  $\mathcal{T}_q$ , quantum evolution time  $t_e$ , number of samples  $M \in \mathbb{N}$ , and GD step size (learning rate)  $\eta$ . For the sake of comparison, the iteration numbers  $\mathcal{T}_c$  and  $\mathcal{T}_q$  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.

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  $\mathcal{T}_c = \Theta(\log^2 n)$  and  $\mathcal{T}_q = \Theta(\log n)$ , 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).

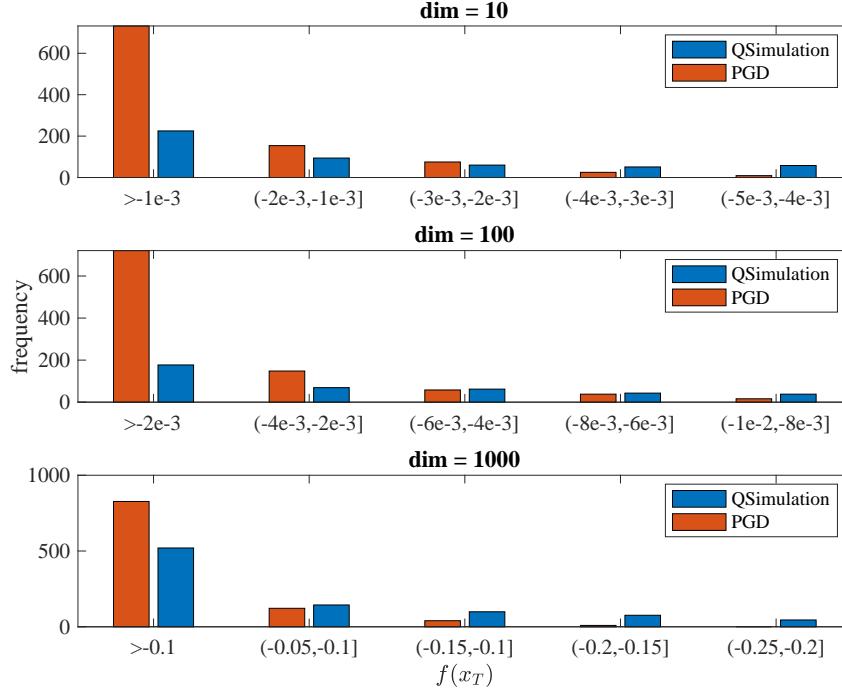


Figure 5: Dimension dependence of classical and quantum algorithms. We set  $\epsilon = 0.01$ ,  $r = 0.1$ ,  $n = 10^p$  for  $p = 1, 2, 3$ . Quantum evolution time  $t_e = p$ , classical iteration number  $\mathcal{T}_c = 50p^2 + 50$ , quantum iteration number  $\mathcal{T}_q = 30p$ , and sample size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

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 [43] has better dimension dependence than  $O(\log^4 n)$ .

## Acknowledgements

We thank Andrew M. Childs, Jin-Peng Liu, and Xiaodi Wu for helpful discussions. 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), and the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Algorithms Teams program.

## 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](#).
- [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](#).
- [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](#).
- [6] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf, *Convex optimization using quantum oracles*, Quantum **4** (2020), 220, [arXiv:1809.00643](#).
- [7] Frank Arute et al., *Quantum supremacy using a programmable superconducting processor*, Nature **574** (2019), no. 7779, 505–510, [arXiv:1910.11333](#).
- [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 Mathematical Physics **270** (2007), no. 2, 359–371, [arXiv:quant-ph/0508139](#).
- [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](#).
- [11] Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro, *Global optimality of local search for low rank matrix recovery*, Proceedings of the 30th International Conference on Neural Information Processing Systems, pp. 3880–3888, 2016, [arXiv:1605.07221](#).
- [12] 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](#).
- [13] 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](#).
- [14] 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](#).
- [15] 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](#).



- [16] 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](#).
- [17] 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](#).
- [18] Nai-Hui Chia, Tongyang Li, Han-Hsuan Lin, and Chunhao Wang, *Quantum-inspired sublinear algorithm for solving low-rank semidefinite programming*, To appear in the proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science, 2020, [arXiv:1901.03254](#).
- [19] Nai-Hui Chia, Han-Hsuan Lin, and Chunhao Wang, *Quantum-inspired sublinear classical algorithms for solving low-rank linear systems*, 2018, [arXiv:1811.04852](#).
- [20] Andrew M. Childs, *Lecture notes on quantum algorithms*, Lecture notes at University of Maryland, 2017, <https://www.cs.umd.edu/%7Eamchilds/qa/qa.pdf>.
- [21] 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](#).
- [22] Andrew M. Childs, Jin-Peng Liu, and Aaron Ostrander, *High-precision quantum algorithms for partial differential equations*, 2020, [arXiv:2002.07868](#).
- [23] 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](#).
- [24] 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.
- [25] 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](#).
- [26] 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](#).
- [27] 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](#).
- [28] Mauger Franois, *Symplectic leap frog scheme*, 2020 (Retrieved May 30, 2020), <https://www.mathworks.com/matlabcentral/fileexchange/38652-symplectic-leap-frog-scheme>.
- [29] 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](#).
- [30] 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](#).

- [31] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan, *Escaping from saddle points – on-line stochastic gradient for tensor decomposition*, Proceedings of the 28th Conference on Learning Theory, Proceedings of Machine Learning Research, vol. 40, pp. 797–842, 2015, [arXiv:1503.02101](#).
- [32] 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](#).
- [33] 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](#).
- [34] 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](#).
- [35] 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](#).
- [36] András Gilyén, Seth Lloyd, and Ewin Tang, *Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension*, 2018, [arXiv:1811.04909](#).
- [37] 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.
- [38] 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](#).
- [39] 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](#).
- [40] 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](#).
- [41] 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](#).
- [42] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan, *How to escape saddle points efficiently*, Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1724–1732, 2017, [arXiv:1703.00887](#).
- [43] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M. Kakade, and Michael I. Jordan, *Stochastic gradient descent escapes saddle points efficiently*, 2019, [arXiv:1902.04811](#).
- [44] 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](#).

- [45] Michael I. Jordan, *On gradient-based optimization: Accelerated, distributed, asynchronous and stochastic optimization*, 2017, <https://www.youtube.com/watch?v=VE2ITg%5FhGnI>.
- [46] Stephen P. Jordan, *Fast quantum algorithm for numerical gradient estimation*, Physical Review Letters **95** (2005), no. 5, 050501, [arXiv:quant-ph/0405146](#).
- [47] 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](#).
- [48] Iordanis Kerenidis and Anupam Prakash, *A quantum interior point method for LPs and SDPs*, 2018, [arXiv:1808.09266](#).
- [49] Alexei Kitaev and William A. Webb, *Wavefunction preparation and resampling using a quantum computer*, 2008, [arXiv:0801.0342](#).
- [50] Kfir Y. Levy, *The power of normalization: Faster evasion of saddle points*, 2016, [arXiv:1611.04831](#).
- [51] Jianping Li, *General explicit difference formulas for numerical differentiation*, Journal of Computational and Applied Mathematics **183** (2005), no. 1, 29–52.
- [52] Seth Lloyd, *Universal quantum simulators*, Science **273** (1996), no. 5278, 1073.
- [53] 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](#).
- [54] Guang Hao Low and Isaac L. Chuang, *Hamiltonian simulation by qubitization*, Quantum **3** (2019), 163, [arXiv:1610.06546](#).
- [55] Guang Hao Low and Nathan Wiebe, *Hamiltonian simulation in the interaction picture*, 2018, [arXiv:1805.00675](#).
- [56] Yurii Nesterov and Boris T. Polyak, *Cubic regularization of Newton method and its global performance*, Mathematical Programming **108** (2006), no. 1, 177–205.
- [57] 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.
- [58] John Preskill, *Quantum computing in the NISQ era and beyond*, Quantum **2** (2018), 79, [arXiv:1801.00862](#).
- [59] 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](#).
- [60] Ewin Tang, *Quantum-inspired classical algorithms for principal component analysis and supervised clustering*, 2018, [arXiv:1811.00414](#).
- [61] 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](#).
- [62] 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](#).

- [63] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J. Cerf, Timothy C. Ralph, Jeffrey H. Shapiro, and Seth Lloyd, *Gaussian quantum information*, Reviews of Modern Physics **84** (2012), no. 2, 621, [arXiv:1110.3234](#).
- [64] Stephen Wiesner, *Simulations of many-body quantum systems by a quantum computer*, 1996, [arXiv:quant-ph/9603028](#).
- [65] Yi Xu, Rong Jin, and Tianbao Yang, *NEON+: Accelerated gradient methods for extracting negative curvature for non-convex optimization*, 2017, [arXiv:1712.01033](#).
- [66] 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](#).
- [67] Christof Zalka, *Efficient simulation of quantum systems by quantum computers*, Fortschritte der Physik: Progress of Physics **46** (1998), no. 6-8, 877–879.
- [68] 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](#).
- [69] 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](#).
- [70] 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](#).
- [71] 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 2.1.** *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 (2.1). Then, at any time  $t \geq 0$ , the position of the quantum particle still follows normal distribution  $\mathcal{N}(0, \sigma^2(t; \lambda))$ , where the variance  $\sigma^2(t; \lambda)$  is given by*

$$\sigma^2(t; \lambda) = \begin{cases} 1 + \frac{t^2}{4} & (\lambda = 0), \\ \frac{(1+4\alpha^2) - (1-4\alpha^2)\cos 2\alpha t}{8\alpha^2} & (\lambda > 0, \alpha = \sqrt{\lambda}), \\ \frac{(1-e^{2\alpha t})^2 + 4\alpha^2(1+e^{2\alpha t})^2}{16\alpha^2 e^{2\alpha t}} & (\lambda < 0, \alpha = \sqrt{-\lambda}). \end{cases} \quad (2.2)$$

*Proof.* Due to the well-posedness of the Schrödinger equation, if we find a solution to the initial value problem (2.1), this solution is unique. We take the following ansatz

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

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

$$p_\lambda(t, x) := |\Psi(t, x)|^2 = \frac{1}{\sqrt{\pi}} \frac{1}{|\delta(t)|} \exp\left(2 \operatorname{Im}(\theta(t))\right) \exp\left(-x^2 \operatorname{Re}(1/y(t))\right), \quad (\text{A.2})$$

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

If the ansatz (A.1) 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  $\int_{\mathbb{R}} p_\lambda(t, x) dx = 1$  for all  $t \geq 0$ . It is now clear that (A.2) is the density of a Gaussian random variable with zero mean and variance

$$\sigma^2(t; \lambda) = \frac{1}{2 \operatorname{Re}(1/y(t))}. \quad (\text{A.3})$$

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 (A.1) to (2.1) 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

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

**Case 1:**  $\lambda = 0$ . The system (A.4) is linear with solutions

$$y(t) = 2 + it. \quad (\text{A.5})$$

It follows that

$$\frac{1}{y(t)} = \frac{2}{4 + t^2} - i \frac{t}{4 + t^2}, \quad (\text{A.6})$$

And by Equation (A.3), the variance is

$$\sigma^2(t; 0) = 1 + \frac{t^2}{4}. \quad (\text{A.7})$$

**Case 2:**  $\lambda \neq 0$ . The equation  $y' + i\lambda y^2 - i = 0$  in (A.4) 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

$$u'' + \lambda u = 0. \quad (\text{A.8})$$

Clearly, the sign of  $\lambda$  matters.

**Case 2.1:**  $\lambda > 0$ . Let  $\alpha = \sqrt{\lambda}$ , the solution to (A.8) is  $u(t) = c_1 e^{i\alpha t} + c_2 e^{-i\alpha t}$  ( $c_1, c_2$  are constants), and

$$y(t) = \frac{-i u'}{\lambda u} = \frac{1}{\alpha} \frac{c_1 e^{i\alpha t} - c_2 e^{-i\alpha t}}{c_1 e^{i\alpha t} + c_2 e^{-i\alpha t}}. \quad (\text{A.9})$$

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

$$y(t) = \frac{1}{\alpha} \left( \frac{e^{2i\alpha t} - \beta}{e^{2i\alpha t} + \beta} \right). \quad (\text{A.10})$$

By (A.3) and (A.10), we attain the variance when  $\lambda > 0$ .

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

$$y(t) = \frac{i}{\alpha} \frac{e^{2\alpha t} - \beta}{e^{2\alpha t} + \beta}, \quad (\text{A.11})$$

where  $\beta = \frac{1+2i\alpha}{1-2i\alpha}$ . And the variance  $\sigma(t; \lambda)$  for  $\lambda < 0$  can be obtained from (A.3) and (A.11).  $\square$

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

**Lemma A.1** ( $n$ -dimensional evolution). *Let  $\mathcal{H}$  be an  $n$ -by- $n$  symmetric matrix with diagonalization  $\mathcal{H} = U^T \Lambda U$ , with  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$  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  $\Phi(0, \mathbf{x}) = (\frac{1}{2\pi})^{n/4} \exp(-\|\mathbf{x}\|^2/4)$ ; 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*

$$\Sigma(t) = U^T \text{diag}(\sigma^2(t; \lambda_1), \dots, \sigma^2(t; \lambda_n)) U. \quad (\text{A.12})$$

The function  $\sigma(t; \lambda)$  is defined in (2.2).

*Proof.* The proof follows the same idea in Lemma 2.1. We take the following ansatz

$$\Phi(t, \mathbf{x}) = \left( \frac{1}{\pi} \right)^{n/4} (\det \Delta(t))^{-1/4} \exp(-i\theta(t)) \exp \left[ -\frac{1}{2} \mathbf{x}^T (\Delta(t))^{-1} \mathbf{x} \right], \quad (\text{A.13})$$

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

The global phase parameter  $\theta(t)$ , together with the factor  $(\frac{1}{\pi})^{n/4} (\det \Delta(t))^{-1/4}$ , will contribute to a scalar factor in the probability density function such that the  $L^2$ -norm of the wave function (A.13) will remain unit 1. It is the matrix  $\Delta(t)$  that controls the covariance matrix (see (A.18)). Regarding this, we do not delve into the derivation of  $\theta(t)$  in this proof.

Substituting the ansatz (A.13) to the Schrödinger equation (2.1), we have the following system of ordinary differential equations:

$$\frac{d}{dt} (\Delta(t)^{-1}) + i\Delta(t)^{-2} - i\mathcal{H} = 0, \quad (\text{A.14})$$

$$\dot{\theta} = \frac{i}{4} (\det \Delta(t))^{-1} \frac{d}{dt} (\Delta(t)) + \frac{1}{2} \text{Tr}[\Delta(t)^{-1}]. \quad (\text{A.15})$$

We immediately observe that Eq. (A.14) is a decoupled system

$$\frac{d}{dt} \left( \frac{1}{\delta_j(t)^2} \right) + i \frac{1}{(\delta_j(t))^4} - i\lambda_j = 0, \text{ for } j = 1, \dots, n. \quad (\text{A.16})$$



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

$$\dot{y}_j + i\lambda_j y_j^2 - i = 0, \text{ for } j = 1, \dots, n. \quad (\text{A.17})$$

They are precisely the same as the first equation in (A.4), thus the calculation of one-dimensional case in Lemma 2.1 applies directly to (A.17).

Given the ansatz (A.13), 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

$$\Sigma(t) = (2 \operatorname{Re} \Delta^{-1}(t))^{-1} = U^T \left( \frac{1}{2 \operatorname{Re}(1/y_1(t))}, \dots, \frac{1}{2 \operatorname{Re}(1/y_n(t))} \right) U. \quad (\text{A.18})$$

It follows from (A.3) and (2.2) that the covariance matrix is given as (A.12).  $\square$

Finally, we state the following proposition with different scales:

**Proposition A.1.** *Let  $\mathcal{H}$  be an  $n$ -by- $n$  symmetric matrix with diagonalization  $\mathcal{H} = U^T \Lambda U$ , with  $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$  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  $\Phi(0, \mathbf{x}) = (\frac{1}{2\pi})^{n/4} r^{-n/2} \exp(-\|\mathbf{x}\|^2/4r^2)$ ; 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 (2.3). 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*

$$\Sigma(t) = U^T \operatorname{diag}(\sigma^2(t; \lambda_1), \dots, \sigma^2(t; \lambda_n)) U. \quad (\text{A.19})$$

The function  $\sigma(t; \lambda)$  is the same as in (2.2).

*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 A.1. Let  $\Phi(t, x)$  be the wave function as in Lemma 2.1, namely, it satisfies the standard Schrödinger equation (2.1). 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 (2.3). To do so, we substitute  $\Psi(t, x)$  back to (2.3). Its LHS is just  $i \frac{\partial}{\partial t} \frac{1}{\sqrt{r}} \Phi(t, x/r)$ , whereas the RHS is

$$\left[ -\frac{r^2}{2} \nabla^2 + \frac{1}{r^2} f(\mathbf{x}) \right] \Psi(t, x) = \frac{1}{\sqrt{r}} \left[ -\frac{1}{2} \nabla^2 + \frac{1}{2} \left( \frac{\mathbf{x}}{r} \right)^T \mathcal{H} \left( \frac{\mathbf{x}}{r} \right) \right] \Phi \left( t, \frac{x}{r} \right). \quad (\text{A.20})$$

Since  $\Phi(t, x)$  satisfies (2.1), 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  $r^2 \sigma^2(t; \lambda)$ .  $\square$

## A.2 Variance of Gaussian wave packets

Although the variance of the Gaussian wave packet  $\sigma(\lambda; t)$  is explicitly given in (2.2), 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 A.2.** *When  $\lambda > 0$ ,*

$$\min \left\{ 1, \frac{1}{2\alpha} \right\} \leq \sigma(t; \lambda) \leq \max \left\{ 1, \frac{1}{2\alpha} \right\}. \quad (\text{A.21})$$

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

$$\frac{1}{\sqrt{2}}\varphi(t; \alpha) \leq \sigma(t; \lambda) \leq \varphi(t; \alpha), \quad (\text{A.22})$$

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

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

□

**Lemma A.3.** When  $\lambda < 0$ ,

$$\sigma^2(t; \lambda) \geq 1 + \frac{t^2}{4}. \quad (\text{A.23})$$

*Proof.* Recall (2.2) that  $\sigma(t; \lambda)$  equals to:

$$\sigma^2(t; \lambda) = \frac{(1 - e^{2\alpha t})^2 + 4\alpha^2(1 + e^{2\alpha t})^2}{16\alpha^2 e^{2\alpha t}}, \quad (\text{A.24})$$

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

$$\sigma^2(t; \lambda) = \frac{(1 + 4\alpha^2)e^{4\alpha t} + (1 + 4\alpha^2) - 2(1 - 4\alpha^2)e^{2\alpha t}}{16\alpha^2 e^{2\alpha t}} \quad (\text{A.25})$$

$$= \frac{(1 + 4\alpha^2)e^{2\alpha t} + (1 + 4\alpha^2)e^{-2\alpha t} - 2(1 - 4\alpha^2)}{16\alpha^2}. \quad (\text{A.26})$$

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

$$e^\mu = 1 + \mu + \frac{\mu^2}{2} + \frac{\mu^3}{6} + \frac{e^\zeta}{24}\mu^4; \quad (\text{A.27})$$

$$e^{-\mu} = 1 - \mu + \frac{\mu^2}{2} - \frac{\mu^3}{6} + \frac{e^{-\xi}}{24}\mu^4. \quad (\text{A.28})$$

Adding these two equations, we have

$$e^\mu + e^{-\mu} \geq 2 + \mu^2 + \frac{\mu^4}{24}(e^\zeta + e^{-\xi}) \geq 2 + \mu^2 + \frac{\mu^4}{24}(1 + e^{-\mu}) \geq 2 + \mu^2. \quad (\text{A.29})$$

In other words,

$$e^{2\alpha t} + e^{-2\alpha t} \geq 2 + (2\alpha t)^2, \quad (\text{A.30})$$

which results in

$$\frac{(1 + 4\alpha^2)e^{2\alpha t} + (1 + 4\alpha^2)e^{-2\alpha t} - 2(1 - 4\alpha^2)}{16\alpha^2} \geq \frac{(1 + 4\alpha^2)(2 + 4\alpha^2 t^2) - 2(1 - 4\alpha^2)}{16\alpha^2} \quad (\text{A.31})$$

$$\geq \frac{16\alpha^2 + 4\alpha^2 t^2}{16\alpha^2} \quad (\text{A.32})$$

$$= 1 + \frac{t^2}{4}; \quad (\text{A.33})$$



or equivalently,

$$\sigma^2(t; \lambda) \geq 1 + \frac{t^2}{4}. \quad (\text{A.34})$$

□

In the proof of [Proposition 2.1](#), we will also use the following fact about multivariate Gaussian distributions:

**Lemma A.4** ([40, Proposition 1]). *Let  $A \in \mathbb{R}^{m \times n}$  be a matrix, and let  $\Sigma := A^T A$ . Let  $\mathbf{x} = (x_1, \dots, x_n)$  be an isotropic multivariate Gaussian random vector with mean zero. For all  $t > 0$ :*

$$\mathbb{P}\left(\|\mathbf{Ax}\|^2 > \text{tr}(\Sigma) + 2\sqrt{\text{tr}(\Sigma^2)t} + 2\|\Sigma\|t\right) \leq e^{-t}. \quad (\text{A.35})$$

### A.3 Existing lemmas

In this subsection, we list existing lemmas from [43, 44] that we use in our proof.

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

**Lemma A.5** ([43, Lemma 19]). *If  $f(\cdot)$  is  $\ell$ -smooth and  $\rho$ -Hessian Lipschitz,  $\eta = 1/\ell$ , then the gradient descent sequence  $\{\mathbf{x}_t\}$  satisfies:*

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq \eta \|\nabla f(\mathbf{x})\|^2, \quad (\text{A.36})$$

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 A.6** ([44, Lemma 7]). *Consider the setting of [Theorem 2.3](#). If  $\|\nabla f(\mathbf{x}_\tau)\| \geq \epsilon$  for all  $\tau \in [0, \mathcal{T}]$ , 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  $E_{\mathcal{T}} - E_0 \leq -\mathcal{E}$ , in which  $\mathcal{E} = \sqrt{\frac{\epsilon^3}{\rho}} \cdot c_A^{-7}$ , and  $E_\tau$  is defined as:*

$$E_\tau := f(\mathbf{x}_\tau) + \frac{1}{2\eta'} \|\mathbf{v}_\tau\|^2 \quad (\text{A.37})$$

where  $\eta' = \frac{1}{4\ell}$  as in [Theorem 2.3](#).

Note that this lemma is not exactly the same as [44, Lemma 7]: to be more specific, they have an extra  $\iota^{-5}$  term appearing in the  $\mathcal{E}$ . 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  $\epsilon$ , it does not make a difference if we ignore this  $\iota^{-5}$  term.

**Lemma A.7** ([44, Lemma 4 and Lemma 5]). *Assume that the function  $f$  is  $\ell$ -smooth. Consider the setting of [Theorem 2.3](#), for every iteration  $\tau$  where quantum simulation was not called, we have:*

$$E_{\tau+1} \leq E_\tau, \quad (\text{A.38})$$

where  $E_\tau$  is defined in (A.37) in [Lemma A.6](#).

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 10](#) of [Algorithm 3](#) will be triggered (same as in [44]) 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.