Quantum Algorithms for Escaping from Saddle Points

arXiv:2007.10253

Joint work with Chenyi Zhang and Jiaqi Leng

Tongyang Li

MIT and Peking University

Quantum Wave in Computing Reunion Workshop, Simons Institute

July 15, 2021

Massachusetts Institute of Technology

Optimization

Problem: $f: \mathbb{R}^n \to \mathbb{R}, \quad \min_x f(x)$

Core topic in math, theoretical computer science, operations research, etc.

Provable guarantee for solving an optimization problem?

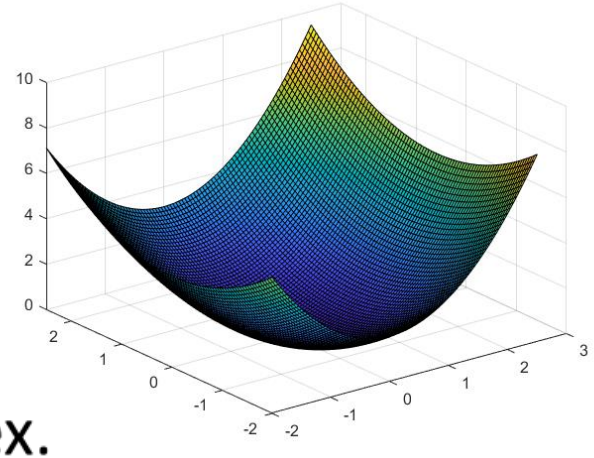
Convex optimization can be solved in polynomial time.
Methods: ellipsoid method, interior point method, etc.

Cost: poly$(n, \log 1/\epsilon)$, state-of-the-art: $\tilde{O}(n^2)$ query and $\tilde{O}(n^3)$ time
[Lee, Sidford, and Vempala]

Quantum: Assume the quantum evaluation oracle $O_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle$,
$\tilde{O}(n)$ query and $\tilde{O}(n^3)$ time [Chakrabarti et al., van Apeldoorn et al.]

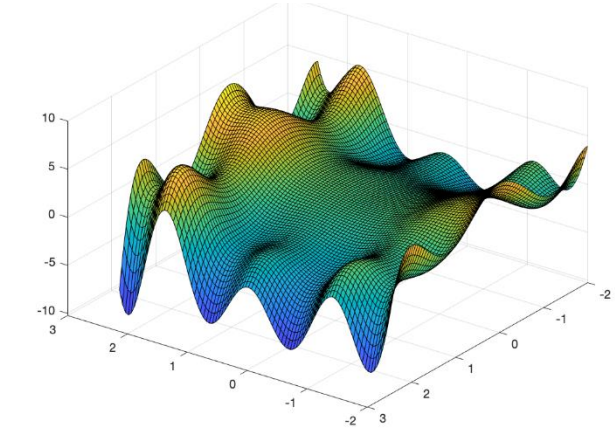
End of the story?

Two concerns about this result in the practice of ML:



Unique local minimum is the global one

- Loss functions of ML models are typically nonconvex.
- Many common cases have large n while can also tolerate reasonably large ϵ.



So many local minima…

Speaking of provable guarantee, maybe we want to pursue algorithms with cost

$$\text{poly}(n, \log 1/\epsilon) \Rightarrow \text{poly}(\log n, 1/\epsilon)$$

Such algorithms are called (almost) dimension-free methods.

Nonconvex optimization

The most common method for nonconvex optimization: **gradient descent (GD)**

$$x_{t+1} = x_t - \eta \cdot \nabla f(x_t).$$

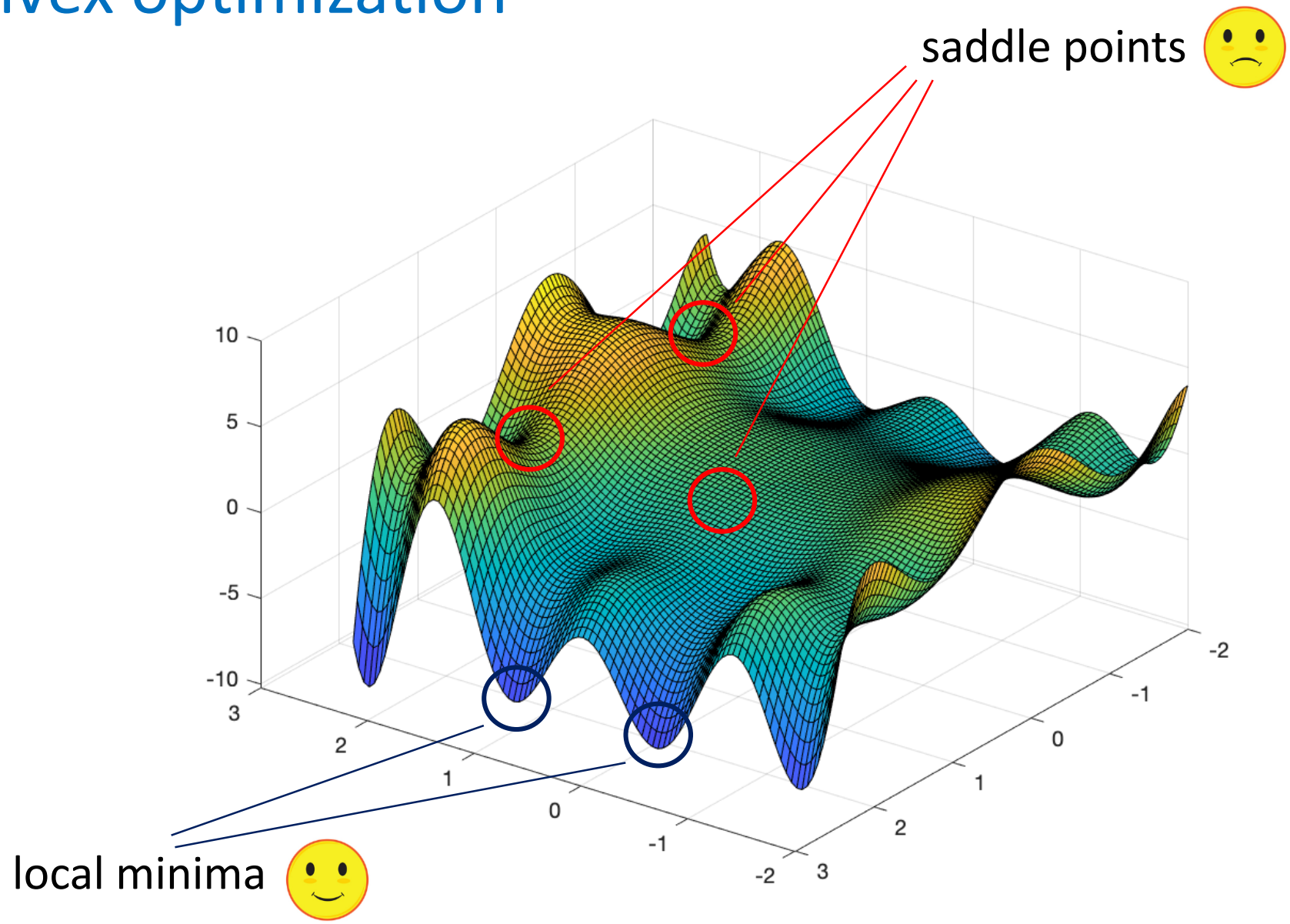
If f is ℓ-smooth: $\|\nabla f(w_1) - \nabla f(w_2)\| \leq \ell \|w_1 - w_2\| \quad \forall\, w_1, w_2 \in \mathbb{R}^n$,

$$t = O(\ell/\epsilon^2) \Rightarrow \|\nabla f(x_t)\| \leq \epsilon.$$

This is an ϵ-approx. first-order stationary point.

Question: Is this good enough?

Nonconvex optimization



Nonconvex optimization

Common facts about many learning problems:

- Local optima are nearly as good as the global minima ("landscape" results in theory);
- Saddle points (and local maxima) can give highly suboptimal solutions;
- Saddle points are ubiquitous;
- Finding the global minima is NP-hard.

Conclusion: We would want to escape from saddle points, but are satisfied with reaching an ϵ-approx. local minimum x_ϵ:

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

Here f is ρ-Hessian Lipschitz: $\|\nabla^2 f(w_1) - \nabla^2 f(w_2)\| \leq \rho\|w_1 - w_2\| \quad \forall\, w_1, w_2 \in \mathbb{R}^n$.

Escaping from saddle points

Reference	Queries	Oracle
Nesterov and Polyak 2006; Curtis et al. 2017	$O(1/\epsilon^{1.5})$	Hessian
Agarwal et al. 2017; Carmon et al. 2018	$\tilde{O}(\log n/\epsilon^{1.75})$	Hessian-vector product
Jin et al. 2017, 2019	$\tilde{O}(\log^4 n/\epsilon^2)$	Gradient
Jin et al. 2018	$\tilde{O}(\log^6 n/\epsilon^{1.75})$	Gradient

Folklore with the Hessian oracle. Issue: n-by-n matrix, too costly in practice

Main consideration: Use simpler and simpler oracle, while keeping poly-log in n

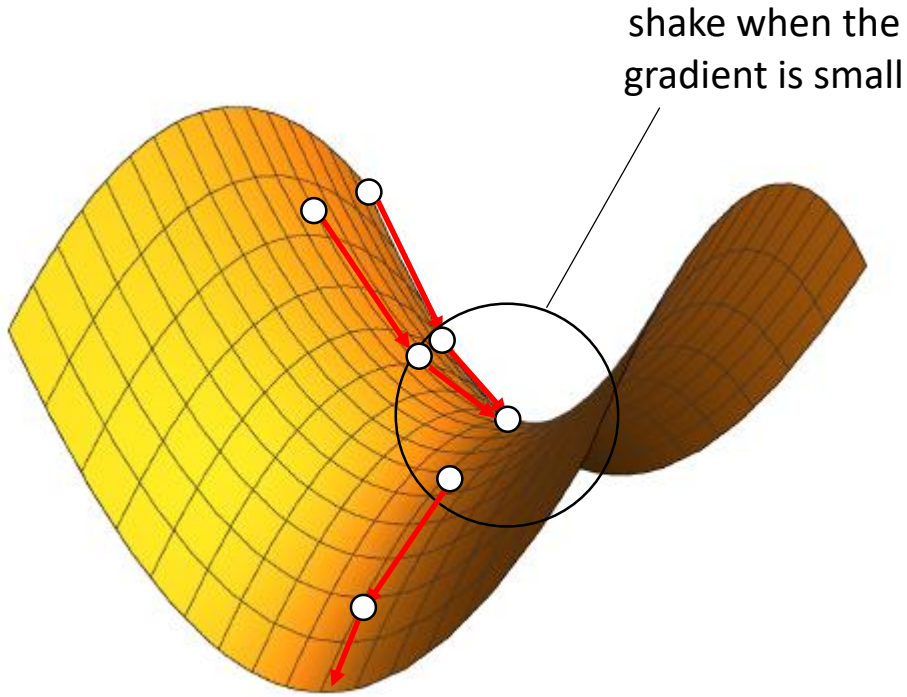
Our result: Using the quantum evaluations $O_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle$,

this work	$\tilde{O}(\log^2 n/\epsilon^{1.75})$	Quantum evaluation

- Cubic quantum speedup in terms of n;
- Using lower order (zeroth-order) information than the classical algorithms.

[Zhang, Leng, L., contributed talk at QIP 2021] arXiv:2007.10253

Escaping from saddle points



The main idea: perturbed gradient descent

Main thoughts:

- **Radius of perturbation:** If it is too large, then we may backtrack too much. If it is too small, we may need many iterations to leave the saddle.
- **Way of perturbation:** What's the most efficient approach?
- **Gradient descent:** Faster versions?

Classical state-of-the-art (simplified)

- Throughout the algorithm, use Nesterov's **accelerated gradient descent** (AGD):

$$y_t \leftarrow x_t + (1-\theta)v_t,\ x_{t+1} \leftarrow y_t - \eta\nabla f(y_t),\ v_{t+1} \leftarrow x_{t+1} - x_t.$$

- If $\|\nabla f(x_t)\| \le \epsilon$ and no perturbation happened in $O(\log n)$ steps:
  Perturb by the **uniform distribution** in the ball of radius $r = \Theta(\epsilon/\log^5 n)$.

Fact: Perturbed AGD takes $O(\log n)$ steps to decrease the the Hamiltonian

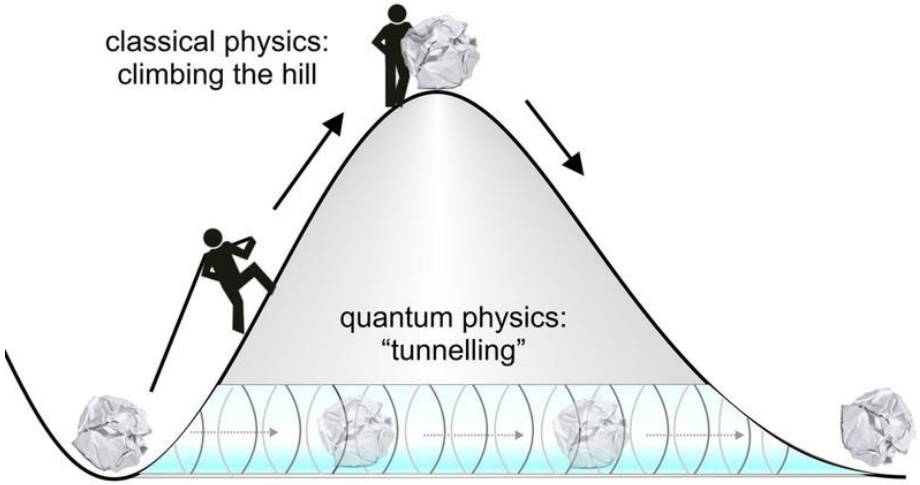
$$f(x_t) + \|v_t\|^2/2\eta$$

by $\Omega(1/\log^5 n)$, convergence rate $O(1/\epsilon^{1.75})$. Total cost: $\tilde{\Theta}(\log^6 n/\epsilon^{1.75})$.

Quantum speedup?

Perturbation plays a crucial role, but the classical radius is too bad. Can we give a quantum speedup for this step? Near saddle points: escape from landscapes.

Quantum tunneling?



We make the idea **algorithmic** using quantum simulation of the Schrödinger equation:

$$i\frac{\partial}{\partial t}\Phi = \Big[-\frac{1}{2}\nabla^2 + f(x)\Big]\Phi.$$

Schrodinger equation

Near a saddle point, the function is well-approximated by a quadratic function.

Lemma

Suppose a quantum particle is in a one-dimensional potential field $f(x) = \frac{\lambda}{2}x^2$ with initial state $\Phi(0, x) = (\frac{1}{2\pi})^{1/4} \exp(-x^2/4)$ (i.e., its initial position follows $\mathcal{N}(0, 1)$). The time evolution of this particle is governed by the Schrödinger equation.

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

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

Fact: Exp. dispersion rate for $\lambda < 0$, quadratic for $\lambda = 0$, and at most constant for $\lambda > 0$.

Escaping from saddle points by quantum simulation

Lemma 2.3. *We have:*

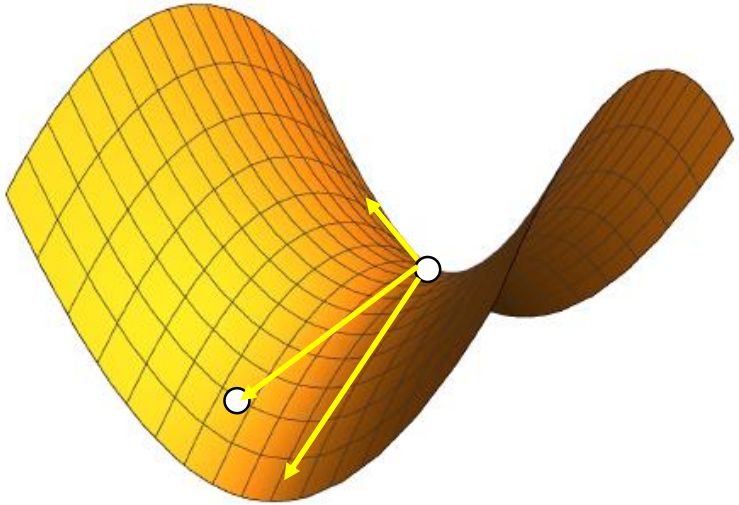
$$\mathbb{P}\Big(\Delta f_\perp \geq \sqrt{\epsilon/\rho^3}\Big) \leq \exp\Big(-\frac{(\log n)^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\Big),$$

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 $\mathscr{T}' = O(l \log n/\sqrt{\rho\epsilon})$. Specifically,

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

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

If we are near a saddle, then at least one eigenvalue is $< -\sqrt{\rho\epsilon}$.



Because the dispersion is so large, even all the other directions go backward (i.e., have positive eigenvalue), with high probability the function value still decreases much.

This is only the perturbation step. How does that combine with GD?

Escaping from saddle points by quantum simulation

Proposition 2.1. *Suppose $f: \mathbb{R}^n \to \mathbb{R}$ is ℓ-smooth and ρ-Hessian Lipschitz. We take*

$$\iota := \Theta(\log(n\ell(f(\mathbf{x_0}) - f^*)/(\rho\epsilon\delta))), \quad \mathscr{T}' := \frac{\ell}{\sqrt{\rho\epsilon}} \cdot \iota, \quad \mathscr{F}' := \sqrt{\frac{\epsilon}{\rho^3}}$$

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:

$$\mathbb{P}\big(f(\mathbf{x}_{\mathscr{T}'}) - f(\tilde{\mathbf{x}}) \leq -\mathscr{F}'\big) \geq 1 - \delta,$$

where $\mathbf{x}_{\mathscr{T}'}$ is the $\mathscr{T}'^{th}$ GD iteration starting from $\mathbf{x}_0$, if Algorithm 1 was called at $t = 0$ in Line 4.

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:

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

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.

Algorithm 2: PGD with Quantum Simulation.

1 $t_{\text{perturb}} = 0$;
2 **for** $t = 0, 1, ..., T$ **do**
3 **if** $\|\nabla f(\mathbf{x}_t)\| \leq \epsilon$ *and* $t - t_{perturb} > \mathscr{T}'$ **then**
4 $\mathbf{x}_t \leftarrow \mathbf{x}_t - \eta\xi_t$,
   where $\xi_t \sim$QuantumSimulation$(\mathbf{x}_t, r_0, \mathscr{T}')$;
5 $t_{\text{perturb}} = t$;
6 $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta\nabla f(\mathbf{x}_t)$;

Query complexity of simulating the Schrodinger equation

Remain question 1: what's the cost of the quantum simulation?

We follow closely to the quantum PDE solver of Childs et al. (arXiv:2002.07868)

If we discretize the space into grids with side-length a (Childs et al.: a can be taken as $\mathrm{poly}(n, \log 1/\epsilon)$), $-\frac{1}{2}\nabla^2$ reduces to $-\frac{1}{2a^2}L$ where L is the Laplacian matrix of the graph of the grids:

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

$$i\frac{\partial}{\partial t}\Phi = \Big(-\frac{1}{2}\nabla^2 + f(\mathbf{x})\Big)\Phi \quad \Rightarrow \quad i\frac{\mathrm{d}}{\mathrm{d}t}\Phi = \Big(-\frac{1}{2a^2}L + B\Big)\Phi,$$

where B is a diagonal matrix such that the entry for the grid at $\mathbf{x}$ is $f(\mathbf{x})$.

Very special structure: $\frac{1}{2a^2}L$ is dominating but independent of f;
B depends on f but is diagonal, very easy to simulate.

Our strategy: Use quantum simulation under the interaction picture (Low and Wiebe, arXiv:1805.00675).

Overall quantum query complexity: $\tilde{O}\big(t \log n \log^2(\frac{t}{\epsilon})\big)$ for time t.

Quadratic approximation

Remain question 2: is the quadratic approximation a too strong assumption?

No, the general case can be approximated by the quadratic case with small overhead.

- Jean Bourgain's result on the growth of Sobolev norms in linear Schrodinger equations: For Schrödinger equations of the form $i\frac{\partial}{\partial t}u + \nabla^2 u + V(x,t)u = 0$ with periodic boundary condition, we have absolute constants C and α such that

$$\|\nabla u(t)\|_2 \leq C(\log t)^\alpha \|\nabla u(0)\|_2.$$

Lemma 3. *Let $\mathcal{H}$ be the Hessian of f at a saddle point $\tilde{\mathbf{x}}$, and $f_q(\mathbf{x}) := f(\tilde{\mathbf{x}}) + \frac{1}{2}(\mathbf{x}-\tilde{\mathbf{x}})^T\mathcal{H}(\mathbf{x}-\tilde{\mathbf{x}})$ to be the quadratic approximation of f near $\tilde{\mathbf{x}}$. Denote the measurement outcome from the quantum simulation on a hypercube with edge length M with potential field f (resp. f_q) and evolution time t_e as a random variable with distribution $\mathbb{P}_\xi$ (resp. $\mathbb{P}_{\xi'}$). Then*

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

Putting everything together

Algorithm 3: Perturbed Accelerated Gradient Descent with Quantum Simulation.

$\mathbf{v}_0 \leftarrow 0$;
for $t = 0, 1, \ldots, T$ **do**
  if $\|\nabla f(\mathbf{x}_t)\| \le \epsilon$ *and no perturbation in last* $\mathscr{T}'$ *steps* **then**
    $\mathbf{x}_t \leftarrow \mathbf{x}_t + \xi_t \qquad \xi_t \sim \text{QuantumSimulation}(\mathbf{x}_t, r_0, \mathscr{T}')$;
  if *a perturbation was added in last* $\mathscr{T}'$ *steps* **then**
    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta \nabla f(\mathbf{x}_t)$ and $\mathbf{v}_{t+1} = 0$;
  else
    $\mathbf{y}_t \leftarrow \mathbf{x}_t + (1-\theta)\mathbf{v}_t$, $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t - \eta' f(\mathbf{y}_t)$, and $\mathbf{v}_{t+1} \leftarrow \mathbf{x}_{t+1} - \mathbf{x}_t$;
    if $f(\mathbf{x}_t) \le f(\mathbf{y}_t) + \langle \nabla f(\mathbf{y}_t), \mathbf{x}_t - \mathbf{y}_t \rangle - \frac{\gamma}{2}\|\mathbf{x}_t - \mathbf{y}_t\|$ **then**
      $(\mathbf{x}_{t+1}, \mathbf{v}_{t+1}) \leftarrow$Negative-Curvature-Exploitation$(\mathbf{x}_t, \mathbf{v}_t, s)$;

Theorem 2.3 (informal). *Algorithm 3 gives an ϵ-approximate local minimum using*

$$\tilde{O}\Big(\frac{(f(\mathbf{x}_0) - f^*)}{\epsilon^{1.75}} \cdot \iota^2\Big)$$

queries to U_f and gradients with probability at least $\ge 1-\delta$, where $\iota = \Theta(\log(n\ell(f(\mathbf{x_0}) - f^*)/(\rho\epsilon\delta)))$, $\epsilon, \delta \in (0,1)$, $\mathbf{x}_0$ is the start point, f is ℓ-smooth and ρ-Hessian Lipschitz, and f^* is the global minimum of f.

Summary

1. When the gradient is large, apply Accelerated GD;
2. When the gradient is small, run quantum simulation with time $O(\log n)$ and measure the outcome;
3. Apply GD for $O(\log n)$ iterations, and go to Step 1 or 2 (depending on the norm of the gradient).

	Perturbation	# iterations/ simulation time	Function decrease	Queries in each iteration/unit time
Classical	Uniform in ball	$O(\log n)$	$\Omega(1/\log^5 n)$	1
Quantum	Quantum simulation	$O(\log n)$	$\Omega(1)$	$\tilde{O}(\log n)$

A comparison between classical and quantum state-of-the-arts, assuming $\epsilon = \Theta(1)$.

Gradient computation by quantum evaluation oracle

Next: Can we simplify classical gradient queries?

Our contribution: Generalize the application of Jordan's algorithm for convex optimization to nonconvex optimization.

The state $e^{if(x)}|x\rangle$ can be prepared by one query to the oracle $O_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle$.

First-order Taylor approximation: $f(x) \approx \sum_{k=1}^{n} \frac{\partial f}{\partial x_k} x_k$;

$$\sum_x e^{if(x)}|x\rangle \approx \sum_x \bigotimes_{k=1}^{n} e^{i\frac{\partial f}{\partial x_k}x_k}|x_k\rangle.$$

Applying the quantum Fourier transform on all n coordinates reveals $\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}$.

Intuitively: Continuous version of the Bernstein-Vazirani algorithm

Gradient computation by quantum evaluation oracle

Our contribution: Generalize the application of Jordan's algorithm for convex optimization to nonconvex optimization.

Lemma 3.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be ℓ-smooth, ρ-Hessian Lipschitz, and let $\eta \le 1/\ell$. Then the gradient outputted by Jordan's algorithm 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:*

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

where $A_q = 400n\sqrt{\delta_q \ell}$ and δ_q stands for the accuracy of the quantum evaluation oracle, i.e., it returns a value $\tilde{f}(x)$ such that $|\tilde{f}(x) - f(x)| \le \delta_q$.

Robust version of function decrease when escaping from saddle points:

Lemma 3.5. *Under the setting of the above lemma, we have:*

$$\mathbb{P}\Big(\Delta f'_\perp \ge \frac{5\mathscr{F}'}{8}\Big) \le \exp\Big(-\frac{\iota^6}{(\rho\epsilon)^{3/2}(1+\ell^2)}\Big) + \mathscr{T}' \cdot \frac{n}{\frac{1}{800n\sqrt{\delta_q}}\sqrt{\frac{\mathscr{F}'}{\mathscr{T}'}} - 1}.$$

Final result

Theorem 3.1 (informal). *Suppose that we have the quantum evaluation oracle U_f with accuracy $\delta_q \leq O\left(\frac{\epsilon^7\delta^2}{n^4\iota^5} \cdot 2^{-\iota}\right)$. Then using Jordan's algorithm for computing the gradients in descent steps, the same query bound for giving an ϵ-local minimum holds with probability at least $1-\delta$.*

Theorem 2.3 (informal). *Algorithm 3 gives an ϵ-approximate local minimum using*

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

queries to U_f [redacted] *with probability at least $\geq 1-\delta$, where $\iota = \Theta(\log(n\ell(f(\mathbf{x_0})-f^*)/(\rho\epsilon\delta)))$, $\epsilon, \delta \in (0,1)$, $\mathbf{x}_0$ is the start point, f is ℓ-smooth and ρ-Hessian Lipschitz, and f^* is the global minimum of f.*

We essentially show the robustness of escaping from saddle points by PGD, which may be of independent interest.

Numerical experiments

Quantum simulation on non-quadratic potential fields

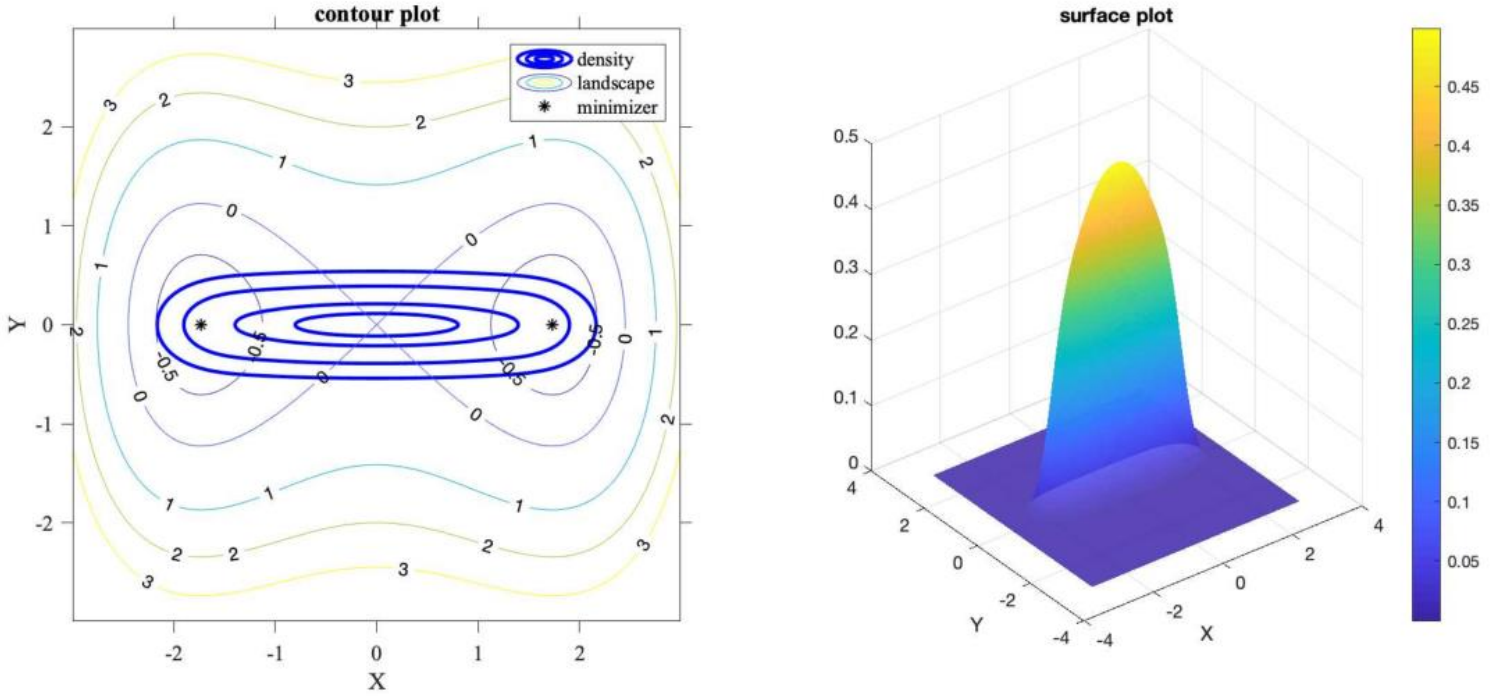


Figure 2: Quantum simulation on lanscape 1: $f(x, y) = \frac{1}{12}x^4 - \frac{1}{2}x^2 + \frac{1}{2}y^2$. Parameters: $r_0 = 0.5$, $t_e = 1.5$. **Left**: The contour of the landscape is placed on the background with labels being function values; the thick blue contours illustrate the wave packet at $t_e = 1.5$ (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 wave packet has been "squeezed" along the x-axis, the negative curvature direction. Compared to the uniform distribution in a ball used in PGD, this "squeezed" bivariant Gaussian distribution assigns more probability mass along the x-axis, thus allowing escaping from the saddle point more efficiently.

Numerical experiments

Quantum simulation on non-quadratic potential fields

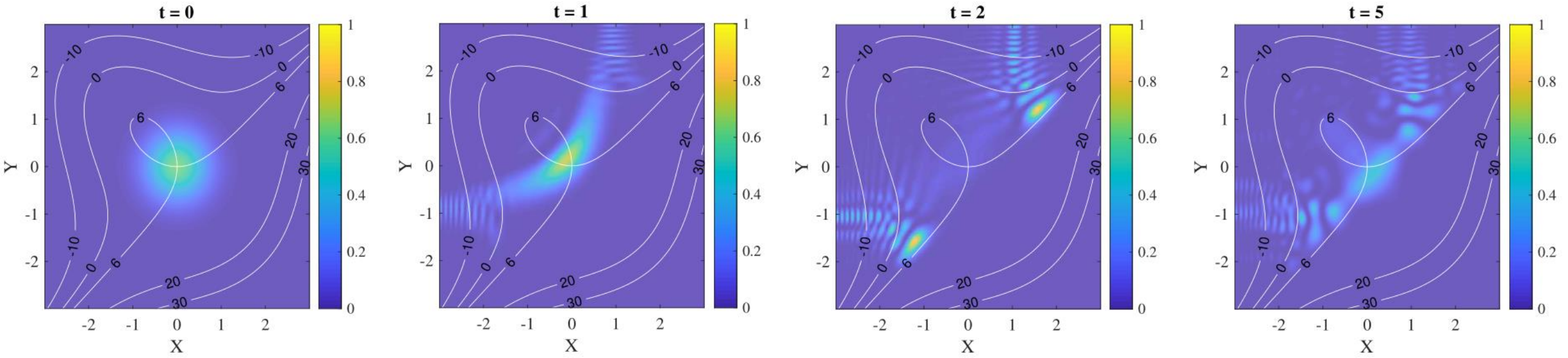


Figure 3: Quantum simulation on lanscape 2: $g(x, y) = x^3 - y^3 - 2xy + 6$. Parameters: $r_0 = 0.5$, $t_e = 5$. It has a saddle point at $(0, 0)$ with no global minimum. This objective function has a circular "valley" along the negative curvature direction $(1, 1)$, and a "ridge" along the positive curvature direction $(1, -1)$. In each subplot, a colored contour plot of the wave packet at a specific time is shown.

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). Nevertheless, an interference pattern can be observed near the upper and left edges of the square, which suggests that Gaussian wave packet is able to adapt to more complicated saddle point geometries.

Numerical experiments

Comparison between classical and quantum algorithms

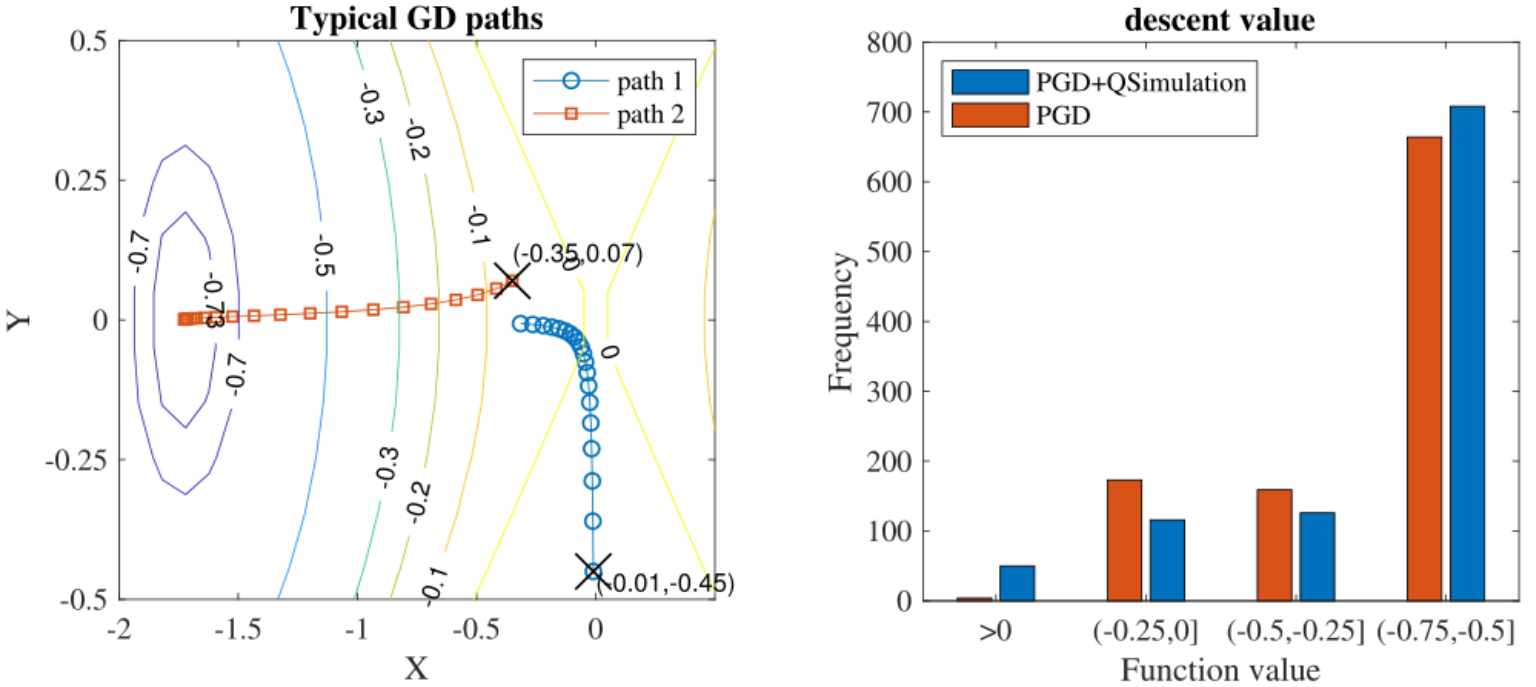


Figure 4: **Left**: Two typical gradient descent paths on the landscape of $f_2 = \frac{1}{12}x^4 - \frac{1}{2}x^2 + \frac{1}{2}y^2$ illustrated as a contour plot. Path 1 (resp. 2) starts from $(-0.01, 0.45)$ (resp. $(-0.35, 0.07)$); both have step length $\eta = 0.2$ and $T = 20$ iterations. Note that path 2 approaches the local minimum $(-\sqrt{3}, 0)$, while path 1 is still far away. Classically, path 1 and 2 will be sampled with equal probability. Quantumly, the dispersion of the wave packet along the x-axis enables a much higher probability of sampling a path like path 2.

Right: A histogram of function values from classical and quantum algorithms. We set step length $\eta = 0.05$, $r = 0.5$, $M = 1000$, $\mathscr{T}_c = 50$, $\mathscr{T}_q = 10$, $t_e = 1.5$. Although we run five more times of iterations in PGD, there are still over 70% of gradient descent paths arriving the neighborhood of the local minimum, while there are less than 70% paths in PGD+QSimulation approaching the local minimum.

Numerical experiments

Dimension dependence

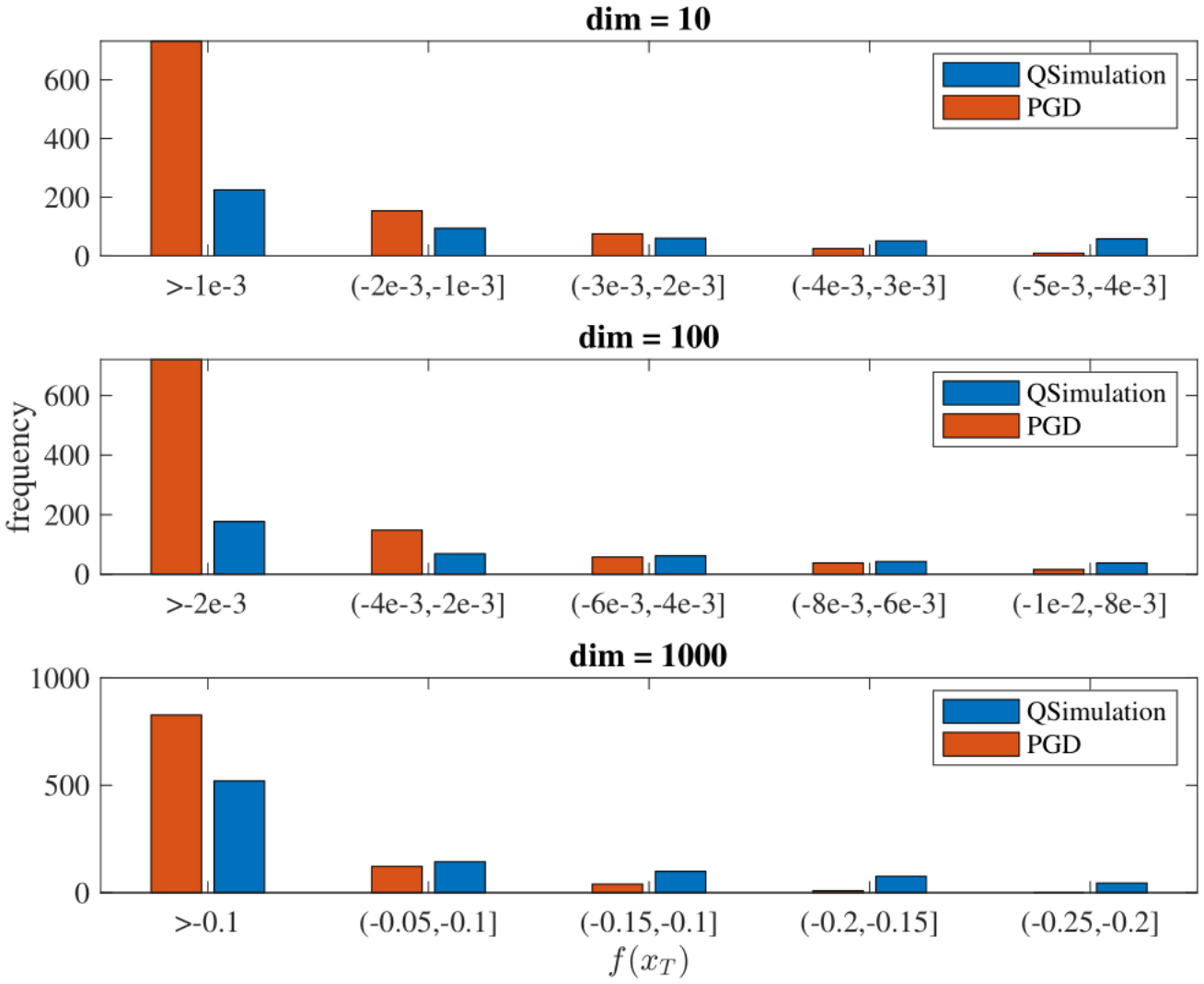


Figure 5: Dimension dependence of classical and quantum algorithms. We set $\epsilon = 0.01$, $r = 0.1$, $n = 10^p$ for $p = 1, 2, 3$. Quantum evolution time $t_e = p$, classical iteration number $\mathscr{T}_c = 50p^2+50$, quantum iteration number $\mathscr{T}_q = 30p$, and sample size $M = 1000$. The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

This numerical evidence might suggest that for a generic problem, the classical PGD method has better dimension dependence than $O(\log^4 n)$.

Conclusions

Main result: A quantum algorithm for ϵ-approx. local minimum by $\tilde{O}\left(\frac{\log^2 n}{\epsilon^{1.75}}\right)$ queries.

Outcome: Cubic quantum speedup in *n*, match the classical best-known in ϵ.

- Achieve speedup by using quantum simulation to escape from saddle points;
- Reduce classical gradients to quantum evaluations by Jordan's algorithm.

Open questions:

- Can we give quantum-inspired classical algorithms for escaping from saddle points?
- Can quantum algorithms achieve speedup in terms of $1/\epsilon$?
- Beyond local minima, does quantum provide advantage for approaching global minima?