A Dynamical Point of View on Quantum Optimization

Jiaqi (Jimmy) Leng

Applied Mathematics & Statistics, and Scientific Computing (AMSC) Program
Joint Center for Quantum Information & Computer Science
University of Maryland, College Park

"Quantum Hamiltonian Descent", arXiv:2303.01471
"A quantum-classical performance separation in nonconvex optimization", manuscript in preparation

UNIVERSITY OF MARYLAND

JOINT CENTER FOR QUANTUM INFORMATION AND COMPUTER SCIENCE

Quantum computing 101

Classical (digital) Computer

- Unit of information: bit ($b \in \{0,1\}$)
- Classical information → bitstrings (e.g., 10001001)
- Computation = manipulation of finite-length bitstrings
- Readout/Return: bitstrings

Quantum Computer

- Unit of information: quantum bit, or **qubit**

$$|\psi> = a|0> + b|1>, a, b \in \mathbb{C}, |a|^2 + |b|^2 = 1$$

- Quantum information → quantum state (i.e., superposition of |*bitstring*>)

$$|GHZ> = \frac{1}{\sqrt{2}}\left(|000> + |111>\right)$$

- Computation = unitary (i.e., preserving 2-norm) operations on quantum states
- Readout: quantum measurement → a sample of bitstrings

Measure GHZ state:
000: 50%, 111: 50%, others: 0%

Digital Quantum Computers

Use quantum gates & quantum circuits, not ready in 5-10 years. Must do error correction.

Analog Quantum Computers

Solving problems by emulating a real quantum system. Ready & easy to scale. No error correction.

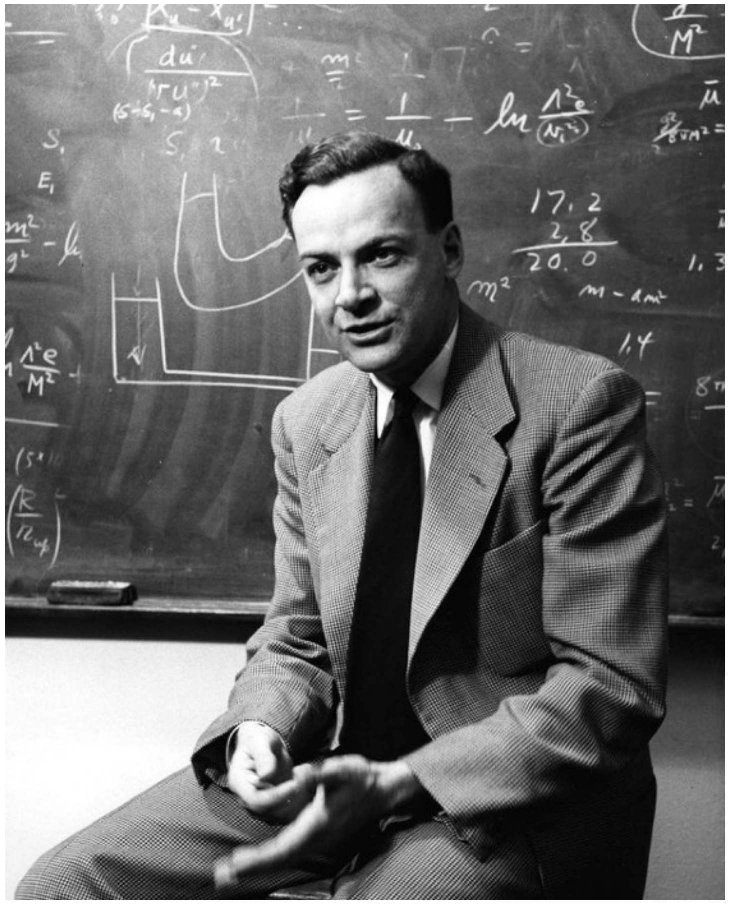
Hamiltonian simulation

- **Quantum simulation** (or **Hamiltonian simulation**): a prominent application of quantum computers. Exponential quantum speedup expected.

$$i\frac{\partial}{\partial t}\psi = \hat{H}(t)\psi$$

$\hat{H}(t)$: *quantum Hamiltonian (Hermitian/self-adjoint)*

- **Feynman** (*“Simulating Physics with Computers”, 1982*): to simulate quantum systems, we would need to build quantum computers.



- **Applications**: quantum chemistry, quantum field theory, condensed matter physics, numerical optimization, etc.

Overview

- Part I: Quantum Hamiltonian Descent (QHD)
- Part II: A Quantum-Classical Performance Separation
- Part III: Implementation on Quantum Computers

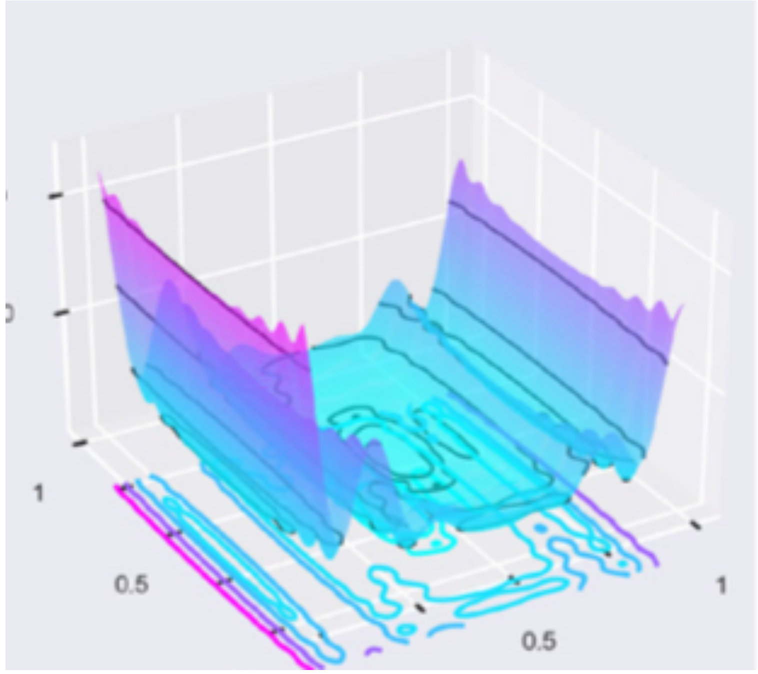
Part I:

Quantum Hamiltonian Descent

Problem formulation

Continuous optimization

$$\boxed{\min_{x \in \mathbb{R}^d} f(x)}$$



- Important in practice: machine learning, operations research, scientific computing, etc.
- A **challenging problem** for quantum: different nature in (quantum) algorithm design, requires new mathematical tools to prove convergence.
- **Opportunity**: new *primitives* of quantum speedups!

Classical algorithms: Gradient Descent (GD)

- Standard GD: $x_{k+1} = x_k - s\nabla f(x_k)$.
- Nesterov's accelerated GD: $x_{k+1} = y_k - s\nabla f(y_k)$, $y_{k+1} = x_{k+1} + \dfrac{k}{k+3}\left(x_{k+1} - x_k\right)$.

A Lagrangian formulation of accelerated methods

Su, Boyd, & Candes
NeurIPS 2014

$$\boxed{\begin{aligned} x_k &= y_{k-1} - s\nabla f(y_{k-1}), \\ y_k &= x_k + \frac{k-1}{k+2}(x_k - x_{k-1}). \end{aligned}} \longrightarrow \boxed{\ddot{X} + \frac{3}{t}\dot{X} + \nabla f(X) = 0,}$$

Nesterov's accelerated GD

ODE model

Wibisono, Wilson, & Jordan
PNAS Nov 2016, 113 (47)

$$\boxed{\begin{aligned} x_k &= y_{k-1} - s\nabla f(y_{k-1}), \\ y_k &= x_k + \frac{k-1}{k+2}(x_k - x_{k-1}). \end{aligned}} \longrightarrow$$

Nesterov's accelerated GD

$$\boxed{\mathcal{L}(t, \dot{X}, X) = t^3\left(\frac{1}{2}|\dot{X}|^2 - f(X)\right)}$$

Lagrangian formulation

$$\boxed{H(t, X, P) = \frac{1}{2t^3}|P|^2 + t^3 f(X)}$$

Hamiltonian formulation

- Accelerated GD can be modeled by accelerated gradient flows.
- GD algorithms can be generated by discretizing continuous-time dynamics.

Accelerated Hamiltonian flows: classical v.s. quantum

We can make the classical dynamics quantum!!

Classical Hamiltonian Systems

Path integral formulation of Quantum Mechanics

Due to Feynman

Quantum Hamiltonian Evolution

$$\boxed{H(t,X,P) = \frac{1}{2t^3}|P|^2 + t^3 f(X)}$$

$$\boxed{\hat{H}(t) = \frac{1}{t^3}\left(-\frac{1}{2}\Delta\right) + t^3 f(x)}$$

Bregman Lagrangian

$$\mathcal{L}(X,\dot{X},t) = e^{-\varphi_t}\left(\frac{1}{2}|\dot{X}|^2\right) - e^{\chi_t} f(X)$$

Quantum propagator

$$K(b,t_b;a,t_a) \propto \sum_{\text{paths from a to b}} e^{\frac{i}{\hbar}S[X(t)]}$$

Infinitesimal expansion

$$\Psi(x,t+\epsilon) \propto \int \exp\left\{\frac{i}{\hbar}\epsilon\mathcal{L}\left(\frac{x+y}{2},\frac{x-y}{\epsilon},t\right)\right\}\Psi(y,t)\,dy$$

Schrodinger equation (QHD)

$$i\hbar\frac{\partial\Psi}{\partial t} = \left[e^{\varphi_t}\left(-\frac{\hbar^2}{2}\nabla^2\right) + e^{\chi_t} f(x)\right]\Psi$$

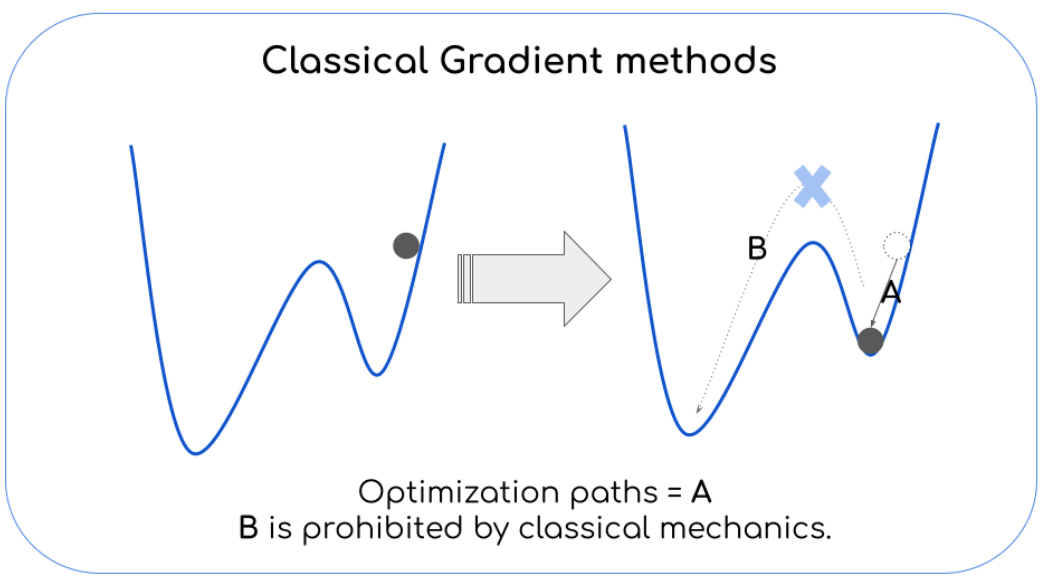
Quantum Hamiltonian Descent

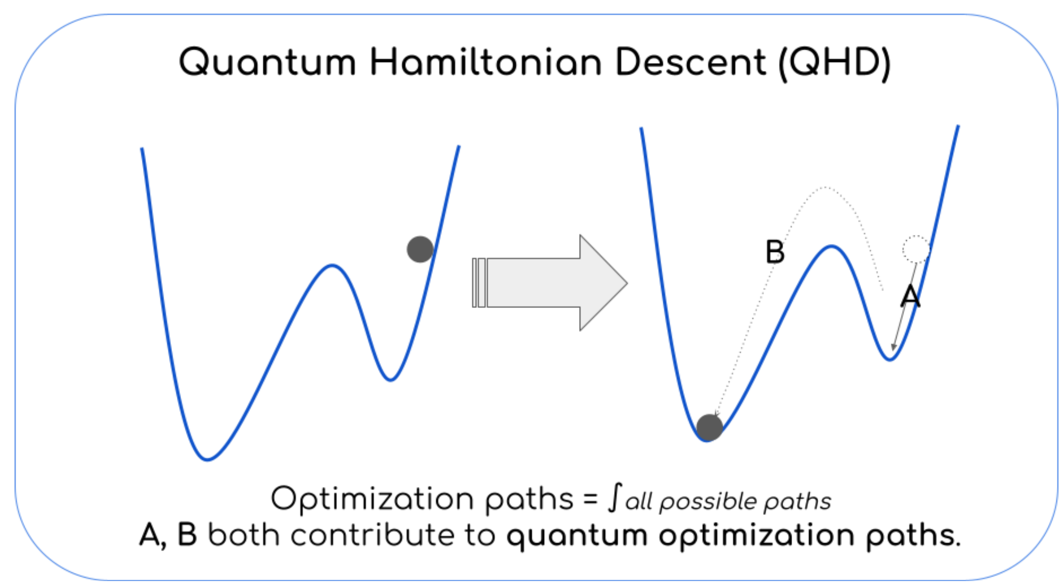
$$H(t, X, P) = e^{\varphi_t}\left(\frac{1}{2}|P|^2\right) + e^{\chi_t} f(X) \longrightarrow \hat{H}(t) = e^{\varphi_t}\left(-\frac{1}{2}\nabla^2\right) + e^{\chi_t} f(x).$$

Accelerated gradient flows

Quantum Hamiltonian Descent (QHD)

- This is a quantum dynamical system with *damping* ($e^{\varphi_t - \chi_t} \to 0$).
- The energy damping allows the quantum system to *converge* to a low-energy configuration (i.e., minimizing the objective f).
- **Intuition**: "Path integral" of classical GD → does it help **nonconvex** optimization?



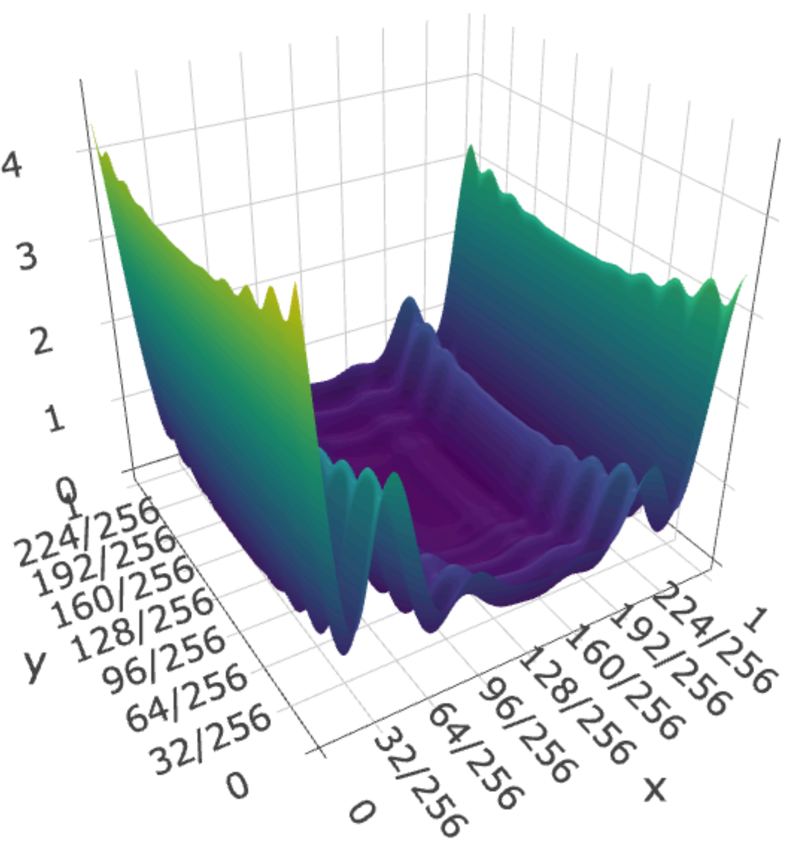


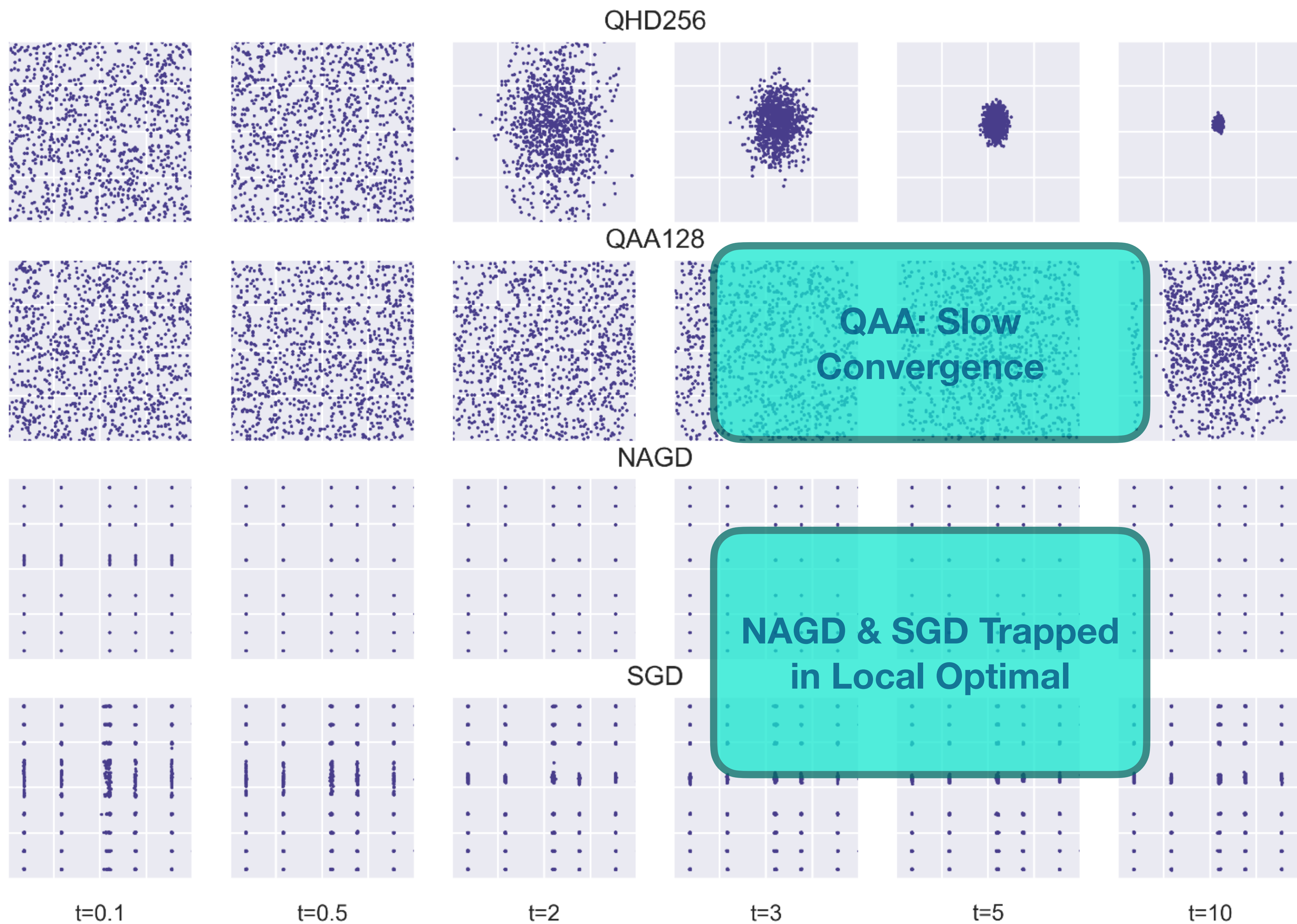
Quantum Hamiltonian Descent - numerical example

QAA: quantum adiabatic algorithm. **NAGD:** Nesterov's accelerated gradient descent.

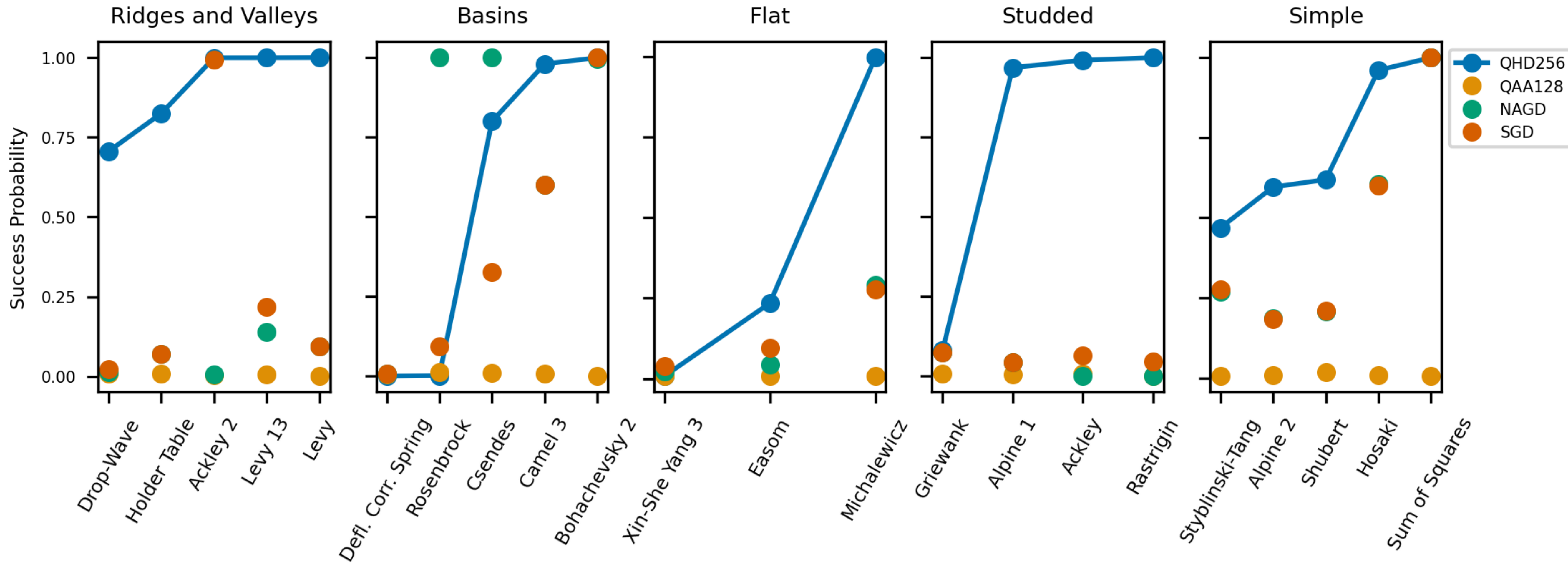
SGD: stochastic gradient descent.

QHD showcase:
the Levy function.





Behavior of QHD on 2D benchmark



Measure in Success Probability for some given time:

the probability of generating a solution within radius r of the optimal solution from random initial points.

QAA: use standard *linear interpolation* in the adiabatic method (i.e., $H_T(t) = (1 - t/T)H_0 + (t/T)H_1$) and simulate $H_T(t)$ for T=10 over a **128-by-128** grid on $[0,1] \times [0,1]$.

QHD: simulate QHD Hamiltonian with Nesterov's AGD parameters (i.e., $H(t) = -\frac{1}{2t^3}\nabla^2 + t^3 f(x)$) from $t = 0$ to 10 over a **256-by-256** grid on $[0,1] \times [0,1]$.

Convergence guarantee: convex optimization

$$H(t,P,X) = e^{\alpha_t}\left(-\frac{e^{-\gamma_t}}{2}P^2 + e^{\gamma_t+\beta_t}f(X)\right) \longrightarrow \frac{d}{dt}\left(e^{-\alpha_t}\dot{X}_t + X_t\right) = e^{\alpha_t+\beta_t}\nabla f(X_t)$$

Theorem [Wibisono, Wilson, Jordan (2016)]

Assume $f(x)$ is continuously differentiable and convex, and $\dot{\beta}_t \leq e^{\alpha_t}$, $\dot{\gamma}_t = e^{\alpha_t}$ (aka, *ideal scaling condition*). Then, the solution $X(t)$ satisfies

$$f(X_t) - f(x^*) \leq O(e^{-\beta_t}).$$

$$\hat{H}(t) = e^{\alpha_t}\left(-\frac{e^{-\gamma_t}}{2}\Delta + e^{\gamma_t+\beta_t}f(x)\right) \longrightarrow \frac{\partial}{\partial t}\Psi(t,x) = \hat{H}(t)\Psi(t,x)$$

Theorem [Leng, Hickman, Li, Wu (2023)]

Assume $f(x)$ is continuously differentiable and convex, and $\dot{\beta}_t \leq e^{\alpha_t}$, $\dot{\gamma}_t = e^{\alpha_t}$ (aka, ideal scaling condition). Define $\mathbb{E}[f]_{\sim\Psi(t)} = \langle\Psi(t)|f|\Psi(t)\rangle = \int f|\Psi(t)|^2\,dx$, then we have

$$\mathbb{E}[f]_{\sim\Psi(t)} - f(x^*) \leq O(e^{-\beta_t}).$$

A Lyapunov function approach

<u>Theorem [Leng, Hickman, Li, Wu (2023)]</u>

Assume $f(x)$ is continuously differentiable and convex, and $\dot{\beta}_t \leq e^{\alpha_t}$, $\dot{\gamma}_t = e^{\alpha_t}$ (aka, ideal scaling condition). Define $\mathbb{E}[f]_{\sim\Psi(t)} = \langle \Psi(t) | f | \Psi(t) \rangle = \int f |\Psi(t)|^2 \, dx$, then we have

$$\mathbb{E}[f]_{\sim\Psi(t)} - f(x^*) \leq O(e^{-\beta_t}).$$

- We construct a Lyapunov function ($< O >_t = < \Psi_t | O | \Psi_t >$, $\hat{p} = -i\nabla$):

$$\mathcal{W}(t) = \langle \hat{J}^2/2 \rangle_t + e^{\beta_t} \langle f \rangle_t$$

$$\hat{J} := e^{-\gamma_t} \hat{p} + \hat{x}.$$

- We can prove this Lyapunov function is **non-increasing in t**.
- Therefore, we have $e^{\beta_t} \mathbb{E}[f]_{\sim\Psi_t} \leq \mathcal{W}(t) \leq \mathcal{W}(0), \quad \mathbb{E}[f]_{\sim\Psi_t} \leq \mathcal{W}(0) e^{-\beta_t} \leq O(e^{-\beta_t}).$

- A sanity check: we have the same converge rate as classical [WWJ16].
- Nesterov's GD: $e^{\alpha_t} = 2/t$, $e^{\beta_t} = e^{\gamma_t} = t^2$, implies the convergence rate $O(t^{-2})$.
- QHD is more stable & robust with larger time discretization steps.

- <u>Question: can we achieve $O(e^{-\sqrt{\beta} t})$ convergence rate for β-strongly convex f ?</u>

Convergence guarantee: nonconvex optimization

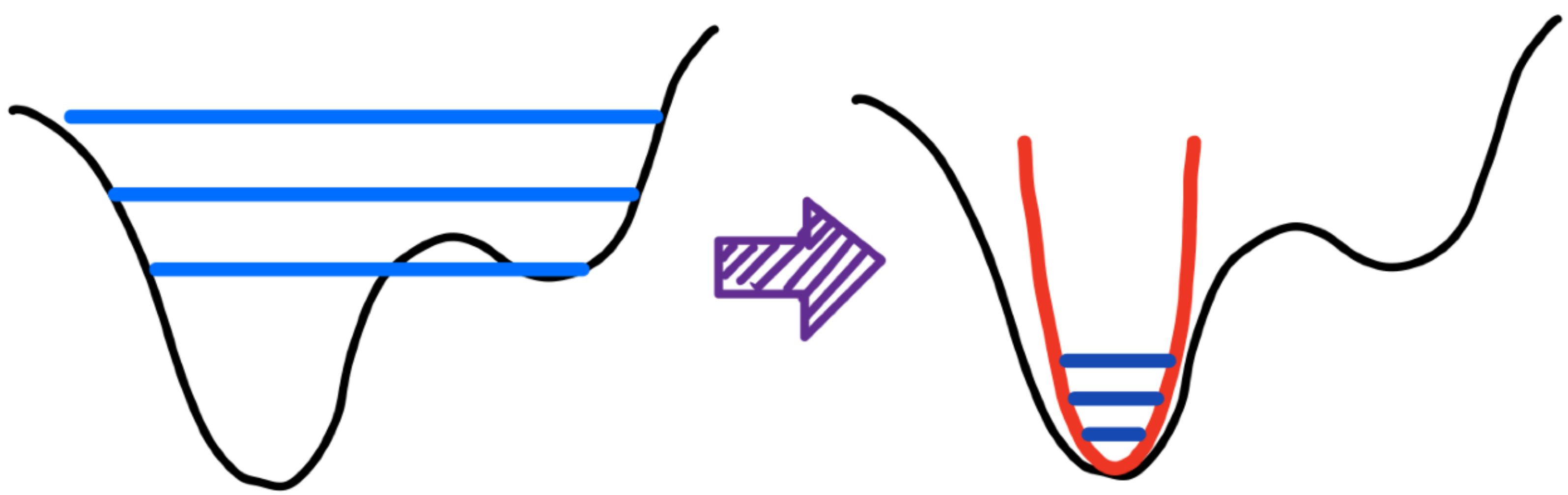
$$\hat{H}(t) = \left(-\frac{e^{\varphi_t}}{2}\Delta + e^{\chi_t} f(x)\right) \longrightarrow \frac{\partial}{\partial t}\Psi(t,x) = \hat{H}(t)\Psi(t,x)$$

Theorem (informal) [Leng, Hickman, Li , Wu (2023)]

Suppose $f(x)$ be smooth, unbounded at infinity, and has a unique non-degenerate global minimum x^*. Let the initial wave Ψ_0 be in the low-energy subspace of $H(0)$ and $e^{\varphi_t - \chi_t} \to 0$, $|\dot{\varphi}_t|, |\dot{\chi}_t| \ll 1$ (i.e., $H(t)$ is slow-varying), then

$$\lim_{t\to\infty} \mathbb{E}[f]_{\sim\Psi_t} = f(x^*)\,.$$

- Slow-varying $H(t)$ -> the wave function $\Psi(t)$ stays in the low-energy subspace for all t.
- The low-energy subspace of $H(t)$ will *migrate* to the global minimum of f as $t \to \infty$.



Quantum Hamiltonian Descent: complexity analysis

1. Prepare an initial state $|\psi_0>$.
2. Simulating the Schrodinger equation: $i\partial_t \psi = \hat{H}(t)\psi,\ \psi(0) = \psi_0$.
3. At time t = T, *measure* the final state $|\Psi_T>$ (i.e., to sample from the corresponding distribution $|\Psi(T)|^2$).
4. Ideally, the measurement results will cluster around the global minimizer of f.

$$\hat{H}(t) = e^{\varphi_t}\left(-\frac{1}{2}\nabla^2\right) + e^{\chi_t} f(x)$$

Runtime of QHD

QHD is comparable to GD in terms of simplicity and resource cost:

- Running QHD = time-dependent Hamiltonian simulation [Childs, Leng, Li, Liu & Zhang, 2022]
- The runtime of QHD is $\tilde{O}(dT)$ (d: dimension of f , T: total evolution time).
- Resource cost (on quantum computer) is comparable to that of classical GD.
- Resource cost (on classical computer) is exponential in d.

Therefore, *QHD is a quantum-upgraded version of GD*. We will demonstrate how powerful QHD could be by itself. One could also build on top of QHD like what we've done with GD.

QHD as accelerated Wasserstein gradient flow

Let's consider a simple QHD evolution:

$$\boxed{i\frac{\partial}{\partial t}\psi(t,x) = \left[e^{-\alpha_t}\left(-\frac{1}{2}\Delta\right) + e^{\alpha_t} f\right]\psi(t,x)}$$

We apply the Madelung transform: $\psi(t,x) = \sqrt{\rho(t,x)}e^{iS(t,x)}$

$$\boxed{\begin{cases} \partial_t \rho_t + e^{-\alpha_t}\nabla\cdot(\rho_t \nabla S_t) = 0, \\ \partial_t S_t + e^{-\alpha_t}\left[\frac{1}{2}(\nabla S_t)^2 + \frac{1}{8}\frac{\delta \mathscr{F}(\rho_t)}{\delta \rho_t}\right] + e^{\alpha_t} f(x) = 0 \end{cases}}$$

$\mathscr{F}(\rho) = \int |\nabla \log \rho|^2 \rho$ is the *Fisher information* functional

The above system of PDEs is the Euler-Lagrange equation generated by:

$$\boxed{\mathcal{L}(\rho,\sigma,t) = e^{\alpha_t}\left(\frac{1}{2}g_W(\sigma,\sigma) - \int f\rho\,\mathrm{d}x\right) - e^{-\alpha_t}\left(\frac{1}{8}\mathscr{F}(\rho)\right)}$$

- QHD = accelerated Wasserstein gradient flow with *Fisher regularization*
- This explains why QHD works well for nonconvex problems (Fisher information is convex, see [Li, Lu, Wang (2020)]).

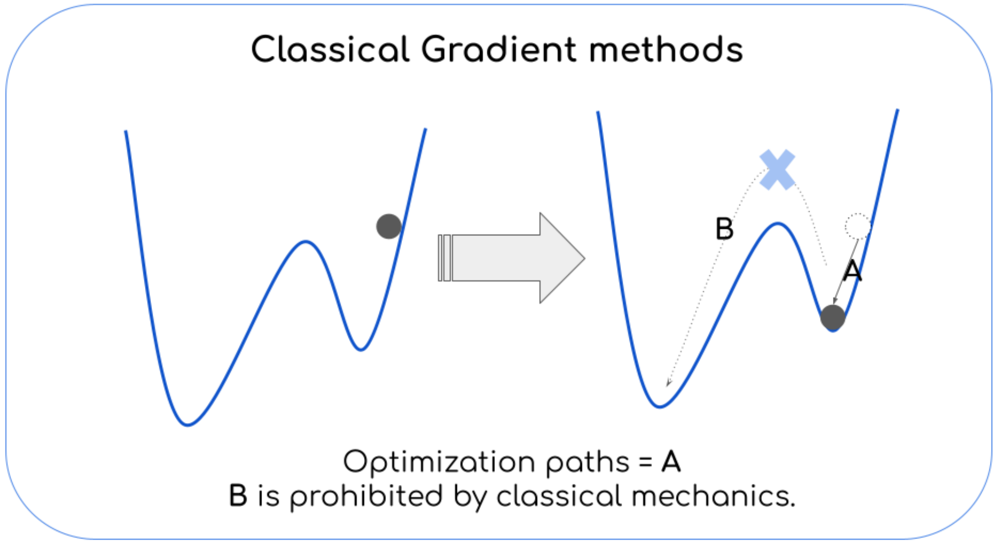
- <u>Question: can we use this interpretation to prove faster convergence rate?</u>

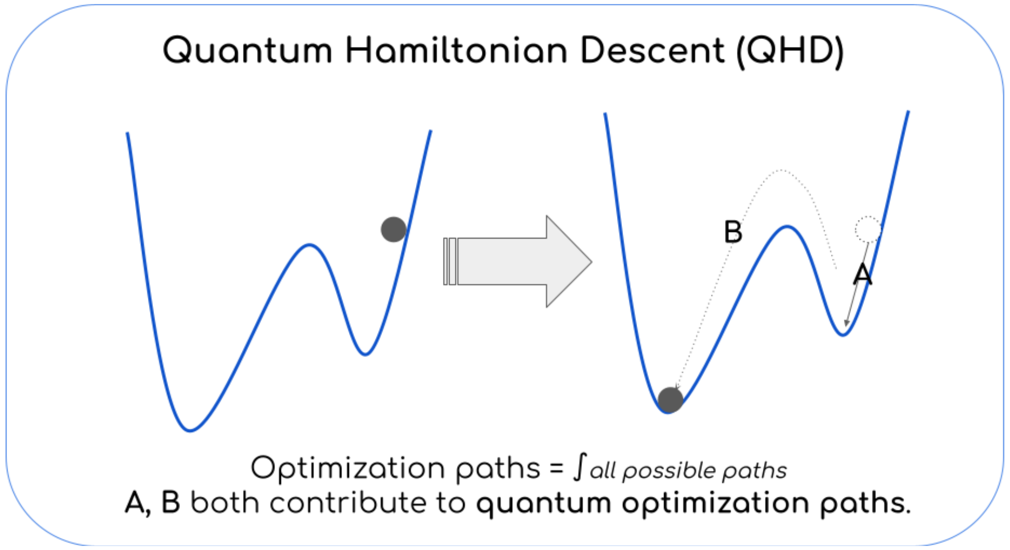
Part II:

A Quantum-Classical Performance Separation

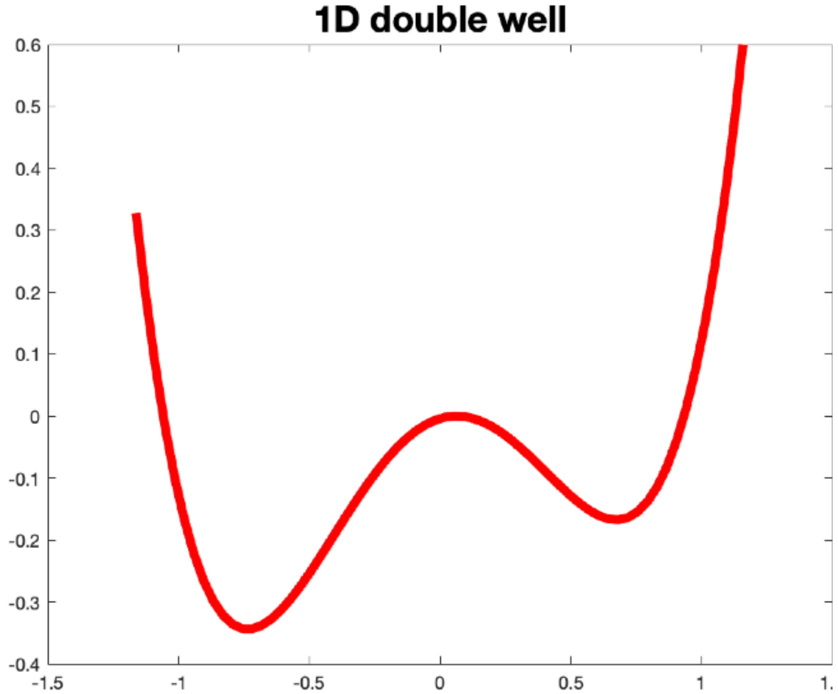
Quantum Hamiltonian Descent — practical setting?

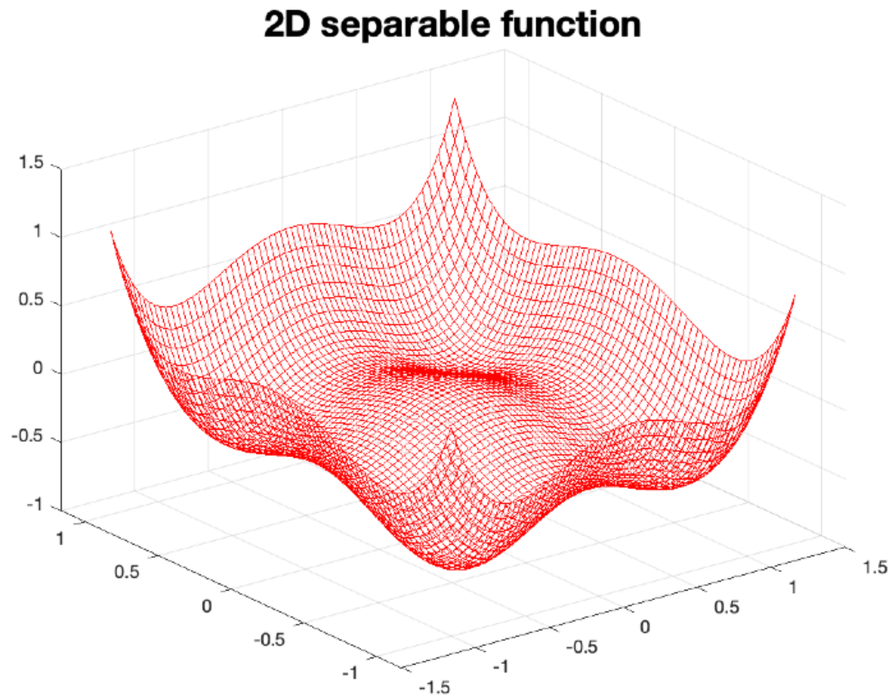
1-dim nonconvex model problem: double-well potential — $f(x)$





d-dim objective function: $F(x_1, \ldots, x_d) = \sum_{k=1}^{d} f(x_k),$

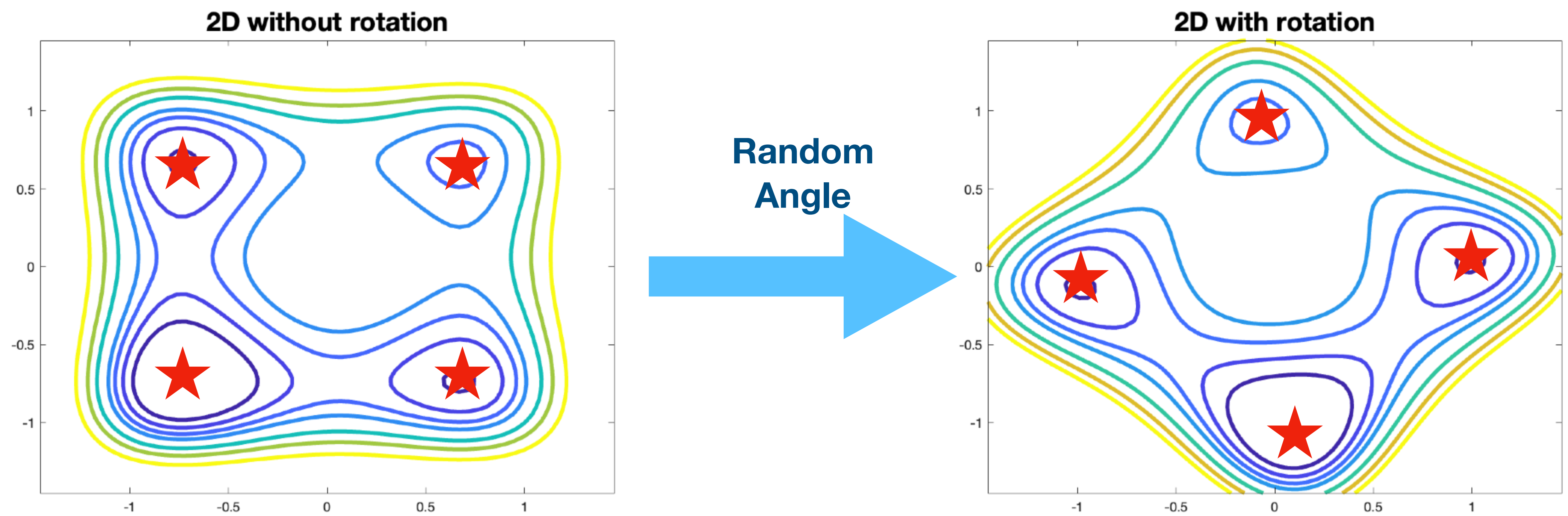




- $F(x)$ has 2^d local minima! (Only **one unique** global minimum).
- $F(x)$ is separable: not difficult for classical algorithms if the closed-form formula is given.

Construction of the optimization instances

Our instances = d-dim separable functions + random rotation



$$F(x_1, \ldots, x_d) = \sum_{k=1}^{d} f(x_k) \mapsto F_U(x) = F(Ux)$$

- $F_U(x)$ still has 2^d local minima, with a **unique** global minimum.
- $F(x)$ is non-separable: difficult to recover the rotation even with the closed-form formula!

QHD: a polynomial-time quantum algorithm

Given an optimization problem $f(x): \mathbb{R}^d \to \mathbb{R}$ with a unique global minimizer x^*.
x is a **δ-approximate** solution if $\|x - x^*\| < \delta$.

Theorem (Informal) [Leng, Zheng, Wu (2023)]

Let $f(x): \mathbb{R} \to \mathbb{R}$ be a double-well potential function. Define $F_U(x) = F(Ux)$ where $F(x) = \sum_{k=1}^{d} f(x_k)$ and U is an arbitrary orthogonal matrix. For any small $\delta > 0$, QHD can produce a δ-approximate solution with probability at least $2/3$ using

- $\tilde{\mathcal{O}}(d^3/\delta^2)$ quantum queries to F_U, and
- $\tilde{\mathcal{O}}(d^4/\delta^2)$ additional 1- and 2-qubit gates.

Manuscript in preparation.

- The QHD Hamiltonian (more precisely, the Laplacian operator) is rotationally invariant.
- The ground state of the QHD Hamiltonian is the *vehicle* of quantum optimization.
- We use an adiabatic theorem for unbounded Hamiltonian.

A quantum-classical performance separation

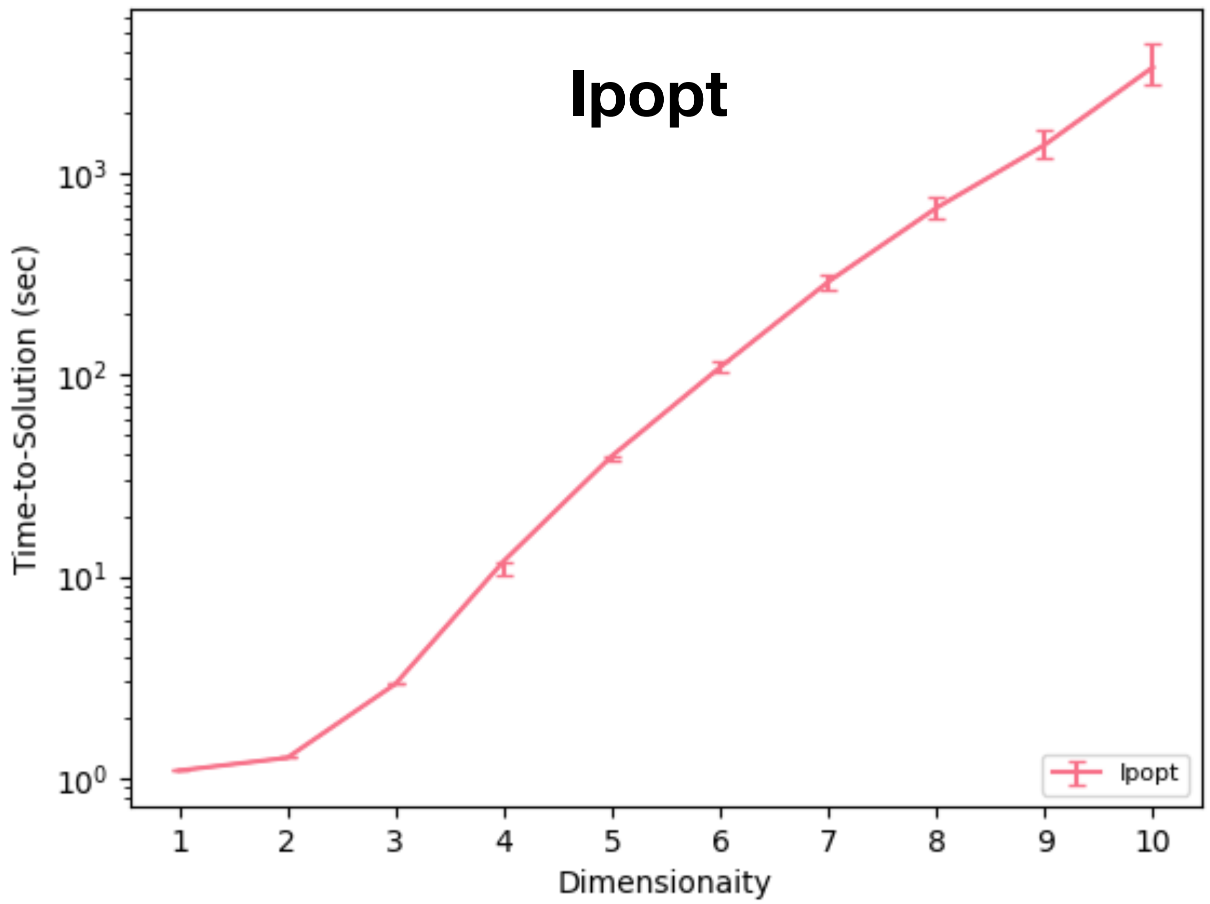
Time-To-Solution (TTS)

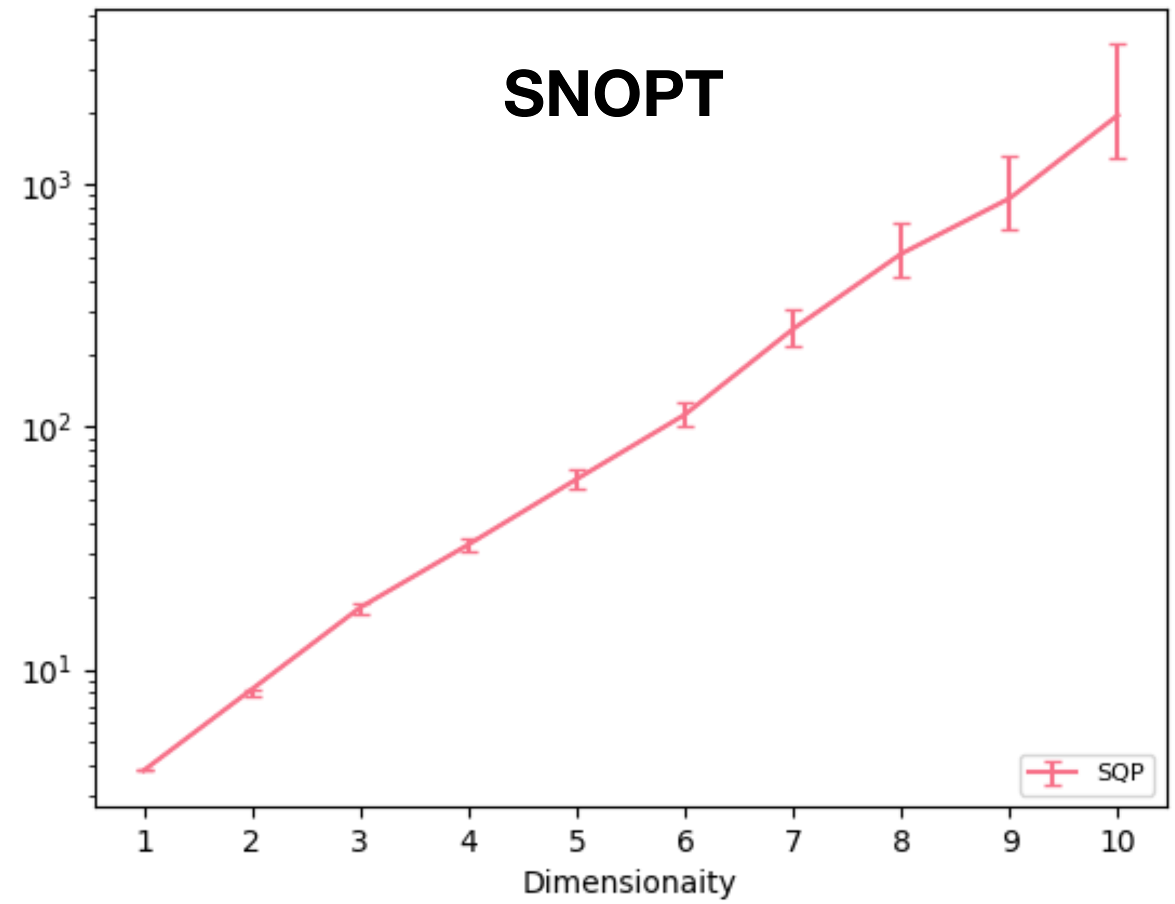
$$\mathrm{TTS} = t_{\mathrm{f}} \left\lceil \frac{\ln(1-0.99)}{\ln(1-P_g)} \right\rceil$$

- t_f - algorithm/solver running time (wall clock time)
- p_g - success probability per run

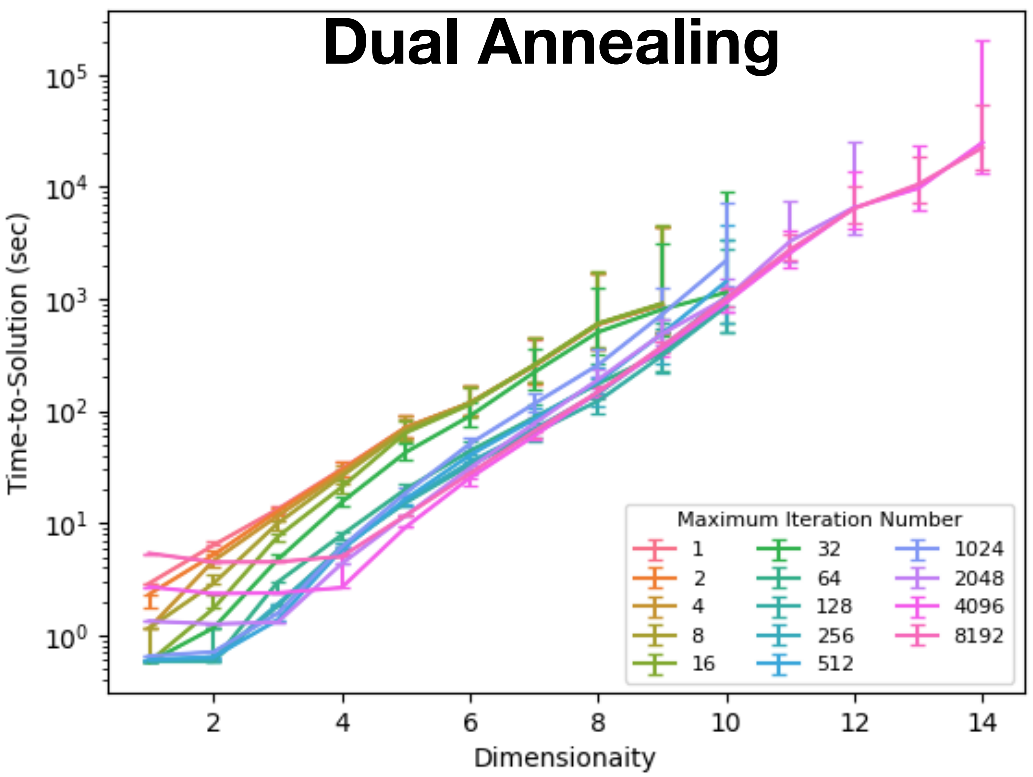
QHD TTS - $\mathcal{O}(d^4)$ (given fixed δ)

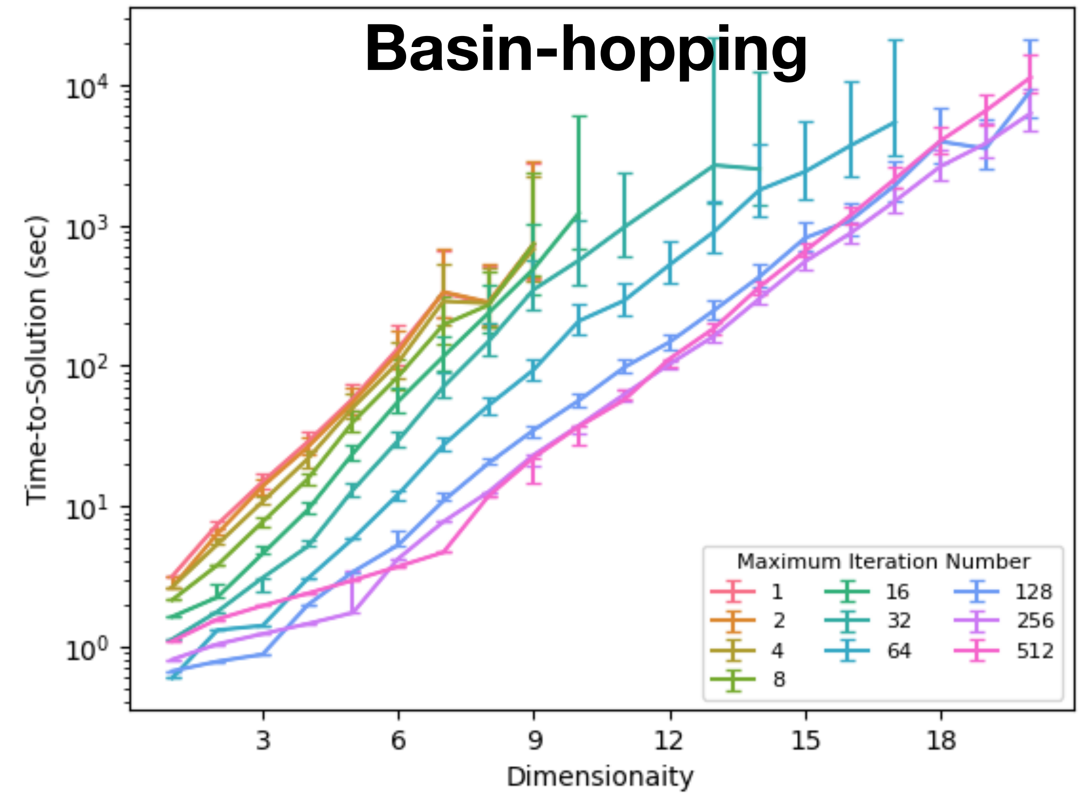
Classical TTS - numerical results suggest super-polynomial scaling in d

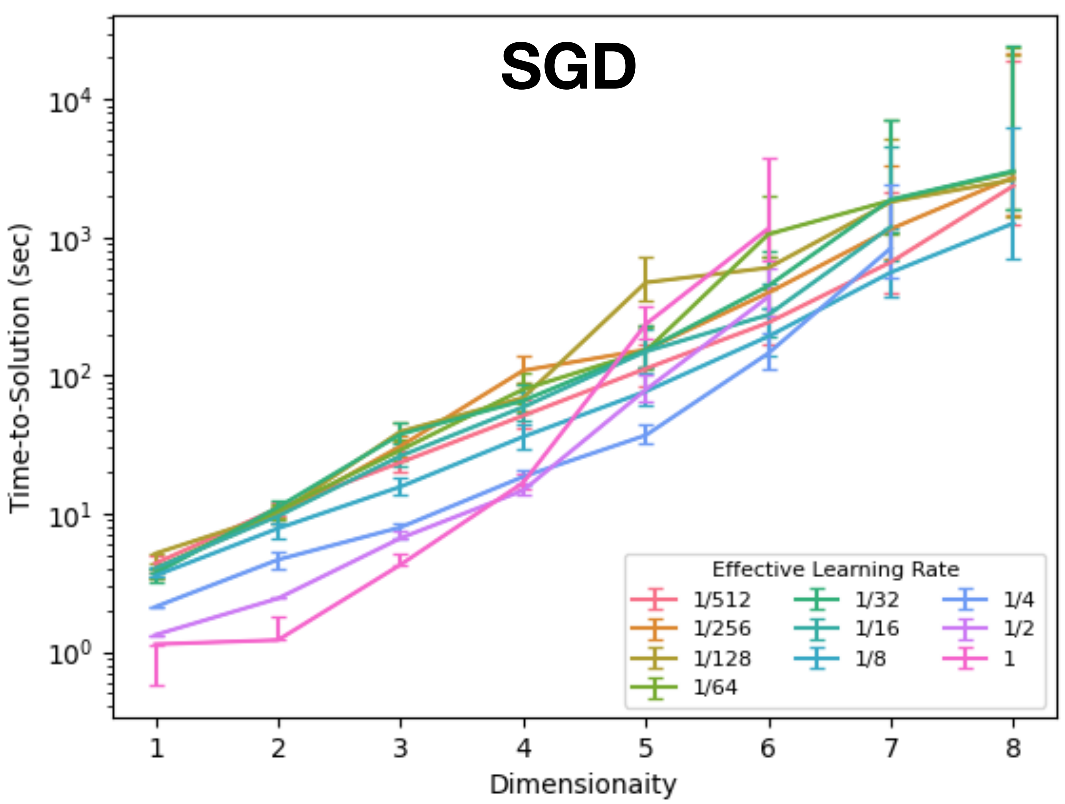


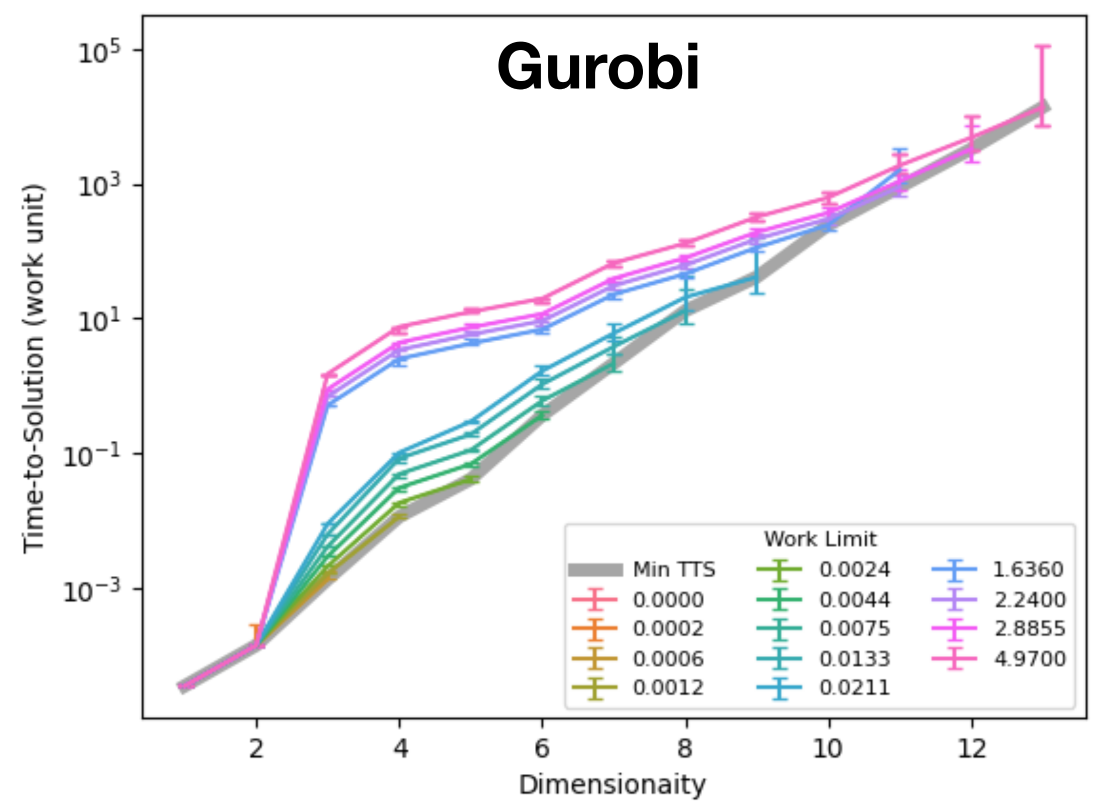


A quantum-classical performance separation









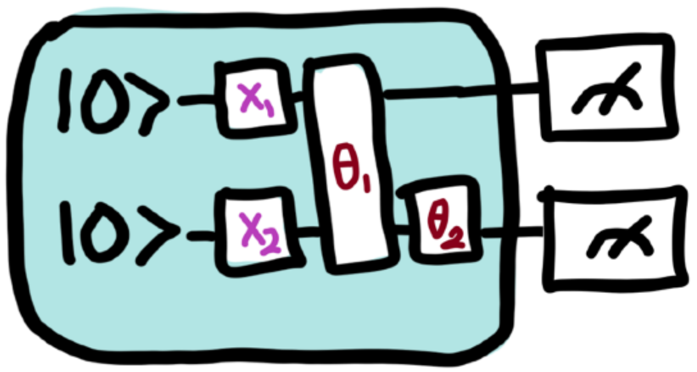
Part III:

Implementation on Quantum Computers

QHD with digital quantum computers

$$H(t) = e^{\varphi(t)}(-\frac{1}{2}\nabla^2) + e^{\chi(t)}f(x), t \geq 0$$

Digital Quantum Implementation



- The algorithm is effectively a **time-dependent Hamiltonian simulation** in the real space (we know poly-time algorithms).
- Requires hundreds of millions of gates!
- **Conclusion**: the digital implementation is **far from being feasible** in near term!

Corresponding **T-gate** Count with Digital Quantum Computing (before fault-tolerance)

Dimensions	**3-qubit format**	**16-qubit format**	**32-qubit format**
50	5.49e+8	7.8386e+9	2.672e+10
60	6.588e+8	9.4063e+9	3.2064e+10
75	8.235e+8	1.1758e+10	4.008e+10

QHD with analog quantum computers

Analog Quantum Implementation

- **Analog simulation**: problem solving by *emulating real quantum systems*.
- Abstraction: **Quantum Ising Machine** (**QIM**, e,g., D-Wave, QuEra, etc.)

$$H(t) = -\frac{A(t)}{2}\left(\sum_j \sigma_x^{(j)}\right) + \frac{B(t)}{2}\left(\sum_j h_j \sigma_z^{(j)} + \sum_{j>k} J_{j,k}\sigma_z^{(j)}\sigma_z^{(k)}\right)$$

- **Programmability:** coefficients $h_j, J_{j,k}$ and functions $A(t), B(t)$.

- QHD is formulated as a quantum Hamiltonian evolution → suitable for analog implementation!
- Patter mismatch: QHD $H(t) = e^{\varphi_t}(-\Delta/2) + e^{\chi_t} f$
- We develop a new technique named **Hamiltonian embedding**: mapping our target Hamiltonian (QHD) to a "diagonal block" of the machine Hamiltonian (QIM).
- If $H = H_0 \oplus H_1$, we have $e^{-iHt} = e^{-iH_0 t} \oplus e^{-iH_1 t}$.
- We implement QHD on today's analog quantum computers (D-Wave's **advantage_system6.1**).
- **Advantage**: resource-efficiency → first large-scale empirical study for nonlinear optimization using quantum computers.

Hamiltonian embedding of QHD

$$i\frac{\partial}{\partial t}\Psi(t,x) = \left[\underbrace{-\frac{e^{\varphi_t}}{2}\nabla^2}_{\text{Kinetic part}} + \underbrace{e^{\chi_t} f(x)}_{\text{Potential part}}\right]\Psi(t,x),$$

$\xrightarrow{\text{Finite difference}}$

$$i\frac{\mathrm{d}}{\mathrm{d}t}|\phi_t\rangle = \left[\underbrace{-\frac{e^{\varphi_t}}{2}\hat{A}}_{\text{Kinetic part}} + \underbrace{e^{\chi_t}\hat{F}}_{\text{Potential part}}\right]|\phi_t\rangle$$

- **Kinetic operator:** $\hat{L} = \frac{1}{(1/r)^2}\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ \cdots & \cdots & \cdots & \cdots \\ & 1 & -2 & 1 \\ & & 1 & -2 \end{bmatrix} \rightarrow \hat{A} = \frac{1}{(1/r)^2}\begin{bmatrix} 0 & 1 & & \\ 1 & 0 & 1 & \\ \cdots & \cdots & \cdots & \cdots \\ & 1 & 0 & 1 \\ & & 1 & 0 \end{bmatrix}$

Tri-diagonal

Diagonal

- **Potential operator:** $\hat{F} = \begin{bmatrix} f(a_0) & & & \\ & f(a_1) & & \\ \cdots & \cdots & \cdots & \cdots \\ & & f(a_{r-1}) & \\ & & & f(a_r) \end{bmatrix}$

Hamming states — an orthonormal basis

$$|H_j> = \frac{1}{\sqrt{C_j}}\sum_{|b|=j}|b> \quad \text{where } C_j = \binom{N}{j} \text{ and } 0 \le j \le N$$

$$|H_0> = |0000>, |H_1> = \frac{1}{2}(|1000> + |0100> + |0010> + |0001>), etc.$$

Lemma (informal version). [Leng, Hickman, Li, Wu (2023)]

Given n qubits, the subspace $\mathcal{S}$ spanned by all $(n+1)$ Hamming states is an invariant subspace of the QIM Hamiltonian. The projection of the QIM Hamiltonian into the subspace $\mathcal{S}$ approximates the discretized QHD Hamiltonian.

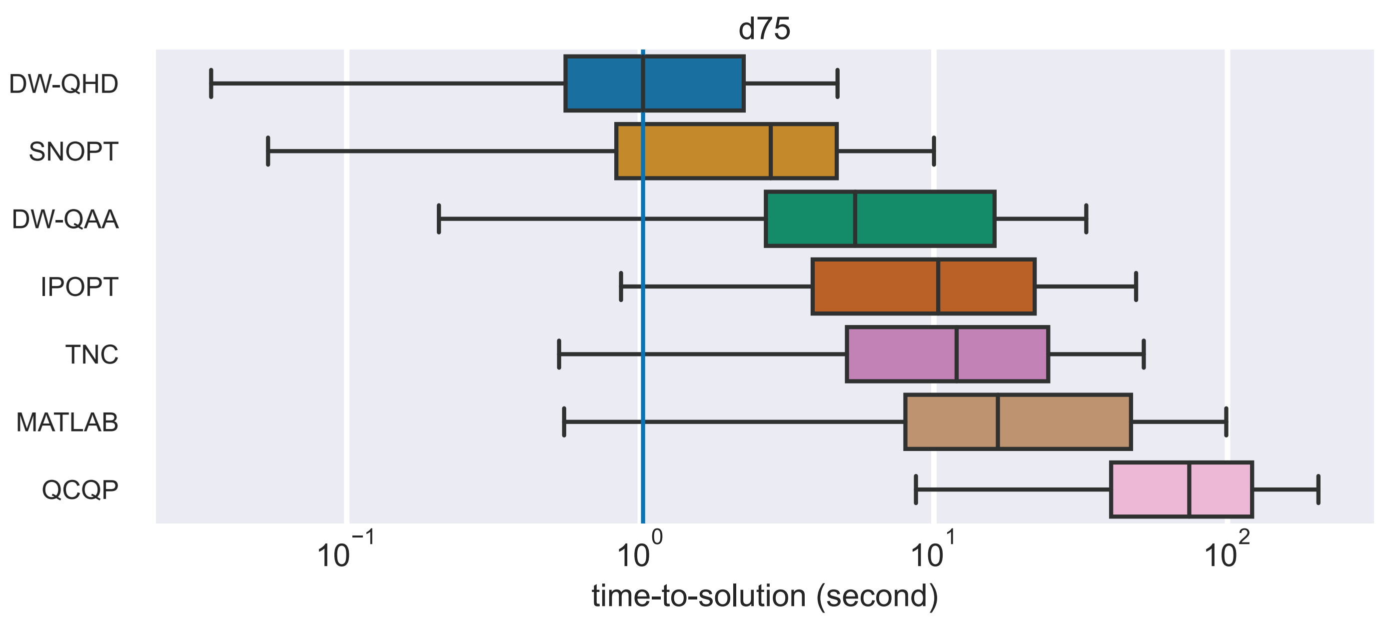
Large-scale empirical study on real quantum computers

We identify a class of *non-trivial* and *self-interesting* optimization problems that can be mapped to QIMs — **Quadratic Programming (QP) with box constraints**.

$$\begin{aligned} &\text{minimize} \quad && f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{b}^\top \mathbf{x} \\ &\text{subject to} \quad && \mathbf{0} \preccurlyeq \mathbf{x} \preccurlyeq \mathbf{1}, \end{aligned}$$

QP — NP-hard with indefinite Q

(For implementation details, see our paper arXiv:2303.01471 **Appendix F**)



Time-To-Solution (TTS)

The lower, the better!!

$$\mathrm{TTS} = t_f \left\lceil \frac{\ln(1-0.99)}{\ln(1-P_g)} \right\rceil$$

t_f - quantum anneal time + post-processing or classical runtime (**wall-clock time**)

p_g - success probability per run

- **DW-QHD** is better than the rest, including **DW-QAA,** and **classical GD, interior points**, and **some local search heuristic.**
- Assuming DW-QHD is no worse than the ideal QHD, this provides **a very strong empirical evidence supporting QHD**.
- QHD does not beat SOTA classical solvers (e.g., Gurobi, CPLEX) in D75. However, such branch-and-bound solvers are **not scalable!**

Summary & Future Work

- **QHD** is an upgraded version of classical GD and variants.
- QHD leverages the continuous structure of the problem and converges faster than QAA.
- QHD has different solution path compared to classical GD.

- **QHD** can be used as *subroutines* for more complicated algorithms like branch-and-bound.

All data & codes are available online!

- Source code (Github): https://github.com/jiaqileng/quantum-hamiltonian-descent
- Raw data (Box): https://umd.app.box.com/s/vq747fvjnt8qrkbxprexhoh44n0q9m0i
- Website: https://jiaqileng.github.io/quantum-hamiltonian-descent/

arXiv:2303.01471

Thank You!

QHD Website