Quantum Hamiltonian Descent

Xiaodi Wu
QuICS & UMD

Based on arXiv: 2303.01471 and 2311.00811
with Jiaqi Leng, Ethan Hickman, Joseph Li, and Yufan Zheng (All UMD)

Website: https://jiaqileng.github.io/quantum-hamiltonian-descent/

UNIVERSITY OF MARYLAND

JOINT CENTER FOR QUANTUM INFORMATION AND COMPUTER SCIENCE

On the Pursue of Quantum Speedups in Optimization

Optimization - a long studied topic in q. applications (e.g., *adiabatic* algorithms originally designed for *discrete* optimization).

We focus on recent progress in **continuous** optimization, applicable to *machine learning, operation research, scientific computing*, and so on.

Continuous optimization - *very different nature* in (quantum) algorithm design:
involving (math) high-dimensional analysis in proving convergence

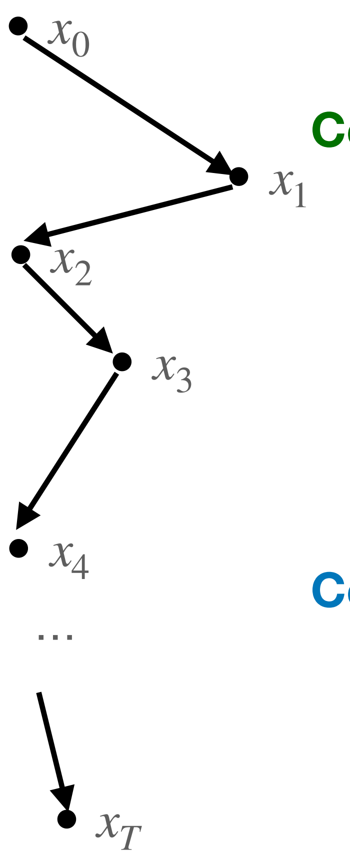
Challenges and Opportunities in new *primitives* of quantum speedups!

Q. Speedup in Optimization

APPROACH I: Quantize Components of Classical Algorithms.

Approach I: Quantizing Components of Classical Algs

Iterative Algorithms:



Classical Computation per step

Quantum Computation per step

Methodology : replace classical by quantum step by step

Difficulty: NOT ALL classical parts can be sped up!!
And not many known quantum speedups!!

Even when it can, some information in the algorithm will need to be *encoded in quantum* to leverage this speedup.

Overheads in *jumping between quantum and classical* might **kill** the potential speed-up, e.g., in tomography.

Therefore, successful examples usually start with classical algorithms with components allowing *quantum speedup*, and make necessary changes to *balance* the potential speedup and corresponding overheads in achieving some real speedups.

An **important** feature of this approach: *almost no change of trajectories of classical algs*.

On the Pursue of Quantum Speedups in Optimization

Q. Speedup in Optimization

APPROACH I: Quantize Components of Classical Algorithms.

Gradient Descent (GD) & its variants are fundamental in both theory and practice!!

Successful Examples (my group):

(1) convex optimization [**C**CL**W** QIP 2019]

(2) semidefinite program [BKLLS**W** QIP 2019]

(3) classification [L**CW**, ICML 2019]

(4) matrix-game solving [L**C**W**W**21, AAAI 2021]

(5) volume estimation [**C**CLHW**W**, QIP 2020]

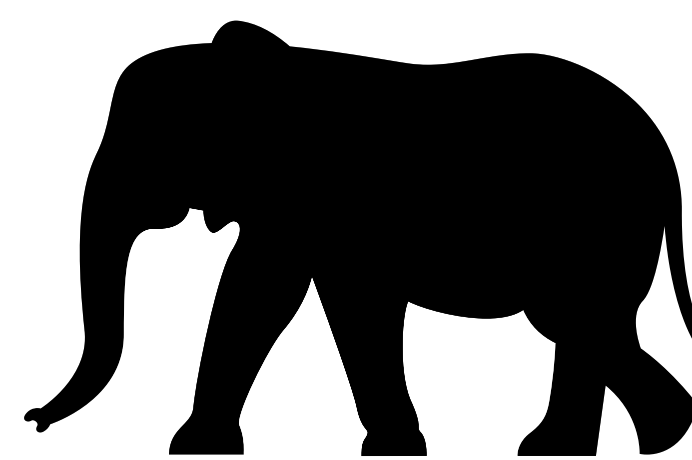
(6) escaping from saddle points [Z**L**L, QIP 2021]

Any GENUINE Quantum Speedup?

Non-ideal Progress

(1) quantum to compute GD but w/ complexity exponential in #iterations [RSW+19]

The parameter range of speedup is very limited and the optimization path is classical.



A Genuine Quantum Gradient Descent

Key Observation - *re-interpret* GD as something w/ physics meaning !!

Gradient Descent —*continuous-time limit*→ **Dynamical System w/ physics law**

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

Nesterov's method
Accelerated methods

←*discretization*—

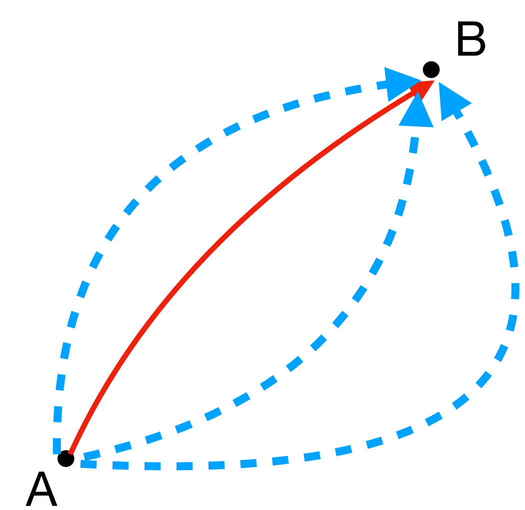
Specific Lagrangian/Hamiltonian
w/ different space metrics

A *generic* correspondence from *physical dynamical systems* to generate many GD algorithms !!

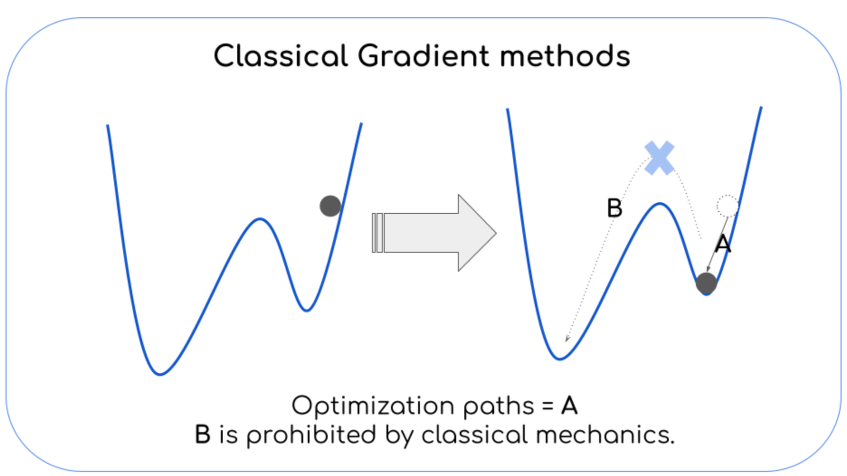
Make It Quantum - **Path Integral Formulation of Quantum Mechanics !!**

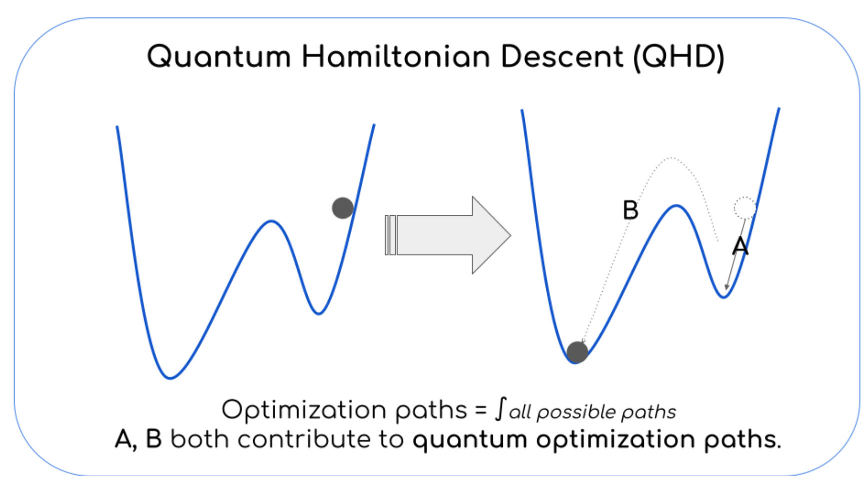
$\int_{paths}$ **Gradient Descent** —*continuous-time limit*→ **Dynamical System w/ quantum law**

Time-Dependent Hamiltonian Simulation



A **systematic** way to design **NEW** quantum algorithms for optimization!





Quantum Hamiltonian Descent

Leng, Hickman, Li, **W.**, arXiv: 2303.01471

One Simple Construction in this family:

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

Goal to **minimize** f(x)

potential energy $\xrightarrow{\text{decaying } e^{\varphi(t)}/e^{\chi(t)}}$ potential energy

$e^{\varphi(0)-\chi(0)} = 1$ and $\lim_{t\to\infty} e^{\varphi(t)-\chi(t)} = 0$

$$e^{\varphi(t)-\chi(t)} \propto \frac{1}{1+\gamma t^2}$$

decaying factor based on [WWJ16, PNAS]

The quantum optimization algorithm, called **Quantum Hamiltonian Descent (QHD)**, will perform time dependent Hamiltonian simulation of $H(t)$ on simple initial quantum states, e.g., uniform super-position.

QHD (Ours)

1. Prepare an initial state $|\psi_0>$.
2. Simulating the Schrodinger equation: $i\partial_t\psi = \hat{H}(t)\psi$,

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

3. At time t = T, measure the final state $|\Psi_T>$ with the position observable $\hat{x}$.
4. The measurement results will cluster around the global minimizer of f.

GD (Classical)

1. Prepare an initial guess x_0.
2. Gradient step: $x_{k+1} = x_k - \eta\nabla f(x_k)$

Equivalent to simulating $\dot{x} = -\nabla f(x)$

3. At k = K (or stopping criterion is met), obtain the final updated result x_K.
4. The results are usually first-order stationary solution (local minimizer/saddle point) of f.

QHD is similar to GD in terms of its simplicity and leveraging first-order gradient information. Thus, *QHD can be deemed as 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.

On the Pursue of Quantum Speedups in Optimization

Q. Speedup in Optimization

APPROACH I:
Quantize Components of Classical Algorithms.

APPROACH II:
Path Integral of Classical Algorithmic Trajectories!

Successful Examples (my group):

(1) convex optimization [**C**CL**W** QIP 2019]

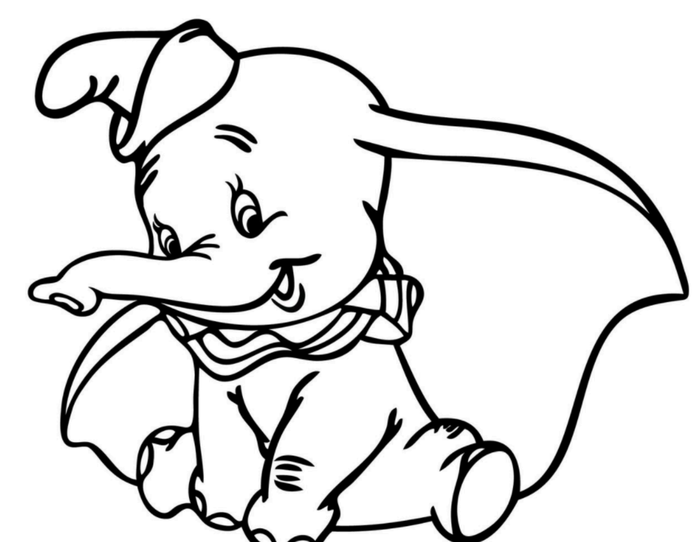
(2) semidefinite program [BKLLS**W** QIP 2019]

(3) classification [L**CW**, ICML 2019]

(4) matrix-game solving [L**C**W**W**21, AAAI 2021]

(5) volume estimation [**C**CLHW**W**, QIP 2020]

(6) escaping from saddle points [Z**L**L, QIP 2021]

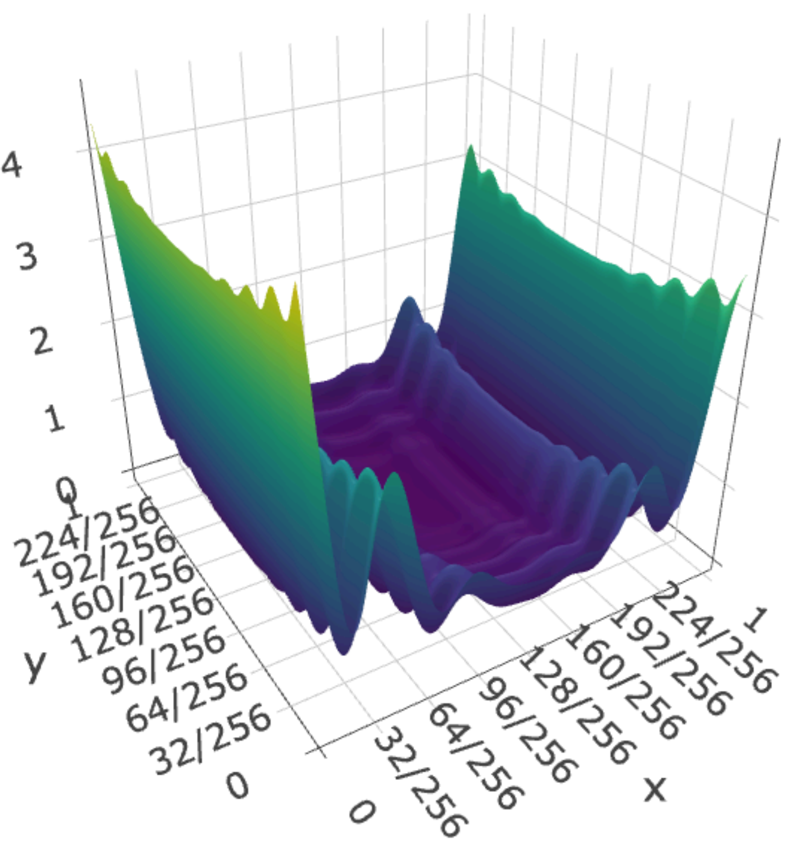


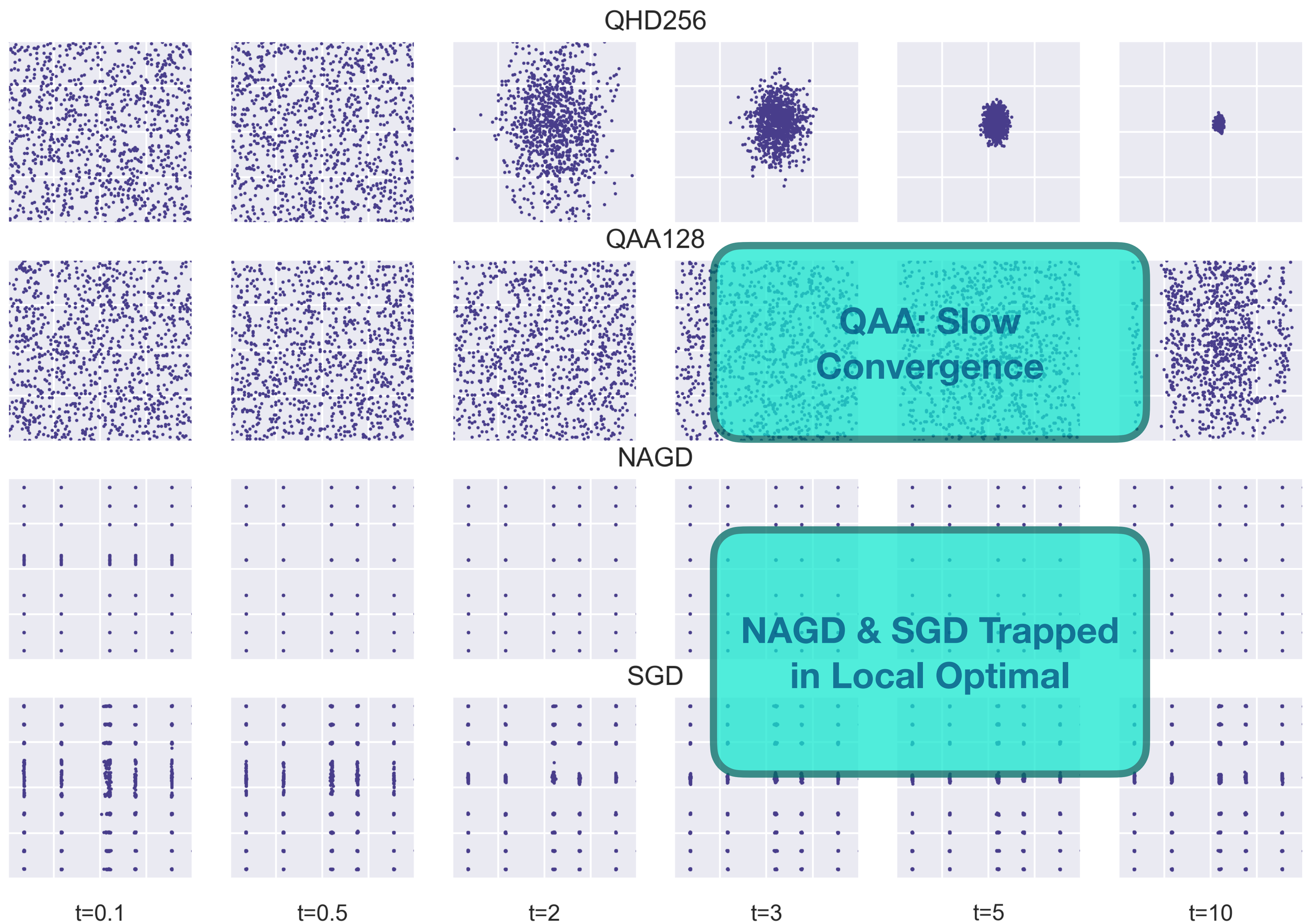
QHD - simple numerical observation for intuition

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

SGD: stochastic gradient descent.

QHD showcase: the Levy function.





Quantum Hamiltonian Descent - Non-Convex Setting

We have similar observations on a benchmark of 29 2D non-convex functions! Please check our project website!

QHD Website

Quantum Hamiltonian Descent — QHD

Home | Executive Summary | Nonconvex 2D | Quadratic Programming | Details

2D Test Functions

We demonstrate the difference between QHD and other algorithms with interactive 3D visualizations and the three-phase diagrams (for details, see Executive Summary).

Test functions & Methodology

We gathered a set of 22 continuous two-dimensional test functions from the optimization literature (e.g., ref1, ref2, ref3). Functions are divided into five categories based on their diversified landscapes: (1) Ridges and Valleys: a mix of functions with steep barriers and/or valleys; (2) Basin: Most of the domain is at a low objective value, so exhaustive search would yield a small improvement in objective value. The gradient signal may be weak in most of the basin; (3) Flat: most of the domain is at a high objective value, so exhaustive search would yield a large improvement in objective value. The gradient signal may be weak, useless, or misleading over most of the domain; (4) Studded: highly non-convex functions, such that the gradient may be large in magnitude but misleading due to high local curvature. Marked by many local minima imposed on a base shape; (5) Simple: Functions solved efficiently by a classical gradient method.

For comparable performance, we normalize all objective functions. Details are available here. We compare QHD with the Quantum Adiabatic Algorithm (QAA), stochastic gradient descent (SGD) and Nesterov's accelerated gradient descent (NAGD) on objective functions with a variety of landscape features. QHD and QAA are numerically simulated (resolution: QHD = 1/256, QAA = 1/128).

For presentation, the surface plots are compressed to a 64×64 grid by taking the sum of probability over square tiles.

Results

	SGD	NAGD	QAA128	QHD256
ackley	0.066	0.000	0.007	**0.992**
ackley2	0.994	0.006	0.005	**>0.999**
alpine1	0.043	0.044	0.005	**0.968**
alpine2	0.185	0.187	0.008	**0.596**
bohachevsky2	**1.000**	0.997	0.000	**>0.999**
camel3	0.601	0.601	0.008	**0.980**
csendes	0.327	**1.000**	0.010	0.800
defl corr spring	**0.009**	**0.009**	0.006	0.000

	SGD	NAGD	QAA128	QHD256
dropwave	0.022	0.014	0.007	**0.706**
easom	0.077	0.035	0.008	**0.188**
griewank	0.074	0.075	0.007	**0.081**
holder	0.071	0.071	0.007	**0.825**
hosaki	0.601	0.606	0.008	**0.961**
levy	0.094	0.095	0.001	**>0.999**
levy13	0.217	0.139	0.006	**>0.999**
michalewicz	0.222	0.233	0.007	**0.797**

	SGD	NAGD	QAA128	QHD256
rastrigin	0.047	0.000	0.000	**>0.999**
rosenbrock	0.094	**1.000**	0.012	0.002
shubert	0.211	0.208	0.017	**0.619**
styblinski tang	0.276	0.269	0.007	**0.469**
sumofsquares	**1.000**	**1.000**	0.006	**>0.999**
xinsheyang3	**0.032**	0.019	0.008	0.007

Success rate of being within r=0.1 of the global minimum in a 1x1 domain. There is a random chance of pi/100 or about 3% that the starting point is already in the radius (though the algorithms can move the point out of the radius). Classical probabilities are estimated from sample means using the same 1000 starting points. Quantum probabilities are calculated from exact full state vector simulation. Functions are sorted alphabetically by name.

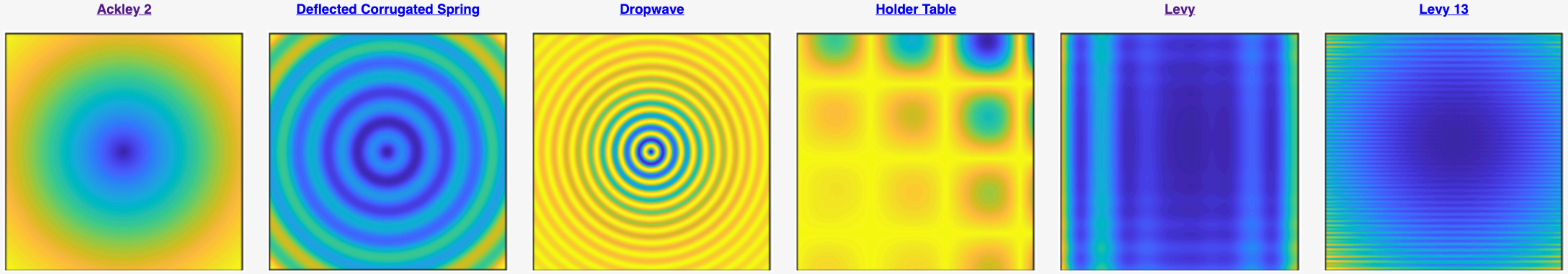
Interactive Visualizations

Click the thumbnails below and have fun!

2D Functions

Ridges or Valleys

Ackley 2 | Deflected Corrugated Spring | Dropwave | Holder Table | Levy | Levy 13



Basin

Quantum Hamiltonian Descent — Basic Theory

arXiv:2303.01471

Convex Setting: Let Ψ_t be the final state of the evolution $H(t)$; $\varphi(t) = \alpha(t) - \gamma(t); \chi(t) = \alpha(t) + \beta(t) + \gamma(t)$

Let $f(x)$ be continuously differentiable and convex, x^* is the unique local minimizer of f,

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

- Same converge rate as classical [WWJ16]
- Some advantages in time discretization (QHD is more stable)

Non-Convex Setting:

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 $H(t)$ be slow-varying (i.e., $|\dot{H}| \ll 1$), then

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

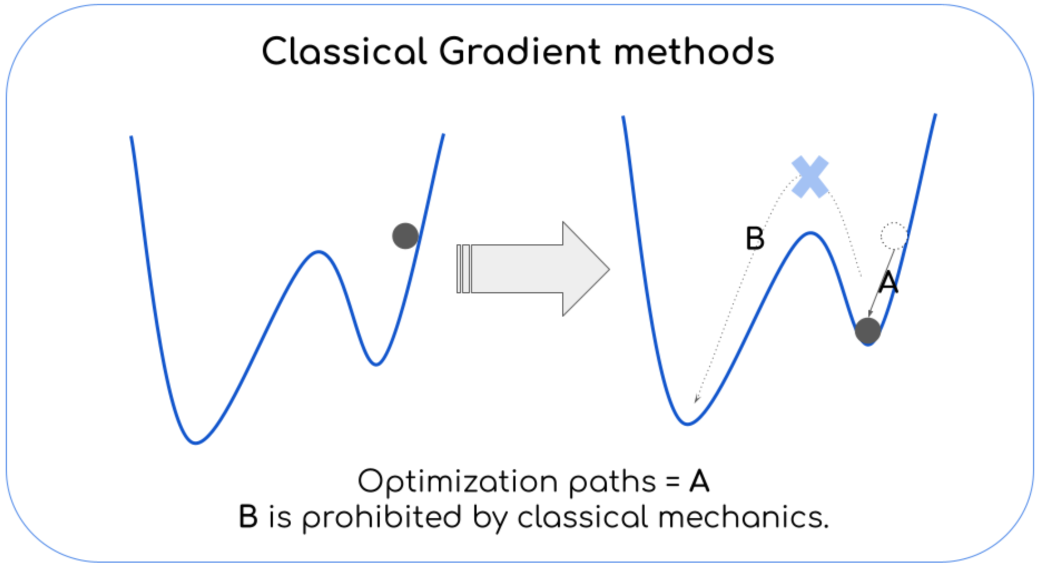
- Convergence to global optimal in the long-term limit. Not the case for classical GD.
- Proof based on time-dependent perturbation theory + spectral theory of Schrödinger operators.

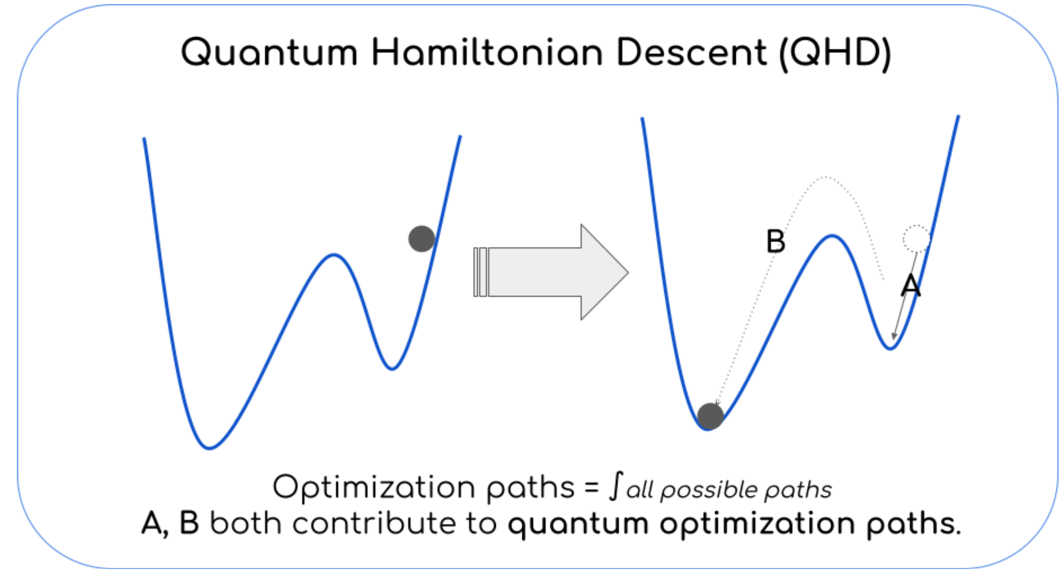
Quadratic (Grover-like) speedup emerged from QHD:

- By embedding a spatial search problem (w/ classical query lower bound $\Omega(n)$) into a non-convex optimization problem, we show QHD can solve it using $O(\sqrt{n})$ quantum queries.

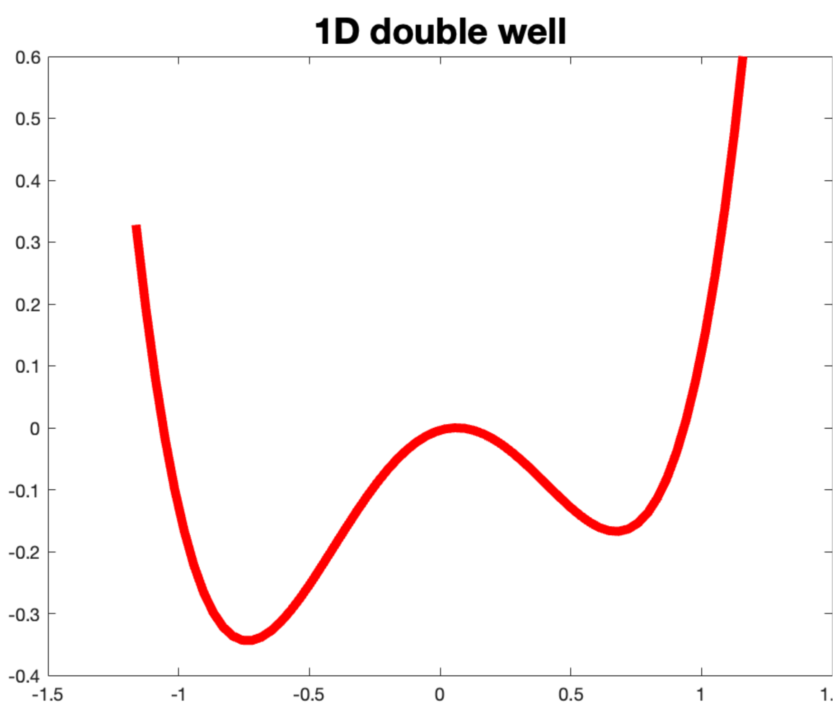
Performance Separation in Non-convex Optimization

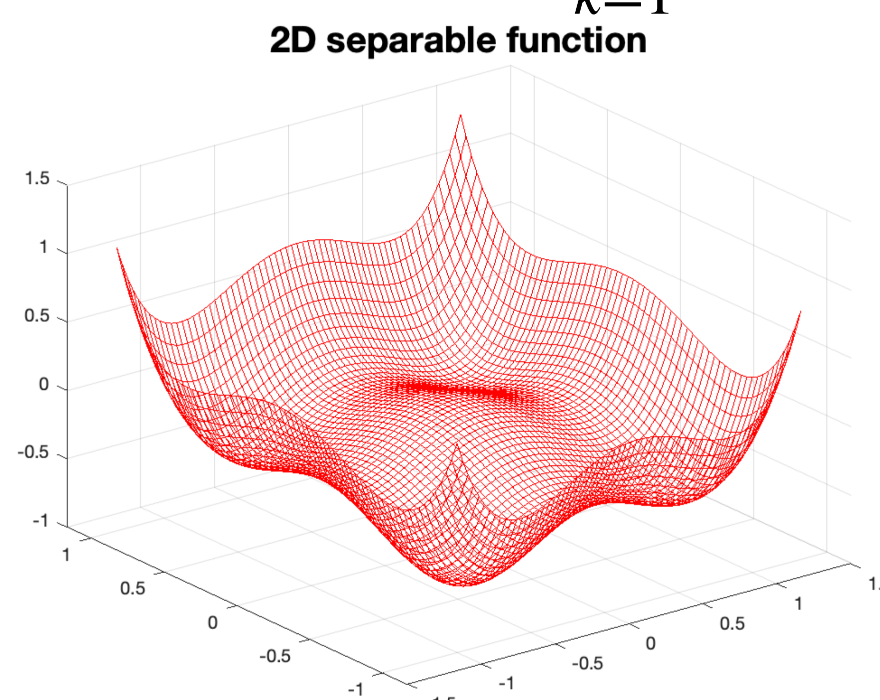
1-dim nonconvex model problem: double-well potential — $f(x)$





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

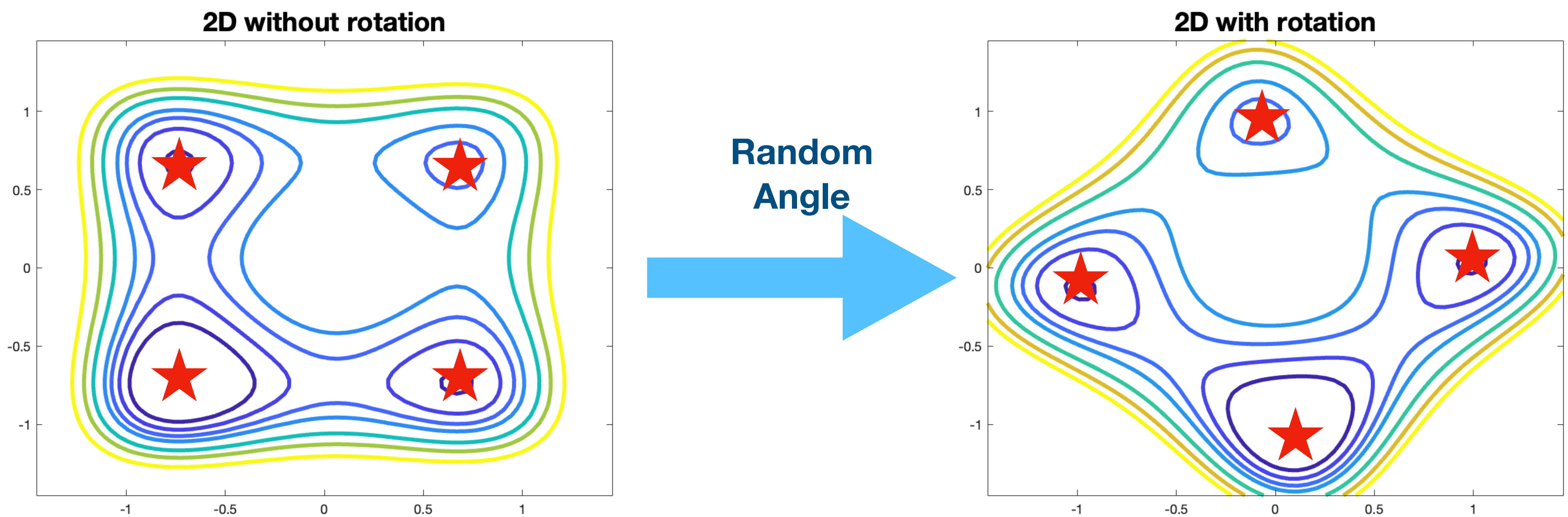




- $F(x)$ has 2^d local minima! (Only **one unique** global minimum).
- $F(x)$ is separable: not difficult for classical algorithms if the closed-form formula is given.

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 available as arXiv: 2311.00811

- 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

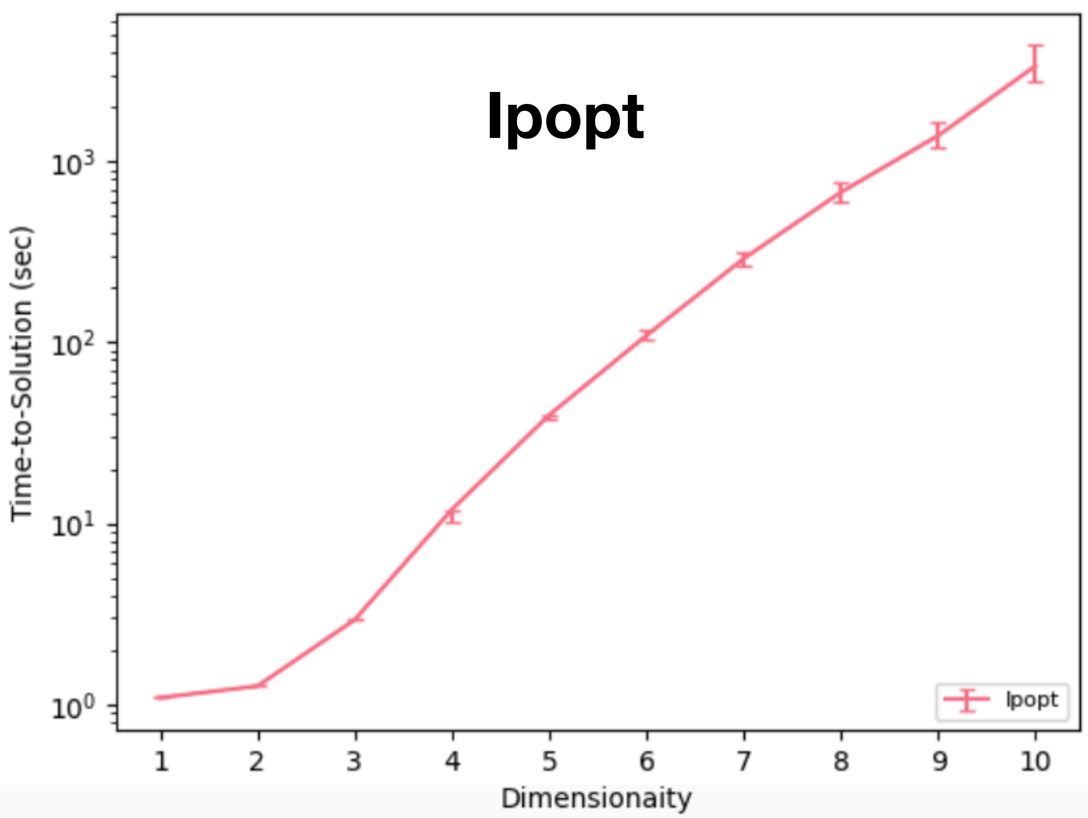
Time-To-Solution (TTS)

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

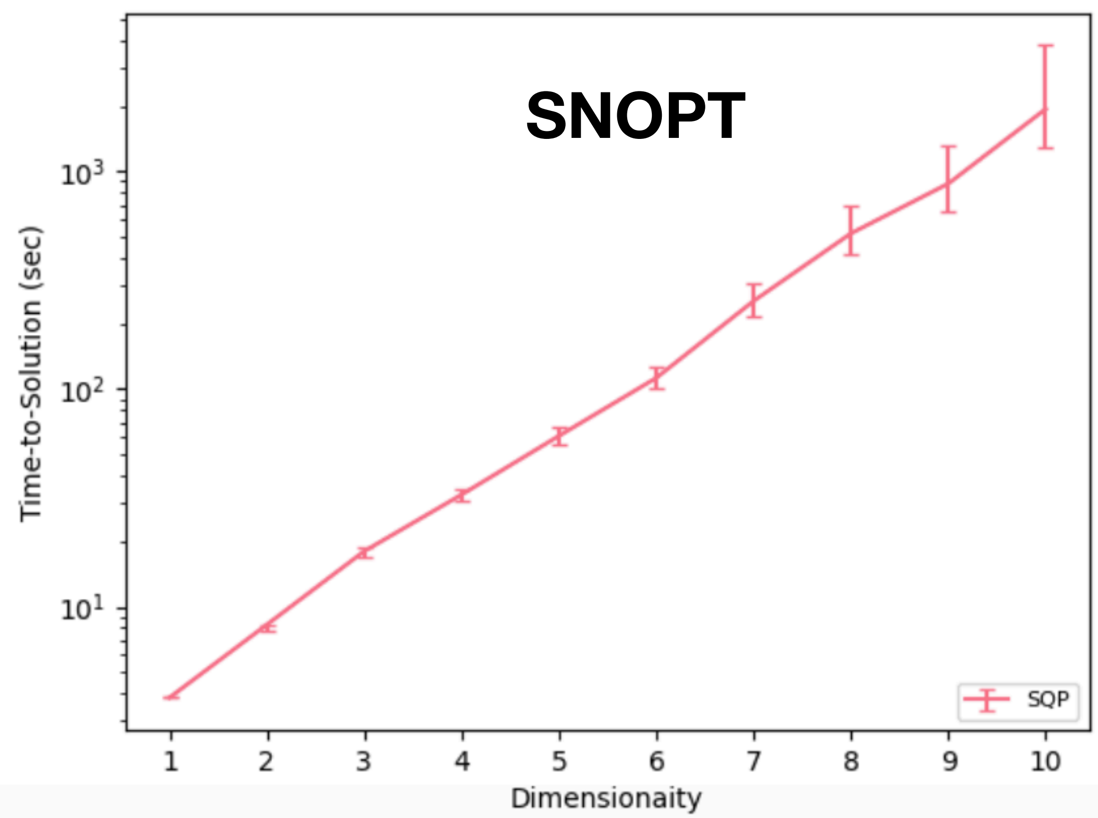
- t_f - algorithm/solver running time (wall clock time)
- p_g - success probability per run

QHD TTS - prove to be $\mathcal{O}(d^3)$ oracle class + $\mathcal{O}(d^4)$ elementary gates

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



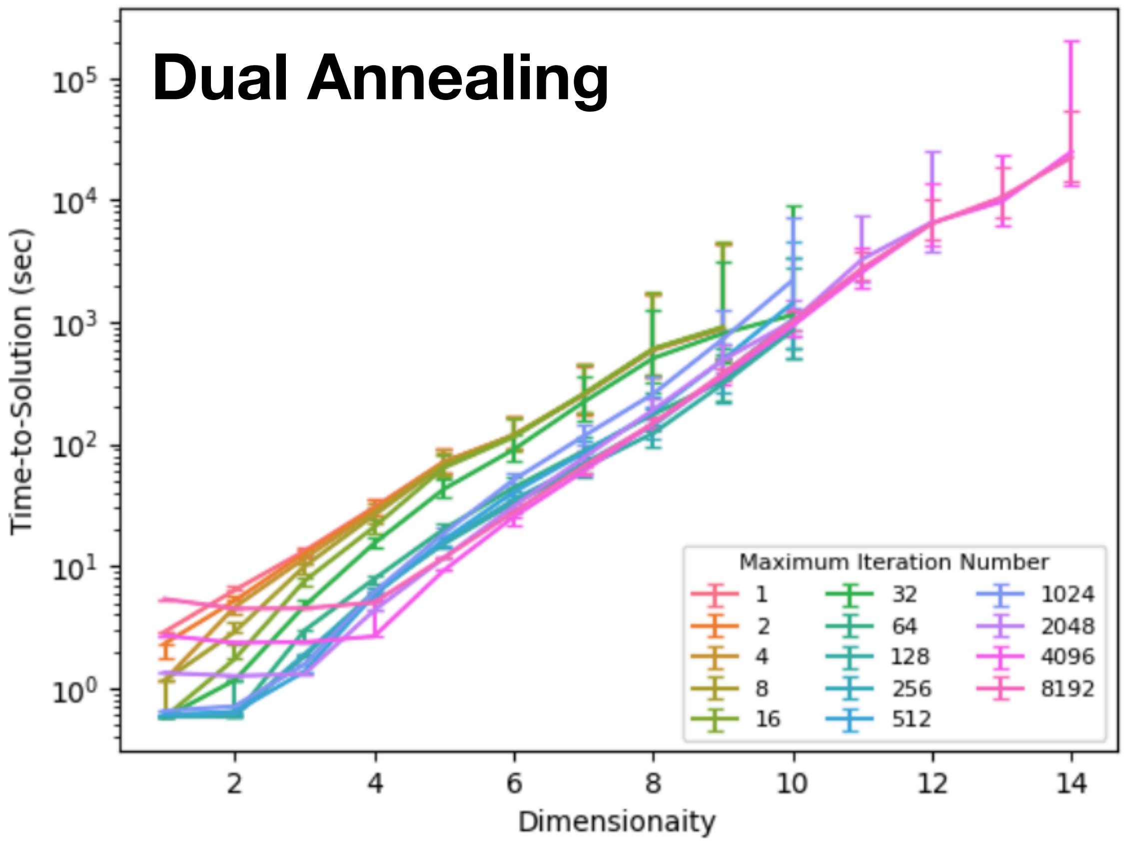
interior-point method



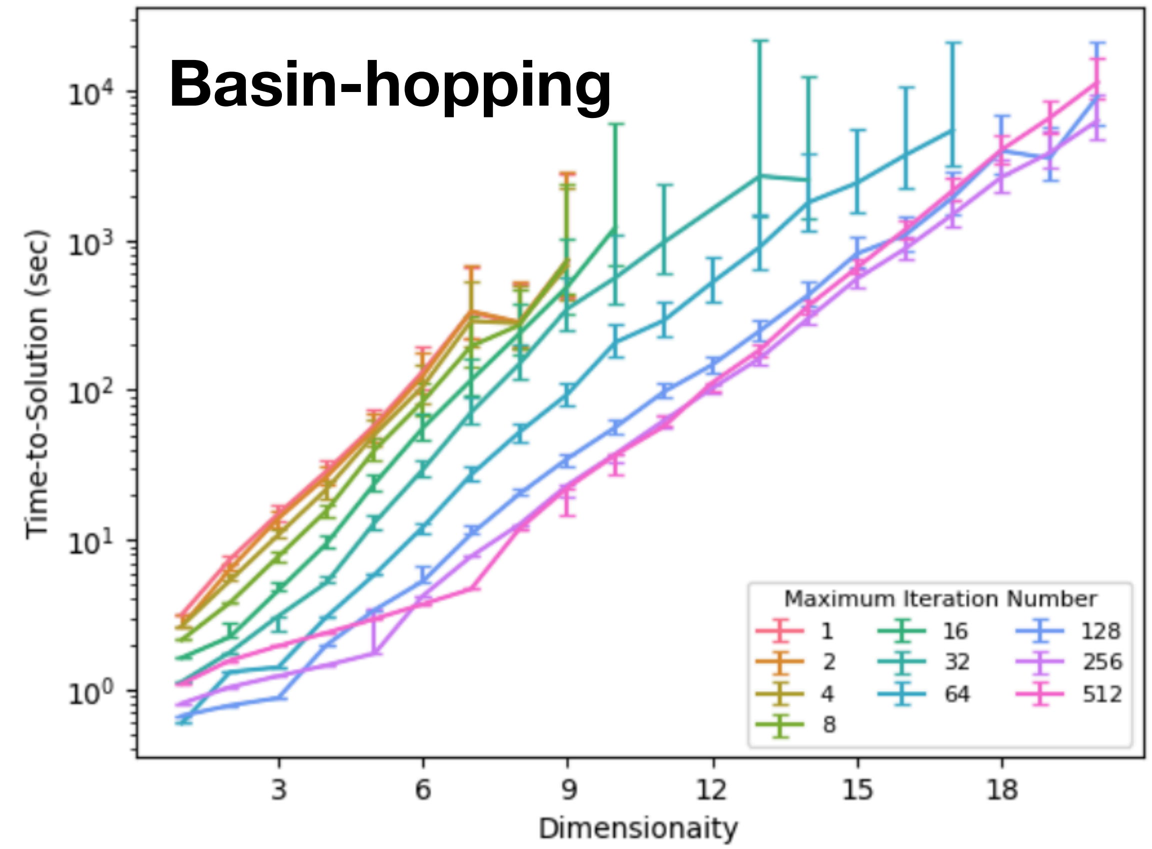
sequential quadratic programming

Newton and quasi-newton Methods

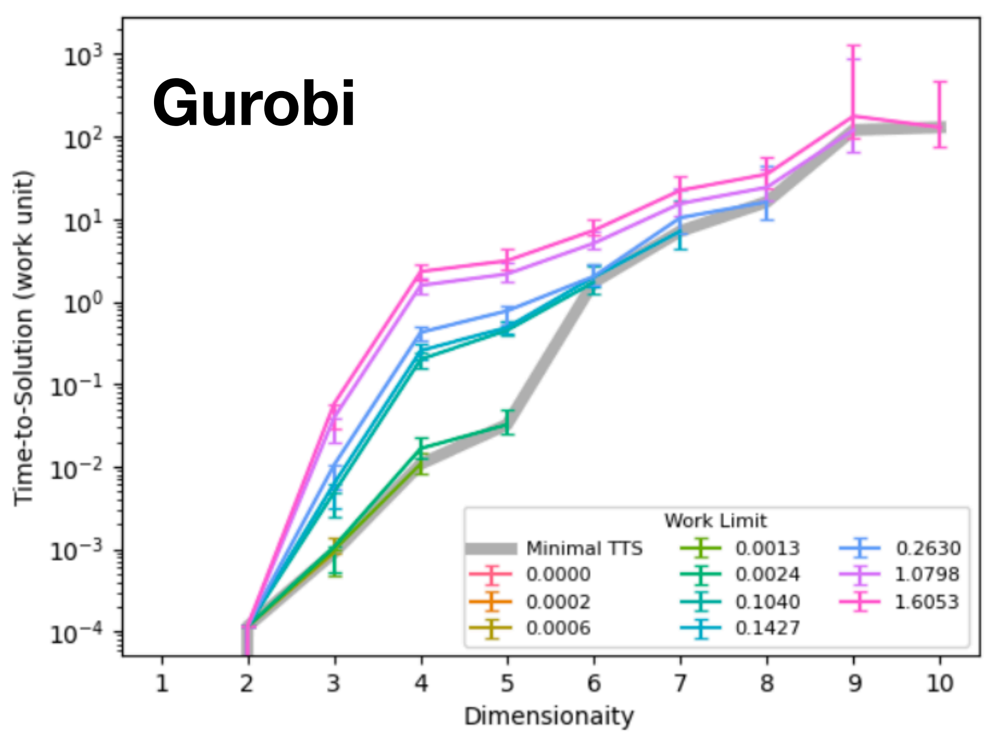
A quantum-classical performance separation



Simulated Annealing (SA) & fast SA



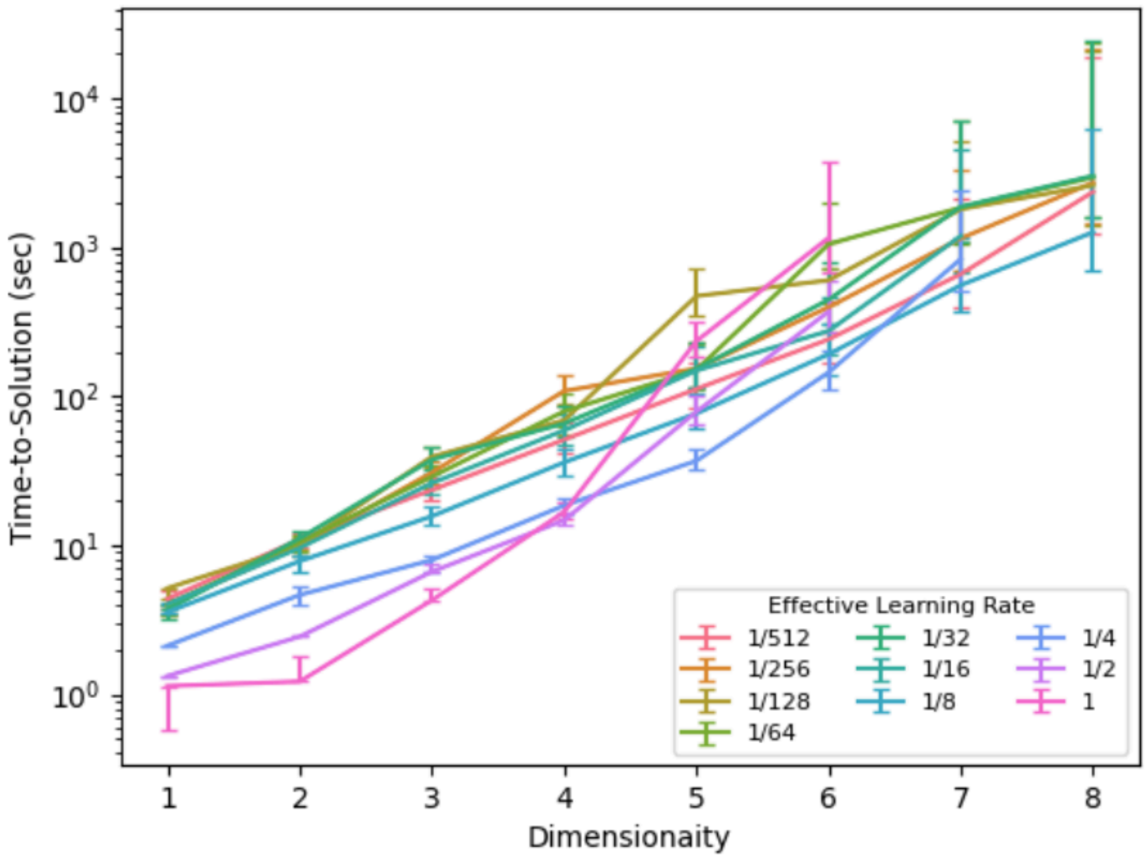
Basin-transformation + Monte Carlo



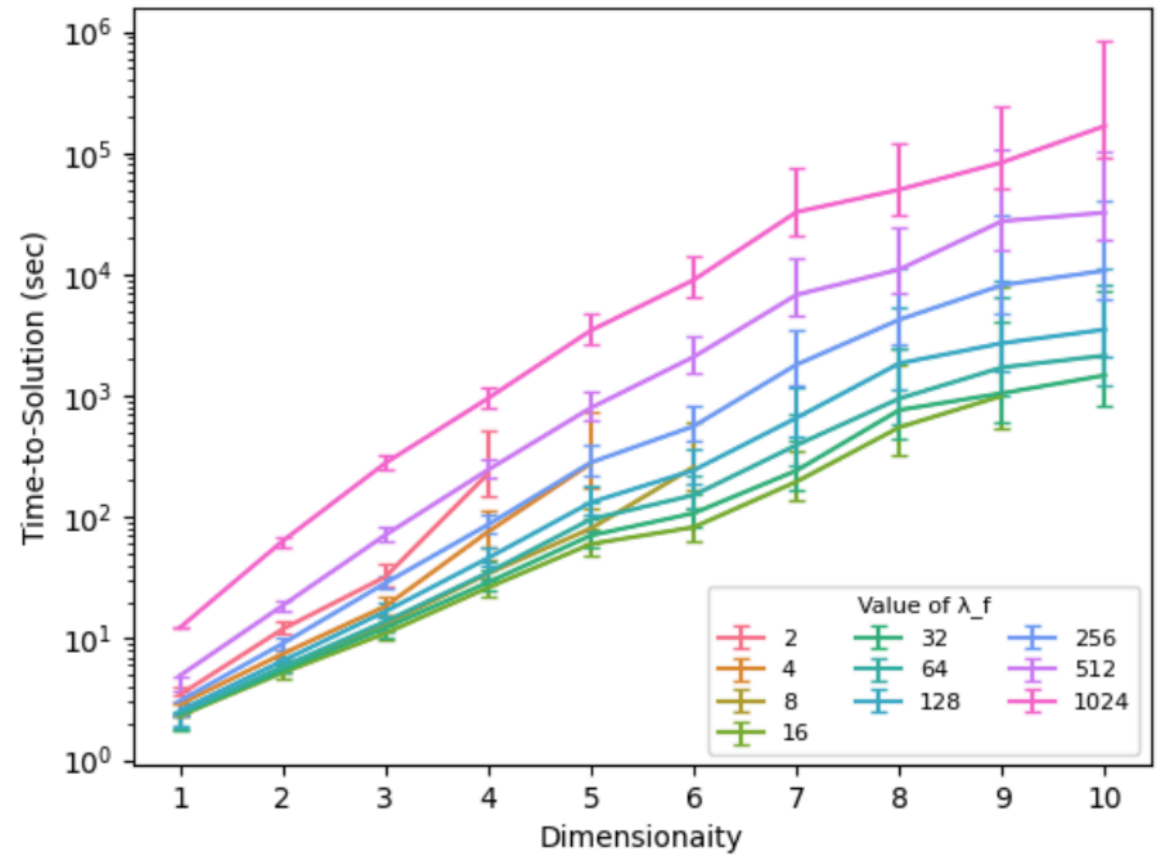
Industrial Leading Branch-and-Bound Solver

TTS scaling of Gurobi which is roughly exponential in dimensionality d. Confidence intervals are calculated at the 95% level. 1 work unit is proportional to 1 second of CPU time.

A quantum-classical performance separation (cont'd)



(a) Fixed learning rate



(b) QHD-type learning rate

TTS for **S**tochastic **G**radient **D**escent (SGD) based on different type of learning rate. Confidence intervals are calculated at the 95% level. It is clear that TTS in both settings (lower envelopes of curves) scale exponentially in dimensionality d.

QHD: Implementation on Quantum Machines

Digital Quantum Machines - assuming coherent access to $f(x)$, the whole algorithm is effectively a time-dependent Hamiltonian simulation with **kinetic energy** terms:

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

which allows a poly-time simulation based on a result from Childs' group. (arXiv: 2203.17006)

Unfortunately, the digital implementation is far from being feasible in near term.

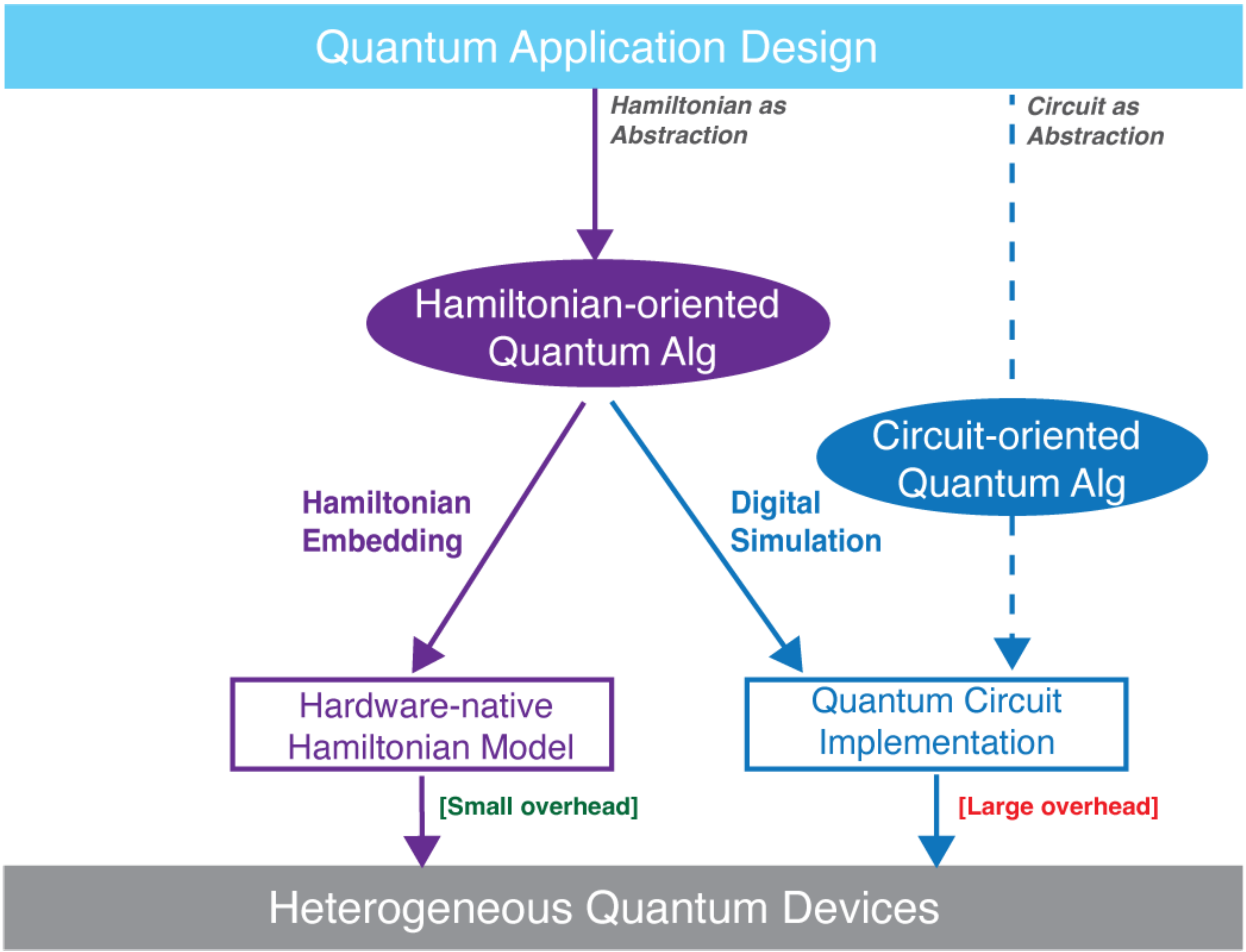
QHD: Hamiltonian Implementation

QHD described as a *time-dependent Hamiltonian evolution*,

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

which makes it possible to directly implement $H(t)$ and **bypass circuit abstraction**.

Hamiltonian-oriented Algorithm Design



QHD: Analog Implementation

Analog Implementation: QHD described as a time-dependent Hamiltonian evolution,

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

which makes it possible to directly implement $H(t)$ and *bypass circuit abstraction*.

Ideally, one would want an **analog** quantum machine designed for H(t) and for high-dimension.

The reality is that no such machine exists. We model an abstract analog machine called **Quantum Ising Machine (QIM)** w/ the following effective Hamiltonian

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

Programmability: (1) initialization to certain quantum state, all-zero or all-|+>

(2) programmable coefficients $h_j, J_{j,k}$ and schedule $A(t), B(t)$

(3) measurement in computational basis

Instantiations of QIM: (1) D-Wave; (2) Rydberg systems (e.g., QuERA)
(3) cold-atom or trapped-ion systems in labs

QHD: Large-scale Empirical Study on Real Machines

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

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

QP — NP-hard with indefinite Q and box constraints.

Benchmark Generation

- **Sparsity of Q** is also required due to the limited qubit connectivity on D-Wave.
- We generate **160 *random QP instances*** (dimensions: 5, 50, 60, 75) with box constraints.

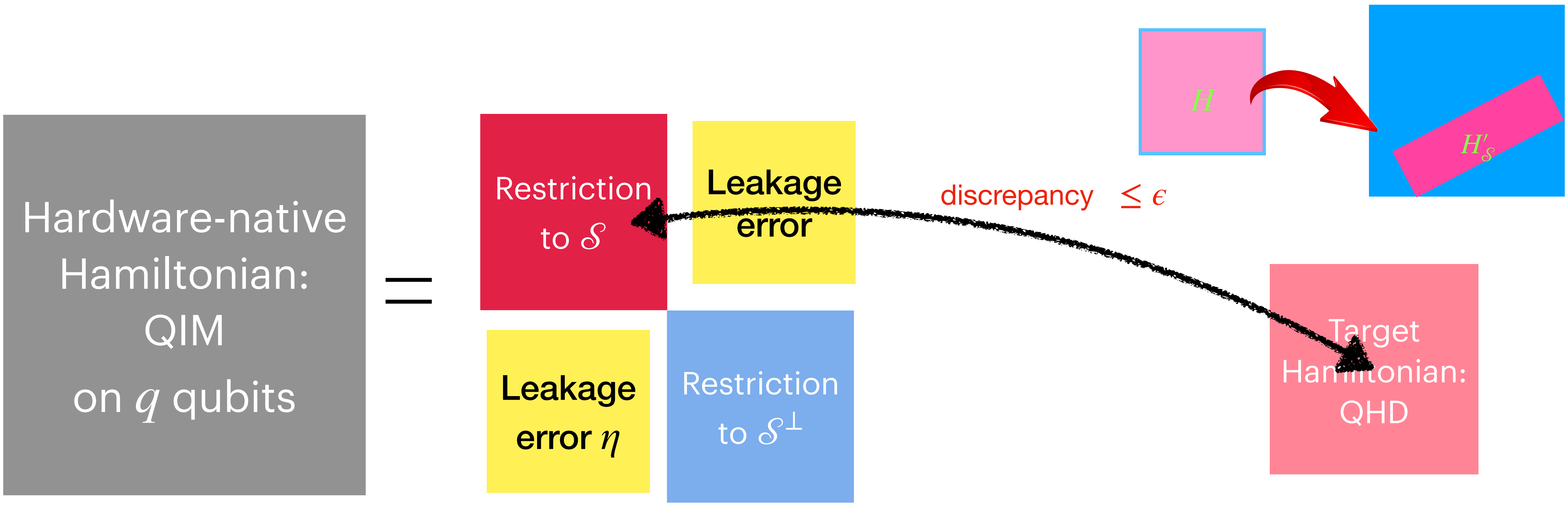
Real-Machine (D-Wave) implementation

$$H_{dwave}(s) = \underbrace{-\frac{A(s)}{2}\left(\sum_j \sigma_x^{(j)}\right)}_{\text{Initial Hamiltonian}} + \underbrace{\frac{B(s)}{2}\left(\sum_j h_j \sigma_z^{(j)} + \sum_{j>k} J_{j,k}\sigma_z^{(j)}\sigma_z^{(k)}\right)}_{\text{Problem Hamiltonian}}$$

- D-Wave Hamiltonian does not look like the QHD Hamiltonian!!
- We propose the **Hamming encoding** scheme to embed QHD into the DW machine Hamiltonian.
- Obtain approximate solution by measuring in the computational basis on DW.

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

Hamiltonian Embedding: math treatment arXiv:2401.08550



1. Find an *invariant* subspace $\mathscr{S}$ of the simulator Hamiltonian.

(Invariance: $Hv \in \mathscr{S}$ for $v \in \mathscr{S}$. Quantum state stays in the subspace in the evolution.

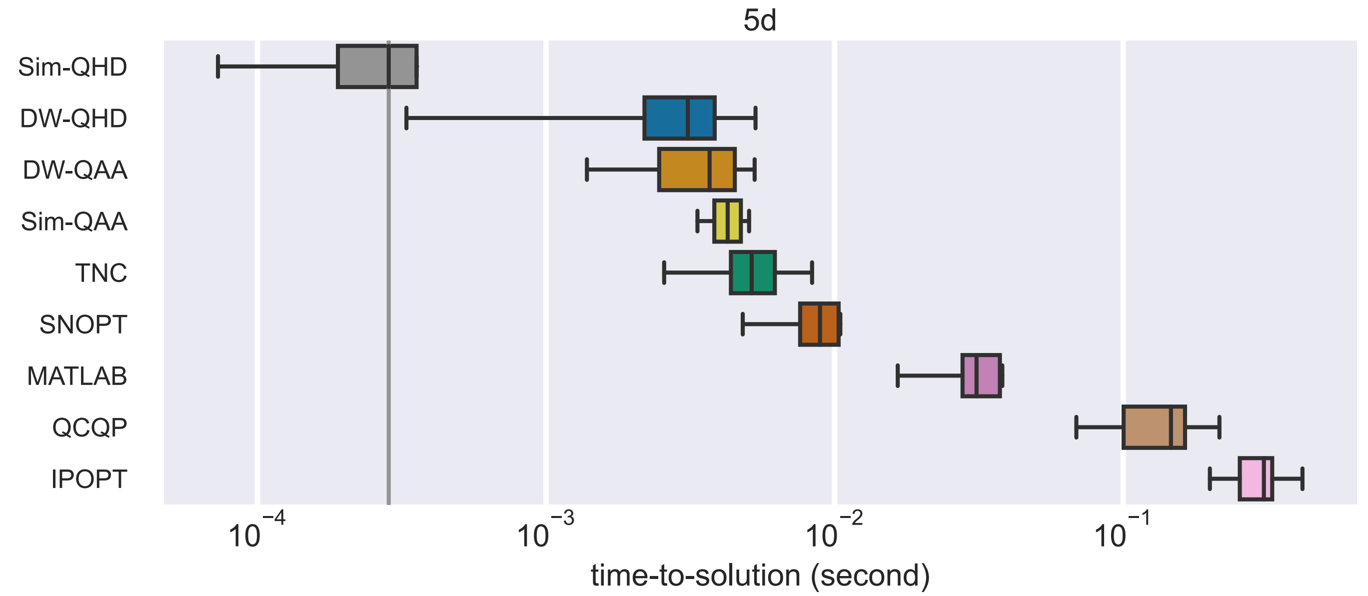
We call $\mathscr{S}$ the *embedding subspace*.

2. The simulator Hamiltonian is block-diagonalized by projecting to $\mathscr{S}$ and $\mathscr{S}^\perp$.

3. Program the simulator Hamiltonian so that its restriction to the invariant subspace matches the target Hamiltonian we want to simulate.

With Hamiltonian embedding, we can simulate QHD in a **sub-system** of the full Quantum Ising Machine.
Measure the QIM in the embedding subspace & **decoding** —> approximate global minimizer of f.

Benchmarking with Quadratic Programming Instances - D5



The evolution time for Sim-QHD/Sim-QAA is 1 μs.
The evolution time for DW-QHD/DW-QAA is 800 μs.

Time-To-Solution (TTS)

The lower, the better!!

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

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

p_g - success probability per run

An Extensive Selection of Solvers

- **DW-QHD/QAA —** DWave implemented QHD/QAA.
- **Sim-QHD/QAA** — Numerically simulated QIM for QHD/QAA (ideal DWave)
- **IPOPT** — SOTA primal-dual interior-point for non-linear optimization.
- **MATLAB** — fmincon - sequential quadratic program.
- **SNOPT** — SOTA large-scale nonlinear optimization using sequential quadratic program routines.
- **TNC** — Truncated Newton. Also used as the post-processing for DW-QHD/QAA to compensate for precision limit. Not very useful by itself.
- **QCQP** — Solve Quadratic Constrained Quadratic Programs using relaxation and local search heuristics.

- **Sim-QHD** is better than the rest, including **DW-QHD,** suggesting that the DWave machine is quite noisy.
- Both Sim-QHD and DW-QHD outperforms QAA, and other classical solvers. QAA has much slower convergence.

Evolution on Random QP Instances D50, D60, D75 with small sparsity

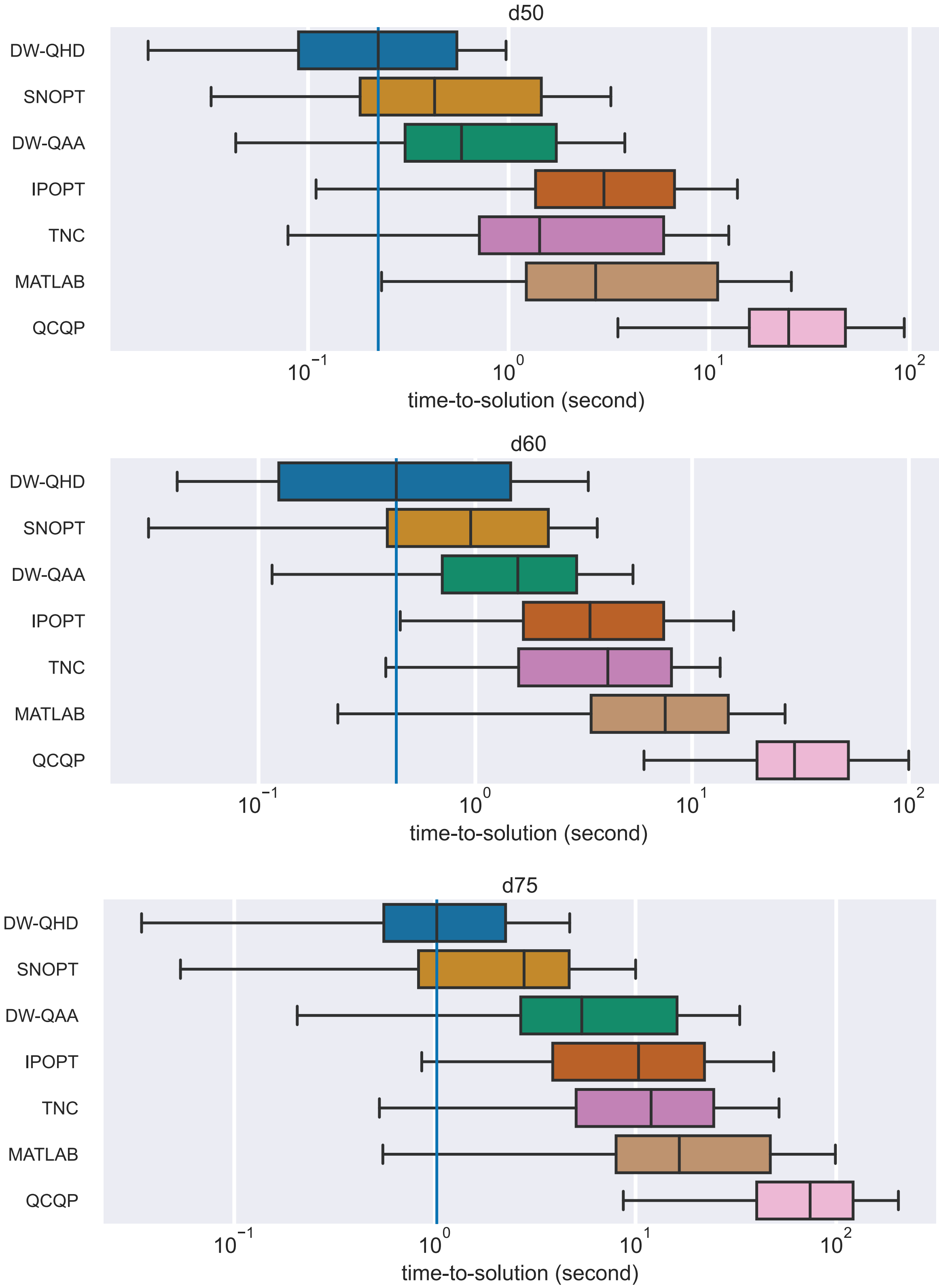


Fig. B/C/E show the success prob metric of 50 randomly generated QP instances for dimension 50, 60, 75. (solid-line: median)

- No classical simulation possible and QP instances need to be sparse due to DWave's constraints.
- **DW- QHD** is still better than the rest, including **DW-QAA**, and **classical GD, interior points,** and **some local search heuristic.**
- Assuming DW-QHD **lower bounds** the performance of QHD, this provides **a very strong empirical evidence supporting QHD**.

QHD does not beat Gurobi on these instances!!

* SOTA *branch-and-bound* algorithms, e.g., Gurobi & CPLEX
These solvers conduct clever enumeration and manage our instances, but should scale exponentially in general.

*** The experimental restriction of D-Wave prevents us from testing more complicated instances in our separation result.**

*** Stay tuned : new experiments solve large sparsity instances w/ Gurobi requiring X hours!!**

Data & Visualization

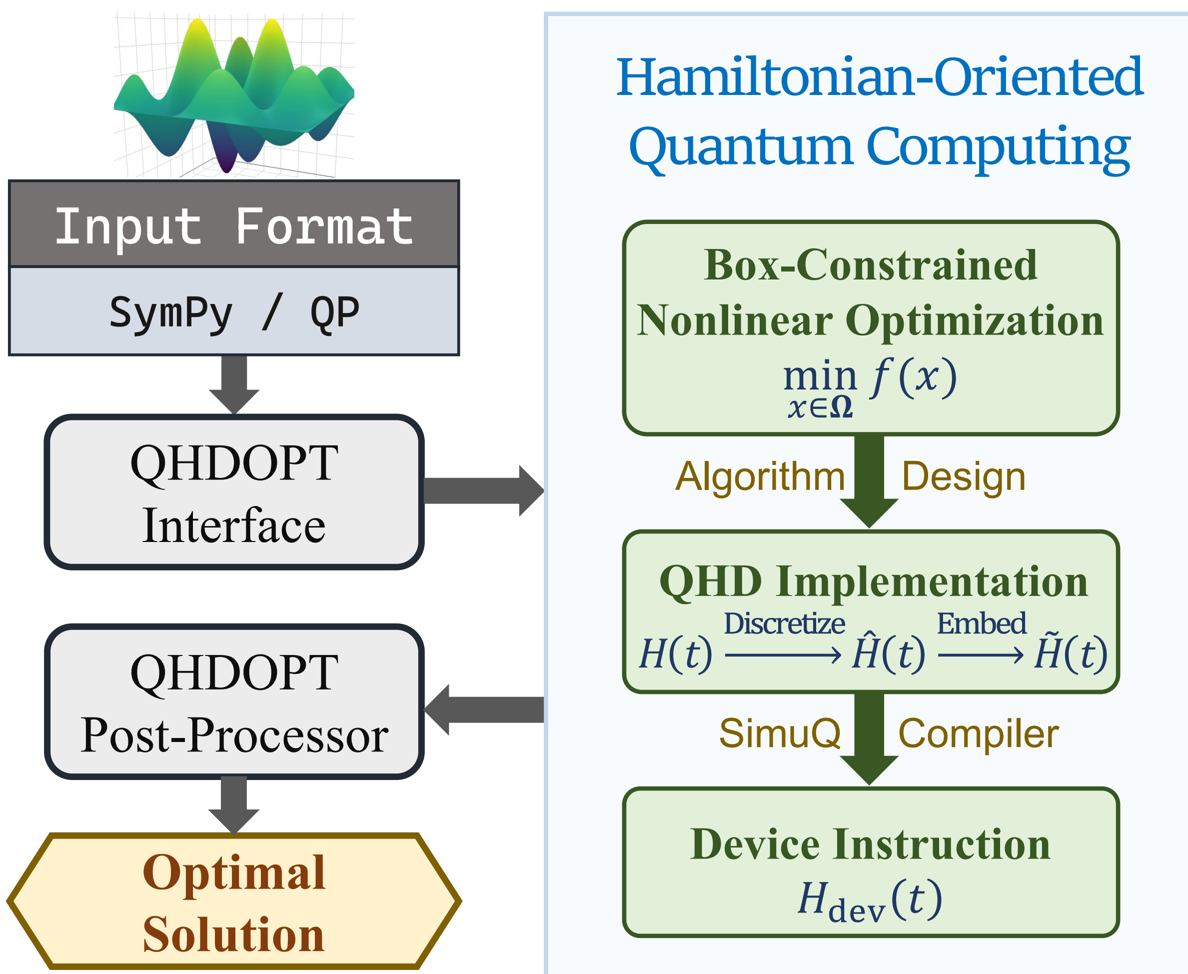
SimuQ Follow-up: QHDOPT

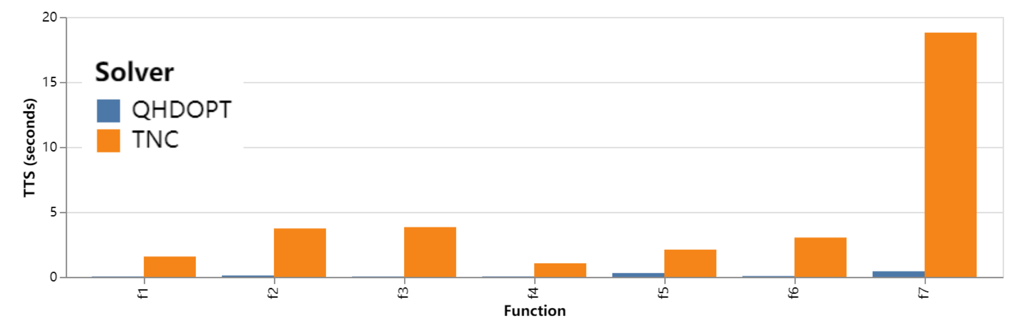
SUPPORTED BY **UNITARY FUND**

Microgrants Co-PI

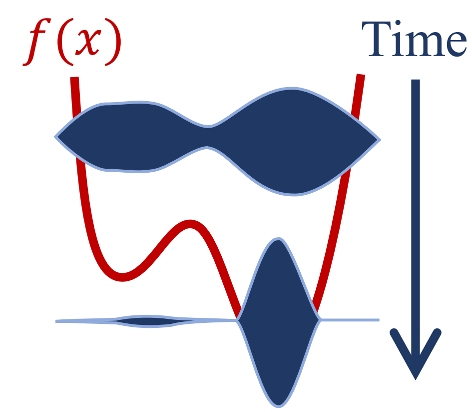
A practical HOQC application

QHDOPT for **nonlinear optimization**

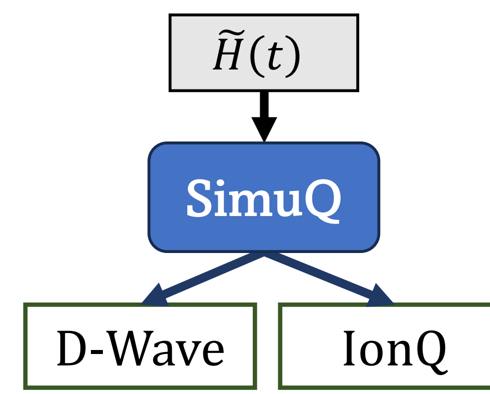




For many nonconvex cases, QHDOPT outperforms classical open-source solvers on wall clock time



Quantum Hamiltonian Descent [LHLW, 23]:
A quantum heavy ball falls to the global minimum



Programming and Deploying with SimuQ

QHD - Summary

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

Summary

- A new proposal of **quantum counterpart of classical gradient methods** with theoretical evidence for quantum/classical performance separation in non-convex optimization.
- QHD is a powerful upgrade of GD and a promising new subroutine.
- Obtain a large scale empirical study with the Hamiltonian-oriented mindset.

Future directions

- Theoretical developments on QHD: e.g., a new quantum interior point method [ALNT**W**'23], identify other non-convex instances for QHD, …
- Implement QHD on *other quantum analog machines: e.g. QuERA, Coherent Ising machines, …*
- Circuit QED co-design for more efficient implementation of QHD.

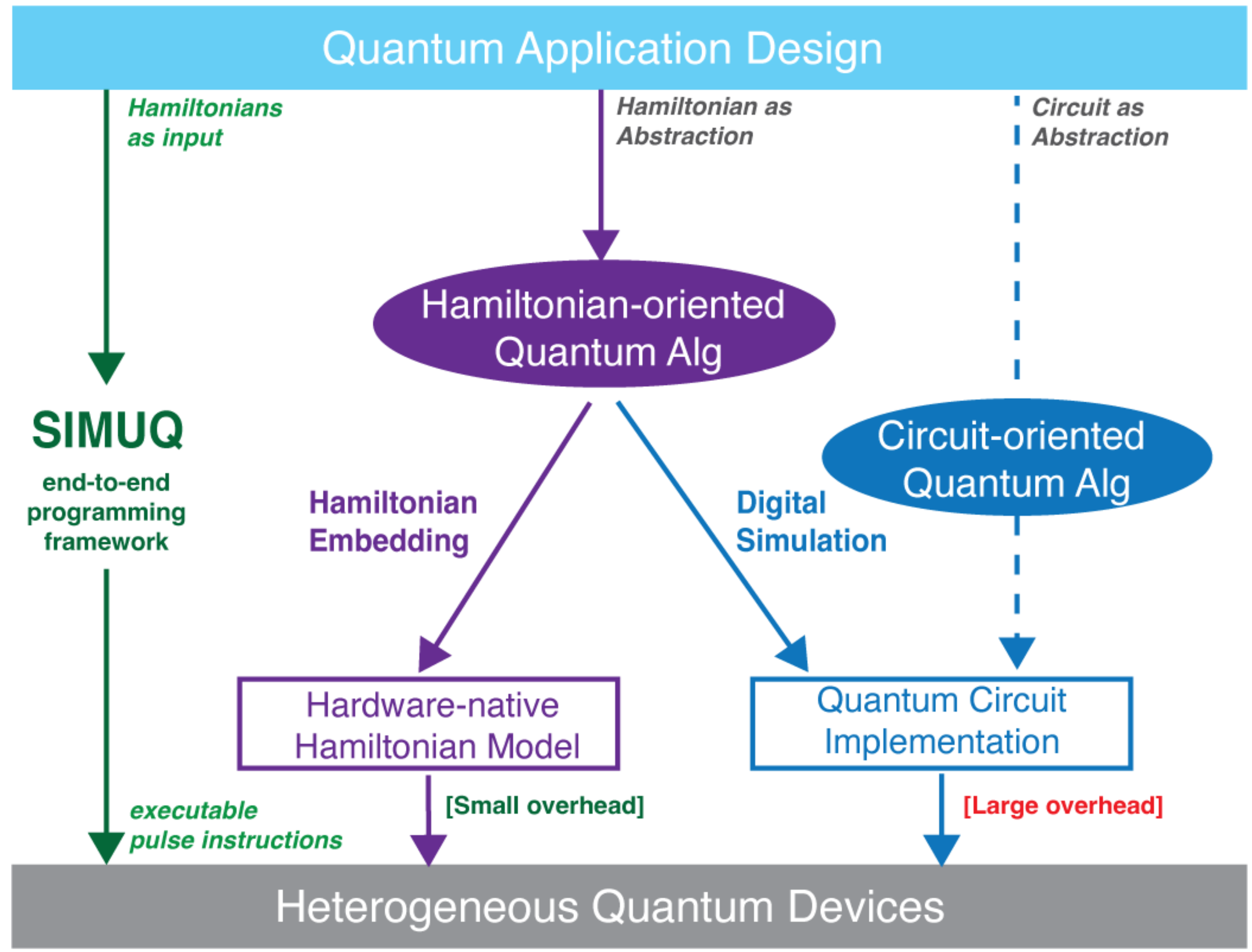
Acknowledgement

Graduate Students

Jacob Young
Ethan Hickman
Haowei Deng
Ethan Lee
Connor Clayton
Yingkang Cao
Joesph Li
Yufan Zheng
Suying Liu
Haohai Shi
Jiaqi Leng (Simons Fellow at UC Berkeley)
Yuxiang Peng (Incoming Asst. Prof. at Purdue)
Xuchen You (Citadel)
Shouvanik Chakrabarti (JP Morgan Quantum)

Postdocs

Liyi Li (2020-2023. Asst. Prof. at Iowa)
Xiong Fan
(2020-2021. Assistant Prof. at Rutgers)
Robert Rand
(2018-2020. Assistant Prof. at U Chicago)



Thank You!