Fast quantum state tomography via accelerated non-convex programming

Joint work with A. Kalev (USC), G. Kollias & K. Wei (IBM), and A. Kyrillidis (Rice)

2021 INFORMS Annual Meeting

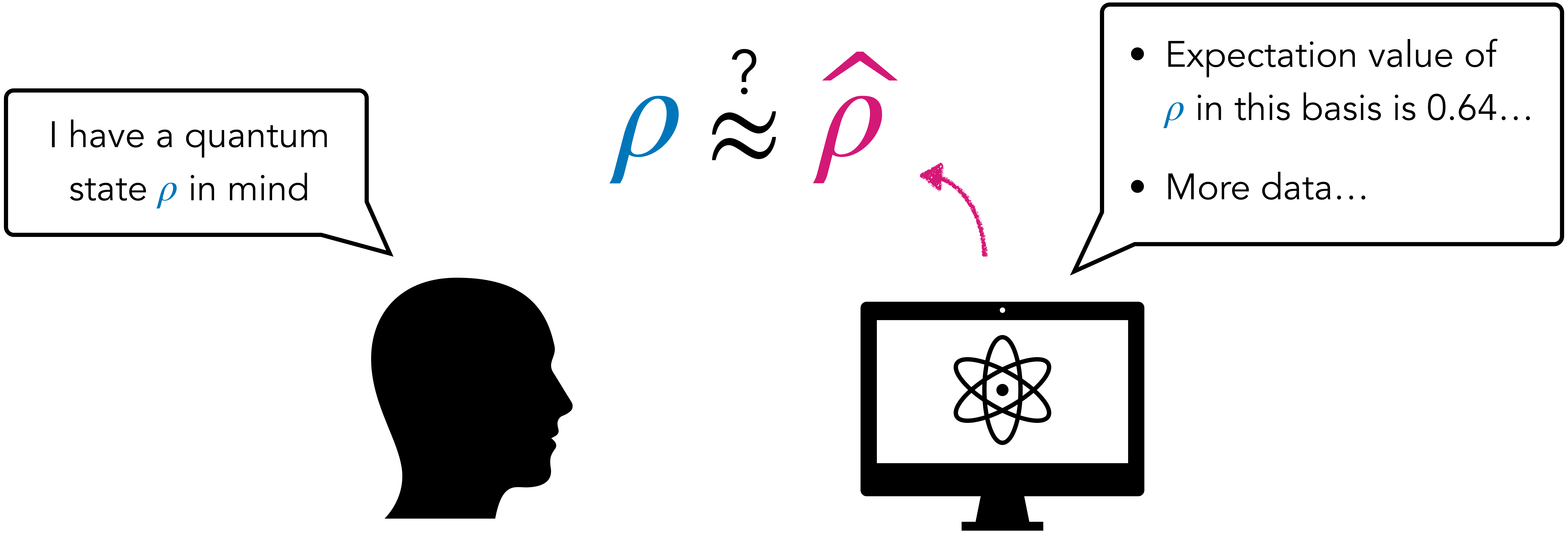
Session: Optimization in Quantum Computing

J. Lyle Kim

Computer Science Department, Rice University

Quantum state tomography (QST)

- Electrical engineers use multimeters and oscilloscopes to verify that circuit works as expected.
- We need similar verification tools in quantum computing. QST is one such tool.
- QST is the task to reconstruct the density matrix of a given quantum state from measurement data.



Quantum state

- We represent quantum bits (qubits) $|0\rangle$ and $|1\rangle$ as vectors:

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}, \quad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

- A state $|\psi\rangle$ can be written as a superposition of $|0\rangle$ and $|1\rangle$, e.g., $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

  Outcome $|1\rangle$ w.p. $|\beta|^2$

- 2-qubit state $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$:

$$|\psi\rangle = \alpha\begin{bmatrix}1\\0\\0\\0\end{bmatrix} + \beta\begin{bmatrix}0\\1\\0\\0\end{bmatrix} + \gamma\begin{bmatrix}0\\0\\1\\0\end{bmatrix} + \delta\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

- A pure state of n qubits can be represented by column vectors in $\mathbb{C}^d$ space with $d = 2^n$

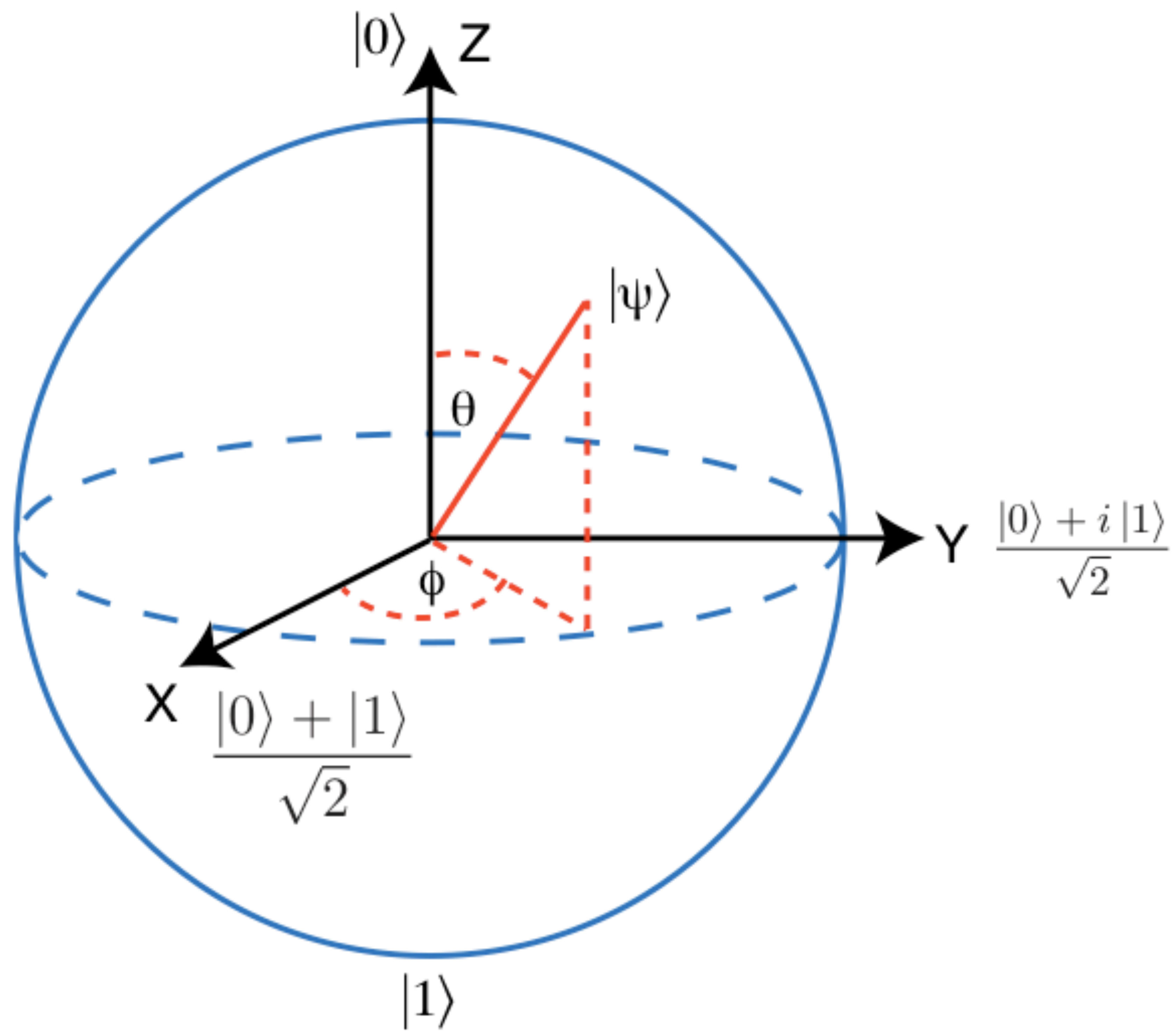


Figure source: https://qiskit.org/textbook/ch-states/introduction.html

Quantum state and density matrix

- $|\psi\rangle$ is a column vector, called "ket"
- $\langle\psi|$ is a row vector, called "bra", with complex conjugates

E.g. $|\psi\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} \end{bmatrix} \rightarrow \langle\psi| = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-i}{\sqrt{2}} \end{bmatrix}$

- Inner product: $\langle\phi|\psi\rangle$ is a number
- Outer product: $|\phi\rangle\langle\psi|$ is a matrix
  - A pure state $|\psi\rangle$ can be written as $\rho = |\psi\rangle\langle\psi|$
  - A mixed state can be written as $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$

"density matrix"

Density matrix ρ

- PSD: $\rho \succeq 0$
- Unit trace: $\mathrm{Tr}(\rho) = 1$

Quantum state tomography (single qubit case)

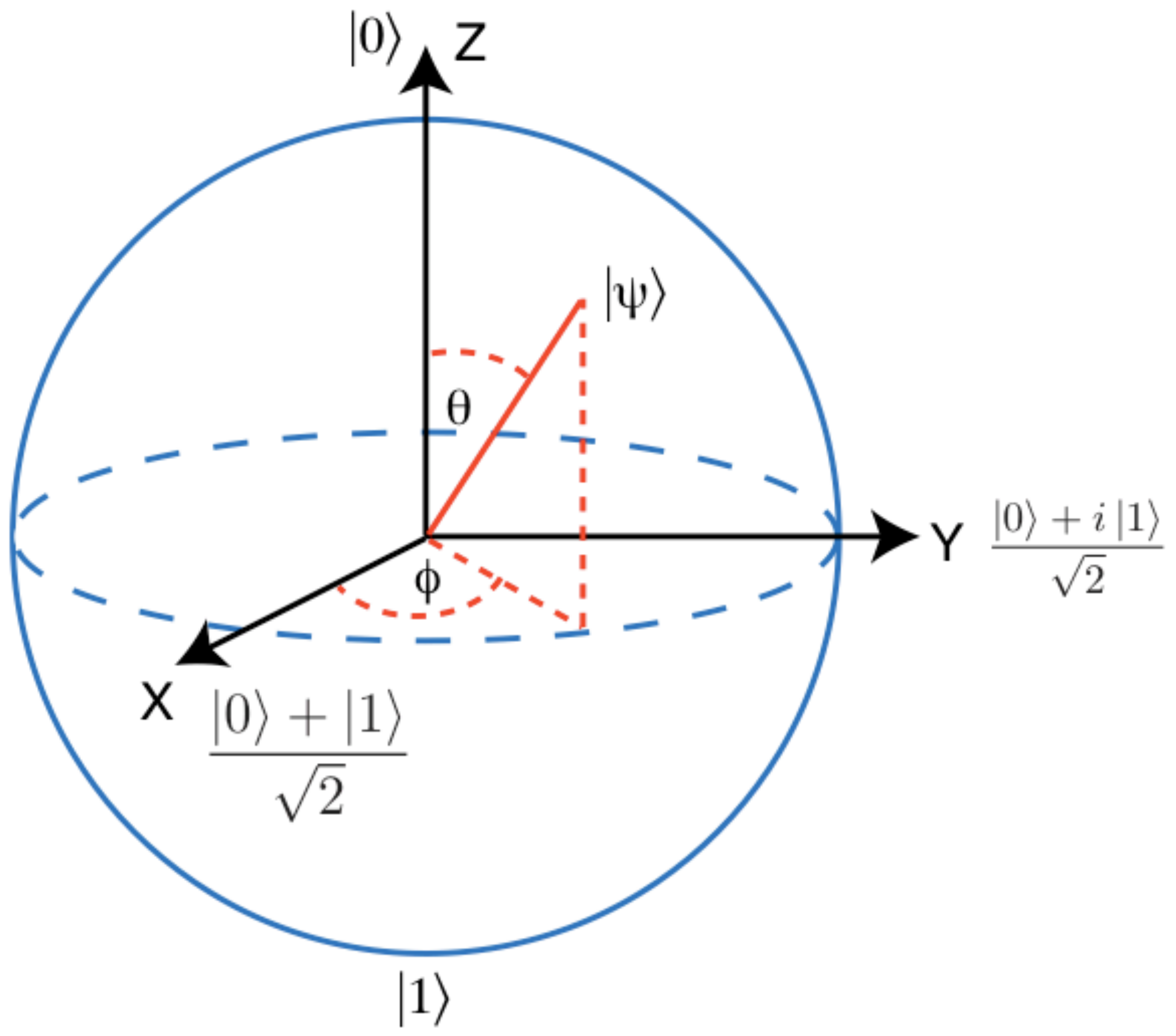
- Any single qubit state can be written as

"Pauli matrices"

$$\rho = \frac{1}{2}\left(I + r_x\sigma_x + r_y\sigma_y + r_z\sigma_z\right) \qquad \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

where $r_\alpha = \mathrm{Tr}(\rho\sigma_\alpha)$, for $\alpha = x, y, z$

Expectation value of σ_α w.r.t ρ



- How do we "measure" $r_\alpha = \mathrm{Tr}(\rho\sigma_\alpha)$?
  - Prepare M number of copies of the state ρ E.g. $M = 1000$
  - Measure the projection of ρ onto eigenvectors of σ_α resulting in $\alpha_1, \alpha_2, \ldots, \alpha_M$
  - Approximation of $\mathrm{Tr}(\rho\sigma_\alpha)$ is given by $\frac{1}{M}\sum_{i=1}^{M} \alpha_i$

Measurement using σ_z, suppose we find the qubit in state $|0_z\rangle$ 400 times, and in state $|1_z\rangle$ 600 times.

We can estimate

$\mathrm{Tr}\left(\rho|0_z\rangle\langle 0_z|\right) \approx \frac{400}{1000} := y_0^z$ and

$\mathrm{Tr}\left(\rho|1_z\rangle\langle 1_z|\right) \approx \frac{600}{1000} := y_1^z$

Quantum state tomography (single qubit case)

- Once we have y_i^α for $i = \{0,1\}$ and $\alpha = \{x, y, z\}$, we can solve:

$$\begin{array}{ll} \underset{\rho \in \mathbb{C}^{d\times d}}{\text{minimize}} & f(\rho) := \sum_{\alpha=x,y,z} \sum_{i=0,1} \left(\mathrm{Tr}(\rho A_i^\alpha) - y_i^\alpha\right)^2 \\ \text{subject to} & \rho \succeq 0,\ \mathrm{Tr}(\rho) = 1 \end{array}$$

$A_i^\alpha = |i_\alpha\rangle\langle i_\alpha|$ "rank-1 sensing matrix"

- More generally we can solve:

$\left(\mathscr{A}(\rho)\right)_i = \mathrm{Tr}(\rho A_i)$ where $A_i \in \mathbb{C}^{d\times d}$, $i = 1,\ldots,m$

$$\begin{array}{ll} \underset{\rho \in \mathbb{C}^{d\times d}}{\text{minimize}} & f(\rho) := \frac{1}{2}\|\mathscr{A}(\rho) - y\|_2^2 \\ \text{subject to} & \rho \succeq 0,\ \mathrm{Tr}(\rho) = 1 \end{array}$$

measured data $y \in \mathbb{R}^m$

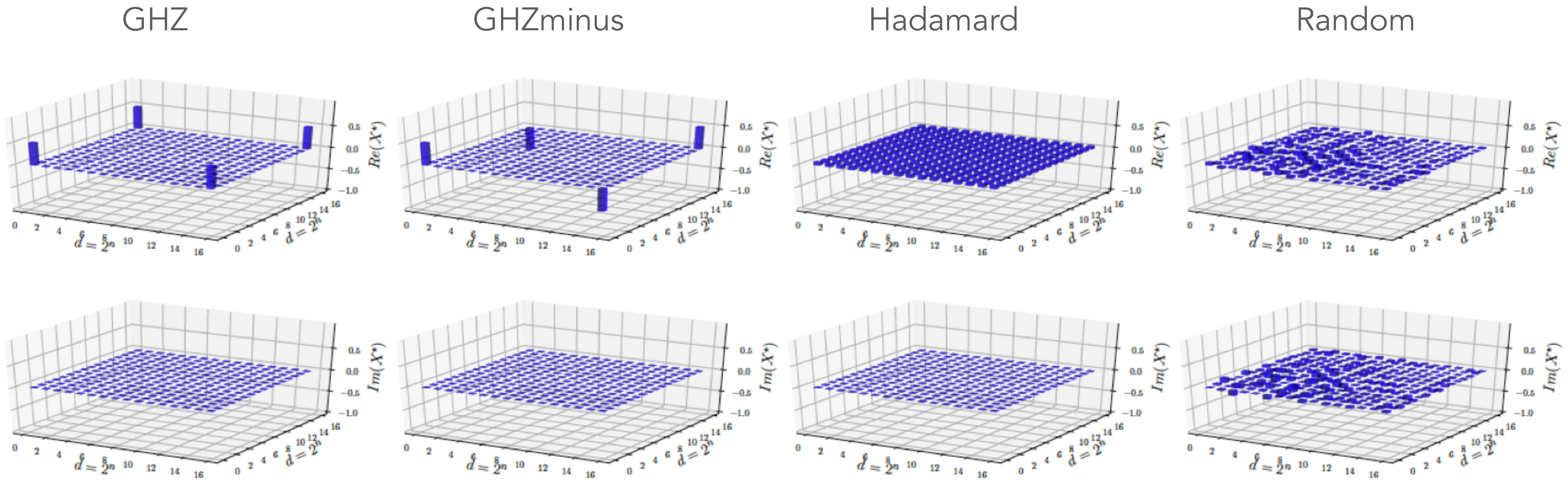
- How does it scale?
  - **Optimization**: the space of $\rho \in \mathbb{C}^{d\times d}$ grows exponentially (recall: $d = 2^n$)
  - **Amount of data**: from $\mathscr{A}(\rho) = y$, if we have access to $y_1, \ldots, y_m$ and $A_1, \ldots, A_m$ that form an orthonormal basis for $\mathbb{C}^{d\times d}$ (i.e. $m = d^2$), we can reconstruct ρ with linear inversion

For $n = 16$ qubits, $\rho \in \mathbb{C}^{d\times d}$ where $d = 65{,}536$

And we need $m = O\left(2^{32}\right) \approx 4{,}294{,}967{,}296$ measurements

- ~~Optimization: $\rho \in \mathbb{C}^{d \times d}$ where $d = 2^n$~~
- Amount of data: $O(d^2)$ without any prior

Structured density matrices



- ~~Optimization: $\rho \in \mathbb{C}^{d\times d}$ where $d = 2^n$~~
- Amount of data: $O(d^2)$ without any prior

Compressed sensing + QST

[Kalev et al., 2015]:

$\mathrm{Tr}(\rho) = 1$ constraint can be ignored without affecting the final estimate

$$\underset{\rho\in\mathbb{C}^{d\times d}}{\text{minimize}} \quad f(\rho) := \frac{1}{2}\|\mathscr{A}(\rho) - y\|_2^2$$

$$\text{subject to} \quad \boxed{\text{rank}(\rho) \leq r,} \ \rho \succeq 0, \ \cancel{\mathrm{Tr}(\rho) = 1}$$

Restricted Isometry Property (RIP) for rank-r matrices

[B. Recht et al., 2010]

A linear operator $\mathscr{A} : \mathbb{C}^{d\times d} \to \mathbb{R}^m$ satisfies the RIP on rank-r matrices, with parameter $\delta_{2r} \in (0,1)$, if the following holds for any rank-r matrix $X \in \mathbb{C}^{d\times d}$, with high probability:

$$(1 - \delta_{2r}) \cdot \|X_1 - X_2\|_F^2 \leq \|\mathscr{A}(X_1 - X_2)\|_2^2 \leq (1 + \delta_{2r}) \cdot \|X_1 - X_2\|_F^2.$$

[D. Gross et al., 2010]: can reconstruct rank-r density matrix $\rho \in \mathbb{C}^{d\times d}$ using $O\left(r \cdot d \cdot \text{poly}(\log d)\right)$ measurements

[Y.K. Liu, 2010]: $P_i \in \{I, \sigma_x, \sigma_y, \sigma_z\}^{\otimes n}$ satisfies RIP for rank-r matrices

- Optimization: $\rho \in \mathbb{C}^{d\times d}$ where $d = 2^n$
- ~~Amount of data: $\mathcal{O}(d^2)$ without any prior~~

Factorized objective and MiFGD

$$\underset{\rho \in \mathbb{C}^{d\times d}}{\text{minimize}} \quad f(\rho) := \frac{1}{2}\|\mathcal{A}(\rho) - y\|_2^2$$

$$\text{subject to} \quad \cancel{\rho \succeq 0, \ \text{rank}(\rho) \leq r} \quad \longleftarrow \quad \rho = UU^\dagger$$

Convex constraint

Non-convex constraint

$$\underset{U \in \mathbb{C}^{d\times r}}{\text{minimize}} \quad f(UU^\dagger) := \frac{1}{2}\|\mathcal{A}(UU^\dagger) - y\|_2^2$$

Smaller space ($\mathbb{C}^{d\times r}$)
than original space ($\mathbb{C}^{d\times d}$)

Constraints automatically satisfied

Factored Gradient Descent

[Kyrillidis et al., 2019]

$$\begin{aligned} U_{i+1} &= U_i - \eta \nabla f(U_i U_i^\dagger) \cdot U_i \\ &= U_i - \eta \mathcal{A}^\dagger \left(\mathcal{A}(U_i U_i^\dagger) - y \right) \cdot U_i \end{aligned}$$

Momentum-inspired Factored Gradient Descent

$$\begin{aligned} U_{i+1} &= Z_i - \eta \mathcal{A}^\dagger \left(\mathcal{A}(Z_i Z_i^\dagger) - y \right) \cdot Z_i \\ Z_{i+1} &= U_{i+1} + \mu \left(U_{i+1} - U_i \right) \end{aligned}$$

- Optimization: $\rho \in \mathbb{C}^{d\times d}$ where $d = 2^n$
- ~~Amount of data: $O(d^2)$ without any prior~~

Convergence theory

Theorem 3 (Accelerated convergence rate). *Assume that $\mathcal{A}$ satisfies the RIP with constant $\delta_{2r} \leq 1/10$. Let U_0 and U_{-1} be such that* $\boxed{\min_{R\in\mathcal{O}} \|U_0 - U^\star R\|_F,\ \min_{R\in\mathcal{O}} \|U_{-1} - U^\star R\|_F \leq \frac{\sqrt{\sigma_r(\rho^\star)}}{10^3\sqrt{\kappa\tau(\rho^\star)}}}$ *where $\kappa := \frac{1+\delta_{2r}}{1-\delta_{2r}}$, $\tau(\rho) := \frac{\sigma_1(\rho)}{\sigma_r(\rho)}$ for rank-r ρ, and $\sigma_i(\rho)$ is the ith singular value of ρ. Set step size η such that*

$$\boxed{\left[1 - \left(\frac{\sqrt{1+\delta_{2r}} - \sqrt{1-\delta_{2r}}}{(\sqrt{2}+1)\sqrt{1+\delta_{2r}}}\right)^4\right] \cdot \frac{10}{4\sigma_r(\rho^\star)(1-\delta_{2r})} \leq \eta \leq \frac{10}{4\sigma_r(\rho^\star)(1-\delta_{2r})},}$$

and the momentum parameter $\boxed{\mu = \frac{\varepsilon}{2\cdot 10^3 r\tau(\rho^\star)\sqrt{\kappa}},}$ *for user-defined $\varepsilon \in (0,1]$. For $y = \mathcal{A}(\rho^\star)$ where $\texttt{rank}(\rho^\star) = r$, $\texttt{MiFGD}$ returns a solution such that*

$$\min_{R\in\mathcal{O}} \|U_{J+1} - U^\star R\|_F \leq \left(1 - \sqrt{\frac{1-\delta_{2r}}{1+\delta_{2r}}}\right)^{J+1} \left(\min_{R\in\mathcal{O}} \|U_0 - U^\star R\|_F^2 + \min_{R\in\mathcal{O}} \|U_{-1} - U^\star R\|_F^2\right)^{1/2}$$

$(1-0.25)^6 \approx 0.1779$ VS. $\left(1-\sqrt{0.25}\right)^6 \approx 0.0156$

$$+ \xi \cdot |\mu| \cdot \sigma_1(\rho^\star)^{1/2} \cdot r \cdot \left(1 - \left(1 - \sqrt{\frac{1-\delta_{2r}}{1+\delta_{2r}}}\right)^{J+1}\right)\left(1 - \sqrt{\frac{1-\delta_{2r}}{1+\delta_{2r}}}\right)^{-1}$$

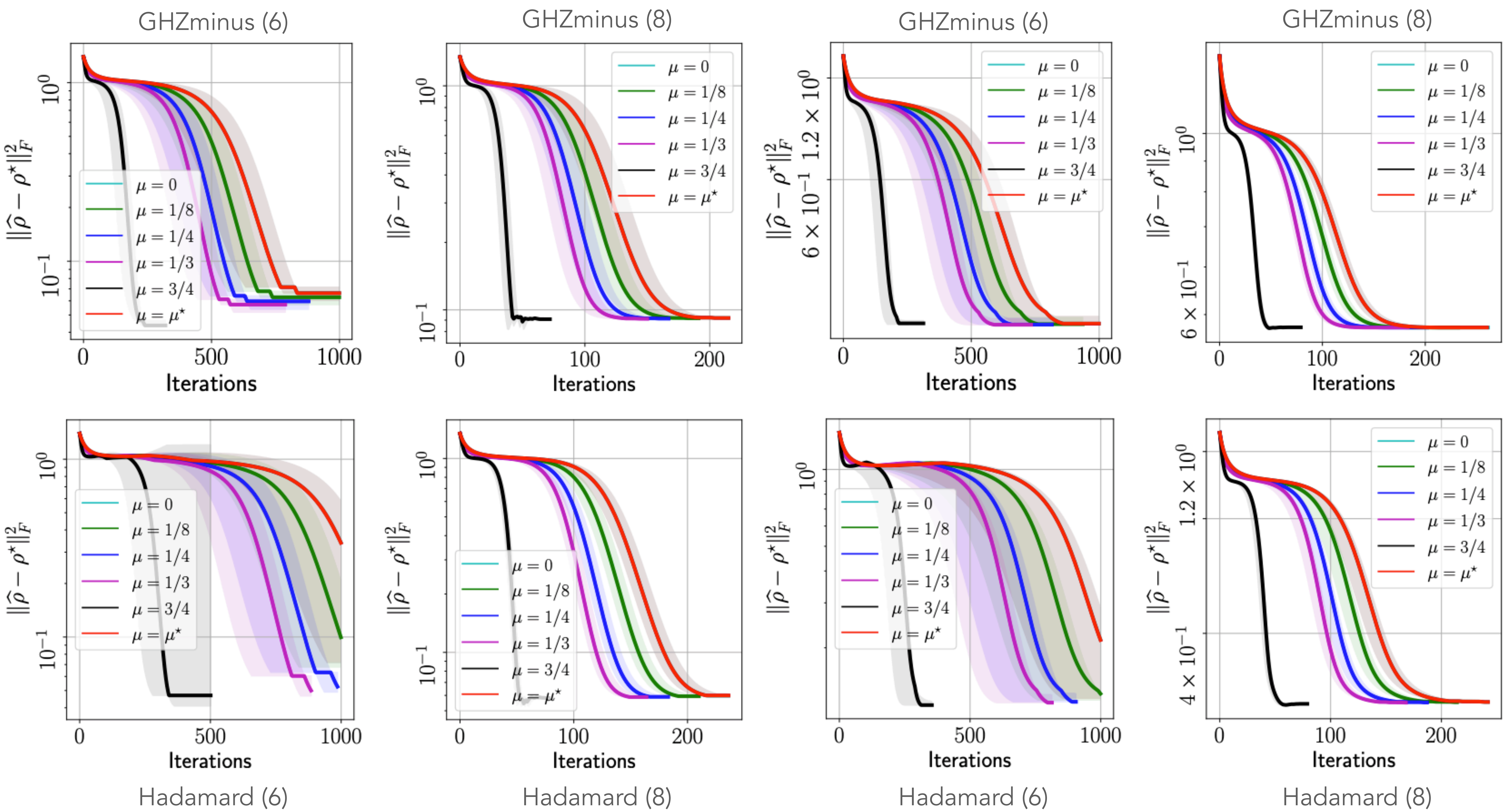
$\left(1 - \frac{1-\delta_{2r}}{1+\delta_{2r}}\right)^{J+1}$ VS.

$$\lessapprox \boxed{\left(1 - \sqrt{\frac{1-\delta_{2r}}{1+\delta_{2r}}}\right)^{J+1} \left(\min_{R\in\mathcal{O}} \|U_0 - U^\star R\|_F^2 + \min_{R\in\mathcal{O}} \|U_{-1} - U^\star R\|_F^2\right)^{1/2} + O(\mu),}$$

where $\xi = \sqrt{1 - \frac{4\eta\sigma_r(\rho^\star)(1-\delta_{2r})}{10}}$. That is, the algorithm has an accelerated linear convergence rate in iterate distances up to a constant proportional to the momentum parameter μ.

Effect of quantum hardware noise

Simulated data
(IBM simulator)



Real data
(IBM QPU)

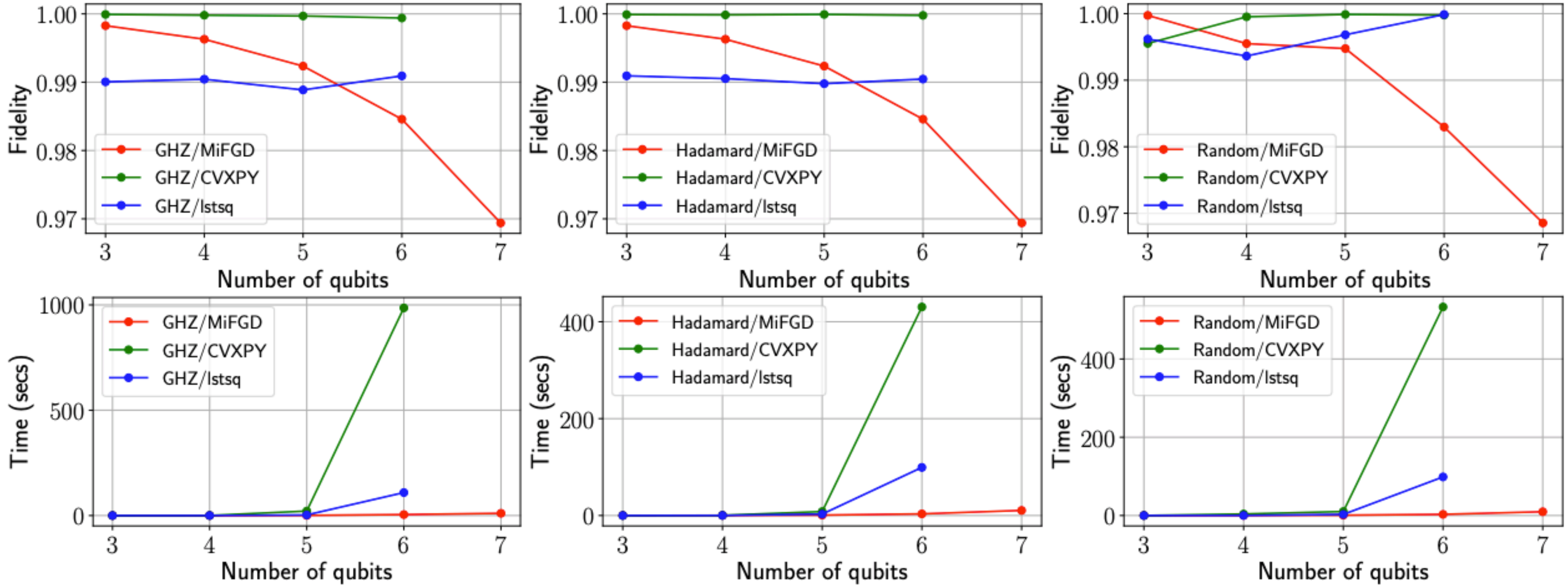
$[m = 20\% \cdot d^2]$

Comparison with `Qiskit`

$$\underset{\rho \in \mathbb{C}^{d \times d}}{\text{minimize}} \quad f(\rho) := \frac{1}{2}\|\mathscr{A}(\rho) - y\|_2^2$$

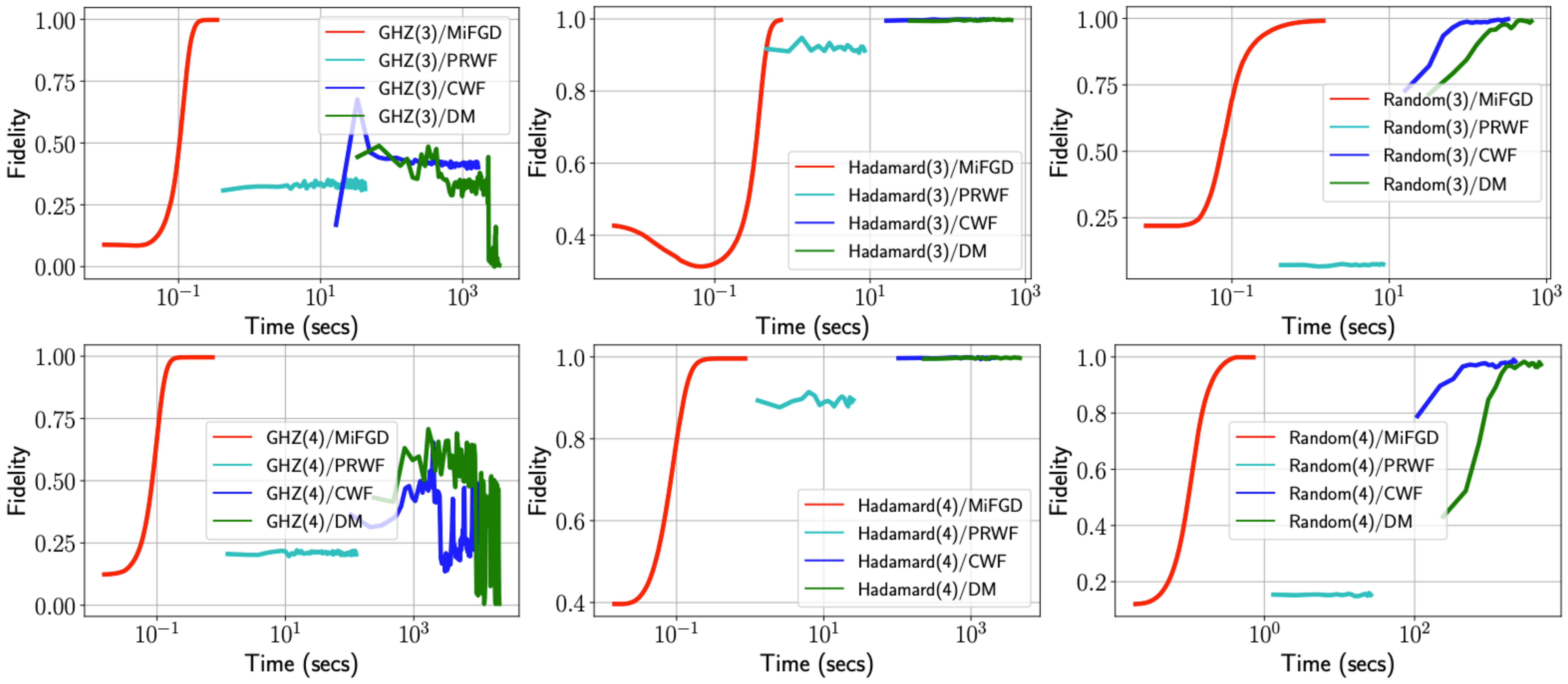
$$\text{subject to} \quad \rho \succeq 0,\ \mathrm{Tr}(\rho) = 1$$

$\text{Fidelity}(\hat{\rho}, \rho) := \mathrm{Tr}(\hat{\rho}\rho)$



Comparison with SOTA: `Qucumber` NN methods

[Torlai et al., 2018]



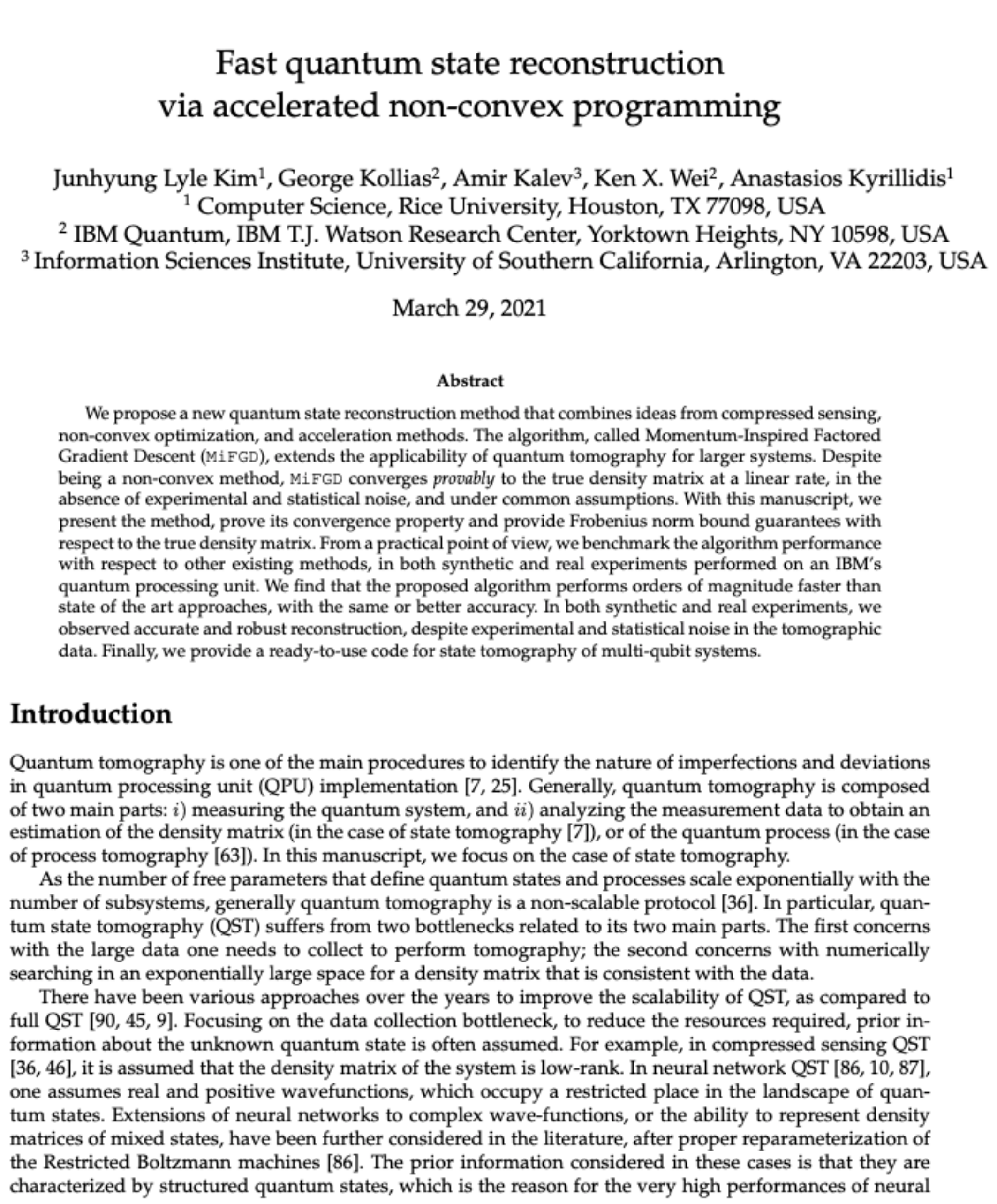
$[m = 50\% \cdot d^2]$

Comparison with SOTA: `Qucumber` NN methods

Circuit		Method			
		MiFGD	PRWF	CWF	DM
GHZ(7)	Fidelity	0.969174	0.058387	0.080648	N/A
	Time (secs)	6.174129	3633.082	$>$ 3h	$>$ 3h
Hadamard(7)	Fidelity	0.969156	0.818174	0.996586	N/A
	Time (secs)	6.324469	713.9404	$>$ 3h	$>$ 3h
Random(7)	Fidelity	0.967640	0.141745	0.06568	N/A
	Time (secs)	6.802577	746.2630	$>$ 3h	$>$ 3h
GHZ(8)	Fidelity	0.940601	0.0400391	N/A	N/A
	Time (secs)	21.16011	$>$ 3h	$>$ 3h	$>$ 3h
Hadamard(8)	Fidelity	0.940638	0.794892	N/A	N/A
	Time (secs)	22.30246	2344.796	$>$ 3h	$>$ 3h
Random(8)	Fidelity	0.939418	0.050521	N/A	N/A
	Time (secs)	22.81059	2196.259	$>$ 3h	$>$ 3h

Summary

Theory



Fast quantum state reconstruction
via accelerated non-convex programming

Junhyung Lyle Kim[1], George Kollias[2], Amir Kalev[3], Ken X. Wei[2], Anastasios Kyrillidis[1]
[1] Computer Science, Rice University, Houston, TX 77098, USA
[2] IBM Quantum, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
[3] Information Sciences Institute, University of Southern California, Arlington, VA 22203, USA

March 29, 2021

Abstract

We propose a new quantum state reconstruction method that combines ideas from compressed sensing, non-convex optimization, and acceleration methods. The algorithm, called Momentum-Inspired Factored Gradient Descent (`MiFGD`), extends the applicability of quantum tomography for larger systems. Despite being a non-convex method, `MiFGD` converges *provably* to the true density matrix at a linear rate, in the absence of experimental and statistical noise, and under common assumptions. With this manuscript, we present the method, prove its convergence property and provide Frobenius norm bound guarantees with respect to the true density matrix. From a practical point of view, we benchmark the algorithm performance with respect to other existing methods, in both synthetic and real experiments performed on an IBM's quantum processing unit. We find that the proposed algorithm performs orders of magnitude faster than state of the art approaches, with the same or better accuracy. In both synthetic and real experiments, we observed accurate and robust reconstruction, despite experimental and statistical noise in the tomographic data. Finally, we provide a ready-to-use code for state tomography of multi-qubit systems.

Introduction

Quantum tomography is one of the main procedures to identify the nature of imperfections and deviations in quantum processing unit (QPU) implementation [7, 25]. Generally, quantum tomography is composed of two main parts: *i*) measuring the quantum system, and *ii*) analyzing the measurement data to obtain an estimation of the density matrix (in the case of state tomography [7]), or of the quantum process (in the case of process tomography [63]). In this manuscript, we focus on the case of state tomography.

As the number of free parameters that define quantum states and processes scale exponentially with the number of subsystems, generally quantum tomography is a non-scalable protocol [36]. In particular, quantum state tomography (QST) suffers from two bottlenecks related to its two main parts. The first concerns with the large data one needs to collect to perform tomography; the second concerns with numerically searching in an exponentially large space for a density matrix that is consistent with the data.

There have been various approaches over the years to improve the scalability of QST, as compared to full QST [90, 45, 9]. Focusing on the data collection bottleneck, to reduce the resources required, prior information about the unknown quantum state is often assumed. For example, in compressed sensing QST [36, 46], it is assumed that the density matrix of the system is low-rank. In neural network QST [86, 10, 87], one assumes real and positive wavefunctions, which occupy a restricted place in the landscape of quantum states. Extensions of neural networks to complex wave-functions, or the ability to represent density matrices of mixed states, have been further considered in the literature, after proper reparameterization of the Restricted Boltzmann machines [86]. The prior information considered in these cases is that they are characterized by structured quantum states, which is the reason for the very high performances of neural

- Non-convex
- Low-rank factorization
- Acceleration

Software

```python
import numpy as np
from mpi4py import MPI
from qiskit.tools.qi.qi import state_fidelity

import projectors
import measurements


import itertools
import sys

############################################################
## Parallel projFGD base class
## XXX WIP
############################################################
class Worker:
    def __init__(self,
                 process_idx,
                 num_processes,
                 params_dict):

        projector_store_path     = params_dict.get('projector_store_path', None)
        num_iterations           = params_dict['num_iterations']
        eta                      = params_dict['eta']

        beta                     = params_dict.get('beta', None)
        trace                    = params_dict.get('trace', 1.0)
        target_state             = params_dict.get('target_state', None)
        convergence_check_period = params_dict.get('convergence_check_period', 10)
        relative_error_tolerance = params_dict.get('relative_error_tolerance', 0.0001)

        parity_flavor            = params_dict.get('parity_flavor', 'effective')

        pauli_correlation_measurements_fpath = params_dict.get('pauli_correlation_measurements_fpath',
                                                                None)

        measurement_store_path   = params_dict.get('measurement_store_path', None)
        tomography_labels        = params_dict.get('tomography_labels', None)
        density_matrix           = params_dict.get('density_matrix', None)

        label_format             = params_dict.get('label_format', 'big_endian')

        store_load_batch_size    = params_dict.get('store_load_batch_size', 1000)
        debug                    = params_dict.get('debug', True)

        seed                     = params_dict.get('seed', 0)

        mu                       = params_dict.get('mu', 0.0)


        if tomography_labels is None:
            tomography_labels  = measurements.MeasurementStore.load_labels(measurement_store_path)

        num_tomography_labels = len(tomography_labels)
        label_list            = split_list(tomography_labels, num_processes)[process_idx]
        num_labels            = len(label_list)
        # print(num_labels, process_idx)

        if projector_store_path is not None:
            # load projectors in batches
            projector_dict = {}
            start = 0
            end   = min(num_labels,  store_load_batch_size)
```

Qiskit

https://github.com/gidiko/MiFGD

Real quantum data