Phase Retrieval Using Unitary 2-Designs

Shelby Kimmel[1,2] and Yi-Kai Liu[1,3]

1. Joint Center for Quantum Info and Computer Science (QuICS), University of Maryland
2. Middlebury College
3. National Institute of Standars and Technology (NIST)

Joint Center for Quantum Information and Computer Science

SAMPTA
04/07/2017

Familiar Problem: Phase Retrieval

Phase Retrieval:

Learn unknown signal $x \in \mathbb{C}^d$, given noisy quadratic measurements:

$$y_i = |a_i^* x|^2 + \epsilon_i$$

Where $a_i \in \mathbb{C}^d$ are chosen by observer, ϵ_i are unknown noise, using as few measurement settings as possible.

Variant: Phase Retrieval using Unitaries

Phase Retrieval:

Learn unknown signal $x \in \mathbb{C}^d$, given noisy quadratic measurements:

$$y_i = |a_i^* x|^2 + \epsilon_i$$

Where $a_i \in \mathbb{C}^d$ are chosen by observer, ϵ_i are unknown noise, using as few measurement settings as possible.

Phase Retrieval using Unitaries:

Learn unknown unitary matrix $U \in \mathbb{C}^{d \times d}$, given noisy quadratic measurements:

$$y_i = |Tr(C_i^* U)\ |^2 + \epsilon_i$$

Where $C_i \in \mathbb{C}^{d \times d}$ are unitary matrices chosen by observer, ϵ_i are unknown noise, using as few measurement settings as possible.

Variant: Phase Retrieval using Unitaries

Phase Retrieval:

Learn unknown signal $x \in \mathbb{C}^d$, given noisy quadratic measurements:

$$y_i = |a_i^* x|^2 + \epsilon_i$$

Where $a_i \in \mathbb{C}^d$ are chosen by observer, ϵ_i are unknown noise, using as few measurement settings as possible.

Phase Retrieval using Unitaries:

Learn unknown unitary matrix $U \in \mathbb{C}^{d \times d}$, given noisy quadratic measurements:

$$y_i = |Tr(C_i^* U)\ |^2 + \epsilon_i$$

Where $C_i \in \mathbb{C}^{d \times d}$ are unitary matrices chosen by observer, ϵ_i are unknown noise, using as few measurement settings as possible.

N.B.:

- $|Tr(C_\mathrm{i}^* U)\ |^2 = |\mathrm{vec}(C_i)^* \mathrm{vec}(U)|^2$
- C_i^* is conjugate transpose of C_i

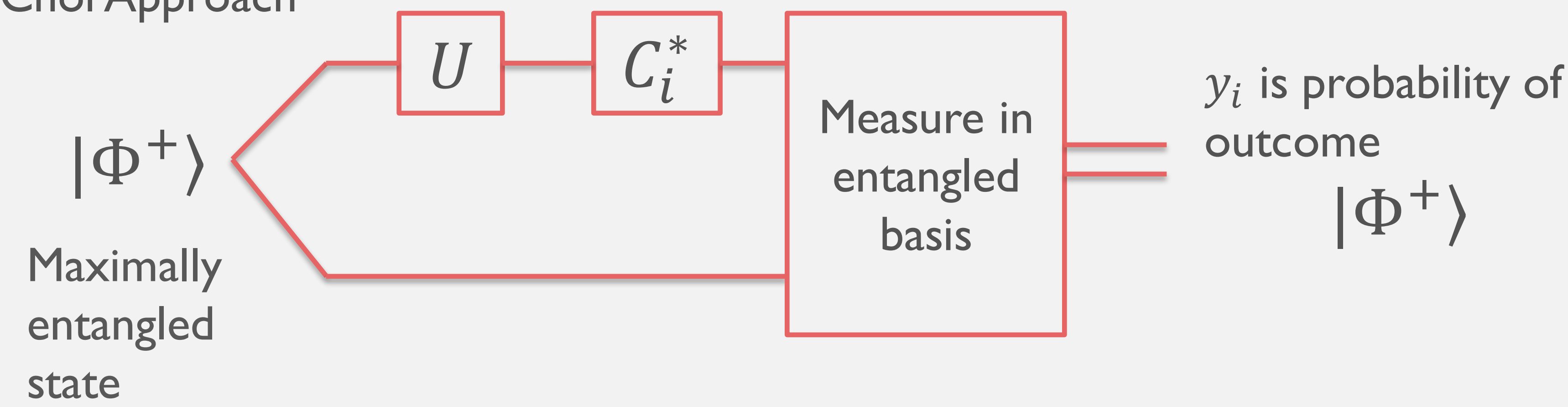
Why Phase Retrieval Using Unitaries?

- Unitaries are basic building blocks of a quantum computer
- Physical implementations often not correct – need to find errors.

How Phase Retrieval Using Unitaries?

Measurement Schemes to obtain $y_i = |Tr(C_i^* U)|^2 + \epsilon_i$

1. Choi Approach

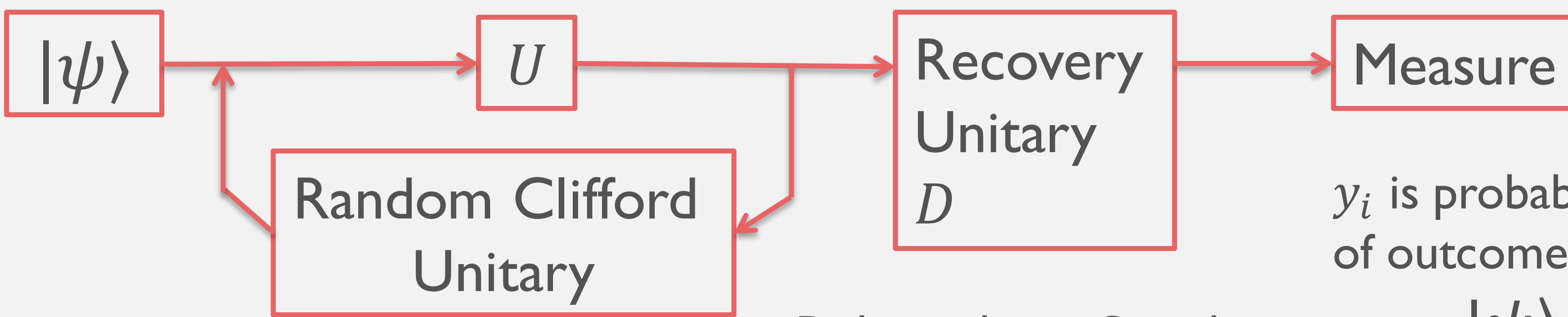


- Difficult to prepare entangled state and measure in entangled basis
- Can't characterize unitaries acting on full system.

How Phase Retrieval Using Unitaries?

Measurement Schemes to obtain $y_i = |Tr(C_i^* U)|^2 + \epsilon_i$

2. Randomized Benchmarking



D depends on C_i and sequence of random unitaries

y_i is probability of outcome $|\psi\rangle$

- Good: inherently protected from SPAM errors, no entanglement needed
- Bad: C_i must be a Clifford Unitary.

Phase retrieval possible when C_i chosen from Cliffords?

Our Results:

How does our choice of C_i affect our ability to learn U?

Our Results:

How does our choice of C_i affect our ability to learn U?

1. Phase retrieval of all unitary matrices, when C_i chosen from a unitary 4-design

2. Phase retrieval works pretty well when C_i chosen from a unitary 2-design.

Our Results:

How does our choice of C_i affect our ability to learn U?

1. Phase retrieval of all unitary matrices, when C_i chosen from a unitary 4-design

   Cliffords form a unitary 3-design! [Zhu; Webb; Kueng and Gross, 2015]

2. Phase retrieval works pretty well when C_i chosen from a unitary 2-design.

Our Results:

How does our choice of C_i affect our ability to learn U?

1. Phase ret … en from a
   unitary 4-

Unitary t-design is a set of unitaries that has same t^{th} order moments as the Haar distribution

2. Phase ret … om a
   unitary 2-

Our Results:

1. Phase retrieval of all unitary matrices, when C_i chosen from a unitary 4-design
   - Use PhaseLift algorithm
   - Matrix analog of vector phase retrieval result using vector 4-designs [Kueng, Rauhut, and Terstiege, 2014]

2. Phase retrieval works pretty well when C_i chosen from a unitary 2-design.

Our Results:

1. Phase retrieval of all unitary matrices, when C_i chosen from a unitary 4-design
   - Use PhaseLift algorithm
   - Matrix analog of vector phase retrieval result using vector 4-designs [Kueng, Rauhut, and Terstiege, 2014]

2. Phase retrieval works pretty well when C_i chosen from a unitary 2-design.
   - Note: no-go result for PhaseLift using vector 2-designs [Gross et al, 2013]
   - PhaseLift is approximately correct, for most unitaries

More Details

- PhaseLift: Lifts vector to matrix, solve convex optimization problem on larger space
- Our case: lifts matrix to larger matrix, solve convex optimization problem on larger space

$$U \to vec(U)vec(U)^* \in \mathbb{C}^{d^2 \times d^2}$$

More Details

- PhaseLift: Lifts vector to matrix, solve convex optimization problem on larger space
- Our case: lifts matrix to larger matrix, solve convex optimization problem on larger space

$$U \rightarrow vec(U)vec(U)^* \in \mathbb{C}^{d^2 \times d^2}$$

- Follow strategy similar to [Keung et al. 2014] for state 4-designs.

More Details

- PhaseLift: Lifts vector to matrix, solve convex optimization problem on larger space
- Our case: lifts matrix to larger matrix, solve convex optimization problem on larger space

$$U \rightarrow vec(U)vec(U)^* \in \mathbb{C}^{d^2 \times d^2}$$

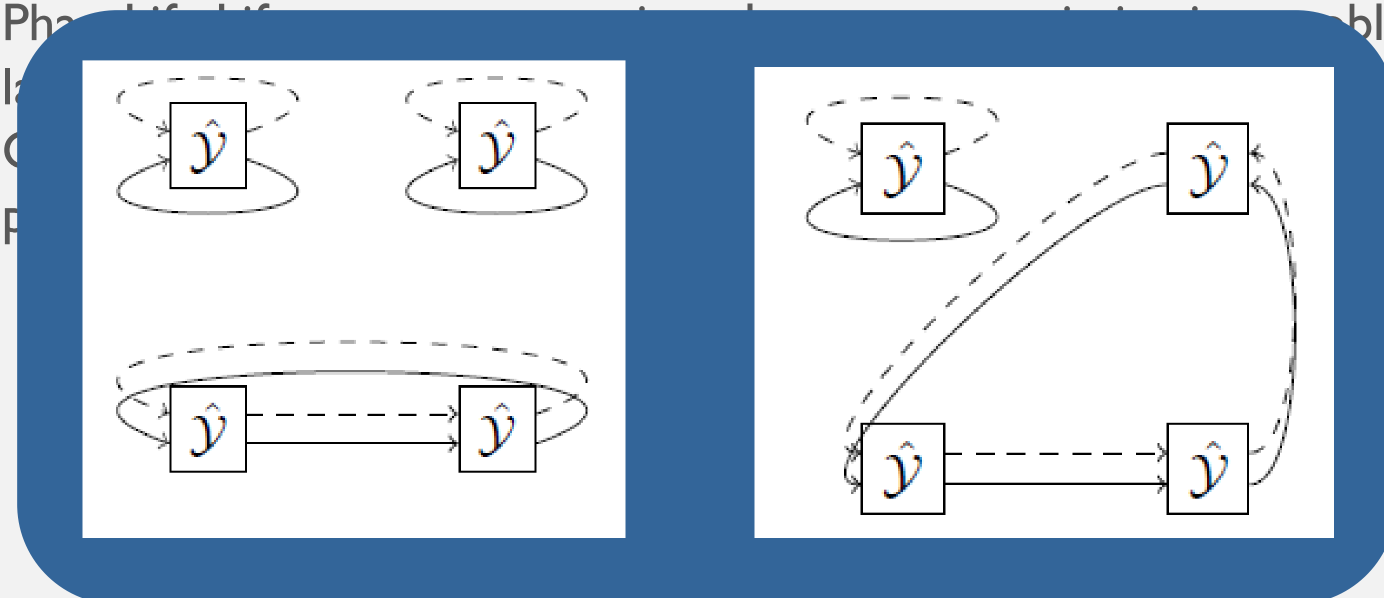
- Follow strategy similar to [Keung et al. 2014] for state 4-designs.
- Key component: bounding expectation of 4^{th} power of certain term
  - ✓ For unitary 4-design, can bound using properties of 4^{th} moment of Haar random unitaries (using Weingarten functions and commutative diagrams)

More Details

- Pha … blem on la …
- C …

  P …

- … .
- … erm … ment



commutative diagrams)

More Details

- PhaseLift: Lifts vector to matrix, solve convex optimization problem on larger space
- Our case: lifts matrix to larger matrix, solve convex optimization problem on larger space

$$U \to vec(U)vec(U)^* \in \mathbb{C}^{d^2 \times d^2}$$

- Follow strategy similar to [Keung et al. 2014] for state 4-designs.
- Key component: bounding expectation of 4^{th} power of certain term
  - ✓ For unitary 4-design, can bound using properties of 4^{th} moment of Haar random unitaries (using Weingarten functions and commutative diagrams)

More Details

- PhaseLift: Lifts vector to matrix, solve convex optimization problem on larger space
- Our case: lifts matrix to larger matrix, solve convex optimization problem on larger space

$$U \to vec(U)vec(U)^* \in \mathbb{C}^{d^2 \times d^2}$$

- Follow strategy similar to [Keung et al. 2014] for state 4-designs.
- Key component: bounding expectation of 4th power of certain term
  - ✓ For unitary 4-design, can bound using properties of 4^{th} moment of Haar random unitaries
  - ✓ For unitary 2-design, can bound using properties of 2^{nd} moment of Haar random unitaries, AND non-spikiness condition.

More Details

Non-spikiness condition:

Let $\tilde{G}$ be a finite set of unitary matrices in $\mathbb{C}^{d\times d}$. We say that a unitary matrix $U \in \mathbb{C}^{d\times d}$ is *non-spiky* with respect to $\tilde{G}$ (with parameter $\beta \geq 0$) if the following holds:

$$|\mathrm{tr}(C^\dagger U)|^2 \leq \beta, \quad \forall C \in \tilde{G}. \tag{I.13}$$

Fact: Almost all unitary matrices are non-spiky when $\beta \sim \log|\tilde{G}|$!

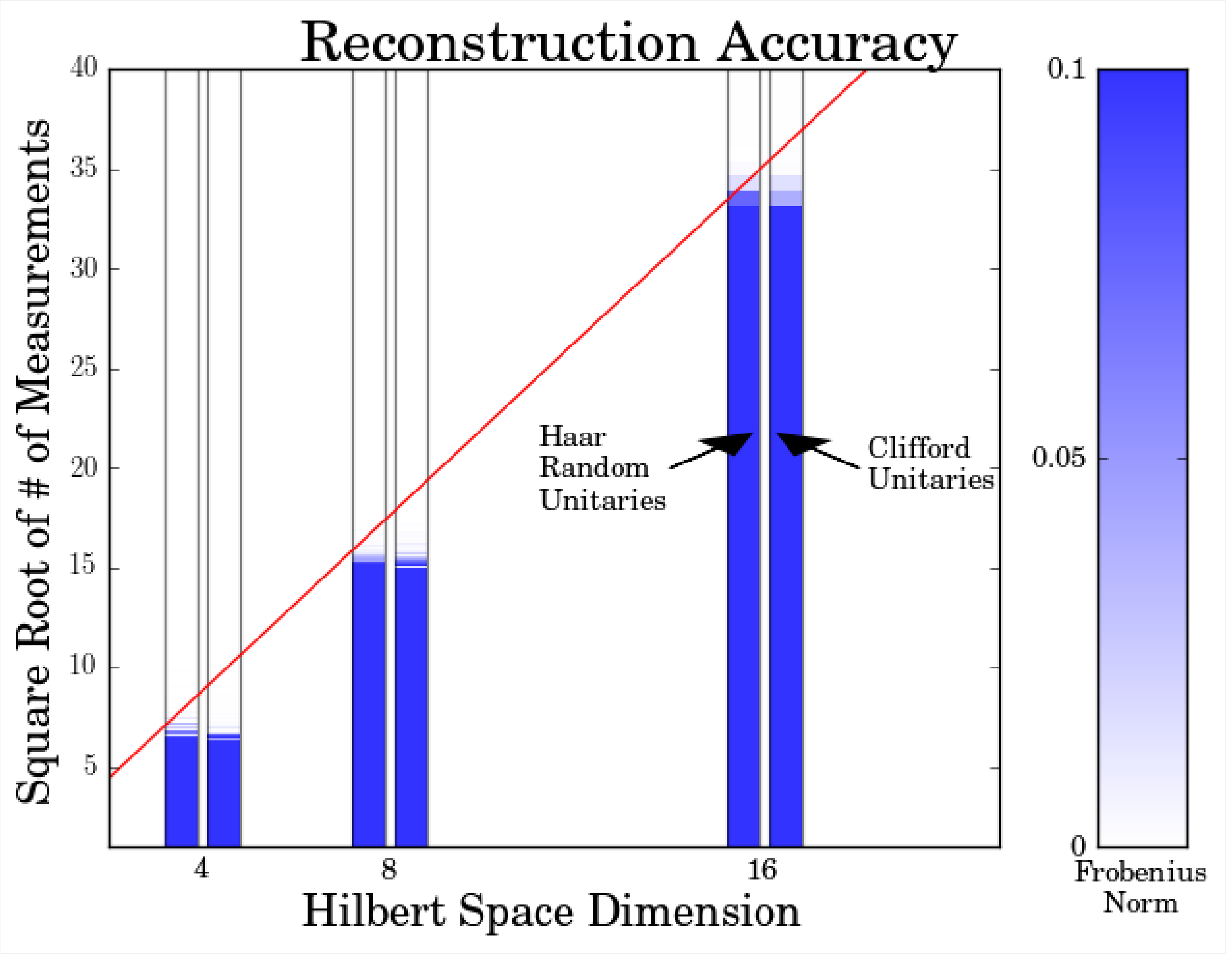
What about Cliffords?

1. Phase retrieval of all unitary matrices, when C_i chosen from a unitary 4-design

Cliffords form a unitary 3-design! [Zhu; Webb; Kueng and Gross, 2015]

2. Phase retrieval works pretty well when C_i chosen from a unitary 2-design.

What about Cliffords?



Summary and Conclusions

- Phase retrieval of unitary matrices
  - Motivation: quantum gate tomography
  - Used variant of PhaseLift algorithm
- Exact recovery using unitary 4-designs, approximate recovery using unitary 2-designs. See **arXiv/1510.08887**
- Outlook:
  - What about 3-designs? What about Cliffords in particular?
  - Different algorithm, e.g. Wirtinger Flow?

PhaseLift for Unitary Matrices

- Measurements $y = \mathcal{A}\big(\mathrm{vec}(U)\mathrm{vec}(U)^\dagger\big) + \varepsilon$
  - Where

$$\mathcal{A} : \mathbb{C}^{d^2\times d^2}_{\mathrm{Herm}} \to \mathbb{R}^m$$

$$\mathcal{A}(\Gamma) = \left[\mathrm{vec}(\sqrt{d}C_i)^\dagger \Gamma \mathrm{vec}(\sqrt{d}C_i)\right]_{i=1}^m$$

- Convex program

$$\begin{aligned} \arg \min_{\Gamma \in \mathbb{C}^{d^2\times d^2}_{\mathrm{Herm}}} \ & \mathrm{tr}(\Gamma) \text{ such that} \\ & \|\mathcal{A}(\Gamma) - y\|_2 \le \eta, \\ & \Gamma \succeq 0, \\ & \mathrm{tr}_1(\Gamma) = (I/d)\,\mathrm{tr}(\Gamma), \\ & \mathrm{tr}_2(\Gamma) = (I/d)\,\mathrm{tr}(\Gamma). \end{aligned}$$

PhaseLift Using Unitary 4-Designs

- “Exact” recovery guarantee:

Suppose that the number of measurements satisfies

$$m \geq \left(64(4!)^2 c_0\right)^2 \cdot d^2 \ln d. \qquad \text{(I.11)}$$

Then with probability at least $1 - \exp\left(-2m\,(4(4!))^{-4}\right)$ *(over the choice of the* C_i*), we have the following uniform recovery guarantee:*

For any unitary matrix $U \in \mathbb{C}^{d\times d}$*, it is the case that any solution* Γ_{opt} *to the convex program (I.7) with noisy measurements (I.6) must satisfy:*

$$\|\Gamma_{opt} - vec(U)vec(U)^\dagger\|_F \leq \frac{128(4!)^2\eta}{\sqrt{m}}\left(1 + \frac{2c_5}{c_0 - 2c_5}\right). \qquad \text{(I.12)}$$

PhaseLift Using Unitary 2-Designs

- We will seek to recover all unitary matrices U that are "non-spiky" with respect to the measurement matrices C_i

Let $\tilde{G}$ be a finite set of unitary matrices in $\mathbb{C}^{d\times d}$. We say that a unitary matrix $U \in \mathbb{C}^{d\times d}$ is *non-spiky* with respect to $\tilde{G}$ (with parameter $\beta \geq 0$) if the following holds:

$$|\mathrm{tr}(C^{\dagger}U)|^2 \leq \beta, \quad \forall C \in \tilde{G}. \tag{I.13}$$

- Fact: Almost all unitary matrices are non-spiky, with parameter $\beta \sim \log|\tilde{G}|$

PhaseLift Using Unitary 2-Designs

- For all β-non-spiky unitary matrices U, we achieve "approximate" recovery, up to error δ
  - Let v = β/δ

Suppose that the number of measurements satisfies

$$m \geq \left(8c_0\nu^2\right)^2 \cdot d^2 \ln d. \qquad \text{(I.16)}$$

Then with probability at least $1 - \exp(-\frac{1}{128}m\nu^{-4})$ *(over the choice of the* C_i*), we have the following uniform recovery guarantee:*

For any unitary matrix $U \in \mathbb{C}^{d\times d}$ *that is non-spiky with respect to* $\tilde{G}$ *(with parameter* β*, in the sense of (I.13)),* [redacted]

$$\|\Gamma_{opt} - vec(U)vec(U)^\dagger\|_F \leq \max\left\{\delta\|vec(U)vec(U)^\dagger\|_F,\ \frac{16\eta\nu^2}{\sqrt{m}}\left(1 + \frac{2c_5}{c_0 - 2c_5}\right)\right\}. \qquad \text{(I.17)}$$