

Title: Quantum machine learning and the prospect of near-term applications on noisy devices

Speakers: Kristan Temme

Collection: Machine Learning for Quantum Design

Date: July 10, 2019 - 2:00 PM

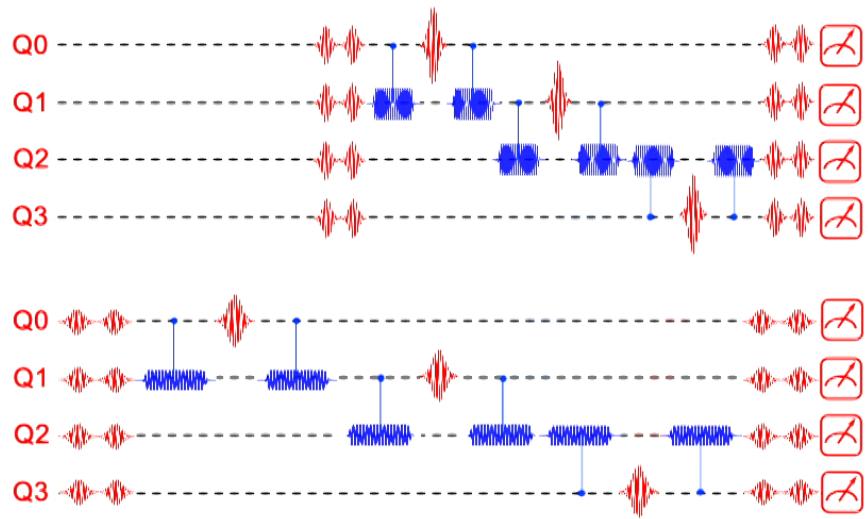
URL: <http://pirsa.org/19070010>

Abstract: Prospective near-term applications of early quantum devices rely on accurate estimates of expectation values to become relevant. Decoherence and gate errors lead to wrong estimates. This problem was, at least in theory, remedied with the advent of quantum error correction. However, the overhead that is needed to implement a fully fault-tolerant gate set with current codes and current devices seems prohibitively large. In turn, steady progress is made in improving the quality of the quantum hardware, which leads to the belief that in the foreseeable future machines could be built that cannot be emulated by a conventional computer. In light of recent progress mitigating the effect of decoherence on expectation values, it becomes interesting to ask what these noisy devices can be used for. In this talk we will present our advances in finding quantum machine learning applications for noisy quantum computers.

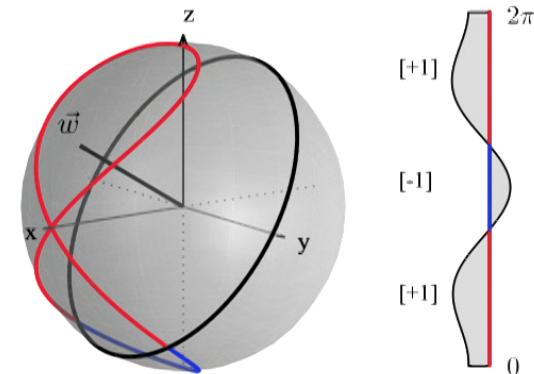
Quantum machine learning and the prospect of near term applications on noisy devices.

Kristan Temme
IBM Research

Error mitigation



Machine learning



$$\Phi : \vec{x} \in \Omega \rightarrow |\Phi(\vec{x})\rangle\langle\Phi(\vec{x})|$$

Noise and quantum error correction

PHYSICAL REVIEW A

VOLUME 51, NUMBER 2

FEBRUARY 1995

Maintaining coherence in quantum computers

W. G. Unruh*

Canadian Institute for Advanced Research, Cosmology Program, Department of Physics,
University of British Columbia, Vancouver, Canada V6T 1Z1
(Received 10 June 1994)

THIRD SERIES, VOLUME 52, NUMBER 4

OCTOBER 1995

RAPID COMMUNICATIONS

The Rapid Communications section is intended for the accelerated publication of important new results. Since manuscripts submitted to this section are given priority treatment both in the editorial office and in production, authors should explain in their submittal letter why the work justifies this special handling. A Rapid Communication should be no longer than 4 printed pages and must be accompanied by an abstract. Page proofs are sent to authors.

Scheme for reducing decoherence in quantum computer memory

Peter W. Shor*

AT&T Bell Laboratories, Room 2D-149, 600 Mountain Avenue, Murray Hill, New Jersey 07974
(Received 17 May 1995)

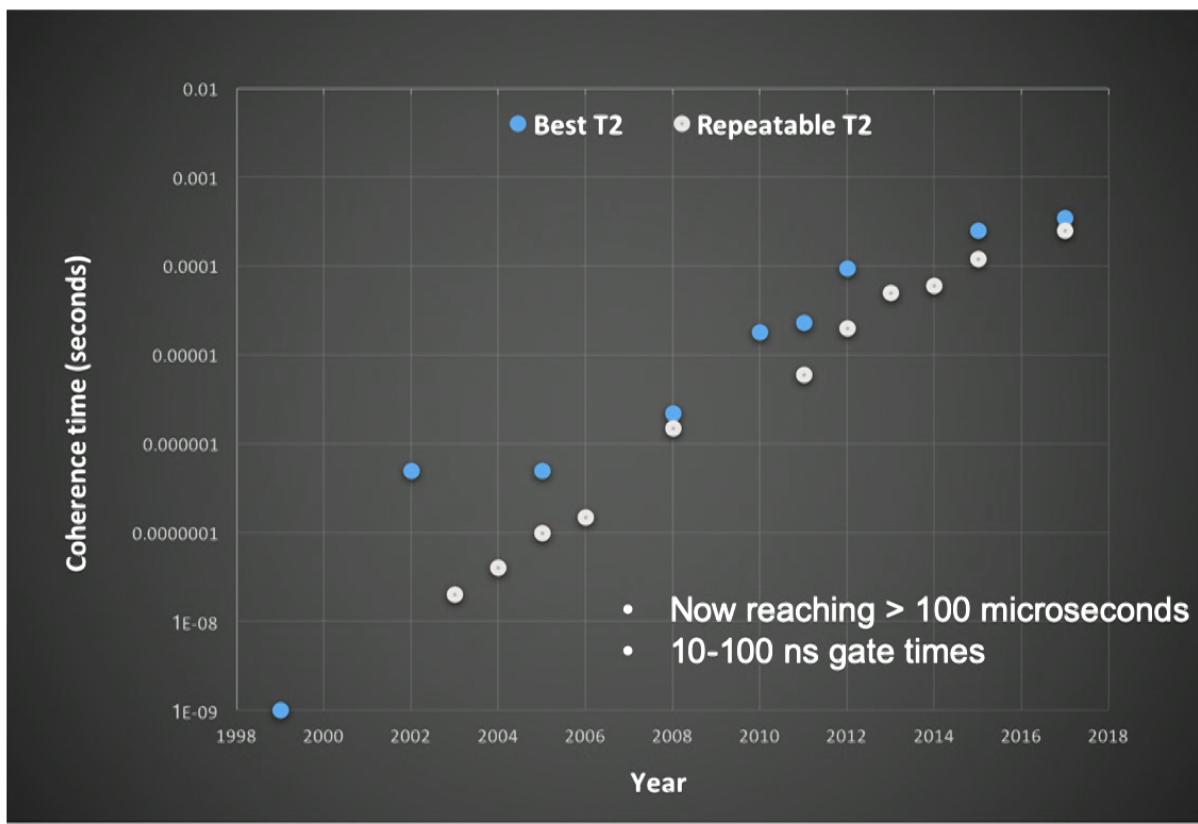
Recently, it was realized that use of the properties of quantum mechanics might speed up certain computations dramatically. Interest has since been growing in the area of quantum computation. One of the main difficulties of quantum computation is that decoherence destroys the information in a superposition of states contained in a quantum computer, thus making long computations impossible. It is shown how to reduce the effects of decoherence for information stored in quantum memory, assuming that the decoherence process acts independently on each of the bits stored in memory. This involves the use of a quantum analog of error-correcting codes.

Table 4. The resources needed to factor a 1024-bit number with Shor's algorithm. Results shown for the surface and Bacon-Shor codes on three technologies.

Technology	Neutral Atoms	Supercond. Qubits	Ion Traps
Gate error	1×10^{-3}	1×10^{-5}	1×10^{-9}
Avg. gate time	19,000 ns	25 ns	32,000 ns
Execution time	2.62 years	10.81 hours	2.22 years
No. qubits	5.29×10^8	4.57×10^7	1.44×10^8
No. gates	1.02×10^{21}	2.55×10^{19}	5.10×10^{19}
Dominant gate	<i>CNOT</i>	<i>CNOT</i>	<i>CNOT</i>
Code distance	17	5	3
Logical gate error	4.99×10^{-11}	2.95×10^{-11}	4.92×10^{-15}
Logical gate time	1.29×10^5 ns	2.10×10^2 ns	5.96×10^5 ns
No. qubits per logical	3.73×10^4	3.23×10^3	1.16×10^3
No. gates per logical	1.11×10^5	9.60×10^3	3.46×10^3
Execution time	N/A	5.10 years	57.98 days
No. qubits	N/A	2.65×10^{12}	4.60×10^5
No. gates	N/A	1.16×10^{32}	4.07×10^{18}
Dominant gate	N/A	<i>SWAP</i>	<i>CNOT</i>
Code concatenations	N/A	5	1
Logical gate error	N/A	3.42×10^{-15}	5.09×10^{-14}
Logical gate time	N/A	1.42×10^7 ns	7.27×10^5 ns
No. qubits per logical	N/A	2.82×10^8	49
No. gates per logical	N/A	1.18×10^{11}	79

M. Suchara et al. arXiv:1312.2316 (2013)

Coherence times of superconducting qubits



- Developments to extend coherence times
 - Materials e.g. [2]
 - Design and geometries e.g. [3]
 - 3D transmon [4]
 - IR Shielding [5,6],
 - Cold normal metal cavities and cold qubits [7]
 - High Q cavities [8]
 - Titanium Nitride (collaboration with David Pappas @ NIST Boulder) [9] ...
- Remarkable progress over the past decade

- [2] J. Martinis *et al.*, PRL **95**, 210503 (2005)
- [3] K. Geerlings *et al.*, APL **192601** (2012)
- [4] H. Paik *et al.*, PRL **107**, 240501 (2011)
- [5] R. Barends *et al.*, APL **99**, 113507 (2011)
- [6] A. Corcoles *et al.*, APL **99**, 181906 (2011)
- [7] C. Rigetti *et al.*, PRB **86**, 100506 (2012)
- [8] M. Reagor *et al.*, arXiv:1302.4408 (2013)
- [9] J. Chang *et al.* APL **103**, 012602 (2013)

Noisy Intermediate Scale Quantum Computing (NISQ)

- **Near – term**

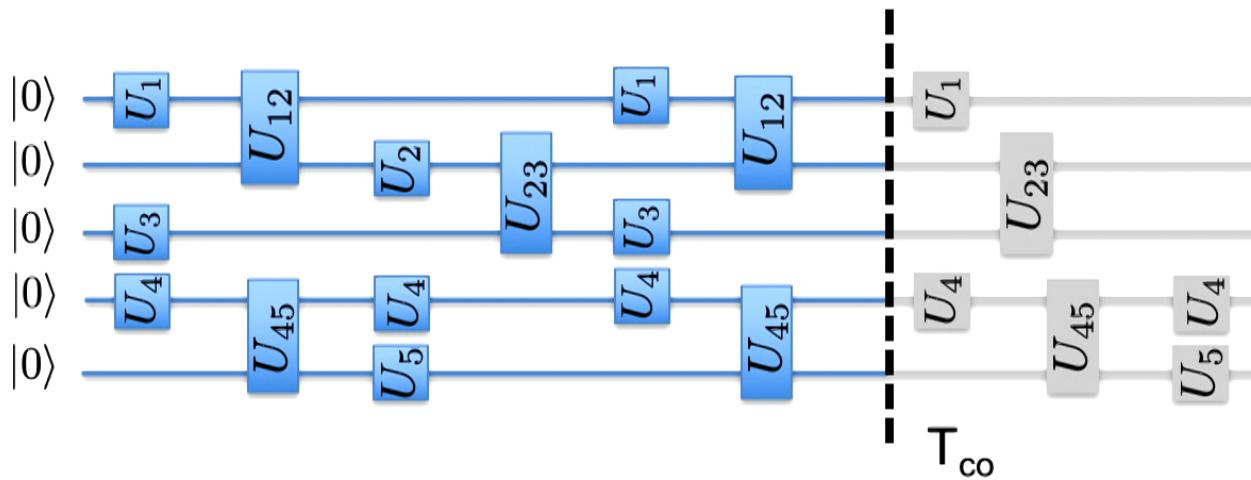
- Noisy devices without full error correction
- Maybe only $O(10^2 + X)$ qubits available
- Decoherence, gate errors and measurement errors limit the usefulness
- Algorithms will need to be designed with noisy hardware in mind

Noisy Intermediate Scale Quantum Computing (NISQ)

- **Near – term**

- Noisy devices without full error correction
- Maybe only $O(10^2 + X)$ qubits available
- Decoherence, gate errors and measurement errors limit the usefulness
- Algorithms will need to be designed with noisy hardware in mind

- **Possible approach: short-depth circuits**

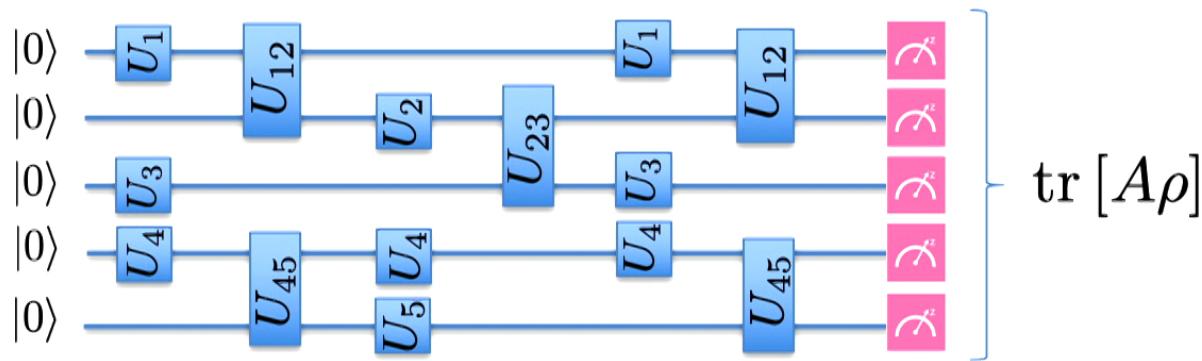


Noisy Intermediate Scale Quantum Computing (NISQ)

- Near – term

- Noisy devices without full error correction
- Maybe only $O(10^2 + X)$ qubits available
- Decoherence, gate errors and measurement errors limit the usefulness
- Algorithms will need to be designed with noisy hardware in mind

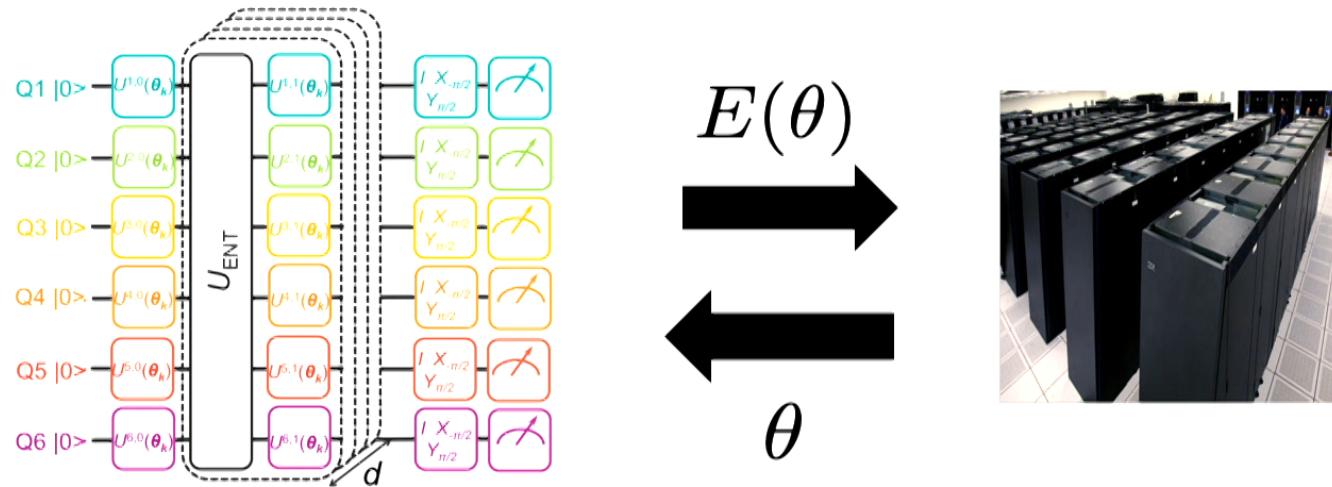
- Possible approach: short-depth circuits



Variational quantum eigensolver

A simple hybrid quantum-classical algorithm can be used to solve problems where the goal is to minimize the energy of a system

$$H = \sum_{\alpha} h_{\alpha} \sigma(\alpha)$$



Map problem to
Paulis

Prepare a trial state $|\psi(\theta)\rangle$
and compute its energy $E(\theta)$

$$|\psi(\theta)\rangle = [U(\theta_k)U_{ent}]^d U(\theta_0)|0\rangle$$

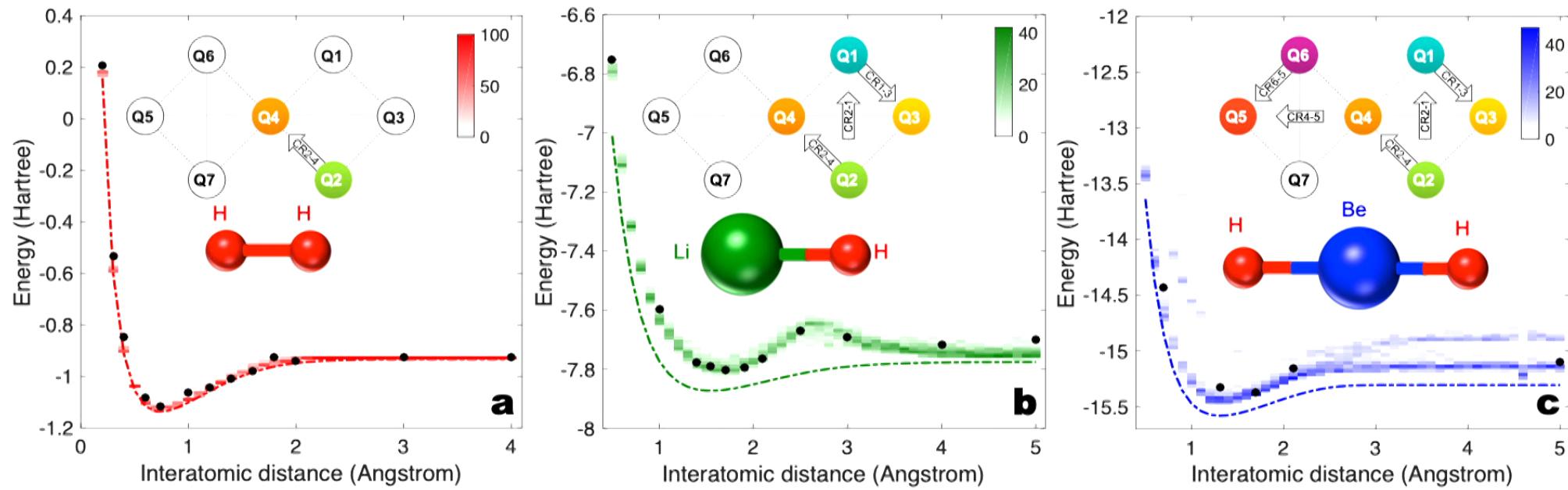
$$E(\theta) = \sum_{\alpha} h_{\alpha} \langle \psi(\theta) | \sigma(\alpha) | \psi(\theta) \rangle$$



Use classical optimizer

- Peruzzo *et al.*, Nat. Commun. 5, 4213 (2014)
- Farhi *et al.*, arXiv:1411.4028 (2014)
-

Application to quantum chemistry



H_2 : 2 qubits
5 pauli terms, 2 sets

- Decoherence
- Sampling error
- Limited iterations

LiH : 4 qubits
100 pauli terms, 25 sets

- Accuracy of the classical optimizer
- Insufficient depth

BeH_2 : 6 qubits
165 pauli terms, 44 sets

Nature **549**, 242-246 (2017)

How could we deal with decoherence?

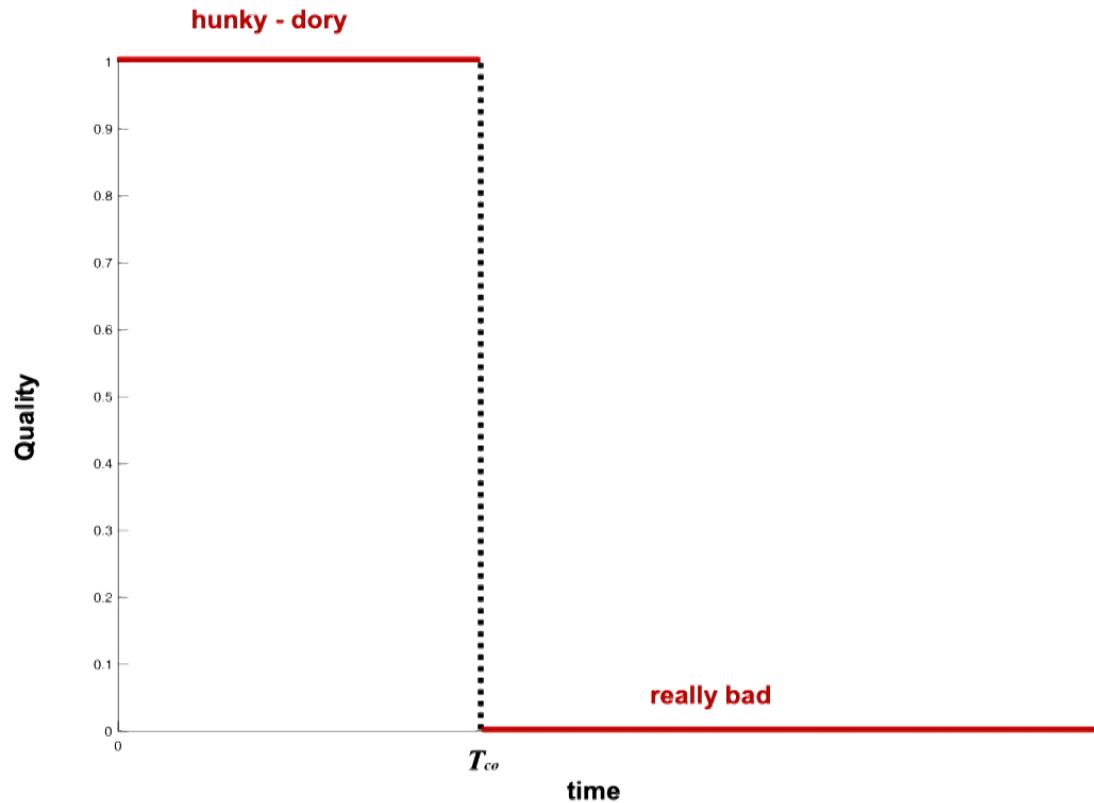
Experiment:

Kandala, Temme, Corcoles, Mezzacapo, Chow, Gambetta, **Nature vol. 567, 491 (2019)**

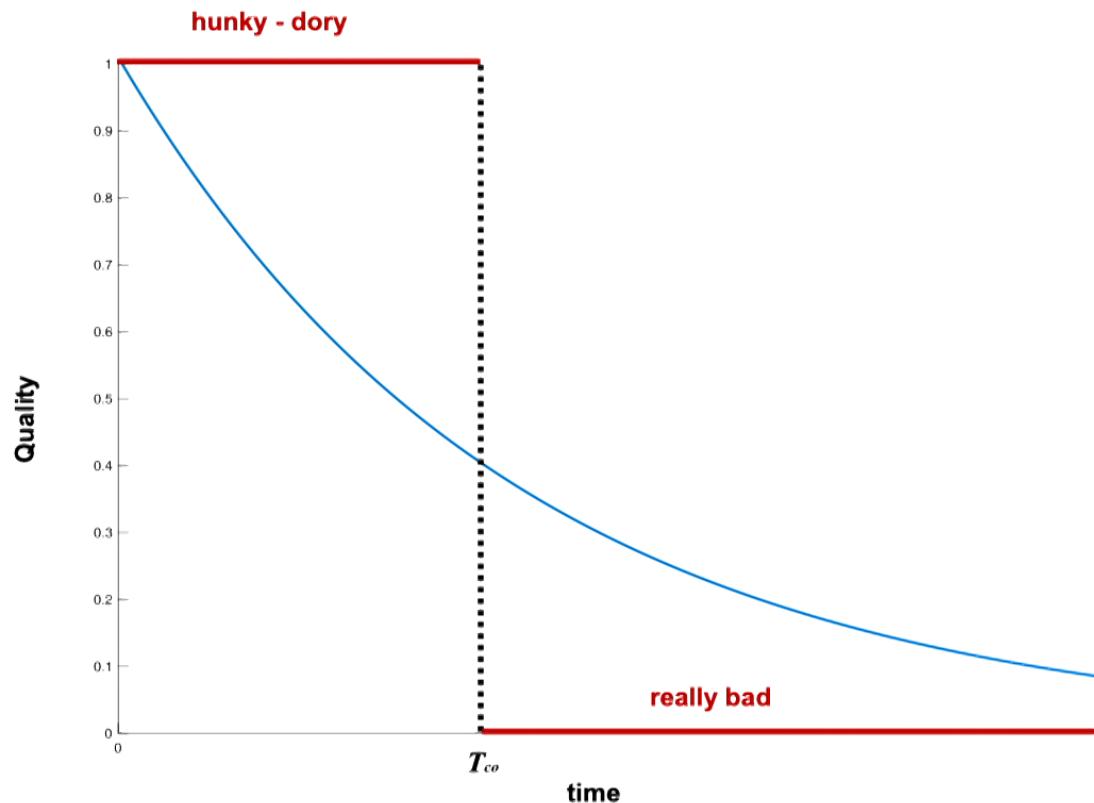
Proposal:

Temme, Brayvi, Gambetta, **Phys. Rev. Lett. 119, 180509 (2017)**

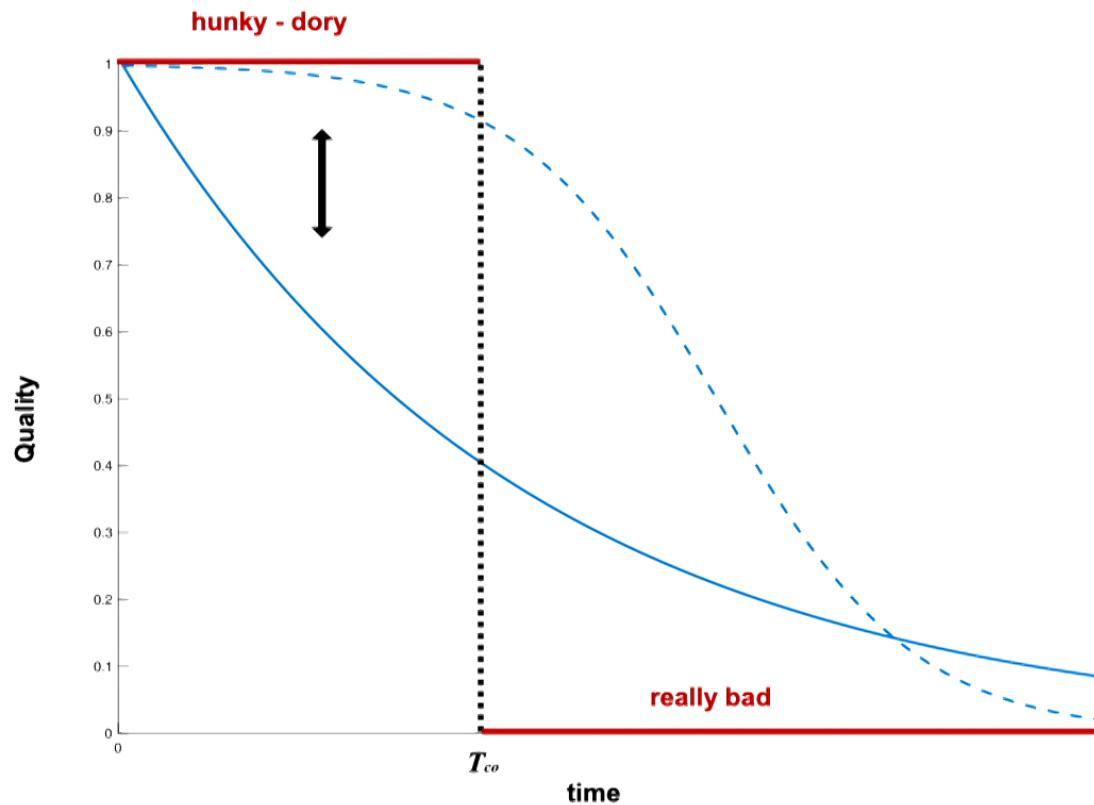
A cartoon



A cartoon



A cartoon



Two methods

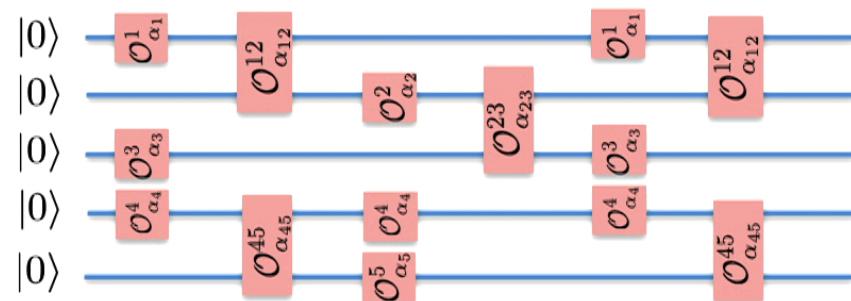
- **Method 1:**
Zero noise extrapolation

$$\text{tr} [A\rho_\lambda] = E(\lambda)$$



- **Method 2:**
Probabilistic error cancellation

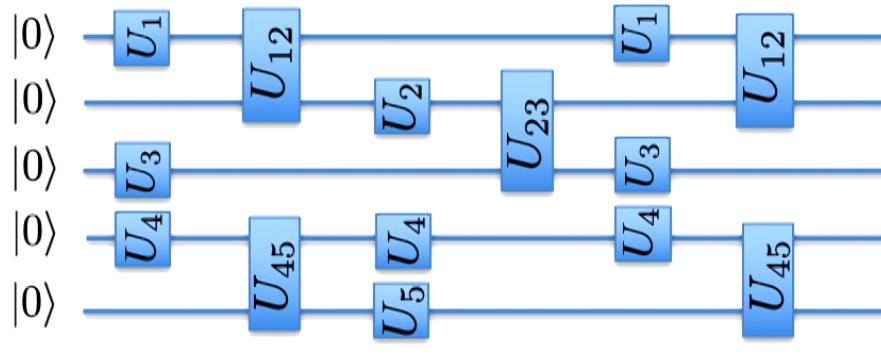
$$\gamma \mathcal{N}^{-1}(\rho)$$



Method 1

Zero noise extrapolation

The model



Dynamics modeled by: $\frac{\partial}{\partial t} \rho(t) = -i[K(t), \rho(t)] + \lambda \mathcal{L}(\rho(t))$ where $\mathcal{L}(\rho) = \sum_{\alpha} L_{\alpha} \rho L_{\alpha}^{\dagger} - \frac{1}{2} \{L_{\alpha}^{\dagger} L_{\alpha}, \rho\}$

$$\mathcal{L}(\rho) = -i[V, \rho]$$

The circuit can be encoded in to a time dependent Hamiltonian evolution with Hamiltonian

$$K(t) = \sum_{\alpha} J_{\alpha}(t) P_{\alpha}.$$

Hence, the circuit is given by

$$U(T) = \mathcal{T} \exp \left(-i \int_0^T K(t) dt \right)$$

Richardson extrapolation

- **Expectation value of observable A:**

$$E_K(\lambda) = \text{tr}(A\rho_\lambda(T))$$

$$E_K(\lambda) = E^* + \sum_{k=1}^n a_k \lambda^k + R_{n+1}(\lambda, \mathcal{L}, T)$$

Richardson extrapolation

- **Expectation value of observable A:**

$$E_K(\lambda) = \text{tr}(A\rho_\lambda(T))$$

$$E_K(\lambda) = E^* + \sum_{k=1}^n a_k \lambda^k + R_{n+1}(\lambda, \mathcal{L}, T)$$

Richardson extrapolation

- **Expectation value of observable A:**

$$E_K(\lambda) = \text{tr}(A\rho_\lambda(T))$$

$$E_K(\lambda) = E^* + \sum_{k=1}^n a_k \lambda^k + R_{n+1}(\lambda, \mathcal{L}, T)$$

Assume we have access to:

$$E_K(c_0\lambda), E_K(c_1\lambda), \dots E_K(c_n\lambda)$$

Richardson extrapolation

- **Expectation value of observable A:**

$$E_K(\lambda) = \text{tr}(A\rho_\lambda(T))$$

$$E_K(\lambda) = E^* + \sum_{k=1}^n a_k \lambda^k + R_{n+1}(\lambda, \mathcal{L}, T)$$

Assume we have access to: $E_K(c_0\lambda), E_K(c_1\lambda), \dots E_K(c_n\lambda)$

- **Richardson's deferred approach to the Limit**

Linear system of equations: $\sum_{l=0}^n \gamma_j = 1 \quad \text{and} \quad \sum_{j=0}^n \gamma_j c_j^k = 0 \quad \text{for } k = 1 \dots n.$

The estimator $\hat{E}_K^n(\lambda) = \sum_{j=0}^n \gamma_j \hat{E}_K(c_j \lambda)$ is a better approximation with $E^* + \mathcal{O}(\lambda^{n+1})$

How does one measure $E_K(c\lambda)$?

- **Noisy dynamics**

$$\frac{\partial}{\partial t}\rho(t) = -i[K(t), \rho(t)] + \lambda\mathcal{L}(\rho(t))$$

Evolve for time T with Hamiltonian:

$$K(t) = \sum_{\alpha} J_{\alpha}(t)P_{\alpha}$$

Measure $E_K(\lambda)$



How does one measure $E_K(c\lambda)$?

- Rescale the desired dynamics in

$$\frac{\partial}{\partial t}\rho(t) = -i[K_c(t), \rho(t)] + \lambda\mathcal{L}(\rho(t))$$

Evolve for time T with Hamiltonian:

$$K(t) = \sum_{\alpha} J_{\alpha}(t) P_{\alpha}$$

Measure $E_K(\lambda)$

Evolve for time cT with rescaled Hamiltonian:

$$K_c(t) = \sum_{\alpha} c^{-1} J_{\alpha}(c^{-1}t) P_{\alpha}$$

Measure $E_K(c\lambda)$



How does one measure $E_K(c\lambda)$?

- Rescale the desired dynamics in

$$\frac{\partial}{\partial t'} \rho(t') = -i[K(t'), \rho(t')] + c\lambda\mathcal{L}(\rho(t'))$$

Evolve for time T with Hamiltonian:

$$K(t) = \sum_{\alpha} J_{\alpha}(t) P_{\alpha}$$

Measure $E_K(\lambda)$

Evolve for time cT with rescaled Hamiltonian:

$$K_c(t) = \sum_{\alpha} c^{-1} J_{\alpha}(c^{-1}t) P_{\alpha}$$

Measure $E_K(c\lambda)$

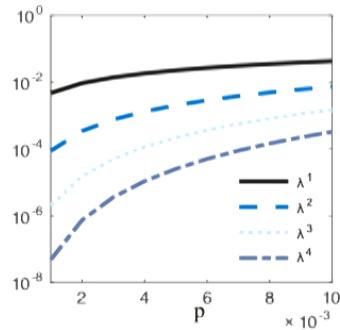


Zero noise extrapolation

If the protocol is performed for $n + 1$ steps, the error between the exact expectation value E^* and the estimator $\hat{E}_K^n(\lambda)$ can be bounded by

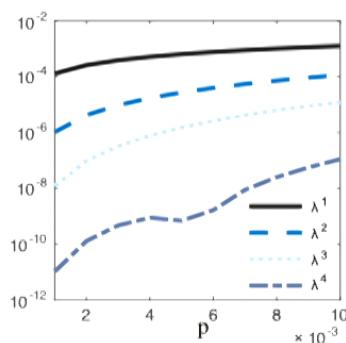
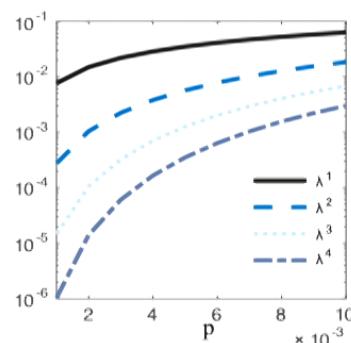
$$|E^* - \hat{E}_K^n(\lambda)| \leq \Gamma_n \left(\delta^* + \|A\| \frac{l_{n+1}(\lambda T)^{n+1}}{(n+1)!} \right).$$

Here $\Gamma_n = \sum_{j=0}^n |\gamma_j| c_j^{n+1}$ and $\delta^* = \max_j |\delta_j|$ is the largest experimental error.



$$\mathcal{L}_i = -\lambda(2^{-1}\text{tr}_i(\rho) - \rho)$$

$$\mathcal{L}_i = \lambda_1 (\sigma_i^- \rho \sigma_i^+ - \tfrac{1}{2} \{\sigma_i^+ \sigma_i^-, \rho\}) + \lambda_2 (Z_i \rho Z_i - \rho)$$



The bound depends on the noise model through

$$\|\mathcal{L}_{I,t_1} \circ \mathcal{L}_{I,t_2} \circ \dots \circ \mathcal{L}_{I,t_n}(\rho_0)\|_1 \leq l_n$$

where at most $l_n = \mathcal{O}(N^n)$

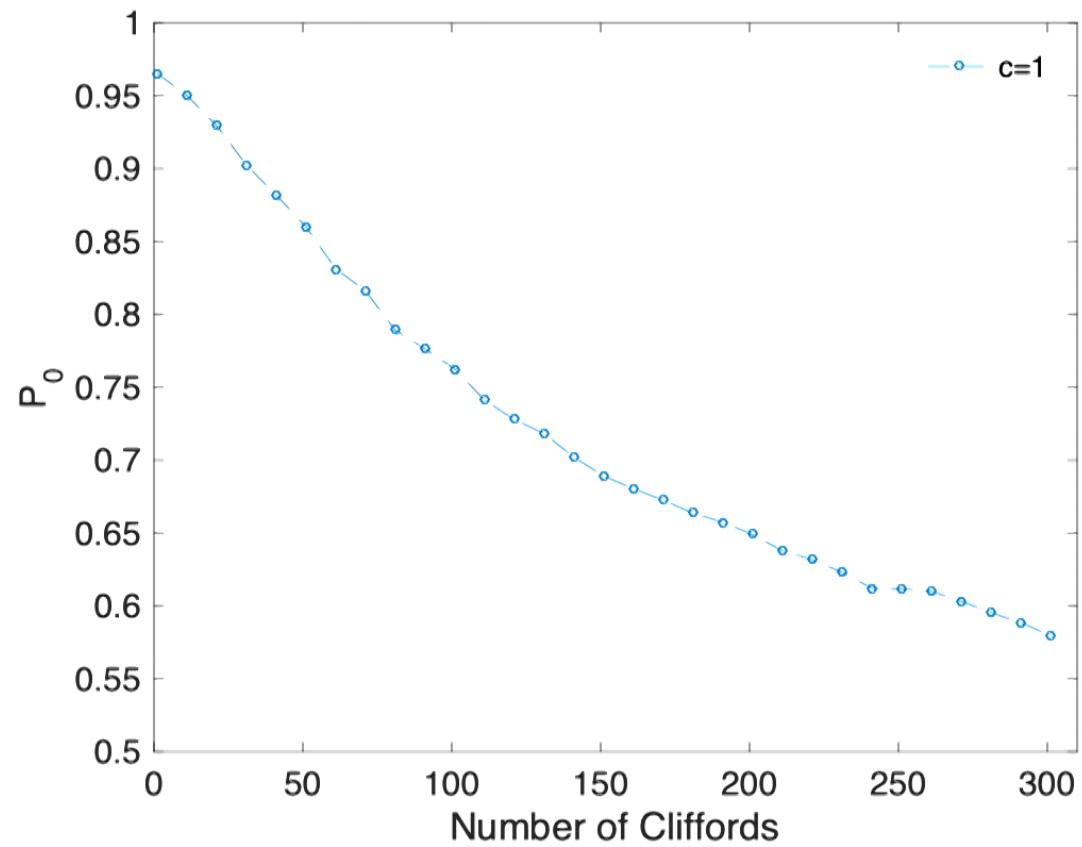
The solution to linear eqn. is given by:

$$\gamma_j = \prod_{m \neq j} c_m (c_j - c_m)^{-1}$$

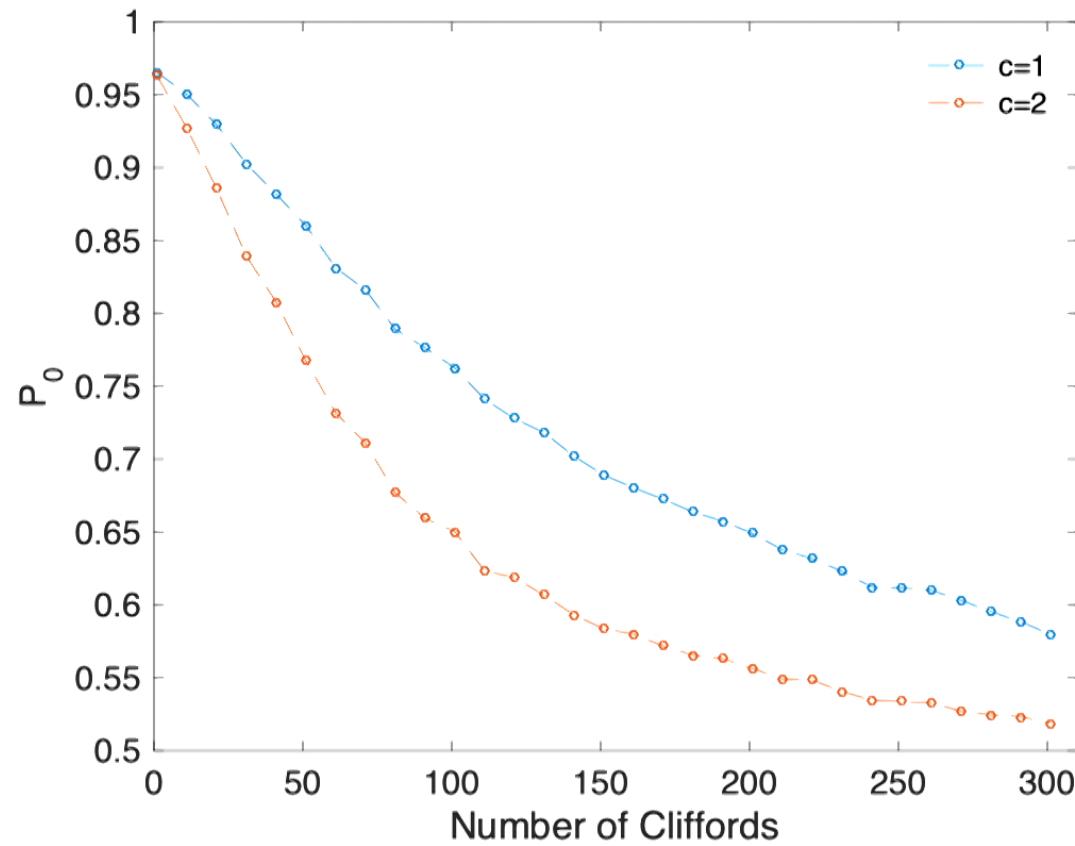
$$V_i = 1/2 X_i \otimes X_{b_i} + 1/2 Z_{b_i}$$

$$\rho_B = (2 \cosh(\beta/2))^{-N} \exp(-\beta \sum_{b_i} \sigma_{b_i}^z)$$

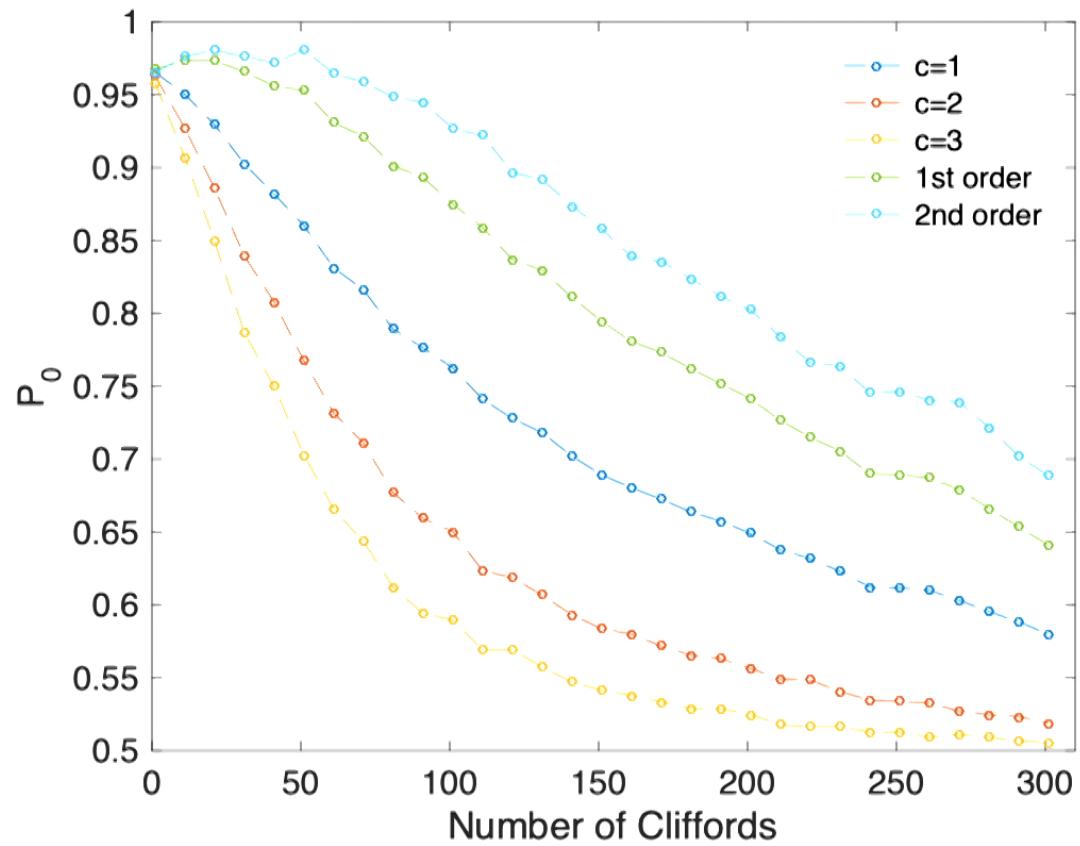
Error Mitigation: 1Q Clifford sequence



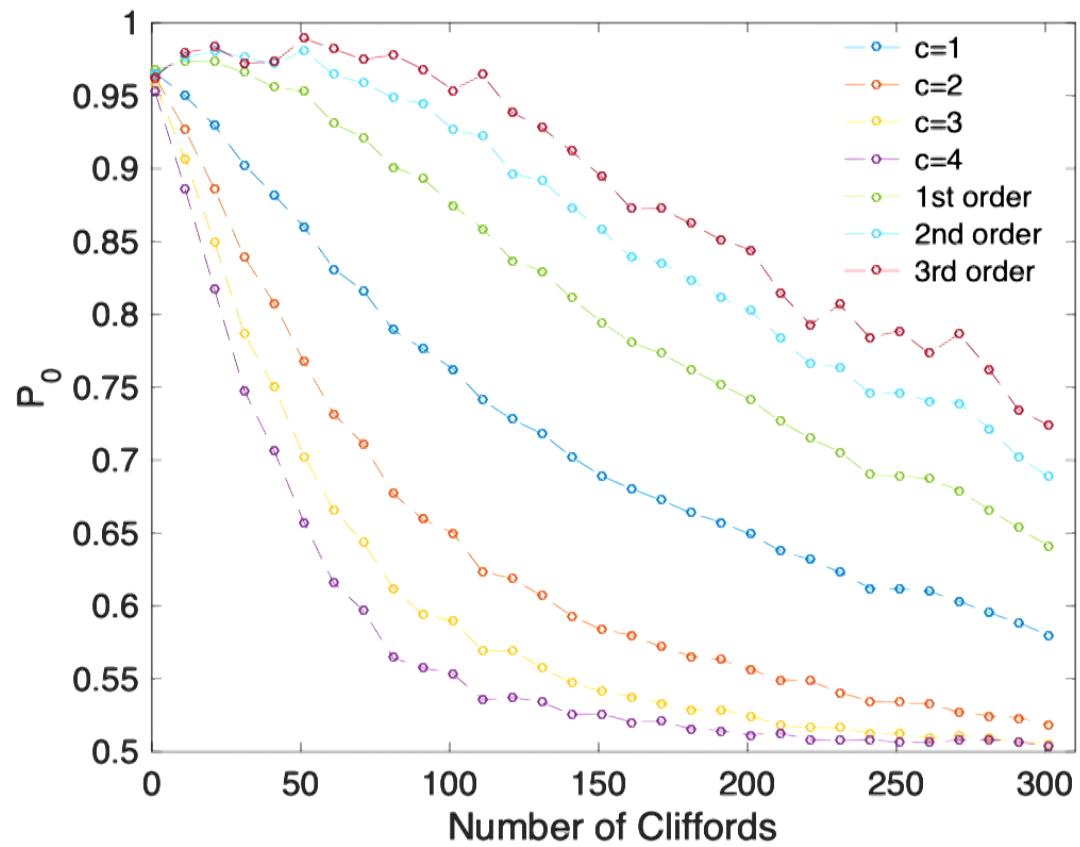
Error Mitigation: 1Q Clifford sequence



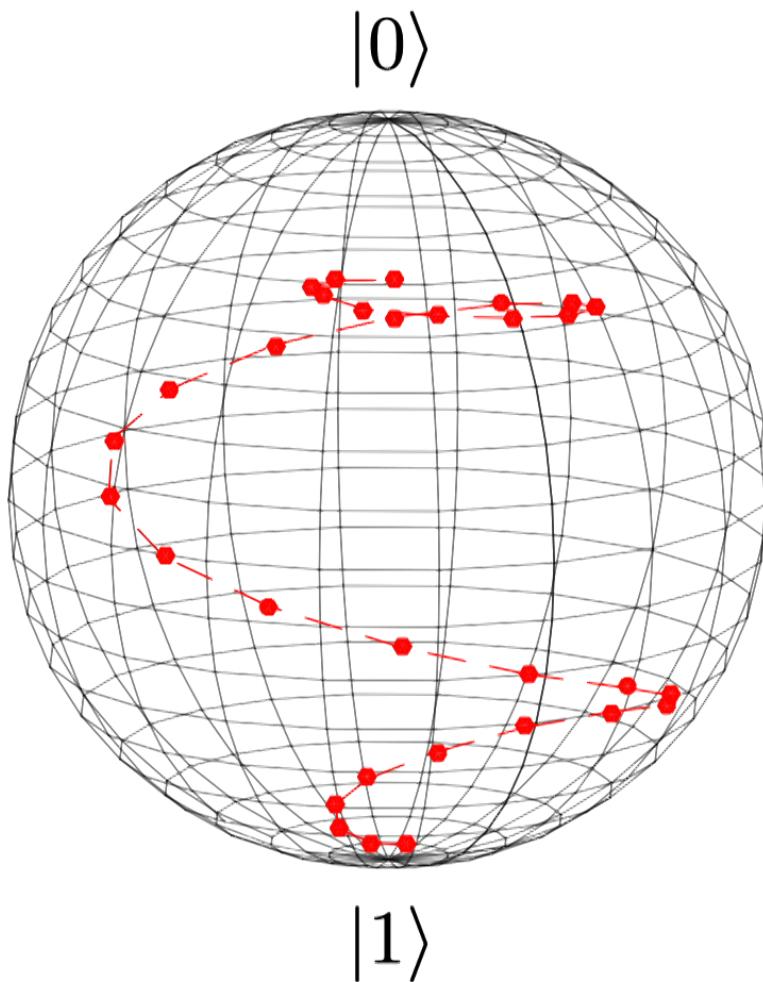
Error Mitigation: 1Q Clifford sequence



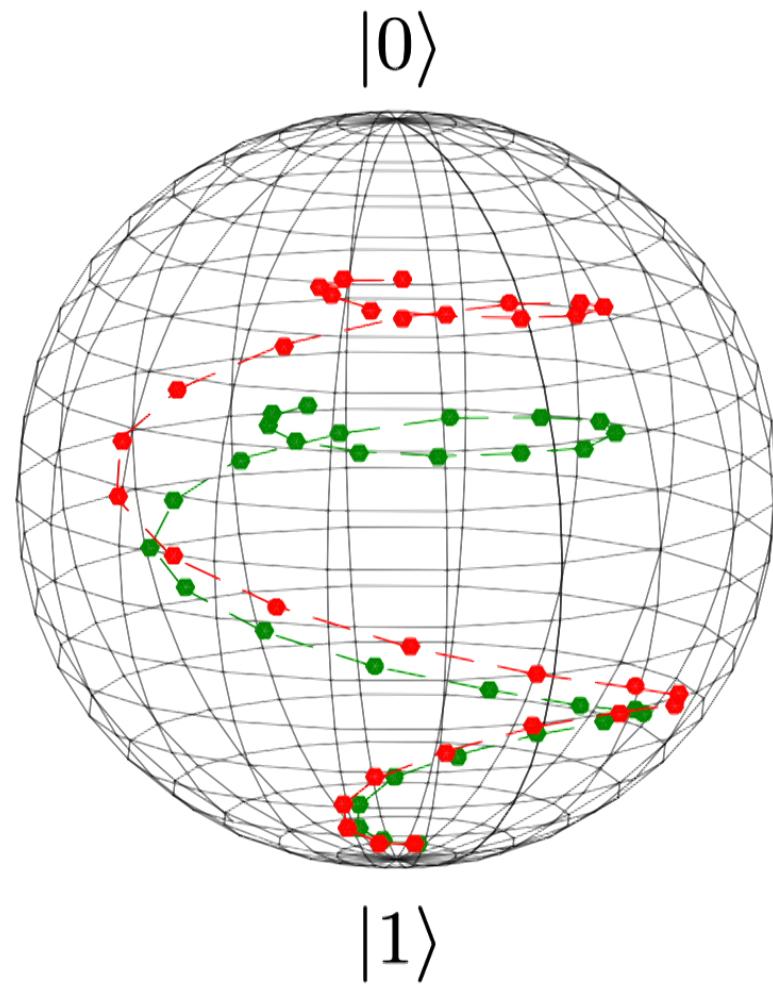
Error Mitigation: 1Q Clifford sequence



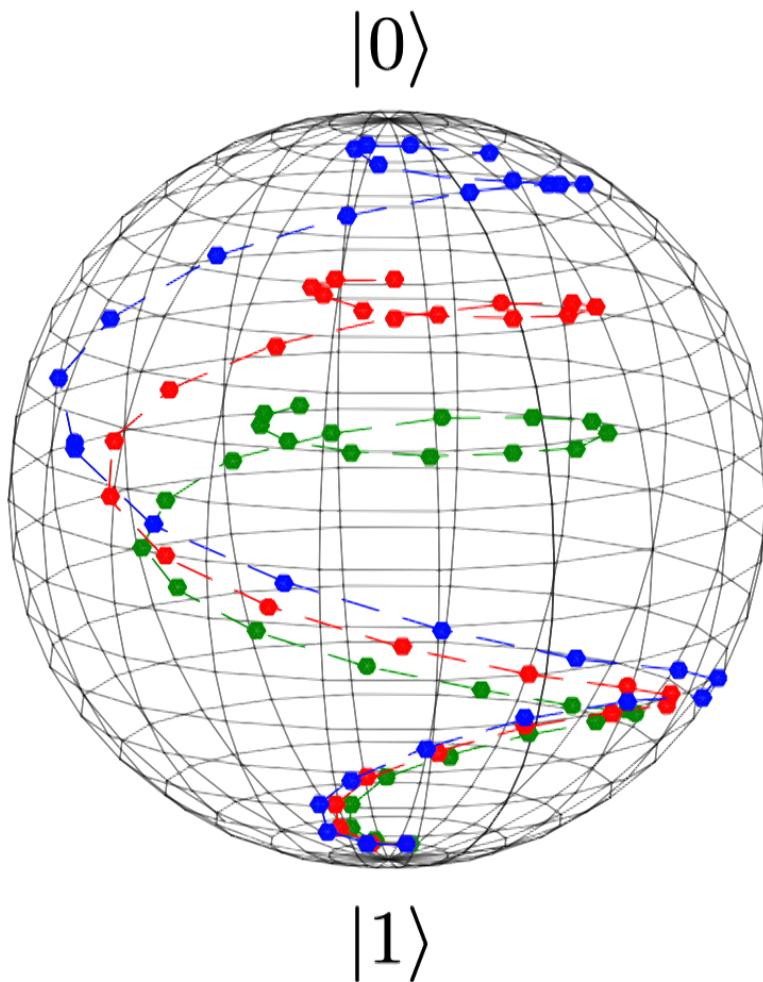
Qubit trajectories



Qubit trajectories

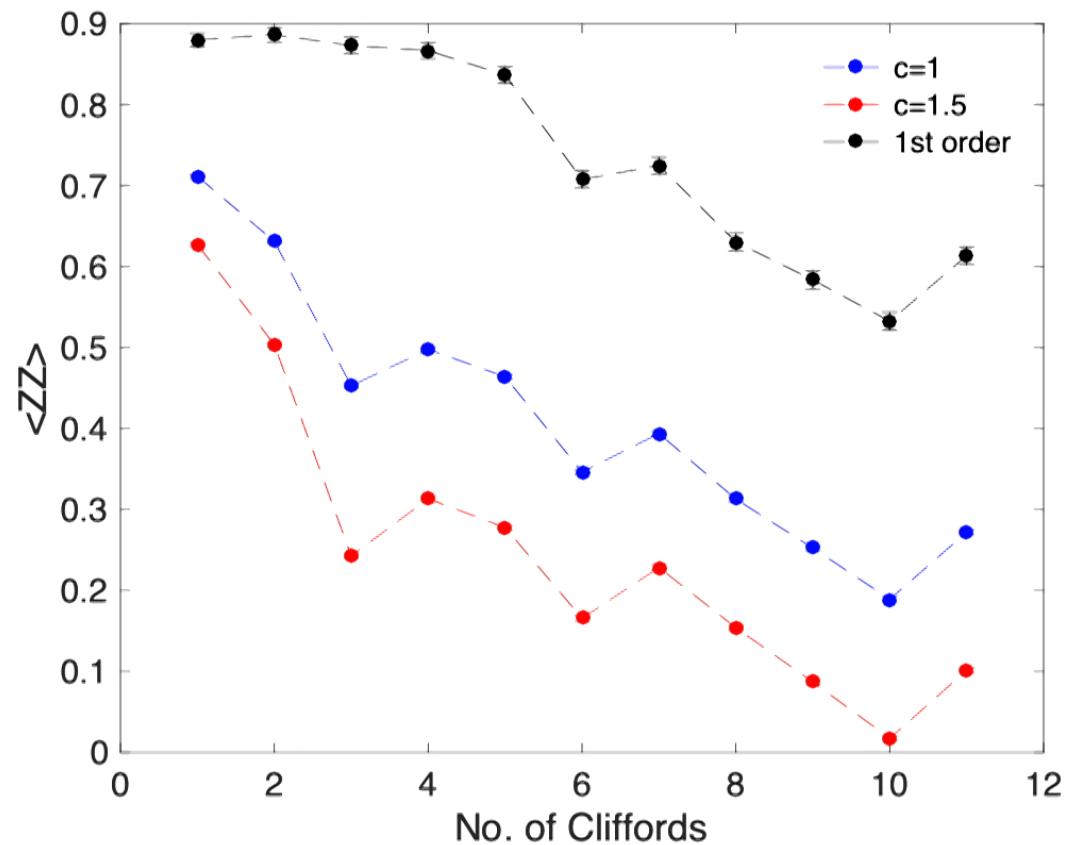


Qubit trajectories

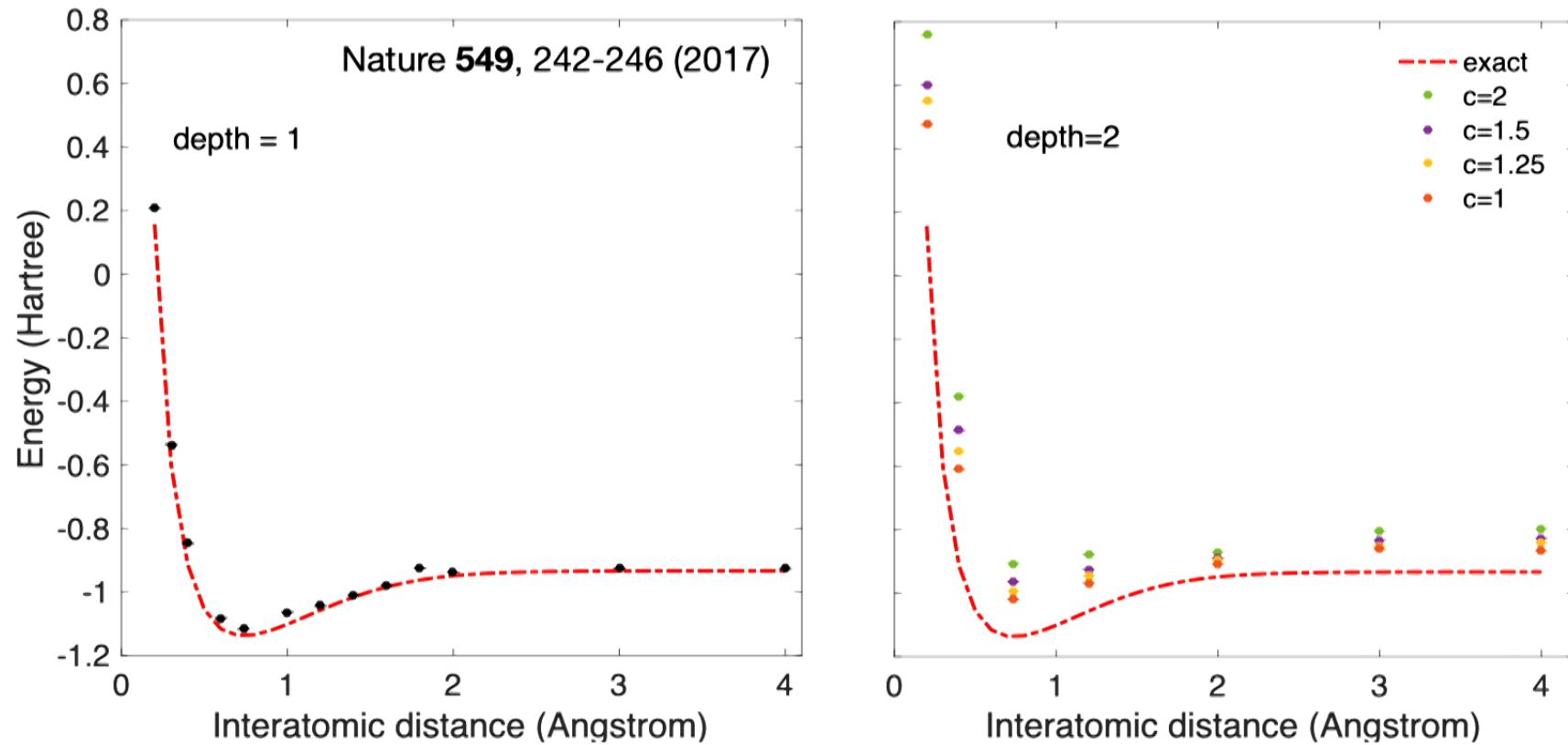


Error Mitigation: 2Q

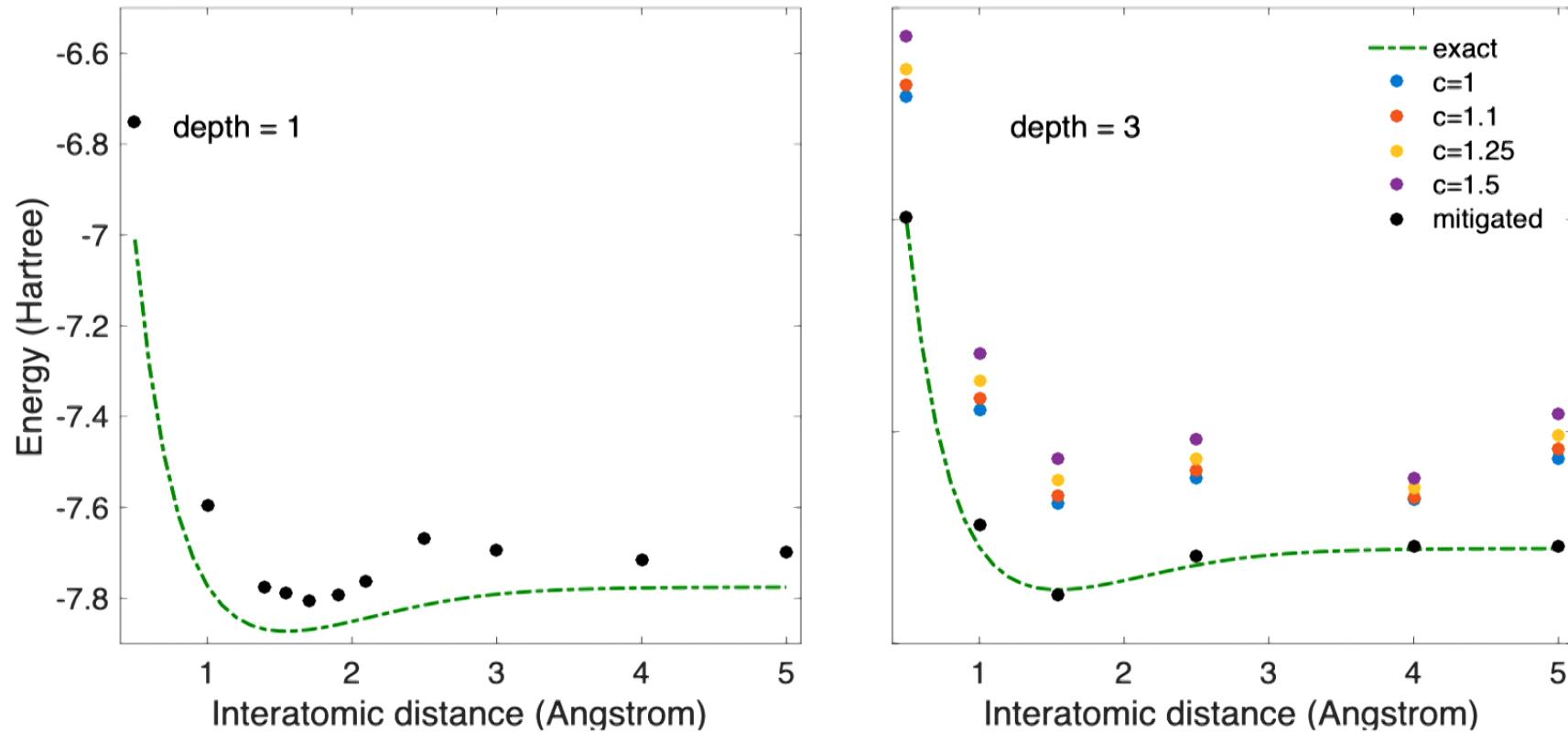
- Identity equivalent random 2Q Clifford sequence on a Bell State



Error Mitigation: 2Q Hydrogen molecule



Error Mitigation: 4Q LiH

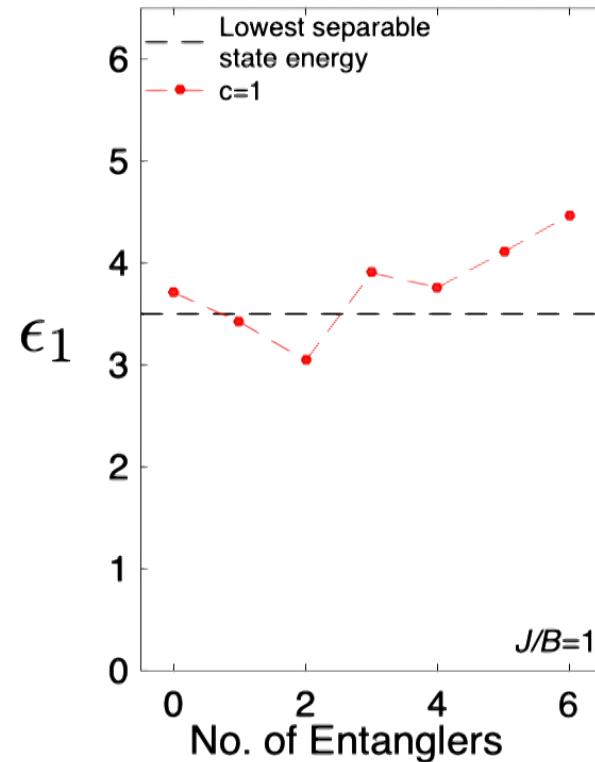


Error Mitigation: 4Q Heisenberg

$$H = J \sum_{\langle ij \rangle} \vec{\sigma}_i \cdot \vec{\sigma}_j + B \sum_i \sigma_i^z$$

$$J/B = 1$$

$$\epsilon_1 = |\langle H \rangle_{Exact} - \langle H \rangle_{Observed}|$$

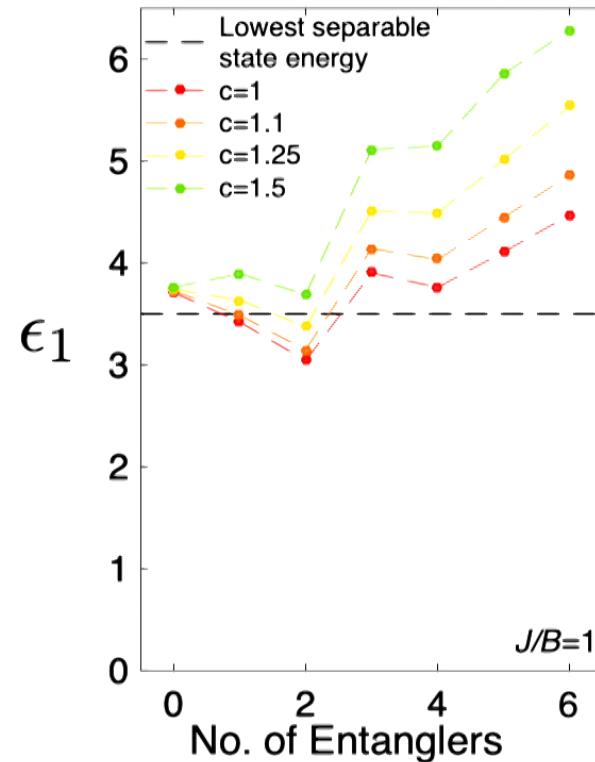


Error Mitigation: 4Q Heisenberg

$$H = J \sum_{\langle ij \rangle} \vec{\sigma}_i \cdot \vec{\sigma}_j + B \sum_i \sigma_i^z$$

$$J/B = 1$$

$$\epsilon_1 = |\langle H \rangle_{Exact} - \langle H \rangle_{Observed}|$$

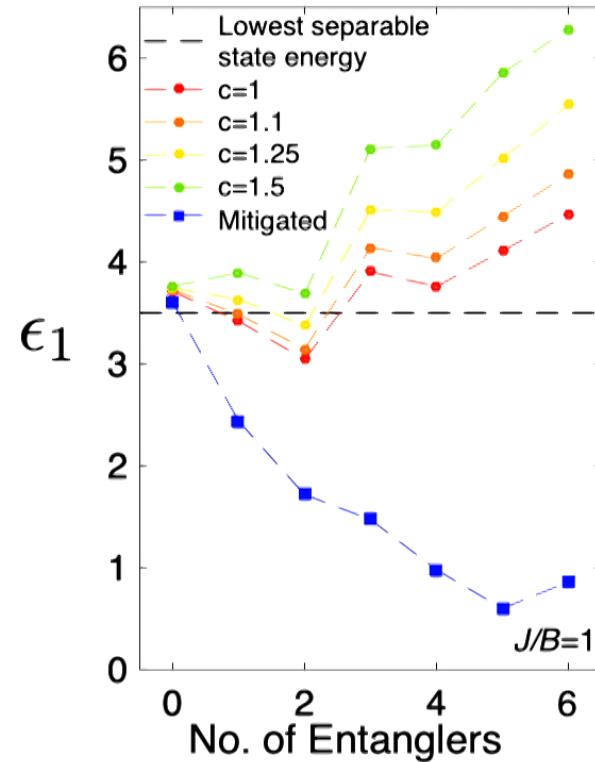


Error Mitigation: 4Q Heisenberg

$$H = J \sum_{\langle ij \rangle} \vec{\sigma}_i \cdot \vec{\sigma}_j + B \sum_i \sigma_i^z$$

$$J/B = 1$$

$$\epsilon_1 = |\langle H \rangle_{Exact} - \langle H \rangle_{Observed}|$$



Supervised learning on NISQ devices

Quantum Kernels:

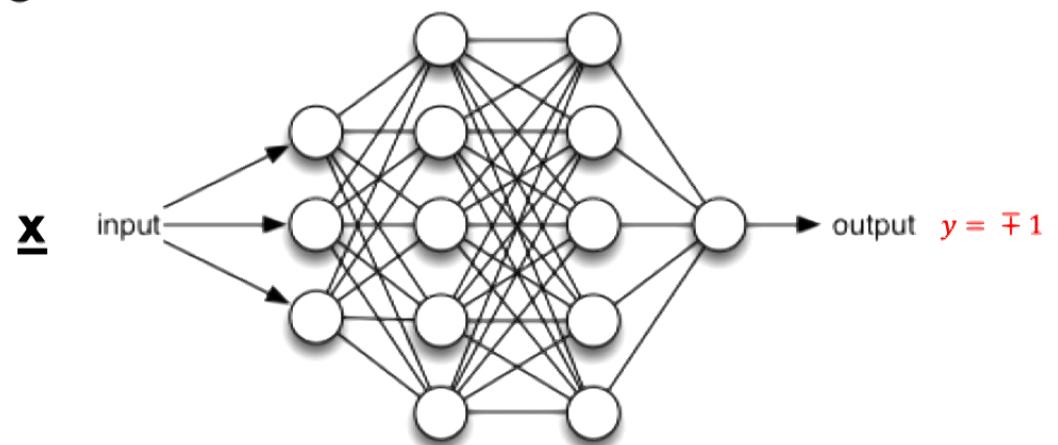
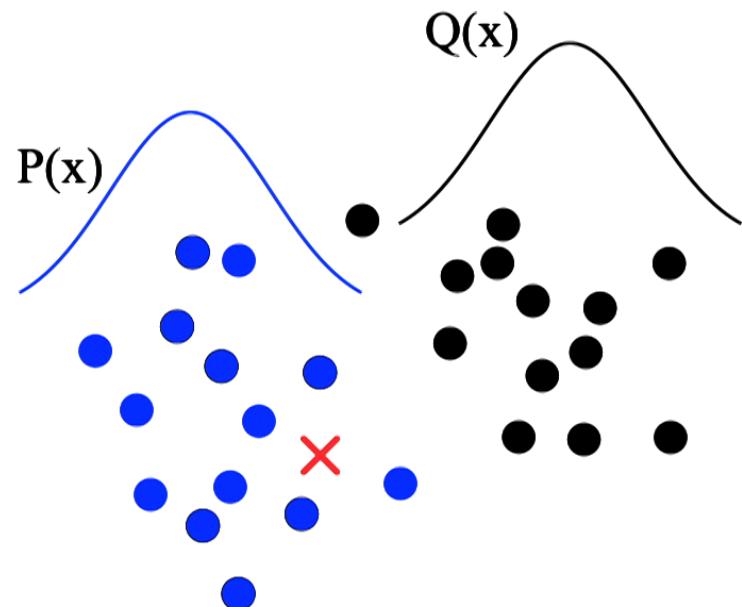
Havlicek, Córcoles, Temme, Harrow, Kandala, Chow, Gambetta -Nature, 567, 209–212 (2019)
Schuld, Killoran - Phys. Rev. Lett. 122, 040504 (2019)

Variational circuits:

Mitarai, Negoro, Kitagawa - Fujii Phys. Rev. A 98, 032309 (2018)
Farhi, Neven - arXiv: 1802.06002
Schuld, Bocharov, Svore, Wiebe arXiv:1804.00633
Havlicek, Córcoles, Temme, Harrow, Kandala, Chow, Gambetta - Nature, 567, 209–212 (2019)
Schuld, Killoran - Phys. Rev. Lett. 122, 040504 (2019)
Grant, Benedetti, Cao, Hallam, Lockhart, Stojovic, Green, Severini - npjqi, 4(1), p.65.

-
-
-

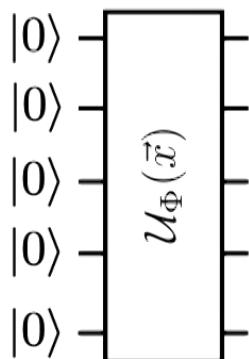
Supervised learning with quantum enhanced feature spaces



A first attempt at a classifier

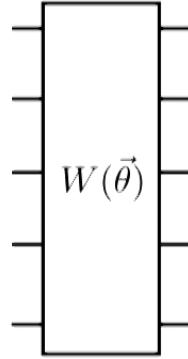
I

Map the data



II

Apply short-depth circuit



III

Measure in canonical basis

$$\mathbf{f} = \sum_{z \in \{0,1\}^N} f(z) |z\rangle\langle z|.$$

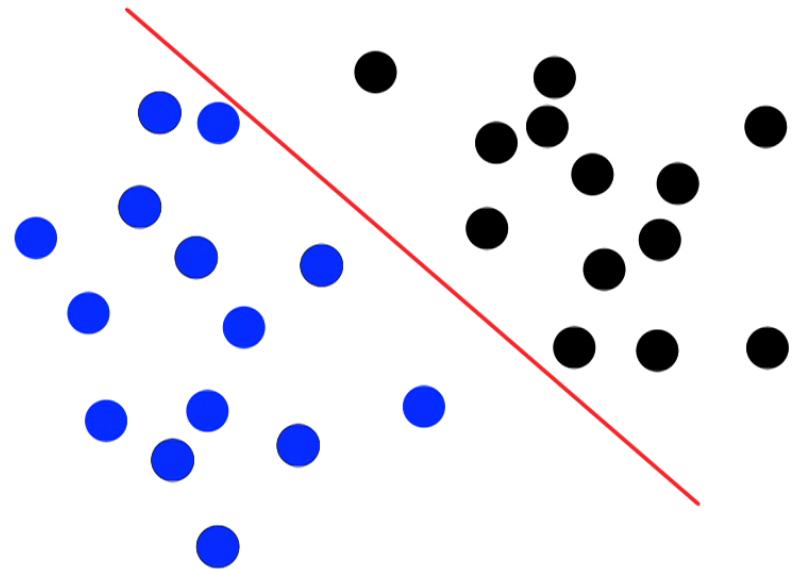
$$\tilde{m}(\vec{x}) = y$$

whenever

$$\hat{p}_y(\vec{x}) > \hat{p}_{-y}(\vec{x}) - yb$$

Classical Support Vector Machines

Training data $S = (\vec{x}_i, y_i)_{i=1,\dots,l}$ $\vec{x}_i \in \mathbb{R}^n$
 $y \in \{+1, -1\}$



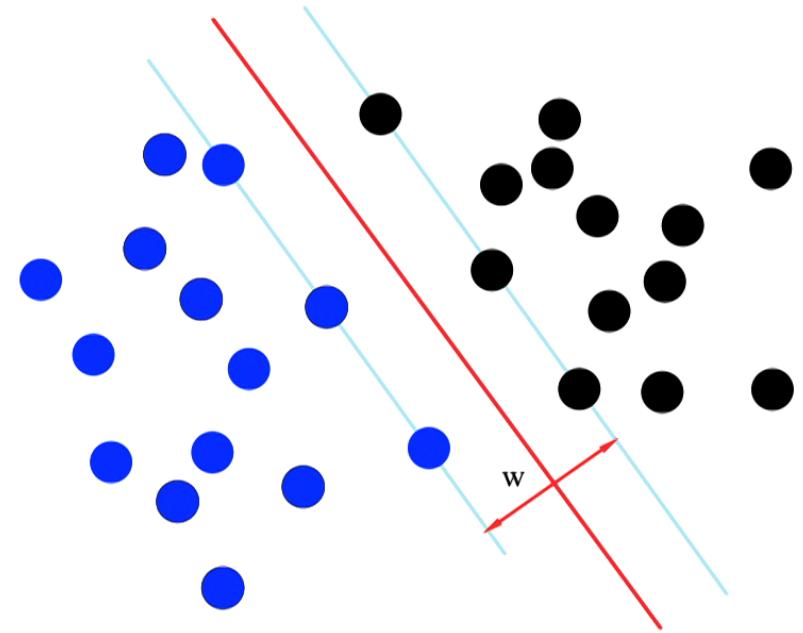
Classical Support Vector Machines

Training data $S = (\vec{x}_i, y_i)_{i=1,\dots,l}$ $\vec{x}_i \in \mathbb{R}^n$
 $y \in \{+1, -1\}$

Training

$$\min_{S,w,b} \frac{1}{2} \|w\|^2$$

$$\text{s.t. } y^{(i)}(w^T x^{(i)} + b) \geq 1, i = 1, \dots, l$$



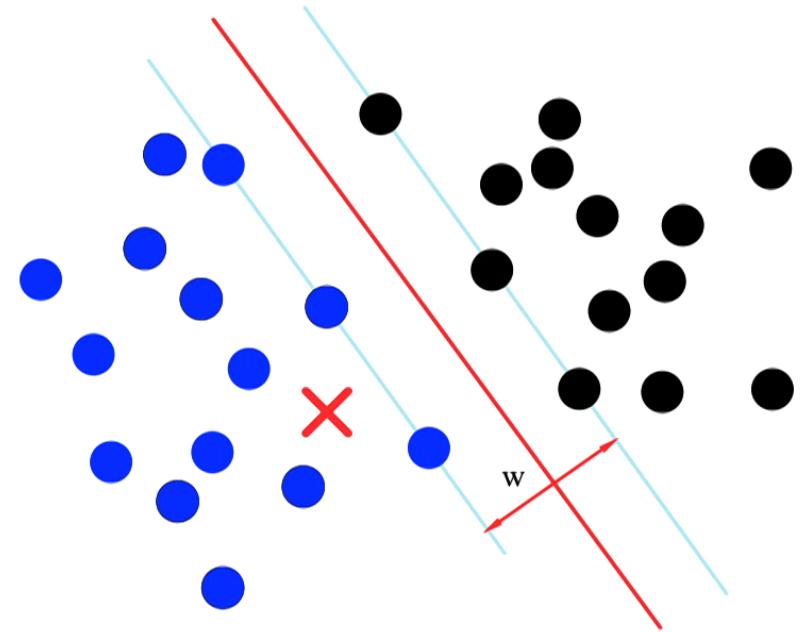
Classical Support Vector Machines

Training data $S = (\vec{x}_i, y_i)_{i=1,\dots,l}$ $\vec{x}_i \in \mathbb{R}^n$
 $y \in \{+1, -1\}$

Training

$$\min_{S,w,b} \frac{1}{2} \|w\|^2$$

$$\text{s.t. } y^{(i)}(w^T x^{(i)} + b) \geq 1, i = 1, \dots, l$$



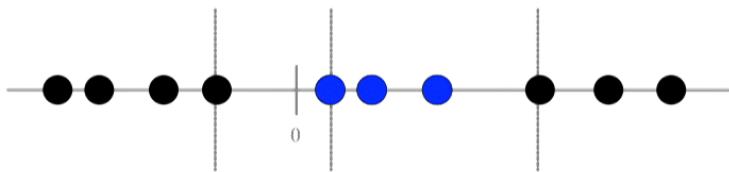
Classification

$$\tilde{m}(\vec{z}) = \text{sign}(\mathbf{w} \circ \vec{z} + b)$$

Choosing the right feature map

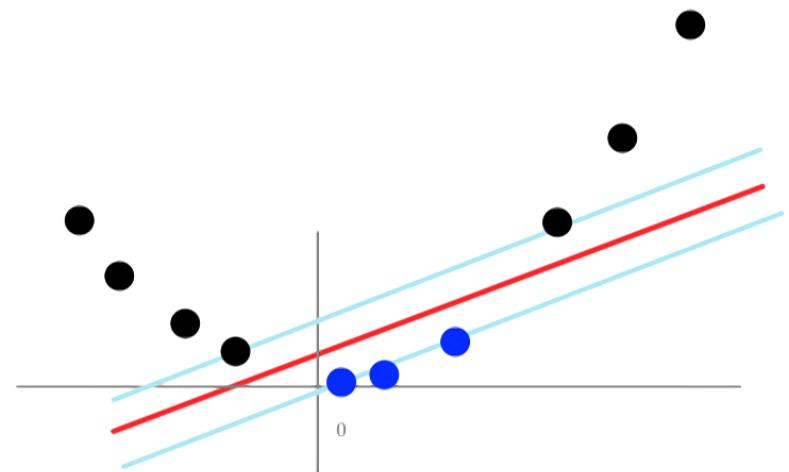
Training data

$$S = (\vec{x}_i, y_i)_{i=1,\dots,t}$$



not linearly separable

$$\Phi : x \rightarrow (x, x^2)$$



linearly separable

Dual optimization problem

maximize

$$L_D(\alpha) = \sum_{i=1}^t \alpha_i - \frac{1}{2} \sum_{i,j=1}^t \alpha_i \alpha_j y_i y_j \underbrace{\Phi(x_i) \circ \Phi(x_j)}_{K(x_i, x_j)}$$

subject to:

$$\sum_{i=1}^t \alpha_i y_i = 0 \quad \alpha_i \geq 0 \quad i = 1 \dots t$$

The first attempt is directly related to linear binary classifiers (SVMs)

- **Decision rule (using the empirical output distribution)**

$$\tilde{m}(\vec{x}) = y \quad \text{whenever} \quad \hat{p}_y(\vec{x}) > \hat{p}_{-y}(\vec{x}) - yb$$

The probability of measuring label y $p_y = \frac{1}{2} (1 + y\langle \Phi(\vec{x}) | W^\dagger(\theta) f W(\theta) | \Phi(\vec{x}) \rangle)$

The first attempt is directly related to linear binary classifiers (SVMs)

- Decision rule (using the empirical output distribution)

$$\tilde{m}(\vec{x}) = y \quad \text{whenever} \quad \hat{p}_y(\vec{x}) > \hat{p}_{-y}(\vec{x}) - yb$$

The probability of measuring label y $p_y = \frac{1}{2} (1 + y \langle \Phi(\vec{x}) | W^\dagger(\theta) f W(\theta) | \Phi(\vec{x}) \rangle)$

- Linear decision function

$$\tilde{m}(\vec{x}) = \text{sign} (\langle \phi(\vec{x}) | W^\dagger f W | \phi(\vec{x}) \rangle + b)$$

The first attempt is directly related to linear binary classifiers (SVMs)

- Decision rule (using the empirical output distribution)

$$\tilde{m}(\vec{x}) = y \quad \text{whenever} \quad \hat{p}_y(\vec{x}) > \hat{p}_{-y}(\vec{x}) - yb$$

The probability of measuring label y $p_y = \frac{1}{2} (1 + y \langle \Phi(\vec{x}) | W^\dagger(\theta) f W(\theta) | \Phi(\vec{x}) \rangle)$

- Linear decision function

$$\tilde{m}(\vec{x}) = \text{sign} (\text{tr} [\Phi(\vec{x}) \mathbf{w}] + b)$$

normal $\mathbf{w} = W^\dagger \mathbf{f} W$
Feature map $\Phi(\vec{x}) = |\phi(\vec{x})\rangle\langle\phi(\vec{x})|$

The first attempt is directly related to linear binary classifiers (SVMs)

- Decision rule (using the empirical output distribution)

$$\tilde{m}(\vec{x}) = y \quad \text{whenever} \quad \hat{p}_y(\vec{x}) > \hat{p}_{-y}(\vec{x}) - yb$$

$$\text{The probability of measuring label } y \quad p_y = \frac{1}{2} (1 + y \langle \Phi(\vec{x}) | W^\dagger(\theta) f W(\theta) | \Phi(\vec{x}) \rangle)$$

- Linear decision function

$$\text{normal} \quad \mathbf{w} = W^\dagger \mathbf{f} W$$

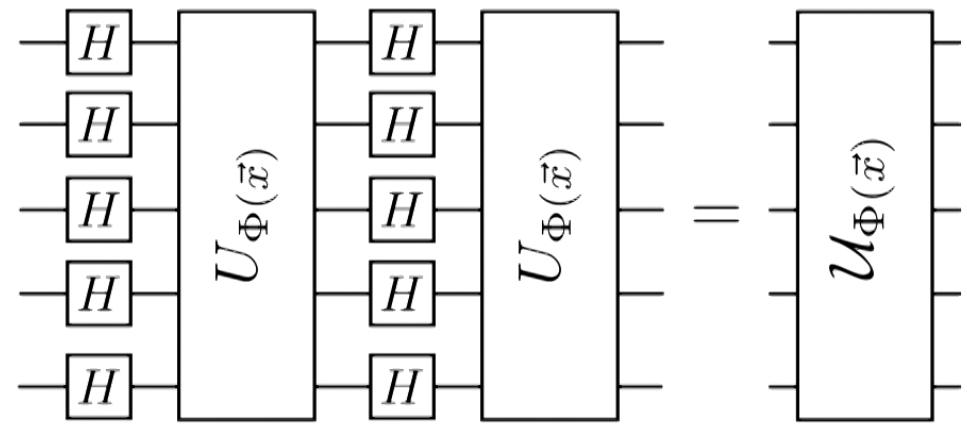
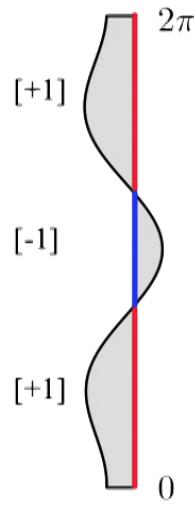
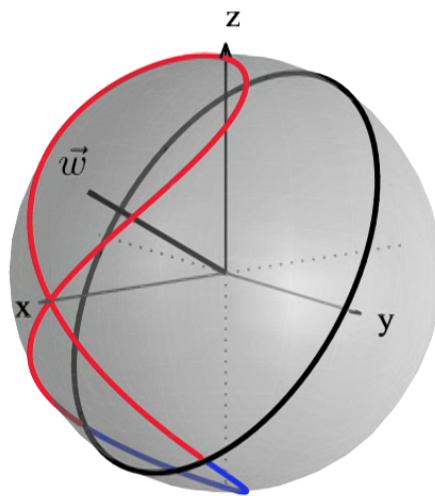
$$\tilde{m}(\vec{x}) = \text{sign} (\text{tr} [\Phi(\vec{x}) \mathbf{w}] + b)$$

$$\text{Feature map} \quad \Phi(\vec{x}) = |\phi(\vec{x})\rangle\langle\phi(\vec{x})|$$

- Construction becomes classically efficient for a “trivial” Kernel

$$K(\vec{x}, \vec{y}) = \prod_{i=1}^n |\langle \phi_i(\vec{x}) | \phi_i(\vec{y}) \rangle|^2$$

Quantum enhanced feature space

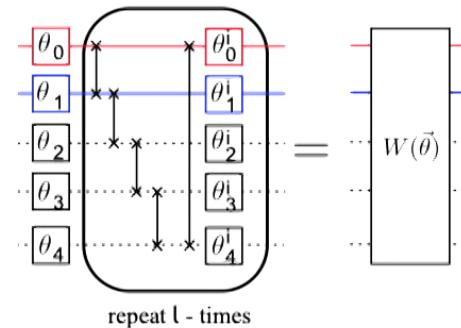


$$\Phi : \vec{x} \in \Omega \rightarrow |\Phi(\vec{x})\rangle\langle\Phi(\vec{x})|$$

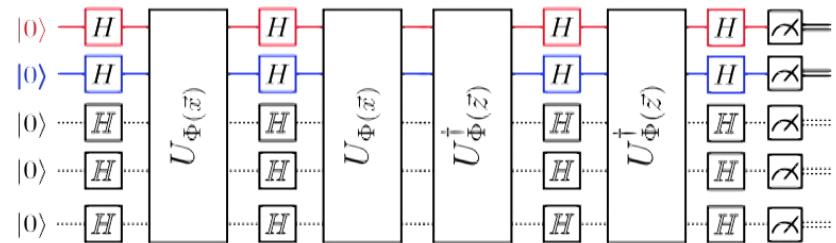
$$U_{\Phi(\vec{x})} = \exp \left(i \sum_{S \subseteq [n]} \phi_S(\vec{x}) \prod_{i \in S} Z_i \right)$$

Two methods

- **Method 1: Quantum variational classification**



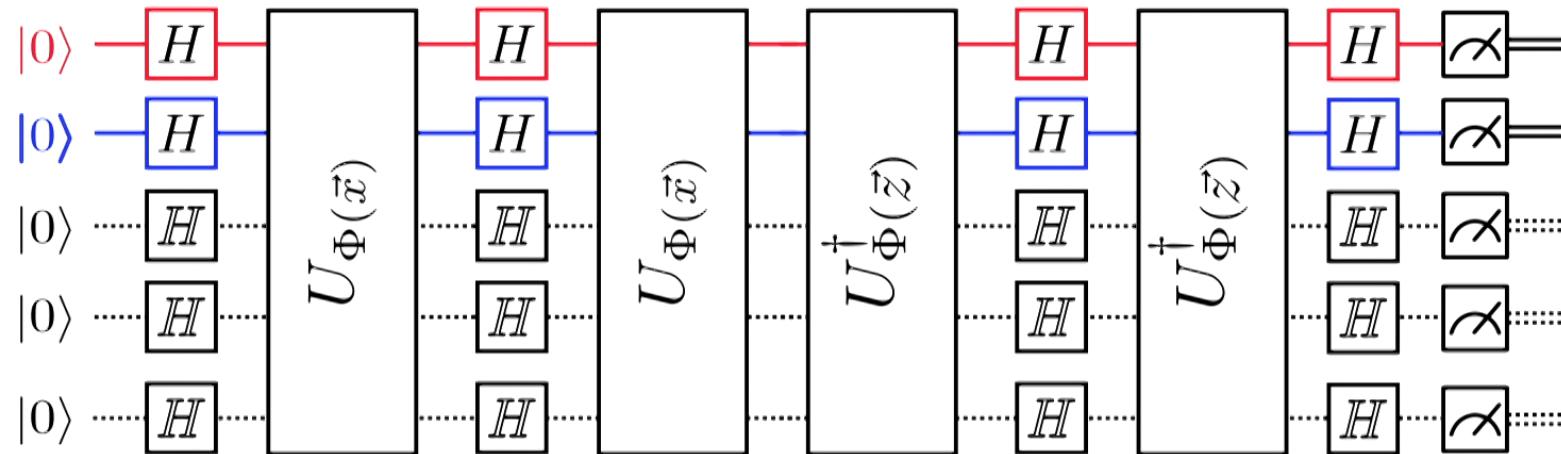
- **Method 2: Direct Kernel estimation**



Method 2: estimating the quantum Kernel

- Direct estimation of the kernel as fidelity

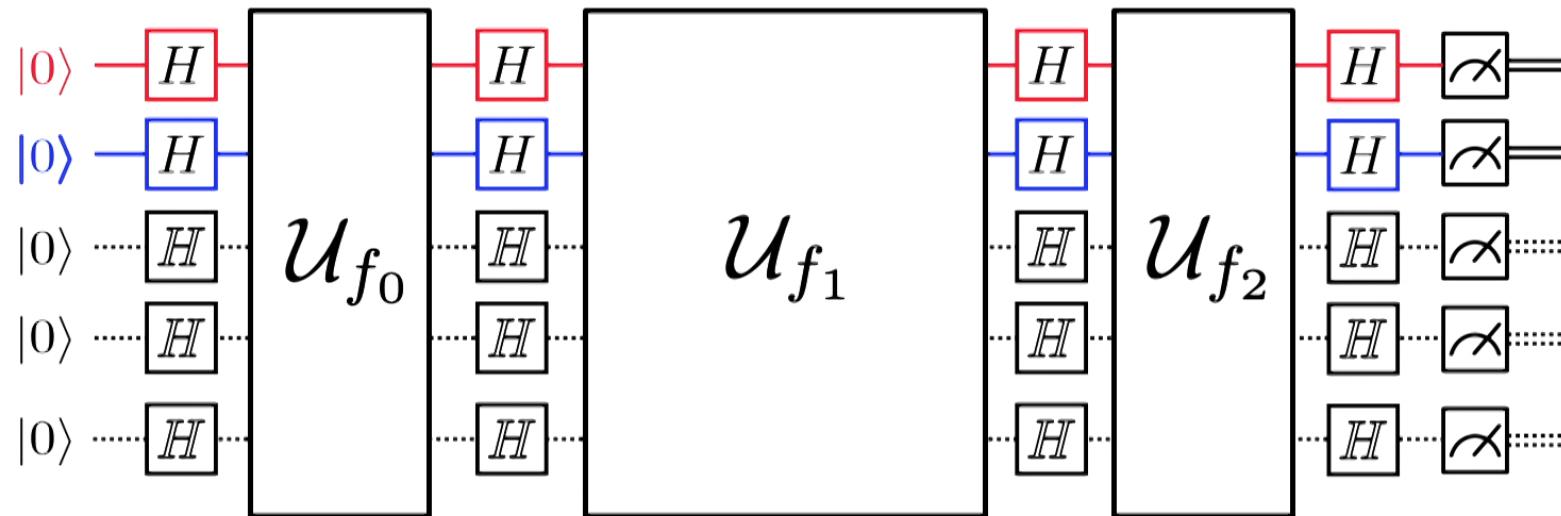
$$K(\vec{x}, \vec{z}) = |\langle \Phi(\vec{x}) | \Phi(\vec{z}) \rangle|^2 = |\langle 0^n | U_{\Phi(\vec{x})}^\dagger U_{\Phi(\vec{z})} | 0^n \rangle|^2$$



Method 2: estimating the quantum Kernel

- Direct estimation of the kernel as fidelity

$$K(\vec{x}, \vec{z}) = |\langle \Phi(\vec{x}) | \Phi(\vec{z}) \rangle|^2 = |\langle 0^n | \mathcal{U}_{\Phi(\vec{x})}^\dagger \mathcal{U}_{\Phi(\vec{z})} | 0^n \rangle|^2$$



Similar to 3-fold Forrelation

S. Aaronson and A. Ambainis SIAM Journal on Computing 47.3 (2018): 982-1038.

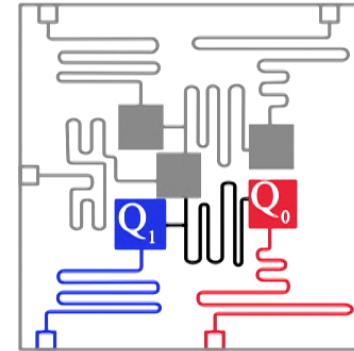
What data do we classify in the experiment?

- We consider “proper learning”

Choose a feature map circuit with

$$\phi_{\{i\}}(\vec{x}) = x_i$$

$$\phi_{\{1,2\}}(\vec{x}) = (\pi - x_1)(\pi - x_2)$$



Assign label according to:

$$m(\vec{x}) = +1 \quad \langle \Phi(\vec{x}) | V^\dagger \mathbf{f} V | \Phi(\vec{x}) \rangle \geq \Delta$$

$$m(\vec{x}) = -1 \quad \langle \Phi(\vec{x}) | V^\dagger \mathbf{f} V | \Phi(\vec{x}) \rangle \leq -\Delta$$

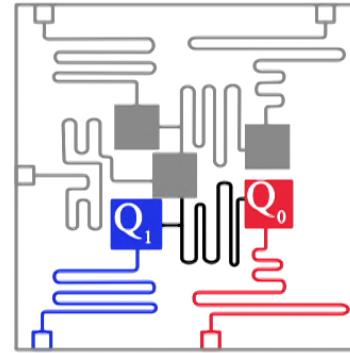
What data do we classify in the experiment?

- We consider “proper learning”

Choose a feature map circuit with

$$\phi_{\{i\}}(\vec{x}) = x_i$$

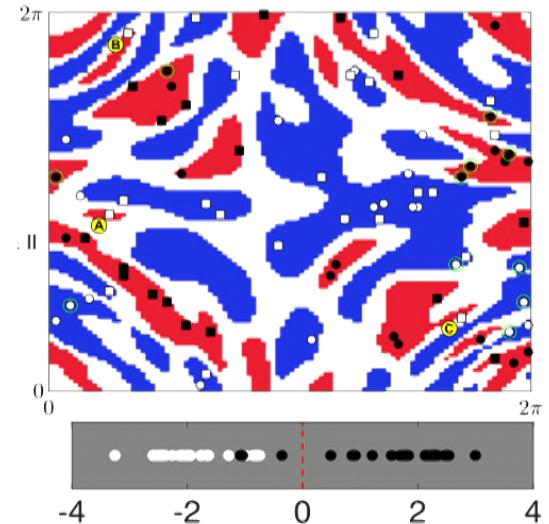
$$\phi_{\{1,2\}}(\vec{x}) = (\pi - x_1)(\pi - x_2)$$



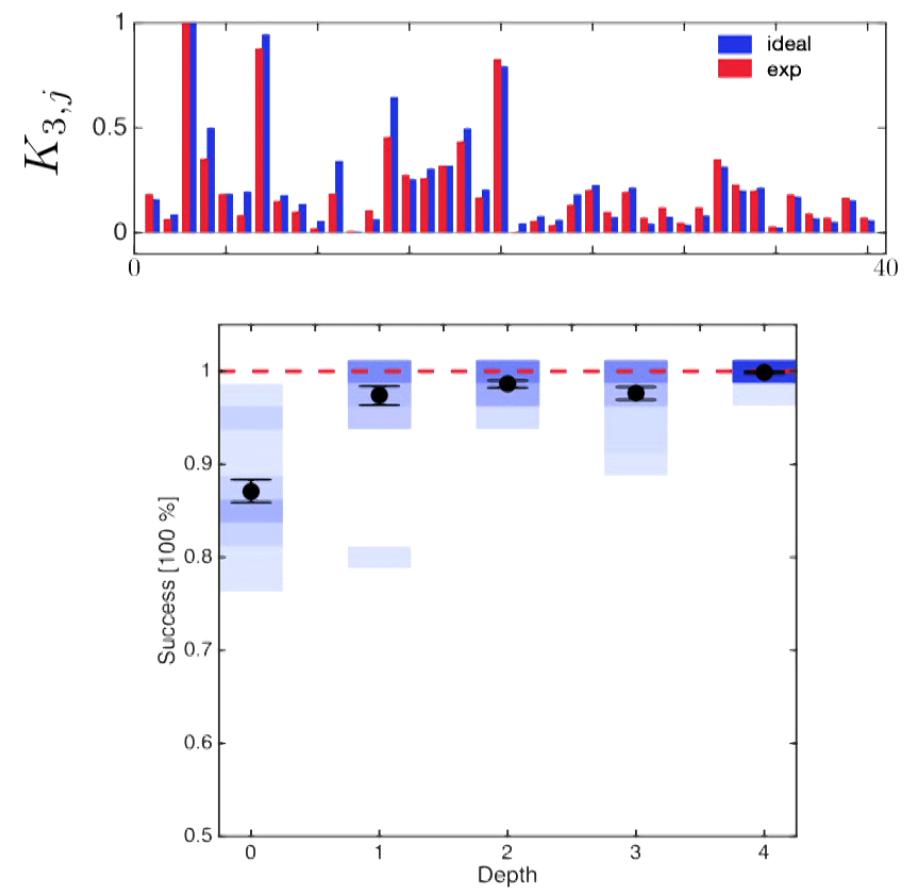
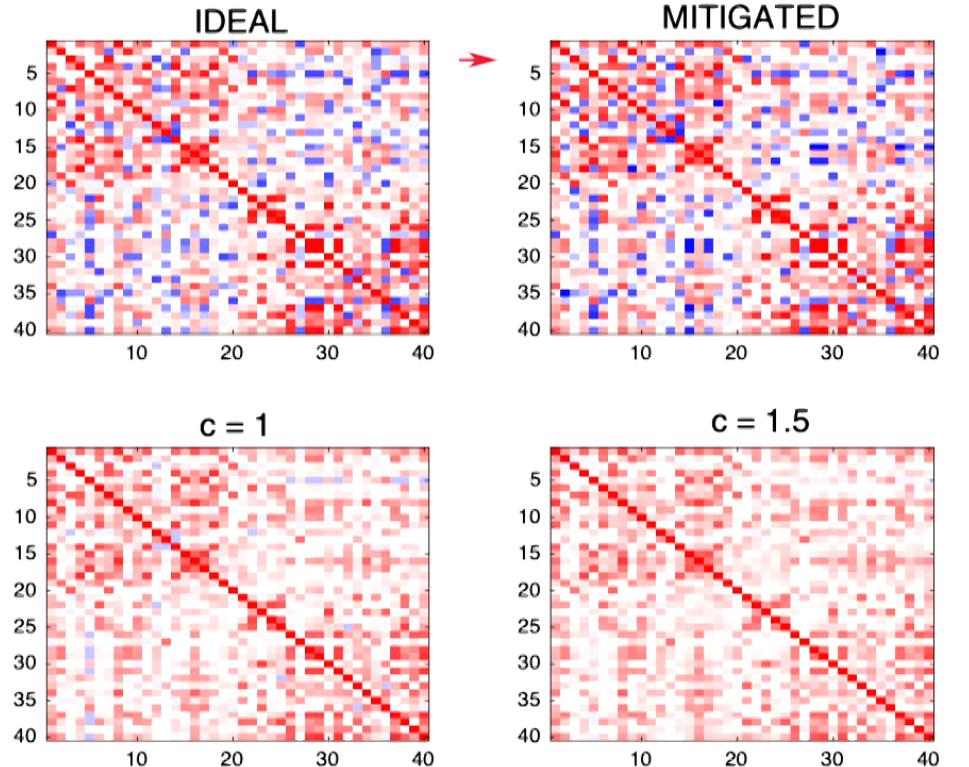
Assign label according to:

$$m(\vec{x}) = +1 \quad \langle \Phi(\vec{x}) | V^\dagger f V | \Phi(\vec{x}) \rangle \geq \Delta$$

$$m(\vec{x}) = -1 \quad \langle \Phi(\vec{x}) | V^\dagger f V | \Phi(\vec{x}) \rangle \leq -\Delta$$



Experimental Results



Conclusions

- Error mitigation can significantly enhance capabilities of near-term quantum hardware for quantum simulation
- A scheme to look for quantum advantage in ML

References:

Error mitigation:

Temme, Brayvi, Gambetta, Phys. Rev. Lett. 119, 180509 (2017),
Kandala, Temme, Corcoles, Mezzacapo, Chow, Gambetta, Nature vol. 567, 491 (2019)
Li and Benjamin, PRX 7, 021050 (2017) ...

Machine learning

Havlicek, Córcoles, Temme, Harrow, Kandala, Chow, Gambetta, Nature vol. 567, 209–212 (2019)
Schuld, Killoran - Phys. Rev. Lett. 122, 040504 (2019)