Circuit Benchmarking is all you need for application-based benchmarking

Joseph Emerson

Professor of Applied Math, Institute of Quantum Computing, University of Waterloo

Director of Quantum Strategy, Quantum Engineering Solutions, Keysight

KEYSIGHT
TECHNOLOGIES

Overview

I. **Intro and Background:**
   - Randomized error diagnostics vs system performance: closing the gap
   - Role of randomized compiling

II. Cycle-level Benchmarking
   - Cycle benchmarking and cycle error reconstruction
     - Learning the process fidelity and Pauli error rates under parallel instruction sets across n-qubits for large n

III. System-level Benchmarking
   - Performance guarantee for applications
   - A scalable quantum volume benchmark?

The Problem of Errors: they are here to stay

- NISQ: Near-term noisy/imperfect qubits: requires error assessment & error suppression & error mitigation

- Fault-tolerance: Long-term error-corrected logical qubits: “better but still noisy”

Myth: FT-QEC will (eventually) deliver essentially “error-free” logical qubits

- Quantum computers *will always* require error assessment, error suppression *and* solution validation, even at logical level
  - Fault-tolerance only guarantees correcting subset of “good’ errors.
  - Physically realistic error models are always a mix of “good” and ”bad” errors

Mind the Gap!

- Application-based benchmarks have very limited usefulness:
  - They are not generally useful to predict any thing other than the very specific algorithm they ran
  - They generally are not scalable unless they are trivial
  - If they are trivial then it is easy to get misleading results: easy to compile away the complexity

- Almost all good/universal benchmarks are based on randomization methods:
  - Randomized Benchmarking, Cycle Benchmarking, Quantum Volume, Quantum Supremacy, etc
  - Application performance can be poorly predicted by these standard methods (without the solution below)

- Do we need application-based benchmarks to solve this problem? Absolutely no we don’t!

- The simple solution is to use randomization methods **for both benchmarking *and* for algorithm implementation** – THIS CLOSES THE GAP, while also *improving algorithm performance*!
  - **Solution: Cycle benchmarking for Circuit Benchmarking via Randomized Compiling**

Explaining the gap: coherent errors

Coherent/calibration error: over-rotation error θ about z-axis:

$$U_z(\theta) = \exp\left(i\frac{\theta}{2}Z\right) = \begin{pmatrix} \cos(\theta/2) + i\sin(\theta/2) & 0 \\ 0 & \cos(\theta/2) - i\sin(\theta/2) \end{pmatrix},$$

has error super-operator: $$\mathcal{E} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Note that the off-diagonal terms (coherent errors) scale as: $\sin(\theta) \sim \theta$

But diagonal terms scale as: $\cos(\theta) \sim 1 - \theta^2$

These properties hold for arbitrary unitary operations in all dimensions!

Pauli Error Channels – a simpler error model

$$\Lambda(\rho) = \sum_{\alpha=1}^{D^2} p_\alpha P_\alpha \rho P_\alpha$$

$$\Lambda(\rho) = \sum_{\alpha,\beta} \tilde{\chi}_{\alpha,\beta} P_\alpha \rho P_\beta$$

where

$$\tilde{\chi}_{\alpha,\beta} = \delta_{\alpha\beta} p_\alpha,$$

- Dominant errors are coherent errors
  - Finite precision (and drift) of the analog classical control (for qubit rotations and qubit couplings) introduces control errors, which are fundamentally *coherent errors*

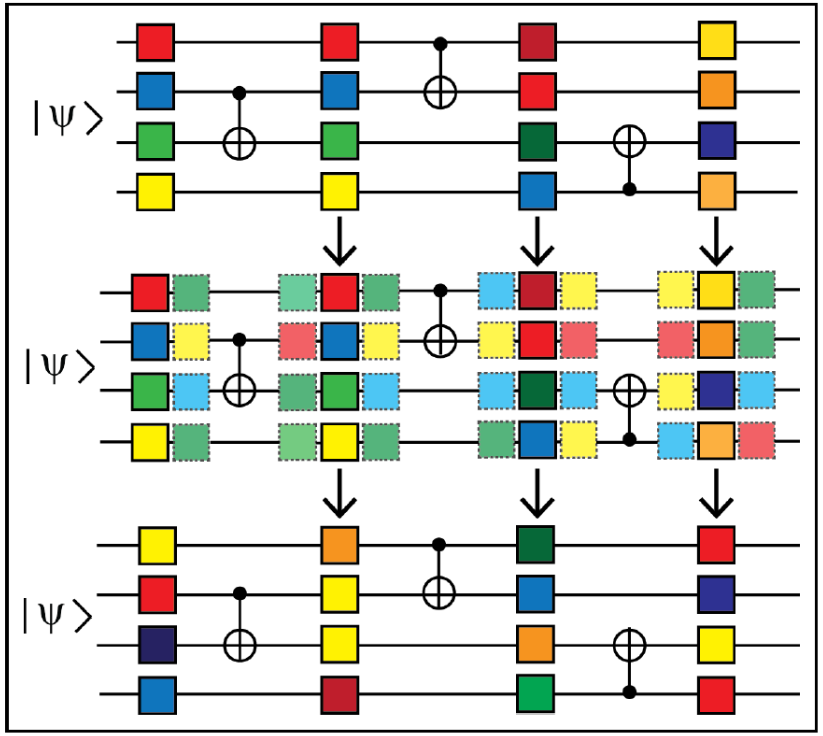
- So Pauli error channels are physically unrealistic errors **unless randomization is applied**

- Randomization removes coherent errors (*no cross-terms in the chi-matrix)*

Randomized Compiling

Randomized Compiling: Twirling Noise during Universal Circuits

- Express circuit as K cycles of alternating rounds of ‘easy gates’ and ‘hard gates’
- Easy gates are subset of single-qubit gates
- Insert randomizing single-qubit twirling gates
- Compute inverse after the hard gate
- Compile inverse, random and original single qubit gate into one operation!



Wallman and Emerson, arXiv:1512.01098

Key ideas:

Averaging over *compensated* Pauli operators suppresses **off-diagonal terms,** eliminating coherent (calibration/drift) errors.

Minimal or no increase to circuit depth.

Unlike DD does not require knowledge of noise axis

Motivation for RC

Calibration/coherent/unitary error:

$$U(z,\theta) = \exp\left(-i\frac{\theta}{2}\sigma_z\right)$$
$$= \cos(\theta/2) - i\sigma_z \sin(\theta/2)$$
$$= \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}.$$

Effect of RC: →

Stochastic (decoherence) error:

$$\mathcal{E}(\rho) = \sum_{P \in \mathbb{P}^{\otimes n}} c_P P \rho P^\dagger,$$

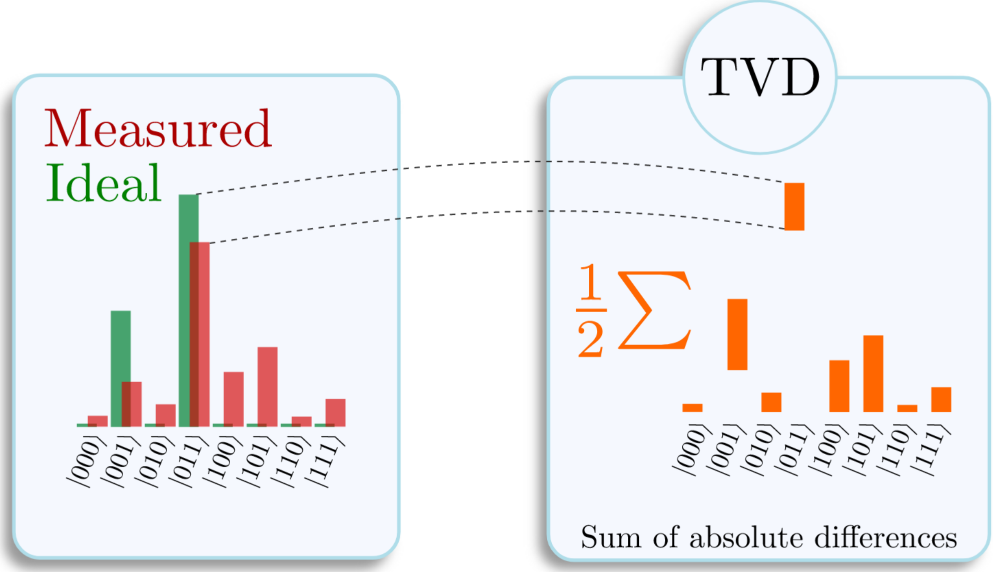
$$\boxed{d_{\mathrm{TV}}(\mathcal{P}_{\mathrm{RC}}, \mathcal{P}_{\mathrm{ideal}}) \leq \epsilon_\diamond(\mathcal{E}_{\mathrm{RC}} - \mathcal{I}) = r(\mathcal{E}_{\mathrm{RC}})\frac{d+1}{d}.}$$

$$r(\mathcal{E})\frac{d+1}{d} \leq \epsilon_\diamond(\mathcal{E} - \mathcal{I}) \leq \sqrt{r(\mathcal{E})}\sqrt{d(d+1)},$$

Very, very good

Very, very bad

$$d_{\mathrm{TV}}(\mathcal{P}, \mathcal{P}_{\mathrm{ideal}}) \leq \epsilon_\diamond(\mathcal{E} - \mathcal{I}).$$



Motivation for RC

EXAMPLE of calibration/coherent/unitary error:

$$\begin{aligned} U(z,\theta) &= \exp\left(-i\frac{\theta}{2}\sigma_z\right) \\ &= \cos(\theta/2) - i\sigma_z \sin(\theta/2) \\ &= \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}. \end{aligned}$$

Why this matters:

$\sin(\theta) \sim \theta$ eg, error of 10^{-2}

$\cos(\theta) \sim 1 - \theta^2$ eg, error of 10^{-4}

$$\sqrt{r(\mathcal{E})} \simeq \theta \text{ to } r(\mathcal{E}) \simeq \theta^2$$

Errors impact on applications

Errors as RB sees them

Pauli-transfer Matrix:

$$\Lambda = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \xrightarrow{\text{Effect of RC:}} \Lambda = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & 0 & 0 \\ 0 & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Hashim et al,
arxiv: 2010.00215

Randomized Compiling

RC works for universal circuits: NISQ circuits and/or logical-level circuits

RC is **necessary for error mitigation**

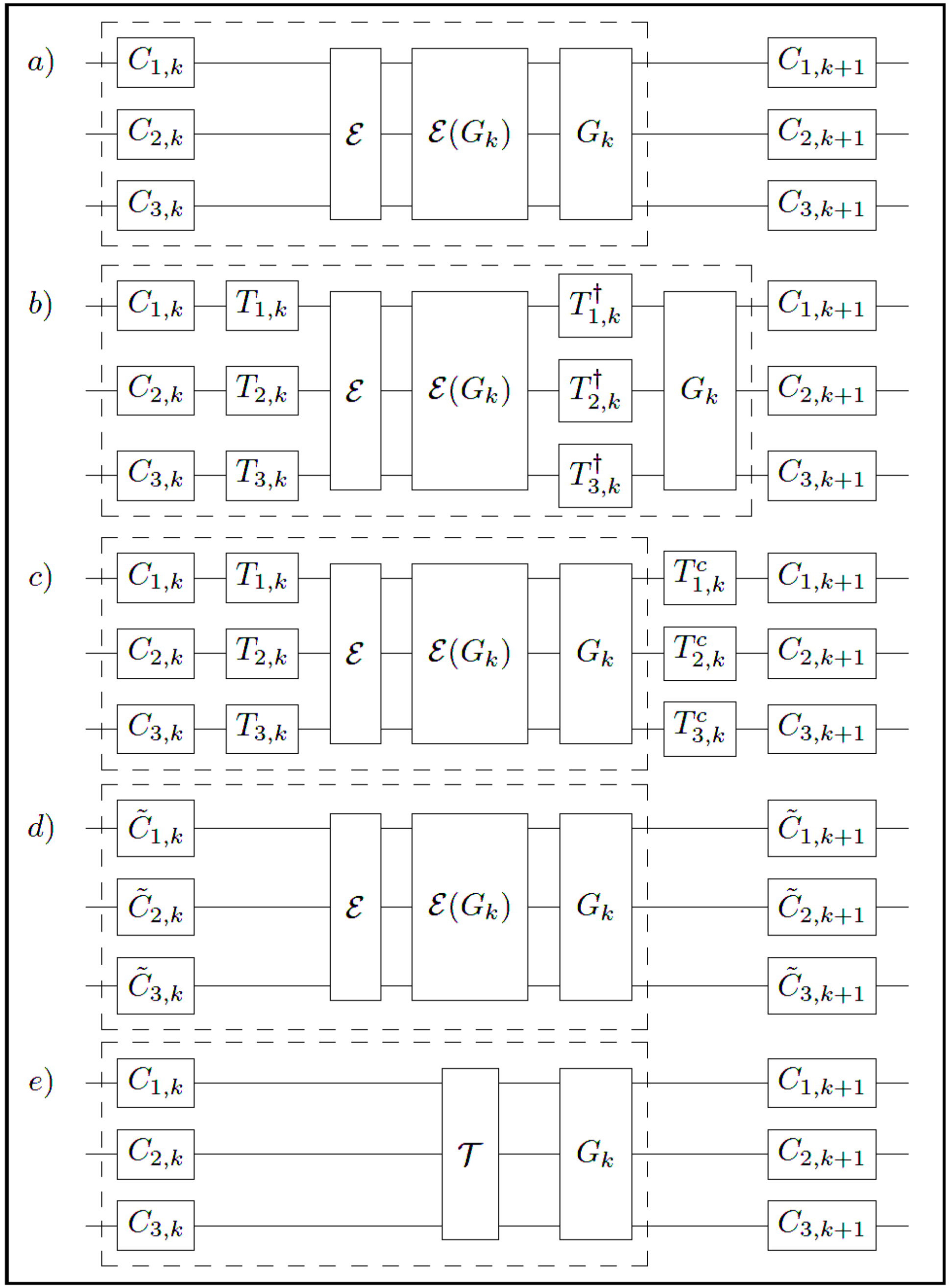
RC is not Pauli Frame Randomization (PFR); PFR only works for Clifford circuits, because you need to be able to track the Pauli frame

In RC, we correct for the frame changes locally via a *deformed* twirl:

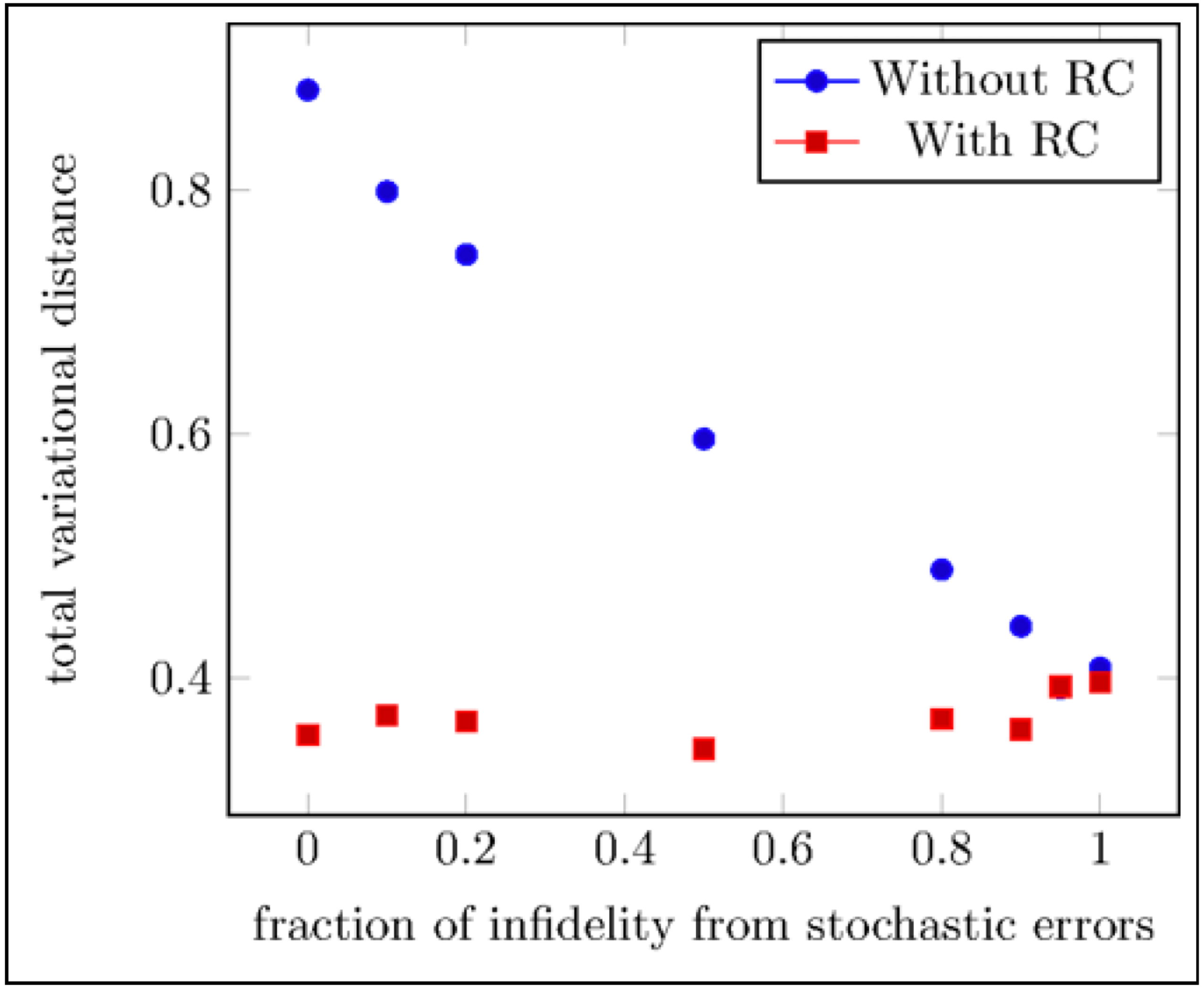
$$\vec{T}_k^c = G_k \vec{T}_k^\dagger G_k^\dagger \qquad \tilde{C}_k = \vec{T}_k \vec{C}_k \vec{T}_{k-1}^c$$

Compensating operations be efficiently pre-computed or on the fly.

Wallman and Emerson, arXiv:1512.01098



RC Suppresses Coherent Errors

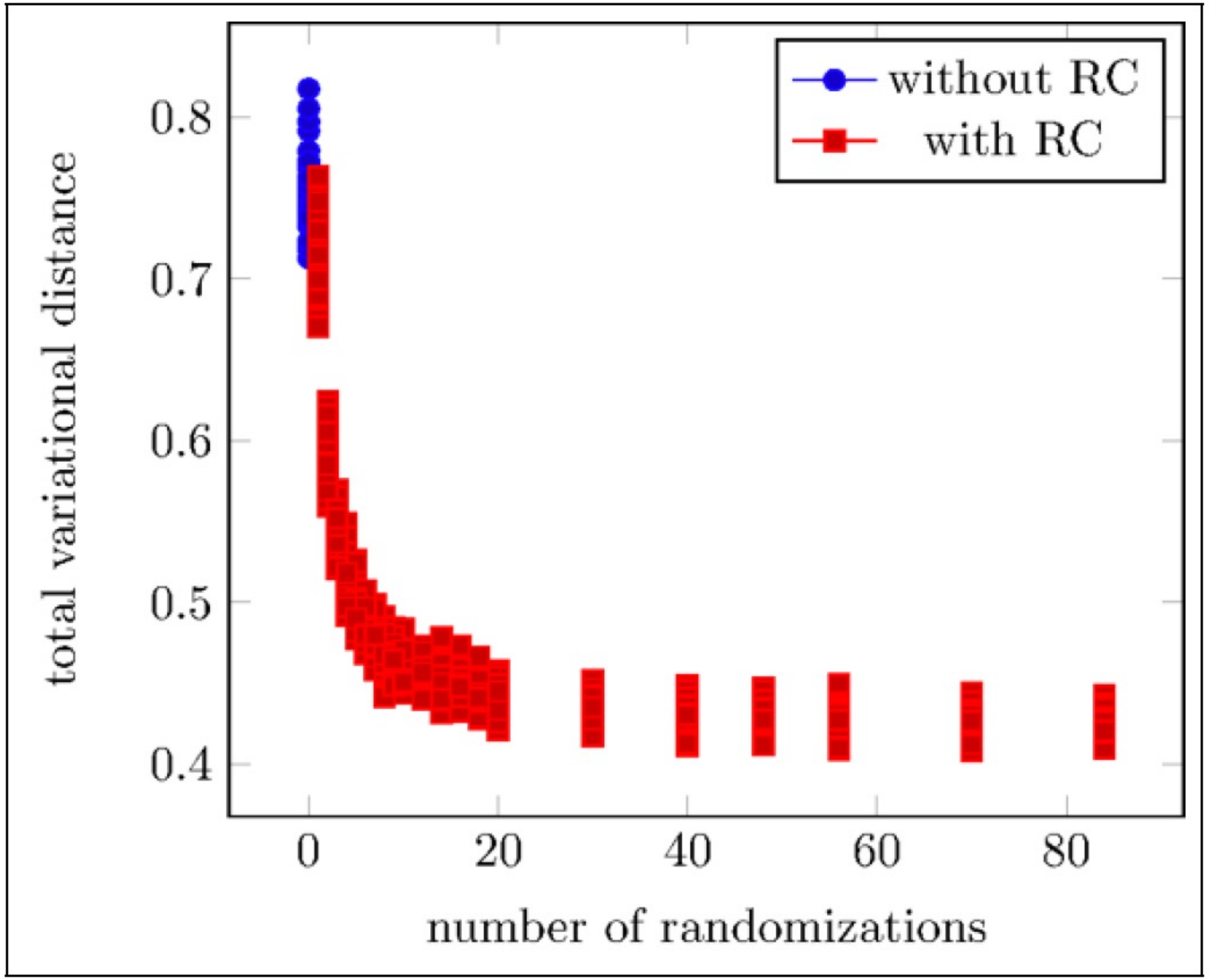


Probability of incorrect solution

Wallman and Emerson, arXiv:1512.01098

RC Requires a Few Randomizations (Hoeffding Inequality)

Probability of incorrect solution



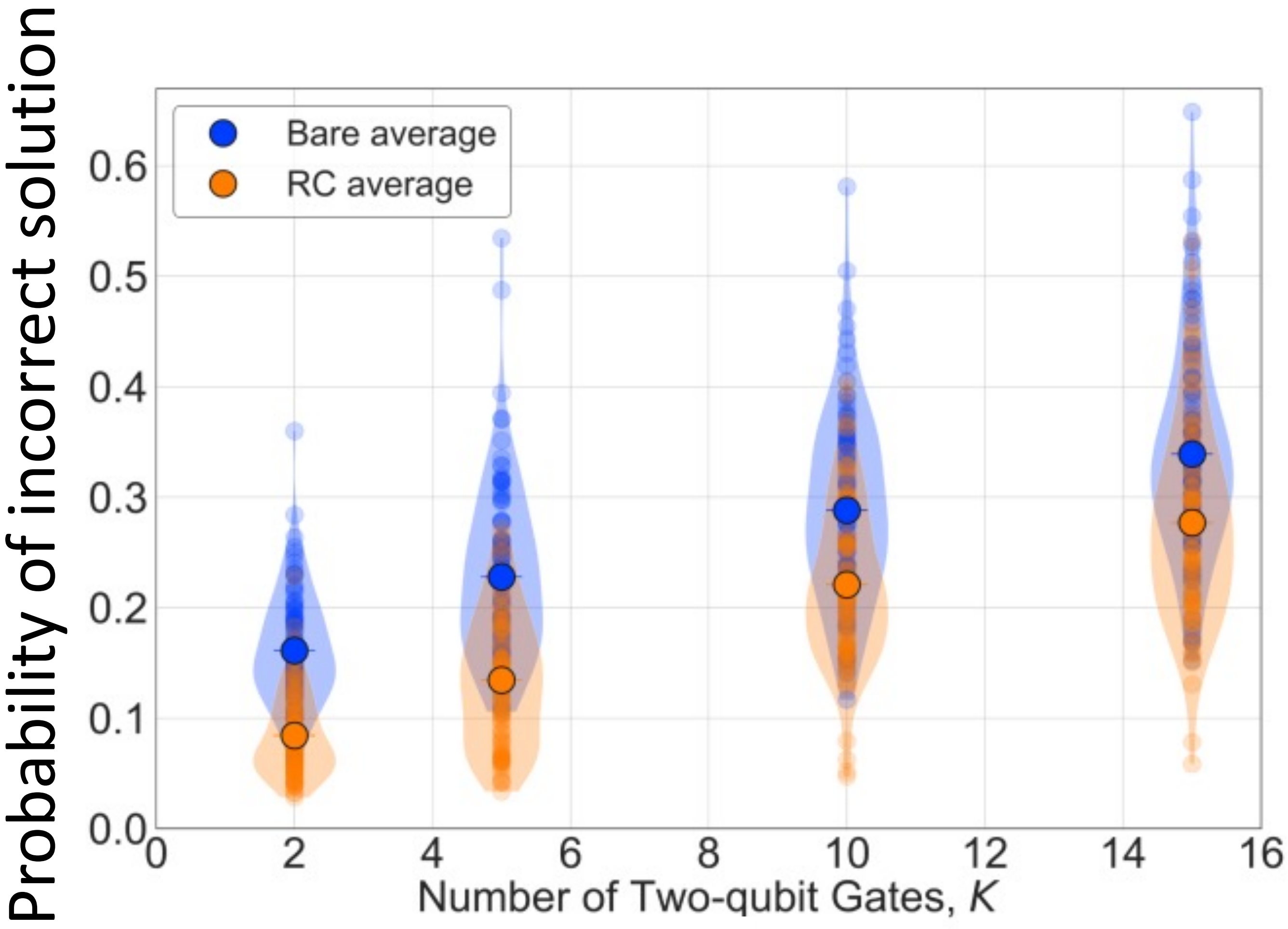
Number of randomizations needed is small, around 20, **and independent of the quantum computer size!**

Proof: Hoeffding Inequality

Wallman and Emerson, arXiv:1512.01098

Example: Randomized Compiling (RC) on LBNL superconducting system

Experimental data from LBNL-AQT comparing circuits with and without *RC*



Suppression works for universal circuits & even for **random circuits**!

Hashim et al, arxiv: 2010.00215

Overview

I. Intro and Background:

- Randomized error diagnostics vs system performance: closing the gap
- Role of randomized compiling

II. **Cycle Benchmarking**

- Cycle benchmarking and cycle error reconstruction
  - Learning the process fidelity and Pauli error rates under parallel instruction sets across n-qubits for large n

III. System-level Benchmarking

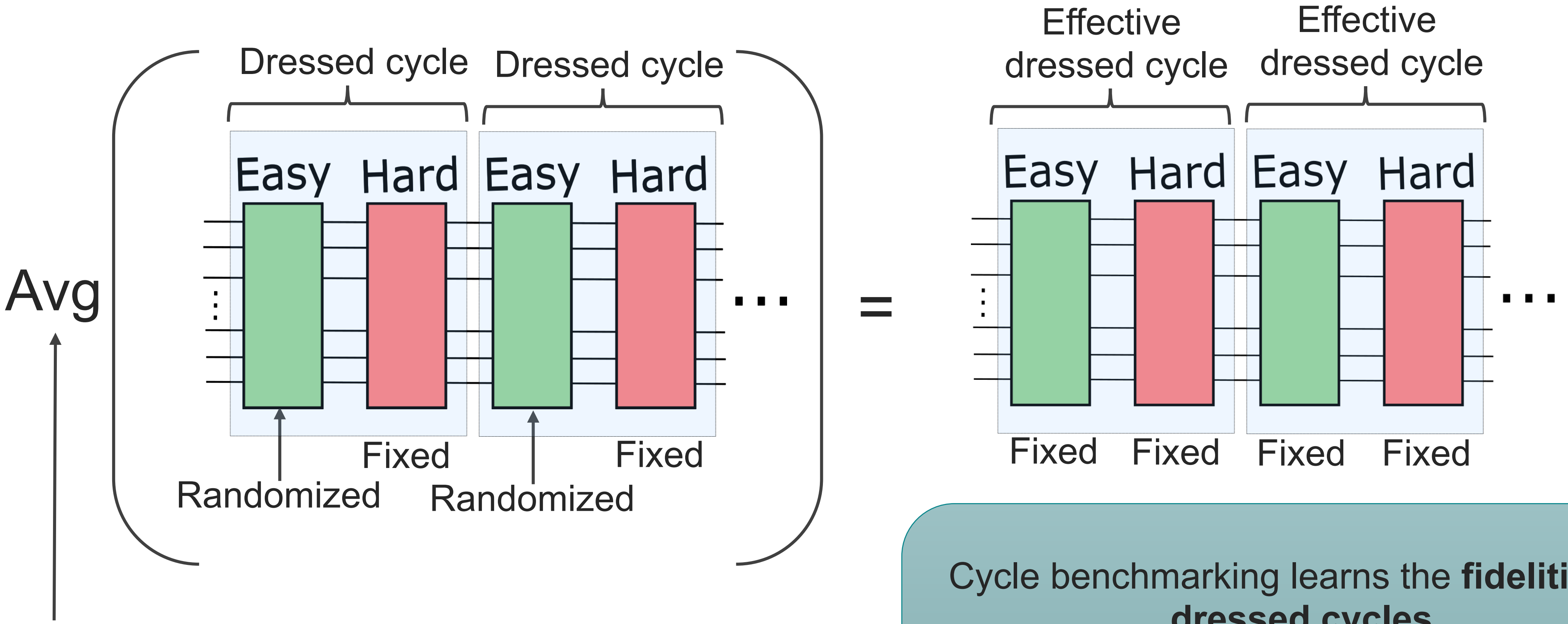
- Performance guarantee for applications
- A scalable quantum volume benchmark

Defining Hard Gate Rounds (hard cycles)

- Any universal circuit can be put into a canonical form, consisting of a sequence of easy and hard gate rounds
- Each round is also called layer or a (clock) cycle
- If the round contains a two-qubit Clifford gate, a Hadadmard, or a T-gate, then it is a “hard gate round”.
- T-gates are not transversal and are usually implemented via magic state injection
- There are very large overheads for reducing the error rate on T-gates
- All other Clifford single qubit gates are easy gate rounds
- Errors on easy gate rounds are typically an order of magnitude smaller than those on hard gate rounds
- A hard gate round preceded by an easy gate round is called a dressed hard gate or dressed cycle

Cycles in the context of randomized compiling

Cycles vs “effective dressed cycles”



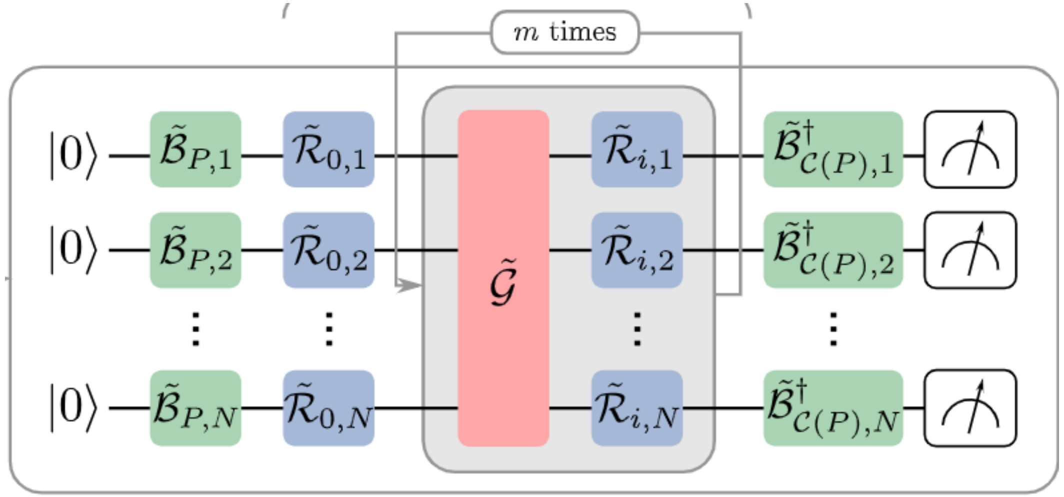
Often improves circuit fidelities by
making error more stochastic, as opposed to coherent.

Cycle benchmarking learns the **fidelities of dressed cycles**

- These dressed cycles match the RC-based implementation of any application circuit.

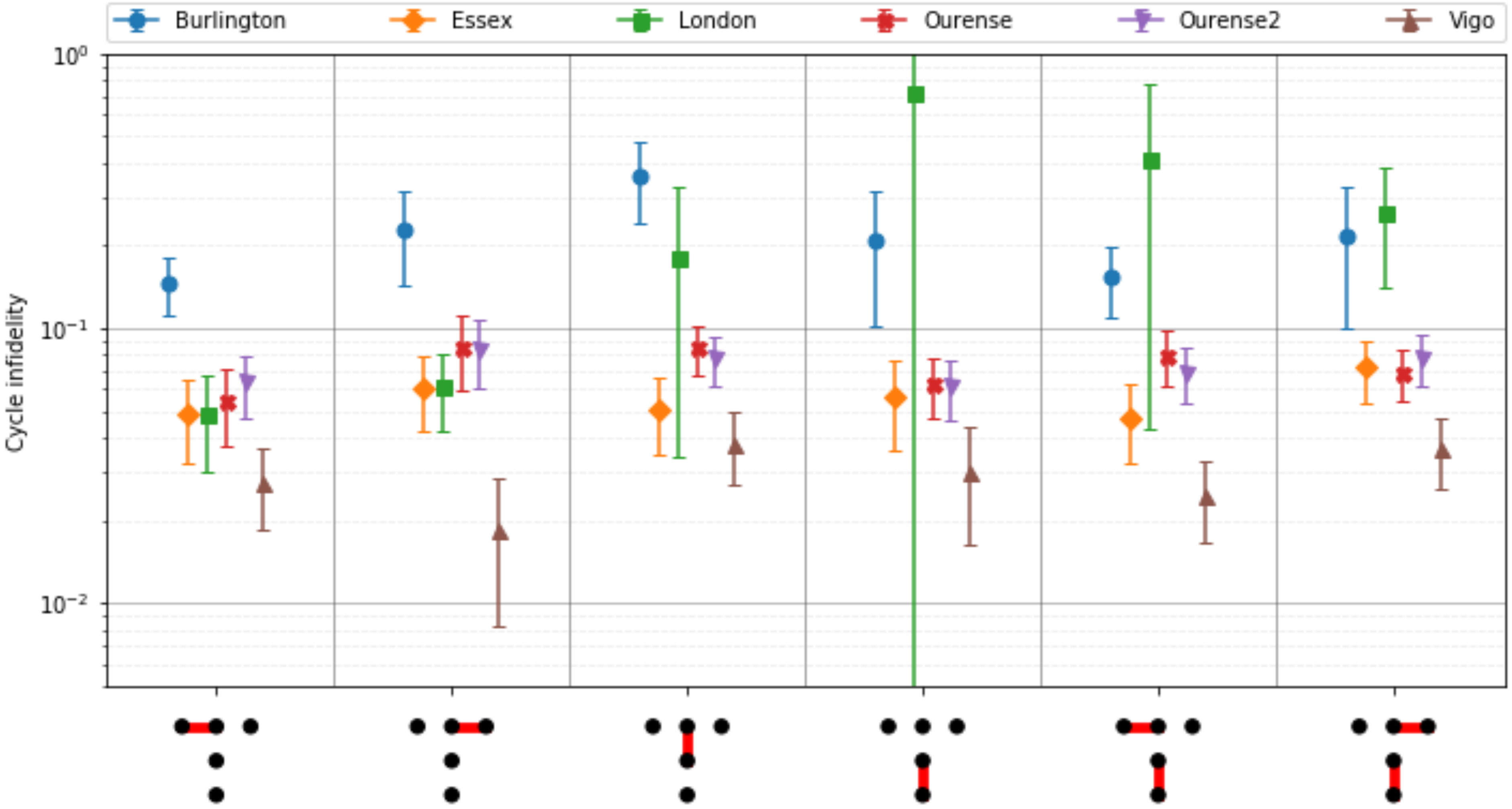
Cycle benchmarking for hard cycle across n-qubits systems

- Cycle benchmarking scales very, very well to arbitrarily large quantum processors
- Estimate fidelity of any combination of operations across n-qubit system
- Bare (interleaved) gates can be any “hard gate round” as in our randomized compiling paper
- Captures cross-talk, unlike simultaneous RB


m times
$|0\rangle$
$|0\rangle$
$|0\rangle$
$\tilde{\mathcal{B}}_{P,1}$
$\tilde{\mathcal{B}}_{P,2}$
$\tilde{\mathcal{B}}_{P,N}$
$\tilde{\mathcal{R}}_{0,1}$
$\tilde{\mathcal{R}}_{0,2}$
$\tilde{\mathcal{R}}_{0,N}$
$\tilde{\mathcal{G}}$
$\tilde{\mathcal{R}}_{i,1}$
$\tilde{\mathcal{R}}_{i,2}$
$\tilde{\mathcal{R}}_{i,N}$
$\tilde{\mathcal{B}}^\dagger_{\mathcal{C}(P),1}$
$\tilde{\mathcal{B}}^\dagger_{\mathcal{C}(P),2}$
$\tilde{\mathcal{B}}^\dagger_{\mathcal{C}(P),N}$

Erhard et al, 2019

Cycle Benchmarking for *each* available Hard Cycle



<u>IBM Q</u>: Process infidelity on all available hard gate rounds on all available 5 qubit chips

<u>Claim 1</u>: From this CB data set we can accurately predict the performance of any application!

<u>Claim 2</u>: The CB approach to benchmarking scales to arbitrarily large quantum computers!

Overview

I. Intro and Background

II. Cycle-level Benchmarking and Cycle Error Reconstruction
- Cycle benchmarking circuits; Pauli infidelity estimation

III. **System-level Benchmarking**
- A performance guarantee for applications
- A *scalable* quantum volume benchmark

Circuit Benchmarking via Cycle Benchmarking of Hard Gate Rounds

- For very broad error models, and assuming the application of twirling to reduce errors to a Pauli channel, then the fidelity of composition is closely approximated by the individual cycle fidelities:

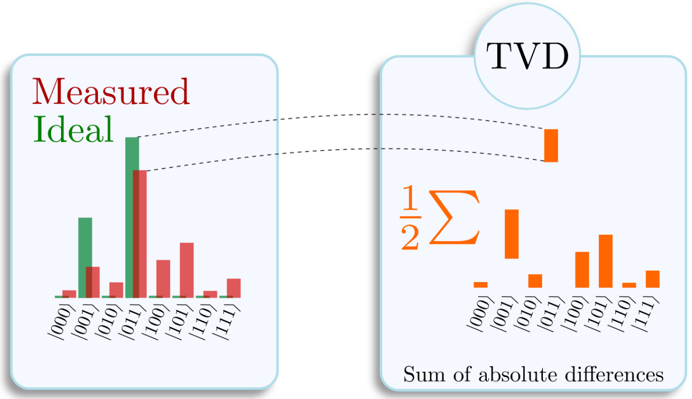
$$\mathcal{C}_L = \Pi_{i=1}^{L} \mathcal{H}_i \mathcal{S}_i \qquad \mathcal{F}_{\text{pro}}(\mathcal{C}_L) \geq \Pi_{i=1}^{L} \mathcal{F}_{\text{pro}}(i) + O(r^2)$$

Carignan-Dugas et al, Quantum 3, 173 (2019)
Emerson et al, forthcoming

- The CB process fidelity data set for the hard gate rounds gives a predictive and scalable system-level benchmark **that predicts performance for any application circuit!**

- We can also bound the TVD for any circuit:

$$\boxed{D(p,q) = \frac{1}{2}\sum_i |p_i - q_i| \leq r(\mathcal{E},\mathcal{I})\frac{(d+1)}{d}}$$



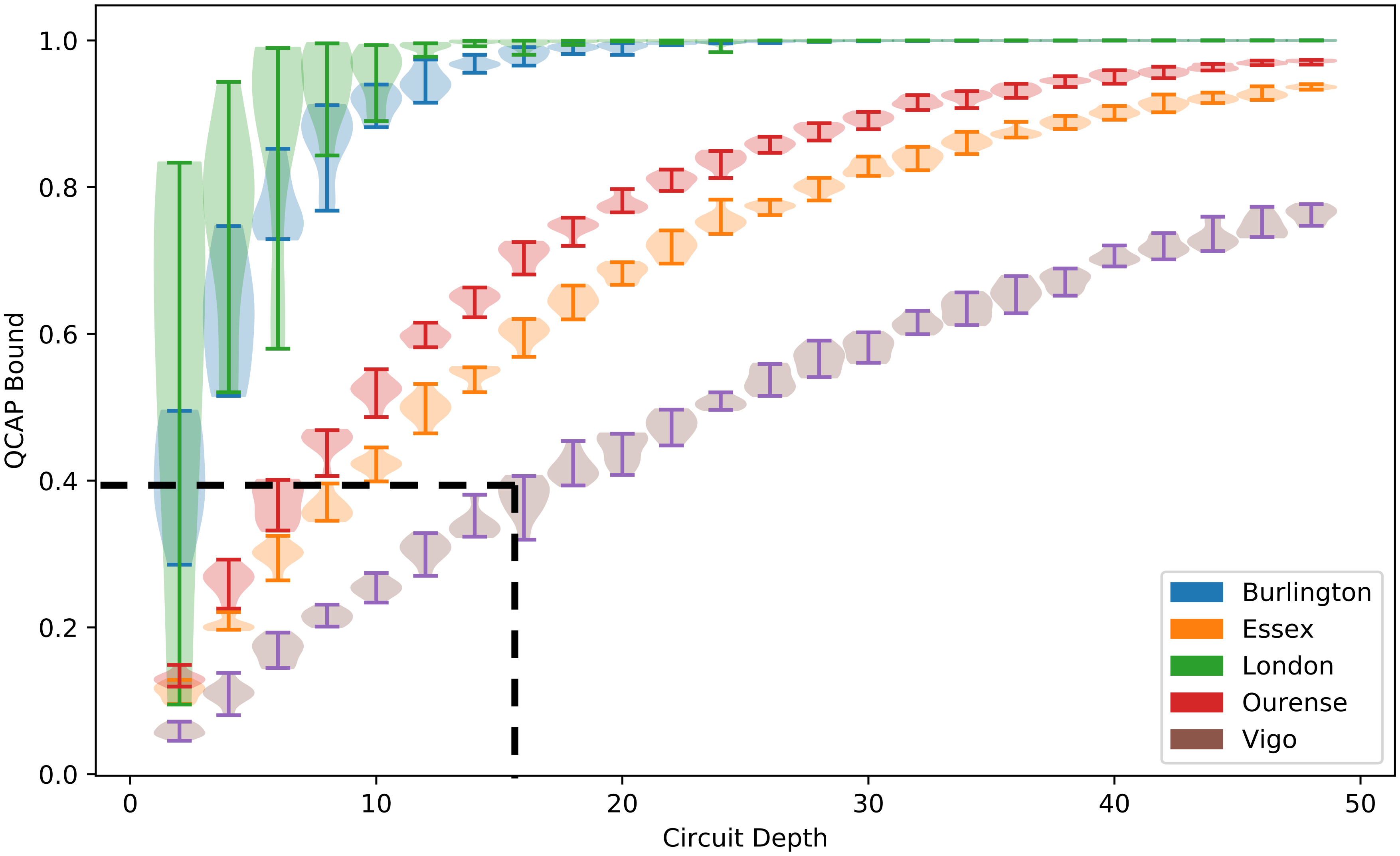
- *This means we can bound the solution accuracy via the TVD for any quantum application!*

Circuit Benchmarking to predict application performance

TVD bound for randomly selected hard gate rounds.

These are the “dressed gate” fidelities relevant to RC circuit performance.

Leads to a natural, **scalable fidelity benchmark & performance bound for any application**



KEYSIGHT TECHNOLOGIES

Circuit Benchmarking for a Scalable Quantum Volume

- Decompose QV circuits into hard and easy rounds: this defines dressed gate rounds
- Apply CB to each dressed hard gate round
- Obtain a **scalable fidelity benchmark** for any application circuit using this (standardized) CB data set

$$\mathcal{F}_{\text{pro}}(\mathcal{C}_L) \geq \Pi_{i=1}^{L} \mathcal{F}_{\text{pro}}(i)$$

- Setting a threshold for this fidelity defines a scalable quantum volume benchmark as a “generic” performance benchmark!

Summary 1/2: The problems with current system-level benchmarks

- Previous system-level randomized benchmarks like Quantum Volume/XEB do not scale
  - Exponential cost in sampling measurement outcomes, predicting ideal outcomes, and/or determining figure of merit “heavy-output” success probability
  - Limited to benchmarking systems with less than ~50 qubits (ie, useless quantum computers)
- Application-based benchmarks, like QED-C, either do not scale and/or have other problems
  - Exponential cost in sampling measurement outcomes, or exponential cost in predicting solutions
  - Or they select overly simplistic circuits to avoid the above problems
    - Easy to **compile to triviality**, so companies **do not need to actually perform most gates**
    - Not representative of eventual use-cases in “utility” regime where we will not know the correct solutions
  - **And open to exponentially expensive** error mitigation and compilation schemes that give misleading system performance estimates: not relevant to future use-cases in the “utility” regime

Summary 2/2: Circuit benchmarking as a fully scalable system-level benchmark

- Cycle benchmarking and Circuit Benchmarking
  - Solves all of the problems with previous system-level benchmarks
  - Works for arbitrarily large Quantum computers
  - Closes the gap between component level randomized benchmarks and application-level performance
  - Predicts the performance of **any and every** application-level benchmark via randomized compiling
  - Bonus: improved application-level performance via randomized compiling
  - Provides robust performance guarantees for applications
  - Can define a scalable quantum volume benchmark

KEYSIGHT
TECHNOLOGIES