Robustly decorrelating errors with mixed quantum gates

Anthony M. Polloreno*
Rigetti Computing, 2919 Seventh Street, Berkeley, CA, 94710-2704 USA

Kevin C. Young
Quantum Performance Laboratory, Sandia National Laboratories, Livermore, CA

(Dated: November 26, 2019)

Coherent errors in quantum operations are ubiquitous. Whether arising from spurious environmental couplings or errors in control fields, such errors can accumulate rapidly and degrade the performance of a quantum circuit significantly more than an average gate fidelity may indicate. As Hastings [1] and Campbell [2] have recently shown, by replacing the deterministic implementation of a quantum gate with a randomized ensemble of implementations, on can dramatically suppress coherent errors. Our work begins by reformulating the results of Hastings and Campbell as a quantum optimal control problem. We then discuss a family of convex programs designed to improve the performance, implementability, and robustness of the resulting mixed quantum gates. Finally, we implement these mixed quantum gates on a superconducting qubit and discuss randomized benchmarking results consistent with a marked reduction in the coherent error.

[1] M. B. Hastings, *Quantum Information & Computation* 17, 488 (2017).
[2] E. Campbell, *Physical Review A* 95, 042306 (2017).

I. INTRODUCTION

The ultimate impact of a gate error on the performance of a quantum circuit depends strongly on both the magnitude and the nature of the error. Systematic, or *coherent*, errors can arise from poorly calibrated controls or approximate gate compilations that induce repeatable, undesired unitary errors on the state of a quantum information processor. Errors of this type are correlated in time and may add up constructively or destructively, depending on details of the circuit in which they appear. This can make it difficult to construct tight analytic bounds on circuit performance [3], and numerical studies are often limited by the high computational cost of modeling coherent errors. Contrast this against random, or *stochastic*, errors which often result from high-frequency noise in the controls or the environment. Systems with stochastic errors can usually be accurately modeled by defining a rate of various discrete errors in the system, such as a bit flips or phase flips. These errors are significantly easier to simulate on a classical computer, and their impact on quantum circuits is much easier to estimate [3].

Campbell [2, 4] and Hastings [1] have developed a technique for suppressing coherent noise by replacing deterministic gate implementations with a *mixed quantum gate* (MQG) consisting of a randomly sampled ensemble of implementations. They focus on errors in gate compilation, such as those that arise from the Solovey-Kitaev algorithm, for which any lingering approximation errors are generally coherent, even if the underlying native gates are perfect. Different approximate gate compilations of the same target unitary will almost certainly result in different unitary errors. So by selecting from these various compilations at random, the resulting quantum channel becomes a mixture of unitaries [5], which can have significantly less coherent error than any single compilation on its own. A very simple example of this reduction in coherent error by mixed quantum gates is illustrated in Fig. 1.

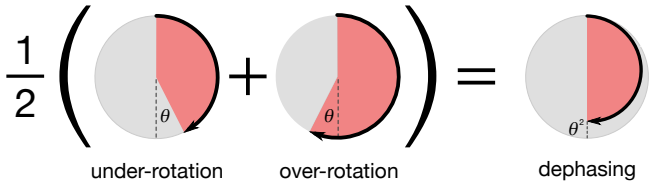

FIG. 1. **Simple example of a mixed quantum gate:** Using optimal control, two implementations of a Z_π gate are designed to have equal and opposite coherent errors (if one implementation over-rotates by a small angle θ, then the other *under*-rotates by θ). Each time the gate is used, one of these two implementations is chosen at random. The resulting effective quantum gate is equivalent to a perfect implementation of the gate followed by dephasing with associated probability $\mathcal{O}\left(\theta^2\right)$.

Randomized protocols have a long history of outperforming their deterministic counterparts[6, 7]. In circuit models, Pauli Frame Randomization [8, 9] has been shown to reduce coherent errors by introducing randomness in a pre-compilation step. While Pauli Frame Randomization shares the some favorable properties with MQGs, we will see that MQGs additionally admit a hardware-efficient implementation and numerous useful modifications when they are posed as a numerical optimization problem.

In this article, we extend the work of Campbell and Hastings to numerically optimized quantum gates and show that the advantages of their approach can be made robust to drift and model uncertainty. We provide an

* Email: ampolloreno@gmail.com; Current address: University of Colorado, Boulder, CO

explicit series of convex programs able to efficiently compute MQGs with a variety of useful properties. We demonstrate our results on a superconducting testbed device at Rigetti Computing. In a simple experiment based on randomized benchmarking circuits, MQGs demonstrate a marked improvement in error rates and a reduced variance in circuit outcome probabilities, consistent with a significant reduction in the coherence of the error [10]. We further apply our methods in simulation, constructing both single- and two-qubit mixed unitary controls that are robust to drift and uncertainty in the control parameters.

II. PRELIMINARIES

A. Representing Errors in Quantum Gates

In order to implement a desired unitary gate, G, on a quantum device, one generally applies a carefully tuned sequence of classical control fields. But fluctuations in the environment or imperfections in these controls can cause the state of the qubits to change in a way that is different from what was intended, *i.e.*, there are errors in the gate. If the device is fairly stable with time[11] and context[12], then we can accurately model the gate action using a completely positive, trace-preserving (CPTP) map, $\tilde{\mathsf{G}}$ acting on the Hilbert space of the target qubits. This map can always be written as $\tilde{\mathsf{G}} = \mathsf{E} \circ \mathsf{G}$, where $\mathsf{E} = \tilde{\mathsf{G}} \circ \mathsf{G}^{-1}$ is the *error map*, which is itself CPTP because G is unitary.

CPTP maps possess a number of useful representations, including Kraus operators[13], Choi matrices[14], and Jamiolkowsi states[15]. But for the purposes of this article, the *process matrix* representation will be particularly convenient[16], and we shall denote the process matrix associated with a given CPTP map G with its corresponding calligraphic character, $\mathcal{G}$. For a d-dimensional quantum state, the process matrix is a $d^2 \times d^2$ matrix. A key feature of process matrices is that they are composable and act through the usual matrix multiplication on the vectorized quantum state:

$$\mathsf{vec}\left(\tilde{\mathsf{G}}(\rho)\right) = \tilde{\mathcal{G}} \cdot \mathsf{vec}\left(\rho\right) \tag{1}$$

$$= \mathcal{E} \cdot \mathcal{G} \cdot \mathsf{vec}\left(\rho\right) \tag{2}$$

The vec operation is typically performed using a basis of matrix units, for which $\mathsf{vec}\left(\rho\right)$ would be the column vector obtained by stacking the columns of ρ. In this work, however, we shall use a basis of Pauli matrices, defining $P = \{I, \sigma_x, \sigma_y, \sigma_z\}^{\otimes n}$ as the collection of all 4^n n-qubit Pauli operators (including the identity). In this basis,

$$\mathsf{vec}\left(\rho\right)_i = \langle P_i \rangle = \mathrm{Tr}\left(P_i\ \rho\right), \tag{3}$$

and

$$(\tilde{\mathcal{G}})_{ij} = \frac{1}{d}\mathrm{Tr}\left(P_i\ \tilde{\mathsf{G}}(P_j)\right). \tag{4}$$

Process matrices written in this Pauli basis are often referred to as *Pauli transfer matrices* (PTMs) [17]. Error maps represented in this basis take the particularly nice form,

$$\mathcal{E} = \left(\begin{array}{c|c} 1 & \vec{0}^T \\ \hline \vec{m} & R \end{array}\right) \tag{5}$$

The top row of all trace-preserving (TP) error maps is fixed to $\{1, 0, 0, 0, \cdots\}$ and the remainder of the first column, $\vec{m}$, describes any deviations from unitality, as might arise from amplitude damping [18]. If the error map is unitary, then the error is called *coherent* and the unital submatrix R is a rotation matrix. Importantly, if R is diagonal, then the error is Pauli-stochastic, with each diagonal entry corresponding to the probability that its associated Pauli error occurs in each application of the gate.

In general an error will neither be fully coherent nor Pauli-stochastic and so it is useful to define what we mean by the coherence of arbitrary maps[19]. One way to measure this is with a quantity called the unitarity[20]:

$$u(\mathsf{E}) = \frac{d}{d-1}\int d\psi ||\mathbf{n}[\mathsf{E}(\psi)] - \mathbf{n}[\mathsf{E}(\mathcal{I}_d/2)]||^2 \tag{6}$$

where $\mathbf{n}$ is the length of the Bloch vector, and the integral is over all d-dimensional pure states using the Haar measure. Yet another way of quantitavely describing how unitary a map is with protocols that assess how errors accumulate on different input states[21]. In this work, we will only discuss the relative unitarity of channels, and therefore are content if the unitarity of our error map decreases by any of these metrics.

In what follows it will be useful to define the *error generator*, $\mathcal{L}$, associated with a faulty gate:

$$\mathcal{E} = \exp\left(\mathcal{L}\right) \tag{7}$$

$$= \mathcal{I}_d + \mathcal{L} + \frac{1}{2}\mathcal{L}^2 + \mathcal{O}\left(\mathcal{L}^3\right). \tag{8}$$

If an implemented gate is relatively close to its target, then the error generator will be small under any of the usual matrix norms, and the Taylor expansion above may reliably be truncated at first or second order.

B. Mixed Quantum Gates

Suppose that we have access to an ensemble of distinct implementations, $\{\tilde{\mathsf{G}}_i\}$, of a target gate, G. Each time the gate is to be applied to the system, we randomly select an implementation from this ensemble such that the probability of drawing $\tilde{\mathsf{G}}_i$ is w_i (and we ensure that $\sum_i w_i = 1$). This procedure is operationally indistinguishable from always applying the effective channel,

$$\tilde{\mathsf{G}}_{\text{eff}} = \sum_i w_i \tilde{\mathsf{G}}_i = \left(\sum_i w_i \mathsf{E}_i\right) \circ \mathsf{G} \tag{9}$$

We call such randomized quantum operations *mixed quantum gates* or MQGs. Error metrics for these MQGs can be computed in terms of their effective error map,

$$\mathsf{E}_{\text{eff}} = \sum_i w_i \mathsf{E}_i. \tag{10}$$

Two important error metrics are the average gate infidelity (AGI), $\epsilon_{\mathcal{F}}$[22], and the diamond distance to the target (or simply, the *diamond distance*), $\epsilon_\diamond$ [23], defined as:

$$\epsilon_{\mathcal{F}}(\mathsf{E}) = \frac{d^2 - \mathrm{Tr}\,\mathcal{E}}{d^2 + d}, \tag{11}$$

$$\epsilon_\diamond(\mathsf{E}) = \frac{1}{2} \sup_\rho ||(\mathsf{I}_d \otimes \mathsf{I}_d)(\rho) - (\mathsf{E} \otimes \mathsf{I}_d)(\rho)||_1, \tag{12}$$

where $d = 2^n$, with n being the number of qubits, and I_d is the d-dimensional identity operator. In Eq. (11), we have written the AGI for an error map, E, in terms of its associated Pauli transfer matrix, $\mathcal{E}$. If the error channel is purely stochastic, then $\epsilon_\diamond(\mathsf{E}) = \epsilon_{\mathcal{F}}(\mathsf{E})$, but if the error channel has a unitary component, then the diamond distance will generically be larger than the average gate infidelity [24]. The diamond distance is subaddative [23] under gate composition, so is particularly useful for constructing bounds on quantum circuit performance: the total variation distance between the outcome probabilities of a faulty and perfect quantum circuit is less than or equal to the sum of the diamond distances for gates that compose the circuit [25].

Because $\epsilon_{\mathcal{F}}$ is linear in the error map, we have:

$$\epsilon_{\mathcal{F}}(\mathsf{E}_{\text{eff}}) = \sum_i w_i\, \epsilon_{\mathcal{F}}(\mathsf{E}_i). \tag{13}$$

That is, the AGI of the effective error channel is simply the weighted average of the component AGIs, so MQGs provide little benefit for reducing the AGI. However, the diamond distance is a nonlinear function of error map. As we show in the appendix (VII A),

$$\epsilon_\diamond(\mathsf{E}_{\text{eff}}) \leq \sum_i w_i\, \epsilon_\diamond(\mathsf{E}_i). \tag{14}$$

So by mixing various implementations, each with a different error channel, the resulting channel can have a diamond distance error that is less than the diamond distance error of any of the component implementations.

Campbell [2] and Hastings [26] independently considered these mixed channels using gates constructed with the Solovey-Kitaev algorithm, for which many approximate gate compilations are possible. Campbell showed that, if the error generators of some ensemble of gate compilations form a convex set containing the origin, then one can construct a MQG with quadratically suppressed diamond distance to the target. Explicitly, the weights are chosen such that the error generator is canceled to first order. Using (10) and (8), the effective error map for a MQG in terms of the component error generators is,

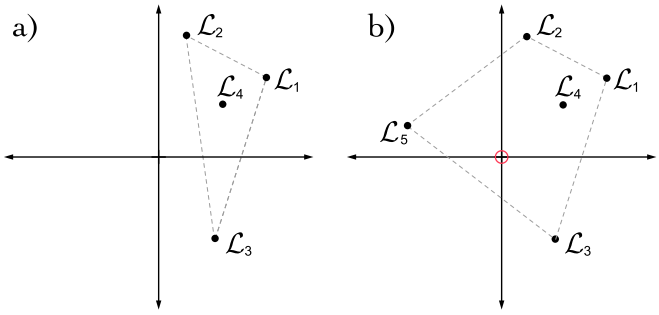

FIG. 2. **Geometric condition for existence:** A target unitary gate can be implemented a number of ways, each associated with a different error generator, $\mathcal{L}_i$. These generators lie in a vector space, which we illustrate as two-dimensional here. a) Four error generators. The origin is not contained in their convex hull, so there are no generator-exact MQGs. b) After including an additional control solution, the convex hull grows to contain the origin, and so a generator-exact MQG exists.

$$\mathcal{E}_{\text{eff}} \simeq \sum_i w_i \left(\mathcal{I}_d + \mathcal{L}_i + \frac{1}{2}\mathcal{L}_i^2\right) \tag{15}$$

These error generators are linear operators and so are elements of a vector space. If, as Campbell required, the origin lies in their convex hull (see Fig. 2), then there exists a choice for the weights, w_i^*, such that $\sum_i w^* = 1$ and,

$$\sum_i w_i^* \mathcal{L}_i = 0 \tag{16}$$

When this condition is satisfied, we call the resulting operation a *generator-exact* MQG.

If the diamond distance error rates of the component gates are bounded by α, then Campbell shows that this first-order cancellation of the error generators can ensure that the diamond distance error rate of the MQG is bounded by α^2. For gate compilation errors, the error channels are dominated by unitary approximation error, so the error generators are Hamiltonians. Hamiltonian generators can have positive or negative coefficients, so it is possible that the origin may lie in their convex hull. If, however, the error map contains stochastic components, then the condition of Eq. (16) might be impossible to satisfy. Such errors always have strictly positive probabilities, so the origin can *never* lie in their convex hull. This can be rectified by restricting the sum in Eq. (16) to only the Hamiltonian component of the generators, or by replacing the exact condition with a minimization, as we discuss in Sec. III.

While generator-exact MQGs can provide a guaranteed suppression of the diamond distance, their effective error channels are unlikely to be a purely stochastic. Second- and higher-order terms in (15) can easily contribute to

lingering coherent errors that impact the efficient simulability of the channel. In order to definitively eliminate these coherent errors, we could instead seek weights, w_i^P, that annihilate the off-diagonal entries of the effective error map. The resulting diagonal error map will be Pauli-stochastic. For a single qubit gate, it takes the form,

$$\mathcal{E}_{\text{eff}} = \sum_i w_i^P \mathcal{E}_i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & p_x & 0 & 0 \\ 0 & 0 & p_y & 0 \\ 0 & 0 & 0 & p_z \end{pmatrix}, \tag{17}$$

where p_x, p_y and p_z are the rates of Pauli X, Y, and Z errors, respectively. We refer to such channels as *Pauli-exact* MQGs, and the same geometric argument we used in constructing generator-exact MQGs is again applicable here. The off-diagonal elements of the error map form a vector space, so we need only check that the origin is in the convex hull of the vectors of off-diagonal elements for each of the component operations, $\mathcal{E}_i$. If the error map contains non-unital components or correlated Pauli-stochastic errors (such as $X+Y$), then this strict geometric condition will not hold and no Pauli-exact MQG will exist. However, MQGs that approximate the Pauli-exact condition can be found via the convex programing techniques discussed in Sec. III A 2.

III. CONSTRUCTING MIXED QUANTUM GATES

We now present a methodology for constructing mixed quantum channels, formalizing the intuitive approach discussed above with a series of explicit convex programs. As mentioned, our method requires two steps. The first is a control synthesis step, in which we construct an ensemble of gate implementations. Campbell and Hastings draw their ensembles from various different gate compilations, but for this work we use utilize quantum optimal control theory. Many standard control-generation algorithms, such as GRAPE [27], take a random initial guess for the control and iteratively improve it to yield a gate that well-approximates a target. By seeding such algorithms with many initial guesses, one can quickly construct a large ensemble of approximate quantum gates, each with a different error channel. We discuss this approach in some detail in Sec. IV B.

The remainder of this section will assume such a capability for generating a large ensemble of gate implementations, and will focus instead on the second step: constructing a distribution over that ensemble that reduces the coherence of the effective error channel. We will begin by discussing the two broad optimization targets introduced above: i) generator-exact MQGs and ii) Pauli-exact MQGs. We then propose a set of secondary targets that can improve the performance by i) incorporating robustness to drift, ii) targeting low-error-rate solutions, and iii) reducing the number of native unitaries that contribute to the effective channel.

A. Convex Programs for Constructing Mixed Quantum Gates

1. Generator-exact MQGs

A compelling reason to construct an MQG is to improve the worst-case performance of a quantum gate. The diamond distance bounds this worst-case performance, and so one might assume minimizing it would be a natural optimization target:

$$\underset{w_i \geq 0, \sum_i w_i = 1}{\textbf{minimize}}: \quad \epsilon_\diamond \left(\sum_i w_i \mathsf{E}_i \right) \tag{18}$$

The diamond distance, however, is a non-linear function that in general requires its own convex optimization to compute[23]. This can dramatically slow down iterative optimizers, and so directly minimizing the diamond distance is a computationally burdensome optimization target. However, for quantum computing purposes the error rates are typically quite small, so we can consider a linearized problem, minimizing the diamond distance at first-order in the effective error generator. As discussed around Eq. (16), if the origin lies in in the convex hull of the generators, $\mathbf{0} \in \mathsf{conv}(\{\mathcal{L}_i\})$, then we can construct a generator-exact MQG. In [2] Campbell presents an iterative algorithm that, given an oracle able to produce approximate unitary operations, will continually generate controls until the geometric constraint is satisfied and then identify the optimal weighting.

As discussed in the previous section, the geometric condition illustrated in Fig. 2 may not be satisfied, due perhaps to stochastic errors in the generators or possibly a situation in which one has access only to m distinct, precomputed gate implementations. In this case, we may simply wish to to identify the MQG with minimal error. A heuristic that is efficiently computable is to minimize the Frobenius norm of the first-order effective error generator:

$$\underset{w_i \geq 0, \sum_i w_i = 1}{\textbf{minimize}}: \quad \left|\left| \sum_i w_i \mathcal{L}_i \right|\right|_F \tag{19}$$

This optimization problem is equivalent to:

$$\underset{w_i \geq 0, |w|_1 = 1}{\textbf{minimize}}: \; ||\mathbf{L} \cdot \vec{w}||_2 \tag{20}$$

Where $\mathbf{L} = (\mathsf{col}(\mathcal{L}_1)\, \mathsf{col}(\mathcal{L}_2)\, \cdots)$ is the $d^2 \times m$-dimensional matrix of column-stacked error generators and $||\cdot||_2$ is the ℓ_2 norm.

The constraints on these optimizations are required to ensure that the weights, w_i, form a proper probability distribution. Linearly constrained minimization problems with quadratic cost functions are convex and have been proven to be efficiently solvable by, e.g. the ellipsoid method [28, 29]. Many existing convex solver software packages are available that can solve these problems efficiently in practice[30, 31].

2. Pauli-exact MQGs

As quantum devices grow in size, it is increasingly expensive to simulate their dynamics. Coherent errors impose a particular burden on classical simulators, as the entire quantum state must be tracked continually. Existing work generally approximates channels as Pauli-stochastic channels, and assesses the quality of these approximations[32, 33]. Pauli Frame Randomization adds additional gates into circuits leaving the computation unchanged but randomizing the noise in such a way that it becomes a Pauli-stochastic channel. MQGs offer another path forward. Pauli-exact MQGs can be constructed with vanishing coherent error and without adding any additional gates or compilation time. Building MQGs with diminished coherent error enables the use of much more efficient Pauli-stochastic simulators that are able to model significantly more qubits than their vector-state counterparts.

In particular, the well-known stabilizer formalism [34] for quantum simluation gives a polynomial-time algorithm to compute expectation values of circuits consisting of only Clifford operations. These circuits are useful because they give examples of quantum circuits whose performance can be efficiently assessed using classical simulation. This has motivated significant development of these techniques, including extensions to simulating stabilizer circuits with mixed state inputs[35] and linear combinations of stabilizer circuits. [36, 37] An interesting property of such simulation techniques is that the cost of the classical simulation is proportional to how non-Clifford the circuit is. Therefore Pauli-exact MQGs have a clear benefit - if the error channels of a circuit can be made to look more like Pauli-stochastic errors, they will be more amenable to efficient classical simulation.

Like generator-exact MQGs, the optimal weights can be found efficiently by a convex optimization. As above, we assume that we have access to m distinct gate implementations, each of which is associated with its own error matrix, $\mathcal{E}_i$. We wish to construct an MQG whose effective error is Pauli-stochastic and is therefore diagonal. This can be done with the following simple minimization:

$$\underset{w_i\geq 0,\sum_i w_1=1}{\textbf{minimize}} : ||\sum_i w_i\mathcal{E}_i - \mathsf{diag}(\sum_i w_i\mathcal{E}_i)||_2 \tag{21}$$

Here $(\mathsf{diag}(\mathcal{A}))_{ij} = \mathcal{A}_{ij}\delta_{ij}$ sets the off-diagonal elements of a matrix to zero. This can be more transparently written as a convex program by constructing the $(d^2 - d) \times m$-dimensional matrix, $\mathbf{E}$, whose i^{th} column is a vector composed of all of the off-diagonal entries of the error generators $\mathcal{E}_i$. [1] Eq. (21) is then equivalent to

$$\underset{w_i\geq 0,|w|_1=1}{\textbf{minimize}} : ||\mathbf{E}\cdot\vec{w}||_2 \tag{22}$$

[1] Exactly *how* the off-diagonal entries are vectorized is unimportant, so long as it is consistent across error matrices.

The similarity with Eq. 20 is clear and the same methods can be used to solve both. If the solver routine finds a solution for which the cost function is equal to 0, then the resulting error channel will be *exactly* a Pauli channel.

B. Secondary Objectives

Often, the cost of generating distinct gate implementations is quite small, so large ensembles of them can be computed rather quickly. It is often the case that one will have available many more distinct gate implementations than there are parameters in the relevant vector space (see Fig. 2). This can lead to the the minimization problems discussed in the previous section being massively under-constrained, yielding a large continuous family of exact solutions. Secondary objectives provide a means for choosing among this family of exact MQGs those with increased performance in a desired area. In this section we present explicit convex programs for such secondary objectives, including adding robustness to drift or model uncertainty, reducing the effective gate error, and minimizing the number of constituent gates utilized by the MQG. We choose to take generator-exact MQGs as our starting point, though the following sections apply equally well to Pauli-exact MQGs with only trivial modifications.

1. Robustness to drift and model uncertainty

Mixed quantum gates can offer significant performance improvements over bare unitary gates, but their construction requires a good knowledge of the errors experienced by the constituent gates. Often, however, the gates are not perfectly well-characterized. This could be due to simple uncertainty about the parameters of the model used to generate the control solutions, or the system could experience some degree of drift [11]. In either case, the performance of the MQG will be negatively impacted. We can describe the effect of this uncertainty by first writing the gate explicitly as a function of a vector of model parameters, $\vec{\lambda}$, and their nominal values, $\vec{\lambda}_0$. These model parameters could include magnetic field strengths, laser or microwave intensities, coupling strengths, etc. If the parameters are close to their nominal values, then we can write a gate's error generator as a Taylor expansion in the deviation vector, $\vec{\delta} = \vec{\lambda} - \vec{\lambda}_0$:

$$\mathcal{L}_i(\vec{\delta}) = \mathcal{L}_i(\mathbf{0}) + \sum_k \delta_k\, \mathcal{L}_{i,k}(\mathbf{0}) + \mathcal{O}\left(\delta^2\right). \tag{23}$$

In the above expression we have used the comma derivative,

$$\mathcal{L}_{i,k}(\vec{\delta}_0) = \frac{\partial}{\partial\delta_k}\mathcal{L}_i(\vec{\delta})|_{\vec{\delta}=\delta_0}. \tag{24}$$

Combining Eq. (23) with Eq. (8) and taking both Taylor expansions to first order, we have:

$$\mathcal{E}_{\text{eff}} = \mathcal{I}_d + \sum_i w_i \left(\mathcal{L}_i(\mathbf{0}) + \sum_k \delta_k \, \mathcal{L}_{i,k}(\mathbf{0}) \right) + \mathcal{O}\left(\mathcal{L}_i^2 + \mathcal{L}_i \delta + \delta^2\right). \tag{25}$$

We would like to choose the weights $\vec{w}$ so that the effective error generator vanishes to first order. We can cast this problem as a convex optimization problem, much as we did in Sec. III A 1 for synthesizing generator-exact MQGs. Following Eq. (20), we have,

$$\underset{w_i \geq 0, |w|_1 = 1}{\textbf{minimize}} : ||\mathbf{L} \cdot \vec{w}||_2 + \sum_k ||\mathbf{L}_{,k} \cdot \vec{w}||_2 \tag{26}$$

Where $\mathbf{L}_{,k} = (\mathsf{col}(\mathcal{L}_{1,k}(\mathbf{0}))\ \mathsf{col}(\mathcal{L}_{2,k}(\mathbf{0}))\ \cdots)$ is the $d^2 \times m$-dimensional matrix of column-stacked derivatives of the error generators with respect to the deviation δ_k.

We can additionally generalize the geometric criterion for guaranteeing the existence of generator-exact MQGs. To construct a robust, generator-exact MQG, there must exist a set of weights, $\{w_i\}$, so that $\sum_i w_i \mathcal{L}_i = \sum_i w_i \mathcal{L}_{i,k} = 0$. This is equivalent to demanding the origin lie in the convex hull of the generators augmented with their derivatives:

$$\mathbf{0} \in \mathsf{conv}\left(\left\{ \mathcal{L}_i \bigoplus_k \mathcal{L}_{i,k} \right\}_i \right). \tag{27}$$

We illustrate this in Fig. 3.

We call a gate constructed in this way a *robust* generator-exact MQG. These techniques can be easily adapted to construct robust Pauli-exact MQGs, where vectors of the off-diagonal elements of the error maps take the role of the error generators used above.

2. *Improving the average gate fidelity*

While the geometric constraint is sufficient for suppressing the diamond norm to first order relative to the *worst* controls in the collection, it does not preferentially select the best controls possible. As an example, consider four erroneous implementations of a Pauli Z gate: $Z_{-2\theta}$, $Z_{-\theta}$, Z_{θ}, and $Z_{2\theta}$, where the subscript indicates the magnitude of the Z-rotation error. An MQG consisting of equally weighted $Z_{-2\theta}$ and $Z_{2\theta}$ is generator-exact, as is $Z_{-\theta}$ and Z_{θ}. But the error rate[2] of the first is $4\theta^2$, while the second achieves an error rate of θ^2.

To incentive the inclusion of controls with smaller error, we may instead minimize the average gate infidelity of the MQG, subject to the condition that any solution we find be generator-exact:

[2] The diamond distance is equal to the AGI here.

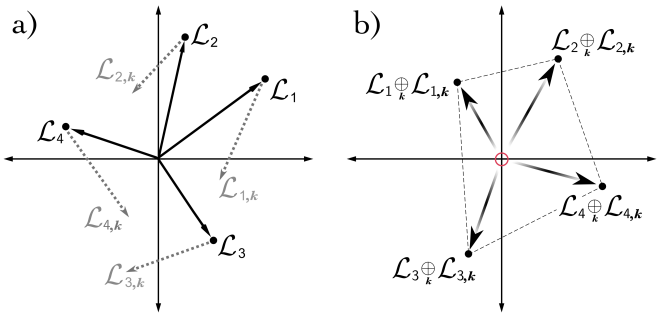

FIG. 3. **Geometric condition for robustness:** A target unitary gate can be implemented a number of ways, each with a different effective error generator, $\mathcal{L}_i$. a) The error generators are shown as elements of a vector space (solid, black arrows). Also shown are their derivatives $\mathcal{L}_{i,k}$ (dashed, grey arrows) with respect to a model parameter, δ_k. As this parameter drifts, the generators may no longer cancel. b) To construct a robust, generator-exact MQG, there must exist a set of weights, $\{w_i\}$, so that $\sum_i w_i \mathcal{L}_i = \sum_i w_i \mathcal{L}_{i,k} = 0$. This is equivalent to demanding that the origin lie in the convex hull $\mathsf{conv}(\{\mathcal{L}_i \bigoplus_k \mathcal{L}_{i,k}\}_i)$. The gradient arrows indicate that this space is higher-dimensional.

$$\begin{aligned} &\underset{w_i \geq 0, |w|_1 = 1}{\textbf{minimize}} : \sum_i w_i \, \epsilon_{\mathcal{F}}(\mathcal{E}_i) \\ &\textbf{subject to} : ||\mathbf{L} \cdot \vec{w}||_2 = 0 \end{aligned} \tag{28}$$

If we do not expect to have a space of generator-exact solutions available, then the constraint above will never be satisfied and the minimization will fail. This could occur if there are simply not enough controls to satisfy the generator-exact criterion, or if the error generators have stochastic components. In such a case, we can choose a weighting parameter, η, to explicitly balance the two competing objectives:

$$\underset{w_i \geq 0, |w|_1 = 1}{\textbf{minimize}} : ||\mathbf{L} \cdot \vec{w}||_2 + \eta \sum_i w_i \, \epsilon_{\mathcal{F}}(\mathcal{E}_i) \tag{29}$$

Both optimizations are convex and efficiently solvable[38].

3. *Sparsity Constraints*

As a practical consideration, we would also like to regularize our objective function to promote sparse weightings. Control electronics often have a limited amount of waveform memory, and thus it is important to be able to construct MQGs as few constituent gates as possible. In many machine learning contexts, lasso regularization [39] can be used to enforce sparsity in solutions. Rather than penalizing the ℓ_0 norm of the solution, which is not convex, lasso regularization penalizes the ℓ_1, which is the convex relaxation of the ℓ_0 norm. However this is insufficient for our purposes, as we require w to be a valid

probability distribution, which constrains the ℓ_1 norm to be equal to 1. Conveniently, the problem of enforcing sparsity in such situations has been considered in [40] and can be expressed via another convex program that extends Eq. (20):

$$\begin{aligned} &\textbf{minimize:} \\ &\underset{m\in[1..N]}{} \\ &\quad\textbf{minimize:} \quad ||\mathbf{L}\cdot\vec{w}||_2 + t \\ &\underset{w_i\geq 0,|w|_1=1,t\geq 0}{} \\ &\quad\textbf{subject to:}\ w_m > \frac{\lambda}{t}. \end{aligned} \tag{30}$$

Here, $\lambda > 0$ is a tunable parameter that can be used to control the degree of sparsity in the solution. Increasing λ will increase the sparsity of the solution, but will generally decrease the likelihood that the solution will be generator-exact.

In the following sections we discuss both experimental and numerical implementations of MQGs, leveraging each of these secondary optimizations.

IV. RESULTS

A. Experimental implementation

We implemented our methods using the 19Q-Acorn superconducting transmon processor at Rigetti Computing. Comprehensive characterization of this device can be found in [41]. While this device consisted of 20 qubits with fixed couplings, we used only qubit #8 for our demonstration. Controls were derived from a single, 10-sample, 50ns Gaussian $X_{\pi/2}$ pulse calibrated via Rabi oscillation. Four intentionally miscalibrated Gaussian pulses, labeled Pulse1 through Pulse4, were derived by scaling the amplitude of the calibrated $X_{\pi/2}$ pulse by a factor $S \in \{1.064, 1.039, 0.937, 0.912\}$, respectively. We model their effect on the qubit state as perfectly unitary rotations, $X_{A\pi/2} = \exp(-iA\pi/2\sigma_x)$. These gates are then used as in Sec. III A 2 to construct a Pauli-exact MQG, which samples from the miscalibrated waveforms with probabilities $P \in \{0.307, 0.283, 0.211, 0.199\}$ respectively. Using these probabilities and pulse amplitudes we have plotted the corresponding Pauli Transfer Matrices in Figure **??**.

We characterize the performance of the MQG using a series of randomized benchmarking (RB) experiments[42]: one for each of the miscalibrated pulses, one using the calibrated pulse, and using the MQG. In each case, the Clifford operations were decomposed into $X_{\pi/2}$ gates, which were implemented directly using the relevant waveforms, and $Y_{\pi/2}$ gates, which were implemented by phase-shifting the relevant waveform $\pi/2$ radians. All circuits were precompiled on the host computer and sent to be executed to the control electronics, rather than using the FPGA or lower-level firmware to sample at run-time.

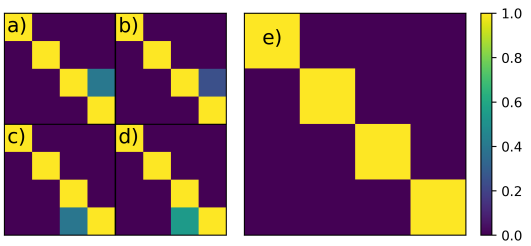

FIG. 4. Pauli Transfer Matrices for a) Pulse1 b) Pulse2 c) Pulse3 d) Pulse4 e) MQG. The four constituent pulses can be seen to have coherent error from the off-diagonal components of their Pauli Transfer Matrices, while the Pauli-exact MQG has been constructed to have minimal coherent error and therefore has predominantly diagonal elements.

An RB experiment consisted of sampling and running 10 random Clifford sequences at each of lengths $L \in \{2, 4, 8, 16, 32, 64\}$. Each sequence was repeated 1000 times. We emphasize that, when benchmarking the MQG, each sequence repetition used a new, randomly sampled sequence of waveforms consistent with its definition. Results of the standard RB analysis are discussed in Table. (IV A). The MQG is seen to perform nearly as well as the calibrated pulse, and better than any of the constituent pulses individually.

Pulse name	Rotation error	RB error rate
Pulse1	6.4%	1.1%
Pulse2	3.9%	0.9%
Pulse3	−6.3%	1.1%
Pulse4	−8.8%	1.5%
Calibrated	·	0.7%
MQG	·	0.8%

TABLE I. **Randomized benchmarking results:** The MQG outperforms all of the individual miscalibrated gates of which it is composed, achieving an RB number nearly as low as the carefully calibrated pulse.

By minimizing the off-diagonal elements of the process matrix, we expect the resulting MQG to display suppressed coherent error. To see that this is in fact the case, we can inspect the variance of the *RB* survival probabilities for the MQG relative to the miscalibrated pulses. As discussed in [10], coherent errors will tend to broaden the distribution of RB survival probabilities over sequences at each length, generally manifesting as a long-tailed gamma distribution. Stochastic noise, such as depolarizing noise, will yield comparatively narrow, Gaussian-distributed success probabilities. In Fig. 5, we plot the experimentally observed distribution of survival probabilities at sequence length 64 for each of the benchmarked gate sets. We see that the intentionally miscalibrated controls in our RB experiment have long tails, consistent with coherent noise, while the calibrated and

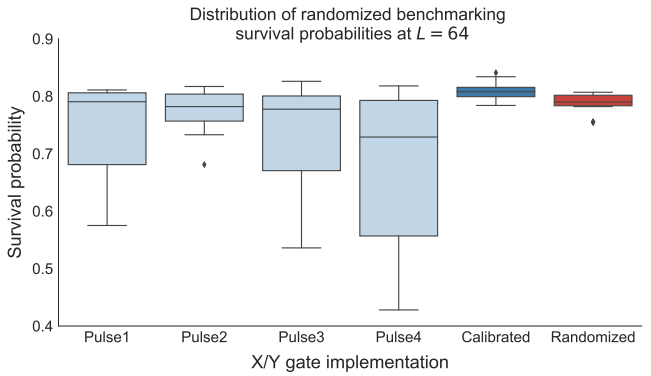

FIG. 5. **Evidence of reduced coherent error:**. Shown here is the distribution of randomized benchmarking survival probabilities at $L = 64$ for each of six native gate implementations. The first four boxes (light blue) result four intentionally miscalibrated $\pi/2$ rotations. The coherent noise present in these implementations leads to large variance in the survival probability over sequences. The fifth (dark blue) box illustrates the survival probability using a highly-tuned gate implementation. It displays improved average survival probability as well as reduced variance. The final box (dark red) illustrates the distribution for a randomized MQG composed of Pulse1 through Pulse4. It performs comparably to the highly-calibrated implementation in both average survival probability and variance over random sequences. The reduced variance of the MQG is a tell-tale sign of reduced coherent error in the effective channel[10].

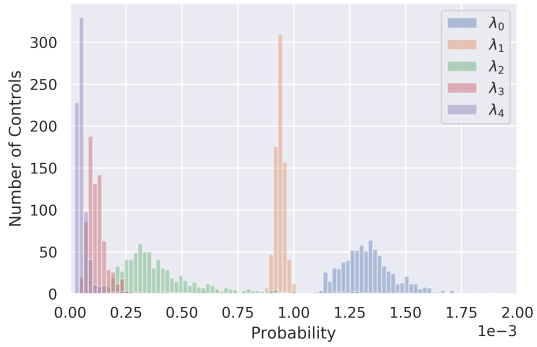

FIG. 6. Sparsity of MQGs as a function of the sparisty parameter λ, with higher weight controls omitted to emphasize the low-weight distribution. Numerical values of λ_i are: $\lambda_0 = 0$, $\lambda_1 = 6.25\text{e-}05$, $\lambda_2 = 0.0001$, $\lambda_3 = 0.00025$, $\lambda_4 = 0.0005$. As the parameter is increased from zero, the number of controls with low probability increases. For λ_4 virtually all of the 750 controls have vanishingly small probability, allowing for the restriction of the MQG to a small number of controls.

randomized implementations are both significantly less dispersed, consistent with stochastic errors. This experiment therefore provides compelling evidence that the MQG suffers considerably less coherent error than any of the miscalibrated gates from which it was constructed.

We note that, while this example is somewhat contrived (the calibrated gate is clearly the best), it nonetheless succeeds in demonstrating that the MQGs are capable of outperforming the individual gates of which they are composed. In future work, we hope to apply this technique to two-qubit gates, which are significantly more difficult to tune up in general and are more likely to possess lingering coherent errors. Tomography or modeling can then be used to construct the process matrix estimates required to build a mixed gate.

B. Numerical implementations

In the following numerical results, we use the various methods in Section III to build a range of one- and two-qubit MQGs for model systems. We analyze the resulting mixed gates for a range of Hamiltonian parameters, demonstrating both a reduction in diamond distance and an improved robustness to model uncertainty.

1. *Single-qubit example*

We consider the following dimensionless model for a single qubit subject to frequency drift:

$$H(\delta, \epsilon, t) = \epsilon\sigma_z + (1+\delta)(c_x(t)\sigma_x + c_y(t)\sigma_y) \qquad (31)$$

where ϵ corresponds to fluctuations in qubit frequency and δ corresponds to fluctuations in the control field.

To generate the initial controls, we use the GRAPE algorithm[27] with N=25 steps and total evolution time of π to generate 750 candidate controls. However, by imposing the sparisty constraint discussed in Section III B 3, we found MQGs consisting of just 10 controls. Fig. 6 shows that as λ increases, the support of the MQG becomes more peaked, with most of their probability mass concentrating on a smaller number of controls. In our implementation of the GRAPE algorithm, we use the performance function presented in [27], and average over different values of δ and ϵ using Gaussian quadrature when computing the gradient, so that we find controls that are naturally robust. The standard deviations considered for all parameters in our numerical experiments were fixed to $\sigma = .001$. Finally, we assume that the errors on σ_x and σ_y are perfectly correlated, as in our experimental implementation. We note that advanced quantum control protocols may provide an even more principled approach to control synthesis. DMORPH [43], for example, explores continuous families of controls on fidelity level sets, thereby enabling further optimizations against secondary criteria, such as the duration of the control pulse or robustness to drift.

Using controls generated in this way, the MQGs produced for $X_{\pi/2}$ and $Y_{\pi/2}$ are qualitatively similar, with the results for $X_{\pi/2}$ shown in Figure 7. By imposing the penalty from Section III B 2, we sought to ensure that the

algorithm preferentially selected controls with smaller errors. Adding this constraint increased the performance of the 0MQG by nearly an order of magnitude at the origin, and produced a 1MQG whose performance is an order of magnitude better than the 0MQG away from the origin. Imposing this constraint allows us to trade off flatness for performance. This shows that through adding constraints to our optimization routine, we can make the MQG practically useful.

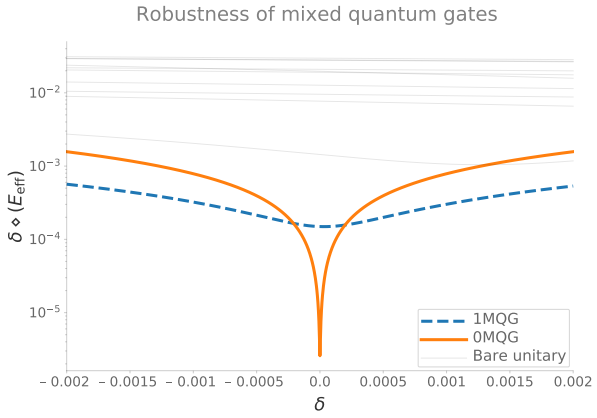

FIG. 7. **One-qubit robust MQG** Numerical results showing the the diamond distance errors for a generator-exact MQG (dashed blue), a robust generator-exact MQG (solid orange), and a large set of bare unitary controls (thin solid grey) as a function of a the fractional amplitude error. The target is an $X_{\pi/2}$ gate on a single qubit. The generator-exact MQGs dramatically outperforms both the robust MQG and any of the bare unitaries at $\delta = 0$. However, even a relatively small amplitude error leads to a regime in which the the robust MQG will yield the better gate. Similar results are obtained for drift in the qubit frequency.

2. *Two-qubit example*

In our two-qubit example we consider a resonant exchange interaction, similar to that in [44]:

$$H(\vec{\delta},\vec{\epsilon},t) = \sum_{j=1}^{2}(\epsilon_j\sigma_z^j + (1+\delta_j)(c_x^j(t)\sigma_x^j + c_y^j(t)\sigma_y^j)) + \frac{1}{10}(XX+YY) \tag{32}$$

In this example it was infeasible to use GRAPE to return non-trivial solutions (always got the same result). Instead we manually selected piecewise constant echoing sequences with 500 steps and total evolution time of $\frac{5\pi}{2}$. In particular, we considered $RX(\pi)$, $RX(-\pi)$, $RY(\pi)$ and $RY(-\pi)$ bang-bang sequences [45], consisting of all combinations of simultaneous π pulses activated at multiples of 8 steps from the beginning of the controls, and the same multiple of 8 steps prior to the end of the controls. To give the control family a variety of RF errors, we also added uniformly distributed amplitude errors to each π pulse, between $-.25\%$ and $.25\%$.

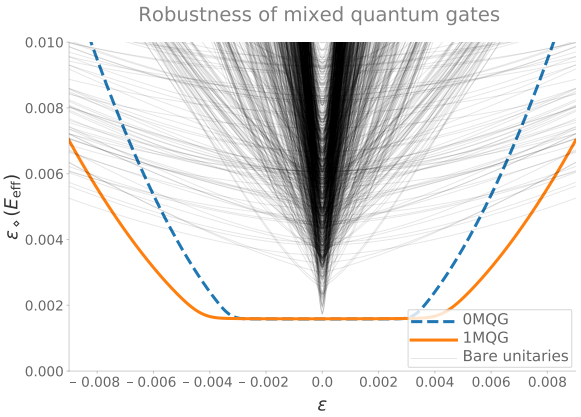

FIG. 8. **Two-qubit robust MQG** Numerical results showing the the diamond distance errors for a generator-exact MQG (dashed blue), a robust generator-exact MQG (solid orange), and a large set of bare unitary controls (thin solid grey) as a function of a spurious detuning (quantified by the ratio of the detuning to the maximum control amplitude). It is intended to models a pair of tunable qubits interacting via resonant exchange. Both MQGs can be seen to outperform the bare unitary control over a wide range of detunings. The robust MQG performs always outperforms the generator-exact MQG, and both MQGs outperform any of the bare unitary controls over a wide range of detunings. Similar results are obtained for drift in the qubit control amplitude.

In this example, we find more modest improvements to performance, as shown in Figure 8. There are now four free parameters to optimize over, and the uncontrolled entangling interaction means that there is little room for variation in the controls. Nonetheless, using an MQG improves performance by half an order of magnitude at the origin relative to the constituent controls, and up to an order of magnitude away from the origin. For all values of the drifting parameters we see that the 1MQG performs as well or better than 0MQG.

V. CONCLUSION AND FUTURE WORK

We have shown numerically that using MQGs can reduce coherent error by more than an order of magnitude in diamond norm, over a wide range of quasi-static values of noise. In addition, we have demonstrated that these approximate controls can be generated through optimal control (GRAPE), and that the minimization problem is tractable.

Future directions for this work include demonstrating the routine experimentally on a two-qubit gate, moving the random gate selection from a pre-compilation step to runtime logic onboard the control electronics, investigat-

ing other optimization routines such as CRAB [46] and GOAT[47], and using more sophisticated benchmarking routines such as GST[48] to quantitatively investigate the performance of our method.

That these methods depend on accurate system models is a clear shortcoming. The robust MQG The numerical work in the paper assumes access to a model of the system, however an experimentalist may not have a model readily available to describe the system, e.g. in the presence of unknown on-chip crosstalk, or an uncalibrated transfer function of the system. Even if a model is available, it might be computationally inconvenient to simulate, i.e. for more than a few qubits. In these situations, one approach would be to use *in-situ* optimal control techniques [49–51]. While performing a complete optimization in this way would require full process tomography, one could instead use partial tomography. By selecting pre– and post –rotations that correspond to measuring Pauli-moments of interest in the Hamiltonian, such as unwanted $Z \otimes Z$ crosstalk terms, one could perform optimization over fewer parameters.

VI. ACKNOWLEDGEMENTS

This material was funded in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research Quantum Testbed Program, as well as by Sandia National Laboratories' Laboratory Directed Research and Development program. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

[1] M. B. Hastings, Quantum Information & Computation **17**, 488 (2017).
[2] E. T. Campbell, B. M. Terhal, and C. Vuillot, Nature **549**, 172 (2017).
[3] S. J. Beale, J. J. Wallman, M. Gutiérrez, K. R. Brown, and R. Laflamme, Physical Review Letters **121** (2018), 10.1103/physrevlett.121.190501.
[4] E. Campbell, "A random compiler for fast hamiltonian simulation," (2018), arXiv:1811.08017.
[5] D. W. Leung, CoRR **cs.CC/0012017** (2000).
[6] L. Viola and E. Knill, Physical Review Letters **94** (2005), 10.1103/physrevlett.94.060502.
[7] L. F. Santos and L. Viola, Physical Review Letters **97** (2006), 10.1103/physrevlett.97.150501.
[8] J. J. Wallman and J. Emerson, Physical Review A **94** (2016), 10.1103/physreva.94.052325.
[9] M. Ware, G. Ribeill, D. Rist, C. A. Ryan, B. Johnson, and M. P. da Silva, "Experimental demonstration of pauli-frame randomization on a superconducting qubit," (2018), arXiv:1803.01818.
[10] H. Ball, T. M. Stace, S. T. Flammia, and M. J. Biercuk, Physical Review A **93** (2016), 10.1103/physreva.93.022303.
[11] T. Proctor, M. Revelle, E. Nielsen, K. Rudinger, D. Lobser, P. Maunz, R. Blume-Kohout, and K. Young, "Detecting, tracking, and eliminating drift in quantum information processors," (2019), arXiv:1907.13608.
[12] K. Rudinger, T. Proctor, D. Langharst, M. Sarovar, K. Young, and R. Blume-Kohout, Physical Review X **9** (2019), 10.1103/physrevx.9.021045.
[13] K. Kraus, A. Böhm, J. D. Dollard, and W. H. Wootters, eds., *States, Effects, and Operations Fundamental Notions of Quantum Theory* (Springer Berlin Heidelberg, 1983).
[14] M.-D. Choi, Linear Algebra and its Applications **10**, 285 (1975).
[15] K. Życzkowski and I. Bengtsson, Open Systems & Information Dynamics (OSID) **11**, 3 (2004).
[16] J. L. O'Brien, G. J. Pryde, A. Gilchrist, D. F. V. James, N. K. Langford, T. C. Ralph, and A. G. White, Physical Review Letters **93** (2004), 10.1103/physrevlett.93.080502.
[17] J. M. Chow, J. M. Gambetta, A. D. Córcoles, S. T. Merkel, J. A. Smolin, C. Rigetti, S. Poletto, G. A. Keefe, M. B. Rothwell, J. R. Rozen, M. B. Ketchen, and M. Steffen, Physical Review Letters **109** (2012), 10.1103/physrevlett.109.060501.
[18] J. Preskill, Accesible via http://www. theory. caltech. edu/people/preskill/ph229 **2015** (1997).
[19] A. Carignan-Dugas, M. Alexander, and J. Emerson, Quantum **3**, 173 (2019).
[20] J. Wallman, C. Granade, R. Harper, and S. T. Flammia, New Journal of Physics **17**, 113020 (2015).
[21] D. Gottesman, "Maximally sensitive sets of states," (2019), arXiv:1907.05950.
[22] N. Johnston and D. W. Kribs, Journal of Physics A: Mathematical and Theoretical **44**, 495303 (2011).
[23] J. Watrous, *The theory of quantum information* (Cambridge University Press, 2018).
[24] J. J. Wallman, "Bounding experimental quantum error rates relative to fault-tolerant thresholds," (2015), arXiv:1511.00727.
[25] D. Aharonov, A. Kitaev, and N. Nisan, arXiv preprint quant-ph/9806029 (1998).
[26] M. B. Hastings, "Turning gate synthesis errors into incoherent errors," (2016), arXiv:1612.01011.
[27] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, Journal of Magnetic Resonance **172**, 296 (2005).
[28] S. Wright and J. Nocedal, Springer Science **35**, 7 (1999).
[29] L. Khachiyan, Soviet Mathematics Doklady **20** (1979).
[30] S. Diamond and S. Boyd, Journal of Machine Learning Research **17**, 1 (2016).
[31] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, Journal of Control and Decision **5**, 42 (2018).
[32] D. Puzzuoli, C. Granade, H. Haas, B. Criger, E. Magesan, and D. G. Cory, Physical Review A **89** (2014),

10.1103/physreva.89.022306.
[33] E. Magesan, D. Puzzuoli, C. E. Granade, and D. G. Cory, Physical Review A **87** (2013), 10.1103/physreva.87.012324.
[34] D. Gottesman, (1998), arXiv:quant-ph/9807006.
[35] S. Aaronson and D. Gottesman, Physical Review A **70** (2004), 10.1103/physreva.70.052328.
[36] T. Yoder, (2012).
[37] R. S. Bennink, E. M. Ferragut, T. S. Humble, J. A. Laska, J. J. Nutaro, M. G. Pleszkoch, and R. C. Pooser, Physical Review A **95** (2017), 10.1103/physreva.95.062337.
[38] S. Shah, J. E. Mitchell, and M. Kupferschmid, Computers & Operations Research **28**, 85 (2001).
[39] R. Tibshirani, Journal of the Royal Statistical Society. Series B (Methodological) , 267 (1996).
[40] M. Pilanci, L. E. Ghaoui, and V. Chandrasekaran, in *Advances in Neural Information Processing Systems 25*, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Curran Associates, Inc., 2012) pp. 2420–2428.
[41] J. S. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. S. Fried, S. Hong, P. Karalekas, C. B. Osborn, A. Papageorge, E. C. Peterson, G. Prawiroatmodjo, N. Rubin, C. A. Ryan, D. Scarabelli, M. Scheer, E. A. Sete, P. Sivarajah, R. S. Smith, A. Staley, N. Tezak, W. J. Zeng, A. Hudson, B. R. Johnson, M. Reagor, M. P. da Silva, and C. Rigetti, "Unsupervised machine learning on a hybrid quantum computer," (2017), arXiv:1712.05771.
[42] E. Magesan, J. M. Gambetta, and J. Emerson, Physical Review Letters **106** (2011), 10.1103/physrevlett.106.180504.
[43] J. Dominy and H. Rabitz, Journal of Physics A: Mathematical and Theoretical **41**, 205305 (2008).
[44] D. C. McKay, S. Filipp, A. Mezzacapo, E. Magesan, J. M. Chow, and J. M. Gambetta, Physical Review Applied **6** (2016), 10.1103/physrevapplied.6.064007.
[45] L. Viola and S. Lloyd, Physical Review A **58**, 2733 (1998).
[46] T. Caneva, T. Calarco, and S. Montangero, Physical Review A **84** (2011), 10.1103/physreva.84.022326.
[47] S. Machnes, E. Assémat, D. Tannor, and F. K. Wilhelm, Physical Review Letters **120** (2018), 10.1103/physrevlett.120.150401.
[48] R. Blume-Kohout, J. K. Gamble, E. Nielsen, K. Rudinger, J. Mizrahi, K. Fortier, and P. Maunz, Nature Communications **8** (2017), 10.1038/ncomms14485.
[49] R.-B. Wu, B. Chu, D. H. Owens, and H. Rabitz, Physical Review A **97** (2018), 10.1103/physreva.97.042122.
[50] J. Kelly, R. Barends, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. Fowler, I.-C. Hoi, E. Jeffrey, A. Megrant, J. Mutus, C. Neill, P. O'Malley, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. White, A. Cleland, and J. M. Martinis, Physical Review Letters **112** (2014), 10.1103/physrevlett.112.240504.
[51] C. Ferrie and O. Moussa, Physical Review A **91** (2015), 10.1103/physreva.91.052306.
[52] K. Fan, Proceedings of the National Academy of Sciences of the United States of America **37**, 760 (1951).

VII. APPENDIX

A. Proof of diamond distance inequality

Here we prove the claim of (14) that:

$$\epsilon_\diamond(\mathsf{E}_{\text{eff}}) \leq \sum w_i\, \epsilon_\diamond(\mathsf{E}_i). \tag{33}$$

The effective error channel for a mixed quantum gate is $\mathsf{E}_{\text{eff}} = \sum w_i\, \mathsf{E}_i$, where E_i are the error channels for the component gates. The diamond distance to the identity of the effective error channel is:

$$\epsilon_\diamond\left(\mathsf{E}_{\text{eff}}\right) = \frac{1}{2} \sup_\rho ||(\mathsf{I}_d \otimes \mathsf{I}_d)(\rho) - (\mathsf{E}_{\text{eff}} \otimes \mathsf{I}_d)(\rho)||_1 \tag{34}$$

$$= \frac{1}{2} \sup_\rho ||\sum w_i\, ((\mathsf{I}_d - \mathsf{E}_i) \otimes \mathsf{I}_d)(\rho)||_1 \tag{35}$$

For qubits, the space of density matrices is compact, so the supremum is achievable. Call a state that achieves the supremum ρ^*. Then

$$\epsilon_\diamond\left(\mathsf{E}_{\text{eff}}\right) = \frac{1}{2}||\sum w_i\, ((\mathsf{I}_d - \mathsf{E}_i) \otimes \mathsf{I}_d)(\rho^*)||_1 \tag{36}$$

$$= \frac{1}{2}||\sum w_i\, \rho_i^*||_1, \tag{37}$$

where we have defined $\rho_i^* = ((\mathsf{I}_d - \mathsf{E}_i) \otimes \mathsf{I}_d)(\rho^*)$. The nuclear norm above is equal to the sum of the singular values of $\sum w_i \rho_i$. Using the Ky Fan singular value inequality [52] , we have

$$\epsilon_\diamond\left(\mathsf{E}_{\text{eff}}\right) \leq \frac{1}{2} \sum_i w_i ||\rho_i^*||_1 \tag{38}$$

$$\leq \sum_i w_i\, \epsilon_\diamond(\mathsf{E}_i) \tag{39}$$

The second inequality above follows because ρ^* defines an explicit lower bound for the diamond distance for each of the component error maps.