Robustly decorrelating errors with mixed quantum gates

Anthony M. Polloreno[*]
Rigetti Computing, Berkeley, CA

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

(Dated: September 10, 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 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 (QIP). Errors of this type are correlated in time may add up constructively or destructively, depending on the circuit. They are are computationally expensive to model and, because coherent errors may amplify when gates are repeated, it can be difficult to place tight analytic bounds on circuit performance [3]. 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 shown that coherent noise can be strongly suppressed by probabilistically mixing several distinct implementations of the target quantum gates. They focus on errors arising from various gate compilations, such as 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 a different unitary errors. So by selecting from these various compilations at random, the resulting quantum channel

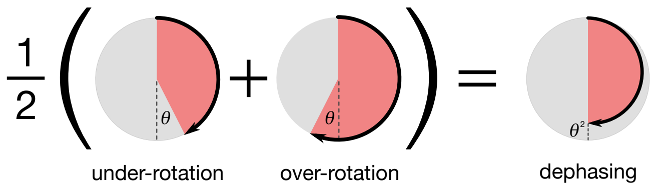

FIG. 1. A simple example of a mixed quantum gate. Using optimal control, two implementations of a Z_π gate are designed to have equal and opposite coherent errors (if one implementation over-rotates by a small angle θ, then the other *under*-rotates by θ). Each time the gate is used, one of these two 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 $\mathcal{O}\left(\theta^2\right)$.

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.

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 in the gate implementations. We demonstrate that, depending on what properties of the resulting channel are desired, different numerical optimization targets may be preferred. We present an experimental implementation of single-qubit mixed unitary controls on a superconducting qubit testbed at Rigetti Computing. Using randomized benchmarking, we are able to show a marked improvement in error rates, as well as a reduced variance in circuit outcome probabilities, consistent with a significant reduction in the coherence of the error [6]. We further apply our methods in simulation, constructing single- and two-qubit mixed unitary controls that are

* Email: ampolloreno@gmail.com

robust to drift and uncertainty in the control parameters.

II. PRELIMINARIES

A. Representing Errors in Quantum Gates

Suppose that one wishes to apply a unitary gate, G, on some quantum device. Implementing this operation on a real system generally involves the application of a sequence of classical control fields to some set of qubits. 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 and context[7], 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[8], Choi matrices[9], and Jamiolkowsi states[10]. But for the purposes of this article, the *process matrix* representation will be particularly convenient[11], 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:

$$\begin{aligned} \mathsf{vec}\left(\tilde{\mathsf{G}}(\rho)\right) &= \tilde{\mathcal{G}} \cdot \mathsf{vec}\left(\rho\right) && (1)\\ &= \mathcal{E} \cdot \mathcal{G} \cdot \mathsf{vec}\left(\rho\right) && (2)\end{aligned}$$

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,

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

and

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

Process matrices written in this Pauli basis are often referred to as *Pauli transfer matrices* [12]. Error maps represented in this basis take the particularly nice form,

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

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

$$\begin{aligned} \mathcal{E} &= \exp\left(\mathcal{L}\right) && (6)\\ &= \mathcal{I}_d + \mathcal{L} + \frac{1}{2}\mathcal{L}^2 + \mathcal{O}\left(\mathcal{L}^3\right). && (7)\end{aligned}$$

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,

$$\tilde{\mathsf{G}}_{\text{eff}} = \sum_i w_i \tilde{\mathsf{G}}_i = \left(\sum_i w_i \mathsf{E}_i\right) \circ \mathsf{G} \quad (8)$$

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,

$$\mathsf{E}_{\text{eff}} = \sum_i w_i \mathsf{E}_i. \quad (9)$$

Two important error metrics are the average gate infidelity (AGI), $\epsilon_{\mathcal{F}}$, and the diamond distance to the target (or simply, the *diamond distance*), $\epsilon_\diamond$ [14], defined as:

$$\begin{aligned} \epsilon_{\mathcal{F}}(\mathsf{E}) &= \frac{d^2 - \mathrm{Tr}\,\mathcal{E}}{d^2 + d}, && (10)\\ \epsilon_\diamond\left(\mathsf{E}\right) &= \frac{1}{2}\sup_\rho ||(\mathsf{I}_d \otimes \mathsf{I}_d)(\rho) - (\mathsf{E} \otimes \mathsf{I}_d)(\rho)||_1, && (11)\end{aligned}$$

where $d = 2^n$ and I_d is the d-dimensional identity operator. In Eq. (10), 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 [15]. The diamond distance is subaddative [14] 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 [16].

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

$$\epsilon_{\mathcal{F}}(\mathsf{E}_{\text{eff}}) = \sum_i w_i \, \epsilon_{\mathcal{F}}(\mathsf{E}_i). \tag{12}$$

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),

$$\epsilon_{\diamond}(\mathsf{E}_{\text{eff}}) \leq \sum_i w_i \, \epsilon_{\diamond}(\mathsf{E}_i). \tag{13}$$

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 [17] 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 (9) and (6), the effective error map for a MQG in terms of the component error generators is,

$$\mathcal{E}_{\text{eff}} \simeq \sum_i w_i \left(\mathcal{I}_d + \mathcal{L}_i + \frac{1}{2}\mathcal{L}_i^2 \right) \tag{14}$$

These error generators lie in a vector space. If, as Campbell required, the origin lies in their convex hull (see Fig. 6), then there exists a choice for the weights, w_i^*, such that $\sum_i w^* = 1$ and,

$$\sum_i w_i^* \mathcal{L}_i = 0 \tag{15}$$

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. Campbell considered error channels that were strictly unitary, so the error generators in Eq. (15) were Hamiltonians. Hamiltonian generators can have

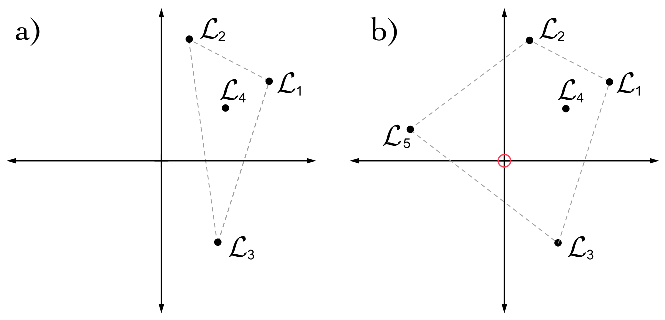

FIG. 2. A target unitary gate can be implemented a number of ways, each associated with a different error generator, $\mathcal{L}_i$. These generators lie in a vector space, which we illustrate as two-dimensional here. a) Four error generators. The origin is not contained in their convex hull, so there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

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. (15) 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. (15) to only the Hamiltonian component of the generators. In the rest of this paper, we will be considering gates subject only to unitary errors, so we will neglect this complication going forward.

While generator-exact MQGs 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 (14) 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 unital submatrix of the effective error map. The resulting diagonal error map is Pauli stochastic. For a single qubit gate, it takes the form,

$$\mathcal{E}_{\text{eff}} = \sum_i w_i^P \mathcal{E}_i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & p_x & 0 & 0 \\ 0 & 0 & p_y & 0 \\ 0 & 0 & 0 & p_z \end{pmatrix}, \tag{16}$$

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, $\mathcal{E}_i$.

In the remainder of this paper, we will formalize this process, presenting efficient convex programs to both generate a family of unitary controls and compute the weights required to yeild either type of MQG. We discuss how one might include additional desiderata, including reduced sensitivity to environmental fluctuation as well as limiting the number of native unitaries that participate in the MQG.

III. RESULTS

A. Experimental

In modern quantum computing platforms, diagnosing and correcting coherent errors remains a challenge. Methodologies like RPE and DRAG are sufficient for routinely achieving high single qubit fidelities in superconducting qubit architectures, however as coherence times increase, the community will continue to be interested in removing coherent errors to achieve higher fidelities. As a proof of principle, we implemented an MQG with a superconducting transmon qubit on the Rigetti 19Q-Acorn chip, whose performance we describe below. While this device consisted of 20 qubits with fixed couplings, here we will give results for a single qubit MQG on qubit 8. For more details about this particular device, characterization can be found in [18].

To implement an MQG on this qubit, we first calibrated a 10-sample, 50ns Gaussian pulse by coherently amplifying and correcting over- and under-rotation errors through repeated application. After the gate was sufficiently calibrated, four intentionally miscalibrated Gaussian pulses were produced by scaling the amplitude of the calibrated $RX(\frac{\pi}{2})$ pulse by 106.4%, 103.9%, 93.7% and 91.2%.

As discussed in Section V A 2, we chose to minimize the off diagonal elements of the process matrix. To benchmark the quality of the MQG, we then performed six randomized benchmarking experiments[19]: one for each over– and under–calibrated pulse, one for the calibrated pulse, and one for the mixed process. We used 1000 shots per experiment and 10 sequences per sequence length, for sequence lengths of $2, 4, 8, 16, 32$ and 64. In each case, our Clifford operations were decomposed into $\mathrm{RX}(\frac{\pi}{2})$ and $\mathrm{RY}(\frac{\pi}{2})$. In our implementation, these gates are implemented using the same pulse envelope definitions and control electronics, phase shifted by $\frac{\pi}{2}$ radians, and are therefore subject to identical miscalibration errors. The results are shown in Figure 3 for sequence lengths $L = 64$. Fitting to the randomized benchmarking decay curves, we find one-qubit gate fidelities of 99.3% for the calibrated pulse, 98.9% for Pulse1, 99.1% for Pulse2, 98.9% for Pulse3, 98.5% for Pulse4, and 99.2% for the MQG, demonstrating that it performs almost as well as the calibrated pulse, and better than the constituent pulses.

More importantly, minimizing the off-diagonal elements of the process matrix, we expect to produce a process with suppressed coherent error. To see that this is the case, we can observe the drastic reduction in the variance at each point for the MQG relative to the miscalibrated pulses. A discussed in [6], for coherent error

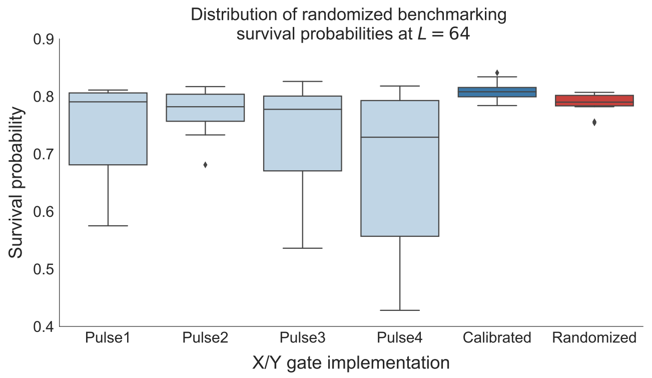

FIG. 3. Randomized benchmarking experiments ran using different pulse definitions. The first four boxes result from using each of four different implementations of the $\pi/2$ rotations. The coherent noise present in these implementations leads to large variance of the survival probability over sequences. 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 over survival probabilities 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.

models, noise will manifest as gamma distributed points for each sequence length. On the other hand, incoherent noise, such as depolarizing noise, will result in Gaussian distributed fidelity estimates for each randomized benchmarking sequence length. We see that the coherently miscalibrated controls in our RB experiment have long tails, consistent with gamma distributed random variables, while the calibrated and randomized implementations both have much shorter tails, consistent with Gaussian distributed random variables. Thus our experiment demonstrates that the MQG has a significantly less-coherent error channel.

B. Numerical Implementation

In the following numerical results, we use the methods in Section V to build MQGs, and then analyze their performance for a range of Hamiltonian parameters. More specifically, we will construct a family of controls with some degree of inherent robustness to drift, and leverage the numerical techniques introduced to assign each control a probability. When drawn from with these probabilities, the controls will form an MQG with enhanced robustness to drift on the control parameters. We consider the following dimensionless model for a single tunable qubit:

$$H(\delta, \epsilon, t) = \epsilon\sigma_z + (1 + \delta)(c_x(t)\sigma_x + c_y(t)\sigma_y) \tag{17}$$

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[20] with N=25 steps and total evolution time of π to generate 100 candidate controls. However, by imposing the sparisty constraint discussed in Section V B 3, we found MQGs consisting of just 10 controls. In our implementation of the GRAPE algorithm, we use the performance function presented in [20], 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 [21], 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 $RX(\frac{\pi}{2})$ and $RY(\frac{\pi}{2})$ are qualitatively similar, with the results for $RX(\frac{\pi}{2})$ shown in Figure 4. By imposing the penalty from Section V 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.

In our two-qubit example we consider the following model for two tunable qubits coupled by a resonant exchange interaction, similar to that in [22]:

$$H(\vec{\delta},\vec{\epsilon},t) = \sum_{j=1}^{2}(\epsilon_j\sigma_z^j + (1+\delta_j)(c_x^j(t)\sigma_x^j + c_y^j(t)\sigma_y^j)) + \frac{1}{10}(XX+YY) \tag{18}$$

[This section is still tough to read, but I'm not sure of a better way to handle it. The constituent controls are clearly contrived, so I think it best to be explicit about how they were constructed.]$_{\mathrm{AMP}}$In this example it was infeasible to use GRAPE to return non-trivial solutions. 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 [23], 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,

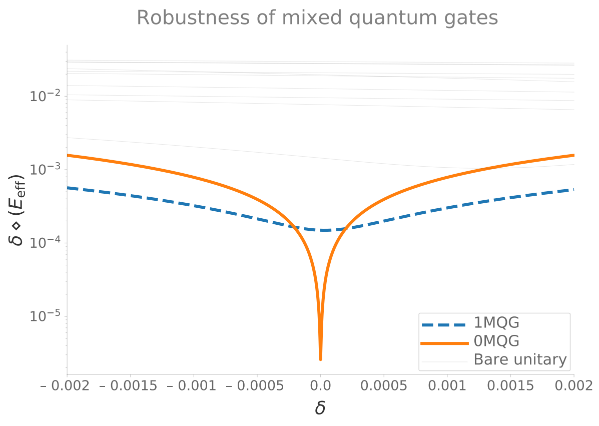

FIG. 4. [needs to be on previous page?]$_{\mathrm{AMP}}$Numerical results comparing a 0MQG to a 1MQG for a single tunable qubit, for $RX(\frac{\pi}{2})$. The results are qualitatively similar to those for $RY(\frac{\pi}{2})$. In this case the 0MQG outperforms both the 1MQG by two orders of magnitude, and the constituent controls by three orders of magnitude at the origin. However, varying over δ we see that the 1MQG outperforms the 0MQG by up to an order of magnitude.

we also added uniformly distributed amplitude errors to each π pulse, between $-.25\%$ and $.25\%$.

In this example, we find more modest improvements to performance, as shown in Figure 5. 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.

IV. 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.

Randomized protocols have a long history of outperforming their deterministic counterparts. In [24, 25] the use of randomness is discussed in the context of dynamical decoupling. In particular, it is shown that both stochastic and hybrid approaches can result in better or more tractable coherent controls. Additionally Pauli Frame Randomization[26] has been shown [27] to reduce coherent errors by introducing randomness in a precompilation step. The techniques discussed in this paper are different from both of these approaches in that they explicitly produce incoherent errors through mixing, and

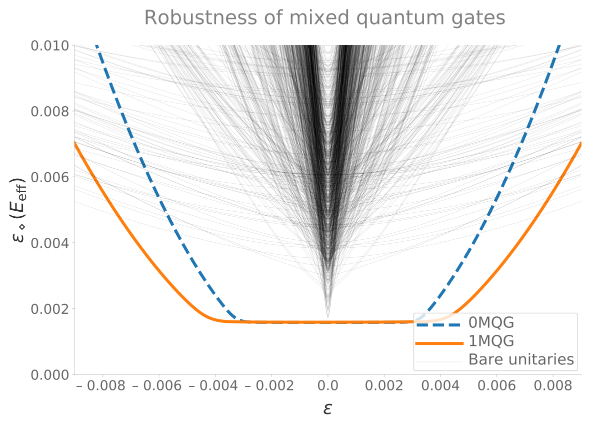

FIG. 5. [needs to be on previous page]KCY Numerical results comparing a 0MQG to a 1MQG for a pair of tunable qubits, with a resonant exchange interaction. Shown with lower alpha values are example constituent controls. The 0MQG and 1MQG can be seen to outperform these controls by half of an order of magnitude at the origin. For all detuning values the 1MQG performs as well or better than the 0MQG. When there is .2% drift in the qubit frequency, the 1MQG outperforms members of the control families by almost an order of magnitude in diamond norm. Similarly, for .2% drift in the qubit control amplitude, we see that the 1MQG outperforms the the constituent controls by over half an order of magnitude.

that the randomness is hardware efficient for any desired gate.

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, investigating other optimization routines such as CRAB [28] and GOAT[29], and using more sophisticated benchmarking routines such as GST[30] to quantitatively investigate the performance of our method.

Another interesting area of research would be using model-free approaches. 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 [31–33] to generate candidate controls, and then use an optimizer like Nelder-Mead to perform the minimization. While performing a complete optimization in this way would require full process tomography, one could instead optimize via 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, one could perform optimization over fewer parameters.

V. CONSTRUCTING MIXED QUANTUM GATES

In this section, we present a methodology for constructing mixed quantum channels, formalizing the intuitive approach discussed above. 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 [20], 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. We discuss this approach in some detail in Sec. III 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 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:

$$\underset{w_i\geq 0, \sum_i w_i = 1}{\textbf{minimize}}: \quad \epsilon_\diamond\left(\sum_i \mathsf{E}_i w_i\right) \tag{19}$$

The diamond distance, however, is a non-linear function that in general requires its own convex optimization to compute[14]. 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. (15), 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 that produces approximate unitary operations, will continually generate controls until the geometric constraint is satisfied and then identify the optimal weighting. We consider instead a situation in which one has access only to m distinct, precomputed gate implementations and we wish to to identify the MQG with minimum error. An efficiently computable heuristic is to minimize the Frobenius norm of the first-order effective error generator:

$$\underset{w_i\geq 0,\sum_i w_i=1}{\textbf{minimize}}: \quad \left|\left|\sum_i w_i\mathcal{L}_i\right|\right|_F \tag{20}$$

This optimization problem is equivalent to:

$$\underset{w_i\geq 0,|w|_1=1}{\textbf{minimize}}: ||\mathbf{L}\cdot\vec{w}||_2 \tag{21}$$

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. Nevertheless, linearly constrained minimization problems with quadratic cost functions are convex and have been proven to be efficiently solvable by, e.g. the ellipsoid method [34, 35]. Many existing convex solver software packages are available that can solve these problems efficiently in practice. [Need some citations here.]$_{\text{KCY}}$

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. As discussed above, Pauli-exact MQGs can be constructed with vanishing coherent error, enabling the use of much more efficient Pauli stochastic simulators that are able to model significantly more qubits than their vector-state counterparts. 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:

$$\underset{w_i\geq 0,\sum_i w_1=1}{\textbf{minimize}}: ||\sum_i w_i\mathcal{E}_i - \mathsf{diag}(\sum_i w_i\mathcal{E}_i)||_2 \tag{22}$$

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 $\mathcal{E}_i$. [1] Eq. (22) is then equivalent to

$$\underset{w_i\geq 0,|w|_1=1}{\textbf{minimize}}: ||\mathbf{E}\cdot\vec{w}||_2 \tag{23}$$

The similarity with Eq. 21 is clear and the same methods can be used to solve both.

[do something with the paragraph]$_{\text{KCY}}$ In this case, if (21) is satisfied, the resulting error channel will be *exactly* a Pauli Channel. If this is achievable, then the circuit can be more efficiently simulated by Monte Carlo methods. Better yet, if the bare circuit consists of only Clifford gates, stabilizer methods can be used in conjunction with the Monte Carlo methods to give sub-exponential simulation times.[36] Other authors have considered the problem of finding the closest Pauli or Clifford Channel to a given process [37, 38] but, unlike our routine, this does not produce a channel that is decomposable into a given family of controls.

B. Additional Constraints

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. 6). This often leads to the the minimization problems discussed in the previous section being massively under-constrained, yielding many possible 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.

1. Robustness to drift and model uncertainty

While mixed quantum gates offer significant performance improvements, their construction assumes a good knowledge of the errors experienced by the constituent gates. But often 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 [cite Tim's drift manuscript]$_{\text{KCY}}$. In either case, the performance of the MQG will be negatively impacted. We can model the effect of this uncertainty by writing the gate

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

If the uncertainty in our estimate of the gate errors is small, then we can write a gate's error generator as a Taylor expansion in the model uncertainty

$$\mathcal{L}_i(\vec{\delta}) = \mathcal{L}_i(0) + \sum_k \delta_k \, \mathcal{L}_{i,k}(\vec{0}) + \mathcal{O}\left(\delta^2\right) \qquad (24)$$

In the above expression we have used the comma derivative,

$$\mathcal{L}_{i,k}(\vec{\delta}_0) = \frac{\partial}{\partial \delta_k} \mathcal{L}_i(\vec{\delta})|_{\vec{\delta}=\delta_0}. \qquad (25)$$

most control electronics experience drift over time scales relevant to QIP performance. Because of this drift, the quality of the MQG will degrade. Thus, we would like to design MQGs that are robust to this drift. To enforce robustness, we can consider the generators in (**??**) to be functions of a control vector $\vec{\delta}$, and if the fluctuations are small enough, it is convenient to write the gate as a Taylor expansion in this small parameter:

$$\begin{aligned}\tilde{\mathcal{G}}_i(d\vec{\delta}) = \exp(-i(\mathcal{L}_i(\vec{0}) + \frac{\partial}{\partial \delta_j}\mathcal{L}_i(\vec{0})d\delta_j \\ + \frac{1}{2}\frac{\partial^2}{\partial \delta_j \partial \delta_k}\mathcal{L}_i(\vec{0})d\delta_j d\delta_k + \ldots))\mathcal{G}\end{aligned} \qquad (26)$$

Now, instead of requiring that only the error generators average to zero, we can additionally impose a similar condition on the higher-order derivatives with respect to parameters that may drift. Specifically, we are interested in the n^{th} order derivatives: [Should the figures change from H to something else?]$_{\text{KCY}}$ [Right. It seems to me they should be written as something like $D^1_{i,k}$. Then in the equation below I can use the comma notation, but for the i^{th} derivative it's going to be some gross multi-index... I think your point is that the comma is standard notation (although I've never used it before.)]$_{\text{AMP}}$

$$D^n_{i,(j_1,\ldots,j_n)} = \frac{1}{n!}\frac{\partial^n}{\partial \delta_{j_1} \ldots \partial \delta_{j_n}} \mathcal{L}_i(\vec{\delta})|_{\vec{\delta}=\vec{0}} \qquad (27)$$

If the dimension of $\vec{\delta}$ is d, the indices $j_1, \ldots, j_n$ take on values in $1, \ldots, d$ and this matrix has d^n entries. We say that an MQG is robust to order ℓ (an ℓMQG) if for all $1 \le k \le \ell$:

$$\sum_i w_i (\sum_{n=0}^{k} D^n_i)^n = 0 \qquad (28)$$

In particular, we see that a 0MQG satisfies (**??**). More generally, these conditions imply that an ℓMQG is insensitive to ℓ^{th} order in drift in $\vec{\delta}$. By Taylor expanding (26) in $d\vec{\delta}$, one can see that the the first ℓ derivatives of an ℓMQG will be zero. Furthermore if we can find controls such that $||D^n_i|| \le \alpha$, (28) is a pproximately satisfied in the sense that:

$$||\sum w_i D^n_i|| + \mathcal{O}\left(\alpha^2\right) = 0 \qquad (29)$$

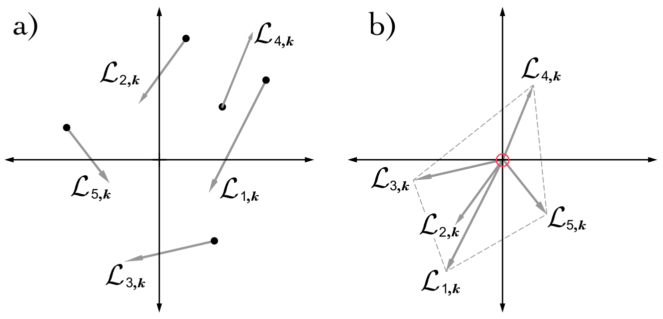

FIG. 6. A target unitary gate can be implemented a number of ways, each with a different effective Hamiltonian error. a) The error Hamiltonians shown with their derivative with respect to a control parameter. As this parameter drifts, a MQG may drift, leading to a first-order error. b) The derivatives also lie in a vector space. If the origin lies in their convex hull $\mathsf{Conv}(\{H_k \oplus H_{k,1} \oplus H_{k,2} \oplus \cdots\}_k)$[I'm not sure what you mean by the $_k$ subscript, nor why the tensor product is there. Also I think your indices are backwards?]$_{\text{AMP}}$ then it may be possible to construct a robust 1-MQG. [This should appear in the main text somewhere. When we add the derivatives we make our space larger, but we still want the origin to be in the higher-dimensional convex hull.]$_{\text{KCY}}$

where $||\cdot||$ is the Frobenius norm. This condition guarantees, as before, that errors will be suppressed to second order in α, but now for all derivatives up to order ℓ. As discussed in [2], the geometric picture for 0MQGs is to consider the error generators as vectors, and to ensure zero is contained in their convex hull. Figure 6 generalizes this picture to 1MQGs and gives intuition for the conditions required to produce them. To generate ℓMQGs, we first define the vectorized derivative matrix $\mathbf{D}^\ell$ in a similar way to (**??**):

$$\mathbf{D}^\ell = \begin{pmatrix} D^\ell_{1_{1,1}} & D^\ell_{2_{1,1}} & \cdots \\ \vdots & \ddots & \\ D^\ell_{1_{d^2,d^2}} & & D^\ell_{m_{d^2,d^2}} \end{pmatrix} \qquad (30)$$

Using this, we can then solve the following convex optimization problem, generalizing (21):

$$\begin{aligned} &\underset{w_i \ge 0, |w|_1 = 1}{\textbf{minimize}} : ||\mathbf{D}^\ell w||_1 \\ &\textbf{subject to: } \forall n < \ell, ||\sum w_i D^n_i|| = 0 \end{aligned} \qquad (31)$$

with D^i_n defined in (27).

2. *Hamiltonian Norm Regularization*

While quadratic programs with equality constraints have analytic solutions, the program we are considering has both constraints on the norm of the weight vector, and inequality constraints. Additionally, casting this problem as a convex optimization problem allows us to add penalty terms to the cost function to promote different properties of the MQG. In particular, while (31) is sufficient for suppressing the diamond norm to first order relative to the *worst* controls in the collection, it does not preferentially select the controls with the least error. As an example, any MQG that mixes equally over equal and opposite errors will satisfy (31), but the magnitude of the error of the resulting channel will depend on the magnitude of the original coherent errors. To encourage the inclusion of controls with smaller error, we may impose a penalty proportional to the norm of the included Hamiltonians. In our case, we choose to penalize for the ℓ_2 ($||\cdot||_2$) norm:

$$\begin{aligned} &\underset{w_i\geq 0, |w|_1=1}{\textbf{minimize}} : ||\mathbf{D}^\ell w||_1 + \eta \sum w_i ||D_i^0||_2 \\ &\textbf{subject to: } \forall n < \ell, ||\sum w_i D_i^n|| = 0 \end{aligned} \quad (32)$$

with $\eta \geq 0$. By increasing η, we can encourage the optimizer to include better controls while minimizing the diamond norm. The value of η that is best for any particular problem instance will depend on the values in $\mathbf{D}^\ell$.

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 that MQGs only require a small number of controls. As an example, consider Figure 6. In b), it is clear that H_4 is unnecessary to contain the origin in the convex hull of the error generators. Thus we would prefer that a 0MQG exclude H_4. However, if we additionally want our controls to form a 1MQG, we see from d) that we need H_4 in our control set, in which case we would like our algorithm to exclude H_1, since its derivative is contained in the convex hull of the other controls' derivatives.

In many machine learning contexts, lasso regularization [39] can be used to enforce sparsity in solutions, however this is insufficient for our purposes, as we already constrain w to be a valid probability distribution. Lasso regularization works by penalizing solutions with large ℓ_1 norm, however if w represents probabilities, it will always have an ℓ_1 norm of one. Conveniently, the problem of enforcing sparsity in such situations has been considered in [40] and can be expressed via another convex program that extends (31):

$$\begin{aligned} &\underset{m\in[N]}{\textbf{minimize}} \\ &\underset{\substack{w_i\geq 0, |w|_1=1, \\ t\geq 0}}{\textbf{minimize}} : ||\mathbf{D}^\ell w||_1 + t \\ &\textbf{subject to: } w_m > \frac{\lambda}{t} \\ &\qquad\qquad \forall n < \ell, ||\sum w_i D_i^n|| = 0 \end{aligned} \quad (33)$$

with $\lambda \geq 0$. As with η in the last subsection, the optimal value of λ is problem specific, depending on $\mathbf{D}^\ell$ and how sparse the solution needs to be. In the following sections we discuss both experimental and numerical implementations of MQGs, leveraging these optimization techniques.

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] H. Ball, T. M. Stace, S. T. Flammia, and M. J. Biercuk, Physical Review A **93** (2016), 10.1103/physreva.93.022303.
[7] K. Rudinger, T. Proctor, D. Langharst, M. Sarovar, K. Young, and R. Blume-Kohout, Physical Review X **9** (2019), 10.1103/physrevx.9.021045.
[8] 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).
[9] M.-D. Choi, Linear Algebra and its Applications **10**, 285 (1975).
[10] K. Życzkowski and I. Bengtsson, Open Systems & Information Dynamics (OSID) **11**, 3 (2004).
[11] 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.

[12] 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.

[13] J. Preskill, Accesible via http://www. theory. caltech. edu/people/preskill/ph229 **2015** (1997).

[14] J. Watrous, *The theory of quantum information* (Cambridge University Press, 2018).

[15] J. J. Wallman, "Bounding experimental quantum error rates relative to fault-tolerant thresholds," (2015), arXiv:1511.00727.

[16] D. Aharonov, A. Kitaev, and N. Nisan, arXiv preprint quant-ph/9806029 (1998).

[17] M. B. Hastings, "Turning gate synthesis errors into incoherent errors," (2016), arXiv:1612.01011.

[18] 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.

[19] E. Magesan, J. M. Gambetta, and J. Emerson, Physical Review Letters **106** (2011), 10.1103/physrevlett.106.180504.

[20] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, Journal of Magnetic Resonance **172**, 296 (2005).

[21] J. Dominy and H. Rabitz, Journal of Physics A: Mathematical and Theoretical **41**, 205305 (2008).

[22] 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.

[23] L. Viola and S. Lloyd, Physical Review A **58**, 2733 (1998).

[24] L. Viola and E. Knill, Physical Review Letters **94** (2005), 10.1103/physrevlett.94.060502.

[25] L. F. Santos and L. Viola, Physical Review Letters **97** (2006), 10.1103/physrevlett.97.150501.

[26] J. J. Wallman and J. Emerson, Physical Review A **94** (2016), 10.1103/physreva.94.052325.

[27] M. Ware, G. Ribeill, D. Ristè, C. A. Ryan, B. Johnson, and M. P. da Silva, "Experimental demonstration of pauli-frame randomization on a superconducting qubit," (2018), arXiv:1803.01818.

[28] T. Caneva, T. Calarco, and S. Montangero, Physical Review A **84** (2011), 10.1103/physreva.84.022326.

[29] S. Machnes, E. Assémat, D. Tannor, and F. K. Wilhelm, Physical Review Letters **120** (2018), 10.1103/physrevlett.120.150401.

[30] R. Blume-Kohout, J. K. Gamble, E. Nielsen, K. Rudinger, J. Mizrahi, K. Fortier, and P. Maunz, Nature Communications **8** (2017), 10.1038/ncomms14485.

[31] R.-B. Wu, B. Chu, D. H. Owens, and H. Rabitz, Physical Review A **97** (2018), 10.1103/physreva.97.042122.

[32] 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.

[33] C. Ferrie and O. Moussa, Physical Review A **91** (2015), 10.1103/physreva.91.052306.

[34] S. Wright and J. Nocedal, Springer Science **35**, 7 (1999).

[35] L. Khachiyan, Soviet Mathematics Doklady **20** (1979).

[36] D. Gottesman, (1998), arXiv:quant-ph/9807006.

[37] E. Magesan, D. Puzzuoli, C. E. Granade, and D. G. Cory, Physical Review A **87** (2013), 10.1103/physreva.87.012324.

[38] D. Puzzuoli, C. Granade, H. Haas, B. Criger, E. Magesan, and D. G. Cory, Physical Review A **89** (2014), 10.1103/physreva.89.022306.

[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] 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 (13) that:

$$\epsilon_\diamond(\mathsf{E}_{\text{eff}}) \leq \sum w_i\, \epsilon_\diamond(\mathsf{E}_i). \tag{34}$$

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:

$$\epsilon_\diamond\,(\mathsf{E}_{\text{eff}}) = \frac{1}{2}\sup_\rho ||(\mathsf{I}_d \otimes \mathsf{I}_d)(\rho) - (\mathsf{E}_{\text{eff}} \otimes \mathsf{I}_d)(\rho)||_1 \tag{35}$$

$$= \frac{1}{2}\sup_\rho ||\sum w_i\,((\mathsf{I}_d - \mathsf{E}_i) \otimes \mathsf{I}_d)(\rho)||_1 \tag{36}$$

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

$$\epsilon_\diamond\,(\mathsf{E}_{\text{eff}}) = \frac{1}{2}||\sum w_i\,((\mathsf{I}_d - \mathsf{E}_i) \otimes \mathsf{I}_d)(\rho^*)||_1 \tag{37}$$

$$= \frac{1}{2}||\sum w_i\,\rho_i^*||_1, \tag{38}$$

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 [41] , we have

$$\epsilon_\diamond\,(\mathsf{E}_{\text{eff}}) \leq \frac{1}{2}\sum_i w_i||\rho_i^*||_1 \tag{39}$$

$$\leq \sum_i w_i\,\epsilon_\diamond(\mathsf{E}_i) \tag{40}$$

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