Suppressing ZZ Crosstalk of Quantum Computers through Pulse and Scheduling Co-Optimization

Lei Xie
Tsinghua University
Beijing, China
xie-l18@mails.tsinghua.edu.cn

Jidong Zhai
Tsinghua University
Beijing, China
zhaijidong@tsinghua.edu.cn

ZhenXing Zhang
Tencent Quantum Laboratory
Shenzhen, China
alanzxzhang@tencent.com

Jonathan Allcock
Tencent Quantum Laboratory
Shenzhen, China
jonallcock@tencent.com

Shengyu Zhang
Tencent Quantum Laboratory
Shenzhen, China
shengyzhang@tencent.com

Yi-Cong Zheng*
Tencent Quantum Laboratory
Shenzhen, China
yicongzheng@tencent.com

ABSTRACT

Noise is a significant obstacle to quantum computing, and *ZZ* crosstalk is one of the most destructive types of noise affecting superconducting qubits. Previous approaches to suppressing *ZZ* crosstalk have mainly relied on specific chip design that can complicate chip fabrication and aggravate decoherence. To some extent, special chip design can be avoided by relying on pulse optimization to suppress *ZZ* crosstalk. However, existing approaches are non-scalable, as their required time and memory grow exponentially with the number of qubits involved.

To address the above problems, we propose a scalable approach by co-optimizing pulses and scheduling. We optimize pulses to offer an ability to suppress *ZZ* crosstalk surrounding a gate, and then design scheduling strategies to exploit this ability and achieve suppression across the whole circuit. A main advantage of such co-optimization is that it does not require special hardware support. Besides, we implement our approach as a general framework that is compatible with different pulse optimization methods. We have conducted extensive evaluations by simulation and on a real quantum computer. Simulation results show that our proposal can improve the fidelity of quantum computing on 4~12 qubits by up to 81× (11× on average). Ramsey experiments on a real quantum computer also demonstrate that our method can eliminate the effect of *ZZ* crosstalk to a great extent.

CCS CONCEPTS

• **Computer systems organization → Quantum computing**.

KEYWORDS

Quantum Computing, ZZ Crosstalk, Error Suppression

---

*Corresponding author

1 INTRODUCTION

Quantum computing (QC), in theory, can be used to solve some classically difficult or intractable problems [4, 58]. However, current quantum computers are susceptible to noise, which impedes their usage in many scenarios [47, 48]. Noise can corrupt qubits, degrade quantum gate fidelity, and lead to computational errors. Finding effective approaches to suppressing noise is therefore crucial for enabling near-term quantum computers to solve practical problems.

Superconducting qubits are one of the leading technologies for building quantum computers. However, devices based on this technology are affected by a destructive type of noise known as *ZZ* crosstalk [9, 18, 43]. This refers to an always-on $\sigma_z \otimes \sigma_z$ interaction between *qubits connected by couplings* (Figure 1), which originates from the interaction between the computational and non-computational energy levels of qubits [4, 18].
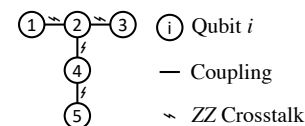
Figure 1: IBMQ Vigo superconducting device topology. Couplings between qubits mediate *ZZ* crosstalk.

ZZ crosstalk is problematic for QC in several ways. Firstly, its presence is often a limiting factor in quantum gate fidelity [46], leading to a ~5× increase in terms of the error rate of two-qubit gates [29, 37, 60]. Secondly, it is also an obstacle limiting the parallel execution of gates in a quantum circuit: when multiple gates are performed simultaneously, *ZZ* crosstalk can further deteriorate the fidelity of each gate by 2~10× [39, 43]. Additionally, *ZZ* crosstalk can cause correlated high-weight errors to occur with high probability [43, 56], which is difficult for quantum error correction (QEC) to deal with [2, 22, 40].

Previous approaches to suppressing ZZ crosstalk have mostly relied on sophisticated and specific chip design. One widely explored approach is to use tunable couplers [39, 52, 60, 65], but this introduces additional decoherence factors [33, 42], shortening the lifetime of qubits. Recently, two other approaches based on heterogeneous qubits [37, 53, 67] and multiple coupling paths [33, 46] have been proposed. However, these approaches suffer from common drawbacks of the extra complexity imposed on chip fabrication and control, and of the fact that none of them can be used on simpler devices with homogeneous qubits connected by a single untunable coupling (e.g., IBMQ devices [16, 61]) which are expected to have much longer coherence time [24, 61].

Superconducting quantum computers use microwave pulses to control qubits [4, 44, 45], and it is thus desirable to suppress ZZ crosstalk via careful manipulation of these pulses, instead of via specific chip design. Since ZZ crosstalk causes error correlation, to minimize its effect on the execution of quantum circuits, an ideal approach is to treat the whole circuit as a single unitary and optimize pulses globally [17, 34]. However, such kind of global optimization is limited to small circuits, as *its time and memory consumption grows exponentially with the number of qubits involved* [38, 57].

To overcome the above problems, we propose *co-optimizing pulses and scheduling*. As shown in Figure 2, we optimize pulses to implement quantum gates and offer an ability to suppress ZZ crosstalk *surrounding the gate*, and then we design effective scheduling strategies for exploiting this ability to achieve suppression *across the whole circuit*.

We manage to tackle ZZ crosstalk in a scalable way. (1) We carefully design the objective of pulse optimization so that we only need to optimize pulses *on small systems* (typically with ≤ 4 qubits), which has low time and memory consumption. (2) We leverage the duality between cuts and odd-vertex pairings of graphs and several heuristics to design a scheduler that can achieve effective suppression in time *polynomial* in the number of qubits and gates of quantum circuits.

We implement our approach as a general framework that can work with different pulse optimization methods (we have tested two existing methods and one proposed in this work). Simulation results show that we can improve the fidelity of QC with 4~12 qubits by up to 81× (11× on average), compared with the state-of-the-art pulse and scheduling policy used on current quantum computers. Ramsey experiments on a real QC device with three transmon qubits also demonstrate that our method can eliminate the effect of ZZ crosstalk to a great extent by reducing effective ZZ strength (defined in Section 7.4) from ~200 kHz to <11 kHz.

2 OVERVIEW AND CHALLENGES

The key idea of our approach is to co-optimize pulses and scheduling so as to suppress ZZ crosstalk in a scalable way. As shown in Figure 2, given a quantum circuit, our ZZ-aware scheduling first partitions it into multiple layers[1] and then supplements each layer with additional identity gates. Next, we translate each gate in the circuit to its corresponding pulses that are optimized under our
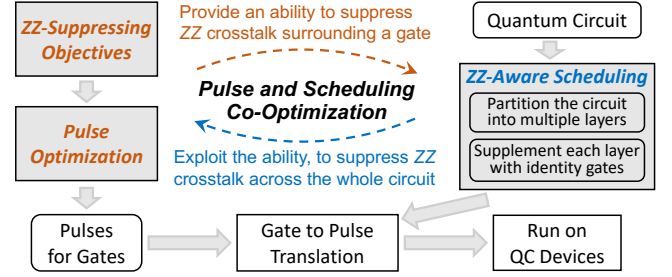
---

[1]A layer contains a set of gates to be executed simultaneously and different layers are executed serially.

Figure 2: Our pulse and scheduling co-optimization approach.

ZZ-suppressing objectives. In the following, we use an example to illustrate our idea.

2.1 A Motivating Example

Figure 3 shows an example where we aim to run a circuit (a) on a device with a 5×3 grid topology.

Optimizing pulses for quantum gates. For now, we consider executing all the 3 gates of (a) simultaneously as 1 layer (Figure 3 (b)). According to qubit status (whether a gate is applied to the qubit or not), we divide the device into two *regions* (marked by gray areas), where a region refers to a connected component composed of qubits with the same status. *We optimize pulses to implement quantum gates while also suppressing cross-region ZZ crosstalk surrounding the gates.* For example, the optimized pulses on qubits 7-10 implement gates $CNOT_{7,8}$, $H_{9,10}$ and suppress ZZ crosstalk on the 9 cross-region couplings (dashed edges) surrounding them.

However, there are still 13 *intra-region* couplings (solid edges), and these are the ones with unsuppressed ZZ crosstalk. We use two metrics to characterize them: #couplings with unsuppressed crosstalk (N_C) and #qubits in the largest region (N_Q). A larger N_C implies more crosstalk errors, and a larger N_Q can produce correlated errors of higher weight. [2] We propose two scheduling strategies to reduce these metrics.

Applying additional identity gates. The first strategy is to *apply identity gates to some of the qubits on which no gates have been performed.* Identity gates (pulses) do not alter the quantum state of qubits, but they can convert some intra-region crosstalk to cross-region crosstalk that can be suppressed by these identity pulses. Figure 3 (c) gives two plans. Take Plan A for example, it applies identity gates on qubits 1 and 11, converting unsuppressed intra-region crosstalk 1-2, 1-6, 11-6 and 11-12 to suppressed cross-region crosstalk.

Partitioning a circuit into multiple layers. Besides applying identity gates, suppression can be further enhanced if we *properly partition the circuit into 2 layers* instead of one, as shown in Figure 3 (d). Compared with (c), both N_Q and N_C of each layer are largely reduced, which indicates much better suppression. As an intuitive demonstration, all the ZZ crosstalk on the device is suppressed in Layer 2.

2.2 Challenges

We have addressed three main challenges for applying our approach in practice.

The first challenge is that pulse optimization requires time and memory exponential in the number of qubits involved.
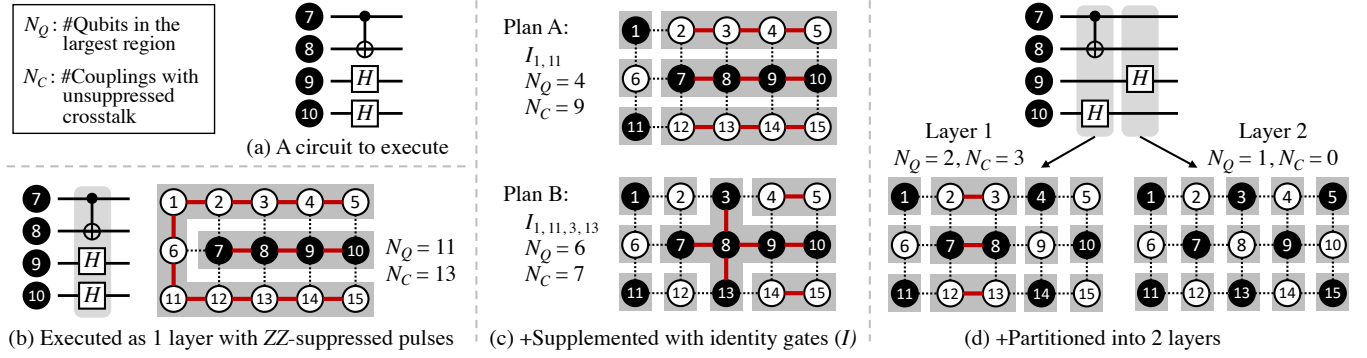
Figure 3: A motivating example. With the introduction of new strategies to (c) and (d), suppression is gradually enhanced (indicated by smaller N_Q and N_C). Black qubits have pulses applied to, while white ones do not. Gray areas of topology graphs represent regions. Crosstalk across regions (dashed edges) is suppressed but that within regions (solid edges) is not.

We address this by setting the objective of pulse optimization to suppressing only cross-region ZZ crosstalk. Section 4 elaborates on the objective and explains why we only need to optimize pulses on small systems (i.e., with only a few qubits) under this objective.

The second challenge is to find a plan to apply identity gates with the resulting N_Q and N_C minimized. It is challenging as the minimization of N_C alone can be reduced to a maximum cut problem (see Section 5) which is NP-complete for general graphs. Besides, Plan A ($N_Q = 4, N_C = 9$) and B ($N_Q = 6, N_C = 7$) in Figure 3 (c) also show a trade-off between N_Q and N_C, and thus consideration of N_Q further complicates the situation. In Section 5, we define an *optimal suppression* problem to characterize this trade-off. And based on the duality of cuts and odd-vertex pairings, we give an *efficient* solution for near-term QC devices with a planar topology.

The third challenge is to strike a balance between parallelism and crosstalk suppression when partitioning circuits. Though, as shown in Figure 3 (d), more layers can improve suppression, they can also decrease parallelism and hence increase the effect of decoherence. For high-fidelity QC, we need to balance these two factors. The challenge is that it is impractical to test all possible partitioning schemes and find an optimal one. We instead propose several heuristics which, according to our evaluations, give very good performance in practice. Section 6 describes these heuristics along with a complete scheduling algorithm.

3 PRELIMINARIES

3.1 Quantum Evolution & Quantum Control

QC is performed by using quantum gates to manipulate qubits, whose evolution is described by the Schrödinger equation

$$i\hbar |\dot{\psi}(t)\rangle = H(t) |\psi(t)\rangle$$

where $|\psi(t)\rangle$ is a state of the qubits, $|\dot{\psi}(t)\rangle$ is its derivative with respect to time t, and $H(t)$ is the Hamiltonian that completely determines the evolution. We discuss our approach based on the effective Hamiltonian below [38, 41]. Drive noise and leakage into higher energy levels will be considered in the evaluation.

In superconducting QC devices, quantum gates are implemented using pulses to control certain Hamiltonian terms [44]. For the two-qubit system in Figure 4, $\sigma_z^{(1)} \otimes \sigma_z^{(2)}$ describes ZZ crosstalk, λ

is its strength, and $\Omega_{x,y}^{(q)}(t)$ and $\Omega^{(1\text{-}2)}(t)$ are control pulses. Single-qubit gates on qubit q can be achieved by controlling the $\sigma_x^{(q)}$ and $\sigma_y^{(q)}$ terms via $\Omega_{x,y}^{(q)}(t)$. Two-qubit gates can be achieved by tuning $\Omega^{(1\text{-}2)}(t)$ for $H_{Coupling}$. The concrete form of $H_{Coupling}$ depends on the specific device and gate type. For example, $\sigma_z \otimes \sigma_x$ is used to implement CNOT gate [15, 55] and $\sigma_x \otimes \sigma_x + \sigma_y \otimes \sigma_y$ is used for iSWAP gate [4, 10].

$$H(t) = \lambda \sigma_z^{(1)} \otimes \sigma_z^{(2)}$$
$$+ \sum_{q \in \{1,2\}} \left[\Omega_x^{(q)}(t)\sigma_x^{(q)} + \Omega_y^{(q)}(t)\sigma_y^{(q)} \right]$$
$$+ \Omega^{(1\text{-}2)}(t) H_{Coupling}$$

Figure 4: A two-qubit system and its (effective) Hamiltonian.

3.2 Graphs, Cuts, and Odd-Vertex Pairings

The topology of a QC device can be represented by a graph $G = (V, E)$, where V is a vertex set corresponding to qubits, and E is an edge set corresponding to couplings. A planar topology corresponds to a planar graph. Every planar graph G has a *dual graph* G^*, where each vertex of G^* corresponds to a face of G (including the outer face), and each edge e^* of G^* corresponds to an edge e of G (see Figure 5 (a)).
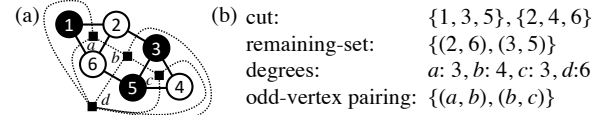
Figure 5: An example of a cut in a planar graph. (The dual graph is drawn by square vertices and dashed edges.)

A *cut* $C = (S, T)$ is a partition of V into two disjoint sets S, T. After removing the edges across the cut, one is left with the *remaining-set* R_C that includes all edges whose endpoints belong to the same partition $R_C = \{(u, v) | u, v \in S \text{ or } u, v \in T\}$ (see Figure 5 (b)).

An edge set whose contraction leaves the remaining graph free of odd-degree vertices is called an *odd-vertex pairing*. (Contracting an edge refers to removing it from a graph and merging the two vertices that it previously joined.) All graphs have an even number of odd-degree vertices, and an odd-vertex pairing covers all the odd-degree vertices. For example, the odd-degree vertices of the

Lei Xie, Jidong Zhai, ZhenXing Zhang, Jonathan Allcock, Shengyu Zhang, and Yi-Cong Zheng

dual graph in Figure 5 are a, c. After contracting the edges $D^* = \{(a, b), (b, c)\}$, G^* will have only two even-degree vertices (a, b, c are merged into one), and thus D^* is an odd-vertex pairing. The following theorem establishes a connection between cuts and odd-vertex pairings.

Theorem 3.1 *An edge set D contains a remaining-set R_C for a cut C of a graph G if and only if the dual edge set D^* of D in the dual graph G^* is an odd-vertex pairing of G^*.*

Refer to [8, 28] for a proof. One can thus find a cut of a graph $G = (V, E)$ as follows. First, find an odd-vertex pairing D^* in the dual graph G^* (e.g., $D^* = \{(a, b), (b, c)\}$). After contracting from G the dual edges D of D^* (e.g., $D = \{(2, 6), (3, 5)\}$), all the remaining edges $E - D$ are across a cut. We can then find the cut by coloring the vertices in the remaining graph using two colors. We will use this method in our scheduling algorithm.

4 ZZ-SUPPRESSING OBJECTIVES FOR PULSE OPTIMIZATION

We elaborate on the objective of pulse optimization in this section: to implement quantum gates and also suppress cross-region ZZ crosstalk surrounding the gates. The key result is that we only need to optimize pulses on small systems when taking cross-region crosstalk as the suppression target.

We first discuss basic regions with only one single-qubit or two-qubit gate, and then consider general regions consisting of multiple gates.
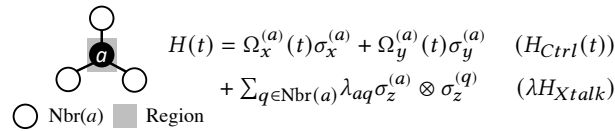
4.1 Basic Region: One Single-Qubit Gate

$$H(t) = \Omega_x^{(a)}(t)\sigma_x^{(a)} + \Omega_y^{(a)}(t)\sigma_y^{(a)} \quad (H_{Ctrl}(t))$$
$$+ \sum_{q\in\text{Nbr}(a)} \lambda_{aq}\sigma_z^{(a)} \otimes \sigma_z^{(q)} \quad (\lambda H_{Xtalk})$$

Figure 6: A basic region with one single-qubit gate U_1 on qubit a and its Hamiltonian.

Figure 6 shows a basic region with one single-qubit gate U_1 on qubit a and the corresponding Hamiltonian $H(t)$. Nbr(a) is a set of all the qubits that are neighbors of a and outside the region, and the other notations are defined in Section 3.1. Using $\lambda = \sum_{q\in\text{Nbr}(a)} \lambda_{aq}$ to denote the total effect of all the cross-region crosstalk, the Hamiltonian can be rewritten as

$$H(t) = H_{Ctrl}(t) + \lambda H_{Xtalk}$$

where $H_{Ctrl}(t) = \Omega_x^{(a)}(t)\sigma_x^{(a)} + \Omega_y^{(a)}(t)\sigma_y^{(a)}$, and $H_{Xtalk} = \sigma_z^{(a)} \otimes \left(\sum_{q\in\text{Nbr}(a)} \lambda_{aq}\sigma_z^{(q)}/\lambda\right)$.

We aim to optimize pulses $\Omega_{x,y}^{(a)}(t)$ to (i) implement U_1 on a and (ii) suppress the effect of the cross-region crosstalk H_{Xtalk} surrounding a.

The implementation of U_1 is achieved by letting $U_{Ctrl}(T) = U_1$ where $U_{Ctrl}(t)$ is defined by $i\hbar\dot{U}_{Ctrl}(t) = H_{Ctrl}(t)U_{Ctrl}(t)$ and T is the duration of the pulses. In other words, the target gate should be implemented when all the crosstalk is suppressed.

To suppress H_{Xtalk}, we maximize the similarity between the actual evolution $U(t)$ of the system, defined by $i\hbar\dot{U}(t) = H(t)U(t)$,

and the desired evolution $U_1 \otimes \mathbf{I}^{\text{Nbr}(a)}$, where $\mathbf{I}^{\text{Nbr}(a)}$ represents identity operations on Nbr(a).

To summarize, the optimization objective is

$$\begin{aligned} \underset{\{\Omega(t)\}}{\text{maximize}} \quad & F\left(U(T), U_1 \otimes \mathbf{I}^{\text{Nbr}(a)}\right) \\ \text{subject to} \quad & U_{Ctrl}(T) = U_1 \end{aligned} \quad (4.1)$$

where $\{\Omega(t)\} = \{\Omega_{x,y}^{(a)}(t)\}$ and F is a measure of similarity.
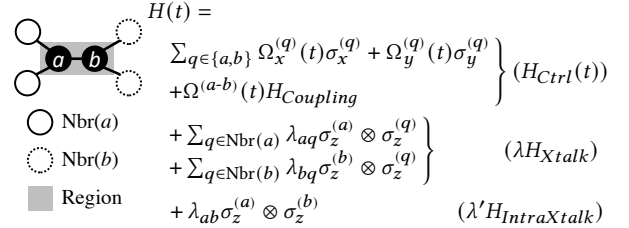
4.2 Basic Region: One Two-Qubit Gate

$$H(t) =$$
$$\left.\begin{aligned}&\sum_{q\in\{a,b\}} \Omega_x^{(q)}(t)\sigma_x^{(q)} + \Omega_y^{(q)}(t)\sigma_y^{(q)}\\&+\Omega^{(a\text{-}b)}(t)H_{Coupling}\end{aligned}\right\} (H_{Ctrl}(t))$$
$$\left.\begin{aligned}&+ \sum_{q\in\text{Nbr}(a)} \lambda_{aq}\sigma_z^{(a)} \otimes \sigma_z^{(q)}\\&+ \sum_{q\in\text{Nbr}(b)} \lambda_{bq}\sigma_z^{(b)} \otimes \sigma_z^{(q)}\end{aligned}\right\} (\lambda H_{Xtalk})$$
$$+ \lambda_{ab}\sigma_z^{(a)} \otimes \sigma_z^{(b)} \quad (\lambda' H_{IntraXtalk})$$

Figure 7: A basic region with one two-qubit gate U_2 on qubits a, b and its Hamiltonian.

Figure 7 shows a basic region with one two-qubit gate U_2 on qubits a, b and its Hamiltonian. The Hamiltonian can be rewritten as

$$H(t) = H_{Ctrl}(t) + \lambda H_{Xtalk} + \lambda' H_{IntraXtalk}$$

where H_{Xtalk} denotes crosstalk across regions, and $H_{IntraXtalk}$ denotes crosstalk within the region.

We aim to optimize pulses to (i) implement U_2 on a, b and (ii) suppress the effect of the cross-region crosstalk H_{Xtalk} surrounding a, b. Similarly, the optimization objective is

$$\begin{aligned} \underset{\{\Omega(t)\}}{\text{maximize}} \quad & F\left(U(T), \tilde{U}_2(T) \otimes \mathbf{I}^{\text{Nbr}(a)} \otimes \mathbf{I}^{\text{Nbr}(b)}\right) \\ \text{subject to} \quad & U_{Ctrl}(T) = U_2 \end{aligned} \quad (4.2)$$

where $\{\Omega(t)\} = \{\Omega_{x,y}^{(a)}(t), \Omega_{x,y}^{(b)}(t), \Omega^{(a\text{-}b)}(t)\}$. The major difference from the basic region with one single-qubit gate is that the desired evolution on a, b is no longer the ideal gate U_2 but $\tilde{U}_2(T)$ defined by $i\hbar\dot{\tilde{U}}_2(t) = [H_{Ctrl}(t) + \lambda' H_{IntraXtalk}] \tilde{U}_2(t)$, since we allow intra-region crosstalk when optimizing pulses.

4.3 General Regions

For a general region composed of multiple single-qubit and two-qubit gates, we do *not* need to optimize again, but can *directly* apply the pulses optimized in basic regions with the same target gates. In this way, all cross-region ZZ crosstalk for that general region can be suppressed simultaneously.

For instance, consider pulses optimized in two basic regions, one with only a single-qubit gate U_1 and the other with a two-qubit gate U_2. Then, for the general region in Figure 8, by directly applying these optimized pulses, the pulse for U_1 on qubit c can suppress c-4 and c-5, and the pulse for U_2 on qubits a, b can suppress a-1, a-2, b-3, b-5. Thus, the application of these two pulses suppresses all of the crosstalk across regions. One can find out, in a similar manner, that this result holds for arbitrary general regions. The behind reason is that each cross-region coupling is connected to
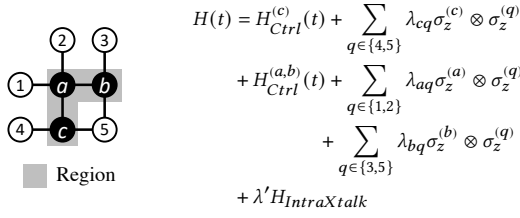
$$H(t) = H_{Ctrl}^{(c)}(t) + \sum_{q \in \{4,5\}} \lambda_{cq} \sigma_z^{(c)} \otimes \sigma_z^{(q)}$$

$$+ H_{Ctrl}^{(a,b)}(t) + \sum_{q \in \{1,2\}} \lambda_{aq} \sigma_z^{(a)} \otimes \sigma_z^{(q)}$$

$$+ \sum_{q \in \{3,5\}} \lambda_{bq} \sigma_z^{(b)} \otimes \sigma_z^{(q)}$$

$$+ \lambda' H_{IntraXtalk}$$

Figure 8: A general region with a two-qubit gate U_2 on qubits a, b and a single-qubit gate U_1 on qubit c.

one qubit with a gate performed on, and we can suppress crosstalk on that coupling by the pulses for that gate.

The above result implies *high scalability* in that we only need to optimize pulses *in basic regions* (i.e., small systems), which avoids the high time and memory consumption of optimization in general regions.

In Section 7, we will give three approaches to implement the optimization objectives for basic regions: quantum optimal control, dynamic corrected gates and a new proposed one based on quantum perturbative theory.

5 OPTIMAL SUPPRESSION

In this section we show how a layer of gates can be executed, assisted by extra identity gates, such that ZZ crosstalk on the whole device is maximally suppressed.

The key idea is to establish a connection between the status of qubits and a cut of the device topology. A qubit's status is either (i) with a gate (pulse) applied to it or (ii) not. The qubits with gates acting on them form a set S, the others constitute a set T, and these sets form a cut $C = (S, T)$ of the topology. Given a layer L of gates, the aim is to find such a cut (S, T) that (i) the qubits acted on by gates in L all lie in S, and (ii) any remaining qubits in S are acted on by identity gates to further improve crosstalk suppression.

Before discussing the general case where the layer contains multiple gates, we first consider a special case where it contains no gates, and one simply wishes to suppress crosstalk by applying identity gates to all the qubits in S.

5.1 Layers Containing No Gates

Ideally, all the crosstalk on a device can be suppressed by applying identity pulses to certain qubits. We refer to this as *complete suppression*. It can be easily proved that complete suppression can be achieved on any device of a bipartite topology (e.g., Figure 9). Fortunately, most near-term QC devices [30, 48] and topological QEC codes [11, 22] indeed have a bipartite topology.
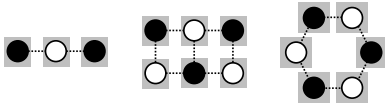

Figure 9: Examples of complete suppression. All crosstalk is suppressed by applying identity pulses to black qubits.

For devices that do not admit complete suppression, there is at least one region with unsuppressed internal crosstalk. In this case, two metrics are important: #qubits in the largest region (N_Q) and #couplings with unsuppressed crosstalk (N_C) corresponding
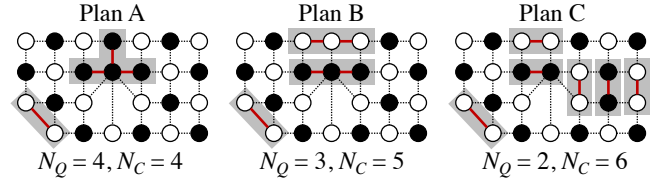

Figure 10: The trade-off between N_Q and N_C.

to all the edges within regions. There is a trade-off between the two metrics, as can be seen in Figure 10. To characterize this trade-off, we define optimal suppression or, more precisely, *α-optimal suppression*.

Definition 5.1 (α-Optimal Suppression) Given $\alpha > 0$, a device has an α-optimal suppression plan if applying pulses to some of its qubits divides its qubits into multiple regions such that $\alpha N_Q + N_C$ is minimized.

To find an α-optimal suppression plan, we convert the problem to a cut problem of topology graphs.

A Cut Problem for α-Optimal Suppression. Given a graph $G = (V, E)$ and $\alpha > 0$, find a cut $C = (S, T)$ of G that minimizes $\alpha N_Q + N_C$, where N_Q is #vertices in the largest connected component of the graph $G' = (V, R_C)$, $N_C = |R_C|$, and R_C is the remaining-set of the cut C.

The two partitions S, T represent qubits with and without pulses applied to respectively. The remaining-set R_C represents couplings with unsuppressed crosstalk since the endpoints of any edge in R_C belong to the same partition. After giving a solution to the cut problem, α-optimal suppression can be achieved by applying identity pulses to the partition S. In Figure 10, the black and white vertices correspond to the partitions S, T respectively, solid edges correspond to the remaining-set R_C, and crosstalk on the solid edges is unsuppressed.

The cut problem is difficult to solve. For example, for $\alpha = 0$, we need to find a cut that has a minimum remaining-set R_C, i.e., a maximum number of cross-cut edges. This is the maximum cut problem, and is NP-complete for general graphs. Noting that the topology of most near-term QC devices [30, 48] and topological QEC codes [11, 22] is planar, we therefore aim to design an efficient algorithm for planar topologies.

The main idea of our algorithm is as follows. Based on Theorem 3.1, we can find cuts by locating odd-vertex pairings in dual graphs. Since an odd-vertex pairing covers all odd-degree vertices, the smallest one contains only simple paths, with each path connecting two odd-degree vertices [28]. By selecting shortest paths, one can minimize the size of an odd-vertex pairing. As a consequence of Theorem 3.1, $N_C = |R_C|$ can also be minimized. To account for N_Q, we consider the top-k shortest paths, and select the path which strikes the desired balance between N_C and N_Q.

Specially, our algorithm consists of three steps (see Figure 11 for an example):

Step 1: Vertex Matching. This step decides which two vertices each path should connect. First, we construct a complete graph containing all the odd-degree vertices of the dual graph. Then, a weight $L - d(u, v)$ is assigned to each edge (u, v), where $d(u, v)$ is the shortest path length between u, v, and $L = 1 + \max_{u,v} d(u, v)$. Finally, the vertices on the complete graph are matched by a maximum
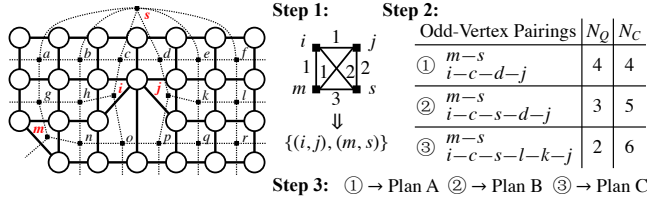
Figure 11: Left: The topology graph (circular vertices, solid edges) of Figure 10 and its dual (square vertices, dashed edges, all edges on the border are connected to vertex s). Right: Steps for running our algorithm on the left topology.

weight matching algorithm [23]. In this way, connecting each pair of matched vertices in the dual graph by shortest paths can give a smallest odd-vertex pairing [28]. In Figure 11, we construct a complete graph of odd-degree vertices $\{i, j, m, s\}$, whose maximum weight matching is $\{(i, j), (m, s)\}$.

Step 2: Path Relaxing. This step relaxes the shortest path restriction, to enable optimizing over both N_Q and N_C. We generate many odd-vertex pairings by connecting each pair of matched vertices with their top-k shortest paths, and then select the one with minimum $\alpha N_Q + N_C$. In more detail, we use a greedy strategy as shown in Algorithm 1, Line 7-21. In Figure 11, we generate three samples corresponding to the shortest path between m, s and top-3 shortest paths between i, j. We select one according to α.

Step 3: Cut Inducing. This step obtains a cut for the selected odd-vertex pairing. Based on Theorem 3.1, after contracting the dual edges of the selected odd-vertex pairing in the topology graph, we obtain a cut by performing breadth-first search to color the vertices of the remaining topology graph using 2 colors. In Figure 11, the odd-vertex pairings ①, ②, ③ obtained from step 2 correspond to Plans A, B, C in Figure 10 respectively.

5.2 Layers Containing Gates

This case can be regarded as a constrained problem that requires all the qubits involved in the gates of a layer belong to the same partition, because we need to apply pulses to all of them. We solve this case by converting it to the unconstrained problem (layers containing no gates). The conversion is based on a property of remaining-sets.

Theorem 5.1 *The remaining-set of a cut comprises multiple connected components, with vertices in the same connected component belonging to the same partition of the cut.*

For example, gray areas in Figure 10 identify connected components, and the vertices in the same gray area have the same color, i.e., they belong to the same partition.

Based on this theorem, denoting the set of qubits involved in the gates of a layer by Q and the edges involved $\{(u, v)|(u, v) \in E$ and $u, v \in Q\}$ by E_Q, if we explicitly add E_Q to a remaining-set, then the vertices of Q in the same connected component of graph (Q, E_Q) necessarily belong to the same partition of the induced cut. Then we only need to check whether the vertices in different connected components belong to the same partition. Specifically, we perform the conversion in the dual graph in three steps. We denote the dual of E_Q by E_Q^*.

Algorithm 1: α-Optimal Suppression Algorithm

Input: Topology graph $G = (V, E)$; Qubits involved in gates Q; Relative importance coefficient α
Output: A cut for α-optimal suppression

1   $G^* \leftarrow$ the dual graph of G
2   $E_Q^* \leftarrow \{(u, v)^* | (u, v) \in E$ and $u, v \in Q\}$
3   **Delete Edges:** $G^* \leftarrow G^* - E_Q^*$
4   **Vertex Pairing:**
5      $G_{odd} \leftarrow$ a complete graph with the odd-degree vertices of G^* and edge weights $w(u, v) = L - d(u, v)$
6      $M \leftarrow$ a maximum matching of G_{odd}
7   **Path Relaxing:**
8   **foreach** pair of matched vertices $(u, v) \in M$ **do**
9          $L^{(u,v)} \leftarrow [L_1^{(u,v)}, L_2^{(u,v)}, \cdots]$ a list of top-k shortest paths between (u, v) sorted by their length in ascending order
10   Odd-vertex pairing $P \leftarrow \{L_1^{(u,v)} | (u, v) \in M\}$
11   **repeat**
12        $candidates \leftarrow \varnothing$
13   **foreach** $(u, v) \in M$ **do**
14           $P' \leftarrow P - L_i^{(u,v)} + L_{i+1}^{(u,v)}$ // relax
            $L_i^{(u,v)} = P \cap L^{(u,v)}$
15   **Cut Inducing:**
16   **Add Edges:** $P' \leftarrow P' + E_Q^*$
17             $G' \leftarrow$
               G with the dual edges of P' contracted
18   Obtain a cut C by coloring the vertices of G' with 2 colors
19   **Check: if** $Q \subset$ a partition of C **then** add P' to $candidates$
20        $P \leftarrow$ the candidate of $candidates$ with minimum $\alpha N_Q + N_C$
21   **until** $\alpha N_Q + N_C$ of P is unchanged
22 **return** a cut C induced from P

Step 1: Delete Edges. Delete E_Q^* from the dual graph. Then preform Vertex Matching and Path Relaxing on the remaining dual graph.

Step 2: Add Edges. Add E_Q^* back to the odd-vertex pairing obtained from Path Relaxing.

Step 3: Check. Perform Cut Inducing according to the modified odd-vertex pairing, and check whether all the vertices of Q belong to the same partition of the induced cut.

For the example in Figure 12, $Q = \{1, 3, 5\}$ and $E_Q = \{(1, 5)^*\}$, graph (Q, E_Q) has two connected components. We first delete the edge $(1, 5)^*$ from the dual graph. After obtaining odd-vertex pairings, we add $(1, 5)^*$, induce a cut and check the constraint. The example shows two odd-vertex pairings, one passes the check while the other does not.

Time Complexity Analysis. We summarize the complete process in Algorithm 1. We use the blossom algorithm [23] for maximum weight matching in $O(n_o^3)$ time, where n_o is #odd-degree vertices. We use Yen's algorithm [66] to generate top-k shortest paths in
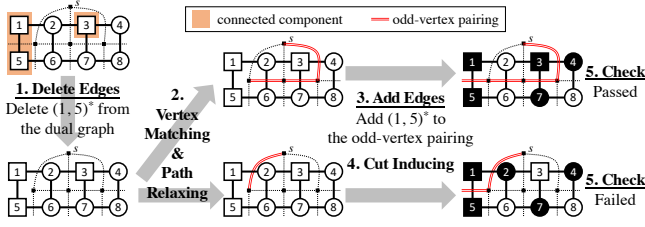
Figure 12: Steps for running our algorithm to perform gates on qubits 1, 3 and 5. (All edges on the border are connected to vertex s.)

$O(kn_d(m_d + n_d \log n_d))$ time, where n_d is #vertices in the dual graph and m_d is #edges. For path relaxing, our greedy strategy takes $O(kn_o/2)$ iterations, since each path can be used in Line 20 at most once. In each iteration, our strategy checks $O(n_o/2)$ paths (Line 13). For cut inducing, breadth-first search takes $O(n_t + m_t)$ time, where n_t is #vertices in the topology graph and m_t is #edges. In total, our algorithm takes $O(kn^3 \log n)$ time, where $n = \max\{n_t, m_t, n_d\}$.

6 ZZ-AWARE SCHEDULING

We present a complete scheduling algorithm in this section, with the focus on how to partition a circuit into multiple layers so as to strike a balance between crosstalk suppression and parallelism.

Our scheduling algorithm is given in Algorithm 2. The main idea is to iteratively schedule the gates in schedulable gate sets[2] by making crosstalk suppression the first priority and then maximizing parallelism. Given a schedulable gate set, two cases can arise: (1) it contains only single-qubit gates, or (2) it contains two-qubit gates. In the first case, we propose a strategy that can always achieve complete suppression for bipartite topologies. In the second case, we propose a heuristic based on the distance between two-qubit gates.
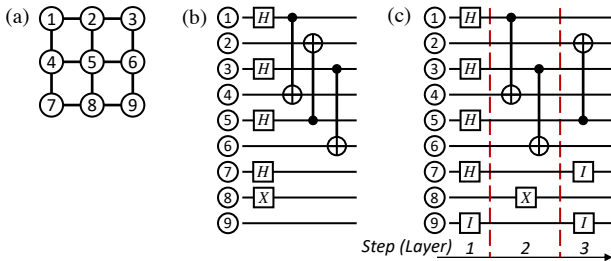

Figure 13: An example for scheduling the quantum circuit (b) on the device (a). Our scheduling plan is shown in (c). (Assume the same duration for all the gates, different durations will be considered in the evaluation.)

Figure 13 shows an example. When no gates have been scheduled, the schedulable gate set is $\{H_{1,3,5,7}, X_8\}$ (Case 1). After scheduling layer 1, the set will be $\{CNOT_{1,4}, CNOT_{5,2}, CNOT_{3,6}, X_8\}$ (Case 2).
Case 1. Only single-qubit gates. The strategy for this case is based on an observation about bipartite topologies.

Observation: Complete suppression can be achieved on bipartite topologies with single-qubit gates. (This is illustrated by Figure 9 in Section 5.)

---

[2]A gate is schedulable if all of its preceding gates have been scheduled, and a schedulable gate set consists of all currently schedulable gates.

Algorithm 2: Schedule a Quantum Circuit

Input: Topology graph G; Quantum circuit QC;
　　　　Suppression requirement R; α for the α-optimal
　　　　suppression algorithm

Output: A scheduling plan: layers of simultaneous gates

1 **while** there exists unscheduled gates of QC **do**
2 　　$SG \leftarrow$ a set of all schedulable gates
3 　　**if** SG has only single-qubit gates **then**　　　// Case 1
4 　　　　Cut $(S, T) \leftarrow \alpha$-optimal suppression with topology
　　　　　　G and qubits $\varnothing$ (S has more qubits involved in SG
　　　　　　than T)
5 　　**else**　　　　// Case 2: SG contains two-qubit gates
6 　　　　$SG_2 \leftarrow$ two-qubit gates of SG
7 　　　　Partition $S \leftarrow$ TwoQSchedule(G, SG_2, R)
8 　　Schedule(S, SG)
9 **Procedure** Schedule(Partition S, Gates SG)
10 　　$Q \leftarrow$ qubits involved in gates SG
11 　　Layer $L = \{$gate $g \in SG \mid$ all qubits for gate $g \in S\}$
12 　　**foreach** qubit $q \in S - Q$ **do** // supplement L with identity
　　　　gates
13 　　　　$L = L \cup \{$an identity gate on $q\}$
14 　　**yield** L
15 **Procedure** TwoQSchedule(Topology G, Two-qubit gates
　　SG_2, Suppression requirement R)
16 　　Cut $C = (S, T) \leftarrow \alpha$-optimal suppression with topology
　　　　G and qubits involved in SG_2 (S contains qubits
　　　　involved in SG_2)
17 　　**if** C satisfies R **then return** S// schedule simultaneously
18 　　**else**　　　　// heuristic selection according to distance
19 　　　　$a, b \leftarrow$ two gates of SG_2 with the minimum distance
20 　　　　Groups $A \leftarrow \{a\}, B \leftarrow \{b\}$
21 　　　　**while** SG_2 is not empty **do**
22 　　　　　　$(M_g, M_G) \leftarrow$ the gate (of SG_2) and group (A or
　　　　　　　　B) with the maximum distance
23 　　　　　　Cut $C \leftarrow \alpha$-optimal suppression with topology
　　　　　　　　G and qubits involved in $M_G + M_g$
24 　　　　　　**if** C satisfies R **then** $M_G += M_g, SG_2 -= M_g$
25 　　　　　　**else break**
26 　　　　**if** $|A| > |B|$ **then** $M \leftarrow A$ **else** $M \leftarrow B$
27 　　　　Cut $(S, T) \leftarrow \alpha$-optimal suppression with topology
　　　　　　G and qubits involved in M (S contains qubits
　　　　　　involved in M)
28 　　**return** S

Strategy: We thus first perform the α-optimal suppression algorithm for layers containing no gates (Line 4), which, for bipartite topologies, gives the cut (partitions of qubits) that can admit complete suppression (Figure 14 (a)). According to the qubits they act on, the schedulable gates are also divided into two partitions (Figure 14 (b)). Then, we schedule the partition that has more gates to maximize the parallelism, i.e., grouping these gates as a layer and supplementing it with additional identity gates (see Procedure Schedule). In Figure 14, we choose the gates on black qubits to
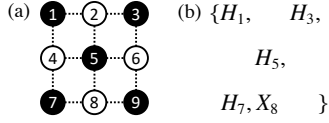
(a) (b) $\{H_1, \quad H_3,$

$H_5,$

$H_7, X_8 \quad \}$

Figure 14: (a) The suppression plan given by the optimal suppression algorithm. (b) The schedulable gates for Layer 1.

schedule, with an identity gate on qubit 9. The remaining gates (e.g., X_8) will be scheduled in the following layers.

Case 2. The schedulable gate set contains two-qubit gates. The strategy for this case is based on an observation about the distance between two-qubit gates.

Definition 6.1 The distance between two-qubit gates a on qubits a_1, a_2 and b on qubits b_1, b_2 is defined by $D(a, b) = \sum_{i,j} d(a_i, b_j)$ where $d(a_i, b_j)$ is the length of the shortest path between qubits a_i and b_j.

Definition 6.2 The distance between a two-qubit gate a and a group G of two-qubit gates is defined by $D(a, G) = \min_{g \in G} D(a, g)$.

Observation: Executing two-qubit gates with a shorter distance usually worsens suppression.

Figure 15 compares the suppression performance and the distances of different ways to execute the CNOT gates in Figure 13. Executing $CNOT_{1,4}$ and $CNOT_{3,6}$ in parallel has a better suppression performance than the other ways, and also corresponds to a larger distance between gates.
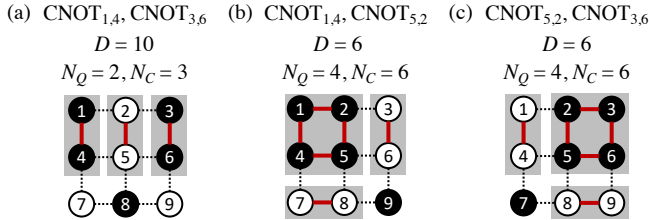
(a) $CNOT_{1,4}, CNOT_{3,6}$ (b) $CNOT_{1,4}, CNOT_{5,2}$ (c) $CNOT_{5,2}, CNOT_{3,6}$

$D = 10$ $\qquad D = 6 \qquad D = 6$

$N_Q = 2, N_C = 3 \quad N_Q = 4, N_C = 6 \quad N_Q = 4, N_C = 6$

Figure 15: Suppression performance (N_Q and N_C, lower is better) and distances (D) of different ways to execute two CNOT gates simultaneously.

Strategy: The main idea is to iteratively select the next farthest gate so that we can schedule as many gates as possible to maximize parallelism before a suppression requirement is violated (Procedure TwoQSchedule). Here we define the suppression requirement by some constraints on N_Q and N_C (e.g., requiring them to be less than certain thresholds), and violating the requirement means the suppression performance is too bad to be accepted.

Specifically, we first separate two closest gates into different groups (Line 19, 20) and then fill the two groups with the next farthest gate iteratively (Line 21-25). For the 3 CNOT gates in Figure 13, we first separate the two closest gates $CNOT_{1,4}$ and $CNOT_{5,2}$ into two groups $A = \{CNOT_{1,4}\}$, $B = \{CNOT_{5,2}\}$, then for $g = CNOT_{3,6}$, since $D(g, A) = 10 > D(g, B) = 6$, we add $CNOT_{3,6}$ to group A. Finally, we get $A = \{CNOT_{1,4}, CNOT_{3,6}\}$, $B = \{CNOT_{5,2}\}$. After obtaining the final groups, we schedule the group that has more gates (Line 26) and leave the other to schedule later.

The following theorem ensures that this strategy is effective.

Theorem 6.1 *For a set of simultaneous two-qubit gates, perform TwoQSchedule multiple times until all the gates are scheduled. If*

they are scheduled into K layers, then the top-K closest gates in this set must belong to different layers.

Proof. Since scheduling once separates two closest gates, which will be executed in different layers, scheduling K times leaves top-K closest gates in different layers. □

Thus, when suppression performance is measured by distance, a decrease in parallelism necessarily leads to an improvement in suppression performance. This is because, for every additional layer, two more closest gates are guaranteed to be executed at different times.

7 EVALUATION

We implement our approach as a general framework to work with different pulse optimization methods. In this section, we first give three methods that can achieve the ZZ-suppressing objectives defined in Section 4, and then present their suppression performance. Next, we integrate them into our framework and evaluate our approach with QC benchmarks. Finally, we present the results of the Ramsey experiments performed on a real quantum computer.

7.1 Methodology

7.1.1 Pulse Optimization Methods.

1. OptCtrl: With average gate fidelity F_g [50] as the measure of similarity, the ZZ-suppressing objective 4.1 can be expressed as minimizing a loss function:

$$L = -F_g \left(U(T), U_1 \otimes \mathbf{I}^{\text{Nbr}(a)} \right) - w F_g \left(U_{Ctrl}(T), U_1 \right)$$

where the first F_g term is to suppress cross-region ZZ crosstalk, the second is to implement U_1, and w is a weighting factor. The objective 4.2 can be converted similarly.

This is the typical setting of quantum optimal control [6, 17]. To suppress a range of ZZ crosstalk strengths, we average the loss function values obtained at many different strengths.

2. Pert: Optimizing the fidelity F_g cannot reflect the order of ZZ strength up to which crosstalk is suppressed. For direct and effective suppression, we propose an approach based on quantum perturbative theory.

The evolution according to $H(t) = H_{Ctrl}(t) + \lambda H_{Xtalk}$ can be written as $U(t) = U_{Ctrl}(t)U_{Xtalk}(t)$, where $U_{Xtalk}(t)$ can be expanded perturbatively according to the strength λ:

$$U_{Xtalk}(t) = I + \lambda U_{Xtalk}^{(1)}(t) + O(\lambda^2)$$

$$U_{Xtalk}^{(1)}(t) = -\frac{i}{\hbar} \int_0^t U_{Ctrl}^{\dagger}(t') \cdot H_{Xtalk} \cdot U_{Ctrl}(t') dt'$$

Thus, when $U_{Xtalk}^{(1)}(T) = \mathbf{0}$, $U(T)$ will differ from $U_{Ctrl}(T)$ by $O(\lambda^2)$, i.e., the effect of the first order of ZZ crosstalk is cancelled out. We achieve this by minimizing a loss function

$$L = \left\| U_{Xtalk}^{(1)}(T) \right\| - w F_g \left(U_{Ctrl}(T), U \right)$$

where U is the target gate to be achieved.

Both the loss functions of OptCtrl and Pert can be solved with gradient-based methods numerically [21] or analytically [38]. We use the numerical method and set $T = 20$ns.

3. DCG: Dynamic corrected gates (DCG) [36] are a generalization of dynamic decoupling [63, 64]. Different from the above two

approaches that optimize pulses from scratch, DCG constructs a sequence of pulses with existing ones (e.g., 20ns Gaussian pulses). The disadvantage of DCG is the long duration of the whole sequence (e.g., 120ns for $R_x(\pi/2)$ and 40ns for I).

7.1.2 Native Gates. We compile quantum circuits to the same native gates as IBMQ devices [31], $\{R_z(\theta), R_x(\pi/2), R_{zx}(\pi/2)\}$, where $R_z(\theta)$ is virtual Z gate implemented by software, not by pulses [44], and $R_{zx}(\pi/2)$ is used to implement CNOT gate [15]. To suppress more crosstalk, we also optimize an identity gate $I = R_x(2\pi)$.

7.1.3 Implementation. We implement our approach in Python. Simulations are performed at the Hamiltonian level with QuTiP [32], and Ramsey experiments are conducted on a real QC device [68] with three transmon qubits.

7.2 ZZ Crosstalk Suppression Performance

We show in this section that pulses optimized under our ZZ-suppressing objectives can effectively suppress cross-region ZZ crosstalk, even with control noise and leakage errors of qubits. We use Gaussian pulses as a reference since (i) they are representative of practical systems [16, 31, 45] and (ii) they are not optimized for ZZ crosstalk.

7.2.1 Single-Qubit Gates. For a two-qubit system ❶-②, with crosstalk 1-2 fully suppressed, the evolution of the whole system would be $U \otimes I$ when applying a gate U to qubit 1. Therefore, the infidelity between $U \otimes I$ and the actual observed evolution characterizes suppression performance.
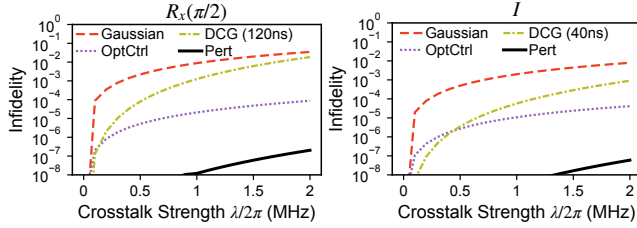
Figure 16: ZZ crosstalk suppression performance of $R_x(\pi/2)$ and I pulses. Lower is better. Precision is truncated to 10^{-8}.

Suppression Performance. Figure 16 compares different pulses for native gates $R_x(\pi/2)$ and I. It is shown that all the optimized pulses achieve lower infidelity than Gaussian pulses. Since OptCtrl suppresses ZZ indirectly via the observed fidelity, and the long DCG sequence accumulates more crosstalk errors than the others, they perform worse than Pert. However, as we will demonstrate later, for the typical ZZ crosstalk strengths observed on real devices ($\lambda/2\pi \approx 200$kHz [3, 5, 15, 29, 60]), OptCtrl is sufficient for our approach to achieve effective suppression of ZZ crosstalk for the whole quantum circuit.

Drive Noise. Frequency detuning (shift) of carrier waves and amplitude fluctuation are two typical kinds of drive noise [12, 38]. We show the results of the Pert $R_x(\pi/2)$ pulse in Figure 17. It is shown that, under drive noise of typical strengths in practice (detuning < 0.1 MHz, amplitude fluctuation < 0.1% [6]), ZZ can still be effectively suppressed.

Leakage Errors. Superconducting qubits can suffer from leakage errors due to the existence of higher energy levels, an issue that can be mitigated using DRAG [25]. DRAG is a protocol that can modify
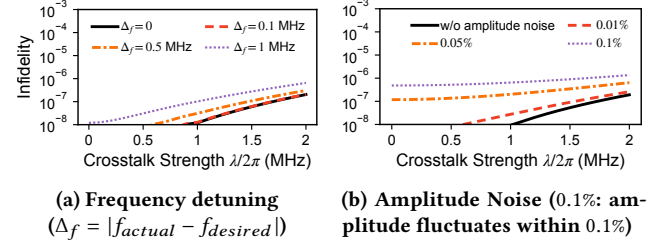
(a) Frequency detuning ($\Delta_f = |f_{actual} - f_{desired}|$)

(b) Amplitude Noise (0.1%: amplitude fluctuates within 0.1%)

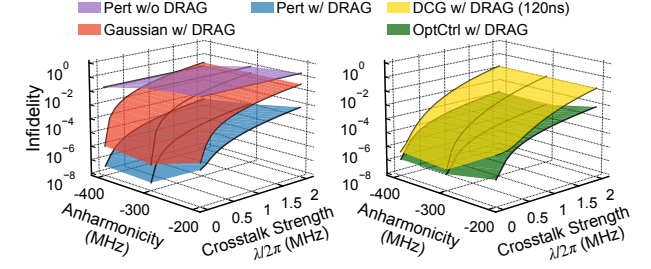**Figure 17: Robustness of the Pert $R_x(\pi/2)$ pulse to drive noise.**

Figure 18: Suppression performance of $R_x(\pi/2)$ pulses under ZZ crosstalk and leakage errors. Lower is better.

pulses optimized for two-level systems and make them robust to leakage errors on multi-level systems [25]. Our work suppresses ZZ crosstalk through optimizing two-level Hamiltonian, and then applies DRAG on top of the pulses generated from such Hamiltonian to overcome leakage errors. We process the optimized pulses with DRAG and then evaluate them on a five-level system with typical anharmonicity [4, 15]. Figure 18 shows that the optimized pulses processed by DRAG can achieve simultaneous suppression of both ZZ crosstalk (versus Gaussian w/ DRAG) and leakage errors (versus Pert w/o DRAG).

Since the effect of drive noise and leakage errors on the suppression performance is limited, we neglect these sources of errors and focus only on ZZ crosstalk when performing the evaluation on quantum computing benchmarks.

7.2.2 Two-Qubit Gates. For a four-qubit system ①-❷-❸-④ with a two-qubit gate on qubits 2 and 3, when (cross-region) crosstalk 1-2 and 3-4 are fully suppressed, the evolution of qubits 1 and 4 would be $I \otimes I$. We thus use the infidelity between $I \otimes I$ and the actual evolution to characterize suppression performance.
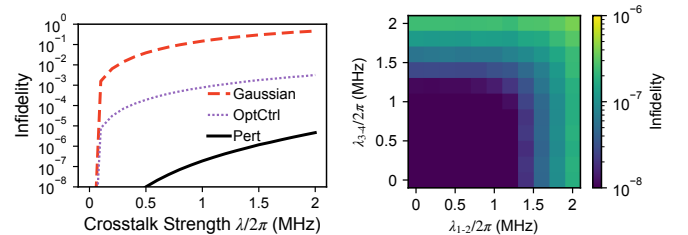
(a) The same crosstalk strength on 1-2 and 3-4

(b) Different strengths on 1-2 and 3-4 (showing the Pert pulse)

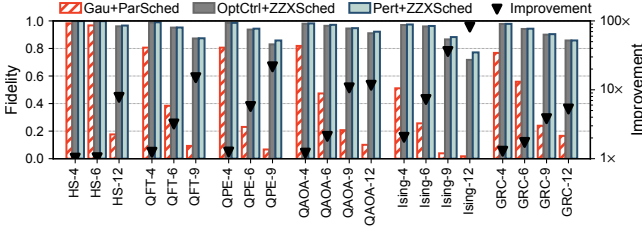**Figure 19: ZZ crosstalk suppression performance of $R_{zx}(\pi/2)$ pulses. Lower is better.**

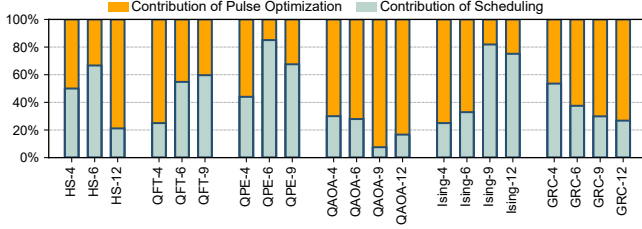Figure 20: Overall improvements in fidelity under *ZZ* crosstalk. Higher is better. (Gau: Gaussian pulses)

Figure 21: Comparisons with using only optimized pulses (Pert+ParSched) or ZZXSched (Gau+ZXSched).

Figure 22: Contribution of pulse optimization/scheduling to the fidelity improvements from Gau+ParSched to Pert+ZZXSched.

Figure 23: 6-qubit benchmarks under *ZZ* crosstalk and decoherence errors ($T_1 = T_2$). Higher is better.

Suppression Performance. Figure 19 shows the results under two configurations. (DCG is omitted since its sequence for two-qubit gates is too complicated and too long in practice.) These results show again the effective suppression of cross-region *ZZ* crosstalk during two-qubit gates.

7.3 Quantum Computing

Benchmarks. We use 6 representative benchmarks for near-term QC devices which have been widely used in recent works [19, 47, 48, 62]: Hidden Shift (HS) algorithm [13], Quantum Fourier Transform (QFT) [51], Quantum Phase Estimation (QPE) [51], Quantum Approximate Optimization Algorithm (QAOA) [20], Ising model simulation (Ising) [7], and Google Random Circuits (GRC) [4]. We vary the number of qubits (4,6,9,12) for each benchmark. The compilation for any of these benchmarks with our approach takes <0.25s on a 2.3GHz CPU.

Setup. Crosstalk strengths $\lambda/2\pi$ for different couplings are sampled from $N(\mu, \sigma^2)$ with μ=200kHz and σ=50kHz [3, 26]. To analyze the effect of decoherence, we consider relaxation and dephasing characterized by the T_1 and T_2 times of qubits. We perform simulations on a 3×4 grid topology $G = (V, E)$, and the suppression requirement for scheduling is $N_Q < \max_{v \in V} degree(v)$ and $N_C \leq \frac{1}{2}|E|$. $\alpha = 0.5$ and $k = 3$ are used for the optimal suppression algorithm.

Comparison. We name our scheduling policy ZZXSched and use it with OptCtrl and Pert pulses. We compare our approach with Gaussian pulses (Gau) + parallel scheduling (ParSched). ParSched maximizes the number of gates executed in parallel, which is the state of the art [49] in Qiskit [1] and Qulic [59]. We also make comparisons on devices with tunable couplers. The fidelity between actual output states and ideal output states (i.e., no error case) is used as the metric.

Improvements in Fidelity. Figure 20 shows the overall improvements from pulse and scheduling co-optimization. There are three key results. **(1) Our approach can largely improve the circuit**
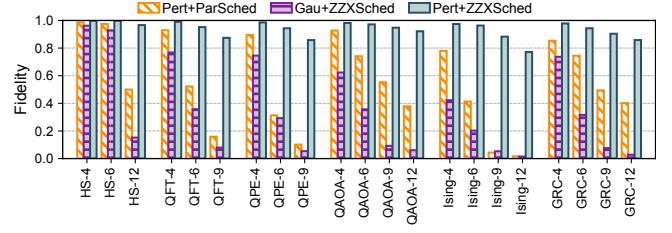
fidelity by up to 81× (11× on average) over Gau+ParSched. More importantly, we achieve > 0.9 fidelity for most of the evaluated benchmarks. **(2)** As revealed by the similar improvements with OptCtrl and Pert pulses, **our approach is insensitive to the pulses used**, as long as they can suppress *ZZ* crosstalk to some extent. This is because, after compiled by our approach, circuits are affected mainly by the remaining intra-region crosstalk, and thus the varying performance of different pulses on cross-region crosstalk suppression becomes relatively unimportant. **(3) Our approach achieves an increasing improvement with more qubits.** This is significant, as it indicates that our approach can play a more important role in future larger devices.

Analysis of Co-Optimization. Figure 21 shows the fidelity when using only optimized pulses (Pert+ParSched) or ZZXSched (Gau+ZZXSched). It highlights the synergistic effect of co-optimization by showing that co-optimization can achieve higher fidelity than using each part individually.

Breakdown. Figure 22 shows the contribution of pulse optimization and scheduling to the overall improvements. Averaged over all the tested benchmarks, pulse optimization and scheduling contribute 43.7% and 56.3% of the improvements respectively. The contribution of pulse optimization is calculated by the ratio of the improvement with only Pert pulses (Pert+ParSched) to the overall improvement.

Analysis of Decoherence. In addition to *ZZ* crosstalk, decoherence is also a significant source of errors. We therefore further evaluate our approach under both the two error sources. Figure 23 presents the fidelity of 6-qubit benchmarks across a series of T_1 and T_2 times. It shows that our approach maintains stable improvements over different T_1 and T_2 times, revealing that the impact of decoherence on the effectiveness of our approach is limited.

Effect on Parallelism. Figure 24 shows the execution time of each benchmark. Compared with ParSched, our ZZXSched typically increases the execution time by <2×, showing a limited sacrifice of
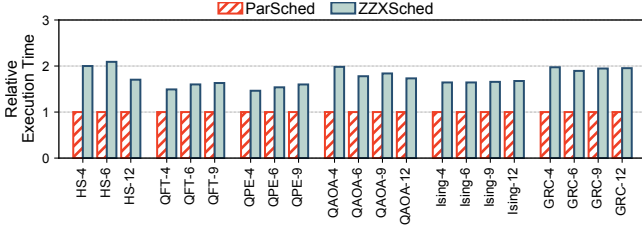
Figure 24: Execution time of benchmarks (relative to the time with ParSched). Results are irrelevant of pulses used.
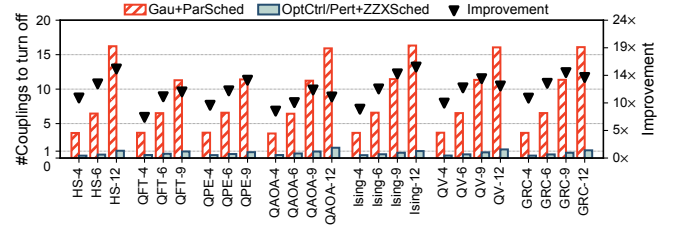

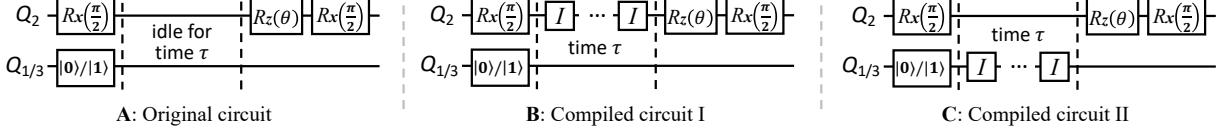**Figure 25: #Couplings to turn off (averaged over all layers). Lower is better.**

Figure 26: Circuits for Ramsey experiments on a real device $Q_1-Q_2-Q_3$. $\theta \propto \tau$. The difference between the compiled circuits B and C is that B applies I gates to Q_2 while C applies I to Q_1 and Q_3.
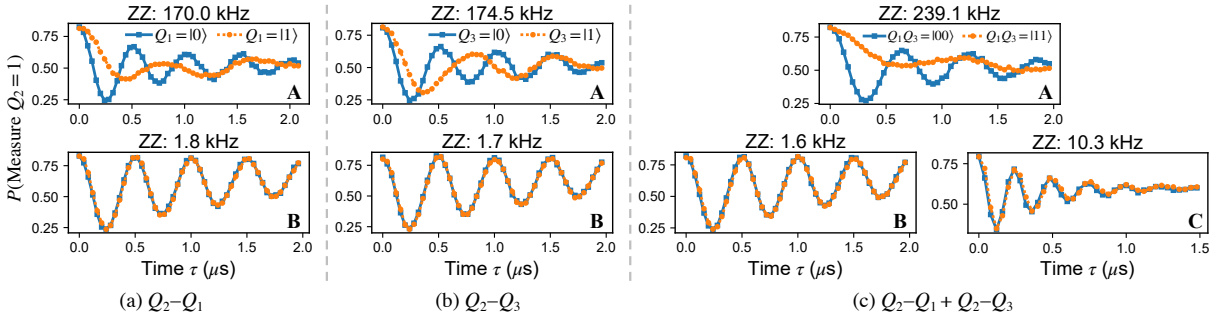

Figure 27: Results of Ramsey experiments. The letter at the lower right corner of each figure indicates the circuit from which the results are obtained. ZZ strength is smaller when the oscillation frequency of the two curves gets closer.

parallelism for better suppression. This trade-off is worthy, as we have already seen in both Figure 20 and 23 that the overall fidelity is improved.

Comparison on Devices with Tunable Couplers. Devices with tunable couplers can suppress ZZ crosstalk by "turning off" couplings [46]. In practice, the process of turning off can incur additional control noise [4, 54]. Our approach can be used on these devices and significantly reduce #couplings to turn off (since only couplings within regions need to be turned off), thus reducing control noise and improving fidelity. Figure 25 shows a 10∼20× reduction over the baseline and a very slow growth of #couplings to turn off under our approach.

7.4 Ramsey Experiments on a Real QC Device

In this section, we further evaluate our approach from the perspective of effective ZZ strength, where *effective strength* refers to the strength that *actually* affects the fidelity of quantum circuits. Thus, suppressing the effect of ZZ crosstalk on QC is equivalent to reducing the effective ZZ strength.

A standard protocol for measuring ZZ strength between two qubits Q_1, Q_2 is to perform two Ramsey experiments on Q_2, with Q_1 in $|0\rangle$ or $|1\rangle$ respectively [14]. For each experiment, the population of $|1\rangle$ on Q_2 will oscillate at some frequency, and ZZ strength can be calculated by the difference between the frequencies obtained

from the two experiments. Figure 26 **A** shows the original circuit for Ramsey experiments. Our approach gives two compiled circuits that apply additional I gates to different qubits, as shown in Figure 26 **B** and **C**.

We experiment on a real QC device [68] with three transmon qubits in a line ($Q_1-Q_2-Q_3$). The qubit frequency is 5.78, 5.12 and 5.81 GHz respectively, and the adjacent qubits are coupled by a capacitor with the coupling strength ∼17 MHz. The default pulses on the device are Gaussian pulses, and our approach uses DCG pulses.

We have conducted three groups of experiments. Figure 27 shows the experimental results. They reveal two important conclusions.

First, the results have verified the foundations of our approach. Figure 27 (a) and (b) show that cross-region crosstalk on one coupling can be suppressed, by pulses applied to one of the qubits on the coupling. Figure 27 (c) shows that cross-region crosstalk on multiple couplings around a qubit can be suppressed simultaneously, by pulses applied to either that qubit or its neighbors. These cases are the foundations of our approach, and the experiments have verified them.

Second, the results have shown the effectiveness of our approach. They show that our approach can reduce the effective ZZ crosstalk strength from ∼200 kHz to <11 kHz, which indicates strong suppression of ZZ crosstalk.

8 RELATED WORKS

ZZ **Crosstalk Suppression.** Previous approaches have mostly relied on sophisticated and specific chip design: tunable couplers [39, 52, 60, 65], heterogeneous qubits [37, 53, 67], and multiple coupling paths [33, 46]. They can bring challenges to chip fabrication and introduce additional decoherence factors [33, 42]. We propose a *software* approach that does not require special hardware to implement. It can be used on both the sophisticated chips described above (see Figure 25) and much simpler devices (see Figure 27). Our approach can potentially open up a new avenue to *ZZ*-resilient QC and simplify the design of QC devices.

Error Mitigation. Pulse optimization [12, 21, 27] or scheduling [49] has been explored by prior error mitigation works, but most of these works exploit only one of them. We take a step towards pulse and scheduling *co-optimization* and show an effective application in *ZZ* crosstalk suppression. We hope to inspire similar co-optimization strategies to tackle other types of errors.

Recently, two works have been proposed to mitigate other types of crosstalk than *ZZ* crosstalk, XtalkSched [49] and ColorDynamic [19]. Our approach can be used with them to simultaneously mitigate multiple types of crosstalk. The following gives some potential ways to combine these approaches.

XtalkSched targets crosstalk arising when multiple gates are executed simultaneously. It appropriately inserts barriers in the circuit. Noting that these barriers actually divide a circuit into multiple sub-circuits, each sub-circuit can then be processed with our approach to further suppress *ZZ* crosstalk.

ColorDynamic mitigates crosstalk due to frequency crowding. It annotates each gate with frequency information (for tuning qubit frequency dynamically) and groups gates into multiple layers. Similar to XtalkSched, each layer can be regarded as a circuit and then processed with our approach.

Dynamic Decoupling. Dynamic decoupling (DD) aims to protect a system from decoherence and dissipation due to unwanted system-environment interactions [63, 64]. Though both DD and our approach target unwanted interactions, the main difference is that DD takes a view of idling, while we take a view of computing. Specifically, DD applies pulses during the *idle* periods of qubits. These pulses are not aimed at implementing a target gate. In contrast, the primary goal of our pulses is to implement a target gate, though we also require the pulses to suppress *ZZ* crosstalk surrounding the gate. With scheduling, we protect qubits in their whole lifetime including both computing and idle periods.

Our approach can use DD to provide protection during idle periods, by substituting DD pulses for the additional identity pulses. Actually, the DCG identity pulse is a special case of DD pulses; when using our approach with DCG pulses, the protection from DCG identity pulses can be regarded to be provided by DD.

9 CONCLUSION

We propose a pulse and scheduling co-optimization approach to suppress the destructive *ZZ* crosstalk of superconducting QC devices in a scalable way. Our approach does not require special hardware to support. We have implemented it as a general framework and shown the compatibility with various pulse optimization methods.

We have also demonstrated the effectiveness of our approach both in simulated settings and on a real device.

A APPENDIX: OPTIMIZED PULSES

Figure 28 shows the optimized pulses for $R_x(\pi/2)$.

(a) OptCtrl   **(b) Pert**
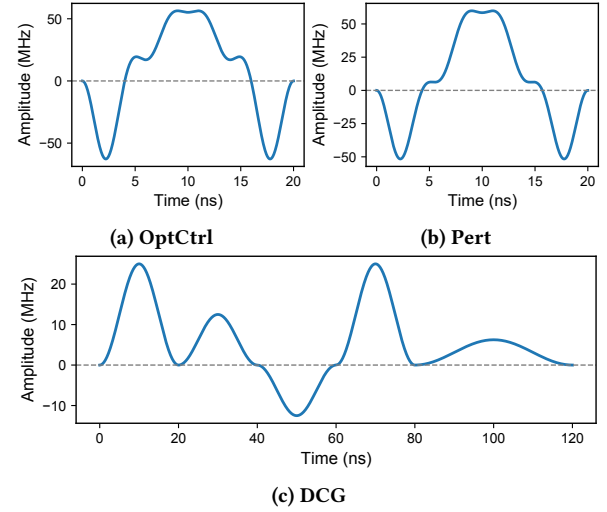
(c) DCG

Figure 28: Optimized pulses for $R_x(\pi/2)$. The amplitude and duration of these optimized pulses are reasonable [6].

We select a Fourier form $\Omega(\mathbf{A}, t)$ for OptCtrl and Pert pulses, which is smooth, of narrow bandwidth and friendly to arbitrary waveform generators.

$$\Omega(\mathbf{A}, t) = \sum_{j=1}^{5} \frac{\mathbf{A}_j}{2} \left[1 + \cos\left(\frac{2\pi j}{T} t - \pi\right) \right]$$

where T denotes pulse duration, and $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_N)$ denote parameters to optimize.

DCG does not optimize pulses from scratch; it leverages existing pulses and combines them into a sequence [35, 36]. Figure 28c shows the sequence for $R_x(\pi/2)$ which is composed of 4 parts. (1) 0~20ns: a Gaussian π pulse; (2) 20~60ns: a Gaussian $\pi/2$ pulse followed by a Gaussian $-\pi/2$ pulse; (3) 60~80ns: a Gaussian π pulse; (4) 80~120ns: a Gaussian $\pi/2$ pulse.

REFERENCES

[1] Héctor Abraham, AduOffei, Rochisha Agarwal, Ismail Yunus Akhalwaya, Gadi Aleksandrowicz, Thomas Alexander, Matthew Amy, Eli Arbel, Arijit02, Abraham Asfaw, Artur Avkhadiev, Carlos Azaustre, AzizNgoueya, Abhik Banerjee, Aman Bansal, Panagiotis Barkoutsos, Ashish Barnawal, George Barron, George S. Barron, Luciano Bello, Yael Ben-Haim, Daniel Bevenius, Arjun Bhobe, Lev S. Bishop, Carsten Blank, Sorin Bolos, Samuel Bosch, Brandon, Sergey Bravyi, Bryce-Fuller, David Bucher, Artemiy Burov, Fran Cabrera, Padraic Calpin, Lauren Capelluto, Jorge Carballo, Ginés Carrascal, Adrian Chen, Chun-Fu Chen, Edward Chen,

Jielun (Chris) Chen, Richard Chen, Jerry M. Chow, Spencer Churchill, Christian Claus, Christian Clauss, Romilly Cocking, Filipe Correa, Abigail J. Cross, Andrew W. Cross, Simon Cross, Juan Cruz-Benito, Chris Culver, Antonio D. Córcoles-Gonzales, Sean Dague, Tareq El Dandachi, Marcus Daniels, Matthieu Dartiailh, DavideFrr, Abdón Rodríguez Davila, Anton Dekusar, Delton Ding, Jun Doi, Eric Drechsler, Drew, Eugene Dumitrescu, Karel Dumon, Ivan Duran, Kareem EL-Safty, Eric Eastman, Grant Eberle, Pieter Eendebak, Daniel Egger, Mark Everitt, Paco Martín Fernández, Axel Hernández Ferrera, Romain Fouilland, FranckChevallier, Albert Frisch, Andreas Fuhrer, Bryce Fuller, MELVIN GEORGE, Julien Gacon, Borja Godoy Gago, Claudio Gambella, Jay M. Gambetta, Adhisha Gammanpila, Luis Garcia, Tanya Garg, Shelly Garion, Austin Gilliam, Aditya Giridharan, Juan Gomez-Mosquera, Gonzalo, Salvador de la Puente González, Jesse Gorzinski, Ian Gould, Donny Greenberg, Dmitry Grinko, Wen Guan, John A. Gunnels, Mikael Haglund, Isabel Haide, Ikko Hamamura, Omar Costa Hamido, Frank Harkins, Vojtech Havlicek, Joe Hellmers, Łukasz Herok, Stefan Hillmich, Hiroshi Horii, Connor Howington, Shaohan Hu, Wei Hu, Junye Huang, Rolf Huisman, Haruki Imai, Takashi Imamichi, Kazuaki Ishizaki, Raban Iten, Toshinari Itoko, JamesSeaward, Ali Javadi, Ali Javadi-Abhari, Wahaj Javed, Jessica, Madhav Jivrajani, Kiran Johns, Scott Johnstun, Jonathan-Shoemaker, Vismai K, Tal Kachmann, Akshay Kale, Naoki Kanazawa, Kang-Bae, Anton Karazeev, Paul Kassebaum, Josh Kelso, Spencer King, Knabberjoe, Yuri Kobayashi, Arseny Kovyrshin, Rajiv Krishnakumar, Vivek Krishnan, Kevin Krsulich, Prasad Kumkar, Gawel Kus, Ryan LaRose, Enrique Lacal, Raphaël Lambert, John Lapeyre, Joe Latone, Scott Lawrence, Christina Lee, Gushu Li, Dennis Liu, Peng Liu, Yunho Maeng, Kahan Majmudar, Aleksei Malyshev, Joshua Manela, Jakub Marecek, Manoel Marques, Dmitri Maslov, Dolph Mathews, Atsushi Matsuo, Douglas T. McClure, Cameron McGarry, David McKay, Dan McPherson, Srujan Meesala, Thomas Metcalfe, Martin Mevissen, Andrew Meyer, Antonio Mezzacapo, Rohit Midha, Zlatko Minev, Abby Mitchell, Nikolaj Moll, Jhon Montanez, Gabriel Monteiro, Michael Duane Mooring, Renier Morales, Niall Moran, Mario Motta, MrF, Prakash Murali, Jan Müggenburg, David Nadlinger, Ken Nakanishi, Giacomo Nannicini, Paul Nation, Edwin Navarro, Yehuda Naveh, Scott Wyman Neagle, Patrick Neuweiler, Johan Nicander, Pradeep Niroula, Hassi Norlen, NuoWenLei, Lee James O'Riordan, Oluwatobi Ogunbayo, Pauline Ollitrault, Raul Otaolea, Steven Oud, Dan Padilha, Hanhee Paik, Soham Pal, Yuchen Pang, Vincent R. Pascuzzi, Simone Perriello, Anna Phan, Francesco Piro, Marco Pistoia, Christophe Piveteau, Pierre Pocreau, Alejandro Pozas-Kerstjens, Milos Prokop, Viktor Prutyanov, Daniel Puzzuoli, Jesús Pérez, Quintiii, Rafey Iqbal Rahman, Arun Raja, Nipun Ramagiri, Anirudh Rao, Rudy Raymond, Rafael Martín-Cuevas Redondo, Max Reuter, Julia Rice, Matt Riedemann, Marcello La Rocca, Diego M. Rodríguez, RohithKarur, Max Rossmannek, Mingi Ryu, Tharrmashastha SAPV, SamFerracin, Martin Sandberg, Hirmay Sandesara, Ritvik Sapra, Hayk Sargsyan, Aniruddha Sarkar, Ninad Sathaye, Bruno Schmitt, Chris Schnabel, Zachary Schoenfeld, Travis L. Scholten, Eddie Schoute, Joachim Schwarm, Ismael Faro Sertage, Kanav Setia, Nathan Shammah, Yunong Shi, Adenilton Silva, Andrea Simonetto, Nick Singstock, Yukio Siraichi, Iskandar Sitdikov, Seyon Sivarajah, Magnus Berg Sletfjerding, John A. Smolin, Mathias Soeken, Igor Olegovich Sokolov, Igor Sokolov, SooluThomas, Starfish, Dominik Steenken, Matt Stypulkoski, Shaojun Sun, Kevin J. Sung, Hitomi Takahashi, Tanvesh Takawale, Ivano Tavernelli, Charles Taylor, Pete Taylour, Soolu Thomas, Mathieu Tillet, Maddy Tod, Miroslav Tomasik, Enrique de la Torre, Kenso Trabing, Matthew Treinish, TrishaPe, Davindra Tulsi, Wes Turner, Yotam Vaknin, Carmen Recio Valcarce, Francois Varchon, Almudena Carrera Vazquez, Victor Villar, Desiree Vogt-Lee, Christophe Vuillot, James Weaver, Johannes Weidenfeller, Rafal Wieczorek, Jonathan A. Wildstrom, Erick Winston, Jack J. Woehr, Stefan Woerner, Ryan Woo, Christopher J. Wood, Ryan Wood, Stephen Wood, Steve Wood, James Wootton, Daniyar Yeralin, David Yonge-Mallo, Richard Young, Jessie Yu, Christopher Zachow, Laura Zdanski, Helena Zhang, Christa Zoufal, Zoufalc, a kapila, a matsuo, bcamorrison, brandhsn, nick bronn, brosand, chlorophyll zz, csseifms, dekel.meirom, dekelmeirom, dekool, dime10, drholmie, dtrenev, echchen, elfrocampeador, faisaldebouni, fanizzamarco, gabrieleagl, gadial, galeinston, georgios ts, gruu, hhorii, hykavitha, jagunther, jliu45, jscott2, kanejess, klinvill, krutik2966, kurarrr, lerongil, ma5x, merav aharoni, michelle4654, ordmoj, sagar pahwa, rmoyard, saswati qiskit, scottkelso, sethmerkel, shaashwat, sternparky, strickroman, sumitpuri, tigerjack, toural, tsura crisaldo, vvilpas, welien, willhbang, yang.luh, yotamvakninibm, and Mantas Čepulkovskis. 2019. Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2562110

[2] Panos Aliferis, Daniel Gottesman, and John Preskill. 2006. Quantum Accuracy Threshold for Concatenated Distance-3 Codes. *Quantum Info. Comput.* 6, 2 (March 2006), 97–165. https://dl.acm.org/doi/10.5555/2011665.2011666

[3] Christian Kraglund Andersen, Ants Remm, Stefania Lazar, Sebastian Krinner, Nathan Lacroix, Graham J. Norris, Mihai Gabureac, Christopher Eichler, and Andreas Wallraff. 2020. Repeated quantum error detection in a surface code. *Nature Physics* 16, 8 (01 Aug 2020), 875–880. https://doi.org/10.1038/s41567-020-0920-y

[4] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (01 Oct 2019), 505–510. https://doi.org/10.1038/s41586-019-1666-5

[5] A. Ash- Saki, M. Alam, and S. Ghosh. 2020. Experimental Characterization, Modeling, and Analysis of Crosstalk in a Quantum Computer. *IEEE Transactions on Quantum Engineering* 1 (2020), 1–6. https://doi.org/10.1109/TQE.2020.3023338

[6] Harrison Ball, Michael Biercuk, Andre Carvalho, Jiayin Chen, Michael Robert Hush, Leonardo A. de Castro, Li Li, Per J. Liebermann, Harry Slatyer, Claire Edmunds, Virginia Frey, Cronelius Hempel, and Alistair Milne. 2021. Software tools for quantum control: Improving quantum computer performance through noise and error suppression. *Quantum Science and Technology* (2021). https://doi.org/10.1088/2058-9565/abdca6

[7] R. Barends, A. Shabani, L. Lamata, J. Kelly, A. Mezzacapo, U. Las Heras, R. Babbush, A. G. Fowler, B. Campbell, Yu Chen, Z. Chen, B. Chiaro, A. Dunsworth, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. C. White, E. Solano, H. Neven, and John M. Martinis. 2016. Digitized adiabatic quantum computing with a superconducting circuit. *Nature* 534, 7606 (01 Jun 2016), 222–226. https://doi.org/10.1038/nature17658

[8] Glencora Borradaile. 2014. CS523: Advanced Algorithms Lecture 2: MAXCUT for planar graphs. https://web.engr.oregonstate.edu/~glencora/wiki/uploads/planar-max-cut.pdf (School of Electrical Engineering and Computer Science, Oregon State University).

[9] T.-Q. Cai, X.-Y. Han, Y.-K Wu, Y.-L. Ma, J.-H Wang, Z.-L. Wang, H.-Y Zhang, H.-Y Wang, Y.-P. Song, and L.-M. Duan. 2021. Impact of Spectators on a Two-Qubit Gate in a Tunable Coupling Superconducting Circuit. *Phys. Rev. Lett.* 127 (Aug 2021), 060505. Issue 6. https://doi.org/10.1103/PhysRevLett.127.060505

[10] S. A. Caldwell, N. Didier, C. A. Ryan, E. A. Sete, A. Hudson, P. Karalekas, R. Manenti, M. P. da Silva, R. Sinclair, E. Acala, N. Alidoust, J. Angeles, A. Bestwick, M. Block, B. Bloom, A. Bradley, C. Bui, L. Capelluto, R. Chilcott, J. Cordova, G. Crossman, M. Curtis, S. Deshpande, T. El Bouayadi, D. Girshovich, S. Hong, K. Kuang, M. Lenihan, T. Manning, A. Marchenkov, J. Marshall, R. Maydra, Y. Mohan, W. O'Brien, C. Osborn, J. Otterbach, A. Papageorge, J.-P. Paquette, M. Pelstring, A. Polloreno, G. Prawiroatmodjo, V. Rawat, M. Reagor, R. Renzas, N. Rubin, D. Russell, M. Rust, D. Scarabelli, M. Scheer, M. Selvanayagam, R. Smith, A. Staley, M. Suska, N. Tezak, D. C. Thompson, T.-W. To, M. Vahidpour, N. Vodrahalli, T. Whyland, K. Yadav, W. Zeng, and C. Rigetti. 2018. Parametrically Activated Entangling Gates Using Transmon Qubits. *Phys. Rev. Applied* 10 (Sep 2018), 034050. Issue 3. https://doi.org/10.1103/PhysRevApplied.10.034050

[11] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. 2017. Roads towards fault-tolerant universal quantum computation. *Nature* 549, 7671 (01 Sep 2017), 172–179. https://doi.org/10.1038/nature23460

[12] Andre R. R. Carvalho, Harrison Ball, Michael J. Biercuk, Michael R. Hush, and Felix Thomsen. 2021. Error-Robust Quantum Logic Optimization Using a Cloud Quantum Computer Interface. *Phys. Rev. Applied* 15 (Jun 2021), 064054. Issue 6. https://doi.org/10.1103/PhysRevApplied.15.064054

[13] Andrew M. Childs and Wim van Dam. 2007. Quantum Algorithm for a Generalized Hidden Shift Problem. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (New Orleans, Louisiana) *(SODA '07)*. Society for Industrial and Applied Mathematics, USA, 1225–1232. https://dl.acm.org/doi/10.5555/1283383.1283515

[14] Jerry Moy Chow. 2010. *Quantum information processing with superconducting qubits.* Ph.D. Dissertation. Yale University.

[15] Jerry M. Chow, A. D. Córcoles, Jay M. Gambetta, Chad Rigetti, B. R. Johnson, John A. Smolin, J. R. Rozen, George A. Keefe, Mary B. Rothwell, Mark B. Ketchen, and M. Steffen. 2011. Simple All-Microwave Entangling Gate for Fixed-Frequency Superconducting Qubits. *Phys. Rev. Lett.* 107 (Aug 2011), 080502. Issue 8. https://doi.org/10.1103/PhysRevLett.107.080502

[16] A. D. Córcoles, Easwar Magesan, Srikanth J. Srinivasan, Andrew W. Cross, M. Steffen, Jay M. Gambetta, and Jerry M. Chow. 2015. Demonstration of a quantum error detection code using a square lattice of four superconducting qubits. *Nature Communications* 6, 1 (29 Apr 2015), 6979. https://doi.org/10.1038/ncomms7979

[17] Domenico D'Alessandro. 2007. *Introduction to quantum control and dynamics.* CRC press.

[18] L. DiCarlo, J. M. Chow, J. M. Gambetta, Lev S. Bishop, B. R. Johnson, D. I. Schuster, J. Majer, A. Blais, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf. 2009. Demonstration of two-qubit algorithms with a superconducting quantum processor. *Nature* 460, 7252 (01 Jul 2009), 240–244. https://doi.org/10.1038/nature08121

[19] Y. Ding, P. Gokhale, S. F. Lin, R. Rines, T. Propson, and F. T. Chong. 2020. Systematic Crosstalk Mitigation for Superconducting Qubits via Frequency-Aware Compilation. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 201–214. https://doi.org/10.1109/MICRO50266.2020.00028

[20] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028* (2014).

[21] T. Figueiredo Roque, Aashish A. Clerk, and Hugo Ribeiro. 2021. Engineering fast high-fidelity quantum operations with constrained interactions. *npj Quantum Information* 7, 1 (08 Feb 2021), 28. https://doi.org/10.1038/s41534-020-00349-z

[22] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. 2012. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A* 86 (Sep 2012), 032324. Issue 3. https://doi.org/10.1103/PhysRevA.86.032324

[23] Zvi Galil. 1986. Efficient Algorithms for Finding Maximum Matching in Graphs. *ACM Comput. Surv.* 18, 1 (March 1986), 23–38. https://doi.org/10.1145/6462.6502

[24] Jay M. Gambetta, Jerry M. Chow, and Matthias Steffen. 2017. Building logical qubits in a superconducting quantum computing system. *npj Quantum Information* 3, 1 (13 Jan 2017), 2. https://doi.org/10.1038/s41534-016-0004-0

[25] J. M. Gambetta, F. Motzoi, S. T. Merkel, and F. K. Wilhelm. 2011. Analytic control methods for high-fidelity unitary operations in a weakly nonlinear oscillator. *Phys. Rev. A* 83 (Jan 2011), 012308. Issue 1. https://doi.org/10.1103/PhysRevA.83.012308

[26] M. Ganzhorn, G. Salis, D. J. Egger, A. Fuhrer, M. Mergenthaler, C. Müller, P. Müller, S. Paredes, M. Pechal, M. Werninghaus, and S. Filipp. 2020. Benchmarking the noise sensitivity of different parametric two-qubit gates in a single superconducting quantum computing platform. *Phys. Rev. Research* 2 (Sep 2020), 033447. Issue 3. https://doi.org/10.1103/PhysRevResearch.2.033447

[27] Pranav Gokhale, Ali Javadi-Abhari, Nathan Earnest, Yunong Shi, and Frederic T. Chong. 2020. Optimized Quantum Compilation for Near-Term Algorithms with OpenPulse. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 186–200. https://doi.org/10.1109/MICRO50266.2020.00027

[28] F. Hadlock. 1975. Finding a Maximum Cut of a Planar Graph in Polynomial Time. *SIAM J. Comput.* 4, 3 (1975), 221–225. https://doi.org/10.1137/0204019

[29] X. Y. Han, T. Q. Cai, X. G. Li, Y. K. Wu, Y. W. Ma, Y. L. Ma, J. H. Wang, H. Y. Zhang, Y. P. Song, and L. M. Duan. 2020. Error analysis in suppression of unwanted qubit interactions for a parametric gate in a tunable superconducting circuit. *Phys. Rev. A* 102 (Aug 2020), 022619. Issue 2. https://doi.org/10.1103/PhysRevA.102.022619

[30] IBM. 2021. IBM Quantum. https://quantum-computing.ibm.com/

[31] IBM. 2021. IBMQ Backend Information. https://github.com/Qiskit/ibmq-device-information

[32] J.R. Johansson, P.D. Nation, and Franco Nori. 2013. QuTiP 2: A Python framework for the dynamics of open quantum systems. *Computer Physics Communications* 184, 4 (2013), 1234–1240. https://doi.org/10.1016/j.cpc.2012.11.019

[33] A Kandala, KX Wei, S Srinivasan, E Magesan, S Carnevale, GA Keefe, D Klaus, O Dial, and DC McKay. 2020. Demonstration of a High-Fidelity CNOT for Fixed-Frequency Transmons with Engineered ZZ Suppression. *arXiv preprint arXiv:2011.07050* (2020).

[34] Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte-Herbrüggen, and Steffen J. Glaser. 2005. Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *Journal of Magnetic Resonance* 172, 2 (2005), 296–305. https://doi.org/10.1016/j.jmr.2004.11.004

[35] Kaveh Khodjasteh and Lorenza Viola. 2009. Dynamical quantum error correction of unitary operations with bounded controls. *Phys. Rev. A* 80 (Sep 2009), 032314. Issue 3. https://doi.org/10.1103/PhysRevA.80.032314

[36] Kaveh Khodjasteh and Lorenza Viola. 2009. Dynamically Error-Corrected Gates for Universal Quantum Computation. *Phys. Rev. Lett.* 102 (Feb 2009), 080501. Issue 8. https://doi.org/10.1103/PhysRevLett.102.080501

[37] Jaseung Ku, Xuexin Xu, Markus Brink, David C. McKay, Jared B. Hertzberg, Mohammad H. Ansari, and B. L. T. Plourde. 2020. Suppression of Unwanted *ZZ* Interactions in a Hybrid Two-Qubit System. *Phys. Rev. Lett.* 125 (Nov 2020), 200504. Issue 20. https://doi.org/10.1103/PhysRevLett.125.200504

[38] Nelson Leung, Mohamed Abdelhafez, Jens Koch, and David Schuster. 2017. Speedup for quantum optimal control from automatic differentiation based on graphics processing units. *Phys. Rev. A* 95 (Apr 2017), 042318. Issue 4. https://doi.org/10.1103/PhysRevA.95.042318

[39] X. Li, T. Cai, H. Yan, Z. Wang, X. Pan, Y. Ma, W. Cai, J. Han, Z. Hua, X. Han, Y. Wu, H. Zhang, H. Wang, Yipu Song, Luming Duan, and Luyan Sun. 2020. Tunable Coupler for Realizing a Controlled-Phase Gate with Dynamically Decoupled Regime in a Superconducting Circuit. *Phys. Rev. Applied* 14 (Aug 2020), 024070. Issue 2. https://doi.org/10.1103/PhysRevApplied.14.024070

[40] Daniel A Lidar and Todd A Brun. 2013. *Quantum error correction*. Cambridge university press.

[41] Easwar Magesan and Jay M. Gambetta. 2020. Effective Hamiltonian models of the cross-resonance gate. *Phys. Rev. A* 101 (May 2020), 052308. Issue 5. https://doi.org/10.1103/PhysRevA.101.052308

[42] Moein Malekakhlagh, Easwar Magesan, and David C. McKay. 2020. First-principles analysis of cross-resonance gate operation. *Phys. Rev. A* 102 (Oct 2020), 042605. Issue 4. https://doi.org/10.1103/PhysRevA.102.042605

[43] David C. McKay, Sarah Sheldon, John A. Smolin, Jerry M. Chow, and Jay M. Gambetta. 2019. Three-Qubit Randomized Benchmarking. *Phys. Rev. Lett.* 122 (May 2019), 200502. Issue 20. https://doi.org/10.1103/PhysRevLett.122.200502

[44] David C. McKay, Christopher J. Wood, Sarah Sheldon, Jerry M. Chow, and Jay M. Gambetta. 2017. Efficient *Z* gates for quantum computing. *Phys. Rev. A* 96 (Aug 2017), 022330. Issue 2. https://doi.org/10.1103/PhysRevA.96.022330

[45] F. Motzoi, J. M. Gambetta, P. Rebentrost, and F. K. Wilhelm. 2009. Simple Pulses for Elimination of Leakage in Weakly Nonlinear Qubits. *Phys. Rev. Lett.* 103 (Sep 2009), 110501. Issue 11. https://doi.org/10.1103/PhysRevLett.103.110501

[46] Pranav Mundada, Gengyan Zhang, Thomas Hazard, and Andrew Houck. 2019. Suppression of Qubit Crosstalk in a Tunable Coupling Superconducting Circuit. *Phys. Rev. Applied* 12 (Nov 2019), 054023. Issue 5. https://doi.org/10.1103/PhysRevApplied.12.054023

[47] Prakash Murali, Jonathan M. Baker, Ali Javadi-Abhari, Frederic T. Chong, and Margaret Martonosi. 2019. Noise-Adaptive Compiler Mappings for Noisy Intermediate-Scale Quantum Computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (Providence, RI, USA) *(ASPLOS '19)*. Association for Computing Machinery, New York, NY, USA, 1015–1029. https://doi.org/10.1145/3297858.3304075

[48] Prakash Murali, Norbert Matthias Linke, Margaret Martonosi, Ali Javadi Abhari, Nhung Hong Nguyen, and Cinthia Huerta Alderete. 2019. Full-Stack, Real-System Quantum Computer Studies: Architectural Comparisons and Design Insights. In *Proceedings of the 46th International Symposium on Computer Architecture* (Phoenix, Arizona) *(ISCA '19)*. Association for Computing Machinery, New York, NY, USA, 527–540. https://doi.org/10.1145/3307650.3322273

[49] Prakash Murali, David C. Mckay, Margaret Martonosi, and Ali Javadi-Abhari. 2020. Software Mitigation of Crosstalk on Noisy Intermediate-Scale Quantum Computers. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems* (Lausanne, Switzerland) *(ASPLOS '20)*. Association for Computing Machinery, New York, NY, USA, 1001–1016. https://doi.org/10.1145/3373376.3378477

[50] Michael A Nielsen. 2002. A simple formula for the average gate fidelity of a quantum dynamical operation. *Physics Letters A* 303, 4 (2002), 249–252. https://doi.org/10.1016/S0375-9601(02)01272-0

[51] Michael A. Nielsen and Isaac L. Chuang. 2011. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (10th ed.). Cambridge University Press, USA.

[52] A. O. Niskanen, K. Harrabi, F. Yoshihara, Y. Nakamura, S. Lloyd, and J. S. Tsai. 2007. Quantum Coherent Tunable Coupling of Superconducting Qubits. *Science* 316, 5825 (2007), 723–726. https://doi.org/10.1126/science.1141324

[53] Atsushi Noguchi, Alto Osada, Shumpei Masuda, Shingo Kono, Kentaro Heya, Samuel Piotr Wolski, Hiroki Takahashi, Takanori Sugiyama, Dany Lachance-Quirion, and Yasunobu Nakamura. 2020. Fast parametric two-qubit gates with suppressed residual interaction using the second-order nonlinearity of a cubic transmon. *Phys. Rev. A* 102 (Dec 2020), 062408. Issue 6. https://doi.org/10.1103/PhysRevA.102.062408

[54] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2 (Aug. 2018), 79. https://doi.org/10.22331/q-2018-08-06-79

[55] Chad Rigetti and Michel Devoret. 2010. Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies. *Phys. Rev. B* 81 (Apr 2010), 134507. Issue 13. https://doi.org/10.1103/PhysRevB.81.134507

[56] Mohan Sarovar, Timothy Proctor, Kenneth Rudinger, Kevin Young, Erik Nielsen, and Robin Blume-Kohout. 2020. Detecting crosstalk errors in quantum information processors. *Quantum* 4 (Sep 2020), 321. https://doi.org/10.22331/q-2020-09-11-321

[57] Yunong Shi, Nelson Leung, Pranav Gokhale, Zane Rossi, David I. Schuster, Henry Hoffmann, and Frederic T. Chong. 2019. Optimized Compilation of Aggregated Instructions for Realistic Quantum Computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (Providence, RI, USA) *(ASPLOS '19)*. Association for Computing Machinery, New York, NY, USA, 1031–1044. https://doi.org/10.1145/3297858.3304018

[58] Peter W. Shor. 1997. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* 26, 5 (Oct. 1997), 1484–1509. https://doi.org/10.1137/S0097539795293172

[59] Robert S Smith, Michael J Curtis, and William J Zeng. 2016. A practical quantum instruction set architecture. *arXiv preprint arXiv:1608.03355* (2016).

[60] Youngkyu Sung, Leon Ding, Jochen Braumüller, Antti Vepsäläinen, Bharath Kannan, Morten Kjaergaard, Ami Greene, Gabriel O. Samach, Chris McNally, David Kim, Alexander Melville, Bethany M. Niedzielski, Mollie E. Schwartz, Jonilyn L. Yoder, Terry P. Orlando, Simon Gustavsson, and William D. Oliver. 2020. Realization of high-fidelity CZ and ZZ-free iSWAP gates with a tunable coupler. *arXiv preprint arXiv:2011.01261* (2020).

[61] Maika Takita, A. D. Córcoles, Easwar Magesan, Baleegh Abdo, Markus Brink, Andrew Cross, Jerry M. Chow, and Jay M. Gambetta. 2016. Demonstration of Weight-Four Parity Measurements in the Surface Code Architecture. *Phys. Rev. Lett.* 117 (Nov 2016), 210505. Issue 21. https://doi.org/10.1103/PhysRevLett.117.210505

[62] Swamit S. Tannu and Moinuddin K. Qureshi. 2019. Not All Qubits Are Created Equal: A Case for Variability-Aware Policies for NISQ-Era Quantum Computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (Providence, RI, USA) *(ASPLOS '19)*. Association for Computing Machinery, New York, NY, USA, 987–999. https://doi.org/10.1145/3297858.3304007

[63] Lorenza Viola, Emanuel Knill, and Seth Lloyd. 1999. Dynamical decoupling of open quantum systems. *Physical Review Letters* 82, 12 (1999), 2417. https://doi.org/10.1103/PhysRevLett.82.2417

[64] Lorenza Viola and Seth Lloyd. 1998. Dynamical suppression of decoherence in two-state quantum systems. *Physical Review A* 58, 4 (1998), 2733. https://doi.org/10.1103/PhysRevA.58.2733

[65] Fei Yan, Philip Krantz, Youngkyu Sung, Morten Kjaergaard, Daniel L. Campbell, Terry P. Orlando, Simon Gustavsson, and William D. Oliver. 2018. Tunable Coupling Scheme for Implementing High-Fidelity Two-Qubit Gates. *Phys. Rev. Applied* 10 (Nov 2018), 054062. Issue 5. https://doi.org/10.1103/PhysRevApplied.10.054062

[66] Jin Y Yen. 1970. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quart. Appl. Math.* 27, 4 (1970), 526–530.

[67] Peng Zhao, Peng Xu, Dong Lan, Ji Chu, Xinsheng Tan, Haifeng Yu, and Yang Yu. 2020. High-Contrast ZZ Interaction Using Superconducting Qubits with Opposite-Sign Anharmonicity. *Phys. Rev. Lett.* 125 (Nov 2020), 200503. Issue 20. https://doi.org/10.1103/PhysRevLett.125.200503

[68] Yu Zhou, Zhenxing Zhang, Zelong Yin, Sainan Huai, Xiu Gu, Xiong Xu, Jonathan Allcock, Fuming Liu, Guanglei Xi, Qiaonian Yu, Hualiang Zhang, Mengyu Zhang, Hekang Li, Xiaohui Song, Zhan Wang, Dongning Zheng, Shuoming An, Yarui Zheng, and Shengyu Zhang. 2021. Rapid and unconditional parametric reset protocol for tunable superconducting qubits. *Nature Communications* 12, 1 (11 Oct 2021), 5924. https://doi.org/10.1038/s41467-021-26205-y