SC19: U: A deep learning approach to noise prediction and circuit optimization for near-term quantum devices

Alexander Zlokapa
azlokapa@caltech.edu
California Institute of Technology

Alexandru Gheorghiu (advisor)
andrugh@caltech.edu
California Institute of Technology

ABSTRACT

Noisy intermediate-scale quantum devices face challenges in achieving high-fidelity computations due to hardware-specific noise. We present a framework for a deep-learning compiler of quantum circuits, designed to reduce the output noise of circuits run on a specific device. Our approach is to first train a convolutional neural network on experimental data from a quantum chip, so as to learn a noise model for that device. We then view the trained network as a noise predictor for quantum circuits and design a compiler that rewrites circuits so as to minimize expected noise, as predicted by the network. We tested this approach using the IBM 5-qubit devices and benchmarked the compiled circuits against the IBM Qiskit compilation algorithm. The results we obtained show a reduction in output noise of 11% (95% CI [10%, 12%]) compared to the Qiskit compiler. Improvement compared to the Qiskit compiler is observed on all available 5-qubit IBM devices, but we find significantly better noise reduction on the device on which the noise model was learned. These results suggest that device-specific compilers designed using machine learning may yield higher fidelity operations, increasing the potential of quantum computing applications.

KEYWORDS

quantum computing, machine learning, artificial intelligence

ACM Reference Format:
Alexander Zlokapa and Alexandru Gheorghiu (advisor). 2020. SC19: U: A deep learning approach to noise prediction and circuit optimization for near-term quantum devices. In *SC '19: ACM/IEEE Supercomputing Conference, November 17–22, 2019, Denver, CO.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 PROBLEM AND MOTIVATION

Quantum computers are expected to offer exponential speedups over classical computers in solving certain computational tasks. The recent demonstration of quantum computational supremacy by the Google group further strengthens the case for the potential of quantum computation [3]. However, the result also highlighted the limitations of current and near-term quantum devices. It showed that *Noisy Intermediate-Scale Quantum* (NISQ) devices are limited in usability and reliability by errors due to thermal relaxation, measurement, and interactions between adjacent qubits [25]. Mitigating the effect of these errors (or noise) is thus a pressing problem of immediate practical relevance. Existing noise models often make simplifying assumptions [8, 27] that limit noise mitigation techniques, motivating the design of a fully learned noise model given by deep learning.[1]

Before describing a quantum circuit compiler, let us take a step back and briefly review the formulation of quantum computation in terms of circuits. Quantum circuits are sequences of unitary operations ("gates") selected from a specific gate set and acting on a number of qubits. If, by composing the gates in the set, one can approximate any unitary operation, we say that the set is *universal*. The most commonly used universal gate set is referred to as the Clifford + T set [22]. To implement a particular quantum algorithm described by a unitary operation, any one of an infinite number of gate sequences can be selected from the universal set; these all form equivalent circuits. Of course, while these circuits correspond to the same unitary operation, each will result in a different operation when run on a specific quantum device. This is because each gate in the circuit is performed imperfectly on the device, thus introducing errors in the application of the unitary. The problem of mitigating noise can thus be phrased as a search process for the *lowest-noise circuit* out of this family of equivalent circuits. In other words, given a quantum circuit, the task is to rewrite it into an equivalent circuit that is expected to run with higher fidelity when implemented on some target quantum hardware.

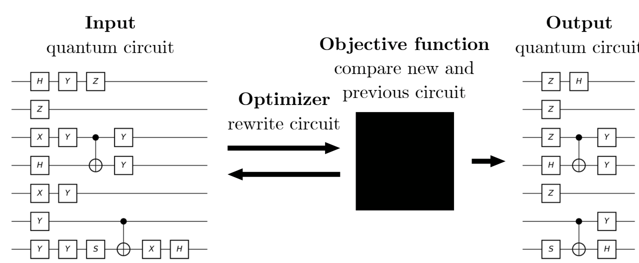


Figure 1: Overview of a quantum circuit compiler. An initial circuit is rewritten by an optimization procedure to satisfy a given objective function — here implemented as a neural network — resulting in a circuit that ideally minimizes noise when run on a quantum device. We address both the optimizer and objective function components in this paper.

A common approach to mitigating noise is to rewrite the initial circuit such that the gate count is minimized as illustrated in Fig. 1.

[1] Code is available at https://github.com/quantummind/deepQ.

SC '19, November 17–22, 2019, Denver, CO

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

This approach works for the simple reason that fewer gates means fewer errors (since each gate is subject to noise) [10, 17, 21]. Other heuristics such as minimizing T-gate count and circuit-depth can also be used [1]. However, these are still just heuristics for minimizing noise. In other words, an objective function, such as gate count, is minimized so as to indirectly minimize the true objective function, noise. It is clear, however, that using a "proxy" objective function (gate count, gate depth, etc.) — although grounded in physical assumptions, whether in the locality or Markovian behavior of the noise — oversimplifies phenomena such as dephasing, amplitude damping, and correlated errors ("cross-talk") between qubits. On the other hand, designing a compiler that accounts for all sources of noise or one that is fine-tuned for a specific device is difficult. What we propose instead, is to use machine learning in order to *learn a noise model* for a given device. We show that this approach can lead to improved results when compared to "heuristic-based" compilers.

In the remainder of the paper, we describe both a method for general compilation of quantum circuits under a *deep learning noise model* and a method for intelligently filling gaps in the circuit with operations that map to the identity (a process known as "dynamical decoupling" [9, 30–32]). A major drawback of gate count and gate depth minimization is that gaps in a quantum circuit will remain empty. Indeed, filling these gaps with gates would increase total gate count. Nevertheless, replacing these sparse regions with operations equivalent to the identity can reduce noise (see Sec. 2.2). We demonstrate this explicitly by showing how our deep learning approach outperforms the circuit optimizer used by IBM as part of their compiler. Specifically, for 5-qubit devices, the circuits we obtain are on average 11% "less noisy" than circuits optimized by the IBM Qiskit compiler.

2 BACKGROUND AND RELATED WORK

2.1 Quantum compilers

One of the most common approaches to noise optimization for quantum circuits is to minimize gate count [10, 17, 21]. Numerous methods exist to minimize gate count or T-gate count, including ZX-calculus [16] or (more commonly) repeatedly applying a set of rewrite patterns of equivalent sequences of gates [21]. For instance, a Hadamard gate, follow by an X gate and followed by another Hadamard gate (HXH) is equivalent to a single Z gate. These rewrite rules are typically applied until gate count can no longer be reduced. The motivation for these methods is to reduce *qubit decoherence*, which increases with time. By minimizing the time taken to evaluate a quantum circuit, noise is approximately suppressed. However, this paradigm neglects other effects: correlated errors between qubits, certain gates being more noisy than other gates, and rates of different types of errors (such as amplitude damping and dephasing) varying across qubits. Other work on noise optimization has focused on other specific aspects, such as minimizing cross-talk caused by ZZ interactions [6]. However, all quantum compilers that we are aware of in the literature assume an error model *a priori* thus restricting the types of noise that can be minimized.

2.2 Noise of quantum devices

The literature on learning noise models for quantum devices is similarly limited. Recent work has quantified correlated errors with a Gibbs random field [13], while more widely used models focus on error rates of individual qubits and gates (single- or two-qubit gates) [4] used in the IBM Qiskit compiler to minimize noise [11]. However, non-Markovian noise [8, 27] is not well-characterized by such models, due to the bidirectional exchange of information to and from the environment in a non-Markovian setting, compared to one-way information leakage into the environment under Markovian assumptions [24]. Nevertheless, non-Markovian dynamics are observed on quantum devices [20]. This motivates a more general approach to a learned noise model, one that can also account for these complex dynamics. Equipped with a more accurate noise model, we may then define a better objective function for noise optimization.

Due to the layout of two-qubit gates on a circuit, gaps can often appear as a qubit must "wait" for a gate to be applied later in the circuit. During this time, the qubit undergoes free evolution and its state will drift. However, dynamical decoupling (DD) may be used to suppress this effect by applying repeated pulses that are equivalent to the identity, canceling interactions between the system and the environment [19]. The optimal design of the DD sequences is dependent on pulse imperfections, non-Markovianity, and other noise model assumptions [2, 12, 15, 29, 29]. Although basic DD sequences such as $XYXY$ have been observed to preserve single-qubit states on IBM and Rigetti devices, they break down due to non-Markovianity [23]. Past optimization of DD sequences [5, 26] has been generally limited to optimization of single-qubit states within theoretical noise models, which does not allow effects such as cross-talk to be accounted for. Since precisely characterizing non-Markovian effects such as cross-talk on multi-qubit states in order to determine optimal DD sequences is theoretically challenging, we propose directly learning the noise model for arbitrary circuits to predict the best DD sequences.

3 APPROACH AND UNIQUENESS

3.1 Overview

In this paper, we propose two methodologies to overcome the shortcomings of existing noise optimization methods: a general compiler and a dynamical decoupling compiler. Central to both approaches is a deep learning noise model, which serves as an objective function during circuit compilation. This noise model is a *convolutional neural network* that is trained to predict the output noise of a quantum circuit. The network is trained with a number of examples of circuits and the noise in their output when run on a specific device[2]. Once the model has been trained, we then design a compiler that uses the learned model as the objective function to be optimized. Given a circuit, C, the compiler attempts to find an equivalent circuit, C', that minimizes the expected output noise as computed by

[2]Output noise can be measured in a number of ways, depending on the application. For small circuits, it is possible to classically simulate the ideal output distribution of a circuit and then compute the trace distance (or some other distance measure like cross-entropy [7] or energy distance [28]) between that distribution and the one resulting from running the circuit on a quantum device. For a large circuit, U, one can quantify noise by running $UU^{\dagger}$ on a known quantum state, such as $|00...0\rangle$, and then counting the number of 1's in the output.

the deep learning model (for a specific quantum device). The circuit C' is then run multiple times on the quantum device to measure the amount of noise in its output. Finally, the results are compared to those obtained from using a traditional quantum circuit compiler.

In our first quantum compiler, we simulate random circuits on the IBM Q Melbourne device and compile the circuits using peephole optimization of rewrite rules (e.g. $HXH = Z$), similar to compilers in the existing literature. Unlike previous work, we do not seek to minimize gate count; instead, we greedily select rewrite rules that are expected to reduce noise according to the deep learning noise model. The resulting compiled circuit is compared to a circuit for which gate count has been minimized. Simulated results are presented below.

This approach, however, does not outperform the IBM Qiskit compiler [11]. The reason is that that IBM compiler performs an additional level of optimization to further minimize gate count, rewriting the circuit into an *approximation* of the original circuit that yields similar (but not necessarily identical) output states using fewer gates. Nevertheless, for large enough circuits run on quantum chips with various geometric constraints, the compiled circuits will still contain many gaps. Hence, we consider a second approach that exclusively fills in gaps of the IBM Qiskit-compiled circuit with dynamical decoupling sequences according to the learned noise model, adding gates so as to further reduce noise. Experimental results on 5-qubit devices are shown below.

3.2 Uniqueness

The novel aspects of our approach are summarized as follows:

(1) *Convolutional neural network noise model of quantum circuits.* By considering qubit operations in the context of neighboring qubits, more complex hardware-specific noise such as cross-talk and phenomena due to entanglement may be directly learned. In contrast, traditional compilers (such as the provided benchmark) only consider the noise of individual gates and qubits.
(2) *Systematic optimization of dynamical decoupling for multi-qubit states.* We provide a new paradigm for hardware-specific optimization of dynamical decoupling sequences on random circuits with highly entangled states, requiring no *a priori* knowledge of dynamical decoupling sequences.
(3) *Experimental realization of dynamical decoupling compilation.* In the recent literature, well-known dynamical decoupling sequences have been measured on single- and two-qubit states for transmon superconducting devices, with mixed results for entangled states [23]. We observe significant noise mitigation with new dynamical decoupling sequences for highly entangled multi-qubit states on the IBM Q device.

3.3 General compiler

3.3.1 Simulated dataset. To generate a training set of equivalent quantum circuits, we first perform an exhaustive search over all 2-qubit circuits with up to 4 gates, each of which may be expressed as a unitary matrix operation. The circuits are divided into equivalence classes labeled by the circuit matrix representation (up to a phase factor). The equalities found in this way may be iteratively applied to regions of a larger quantum circuit to create a family of equivalent circuits.

A simple noise model is simulated in Qiskit [11], calibrating readout errors, depolarization errors and thermal relaxation errors to the IBM Q Melbourne device. A ground-truth dataset is generated with 668 equivalent families of 50 circuits, each with up to 8 qubits and 200 gates selected from the universal gate set $\{X, Y, Z, H, S, T, CNOT\}$. Hence, 1.6 million unique pairs of equivalent circuits are produced, labeling each pair with the noise difference as measured by the energy distance between readout probability distributions [28].

3.3.2 Noise model. We train a deep learning model on pairs of equivalent circuits and predict the difference in noise from the simulated dataset. To establish encoding consistency, we perform a lexicographic topological sort on a directed acyclic graph representation of the circuit. Each circuit is then represented by an multi-channel image (one channel for each gate type), allowing pairs of images corresponding to pairs of quantum circuits to be provided to a noise estimation model. Since correlated errors and other local phenomena contribute to noise, we use a convolutional neural network to compare which of two given circuits are noisier (Fig. 2).

Given the ability of the ResNet convolutional neural network (CNN) architecture [14] to achieve state-of-the-art results on image regression problems [18], we propose an image encoding scheme for quantum circuits and train a ResNet-18 model to learn the noise difference between a pair of equivalent circuits. Because of its importance as a source of noise, the $CNOT$ gate count is concatenated before the fully-connected layer of the ResNet architecture.

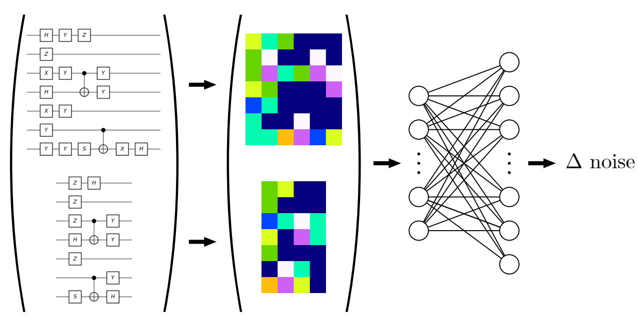


Figure 2: Deep learning noise prediction model consisting of random circuit generation, image encoding via a topological sort, and convolutional neural network prediction of the difference in noise between the initial circuits.

3.3.3 Compilation. An iterative peephole optimization algorithm to greedily rewrite regions of the main circuit according to the exhaustively found equivalence classes of up to 4 gates on 2 qubits (Fig. 3). After identifying equivalence classes of each possible region, the substitution that is expected to cause the most favorable noise reduction according to the ResNet prediction is selected. This process is repeated until no further substitutions can be made.

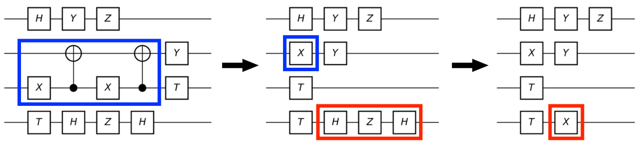


Figure 3: Peephole optimization of a quantum circuit. Precomputed rewrite patterns replace local regions of the circuit to form an equivalent circuit with lower noise. In the example above, the blue and red regions are successively rewritten.

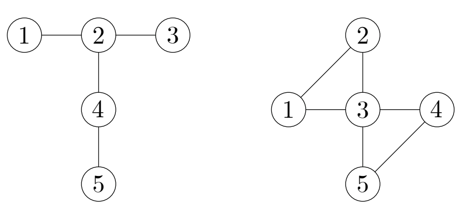
To increase robustness, greedy peephole optimization may be substituted for a scheme such as a *Monte Carlo Tree Search*, increasing the number of circuits explored and involving a larger number of comparisons to average out errors in noise prediction.

3.4 Dynamical decoupling compiler

By first applying the fully optimized IBM Qiskit compiler, we begin with an approximate circuit whose structure is maintained. The deep learning model is trained on circuits with random dynamical decoupling (DD) sequences equivalent to the identity, which are then used to fill gaps of free evolution in the circuit. Thus, gates are added to the result of the IBM Qiskit compiler, minimizing noise according to the learned noise model.

3.4.1 Experimental dataset. We generate random circuits of the form $UU^\dagger$ to ensure an ideal state of $|0\rangle^{\otimes n}$ for an n-qubit circuit. This allows noise to be easily measured as the trace distance between the observed and ideal state by counting the number ones in the bitstring resulting from measuring the output.

The circuits we generated are similar to the recent Google quantum supremacy experiment [3]. The circuits consist of multiple *cycles*. Each cycle is a layer of single-qubit gates and a two-qubit gate. The single-qubit gates are selected randomly from $\{\sqrt{X}, \sqrt{W}, \sqrt{Z}\}$, where $W = (X + Y)/\sqrt{2}$; the two-qubit gate is a CX gate between an arbitrary pair of qubits, despite the limited connectivity of the 5-qubit IBM Q Burlington architecture (Fig. 4).



(a) T-shape (b) Bowtie

Figure 4: Two 5-qubit IBM Q connectivity architectures, corresponding (a) Burlington, Essex, London, Ourense, Vigo and (b) Yorktown. Lines indicate possible two-qubit gate connections. To implement a single CX gate between an arbitrary pair of qubits, multiple qubits and two-qubit gates are often required to be run due to partial connectivity.

We generate 1,000 random 5-qubit circuits with 5 cycles each[3]. Each circuit is then compiled to full optimization with the IBM Qiskit compiler (`optimization_level=3`) optimized to minimize noise on the IBM Q Burlington device. This yields a circuit expected to have the lowest possible noise, both by relabeling qubits to apply fewer operations to noisier couplings and by rewriting the circuit into an approximate form that is expected to result in a state closer to the ideal state. All gates are expressed in the native gate basis $\{U_1, U_2, U_3, CX\}$ of the IBM Q devices, where the single-qubit gates U_i are parameterized.

Random dynamical decoupling (DD) sequences are constructed from U_3 gates. To generate an l-length DD sequence, an $(l-1)$-length sequence of U_3 gates is generated with random parameters. The inverse of this $(l-1)$-length sequence is then compiled to a single gate. If more than one gate results from the compilation, the DD sequence is rejected; otherwise, the l gates are accepted as a valid DD sequence that is equivalent to the identity. For each of the 1,000 random circuits, we generate an additional 15 circuits that include random DD sequences in all gaps (Fig. 5).

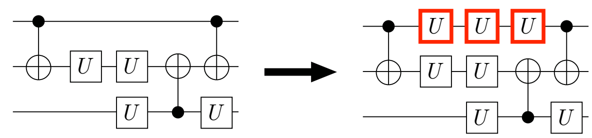


Figure 5: Dynamical decoupling compilation of a quantum circuit. A fully optimized circuit outputted by the IBM Qiskit compiler (left) is padded with a random sequence equivalent to the identity (right), i.e. the gates in red multiply to the identity.

3.4.2 Noise model. As in the general paradigm, we train a neural network on pairs of equivalent circuits and predict the difference in noise from the IBM Q Burlington dataset. Working in the native gate set of dimension 4, we encode each topologically sorted circuit as a 4-channel image and input circuit pairs to the network during training similarly to Fig. 2.

Since the dataset is smaller and more prone to overfitting, we use a smaller CNN with only two convolutional layers (5×5 and 3×3 filters) and three fully-connected layers for the circuits. A step decay is used to schedule learning rates, and batch size is optimized for stable training. Early stopping on a validation set is used to determine the best network.

3.4.3 Compilation. As in dataset generation, random DD sequences are generated from parameterized U_3 gates to fill all gaps in the circuit (Fig. 5). A tournament selection is held between pairs of circuits to determine the circuit with the lowest expected noise out of 1,000 candidate circuits.

4 RESULTS AND CONTRIBUTIONS

We present simulated results for the general compiler and experimental results from IBM Q 5-qubit devices for the dynamical decoupling compiler, summarized in Table 1.

[3]Since we're running $UU^\dagger$ on the device and each cycle consists of two gates, the overall depth of the circuit we run is 20.

Table 1: Comparison of deep learning noise model and gate count heuristic performance. *General compilation* indicates the proposed rewrite of the entire circuit via peephole optimization, evaluated on simulated data. *Random DD* indicates the insertion of random dynamical decoupling sequences in gaps. *DD compilation* indicates the insertion of dynamical decoupling sequences according to the noise model trained on IBM Q Burlington. Boldface row indicates the device for which the DD compiler model performed best, which also corresponds to the device whose noise model was learned.

Proposed compiler	Traditional compiler	Device	Noise improvement [95% CI]
General compilation	Gate minimization	Simulated	15% [6%, 21%]*
Random DD	IBM Qiskit	Burlington	4.5% [4.4%, 4.6%]
DD compilation	**IBM Qiskit**	**Burlington**	**11% [10%, 12%]**
DD compilation	IBM Qiskit	Essex	6% [4%, 7%]
DD compilation	IBM Qiskit	London	6% [5%, 6%]
DD compilation	IBM Qiskit	Ourense	6% [5%, 7%]
DD compilation	IBM Qiskit	Vigo	5% [4%, 6%]
DD compilation	IBM Qiskit	Yorktown	6% [6%, 8%]†

* Noise measured by energy distance. All other rows use trace distance.
† 5-qubit device has bowtie architecture. All other devices have T-shaped architectures.

4.1 General compiler

Greedy noise minimization with deep learning is found to be 15% ± 6% (95% CI) more effective than with gate count minimization on a random sample of 250 circuits, significantly outperforming the standard method of noise reduction in the literature. However, we note that gate count minimization is not the state-of-the-art in circuit compilation, because it preserves the true ideal output state. The fully optimized IBM Qiskit compiler yields an approximate circuit that is sufficiently less noisy to output a state closer to the ideal state. Having established a general paradigm for deep learning compilers, we now propose a compiler that specifically aims to *increase* gate count to reduce noise.

4.2 Dynamical decoupling compiler

We generated 500 circuits outside the training set, which were compiled to full optimization in IBM Qiskit. We then compiled all circuits according to the noise model learned on IBM Q Burlington, and evaluated both the Qiskit and deep learning compilations on all available 5-qubit IBM Q devices. The deep learning DD compiler is shown to outperform the IBM Qiskit compiler on all devices (Table 1). We also include a benchmark of the application of random DD sequences in circuit gaps on the Burlington device.

The DD compiler performs significantly better on IBM Q Burlington — reducing noise around twice as effectively as on other devices — corresponding to the device on which the noise model was learned. Although noise is improved osn all devices, this discrepancy in hardware suggests that a device-specific noise model was indeed learned, instead of simply recognizing states that benefit from being preserved by DD sequences. Additionally, significant improvement is observed over the insertion of random DD sequences, confirming that the compiler optimized the selection of DD sequences.

4.3 Conclusions and future work

Compared to the IBM Qiskit compiler and widely used heuristics such as gate count minimization, we find that deep learning provides significant improvement in noise reduction through learning a hardware-specific noise model. Specifically, we've seen that our approach leads to an 11% reduction in noise on IBM's 5-qubit devices. Our results suggest that deep learning may substantially improve noise mitigation on real-world quantum hardware, addressing a major obstacle in the applicability of NISQ devices.

We think this work opens up several exciting directions for future research. First, it would be interesting to see what information about the noise model can be recovered from the trained CNN. Currently, we are using the CNN as a black box for circuit compilation, however it would be useful to understand how effects such as cross-talk and non-Markovianity are represented within the network itself. Secondly, while dynamical decoupling was already known as a way of mitigating the effects of noise, we would like to see whether the general machine learning approach can provide us with new techniques for noise mitigation. Thirdly, for this framework to be useful in practice, we would need to scale-up this approach and perform tests on larger devices, allowing circuits of arbitrary width and depth to be compiled with a single noise model trained on a given device. Finally, it is worth exploring whether other machine learning techniques (for instance, reinforcement learning) are useful for addressing the noise-minimization problem.

ACKNOWLEDGMENTS

We wish to thank Professor Thomas Vidick of Caltech, who provided helpful comments and support for this project.

REFERENCES

[1] Nabila Abdessaied, Mathias Soeken, and Rolf Drechsler. 2014. Quantum Circuit Optimization by Hadamard Gate Reduction. In *Reversible Computation*, Shigeru Yamashita and Shin-ichi Minato (Eds.). Springer International Publishing, Cham, 149–162.

[2] Carole Addis, Francesco Ciccarello, Michele Cascio, G Massimo Palma, and Sabrina Maniscalco. 2015. Dynamical decoupling efficiency versus quantum non-Markovianity. *New Journal of Physics* 17, 12 (2015), 123004.

[3] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (2019), 505–510.
[4] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (2019), 505–510.
[5] Moritz August and Xiaotong Ni. 2017. Using recurrent neural networks to optimize dynamical decoupling for quantum memory. *Physical Review A* 95, 1 (2017), 012335.
[6] Harrison Ball, Michael J Biercuk, Andre Carvalho, Rajib Chakravorty, Jiayin Chen, Leonardo A de Castro, Steven Gore, David Hover, Michael Hush, Per J Liebermann, et al. 2020. Software tools for quantum control: Improving quantum computer performance through noise and error suppression. *arXiv preprint arXiv:2001.04060* (2020).
[7] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J Bremner, John M Martinis, and Hartmut Neven. 2018. Characterizing quantum supremacy in near-term devices. *Nature Physics* 14, 6 (2018), 595–600.
[8] Heinz-Peter Breuer, Elsi-Mari Laine, Jyrki Piilo, and Bassano Vacchini. 2016. Colloquium: Non-Markovian dynamics in open quantum systems. *Reviews of Modern Physics* 88, 2 (2016), 021002.
[9] Lu-Ming Duan and Guang-Can Guo. 1999. Suppressing environmental noise in quantum computation through pulse control. *Physics Letters A* 261, 3-4 (1999), 139–144.
[10] Gerhard W. Dueck, Anirban Pathak, Md. Mazder Rahman, Abhishek Shukla, and Anindita Banerjee. 2018. Optimization of Circuits for IBM's Five-Qubit Quantum Computers. *2018 21st Euromicro Conference on Digital System Design (DSD)* (2018), 680–684.
[11] Gadi Aleksandrowicz et al. 2019. Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2562110
[12] Genko T Genov, Daniel Schraft, Nikolay V Vitanov, and Thomas Halfmann. 2017. Arbitrarily accurate pulse sequences for robust dynamical decoupling. *Physical review letters* 118, 13 (2017), 133202.
[13] Robin Harper, Steven T Flammia, and Joel J Wallman. 2019. Efficient learning of quantum noise. *arXiv preprint arXiv:1907.13022* (2019).
[14] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90
[15] Chingiz Kabytayev, Todd J Green, Kaveh Khodjasteh, Michael J Biercuk, Lorenza Viola, and Kenneth R Brown. 2014. Robustness of composite pulses to time-dependent control noise. *Physical Review A* 90, 1 (2014), 012316.
[16] Aleks Kissinger and John van de Wetering. 2019. Reducing T-count with the ZX-calculus. *arXiv preprint arXiv:1903.10477* (2019).
[17] Vadym Kliuchnikov and Dmitri Maslov. 2013. Optimization of Clifford circuits. *Phys. Rev. A* 88 (Nov 2013), 052307. Issue 5. https://doi.org/10.1103/PhysRevA.88.052307
[18] S. Lathuiliére, P. Mesejo, X. Alameda-Pineda, and R. Horaud. 2019. A Comprehensive Analysis of Deep Regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), 1–1. https://doi.org/10.1109/TPAMI.2019.2910523
[19] Daniel A Lidar. 2014. Review of decoherence free subspaces, noiseless subsystems, and dynamical decoupling. *Adv. Chem. Phys* 154 (2014), 295–354.
[20] Joshua Morris, Felix A Pollock, and Kavan Modi. 2019. Non-Markovian memory in IBMQX4. *arXiv preprint arXiv:1902.07980* (2019).
[21] Yunseong Nam, Neil J. Ross, Yuan Su, Andrew M. Childs, and Dmitri Maslov. 2018. Automated optimization of large quantum circuits with continuous parameters. *npj Quantum Information* 4, 1 (2018), 23. https://doi.org/10.1038/s41534-018-0072-4
[22] Michael A Nielsen and Isaac Chuang. 2002. Quantum computation and quantum information.
[23] Bibek Pokharel, Namit Anand, Benjamin Fortman, and Daniel A Lidar. 2018. Demonstration of fidelity improvement using dynamical decoupling with superconducting qubits. *Physical review letters* 121, 22 (2018), 220502.
[24] Felix A Pollock, César Rodríguez-Rosario, Thomas Frauenheim, Mauro Paternostro, and Kavan Modi. 2018. Operational Markov condition for quantum processes. *Physical review letters* 120, 4 (2018), 040405.
[25] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2 (2018), 79.
[26] Gregory Quiroz and Daniel A Lidar. 2013. Optimized dynamical decoupling via genetic algorithms. *Physical Review A* 88, 5 (2013), 052306.
[27] Ángel Rivas, Susana F Huelga, and Martin B Plenio. 2014. Quantum non-Markovianity: characterization, quantification and detection. *Reports on Progress in Physics* 77, 9 (2014), 094001.
[28] Maria L. Rizzo and Gábor J. Székely. 2016. Energy Distance. *WIREs Comput. Stat.* 8, 1 (Jan. 2016), 27–38. https://doi.org/10.1002/wics.1375
[29] Alexandre M Souza, Gonzalo A Alvarez, and Dieter Suter. 2011. Robust dynamical decoupling for quantum computing and quantum memory. *Physical review letters* 106, 24 (2011), 240501.
[30] Lorenza Viola, Emanuel Knill, and Seth Lloyd. 1999. Dynamical decoupling of open quantum systems. *Physical Review Letters* 82, 12 (1999), 2417.
[31] Lorenza Viola and Seth Lloyd. 1998. Dynamical suppression of decoherence in two-state quantum systems. *Physical Review A* 58, 4 (1998), 2733.
[32] Paolo Zanardi. 1999. Symmetrizing evolutions. *Physics Letters A* 258, 2-3 (1999), 77–82.