



**HAL**  
open science

# Effects of Dynamical Decoupling and Pulse-Level Optimizations on IBM Quantum Computers

Siyuan Niu, Aida Todri-Sanial

► **To cite this version:**

Siyuan Niu, Aida Todri-Sanial. Effects of Dynamical Decoupling and Pulse-Level Optimizations on IBM Quantum Computers. *IEEE Transactions on Quantum Engineering*, 2022, 3 (3102510), pp.1-10. 10.1109/TQE.2022.3203153 . hal-03645284

**HAL Id: hal-03645284**

**<https://hal.science/hal-03645284>**

Submitted on 17 Oct 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Effects of Dynamical Decoupling and Pulse-level Optimizations on IBM Quantum Computers

Siyuan Niu\* and Aida Todri-Sanial†  
LIRMM, University of Montpellier, France  
(Dated: September 5, 2022)

Currently available quantum computers are prone to errors. Circuit optimization and error mitigation methods are needed to design quantum circuits to achieve better fidelity when executed on NISQ hardware. Dynamical decoupling (DD) is generally used to suppress the decoherence error and different DD strategies have been proposed. Moreover, the circuit fidelity can be improved by pulse-level optimization, such as creating hardware-native pulse-efficient gates. This paper implements all the popular DD sequences and evaluates their performances on IBM quantum chips with different characteristics for various well-known quantum applications. Also, we investigate combining DD with pulse-level optimization method and apply them to QAOA to solve Max-Cut problem. Based on the experimental results, we found that DD can be a benefit for only certain types of quantum algorithms, while the combination of DD and pulse-level optimization methods always has a positive impact. Finally, we provide several guidelines for users to learn how to use these noise mitigation methods to build circuits for quantum applications with high fidelity on IBM quantum computers.

## I. INTRODUCTION

Quantum computing is rapidly growing in recent years and various technologies have been developed for different quantum platforms, the leading candidates being superconducting and trapped-ion devices. Several companies such as IBM, Rigetti, and IonQ provide publicly available cloud-based services which allow users to access their platforms remotely. But, today's quantum computers are still prone to errors caused by either unavoidable interactions with the environment or imperfect quantum controls. They are qualified as Noisy Intermediate-Scale Quantum (NISQ) computers [1]. The largest quantum chip to date has 127 qubits released by IBM [2]. Quantum error correction (QEC) has been proposed to eliminate the noise impact and help achieve a fault-tolerant quantum device [3, 4]. However, the implementation of quantum error correction codes requires a large number of ancilla qubits, which is not feasible on current hardware. Therefore, alternative approaches are needed to tackle the noise issue.

In order to reduce the noise impact and improve the quantum circuit fidelity, several quantum software were designed and they have made a great contribution to different circuit design processes, such as circuit synthesis [5, 6], or qubit mapping [7, 8]. In addition, quantum error mitigation (QEM) was introduced for error suppression on NISQ devices [9]. There are different QEM techniques, such as readout error mitigation [10], dynamical decoupling (DD) [11], crosstalk mitigation [12, 13], zero-noise extrapolation (ZNE) [14], etc. Most methods require supplementary circuit executions to build the error map for mitigation. Whereas DD is one of the simplest

strategies, which aims at mitigating the decoherence error (also called idle error) without any circuit overhead, and is the focus of this paper. The thrust of DD is to insert periodically a series of pulses to the idle qubits and return the qubits to their original states. There is a high probability to have idle qubits during execution due to the variation of gate latencies and limited parallelism caused by the anti-commutative gates. It has been shown in [15] that the idle qubits can be almost 10 times more subject to errors if adjacent two-qubit operations are executing at the same time on IBM superconducting device. DD plays an important role in reducing the idle error and has been used in quantum volume experiments [16], noise spectrum characterization [17], and decoherence-protected quantum gate implementation [18], etc. Moreover, there are different DD strategies, such as Hahn echo [19], CPMG [20], XY4 [21], robust KDD [22], etc. They have diverse impacts on decoherence error suppression for different quantum devices. It has been shown in [23] that DD is able to extend the lifetime of one-qubit states as well as entangled two-qubit states for IBM and Rigetti devices using XY4 sequence. But this sequence is shown to be vulnerable to experimental imperfections, while the use of robust DD was demonstrated to be capable of correcting the pulse errors on Rigetti devices [24].

Recently, IBM released Qiskit Pulse [25, 26] allowing users to design and customize the gate pulse implementations. Some works have already attempted to optimize the pulse controls and reduce the pulse durations with the help of Qiskit Pulse [27–29]. However, such method often needs additional calibrations which is time-consuming and requires a deep familiarity with quantum control. In [30, 31], the authors proposed a new technique to create more hardware-native pulse-efficient gates, which improves the gate fidelity without the overhead of extra calibrations.

So far, the aforementioned techniques were only tested separately for limited benchmarks and quantum hard-

---

\* siyuan.niu@lirmm.fr

† aida.todri@lirmm.fr

ware. Several questions still remain unclear: (1) What are the impacts of different DD sequences on specific quantum algorithms? (2) For a certain benchmark, does the impact of different DD sequences vary across different quantum chips? (3) Will the combination of DD and pulse-level optimization methods further improve the circuit fidelity? In our work, we address these questions and our main contributions can be listed as follows:

- We explore the performance of different DD sequences on various quantum applications and evaluate the experiments on several IBM devices with different qubit numbers and quantum volumes. To the best of our knowledge, this is the first attempt to illustrate the behavior of applying robust KDD sequence to IBM quantum devices.
- We combine the DD technique with pulse-efficient optimization method to demonstrate their benefits on Quantum Approximate Optimization Algorithm (QAOA) to solve Max-Cut problem.
- Based on the experimental results, we provide guidelines and insights for users to apply application-oriented dynamical decoupling and pulse-level optimization techniques.

## II. BACKGROUND

### A. Dynamical decoupling

Dynamical decoupling is widely used in suppressing the decoherence error by reducing the interaction between the system and the environment. Considering a system-bath Hamiltonian  $H$  shown in (1),  $H_s$  and  $H_B$  being the Hamiltonian of the system and the bath respectively, and the interacting term being  $H_{SB}$  (see (2)), where  $\sigma_i^\alpha$  is the Pauli matrix acting on qubit  $i$ ,  $B_i^\alpha$  is the operator of the environment, and  $\alpha \in \{x, y, z\}$ . DD aims at reducing the impact of system-environment interaction and various DD protocols have been developed to improve the performance of quantum computers. Here, we review the main strategies of DD implementations and a summary of these DD sequences is shown in Table I.

$$H = H_s + H_B + H_{SB} \quad (1)$$

$$H_{SB} = \sum_{i=1}^N \sum_{\alpha \in \{x, y, z\}} \sigma_i^\alpha \otimes B_i^\alpha \quad (2)$$

- **Hahn echo.** The spin-echo sequence [19] is used to reduce the inhomogeneous effects from the environmental magnetic field. It applies a  $\pi$  pulse to the spin system to inverse the spins after a period

TABLE I: A summary of different DD sequences.

DD sequences	Pulse implementation
Hahn echo [19]	$X$ or $Y$
CP/CPMG [20, 32]	$(X)^n$ or $(Y)^n$
XY4 [21]	$(X - Y - X - Y)^n$
XY8 [33]	$(XY4 - \overline{XY4})^n$
XY16 [33]	$(XY8 - \overline{XY8})^n$
UDD [34]	$(X)^n$ or $(Y)^n$ with space $\tau$ where $\tau = \sin^2(\frac{\pi j}{2n+2}), j \in \{1, 2, \dots, n\}$
KDD [22]	$(\pi)_{\frac{\pi}{6}+\phi} - (\pi)_\phi - (\pi)_{\frac{\pi}{2}+\phi} - (\pi)_\phi - (\pi)_{\frac{\pi}{6}+\phi}$ where $(\theta)_\phi = e^{\frac{i\theta}{2}[\cos(\theta)\sigma_x + \sin(\theta)\sigma_y]}$ , $\phi = 0, \frac{\pi}{2}, 0, \frac{\pi}{2}$

$n$ : repetition time of the basic DD cycle.

$(\theta)_\phi$ : a rotation of  $\theta$  around the axis defined by  $\phi$ .

of time  $t$  and let the system be refocused during the same duration. At that time,  $T_2^*$  effects are removed and we can obtain a  $T_2$  echo.

- **CP and CPMG.** Carr and Purcell proposed a series of  $\pi$  pulses separated by a constant interval, known as the CP sequence, to further reduce the effect of self-diffusion in the inhomogeneous field [32]. But, extra pulses can introduce more errors and destroy the state of the system. Therefore, Meiboom and Gill improved the CP pulse by developing the CPMG sequence [20], retaining the CP pulse but introducing an additional phase shift in the first pulse to compensate the pulse errors.
- **UDD.** Hahn echo and CP/CPMG are equidistant pulses. While UDD aims at optimizing the  $\pi$  sequence based on CP/CPMG by varying the intervals between each pulse [34]. It is proven to be the optimal pulse to suppress low frequency noise and is insensitive to thermal fluctuations. It outperforms the equidistant DD sequences especially for systems whose spectral densities have high frequencies with sharp cutoff. For other more general cases, equidistant DD sequences were shown to perform better [35].
- **XY4.** All the DD sequences above only rotate around one single axis. They are exclusively useful when the system-environment interaction is orthogonal to the rotation axis. XY4 is the simplest DD sequence to generally suppress system-environment interaction along three directions [21]. It inserts alternatively  $\pi$  rotations around X and Y axes independently of the initial state.
- **XY8 and XY16.** DD can sometimes detriment the fidelity due to the accumulation of errors caused by pulse imperfections. One approach to reducing the errors is to combine the basic sequence with its inverse for self-correction so that the unwanted

terms can be canceled [33]. XY4 is usually chosen as the basic cycle. XY8 is composed of XY4 sequence and its inverse while XY16 contains XY8 and its inverse.

- **KDD.** The other approach to avoid pulse imperfections is to replace each pulse of a DD sequence by a robust composite pulse, which is designed to generate ideal rotations even if there exist pulse imperfections [36]. KDD constructs a DD block using a 5-pulse composite  $\pi$  pulse, and combine two of such DD block with and without a phase shift  $\pi/2$  as  $(KDD_\phi, KDD_{\phi+\pi/2})^2$  [22]. It is composed of 20 pulses in total.

DD has been demonstrated to have the capability to mitigate the decoherence errors on IBM and Rigetti platforms using XY4 sequences [23]. The experiment is performed as follow: it prepares different initial states by varying the angles of the rotation gates, inserts XY4 sequences, and compares the difference between input and output states. The impacts of different DD sequences on Rigetti device are reported in [24]. A large number of DD sequences are evaluated, including XY4, XY8, KDD, etc., and quantum process tomography is used to characterize the evolution. KDD is shown to be the most robust pulse sequence against pulse imperfections. Both papers evaluate DD sequences on simple benchmarks instead of real quantum applications, and there is no experiment to illustrate applying KDD to IBM quantum devices. Moreover, the experiments in [15] show that the naive implementation of DD (inserting DD sequences to all the idle qubits when it is possible) can not always improve the circuit fidelity. Therefore, Das et al. proposed an Adaptive Dynamical Decoupling framework to estimate the DD impact for each circuit and adjust DD sequence to ensure it improves the circuit fidelity [15]. This method achieves fidelity improvement but introduces a large overhead of DD impact characterization for a given application. Also, Ravi et al. proposed VAQEM [37], an approach that dynamically inserts DD sequences for variational algorithm with the overhead of tuning DD sequence. In our paper, instead of carefully adjusting DD sequences for better error mitigation with extra circuit overhead, we exhaustively evaluate DD sequences on extensively used quantum applications, to give a high-level idea about if general DD techniques can really be beneficial for these applications. It is also the first attempt to implement KDD on IBM devices to check its performance. The experiments are performed on various IBM devices with different qubit numbers and quantum volumes to provide general insights about application-oriented DD noise mitigation on IBM quantum devices.

## B. Pulse-efficient technique

*CNOT* gate is the only two-qubit operation included in the basis gates for IBM quantum devices and its cal-

ibrations are provided through the IBM quantum dashboard. It is implemented by a  $R_{ZX}(\frac{\pi}{2})$  gate along with some single-qubit gates [38] on the device.  $R_{ZX}$  gate is realized by the echoed cross-resonance gate [39], which is specific for the IBM fixed-frequency superconducting transmon qubit device due to its low overhead and high fidelity. When executing a circuit on IBM quantum hardware, every multi-qubit operation needs to be transpiled to the *CNOT* basis which is not flexible and less efficient.

Therefore, [30, 31] proposed a pulse-efficient circuit transpilation framework and has shown to achieve higher fidelity than *CNOT* basis transpilation for certain benchmarks. Instead of only using a  $R_{ZX}(\theta)$  with a fixed angle  $\theta = \frac{\pi}{2}$  for *CNOT* gate, a flexible echoed  $R_{ZX}$  gate is implemented with arbitrary angle and  $R_{ZX}(\theta) = XR_{ZX}(-\frac{\theta}{2})XR_{ZX}(\frac{\theta}{2})$  to enable transpilation to  $R_{ZX}$  basis. This method does not require any additional pulse calibration since the calibration of  $R_{ZX}$  gate can be easily calculated from the *CNOT* calibrations.

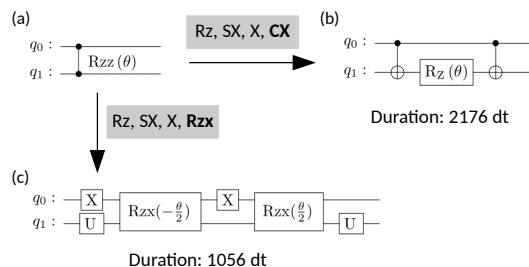


FIG. 1: (a) Original  $R_{ZZ}$  gate with  $\theta = \frac{\pi}{3}$ . The circuit is mapped to the first two qubits of IBM Q 27 Toronto.

(b) The transpilation result with basis gate sets  $\{Rz, SX, X, CX\}$ , and the duration is 2176 dt. dt is the sampling time for Qiskit pulses and is set to 0.2 ns.

(c) The transpilation result with basis gate sets  $\{Rz, SX, X, R_{ZX}\}$ , and the duration is 1025 dt. We replace  $R_{ZX}(\frac{m\pi}{2})\sqrt{X}R_{ZX}(\frac{n\pi}{2})$  by  $U$  to simplify the circuit.  $m, n \in \mathbb{R}$  and can be varied across different transpilation passes.

An example of transpiling  $R_{ZZ}$  gate to two different basis gate sets is shown in Fig. 1. We first transpile the  $R_{ZZ}$  gate to the IBM *CNOT*-basis gate sets (Fig. 1(b)) and then transpile again the original circuit to  $R_{ZX}$ -basis gate sets (Fig. 1(c)). The duration of the second circuit is reduced by 51.1% compared to the first gate sets. This technique exposes the echo of the cross-resonance gates which enables at most one single-qubit gate between each non-echoed  $R_{ZX}$  gate so that it can shorten the total circuit duration.

Quantum algorithms requiring a lot of two-qubit control-rotation gates, such as  $R_{ZZ}$  or  $R_{YY}$ , which can be directly compiled to  $R_{ZX}$  gates along with some single-qubit gates, are particularly benefited from this pulse-efficient transpilation framework, for example QAOA [40], quantum hamiltonian simulation [41], etc.

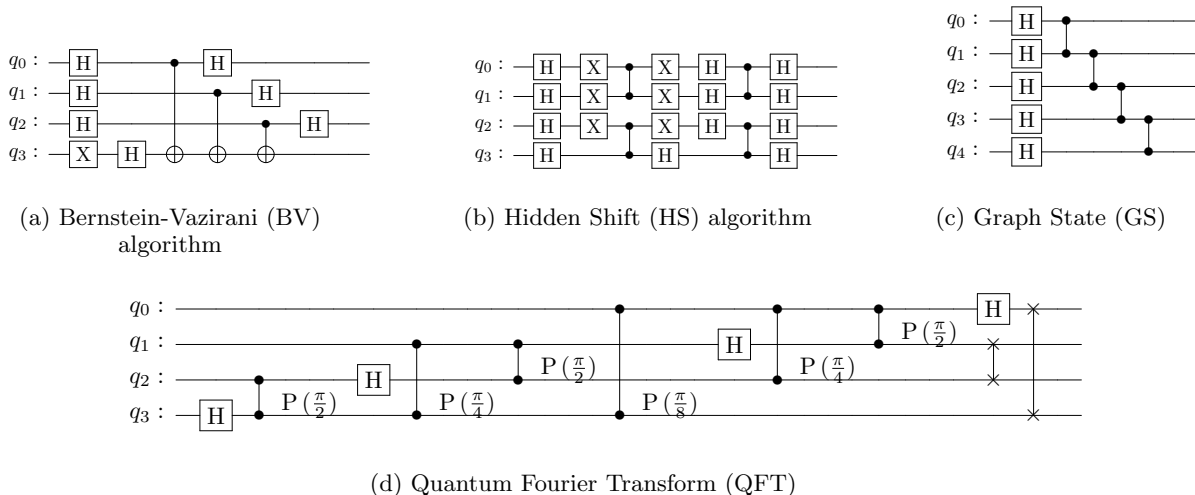


FIG. 2: Examples of different quantum circuits. Note that, the Graph State circuit is constructed linearly based on the target hardware topology.

### III. METHODS

We construct DD sequences according to Table I, and  $n$  is the number of repetition time for the basic DD pulse. For Hahn echo, we insert one  $X$  or  $Y$  gate during the idle time. We set  $n = 2$  for CP/CPMG, CP (resp. CPMG) being implemented as the sequence  $X - X$  (resp.  $Y - Y$ ). For XY4, XY8, and XY16,  $n$  is set to 1 so that the sequence of pulses is inserted once. All the aforementioned DD sequences are equidistant and symmetrical: there is a delay of  $\tau/2$  at the beginning and the end of the idle time, and a delay of  $\tau$  between each DD pulse. Whereas UDD is composed of non-equidistant pulses, it acts differently from CPMG for all  $n > 2$ . As there is no clear conclusion about which repetition number  $n$  we should pick to have the best performance of UDD on IBM quantum devices, we set  $n = 8$  as demonstrated in Qiskit tutorial. The repetition gate is set to  $X$  and  $Y$ , and is marked as UDD\_X and UDD\_Y in the results. For KDD,  $(\theta)_\phi$  represents a rotation of  $\theta$  around the axis defined by  $\phi$ . Since  $\theta_\phi = R_Z(\phi)R_X(-\theta)R_Z(-\phi)$ , we insert 60 gates in total, where delay is only introduced between each  $\theta_\phi$ , which means there is no space between the three gates constructing  $\theta_\phi$ . Note that, since  $R_Z$  gate is free on IBM quantum devices, KDD is translated to “only” 20 pulses in the end.

First, we evaluate the DD effects on IBM quantum computers by applying different DD sequences to various well-known quantum applications, including Bernstein-Vazirani (BV) algorithm, Hidden Shift (HS) algorithm, Quantum Fourier Transform (QFT), and Graph State (GS). The basis quantum circuit structures of these quantum applications are shown in Fig. 2. BV algorithm implements an oracle function  $f(x)$ , which represents the dot product between  $x$  and a secret string  $s$ , with the objective of finding  $s$ . HS algorithm constructs an oracle

that encodes two functions  $f$  and  $g$ , and there exists a secret string  $s$  such that  $g(x) = f(x + s)$ . The goal of the algorithm is to find  $s$ . QFT is the quantum version of discrete Fourier transform, and is the essential part for many other quantum algorithms, such as Shor’s algorithm [42], quantum phase estimation algorithm [43], etc. Graph State is a quantum state prepared based on a graph. Specifically, we can build a Graph State according to a given hardware topology, where there is an edge on the graph when the two qubits are coupled, and the connection is represented by a  $CZ$  gate. It can entangle all the qubits of the device and is important for error correction.

Second, we evaluate the performance of combining different DD sequences with pulse-efficient transpilation technique. We apply them to QAOA to solve the Max-Cut problem, since QAOA ansatz is composed of  $R_{ZZ}$  gates along with some single-qubit gates, which can be profited from the pulse-efficient method. We generate randomly 3-regular graphs using ReCirq [44] and random graphs using Networkx with different degrees as our benchmarks. For example, we use *3-reg-4* to represent a 3-regular graph with degree of 4 (qubits) and *rand-4-0.5* for a random graph with 4 nodes/qubits and a probability for edge creation of 0.5 (see Fig. 3).

We choose the following metrics for different benchmarks to demonstrate the impact of DD and pulse-level optimization technique on application fidelity.

- Probability of Successful Trial (PST) [45]. This metric is defined by the ratio of the number of trials that give the expected result to the total number of trials, and higher is better. It is dedicated to the benchmarks with one certain correct result, such as BV algorithm and HS algorithm.
- Jensen-Shannon Divergence (JSD). It is used to

measure the similarity between two probability distributions, and lower is better. It is suitable for Graph State circuit and QFT whose output is a distribution.

- Approximation ratio [46]. It is specifically designed to evaluate the performance of QAOA circuit and is defined as  $\langle C \rangle / C_{min}$ , where  $\langle C \rangle$  is the expectation value obtained by the quantum computer and  $C_{min}$  is calculated by the classical *NumPyMinimumEigensolver*. We aim at maximizing the approximation ratio and 1 means a perfect solution.

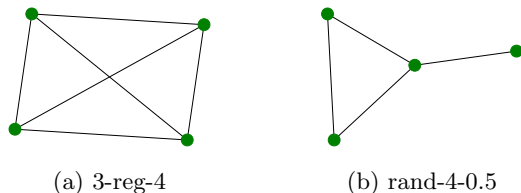


FIG. 3: An example of (a) 3-regular graph with degree of 4, (b) random graph with degree of 4 and edge creation probability of 0.5.

TABLE II: The information of different IBM Q devices.

IBM Q	Jakarta	Guadalupe	Toronto	Montreal
$n$	7	16	27	27
QV	16	16	32	128

#### IV. EXPERIMENTAL RESULTS

We perform the experiments on various IBM quantum devices with different qubit numbers and quantum volumes (QV) (see Table II). The size of the benchmarks varies depending on the quantum device. For the first experiment when evaluating the effects of applying DD sequences to different applications including BV, HS, QFT, and GS, the circuit size varies from 3 to 6 for IBM Q 7 Jakarta. Whereas for other devices, the circuit size changes from 3 to 12, since there are too many noises accumulating to obtain meaningful results for benchmarks with more than 12 qubits. When exploring the performance of combining DD and pulse-efficient optimization technique on QAOA for the second experiment, the degree of the two types of graphs (3-regular graph and random graph) ranges from 4 to 6 for IBM Q 7 Jakarta, and 4 to 12 for other devices. The QAOA ansatz has one layer with parameters initialized randomly and optimized using *COBYLA* optimizer on the simulator. We only execute the final ansatz with optimized parameters on the quantum hardware. All the benchmarks are compiled using *Qiskit* with the highest optimization level and executed 8192 times.

There are some limitations when inserting certain DD pulses to the idle time. If the DD sequences contain a large number of pulses, such as XY16 and KDD, it might not be possible to insert them to some small benchmarks whose idle time might not be long enough. Moreover, Hahn echo can only be applied if the inverse of the inserted single  $X$  or  $Y$  gate is able to be absorbed into the neighboring gates to ensure the equivalence of the quantum state. Therefore, we check if the DD pulses are actually inserted to the benchmark for each experiment, and we remove the circuit without any DD pulses inserted from the result.

In order to clearly show the impact of various DD sequences and pulse-efficient optimization method, we use the relative results for the three metrics. The original benchmark with no optimization method applied is marked as the baseline. PST is divided by the baseline and we use JSD results to divide the baseline, so that we can obtain the relative results when applying DD sequences. If the relative result is larger than one, it means that there is an improvement, and larger is better. Whereas for QAOA experiments, we use the difference between approximation ratio with DD sequence or pulse-efficient technique and the baseline as the relative result due to the possible negative value of the approximation ratio. If the difference is larger than zero, it indicates an enhancement and also larger is better. Each experiment has been repeated three times and results with similar trends were obtained. We show the average of the three experiments for all the results.

We first demonstrate applying DD sequences to various quantum applications. Hahn echo is not applicable for all the benchmarks. XY16 and KDD are too long to insert for certain small BV circuits (less than 4 qubits). All the DD sequences cannot be applied to HS algorithm, since the duration of the idle time is always equal to the duration of one single-qubit gate. Thus, the idle time is too short to fit any DD sequences. The relative PST results for BV circuits, the relative JSD results for QFT and Graph State circuits are shown in Fig. 4, Fig. 5, and Fig. 6.

The quantitative analysis of the relative results is shown in Table III. For BV algorithm (see Fig. 4), if the BV circuit involves a large number of qubits, the PST fidelities can be dropped dramatically such that all the PSTs are below 0.1 even with DD applied for error mitigation. Therefore, we only show the results whose PSTs are larger than 0.1. The relative PST results demonstrate that all the DD sequences are able to enhance the BV circuit fidelity, especially UDD\_X. On average, the fidelity is improved by 1.09x, 3.82x, 2.79x, and 1.68x compared with the baseline for IBM Q Jakarta, Guadalupe, Toronto, and Montreal respectively. Whereas for QFT, inserting DD sequence is more favorable for circuits with more than 5 qubits. Therefore, the relative JSD results on IBM Q 7 Jakarta are not encouraging. Overall, DD can be beneficial for QFT circuits but are not as stable as for BV circuits. The fidelity is improved by 1.14x, 1.18x,

and 1.28x on average for IBM Q Guadalupe, Toronto, and Montreal. For Graph State circuit, the performance of DD is also not always steady and different across IBM quantum devices. DD is helpful for Graph State circuits on IBM Q Guadalupe but not for other devices, with an increase of fidelity by 1.17x on average. The variance of the results for different DD techniques becomes smaller for relatively large-scale QFT and Graph State circuits which might be due to the accumulating errors.

To conclude, DD can always improve the fidelity for BV algorithm to different extents across IBM quantum chips. But for other applications, the performance of DD remains uncertain and can be different on various quantum devices. Moreover, for certain benchmark like HS algorithm, the idle time is always short so that DD is not applicable regardless of the circuit size.

Second, we show the results of using QAOA to solve Max-Cut problem for different graphs by employing: (1) diverse DD sequences (see Fig. 7), (2) DD sequences + pulse-efficient optimization technique (see Fig. 8). The quantitative analysis of the relative results is shown in Table IV. Most of the DD sequences can help enhance the approximate ratio for QAOA, especially CP sequence. The approximate ratio of all the graphs is improved by 6.8%, 5.9%, 3.7%, and 3.7% on average for IBM Q Jakarta, Guadalupe, Toronto, and Montreal respectively. There is no clear relationship between the level of improvement and the size of the benchmark. Comparing to the baseline circuit without applying any error mitigation technique, purely the pulse-efficient method is already able to raise the approximation value by 8.9%, 5.5%, 5.3%, and 4.8% on average for the four devices. Associating pulse-efficient technique with DD sequences can further improve the approximation ratio and the negative impact induced by DD can mostly be canceled. The combination of DD and pulse-efficient technique can improve the approximation ratio by 26.5%, 49.2%, 81.1%, 75.7% compared with only applying DD for the four IBM devices.

One interesting thing is that, even though KDD is a robust DD sequence that is designed to be robust against pulse imperfection and has shown improvements on Rigetti device, its performance is not as good as expected on IBM quantum devices. For all the quantum applications tested including QAOA circuit, there is no growth of circuit fidelity using KDD compared with other DD strategies and actually it decreases the circuit fidelity for most of the cases.

## V. DISCUSSION

### A. Design guideline

It is important to explore different noise mitigation and pulse-level optimization methods, including DD and pulse-efficient optimization technique to better build the quantum circuits. Here, we list several guidelines that

can help the community utilize these methods to design circuits with higher fidelity.

- DD cannot always improve the circuit fidelity and it is highly application-dependent. It is recommended to use DD on specific quantum algorithms, including Bernstein-Vazirani algorithm, and QAOA.
- When applying DD sequences, it is recommended to check the transpiled circuit to verify if DD is suitable to insert.
- The robust sequence KDD does not work well on IBM quantum devices for most of the quantum algorithms tested.
- The pulse-efficient optimization technique is favorable for QAOA circuits and combined with DD, it can improve the approximation ratio.

### B. Future work

As the experiments have demonstrated some discouraging effects when inserting DD sequences for certain quantum applications, exploring the hardware physics behind them becomes important. Application-dependent adaptive dynamical decoupling [15] and variational algorithm-focused VAQEM [37] have been proposed to provide the most beneficial DD sequences but with large overhead and limited analysis of different DD strategies. A smarter approach for designing appropriate application-oriented DD sequences considering hardware physics is left to future works. Recently, DD has been proven to be able to suppress ZZ-crosstalk for fixed frequency transmon superconducting device [47]. It is interesting to further investigate the performance of DD-based crosstalk suppression at application-level, so that it can contribute to the quantum parallel executions [48, 49]. Finally, the pulse-level optimization experiments can be delved to other quantum applications to provide more guidelines, such as VQE [50], quantum simulation [51], etc.

## VI. CONCLUSION

Today's quantum hardware is prone to noise in the NISQ era. Therefore, circuit optimization and error mitigation approaches are required to increase the output fidelity. In this paper, we focus on two pulse-level circuit optimization methods: dynamical decoupling and pulse-efficient optimization technique. First, we implement various DD strategies on IBM quantum devices including non-universal, universal, and robust ones. Second, we apply these DD sequences to several well-known quantum applications, such as QFT and Bernstein-Vazirani algorithm, to evaluate comprehensively the impact of diverse DD techniques on IBM quantum devices with dif-

TABLE III: The relative results of applying various DD sequences to different quantum applications on four IBM Q hardware.

Benchmarks	IBM Q Hardware											
	Jakarta			Guadalupe			Toronto			Montreal		
	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg
BV	1.1 (UDD_X)	1.05 (KDD)	1.09	4.53 (UDD_X)	3.16 (CPMG)	3.82	3.18 (UDD_X)	2.44(XY16)	2.79	1.77(UDD_X)	1.55 (KDD)	1.68
QFT	1.1 (XY16)	0.89 (UDD_Y)	1.01	1.17 (CPMG)	1.06 (UDD_X)	1.14	1.23 (KDD)	1.14 (UDD_X)	1.18	1.33 (XY4)	1.22 (UDD_X)	1.28
GS	1 (CP)	0.9 (UDD_Y)	0.95	1.21 (XY4)	1.06 (UDD_Y)	1.17	1.06 (CPMG)	1 (XY8)	1.03	1.02 (CP)	0.96 (XY16)	0.99
HS	DD is not applicable											

If the relative results  $\geq 1$ , it means that the DD sequence has a positive impact on the fidelity. Otherwise, a negative impact is introduced by DD.

TABLE IV: The relative results of applying different methods, including only DD, only pulse-efficient method (PE), and DD+PE to QAOA circuits on four IBM Q hardware.

Methods	IBM Q Hardware											
	Jakarta			Guadalupe			Toronto			Montreal		
	Max (%)	Min (%)	Avg (%)	Max (%)	Min (%)	Avg (%)	Max (%)	Min (%)	Avg (%)	Max (%)	Min (%)	Avg (%)
Only DD	8.6 (UDD_X)	2.4 (KDD)	6.8	7.4 (CP)	2.5 (KDD)	5.9	4.4 (CP)	2.3 (KDD)	3.7	4.7 (CP)	1.6 (KDD)	3.7
Only PE	8.9			5.5			5.3			4.8		
DD+PE	10.4 (UDD_X)	6 (KDD)	8.6	9.7 (CP)	6.8 (KDD)	8.8	7.2 (XY8)	5.8 (KDD)	6.7	7.4 (CP)	4.5 (KDD)	6.5

If the relative results  $\geq 0$ , it means an improvement on fidelity. Otherwise, a negative impact is introduced on fidelity.

ferent characteristics. We also merge DD with pulse-efficient transpilation method and investigate them on QAOA circuits to solve Max-Cut problem. Based on the experimental results, we found that DD techniques always show positive impact for some benchmarks, while for others, DD demonstrate some discouraging effects. As there is no overhead-free application-oriented DD approach, we provide a list of design guidelines for users to better understand these pulse-level optimization methods and figure out how to improve the circuit design for various quantum applications.

## ACKNOWLEDGMENT

This work is funded by the QuantUM Initiative of the Region Occitanie, University of Montpellier and IBM Montpellier. The authors are very grateful to Adrien Suau for the helpful suggestions and feedback on an early version of this manuscript.

## SUPPLEMENTARY INFORMATION

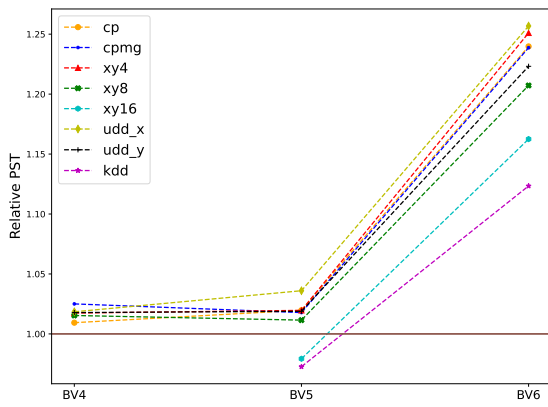
All the scripts to reproduce the experimental results are publicly available in [https://github.com/peachnuts/DD\\_PE](https://github.com/peachnuts/DD_PE).

- 
- [1] J. Preskill, Quantum computing in the nisq era and beyond, *Quantum* **2**, 79 (2018).
- [2] IBM quantum unveiled eagle, a 127-qubit quantum processor. (2021).
- [3] A. R. Calderbank, E. M. Rains, P. Shor, and N. J. Sloane, Quantum error correction via codes over  $GF(4)$ , *IEEE Transactions on Information Theory* **44**, 1369 (1998).
- [4] B. M. Terhal, Quantum error correction for quantum memories, *Reviews of Modern Physics* **87**, 307 (2015).
- [5] O. Di Matteo and M. Mosca, Parallelizing quantum circuit synthesis, *Quantum Science and Technology* **1**, 015003 (2016).
- [6] J. Zhang, J. Vala, S. Sastry, and K. B. Whaley, Optimal quantum circuit synthesis from controlled-unitary gates, *Physical Review A* **69**, 042309 (2004).
- [7] S. Niu, A. Suau, G. Staffelbach, and A. Todri-Sanial, A hardware-aware heuristic for the qubit mapping problem in the nisq era, *IEEE Transactions on Quantum Engineering* **1**, 1 (2020).
- [8] G. Li, Y. Ding, and Y. Xie, Tackling the qubit mapping problem for nisq-era quantum devices, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (2019) pp. 1001–1014.
- [9] S. Endo, S. C. Benjamin, and Y. Li, Practical quantum error mitigation for near-future applications, *Physical Review X* **8**, 031027 (2018).
- [10] S. Bravyi, S. Sheldon, A. Kandala, D. C. McKay, and J. M. Gambetta, Mitigating measurement errors in multiqubit experiments, *Physical Review A* **103**, 042605 (2021).
- [11] A. M. Souza, G. A. Álvarez, and D. Suter, Robust dynamical decoupling, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineer-*

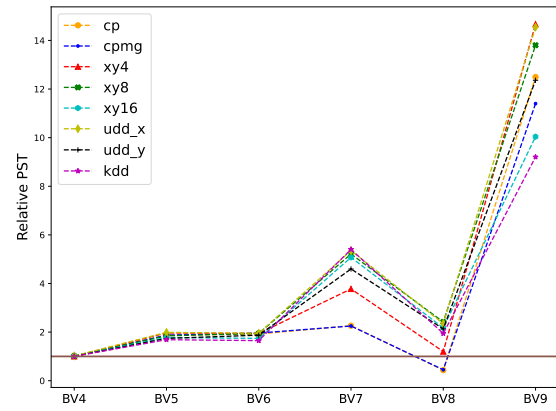


- ing Sciences **370**, 4748 (2012).
- [12] P. Murali, D. C. McKay, M. Martonosi, and A. Javadi-Abhari, Software mitigation of crosstalk on noisy intermediate-scale quantum computers, in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems* (2020) pp. 1001–1016.
- [13] S. Niu and A. Todri-Sanial, Analyzing crosstalk error in the nisq era, in *2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)* (IEEE Computer Society, Los Alamitos, CA, USA, 2021) pp. 428–430.
- [14] Y. Li and S. C. Benjamin, Efficient variational quantum simulator incorporating active error minimization, *Physical Review X* **7**, 021050 (2017).
- [15] P. Das, S. Tannu, S. Dangwal, and M. Qureshi, Adapt: Mitigating idling errors in qubits via adaptive dynamical decoupling, in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture* (2021) pp. 950–962.
- [16] P. Jurcevic, A. Javadi-Abhari, L. S. Bishop, I. Lauer, D. F. Bogorin, M. Brink, L. Capelluto, O. Günlük, T. Itoko, N. Kanazawa, *et al.*, Demonstration of quantum volume 64 on a superconducting quantum computing system, *Quantum Science and Technology* **6**, 025020 (2021).
- [17] Y. Sung, F. Beaudoin, L. M. Norris, F. Yan, D. K. Kim, J. Y. Qiu, U. von Lüpke, J. L. Yoder, T. P. Orlando, S. Gustavsson, *et al.*, Non-gaussian noise spectroscopy with a superconducting qubit sensor, *Nature communications* **10**, 1 (2019).
- [18] T. Van der Sar, Z. Wang, M. Blok, H. Bernien, T. Tamini, D. Toyli, D. Lidar, D. Awschalom, R. Hanson, and V. Dobrovitski, Decoherence-protected quantum gates for a hybrid solid-state spin register, *Nature* **484**, 82 (2012).
- [19] I. Solomon, Multiple echoes in solids, *Physical Review* **110**, 61 (1958).
- [20] S. Meiboom and D. Gill, Modified spin-echo method for measuring nuclear relaxation times, *Review of scientific instruments* **29**, 688 (1958).
- [21] L. Viola, E. Knill, and S. Lloyd, Dynamical decoupling of open quantum systems, *Physical Review Letters* **82**, 2417 (1999).
- [22] A. M. Souza, G. A. Alvarez, and D. Suter, Robust dynamical decoupling for quantum computing and quantum memory, *Physical review letters* **106**, 240501 (2011).
- [23] B. Pokharel, N. Anand, B. Fortman, and D. A. Lidar, Demonstration of fidelity improvement using dynamical decoupling with superconducting qubits, *Physical review letters* **121**, 220502 (2018).
- [24] A. M. Souza, Process tomography of robust dynamical decoupling with superconducting qubits, *Quantum Information Processing* **20**, 1 (2021).
- [25] D. C. McKay, T. Alexander, L. Bello, M. J. Biercuk, L. Bishop, J. Chen, J. M. Chow, A. D. Córcoles, D. Egger, S. Filipp, *et al.*, Qiskit backend specifications for openqasm and openpulse experiments, arXiv preprint arXiv:1809.03452 (2018).
- [26] T. Alexander, N. Kanazawa, D. J. Egger, L. Capelluto, C. J. Wood, A. Javadi-Abhari, and D. C. McKay, Qiskit pulse: programming quantum computers through the cloud with pulses, *Quantum Science and Technology* **5**, 044006 (2020).
- [27] Y. Shi, N. Leung, P. Gokhale, Z. Rossi, D. I. Schuster, H. Hoffmann, and F. T. Chong, 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* (2019) pp. 1031–1044.
- [28] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms, *Journal of magnetic resonance* **172**, 296 (2005).
- [29] D. J. Egger and F. K. Wilhelm, Adaptive hybrid optimal quantum control for imprecisely characterized systems, *Physical review letters* **112**, 240503 (2014).
- [30] J. P. Stenger, N. T. Bronn, D. J. Egger, and D. Pekker, Simulating the dynamics of braiding of majorana zero modes using an ibm quantum computer, *Physical Review Research* **3**, 033171 (2021).
- [31] N. Earnest, C. Tornow, and D. J. Egger, Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware, arXiv preprint arXiv:2105.01063 (2021).
- [32] H. Y. Carr and E. M. Purcell, Effects of diffusion on free precession in nuclear magnetic resonance experiments, *Physical review* **94**, 630 (1954).
- [33] T. Gullion, D. B. Baker, and M. S. Conradi, New, compensated carr-purcell sequences, *Journal of Magnetic Resonance* (1969) **89**, 479 (1990).
- [34] G. S. Uhrig, Keeping a quantum bit alive by optimized  $\pi$ -pulse sequences, *Physical Review Letters* **98**, 100504 (2007).
- [35] A. Ajoy, G. A. Álvarez, and D. Suter, Optimal pulse spacing for dynamical decoupling in the presence of a purely dephasing spin bath, *Physical Review A* **83**, 032303 (2011).
- [36] D. M. Grant and R. K. Harris, *Encyclopedia of nuclear magnetic resonance*, volume 9, advances in nmr, Spectroscopy (1996).
- [37] G. S. Ravi, K. N. Smith, P. Gokhale, A. Mari, N. Earnest, A. Javadi-Abhari, and F. T. Chong, Vaqem: A variational approach to quantum error mitigation, arXiv preprint arXiv:2112.05821 (2021).
- [38] J. M. Chow, J. M. Gambetta, E. Magesan, D. W. Abraham, A. W. Cross, B. R. Johnson, N. A. Masluk, C. A. Ryan, J. A. Smolin, S. J. Srinivasan, *et al.*, Implementing a strand of a scalable fault-tolerant quantum computing fabric, *Nature communications* **5**, 1 (2014).
- [39] S. Sheldon, E. Magesan, J. M. Chow, and J. M. Gambetta, Procedure for systematically tuning up crosstalk in the cross-resonance gate, *Physical Review A* **93**, 060302 (2016).
- [40] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv preprint arXiv:1411.4028 (2014).
- [41] A. Papageorgiou and C. Zhang, On the efficiency of quantum algorithms for hamiltonian simulation, *Quantum Information Processing* **11**, 541 (2012).
- [42] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM review* **41**, 303 (1999).
- [43] A. Y. Kitaev, Quantum measurements and the abelian stabilizer problem, arXiv preprint quant-ph/9511026 (1995).

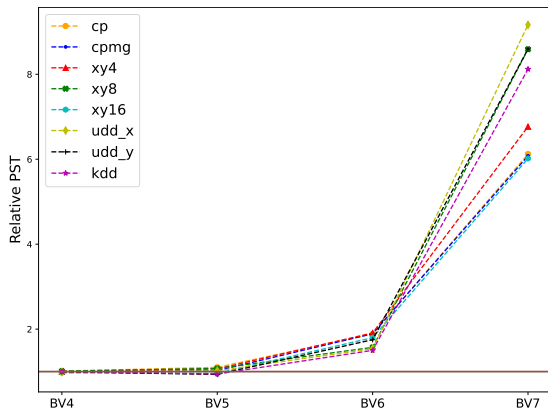
- [44] Quantum AI team and collaborators, Recirq (2020).
- [45] S. S. Tannu and M. K. Qureshi, Not all qubits are created equal: a case for variability-aware policies for nisq-era quantum computers, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (2019) pp. 987–999.
- [46] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo, *et al.*, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, *Nature Physics* **17**, 332 (2021).
- [47] V. Tripathi, H. Chen, M. Khezri, K.-W. Yip, E. Levenson-Falk, and D. A. Lidar, Suppression of crosstalk in superconducting qubits using dynamical decoupling, arXiv preprint arXiv:2108.04530 (2021).
- [48] S. Niu and A. Todri-Sanial, Enabling multi-programming mechanism for quantum computing in the nisq era, arXiv preprint arXiv:2102.05321 (2021).
- [49] S. Niu and A. Todri-Sanial, How parallel circuit execution can be useful for nisq computing?, arXiv preprint arXiv:2112.00387 (2021).
- [50] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, *Nature* **549**, 242 (2017).
- [51] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, *Reviews of Modern Physics* **86**, 153 (2014).



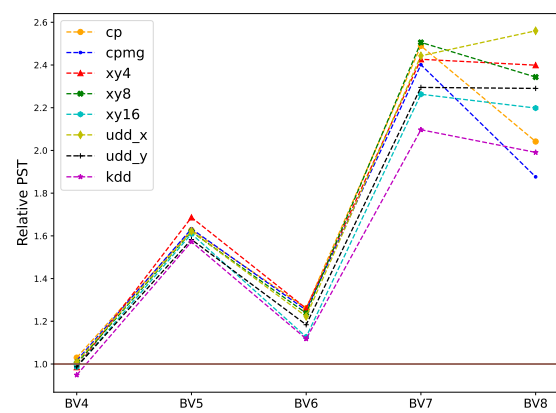
(a) ibmq\_jakarta



(b) ibmq\_guadalupe



(c) ibmq\_toronto



(d) ibmq\_montreal

FIG. 4: Relative PST results for BV circuits on different IBM quantum devices. Higher is better.

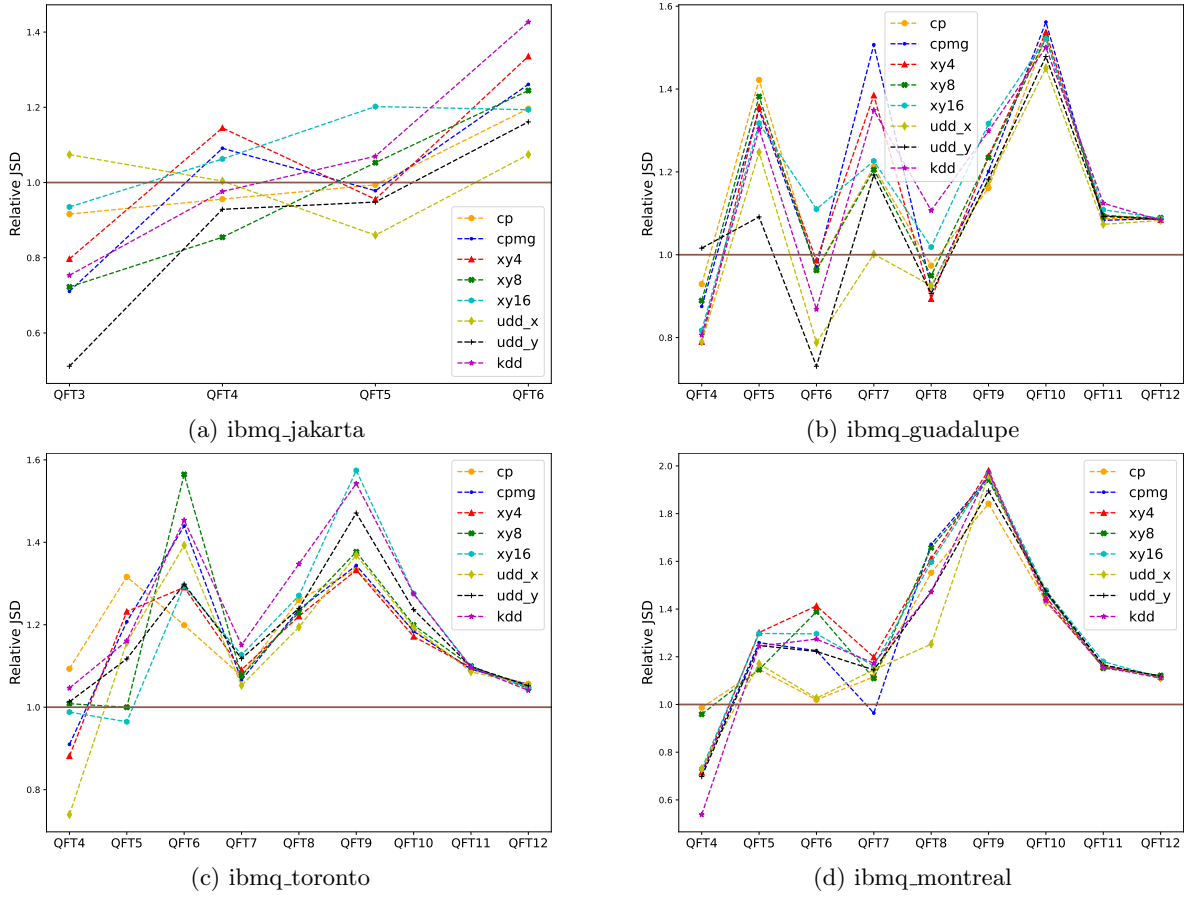
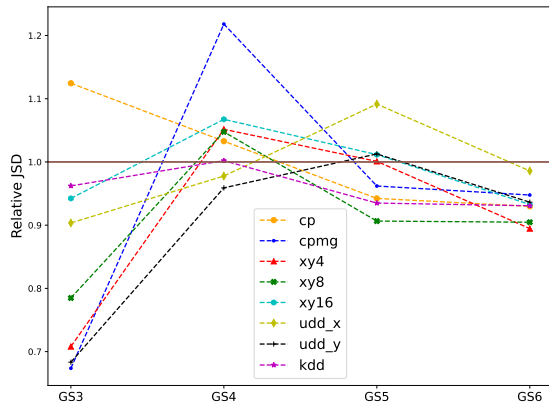
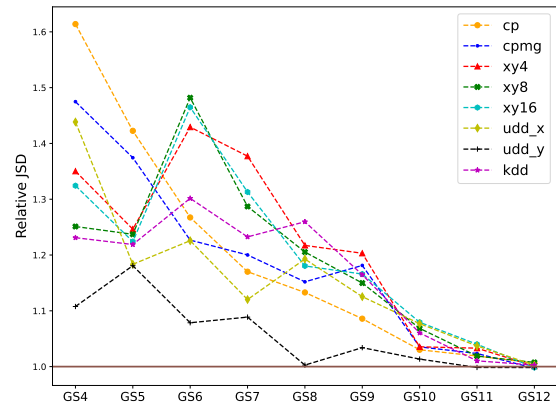


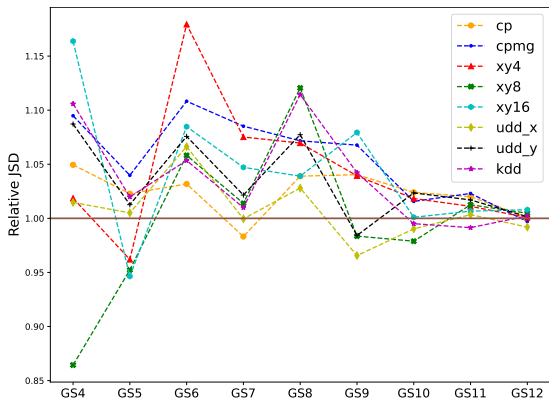
FIG. 5: Relative JSD results for QFT circuits on different IBM quantum devices. Higher is better.



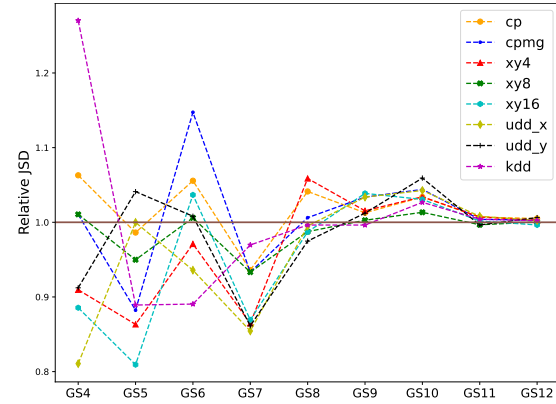
(a) ibmq\_jakarta



(b) ibmq\_guadalupe

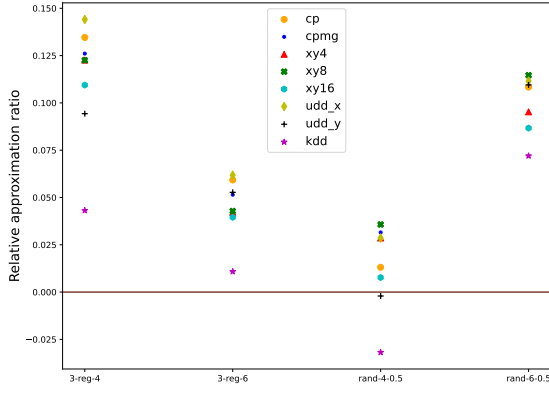


(c) ibmq\_toronto

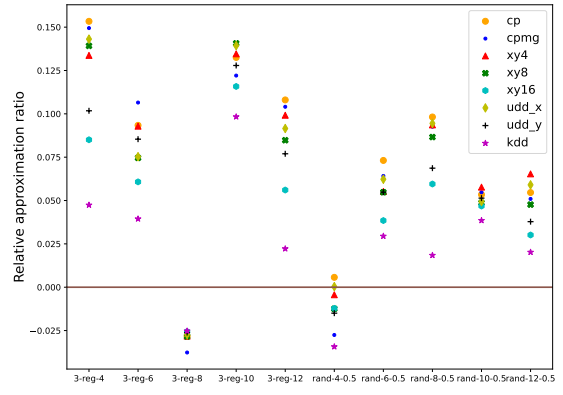


(d) ibmq\_montreal

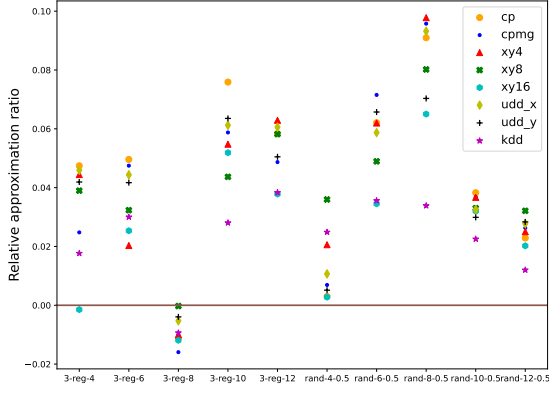
FIG. 6: Relative JSD results for Graph State circuits on different IBM quantum devices. Higher is better.



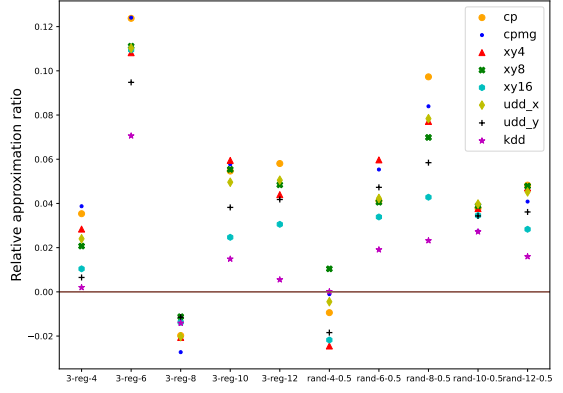
(a) `ibmq_jakarta`



(b) `ibmq_guadalupe`



(c) `ibmq_toronto`



(d) `ibmq_montreal`

FIG. 7: Relative approximation ratio of applying DD sequences for QAOA circuit on different IBM quantum devices. Higher is better.

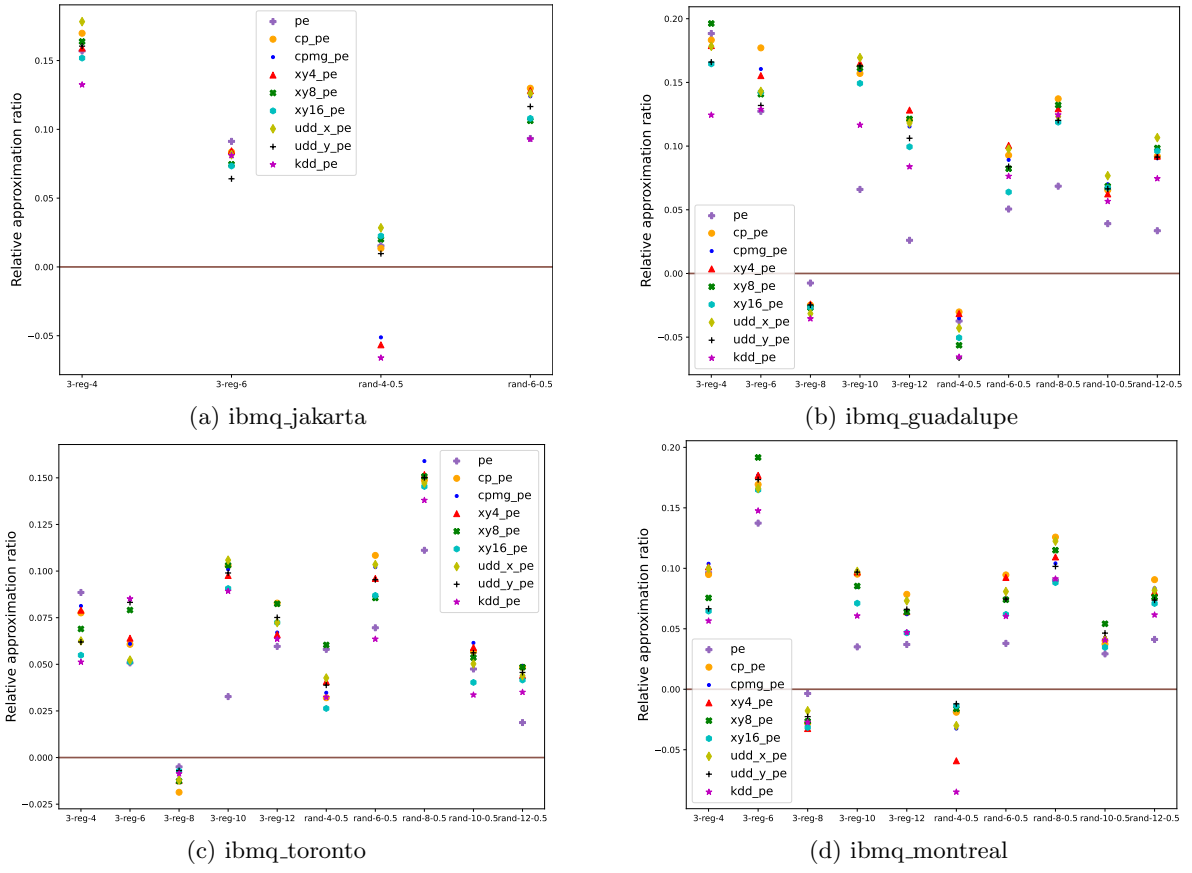


FIG. 8: Relative approximation ratio of applying DD sequences + pulse-efficient technique for QAOA circuit on different IBM quantum devices. Higher is better.