Journal of Physics Astronomy

Research Article | Vol 14 Iss 1

Comparing the Performance of Three Error Mitigation Techniques on Grover's Quantum Search Algorithm

Mihail Mircheski*

Department of Physics, Duke University, Durham, Macedonia

***Corresponding author:** Mihail Mircheski, Department of Physics, Duke University, Durham, Macedonia; E-mail: m.mirceski06@gmail.com

Received: January 21, 2025, Manuscript No. TSPA-25-159467; **Editor assigned:** January 24, 2025, PreQC No. TSPA-25-159467 (PQ); **Reviewed:** February 10, 2025, QC No. TSPA-25-159467; **Revised:** March 21, 2026, Manuscript No. TSPA-25-159467 (R); **Published:** March 28, 2026; DOI: 10.37532/2320-6756.2026.14(1).454

Abstract

Quantum algorithms have the potential of outperforming classical algorithms for solving certain tasks. Grover's Quantum Search Algorithm (QSA) is one such algorithm that provides a quadratic speedup over the classical algorithm, for the problem of searching an unstructured list to find a marked item. However, its accuracy is severely affected by noise, which is apparent especially when running it on a real quantum processor. This paper seeks to expand previous efforts to implement error mitigation techniques to the algorithm, by comparing the performance of three such techniques on the 2, 3 and 4-qubit Grover's QSA: Measurement error mitigation with calibration matrices, matrix-free measurement error mitigation (Mthree) and dynamical decoupling. Along the way, Quantum State Tomography (QST) is introduced and used to compare the fidelity of the prepared state for different numbers of qubits. The results indicate that dynamical decoupling is the most effective at mitigating errors in the algorithm, while measurement error mitigation with calibration matrices becomes less effective with increasing number of qubits. The results of implementing Mthree are inconclusive and merit further investigation. Regarding the state fidelity, it decreased with increasing number of qubits.

Keywords: Core collapse supernova; Neutron degeneracy pressure; Gravitational binding energy; Quantum chromo-dynamic binding energy

Introduction

Quantum computing is one of the most promising areas of research in the 21st century. The fundamental difference between quantum and classical computing is the nature of the entities used to encode information [1]. While a classical bit can be in a state of either 0 or 1, a quantum bit (or qubit) can be in a superposition state between 0 and 1 (*i.e.*, both states simultaneously). This phenomenon has roots in quantum mechanics, with the concept of wave-particle duality: electrons, for example, can be described as both a wave and a particle [2].

A bit can be either 0 or 1 (spin up or spin down in the image above). A qubit, however, can be in any number of superposition states between 0 and 1, with different amplitudes (represented by the quantities alpha and beta above).

In Figure 1, it can be seen that an arbitrary state $|\Psi\rangle$, has different amplitudes for the states 0 and 1 (alpha and beta for $|0\rangle$ and $|1\rangle$, respectively). These amplitudes are complex numbers that, when multiplied by their complex conjugate, give the probability of finding the qubit in the state 0 or 1 [3,4].

Citation: Tang J. Derivation of The Fine-Structure Constant Beyond 12 Decimal Digits Based On Quantization of Hypercomplex Gauges. J Phys Astron. 2026;14(1):454.
© 2026 Trade Science Inc.

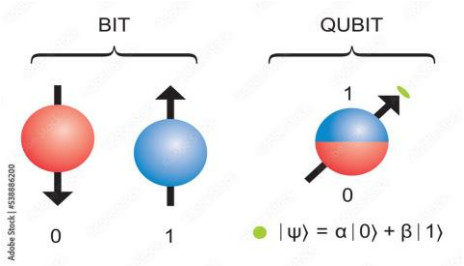


FIG. 1. **Superposition.**

A more complex representation of the states that a qubit can be in is illustrated by a Bloch sphere (Figure 2).

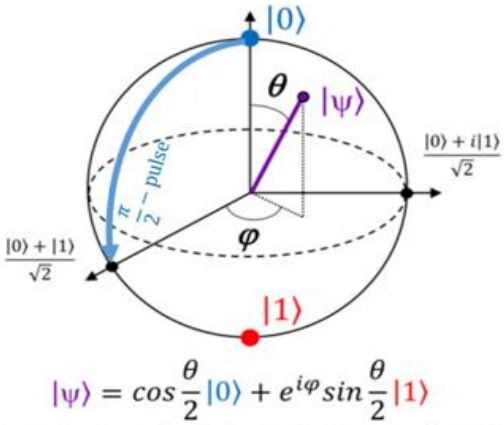


FIG. 2. **Bloch sphere.**

Any point on the Bloch sphere corresponds to a state with a linear combination of the basis states with complex coefficients. A$\pi/2$ pulse transforms the $|0\rangle$ state to the $|+\rangle$ state.

It is an imagined 3D representation of the vector space of a single qubit, where rotations around the different axes (X, Y or Z), represent different states. At the poles of the Bloch sphere, there are the states $|0\rangle$ and $|1\rangle$, which are the eigenstates of the Pauli Z operation, otherwise known as a phase flip (which leaves the $|0\rangle$ state the same and adds a phase of -1 to the $|1\rangle$ state). At the poles of the X axis there are $|+\rangle$ and $|-\rangle$ states, which are abbreviated notations for an equal superposition between the 0 and 1 states ($|0\rangle + |1\rangle/\sqrt{2}$ and $|0\rangle - |1/\sqrt{2}\rangle$, respectively). In both cases, the amplitudes for the $|0\rangle$ and $|1\rangle$ states are equal to $1/\sqrt{2}$, which equates to a probability of 1/2 when squared - hence, equal superposition. The "plus" and "minus" states are eigenstates of the Pauli X operation, otherwise known as a bit flip (transforms a 0 into a 1 and vice-versa). Expectedly, the eigenstates of the Y operation, which is both a phase flip and a bit flip, are at the poles of the Y axis. The Pauli operations and their matrices are some of the basic operations on qubits, along with the Hadamard (H) operation, which transforms a $|0\rangle$ state into the $|+\rangle$ state and a $|1\rangle$ state into the $|-\rangle$ state - in other words, placing the qubit in an equal superposition between the $|0\rangle$ and $|1\rangle$ states, or performing a basis change from the Z basis to the X basis [6]. There are also operations acting on multiple qubits, like the Controlled X operation (CX), which performs a bit flip on a target qubit only if the control qubit is 1, otherwise it remains in the same state [7].

These primitive operations on qubits are called quantum gates - these transform the states of the qubits and manipulate the superpositions to create various quantum interference effects, like entanglement (in other words, rotating the vectors around the Bloch sphere). The end result is the construction of quantum circuits and quantum algorithms that can perform certain tasks much faster than a classical computer (sometimes exponentially so). While a classical computer might need a huge number of iterations and queries in order to compute a certain task, like finding the global maximum or minimum of a function, a quantum computer can utilize superpositions to calculate the values for many different inputs simultaneously and therefore compute the property in much less time. For the case of an exponential speedup, the calculation time can potentially be reduced from years to minutes. This can be utilized for many applications, like synthesizing new drugs, machine learning and optimization problems (like vehicle routing, for example), many of which can be described as unstructured searches for some global property. This implies tremendous

commercial value for the technology [8].

Materials and Methods

Noise and types of errors

However, quantum computers still have some serious problems. One of the foremost challenges in the development of the field is the phenomenon of "noise". In real quantum processors, the physical qubits can interact with each other and with their environment to produce results that differ from the ideal outcome. Roughly speaking, the higher the number of qubits involved in a circuit or algorithm, and the higher the circuit depth, the higher the error rate observed [9]. A graphical representation of this can be seen in Figure 3. Circuit depth is the measure of the time it takes to execute the quantum gates in an algorithm. Thus, if there are N sequential gates, the circuit depth will scale linearly with the number N.

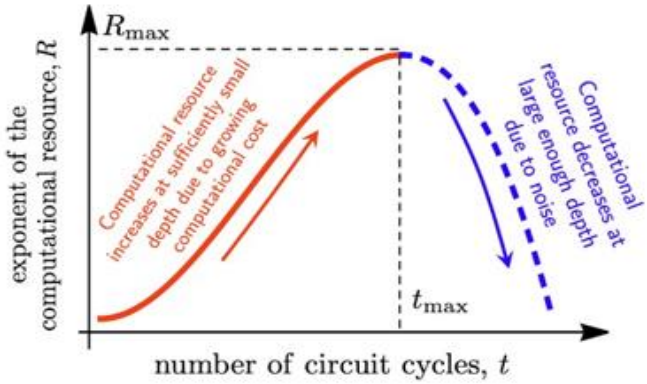


FIG. 3. **A graph of the number of circuit cycles against the exponent of the computational resource. At lower cycles and thus lower circuit depth, the computational resource that can be extracted from a quantum computer to gain advantage over a classical one is increasing. However, it peaks somewhere at t_{max}, after which it decreases with increasing circuit depth due to noise.**

Two-qubit and multi-qubit gates are often one of the greatest contributors to noise, since they involve interactions between physical qubits by definition. This poses a significant barrier to utility-scale quantum computing, whereby scaling a quantum computer to a higher number of qubits does not necessarily translate to quantum advantage over classical computers, because of increasing levels of noise.

This is the reason why quantum computers are currently labeled to be in the NISQ era (Noisy Intermediate-Scale Quantum computing), many years (or even decades) away from the so-called fault-tolerant quantum computing, where noise will have little to no effect on results and will be corrected for.

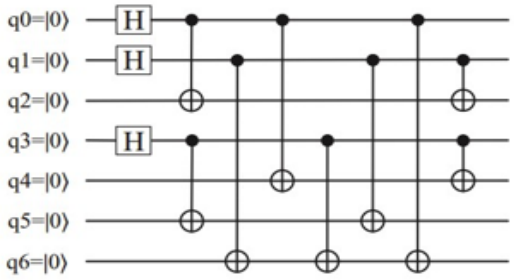
There are several different types of noise. Decoherence refers to interactions with the environment that alter the superposition state of the qubit in some way. Usually, decoherent error is modeled by an entanglement with the "state" of the environment that disrupts the original state. In addition, every gate and operation on the qubits has its own associated error.

Out of all of these, measurement errors are one of the most dominant, which is why it is addressed heavily in this paper. Finally, there are also thermal relaxation errors. Every quantum processor has its own T1 and T2 time. The T1 time is how long it takes for a qubit to relax to its ground state, thus releasing energy. In terms of quantum states, it is roughly the period needed for a state to decay to the $|0\rangle$ state. The T2 time is a measure of how long a phase can be controlled. The lower these values are, the more noise there is in the system and the more unstable it is.

Proposed solutions for the problem of noise

There have been several methods proposed for solving this problem. One of them is error correction, which attempts to do away with noise altogether. There are many error correction codes. The simplest version is the repetition code, which encodes a "logical"

qubit (noiseless, idealized qubit), into multiple physical qubits. At the end, a decoder reverts the qubit back to its original state by utilizing the exact inverse operations as those in the encoder (Figure 4).



(a)

#	Instructions
	QUBIT q0, 0
	QUBIT q1, 0
	QUBIT q2, 0
	QUBIT q3, 0
	QUBIT q4, 0
	QUBIT q5, 0
	QUBIT q6, 0
1	H q0
2	H q1
3	H q3
4	CNOT q2, q0
5	CNOT q5, q3
6	CNOT q6, q1
7	CNOT q4, q0
8	CNOT q6, q3
9	CNOT q5, q1
10	CNOT q6, q0
11	CNOT q2, q1
12	CNOT q4, q3

(b)

FIG. 4. **The encoder for an error correction code called the Steane code, using six physical qubits. Different gates are listed under instructions.**

A more sophisticated code, and the likely candidate for the "ultimate" QEC (Quantum Error Correction) code is the surface code, which uses graph theory to map logical qubits onto different physical qubits. There is also the Shor code, the Steane code, and many others. A feature of all is the fact that they introduce a lot of overhead, or additional physical qubits, just to accomplish error correction and its other features like syndrome measurements. For example, Shor's code uses 9 physical qubits for one logical qubit. This is ultimately counterproductive, as it can introduce more sources of error into the system, and is also not very practical for the current available devices, the best of which have around 1000 qubits at most. So, error correction is certainly one of the ways to achieve fault-tolerant quantum computing, but is not very practical at the present moment.

Another interesting proposal is QEM (Quantum Error Mitigation). As the name suggests, these methods only serve to reduce the errors, not correct them completely. In addition, they do not require as much overhead as quantum error correction. There are many techniques that fall under the umbrella term of QEM. One of them is readout error mitigation, or measurement error mitigation, which runs the circuit for the basis states and collects the noisy results, constructing a calibration matrix (otherwise known as an assignment matrix) with the error probabilities as columns. The inverse of this matrix is then computed and is applied to noisy results, to extract the ideal outcomes. Measurement error mitigation can also be done without matrices. The package Mthree, used in this paper, compatible with Qiskit (IBM's Software Development Kit for quantum computers), attempts to accomplish this.

Other methods exist to mitigate gate errors, like Pauli twirling (or randomized compilation), which sandwiches a chosen gate (usually a two-qubit or multiple-qubit gate), between two different randomly chosen Pauli gates that, when applied, are identical to the identity operation (*i.e.*, the ideal result remains unchanged), but they have the effect of transforming coherent noise into a Pauli noise channel that is easier to deal with. ZNE (Zero-Noise Extrapolation) is used for more sophisticated algorithms, where circuits are run under different progressively increasing error rates and the results are then extrapolated to zero noise. IBM's Qiskit has inbuilt functions that can incorporate these mitigation techniques while post-processing results, but they are only applicable to tasks run on real IBM hardware, not on simulations.

Explanation of Grover's algorithm

Grover's quantum search algorithm (hereafter referred to as Grover's QSA) is an algorithm for searching through an unstructured list of N members, to find a member of a list that fulfills a certain property. It was first designed by Lov Grover, a computer scientist, in 1996. Classically, the time to solve this problem would be proportional to N, since all members of the list would have to be

checked by the algorithm in the worst case scenario. On average, only half of the N items will need to be checked to obtain a 50% probability of finding the special member. This is equivalent to saying that the problem has a computational complexity of O (N), using the so-called "big O" notation. However, what Grover's algorithm manages to accomplish is reduce this complexity to O ($\sqrt{N}$), offering a quadratic speedup. For example, a task that would take a classical computer 180 minutes, or 3 hours to compute, can be done in 13 minutes with Grover's QSA. The following is an explanation of how Grover's algorithm succeeds in doing this (Figure 5).

Let us assume that the list to be searched is the vector space of all states that n qubits can be in (the total number of this space is 2^n).

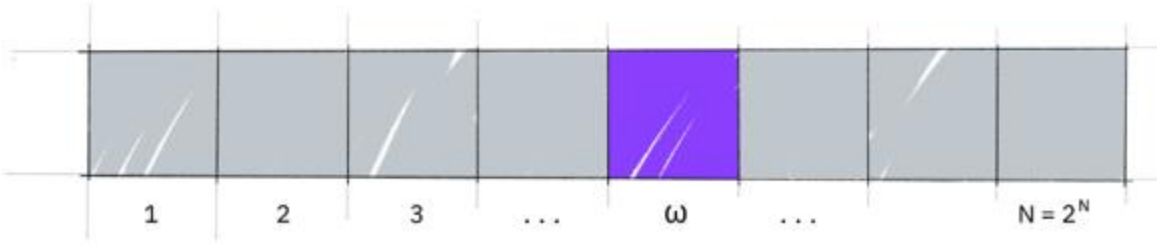


FIG. 5. **Search space with the marked state "W".**

Let us denote the marked state(s) that need(s) to be found by the algorithm with the letter "W". To isolate W with a high probability from the rest of the search space, Grover's algorithm uses four steps: initialization, the oracle, the diffuser and measurement.

The first step, initialization, uses a Hadamard gate on all n qubits to place them in an equal superposition of all possible states, thereby creating the search space. There will be 2^n states in total and the probability of randomly guessing the marked state will therefore be $1/2^n$.

The probability decreases exponentially with the number of qubits.

The second step is the introduction of the oracle. An oracle is a "black box", or a function whose construction is not accessible - only its outputs can be read off when it is queried with an input. The way that the outputs of the oracle in Grover's algorithm are defined is that they are equal to 0 for all states not equal to "W", and equal to 1 for "W". This is implemented by adding a phase of $(-1)^{f(x)}$ to all the states, which would only mark the "W" state with a phase of −1 as it is the only one with $f(x) = 1$.

The third step is the diffuser. This part of Grover's algorithm performs the subroutine of "amplitude amplification". It inverts the amplitude of the marked state "W" about the mean and increases it relative to the amplitudes of the other states, thereby lowering the mean amplitude, which is why this process is also called inversion about the mean.

The final step is the measurement. After multiple iterations of the three steps above, the amplitude of the marked state is increased until the chance of measuring the marked state is very high. After that, a measurement will collapse the vector space to the marked state, meaning that the algorithm has successfully searched the list.

There is a concise geometrical representation of Grover's algorithm. If we denote the state of the n qubits after the initialization step as |s⟩, and denote the set of states excluding the marked state as |s'⟩, then |s'⟩ and |w⟩ would be two perpendicular vectors that would form an orthonormal basis. This allows for the representation of the operations performed by the oracle and by the diffuser simply as reflections of the |s> vector across the 2D plane formed by the two vectors (Figure 6).

$$|s\rangle = sin(\theta)|w\rangle + cos(\theta)|s'\rangle$$

The angle θ seen in Figure 6 is thus equal to:

$$\theta = arcsin(<w|s>) = arcsin(\frac{1}{\sqrt{N}})$$

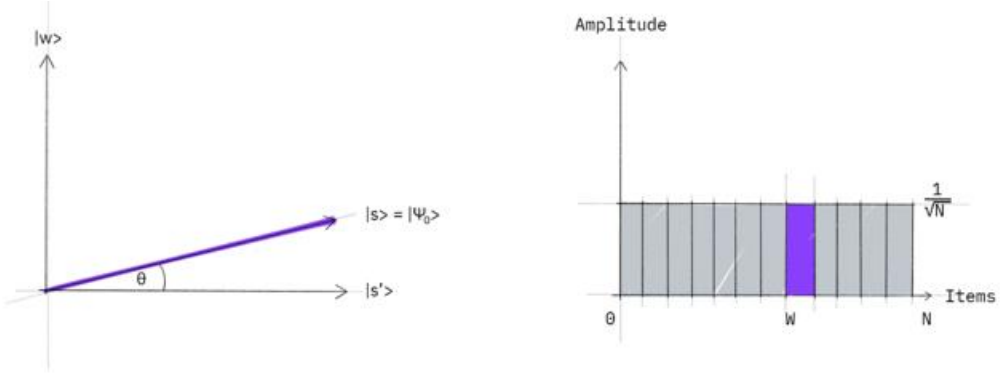


FIG. 6. **Graphical representation of Grover's algorithm: The initialization step.**

The operation executed by the oracle would constitute a reflection across the $|s'\rangle$ axis, since only the component of $|s\rangle$ along the $|w\rangle$ axis is made negative with a phase of negative 1 (Figure 7).

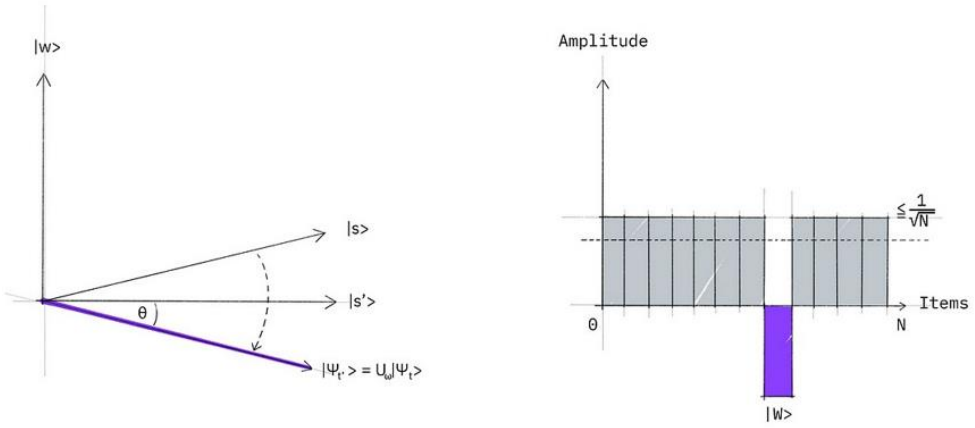


FIG. 7. **Graphical representation of Grover's algorithm: The oracle step.**

Afterwards, the diffuser reflects the new vector across the original $|s\rangle$ vector, ending up with a final vector that has been moved toward the $|w\rangle$ axis by an angle of 2θ, where θ is the angle of reflection. Therefore, it could be seen that after a set number t iteration, the vector would get closer to the $|w\rangle$ axis by $2\theta t$. When $\theta(2t + 1)$ becomes equal to $\pi/2$, *i.e.*, when the vector is oriented completely along the $|w\rangle$ axis, the final measurement step would be applied to extract the marked state $|w\rangle$ with high probability (Figure 8).

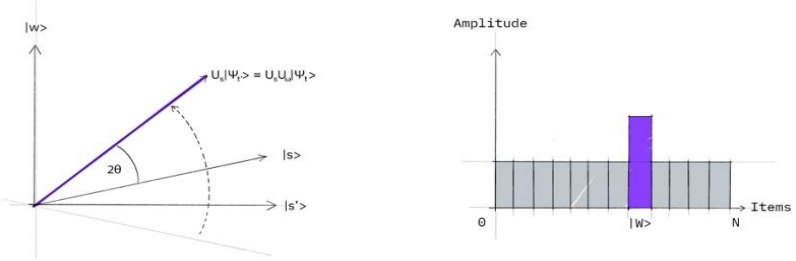


FIG. 8. **Graphical representation of Grover's algorithm: The diffuser step**.

Taking the previous into account, the optimal number of iterations t can be calculated as the following:

$$t = \left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{m}} \right\rfloor,$$

where N is the total number of states and m is the number of marked states (there will not be more than one marked state used in this paper). The brackets denote the floor function (rounding down to the nearest whole number). Since t is proportional to $\sqrt{N}$, this algorithm indeed has a complexity of $O(\sqrt{N})$.

Based on the above equation, the optimal number of iterations for 2, 3 and 4 qubits are computed to be 1, 2 and 3, respectively.

Implementation of Grover's algorithm in Qiskit

In this paper, Grover's algorithm is implemented up to 4 qubits, starting from 2. The circuits are constructed and executed in IBM's Qiskit Software Development Kit (SDK). Below are images showing the construction of the circuits for 2, 3, 4 and 5 qubits for the marked states |11⟩, |111⟩, |1111⟩, respectively.

The circuits shown in Figure 9 are missing the initialization step, which is done once at the beginning, followed by several iterations of the oracle and diffuser. Since the circuits follow the same pattern, the following is a breakdown of the simplest case involving 2 qubits.

After the initialization step, which simply features 1 Hadamard gate per qubit, there is a Hadamard on the most significant qubit (according to Qiskit's ordering convention), followed by a CX (controlled-not) gate, followed by another Hadamard. The three gates together form a Controlled-Z (CZ) gate, which performs a Z operation (phase flip), only if the control qubit(s) are in the |1⟩ state. Therefore, no matter how many qubits are involved, the phase will always be added to the state consisting of n "1" s, where n is the number of qubits - in other words, the required marked state will always be targeted by the oracle.

Moving on to the diffuser, the Hadamard gates convert the |s⟩ state back into a 2-qubit tensor product of |0⟩ (|00⟩), which was the starting state before the initialization step. Then, the Z-gate (decomposed into HXH in Figure 9) and the Controlled-Z gate perform a reflection across the |s> state, by adding a phase to all states orthogonal to |00⟩. Finally, the Hadamard gates again convert the |00⟩ state back into the |s⟩ state. All combined, the cumulative effect of the gates is a reflection across the |s⟩ state, or $2|s><s| - I$, where I is the identity matrix.

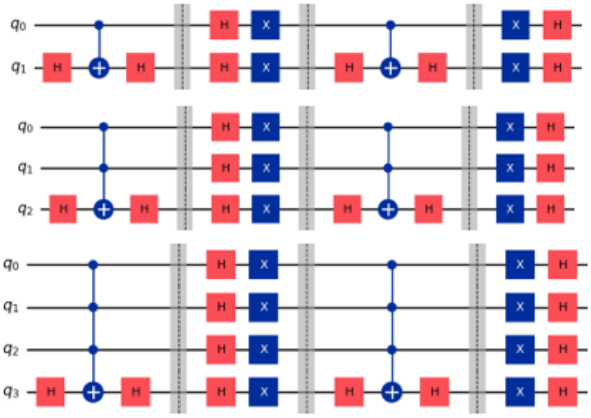


FIG. 9. **Circuit implementation of Grover's algorithm for 2, 3 and 4 qubits, top to bottom.**

For marked states other than an n-qubit tensor product of |1⟩, a different circuit construction should be used. First, an oracle is created by a combination of a multi-controlled Z gate as before, combined with bit flips (X gates), wherever the marked bitstring has "0" s. This way, all cases are covered. Subsequently, the Qiskit GroverOperator class is used to automatically create a diffuser, based on the oracle as an input. Finally, the circuit is raised to the power of t, where t is the optimal number of iterations as discussed above.

Error mitigation techniques

In this paper, three error mitigation techniques will be used to reduce the effect of errors on the results of the implementation of Grover's algorithm - namely, measurement error mitigation as implemented through the Qiskit experiment module (with the use of calibration matrices), matrix-free measurement error mitigation through Mthree (without the use of calibration matrices), and dynamical decoupling. An explanation of each of these methods separately is provided in this section.

Measurement error mitigation with the use of calibration matrices: This is implemented through the Qiskit experiments package. In general, it works by first executing (on the real, noisy backend) basic circuits for storing each of the possible states in a sample space that the specified number of qubits could be in. These circuits consist of just a measurement operation, with additional X operations as needed for the particular state. For example, when the number of qubits is 2, 4 circuits are created, for storing each of the 4 possible states (|00⟩, |01⟩, |10⟩ and |11⟩). X operations are performed for the latter three cases to switch the qubits from the |0⟩ to the |1⟩ state, as needed. After this step is completed, the noisy results are stored in a calibration matrix, also known as an assignment matrix. The entries of this matrix are the probabilities that a certain state will be stored as another state. Predictably, for low levels of noise the probability of storing the state as itself is the highest - however, as the noise increases, the probability of the state being "misread" as other states (due to readout errors), increases. The sum of the entries in each of the columns of the matrix is 1, so the matrix is normalized. An example of an assignment matrix is shown below.

There are different types of assignment matrices. The local one is used under the assumption that each error affects each qubit separately (*i.e.*, the errors are independent). For this type of error, it suffices to use n 2×2 matrices, where n is the number of qubits. This requires less computational resources, although it is less useful for modeling real-life situations, where the errors of each qubit are intertwined. For this case, a correlated readout error assignment matrix is used, which is a $2^n \times 2^n$ matrix, where n is the number of qubits. Obviously, this will require more computational resources and execution time. In this paper, however, correlated assignment matrices will be used, since the algorithm is run on a real backend, not simplified noise models (Figure 10).

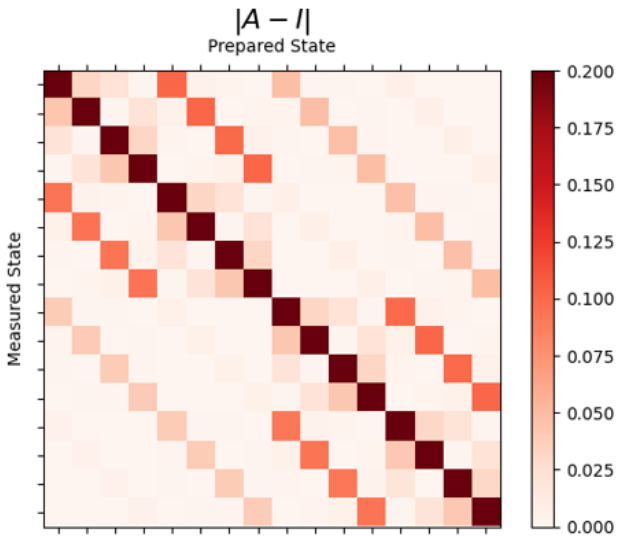


FIG. 10. **Local error assignment matrix for 4 qubits (16 states in total). Darker red indicates higher probability. Notice that the probability entries in the diagonals are highest (*i.e.*, the probability of measuring a prepared state as itself), as expected.**

Regardless of the type, the role of both matrices is the same. Namely, if this matrix is applied on ideal results, it will convert them to noisy results. So, in order to convert noisy results to ideal results, the inverse of the matrix is computed and used to mitigate the errors in whatever algorithm needs to be studied (Grover's algorithm in this case). The simple operation is shown in the equation below.

$p(ideal) = A^{-1}p(noisy)$,

where A^{-1} is the inverse of the assignment matrix, $p(noisy)$ are the noisy probabilities and $p(ideal)$ are the ideal probabilities.

Measurement error mitigation without the use of calibration matrices: This is implemented through the Mthree package. It works by reducing the sample space from n to k, by recognizing that certain outcomes are not obtained among the results. This reduction is significant in the amount of time that it saves to search the sample space, even with just a small reduction in the states to be searched. It works well when the measurement error probability is relatively low. In this paper, it will be tested how well it works for a real backend.

Dynamical decoupling: This is implemented through Qiskit's inbuilt functionality for post-processing and error mitigation of results, applicable only when the desired workloads are run on real IBM quantum processors. This technique is used to mitigate the errors arising where certain qubits are experiencing idle periods, when they are not used as much as the other qubits. This can lead to more frequent unwanted interactions with other qubits or the environment, which can significantly skew the results. The higher the circuit depth (or the number of operations), the greater these idle periods. It works by inserting pulse sequences to the physical qubits in idle periods. These sequences generally compute to be equal to the identity operation, which means that one pulse used successively after another one is the same as applying an identity operation to the qubit, leaving it unchanged. In this paper, an "XX" sequence was used. The real effect of these sequences is reducing the interactions with the environment on a physical level (Figure 11).

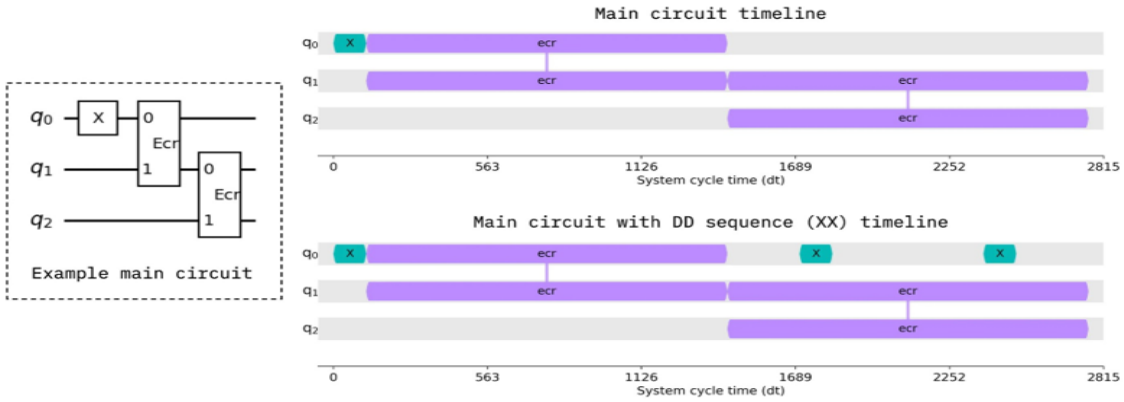


FIG. 11. **Example of an "XX" pulse sequence being applied to a circuit with one X gate and two ECR operations. The zeroth qubit is idle after the second gate.**

Note that applying error correction to the algorithm was also considered, and deemed infeasible in that it requires a great overhead. For example, with Shor's code, with a physical-to-logical qubit ratio of 9:1, for the least complex version of the 2 logical qubits, 18 physical qubits would be required. This means that the matrix showing the whole sample space of possible states would be equal to $(2^{18})^2$=68719476736 elements. If each of these states is stored as a 9-bit complex float, more than 600 gigabytes of memory would be needed to run the algorithm.

Quantum state tomography

Quantum State Tomography (hereafter abbreviated as QST) is another feature that can be applied through the Qiskit experiments module. In this paper, it is used to compare the state fidelities across implementations of Grover's algorithm for different numbers of qubits. State fidelity is a measure of how close the observed, "noisy" state is to the intended outcome. It is essentially a measure of the overlap between two states, or how well a certain state was prepared. It involves an extraction of the density matrix representation of the state just before the measurement process collapses it. Solving this problem is highly nontrivial and for the purposes of this paper, is done automatically through the use of the Qiskit experiments package. The following is a simplified equation for state fidelity:

$$F = |\langle \psi | \boldsymbol{\phi} \rangle|^2,$$

where ψ is the target ("noisy") state and ϕ is the ideal state.

Results and Discussion

A real IBM processor was used for this purpose - IBM Kyoto. The algorithm for 5 qubits was not implemented due to the significant increase in execution time and complexity, and the limits imposed by IBM - only 10 minutes of free time per month are allowed for workflows implemented on real backends.

The following are the results of a noiseless simulation, run on a perfect Qiskit high-performance Aer Simulator. The target state is |1111>. It can be seen that, even for a large number of qubits (4), the results are nearly perfect (Figure 12).

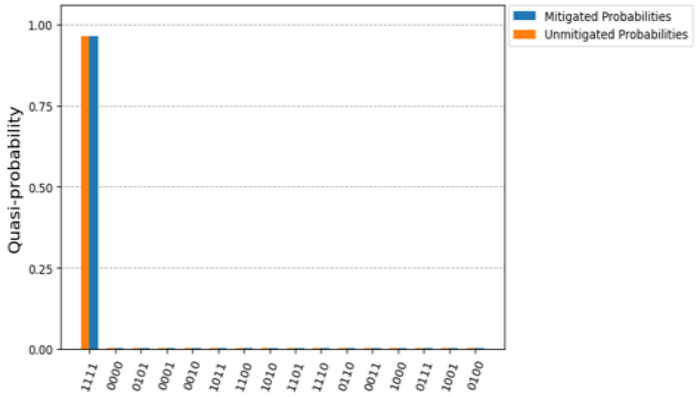


FIG. 12. **Perfect simulation of Grover's QSA for 4 qubits (target state |1111>). The orange bars indicate the unmitigated probabilities, while the blue bars indicate the mitigated probabilities (even though they are the same for a noiseless simulation). Predictably, the target state measurement probabilities are nearly 1.**

Hereafter the results are executed on the real backend, IBM Kyoto. The following are the results for measurement-error mitigation with calibration matrices for the 2-qubit implementation of the algorithm. The target state is |11⟩. The number of shots used for the experiments was 8192 (this is the number of times that the algorithm as a whole was repeated), since that is the default used by Qiskit experiments to construct its calibration matrices.

In Figure 13, it can be seen that the probability for measuring the |11⟩ state is the highest, and it is boosted by the implementation of measurement error mitigation with the use of calibration matrices. In fact, it has increased by 18.29%.

The other two methods produce unsatisfactory results. Mthree introduces negative probabilities, which shouldn't happen, while the use of dynamical decoupling records a percentage decrease in the probability of measuring the target state, |11⟩, meaning that using this method worsens the performance of the algorithm.

The state fidelity of this experiment was found to be 0.62.

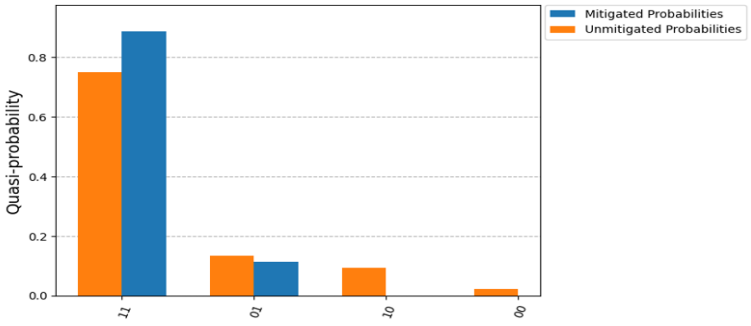


FIG. 13. **Measurement results showing the quasi-probabilities for each of the possible states in the sample space. The orange bars indicate the unmitigated probabilities, while the blue bars indicate the mitigated probabilities.**

The following are the results for measurement-error mitigation with calibration matrices for the 3-qubit implementation of the algorithm. The target state is |111⟩. The number of shots used for the experiments was 8192 (this is the number of times that the algorithm as a whole was repeated), since that is the default used by Qiskit Experiments to construct its calibration matrices.
In Figure 14a), it can be seen that the probability of measuring the |111⟩ state is the highest, and it is boosted by the implementation of measurement error mitigation with the use of calibration matrices. In fact, it has increased by 18.46%.

In Figure 14b), it can be seen that the probability of measuring the |111⟩ state is also the highest, and it is boosted by the implementation of measurement error mitigation with the use of Mthree. It has increased by 29.34%.

In Figure 14c), it can be seen that the probability of measuring the |111⟩ state after implementing dynamical decoupling is the highest. In fact, it is considerably higher than the probability of measuring any other outcome. The percent increase with dynamical decoupling is 34.92%.

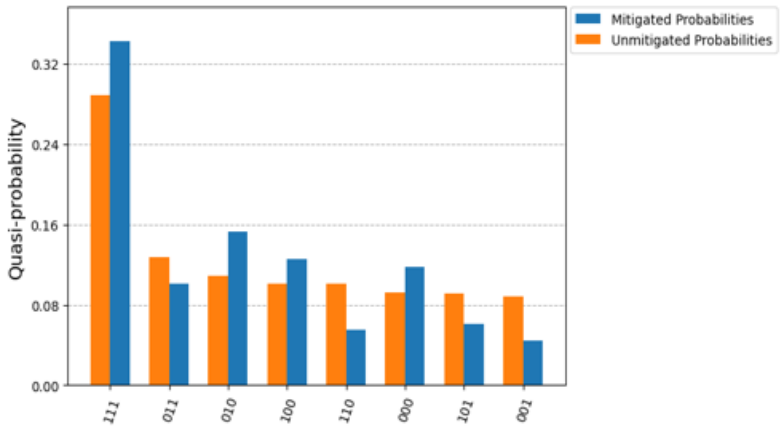


FIG. 14a). **Measurement results showing the quasi-probabilities for each of the possible states in the sample space, with and without readout error mitigation through calibration matrices. The orange bars indicate the unmitigated probabilities, while the blue bars indicate the mitigated probabilities.**

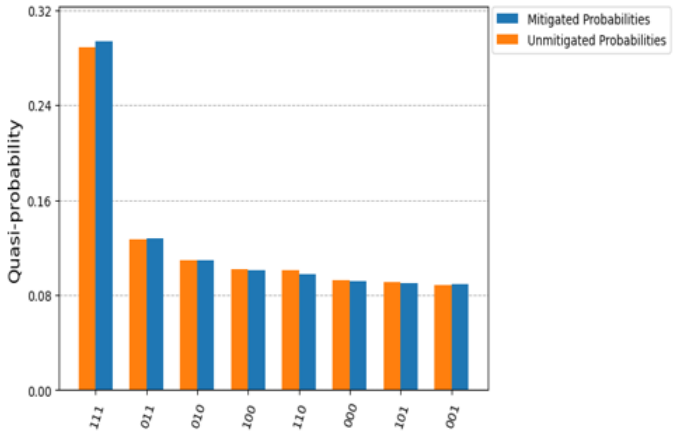


FIG. 14b). **Measurement results showing the quasi-probabilities for each of the possible states in the sample space, with and without readout error mitigation through Mthree. The orange bars indicate the unmitigated probabilities, while the blue bars indicate the mitigated probabilities.**

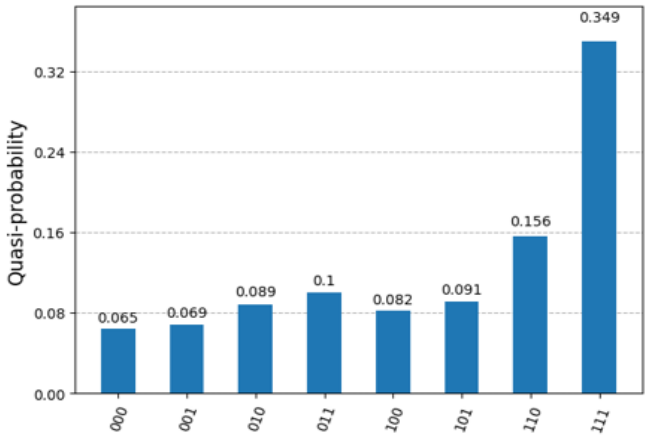


FIG. 14c). **Measurement results showing the quasi-probabilities for each of the possible states in the sample space, with error mitigation through dynamical decoupling. The blue bars indicate the mitigated probabilities.**

The state fidelity of this experiment is 0.59.

The following are the results for measurement-error mitigation with calibration matrices for the 4-qubit implementation of the algorithm. The target state is |1111⟩. The number of shots used for the experiments was 8192 (this is the number of times that the algorithm as a whole was repeated), since that is the default used by Qiskit experiments to construct its calibration matrices.

In Figure 15a), it can be seen that the probability of measuring the |1111⟩ state is not the highest, and it is in fact reduced by the implementation of measurement error mitigation with the use of calibration matrices. It is reduced by around 10.12%. Since the probability of measuring the |1001⟩ state is drastically increased, indicating systematic error, it seems that the increased noise arising from the larger circuit depth has resulted in the wrong state being mitigated.

In Figure 15b), it can be seen that the probability of measuring the |1111⟩ state is not the highest, and it is reduced by the implementation of measurement error mitigation with the use of Mthree. It is reduced by 1.95%. Mthree's results give a more even spread between all outcomes, indicating random error.

In Figure 15c), it can be seen that the probability of measuring the |1111⟩ state after implementing dynamical decoupling is still the highest. In fact, it is considerably higher than without the implementation of the mitigation technique. The percent increase with dynamical decoupling is a whopping 351.6%.

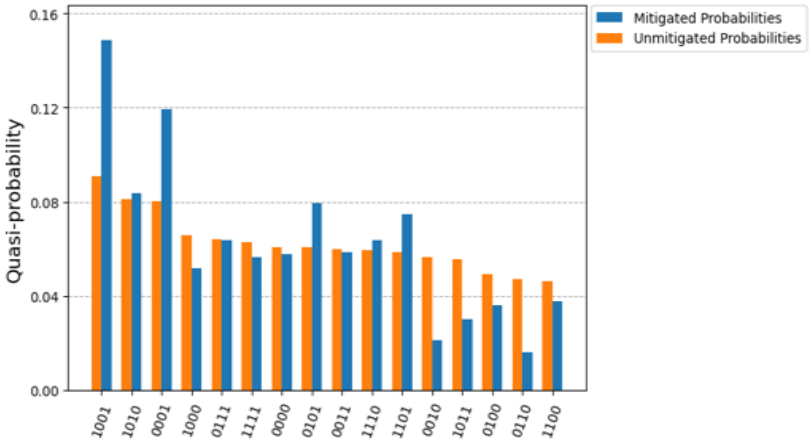


FIG. 15a). **Measurement results showing the quasi-probabilities for each of the possible states in the sample space, with and without readout error mitigation through calibration matrices. The orange bars indicate the unmitigated probabilities, while the blue bars indicate the mitigated probabilities.**

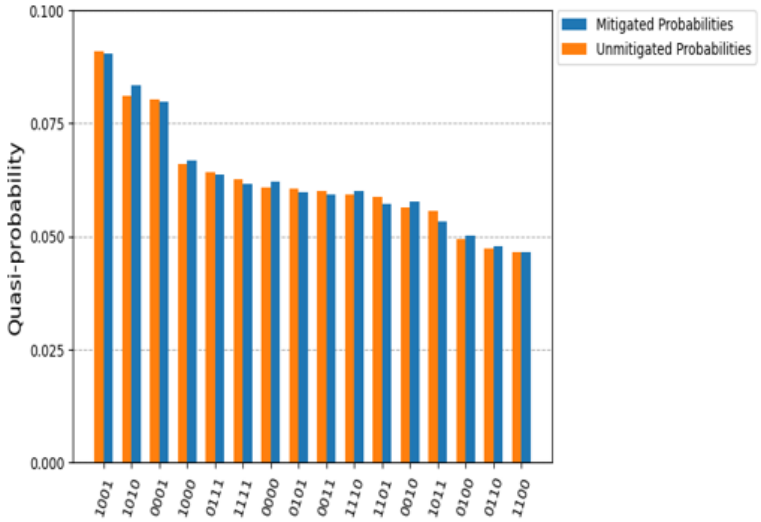


FIG. 15b). **Measurement results showing the quasi-probabilities for each of the possible states in the sample space, with and without readout error mitigation through Mthree. The orange bars indicate the unmitigated probabilities, while the blue bars indicate the mitigated probabilities.**

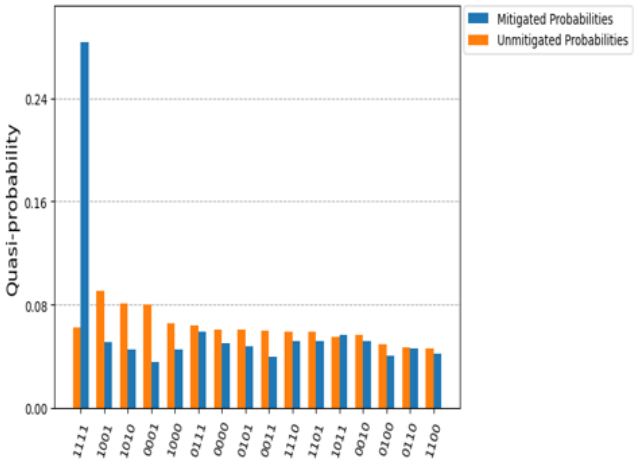


FIG. 15c). **Measurement results showing the quasi-probabilities for each of the possible states in the sample space, with error mitigation through dynamical decoupling. The blue bars indicate the mitigated probabilities.**

The state fidelity of this experiment is 0.53.

Conclusions

From the results of the three experiments, the first notable trend is the declining state fidelity (from 0.62 to 0.59 to 0.53) with an increasing number of qubits, which is to be expected. With an increasing number of qubits, the circuit depth increases and consequently, noise increases. Thus, a state cannot be prepared as well as it would be with a lesser number of qubits.

Secondly, the fact that M three and dynamical decoupling show unexpected behavior for 2 qubits may stem from the fact that these techniques are trained to work on circuits with larger numbers of qubits, where the issue of noise would be more apparent. A likely explanation for the fact that dynamical decoupling did not work as desired may lie in the fact that when the circuit depth is short,

as in the case of a 2-qubit Grover's QSA, the number of idle periods of the qubits are shorter and in these cases, dynamical decoupling can actually introduce unwanted error. This explains the reduction in the probability of measuring the target state. However, the introduction of negative probabilities with M three merits further study, especially as this was only found to happen for the case of 2 qubits.

Thirdly, all three techniques work as desired for the case of 3 qubits. This may be due to the fact that in this case, the circuit depth is intermediate (not too short to be an outlier, not too long for noise to be too apparent - seemingly confirming Figure 3). Notably, dynamical decoupling was found to outperform readout mitigation with calibration matrices, which in turn outperformed readout mitigation with M three.

And lastly, for the case of 4 qubits, where the circuit depth is high enough for noise to become an issue, only dynamical decoupling worked as desired. In fact, it worked better than it did for the case of 3 qubits. This may be due to the fact that as circuit depth increases, the number of idle periods becomes larger, so the effect of mitigating any errors that might arise during these periods is larger.

In this paper, Grover's quantum search algorithm was implemented in Qiskit on the IBM Kyoto quantum processor, and three techniques of error mitigation - readout error mitigation with calibration matrices, readout error mitigation without such matrices (through the package M three) and dynamical decoupling - were compared in terms of their effect in mitigating the probabilities of measuring the target state for 2, 3 and 4-qubit implementations. Dynamical decoupling was found to be the most promising candidate relative to the other two, while M three was the least satisfactory, perhaps owing to the fact that it works the best when the measurement error probability is quite low. Still, M three's paradoxical effects merit further investigation. Additionally, quantum state tomography was applied to track the evolution of the state fidelity across the implementations with different numbers of qubits. Predictably, the state fidelity steadily decreased with an increasing number of qubits.

References

1. Kumar T, Kumar D, Singh G. Error mitigation of Grover's quantum search algorithm. Indian J Pure Appl Phys. 2022;60(8).
2. Azses D, Dupont M, Evert B, et al. Navigating the noise-depth tradeoff in adiabatic quantum circuits. Phys Rev B. 2023;107(12):125127.
3. Preskill J. Quantum computing in the NISQ era and beyond. Quantum. 2018;2:79.
4. Yang B, Raymond R, Uno S. Efficient quantum readout-error mitigation for sparse measurement outcomes of near-term quantum devices. Phys Rev A. 2022;106(1):012423.
5. Goudarzi H, Dousti MJ, Shafaei A, et al. Design of a universal logic block for fault-tolerant realization of any logic operation in trapped-ion quantum circuits. Quantum Inf Process. 2014;13(5):1267-1299.
6. Kumar T, Kumar D, Singh G. Noise analysis of Grover's quantum search algorithm. Indian J Pure Appl Phys. 2023;61(5):355-365.
7. Beisel M, Barzen J, Leymann F, et al. Configurable readout error mitigation in quantum workflows. Electronics. 2022;11(19):2983.
8. Pokharel B, Lidar DA. Better-than-classical Grover search *via* quantum error detection and suppression. npj Quantum Information. 2024;10(1):23.
9. Zhang K, Rao P, Yu K, et al. Implementation of efficient quantum search algorithms on NISQ computers. Quantum Inf Process. 2021;20(7):233.