Future Generation Computer Systems 182 (2026) 108451

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Applying quantum error-correcting codes for fault-tolerant blind quantum cloud computation

Qiang Zhao [a,*], John C.S. Lui [b,1]

[a] Department of Computer Science and Technology, Zhejiang Ocean University, ZhouShan, 316000, ZheJiang, China
[b] Department of Computer Science and Engineering, The Chinese University of Hongkong, 999077, HongKong, China

ARTICLE INFO

Keywords:
Blind quantum computation
Measurement-Based quantum computation
Quantum error-correcting code
Cluster state.

ABSTRACT

Blind Quantum Computation (BQC) is a delegation computing protocol that allows a client to utilize a remote quantum server to implement desired quantum computations while keeping her inputs, outputs, and algorithms private. However, qubit errors during quantum computation are important issues that need to be addressed. In this paper, we propose a fault-tolerant blind quantum computation protocol with quantum error-correcting codes to avoid the accumulation and propagation of qubit errors during the computational process. We further propose to apply the concatenated codes in the protocol to improve the error correction performance. More importantly, we quantify the resource consumption of photon pulses by our optimal level concatenation codes. Extensive simulation results show that our scheme not only can improve the preparation efficiency but also reduce quantum resource consumption, which shows a significant improvement to realize a fault-tolerant BQC.

1. Introduction

Quantum cloud computing is a promising computing paradigm that can bring revolutionary improvements in communications and computations, as well as deal with tasks that are computationally impossible for traditional computers. Quantum cloud computing is a distributed system composed of quantum computers as servers and simple quantum devices as clients that exchange quantum information and computing instructions through quantum and classical channels. It allows clients to utilize remote quantum computing resources to achieve their desired quantum computation. However, classical communication network protocols can not ensure the security of quantum information and computation. One possible solution to address this problem is to design a Blind Quantum Computation (BQC) protocol for quantum cloud computing. BQC enables a client (say Alice) with limited quantum resources to delegate a computing task to a remote quantum server (say Bob) while ensuring privacy during the computing process [1–3]. In recent years, a number of BQC protocols have been proposed [4–9]. Among these protocols, Universal Blind Quantum Computation (UBQC) is of particular interest as it not only can guarantee that Alice's inputs, outputs, and quantum algorithms are private to Bob but also only requires Alice to prepare single photon states for easy realization.

In recent years, the architecture of BQC protocols has evolved from single-server to multi-server and multi-client frameworks to address increasingly diverse application scenarios. Polacchi et al. proposed and experimentally demonstrated a lightweight two-client blind quantum computation protocol based on the Qline, highlighting its potential for scalability in larger quantum networks [10]. This work was subsequently extended to an experimental implementation of verifiable multi-client BQC on the same Qline architecture [11]. However, the multiple servers in BQC may also increase the communication complexity and resource-coordination overhead of such protocols. To enhance the security of BQC, Drmota, Nadlinger, and Main et al. demonstrated a verifiable blind quantum computing scheme using trapped ions and single photons, providing a quantitative assessment of information leakage [12]. In practice, BQC also depends critically on computational efficiency and resource overhead. Ma, Zhu, Liu et al. constructed three improved brickwork states for UBQC that significantly reduce client qubit consumption [13]. Lee and Chuang proposed a partial BQC protocol, in which only the privacy-sensitive components of a quantum circuit are blinded, thus lowering the overall computational overhead [14]. These efficiency-oriented advances–whether through refined resource states, partial blinding, or optimized measurement strategies–aim to relax the demands on client-side resources and server-side operational complexity, thereby advancing the feasibility of BQC on near-term Noisy Intermediate-Scale Quantum (NISQ) devices.

Real quantum systems are inevitably subject to noise, which introduces computational errors. The goal of Fault-Tolerant Blind Quantum

* Corresponding author.
E-mail address: qiangzhaoict@126.com (Q. Zhao).
[1] Fellow, IEEE, ACM

https://doi.org/10.1016/j.future.2026.108451
Received 13 December 2024; Received in revised form 9 January 2026; Accepted 21 February 2026
Available online 24 February 2026

Computation (FT-BQC) is to integrate quantum error correction within the BQC framework, ensuring the correctness of the results and the blindness of the protocol. Detailed analysis of error sources and their impact on computational outcomes in existing BQC experiments [11,12] provides a valuable reference to evaluate FT-BQC performance under realistic noise conditions. In the early work, Broadbent et al. proposed using error-correcting encodings based on brickwork states to correct errors arising during computation [3], but this method consumes substantial quantum resources. Lightweight error-mitigation strategies have also been proposed [15,16], but these primarily serve to warn or filter errors rather than correct them completely. To ensure blindness during error correction, researchers have pursued the use of quantum error-correcting codes (QECCs) to realize FT-BQC [7,17,18], focusing both on optimizing existing code families and exploring novel coding structures [19–21]. Google's quantum computing team has experimentally validated that surface codes can be used to implement fault-tolerant quantum computation [22]. Although surface codes offer a high error threshold, their encoding scheme requires significant resource overhead. Barane et al. used the surface code and a hybrid light-matter approach to develop an architecture for scalable FT-BQC [23]. This approach improves the error-correction threshold and increases the speed and depth of blind logical circuits. Ruiz et al. proposed a novel LDPC-cat code architecture combining LDPC codes with cat-state qubits [24], which can substantially reduce physical qubit overhead and hardware complexity. However, within the measurement-based quantum computation (MBQC) framework of BQC protocols, LDPC codes face challenges related to complex syndrome extraction and a lack of fault-tolerant gate sets compatible with graph-state operations. Concatenated codes achieve very low logical error rates by combining two or more layers of encoding. Pato et al. proposed a concatenated Steane code with single-flag syndrome checks [25], capable of correcting up to four arbitrary physical errors and thereby drastically suppressing the logical error rate. Furthermore, the Steane code belongs to the CSS code family, allowing for the transversal implementation of Clifford gates, which simplifies the realization of logical operations in BQC. Nevertheless, integrating QECCs into BQC requires careful design to ensure that new protocols preserve the blindness within a fault-tolerant framework [26,27].

Note that the current quantum technology still struggles with noises and imperfect measurement, which results in a high error rate in the FT-BQC. To address this problem, we propose to use the concatenated stabilizer codes to reduce qubit requirements by tailoring circuits to suppress the dominant effect of qubit errors [28,29]. Chamberland, Jochym, and Laflamme et al. concatenated the 7-qubit Steane code with the 15-qubit Reed-Muller code to implement universal fault-tolerant quantum computation without state distillation [30]. Paul Webster et al. presented a general framework for universal fault-tolerant logic with no-go theorem and stabilizer codes [31], which can be applied to a wide range of stabilizer code families, including concatenated codes and conventional topological stabilizer codes. Fault-tolerant quantum computation with concatenated quantum codes was proposed by Chamberland, Noh, and Preskill et al. [29], which can reduce the consumption of qubits by tailoring the quantum error-correcting codes to suppress the dominant phase-flip errors. To avoid coherent errors, Yingkai Ouyang proposed rotated concatenated stabilizer codes [32], namely, concatenating an [[n,k,d]] stabilizer outer code with constant-excitation inner codes. It was shown that when the stabilizer outer code is fault-tolerant, the concatenated codes are immune from coherent phase errors. The concatenation codes above are useful for improving the fault-tolerant threshold and reducing quantum resource consumption in a realistic UBQC system.

Recently, many encoding methods [25,33,34]were proposed to deal with the high error rate for qubits, such as entanglement-assisted code [35], surface code [36] and concatenated code [37]. Since quantum error correction imposes a heavy overhead cost in many qubits and gates to ensure the required fault tolerance, there is a fundamental trade-off between fault tolerance and high resource overhead (the number of required pulses). In this context, this work provides a general fault-tolerant blind quantum computation with concatenated stabilizer codes to reduce quantum resource consumption, in terms of the number of photon pulses. Moreover, the concatenated stabilizer codes we design can also be applied to other types of BQC and perform fault-tolerant quantum computation. The main contributions of this paper are as follows:

- We present a remote blind quantum error-correcting code preparation protocol on cluster states to correct qubit errors and improve the successful probability of desired qubits. The encoded circuit is transformed to cluster states, i.e. multi-particle entangled graph states. Then, the server prepares quantum error-correcting codes in the framework of measurement-based quantum computation.
- Based on the above preparation, we propose a fault-tolerant blind quantum computation protocol with quantum error-correcting codes to avoid the accumulation and the propagation of qubit errors to improve the performance in a delegated quantum computation. Since these quantum error-correcting codes are unknown logical qubits to the server, each of them is able to be considered as a logical unit to take the place of the original qubit on the brickwork state to perform a fault-tolerant computation. Furthermore, we provide a demonstration of the ϵ-blindness property in the presence of a malicious server, highlighting the robustness of our protocol even in adversarial noise settings.
- To achieve higher fault-tolerant performance, we propose to utilize the concatenated stabilizer codes, that is to concatenate the same or different stabilizer codes in a multi-level encoding manner, and to prepare high-quality encoded logical qubits to do fault-tolerant computation. Theoretically, we also derive the lower bound of quantum resource consumption in each level of the concatenation codes, i.e. the number of required pulses.
- To reduce quantum resource overhead in a fault-tolerant quantum computation, we formulate an optimization model of quantum resource consumption, which can describe the ratio of the number of pulses for concatenation codes to the non-coding case under the condition of having the same probability of successful qubit preparation. When resource consumption reaches the minimum, one can achieve the optimal number of levels in the concatenated stabilizer codes.

This paper is organized as follows: In Section 2, the background and challenges are introduced. In Section 3, we present a remote blind quantum error-correcting codes preparation protocol on cluster state to correct qubit errors, and then propose a fault-tolerant blind quantum computation with quantum error-correcting codes to perform the delegated computing. Furthermore, we also present a proof of achieving blindness for our fault-tolerant protocol. In Section 4, the multi-level quantum error-correcting codes are concatenated to reduce the error rate of the generated logical qubits. we derive a lower bound of the number of required pulses to evaluate quantum resource consumption for different levels of concatenation codes. In Section 5, we present simulation results to validate the theoretical model of the number of required pulses with concatenated codes, and give an optimal level of the concatenation codes. In Section 6,we state our conclusions and future work.

2. Background

In the framework of MBQC, the underlying resource of UBQC is the brickwork states, which are multi-particle entangled graph states constructed from the prepared qubits [3,38]. A delegated quantum computation can be implemented by measurements on the brickwork state. To illustrate, consider a client (say Alice) and a server (say Bob) with a full-fledged quantum computer. In the preparation stage, Alice prepares a series of single qubits $|+_\theta\rangle$ with random polarization $\theta \in \{0, \pi/4, \ldots, 7\pi/4\}$, and sends them to Bob through a one-way quantum channel, and Bob uses Controlled Z (CZ) gates to act on these received qubits to build a brickwork state. In the interactive measurement stage, Alice meticulously designs the quantum circuit tailored to

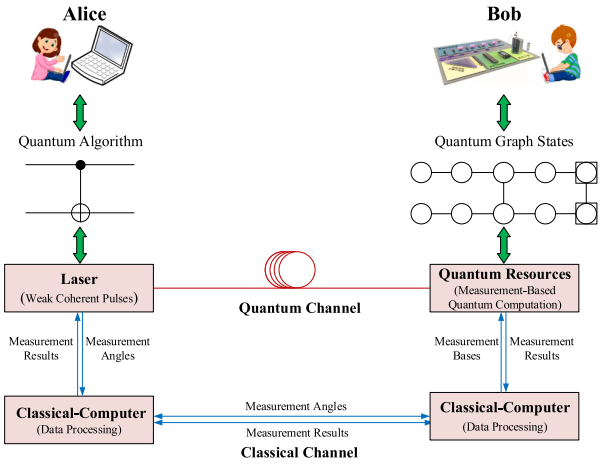

Fig. 1. The schematic diagram of UBQC.

the computational task at hand, and transforms the sequence of ordered quantum gates in the circuit into corresponding measurement angles, and then sends them to Bob through a two-way classical channel. Bob performs measurement operations on each qubit of the brickwork state, and returns the results to Alice. The measurement procedure is repeated on each qubit of the brickwork state until the desired quantum computation result is obtained. The schematic diagram of UBQC is shown in Fig. 1.

Note that for each qubit in a brickwork state, Alice is able to calculate the measurement angle $\phi'_{x,y}$ based on her desired angle $\phi_{x,y}$ and the previous measurement outcomes [3]. In order to ensure the privacy of the computation, the measurement angle $\phi_{x,y}$ needs to be further processed as $\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$, $r_{x,y} \in_R \{0,1\}$. Then, the real measurement angle $\delta_{x,y}$ is sent to Bob to perform the fault-tolerant measurement on each encoded logical qubit $\left|\psi_{x,y}\right\rangle_L$ of the encoded brickwork state. The measurement result $s_{x,y}$ is returned to Alice. If the random bit chosen by Alice is $r_{x,y} = 1$, Alice then flips the measurement result $s_{x,y}$; otherwise, she does nothing.

2.1. Remote blind qubit state preparation protocol with two decoy states

For a UBQC, qubits are encoded via the polarization of the single photons using a realistic single-photon source. However, in real implementation, it is possible to send two or more identically polarized photons instead of one, and this may destroy the privacy of the client. Let the ideal joint state shared by the client and the server be described by the state π_{AB}^{ideal}, and that a malicious server can not infer anything about the client's information. For any UBQC protocol, the ideal joint state π_{AB}^{ideal} is evolved as one of the family of states, i.e. $\mathcal{F}(\pi_{AB}^{ideal})$. If the security can be held for any action of the server, any state of the family should be equally blind. In order to analyze the security in a realistic implementation, we consider the settings where the client sends a general state ρ^{θ_i} instead of a perfect state $|+_{\theta_i}\rangle$. The definition of ϵ blindness is described as follows [4]:

Definition 1. A UBQC protocol with imperfect preparation is ϵ-blind if the trace distance between the ideal state $\pi_{AB}^{\theta_i}$ and realistic state $\pi_{AB}^{\rho^{\theta_i}}$ is less than $\epsilon > 0$. Finally, we have:

$$\min_{\pi_{AB}^{\theta_i} \in \mathcal{F}} \frac{1}{2} \left\| \pi_{AB}^{\{\rho^{\theta_i}\}} - \pi_{AB}^{\{\theta_i\}} \right\| \leq \epsilon. \tag{1}$$

To achieve the above security, a Remote Blind qubit State Preparation (RBSP) protocol [4] was proposed to prepare ϵ-blind qubits for any security parameter $\epsilon > 0$. Although one has an ϵ-blind UBQC protocol, the client's overhead, which is quantified by the number of pulses needed for preparing a qubit, is usually very large for long-distance communication. Therefore, a modified RBSP with two decoy states was presented by Zhao and Li [5,6] to optimize the number of required pulses. They showed that the two-decoy state protocol only needs a smaller number of pulses than one decoy state. In addition, Yang-Fan Jiang, Kejin Wei and Pan jian-wei etc. have demonstrated a resource-efficient RBSP protocol with decoy states in the field test over 100km of fiber [39], which validates the feasibility of UBQC over long distances, thus taking a key step towards secure cloud quantum computing. The RBSP with two decoy states is summarized as follows:

1. Alice generates N weak coherent pulses which contain the signal and two decoy states, and sends them to Bob.
2. For each received state, Bob reports to Alice which pulses were received. Based on the reported statistic values, Alice calculates the gains in the signal state and two decoy states, and determines to continue the protocol if they are within the specified thresholds.
3. Bob discards the decoy states declared by Alice and separates the remaining signal states into S groups (in fact S is the computational scale). Each group only needs to perform the interlaced 1-D cluster computation subroutine(I1DC) [4] so that the desired qubit $|+_\theta\rangle$ can be prepared.
4. With Alice's knowledge about the angles of signal states in the I1DC procedure and outcomes reported by Bob, Alice can compute the polarization angle θ of the prepared qubit.

For a UBQC protocol with computation size S, if the client's preparation is ϵ robust and ϵ blind for a chosen $\epsilon > 0$, the lower bound of the total number of required pulses in RBSP with two decoy states, denoted as N^{L,v_1,v_2}, satisfies this following inequality:

$$N \geq N^{L,v_1,v_2} = \frac{S}{p_\mu Q_\mu} \frac{\ln(\epsilon/S)}{\ln\left(1 - p_1^{L,v_1,v_2}\right)}, \tag{2}$$

where the average photon number of signal states is denoted as μ, and the average photon number of two decoy states is denoted as v_1, v_2. The gain of signal states is Q_μ, which is the probability of the detecting event of the signal pulses. The proportion of single photons in the signal states is described as p_1, and the lower bound is p_1^{L,v_1,v_2}. The probabilities of signal and two decoy states chosen by the client are defined as p_μ, p_{v_1}, p_{v_2}, respectively, with $p_\mu + p_{v_1} + p_{v_2} = 1$. The detailed discussion is shown in [6,40].

2.2. Quantum error-correcting codes

To correct qubit errors, one needs to first establish an error model for quantum computation. Based on the result of [41], we know that the evolution of the qubit state $|\psi\rangle$ can be expressed as a linear combination of four possibilities: (1) no error occurs, (2) the bit flip $|0\rangle \leftrightarrow |1\rangle$, (3) the phase flip $\alpha|0\rangle + \beta|1\rangle \leftrightarrow \alpha|0\rangle - \beta|1\rangle$, (4) both a bit flip and a phase flip occur. Therefore, the quantum state after noise $E_i|\psi\rangle$ can be written as a superposition of four terms $I|\psi\rangle, X|\psi\rangle, Z|\psi\rangle, XZ|\psi\rangle$. We can use a trace-preserving quantum operation $\mathcal{E}$ to describe the noise, which is convenient to analyze the error model by expanding $\mathcal{E}$ in an operator-sum representation with operation elements$\{E_i\}$. The error model has the following form

$$\mathcal{E}(|\psi\rangle\langle\psi|) = \sum_{E_i \in E} E_i |\psi\rangle\langle\psi| E_i^\dagger, \tag{3}$$

where all error $\{E_i\}$ are the linear combinations of Pauli operator, $E_i = p_{i,1} I + p_{i,2} X + p_{i,3} Z + p_{i,4} XZ$, $\sum_{k=1}^{4} p_{i,k} = 1$, and $p_{i,k}$ is the probability for each case to occur. In Eq. (3), note that $\mathcal{E}$ is the trace-preserving operation, so the operation elements $\{E_i\}$ satisfy $\sum_i E_i^\dagger E_i = \mathbf{I}$. For correcting qubit errors, we need to determine which of these four possibilities

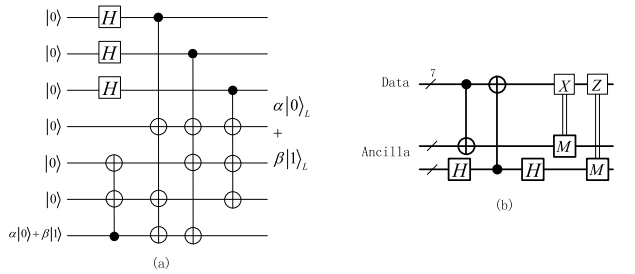

Fig. 2. The encoding and correction circuits of the [[7,1,3]] code [41,44]. (a) An unknown logical qubit and 6 ancilla qubits can be used to encode into [[7,1,3]] code. (b) Two 7-qubit Steane states are used to correct error qubits for a 7-qubit data block. Each CNOT gate in the diagram represents 7 CNOT gates performed in parallel.

occurred, and then correct the error by applying the appropriate Pauli basis.

In the procedure for determining an error syndrome, one needs to make sure that the quantum state can not be destroyed for subsequent quantum computation, and at the same time, its information has to be private. Therefore, one needs to use a suitable quantum code to handle these qubits, which may occur with errors. A popular quantum code is the $[[n, k, d]]$ stabilizer code [41–43], which can encode k qubits into n qubits, when $n > k$. The parameter d is the distance of the code. If the distance of a code is $d \in \mathbb{N}$, it can correct up to $\lfloor d/2 \rfloor$ simultaneous errors from the error set E. In quantum error correction, the diagnosis of error is often called the error syndrome measurement, which is increasingly difficult as the number of logical qubits increases in an encoded block. To illustrate, consider a common stabilizer code with seven qubits, i.e. the 7-qubit Steane code [[7,1,3]], which can encode one data qubit in these seven qubits, and can correct the one-qubit error. The encoded logical qubit basis is denoted as $\{|0\rangle_L, |1\rangle_L\}$, and they are Eq. (4).

$$\begin{aligned}|0\rangle_L = \frac{1}{2\sqrt{2}}(&|0000000\rangle + |0001111\rangle + |0110011\rangle \\ &+ |0111100\rangle + |1010101\rangle + |1011010\rangle \\ &+ |1100110\rangle + |1101001\rangle) \\ |1\rangle_L = \frac{1}{2\sqrt{2}}(&|1111111\rangle + |1110000\rangle + |1001100\rangle \\ &+ |1000011\rangle + |0101010\rangle + |0100101\rangle \\ &+ |0011001\rangle + |0010110\rangle)\end{aligned} \tag{4}$$

Note that the [[7,1,3]] encoding circuit of quantum error-correcting codes can be designed as shown in Fig. 2(a). An unknown logical qubit $\alpha|0\rangle + \beta|1\rangle$ and six ancilla qubits $|0\rangle$ are encoded into $\alpha|0\rangle_L + \beta|1\rangle_L$ [41]. In Fig. 2(b), two ancilla states are prepared to perform error syndrome measurements of bit flip and phase flip. The measurement results of ancilla qubits are multiplied by the row vectors of $G_{[[7,1,3]]}$ to obtain the parity bits, which can diagnose the error syndrome. Finally, one can use the Pauli operator to correct the qubit errors in the encoded data block.

2.3. Challenges

In quantum computation, a basic qubit unit is often stored as the polarization of a single photon or the spin of a single electron. A qubit has the superposition property, which can be described by the two-dimension Hilbert space. A quantum algorithm is described by the quantum circuit which consists of a sequence of quantum gates. In UBQC, the quantum gates are first transformed into the brickwork, and then the server performs measurements on each qubit of the brickwork state to implement the client's computation based on MBQC. Note that quantum gates can be realized by performing measurements on qubits. In a practical UBQC system, one only needs to consider the qubits used. Due to the influence of noise, qubits are prone to polarization errors. Therefore, it is necessary to integrate the quantum error correction feature into UBQC to ensure the correctness of a delegated computation. In UBQC, the main processes of quantum computation consist of preparation and measurement. Hence, we state the challenges in terms of the following two processes.

Challenge I: In the preparation, we modify the above RBSP protocol to prepare the quantum error-correcting codes instead of single qubits. To prepare quantum error-correcting codes, we first design a quantum encoding circuit and then transform it into graph states. On the basis of the MBQC model, we can delegate a server to prepare quantum error-correcting codes on graph states. However, both good encoded circuits and good graph states require a large number of data qubits and ancilla qubits, which directly determine the size of the computation. The challenge is to achieve better error correction performance while having a low demand for quantum resources.

Challenge II: In the measurement process, the client sends measurement angles to the server, which uses these angles to perform measurement-based computations on the brickwork state. To prevent the accumulation and propagation of qubit errors in computing, it is crucial to perform fault-tolerant quantum computation. This requires the qubits, quantum gates, and measurements in the computation to be fault-tolerant in the MBQC setting. To achieve this, each qubit of the brickwork state can be encoded as a logical unit using quantum error-correcting code. In this process, one challenge is to ensure the proposed fault-tolerant blind quantum computation protocol is ϵ-blind. Another challenge is to balance the trade-off between the quantum resources consumption and the level of fault tolerance, as a higher fault tolerance often requires a larger number of redundant qubits to counteract the spread of qubit errors.

3. Fault-tolerant blind quantum computation with quantum error-correcting codes

To effectively eliminate qubit errors, the encoded quantum gates need to ensure that a failure in executing any encoded gate can only be propagated to a small number of qubits in each encoded data block so that the aggregated error rate does not exceed the designed fault-tolerant threshold. In addition, since any error correction scheme may introduce errors in the encoded qubits, one must be careful in designing error correction procedures. As a result, we illustrate the following steps to prevent the accumulation and propagation of errors in fault-tolerant quantum computing [41]: (1) fault-tolerant preparation of quantum error-correcting codes, (2) fault-tolerant error correction operation, (3) fault-tolerant quantum gates application, and (4) fault-tolerant measurements. This general paradigm is shown in Fig. 3.

In UBQC, a fault-tolerant computation usually needs to follow the paradigm in Fig. 3. Alice first delegates to Bob to perform the fault-tolerant preparation of quantum error-correcting codes. Then Bob uses these encoded logical qubits to create an encoded brickwork state to realize fault-tolerant quantum computing. In the fault-tolerant preparation, the CNOT gates between nonadjacent qubits are repeatedly used in the fault-tolerant encoded circuit, so a large number of SWAP gates are needed to ensure CNOT gates can act on neighboring qubits, which requires many auxiliary qubits. If these CNOT gates of the encoding circuit can be implemented with high probability, then the error rate of the prepared quantum error-correcting code will be greatly reduced. In fault-tolerant computation, fault-tolerant quantum gates of the desired quantum circuit need to be equivalently transformed into the brickwork state. Then, fault-tolerant measurements are performed on each encoded logical qubit of the brickwork state in the framework of MBQC. Consequently, with limited quantum resources, a fault-tolerant blind quantum computation can be implemented by using a combination of quantum error-correcting code preparation and fault-tolerant measurements.

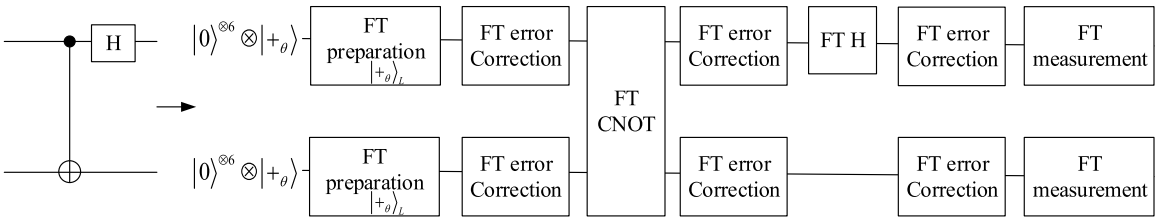

Fig. 3. The general process of fault-tolerant quantum computation.

3.1. Delegated preparation of quantum error-correcting codes

In quantum computation, any arbitrary quantum gate can be realized via a cluster state. The essence of a quantum circuit is a series of ordered quantum gates. Hence, each quantum circuit can be implemented via a cluster state [45]. The encoded circuit can also be realized via a cluster state to generate the encoded logical qubits. These codes can be considered as "*logical units*" to take the place of the original qubits on a brickwork state to perform fault-tolerant computing. Hence, we propose a remote quantum error-correcting code preparation protocol on a cluster state, which is shown in Protocol 1

Protocol 1: A remote quantum error-correcting code preparation on cluster state.

Input: data weak coherent pulses (signal states and two decoy states) with polarization ρ^σ, $\sigma \in_R \{k\pi/4 : 0 \le k \le 7\}$; ancilla pulses with polarization $|+\rangle$.

Output: quantum error-correcting codes $\left\{|+_{\theta_i}\rangle_L\right\}_1^S$.

1 Alice sends data and ancilla pulses to Bob.
2 **for** $i = 1$ *to* S; **do**
3 Bob prepares qubit $|+_{\theta_i}\rangle$ based on RBSP with two decoy states, and uses it and a group of ancilla qubits $\{|+\rangle\}$ to build cluster state.
4 **for** $x = 1$ *to* m, $y = 1$ *to* n; **do**
5 Realization of the quantum encoded circuit on cluster state.
6 **if** q_{xy} *is white* **then** q_{xy} is measured with $\{|0\rangle, |1\rangle\}$;
7 **if** q_{xy} *is green* **then** q_{xy} is measured with $M(0)$;
8 **if** q_{xy} *is red* **then** q_{xy} is measured with $M(\pi/2)$;
9 **end**
10 Bob prepares the encoded logical qubit $\left\{|+_{\theta_i}\rangle_L\right\}$.
11 **end**
12 **return** $\left\{|+_{\theta_i}\rangle_L\right\}$.

Note that the depth of [[7,1,3]] circuit in Fig. 2(a) is about four, and the most commonly used gate is the CNOT gate. If these CNOT gates can be operated with a high success rate, then the success rate of qubit preparation will also be high. According to the quantum computation model on cluster states [45], the quantum gates of [[7,1,3]] circuit can be transformed into a cluster state, and sequentially measure each qubit on the cluster state to prepare the quantum error-correcting code. In the following, we will use the [[7,1,3]] circuit as an example to illustrate how to prepare quantum error-correcting code.

Based on Protocol 1, Alice sends data photon pulses and ancilla photon pulses to Bob. Bob prepares data qubits based on RBSP with two decoy states [6], and then utilizes a series of ancilla qubits and a prepared data qubit to build the initial cluster state. According to the [[7,1,3]] circuit in Fig. 2(a), Alice can transform each quantum gate into measurement angles on cluster state, as shown in Fig. 4. After that, Bob uses the computational basis $\{|0\rangle, |1\rangle\}$ to eliminate the redundant qubits according to Alice's requirements. The remaining qubits on this cluster state are used to prepare quantum error-correcting code based on Alice's measurement basis $M(\delta)$, which is defined by orthogonal projections on $|\pm_\delta\rangle = (|0\rangle \pm e^{i\delta}|1\rangle)/\sqrt{2}$. The parameter $\delta \in [0, 2\pi]$ is called the measurement angle. For $\delta = 0$ or $\pi/2$, one obtains the X or Y Pauli measurement. The measurement outcome at qubit i will be denoted as $s_i \in \mathbb{Z}_2$. We take the specific convention that $s_i = 0$ if the state collapses to $|+_\delta\rangle$, and $s_i = 1$ if the state collapses to $|-_\delta\rangle$.

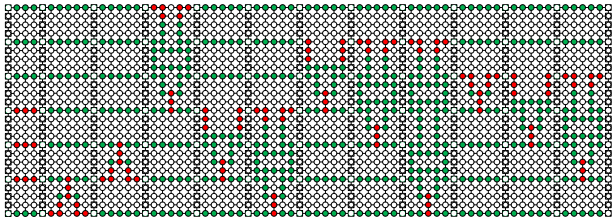

Fig. 4. Transformation of the [[7,1,3]] encoded circuit in Fig. 2(a) on cluster state. Circles in green, red and white represent qubits measured in the eigenbasis of X, Y, and Z, respectively.

Note that in the process of eliminating redundant qubits, the structure of the underlying cluster state can be inferred by Bob, and this may cause the leakage of what quantum gates were used. To avoid information leakage, Alice only needs to ensure that, the encoding logical qubit $|+_{\theta_i}\rangle_L$ is unknown to Bob during the qubit preparation, while the quantum gates in the encoding circuit can be public. To achieve this goal, one needs to show that the measurement is independent of the polarization angle θ in the qubit preparation. There are three kinds of measurement bases in the process, including (a) the computational basis $\{|0\rangle, |1\rangle\}$, (b) measurement bases $M(0)$, and (c) $M(\pi/2)$, which correspond to the eigenstates of Z, X, Y in Pauli gates, respectively. These measurement bases are independent of the polarization angle θ_i, which implies that the information of θ_i will not be leaked to Bob in the preparation process.

Based on the RBSP with two decoy states, the polarization angle of the prepared qubit θ is ϵ-blind as defined in Section 2.1 which means Bob can not derive any information from it. The prepared quantum state as data qubit will be combined with a series of ancilla qubits to prepare the required quantum error-correcting code $|+_\theta\rangle_L$ according to Protocol 1. Hence, the quantum error-correcting code is also ϵ-blind.

3.2. Fault-tolerant computation of quantum error-correcting codes

From the above quantum error-correcting code preparation, we can get that Bob is unable to obtain any information about θ. Hence, these prepared quantum error-correcting codes are all unknown encoded logical qubits to Bob. Each encoded logical qubit can be considered as a "*logical unit*" to take the place of the original qubit on the brickwork state. This new brickwork state can be used to perform fault-tolerant quantum computation. Based on the original UBQC, we propose a novel fault-tolerant blind quantum computation with quantum error-correcting codes, which is shown in Protocol 2.

To illustrate this protocol, consider a simple example as shown in Fig. 5(a). We use our Protocol 2 to demonstrate the delegated

Protocol 2: Fault-tolerant blind quantum computation with quantum error-correcting codes.

Input: data weak coherent pulses with polarization ρ^{σ}, $\sigma \in_R \{k\pi/4 : 0 \le k \le 7\}$, and ancilla pulses with polarization $|+\rangle$.

Output: measurement result $(s_{x,y})_{1,1}^{nm}$.

1 Alice sends data and ancilla pulses to Bob.
2 **for** $i = 1$ *to* S; **do**
3 Bob uses data pulse ρ^{σ} to prepare the required qubit $|+_{\theta_i}\rangle$, and uses it and a group of ancilla qubits $\{|+\rangle\}$ to prepare the desired encoded logical qubit $|+_{\theta_i}\rangle_L$ based on Procotol.1.
4 **end**
5 Bob uses these encoded logical qubits $\left\{|+_{\theta_i}\rangle_L\right\}_1^S$ to build the encoded brickwork state.
6 **for** $x = 1$ *to* n; $y = 1$ *to* m; **do**
7 Alice calculates angle $\phi'_{x,y}$, the beginning values $s^X_{0,y} = s^Z_{0,y} = 0$; and then calculates measurement angle $\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}, r_{x,y} \in_R \{0,1\}$.
8 Bob measures encoded logical qubit $\left|\psi_{x,y}\right\rangle_L$ in the base $\{|+_{\delta_{x,y}}\rangle_L, |-_{\delta_{x,y}}\rangle_L\}$, return measurement result $s_{x,y} \in \{0,1\}$.
9 **if** $r_{x,y} = 1$ **then** flip $s_{x,y}$;
10 **if** $r_{x,y} = 0$ **then** continue;
11 **end**
12 **return** $s_{x,y}$.

fault-tolerant quantum computation. Firstly, Alice sends weak coherent pulses to Bob, which include data states and ancilla states. The polarization angles of data pulses are selected randomly from $\{k\pi/4 : 0 \le k \le 7\}$, and the ancilla is $|+\rangle$. Secondly, Bob uses the data pulse ρ^{σ} to prepare the required qubit $|+_{\theta_i}\rangle$ based on RBSP with two decoy states [6], and uses it and a group of ancilla qubits $\{|+\rangle\}$ to build cluster state to prepare for the desired encoded logical qubit $|+_{\theta_i}\rangle_L$, as shown in Fig. 4. Bob uses these encoded logical qubits $\left\{|+_{\theta_i}\rangle_L\right\}_1^S$ to build the encoded brickwork state, as shown in Fig. 5(c). Finally, interactive measurements are performed between Alice and Bob. Alice transforms the quantum circuit to a fault-tolerant quantum circuit, as shown in Fig. 5, and then calculates the angle $\phi'_{x,y}$ based on the desired angle $\phi_{x,y}$ and previous measurement outcomes. Let $s^X_{x,y} = \oplus_{i \in D_{x,y}} s_i$ be the parity of all measurement outcomes for qubits in $X_{x,y}$, where $D_{x,y} \subseteq [x-1] \times [m]$ is a set with X-dependencies. Similarly, $s^Z_{x,y} = \oplus_{i \in D'_{x,y}} s_i$ is the parity of all the measurement outcomes for qubits in $Z_{x,y}$, where $D'_{x,y} \subseteq [x-1] \times [m]$ is a set with Z-dependencies. We assume that the dependency sets $X_{x,y}$ and $Z_{x,y}$ are obtained via UBQC [3]. Therefore, the actual measurement angle is $\phi'_{x,y} = (-1)^{s^X_{x,y}} \phi_{x,y} + s^Z_{x,y}\pi$. The measurement angles $\{\delta_{x,y}\}$ can be calculated according to the original UBQC [3], i.e. $\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi r_{x,y}$, where $r_{x,y}$ is randomly chosen in $\{0,1\}$, as shown in Fig. 5(c). Finally, the fault-tolerant measurements on the encoded brickwork state are used to realize fault-tolerant quantum computing. This procedure is repeatedly performed on the encoded brickwork state until the desired quantum computation result is obtained. Not that arbitrary quantum rotation gate and CNOT gate can be used to build a set of universal logical gates [46]. Hence, the fault-tolerant logical gates in the quantum circuit can also be constructed using a series of gates from the set of universal logical gates, which can be used to implement arbitrary fault-tolerant quantum gates, as shown in Fig. 5(d).

In Protocol 2, we note that Alice only uses cluster states to prepare the quantum error-correcting codes according to Protocol 1. In contrast to the original brickwork state, the cluster state can reduce quantum resource consumption. One important note we need to emphasize is that

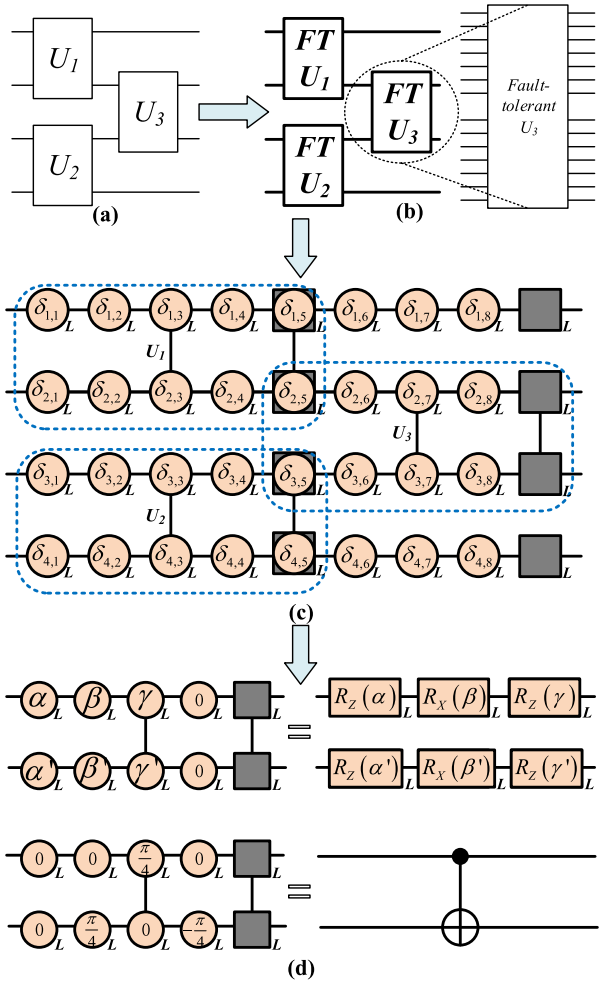

Fig. 5. An example illustration of fault-tolerant blind quantum computation with quantum error-correcting codes. (a) A simple quantum circuit includes three quantum gates U_1, U_2, U_3, and each thin line represents a qubit. (b) In a fault-tolerant quantum circuit, every thick box represents a fault-tolerant quantum gate, and each thick line represents an encoded logical qubit with 7 qubits, i.e. [[7,1,3]] code. (c) Implementation of the fault-tolerant quantum computation on brickwork state, and each angle δ in circles presents a measurement basis $M(\delta)$, each orange thick circle with subscript L represents an encoded logical qubit, each gray thick box with subscript L represents an output logical qubit. (d) Two brick models built with the encoded logical qubits, i.e. arbitrary rotation logical gate and CNOT logical gate, which can be used to form a universal logic gate group.

although the encoded circuit in preparation may be leaked to Bob, the prepared quantum error-correcting codes are unknown to Bob. Hence, these codes can be used as "*logical units*" to build an encoded brickwork state to perform fault-tolerant quantum computation.

3.3. Security analysis

Based on Protocol 2, one can delegate Bob to carry out a fault-tolerant quantum computation. One interesting question is: *If Bob is malicious or dishonest, does Protocol 2 still ensure the privacy of Alice's information?* To answer this question, we present the security proof of Protocol 2 which can ensure that Bob cannot get any computational information except the *scale* of the encoded brickwork state (i.e., say the number of qubits). Note that in this fault-tolerant computing, Bob only has some information about the measurement bases and the

encoded states. Therefore, one needs to show that Alice's information is independent of Bob's information at hand.

Theorem 1. *The fault-tolerant Protocol 2 is ϵ-blind while leaking at most (n, m), where n and m are numbers of rows and columns of the brickwork state respectively, provided that each prepared encoded logical qubit $|+_\theta\rangle_L$ is ϵ-blind.*

Proof. Let (n, m) be the dimension of the brickwork state. According to the definition of brickwork state [3], it ensures that Bob does not get any information on the underlying computation except n and m.

Note that on the one hand, one needs to prove that Bob's measurement bases are ϵ-independent of Alice's computing information. Alice has two secret bit strings in fault-tolerant quantum computing. One is the polarization angle θ of quantum error-correcting code, while the other is Alice's computing information, i.e. the actual measurement base ϕ', which is determined by computational tasks and from previous measurement results. The measurement angle sent to Bob can be calculated according to $\delta = \phi' + \theta + \pi r, r \in_R \{0, 1\}$, and then it is used to perform fault-tolerant measurements on the encoded brickwork state, where r is a random bit string determined by Alice, and it is unknown to Bob. Since each prepared encoded logical qubit is ϵ-blind to Bob, the polarization angle θ is ϵ-uniform in $\{k\pi/4 | k = 0, 1, 2, \ldots, 7\}$. Hence, $\theta + \pi r$ is ϵ-uniform random, and dependent of ϕ', which implies that Bob's measurement angle δ is ϵ-independent of Alice's computing information ϕ'.

On the other hand, one needs to demonstrate the quantum system for Bob is also ϵ-dependent of ϕ'. Since the bit $r_{x,y}$ is random, each encoded logical qubit for Bob will have one of the following two possibilities:
1)If $r_{x,y} = 0$ and $\delta_{x,y} = \phi'_{x,y} + \theta_{x,y}$, then $|\psi_{x,y}\rangle_L = \frac{1}{\sqrt{2}}\left(|0\rangle_L + e^{i\left(\delta_{x,y} - \phi'_{x,y}\right)}|1\rangle_L\right)$.
2)If $r_{x,y} = 1$ and $\delta_{x,y} = \phi'_{x,y} + \theta_{x,y} + \pi$, then $|\psi_{x,y}\rangle_L = \frac{1}{\sqrt{2}}\left(|0\rangle_L - e^{i\left(\delta_{x,y} - \phi'_{x,y}\right)}|1\rangle_L\right)$.

Without loss of generality, if the angle δ is fixed, ϵ-blind θ depends on ϕ', but the random bit $r_{x,y}$ is unknown to Bob, so the quantum system for Bob is a ϵ-blind mixed state of two cases, which implies it is ϵ-independent of Alice's computing information ϕ'. □

In summary, using the Protocol 2, Bob first can be delegated to prepare the ϵ-blind quantum error-correcting codes, and then utilize these codes to perform ϵ-blind fault-tolerant quantum computation. In addition, Protocol 2 only requires Alice to send weak coherent pulses, which can decouple Alice's dependency on quantum computing and quantum memory. The number of ancilla qubits increases only by a constant factor with the increasing computational scale, rather than a linear increase in the original UBQC [3]. Hence, Protocol 2 has the advantage of reducing Alice's quantum resource consumption and removing Alice's quantum computing requirements.

4. Concatenation codes and resource requirements

Here, we propose to utilize a concatenated code that further reduces the qubit error rate in quantum computation. The idea of our concatenated code is to recursively execute the above encoding circuit to reconstruct a multi-level quantum circuit. When the number of levels is 1, it represents the initial encoded circuit. In the construction, each qubit in the original circuit is encoded in a quantum code whose own qubits are encoded again, and so on, as illustrated in Fig. 6. Hence, in order to improve the error correction performance of qubits, we propose using concatenation codes in UBQC, as shown in Protocol 3.

To illustrate our idea, let us consider two-level concatenated [[7,1,3]] codes to reduce the error rate, and each [[7,1,3]] code encodes a single qubit using a block of 7 qubits. When examining one of the 7 qubits in this block at a higher level, we note that it is an encoded sub-block. With this method, the growth of complexity in quantum error correction is not as sharp as the increase in the error-correcting capacity

Protocol 3: Fault-tolerant blind quantum computation with concatenation codes.

Input: Data pulses and ancilla pulses
Output: Measurement results $(s_{x,y})_{1,1}^{nm}$

for $i = 1$ *to* S; **do**
  Bob uses data pulse ρ^σ to prepare the qubit $|+_{\theta_{i,k}}\rangle$ based on RBSP with two decoy states, the initial value $k = 0$.
  while $k =< l - 1$; **do**
    Bob uses $|+_{\theta_{i,k}}\rangle_L$ and a group of ancilla qubits $\{|+\rangle\}$ to build cluster state, and then prepare the desired encoded logical qubit $|+_{\theta_{i,k+1}}\rangle_L$ based on the lth-level concatenated circuit;
    $k++$.
  end
  Output$(|+_{\theta_{i,l}}\rangle_L)$
end
Bob uses these encoded logical qubits $\left\{|+_{\theta_{i,l}}\rangle_L\right\}_1^S$ to build the encoded brickwork state.
The interactive measurement process refers to Protocol 2 to perform.
return $s_{x,y}$.

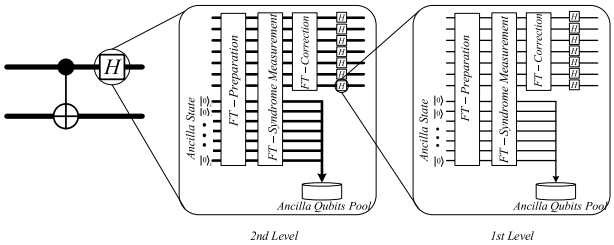

Fig. 6. The two-level concatenated [[7,1,3]] codes. When inspected at higher resolution, each qubit or quantum gate in the block is itself an encoded sub-block. Each block comprises fault-tolerant syndrome measurement, fault-tolerant correction, and fault-tolerant quantum gate(a transversal application of quantum gate).

of quantum code. Note that the [[7,1,3]] code can correct one error, so a correction will fail if more than two errors occur in the block. If the probability of error per actual physical qubit at the lowest level is e_0, and the correction is fault-tolerant, then the probability of the correction failure in [[7,1,3]] code is of order e_0^2. If we concatenate the code to construct a new block, then an error will occur in the block if two of the sub-blocks of size 7 fail, which occurs with a probability of order e_0^4.

Let the error probability of each block at level i be denoted as e_i. At each level of the concatenated code, the block fails if there are errors in at least two contained sub-blocks. For the i-level concatenated code, the failure probability can be estimated according to Eq. (5).

$$\begin{aligned} e_i &= \sum_{k\geq 2}^{7} \binom{7}{k} e_{i-1}^k \\ &\approx \binom{7}{2} e_{i-1}^2 + o\left(e_{i-1}^2\right) \\ &\approx \frac{(21e_0)^{2^i}}{21} + o\left(e_0^{2^i}\right). \end{aligned} \tag{5}$$

In order to significantly reduce the probability of error, the condition $e_0 < 1/21$ must be satisfied in the concatenation codes (the concatenation codes hold if and only if $(21e_0)^{2^i} \ll 1$). Further, we can also make the error rate arbitrarily small by adding sufficient levels of concatenation. If the concatenated [[7,1,3]] code circuit is delegated to Bob to prepare

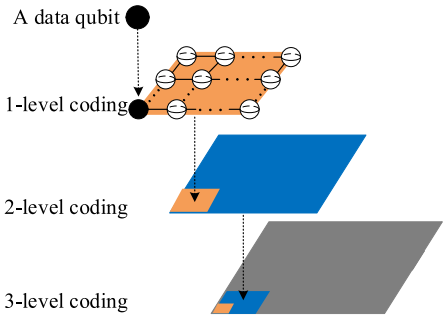

Fig. 7. The encoding structure for a date qubit in 3-level concatenation code.

quantum error-correcting code on the cluster state in Protocol 2, then the number of required ancilla pulses will increase with the number of levels. Since a data qubit is used to prepare an encoded logical qubit using [[7,1,3]] encoding circuit, the number of required data qubits is consistent with that of the non-coding case, which remains unchanged.

Note that in Protocol 3, weak coherent pulses are sent to Bob, which contain data pulses and ancilla pulses. According to RBSP with two decoy states, data pulses are used to prepare the required data qubits $\{|+_{\theta_i}\rangle\}_1^S$, and each data qubit is encoded into a required logical qubit. Hence, the number of data pulses is equal to RBSP with two decoy states, which is denoted as N^d [7]. The number of ancilla pulses depends on the encoding circuit on cluster state, as shown in Fig. 4. Note that the required number of ancilla qubits is a constant when preparing an encoded logical qubit, and we denoted it as C. If the transmittance of the quantum channel is T, and the computational scale is S, then the number of ancilla pulses satisfies $N^a = CS/T$. Hence, the total number of required pulses in Protocol 3 is the sum of data and ancilla pulses, i.e.N=$N^d + N^a$.

The ancilla pulses are used for preparations, syndrome measurements, and error corrections. Note that, the required ancilla qubits in the syndrome measurements and corrections can be reused in our fault-tolerant protocol, so this part of the resource consumption of ancilla qubits can be ignored, and only the number of ancilla qubits in the preparation is needed for analysis. In Protocol 3, we use cluster state to prepare the encoded logical qubits. Based on the circuit model, the block size of ancilla qubits to prepare an encoded logical qubit is a constant C. According to the encoding structure for data qubits in Fig. 7, the block size of ancilla qubits is the sum of the required number at all levels. If we use an n-level concatenation code to prepare an encoded logical qubit, the number of ancilla qubits is denoted as N_n^a, and its value can be calculated as follows:

$$N_n^a = \sum_{i=1}^{n} 7^{i-1} C = \frac{(7^n - 1)C}{6}. \tag{6}$$

The number of required data pulses is consistent with RBSP with two decoy states. According to Eq. (2), we have $N^d \geq N^{L,v_1,v_2}$. The required total number of pulses in the n-level concatenated [[7,1,3]] code, i.e. N_n, is estimated as follows:

$$\begin{aligned} N_n &= N^d + N_n^a \\ &\geq N^{L,v_1,v_2} + \frac{(7^n - 1)CS}{6T} \\ &\approx \frac{S}{T}\left[\frac{\ln(\epsilon/S)}{p_\mu \mu \ln\left(1 - p_1^{L,v_1,v_2}\right)} + \frac{(7^n - 1)C}{6}\right], \end{aligned} \tag{7}$$

where S is the computational scale, ϵ is the security parameter of the blind quantum computation, T is the transmittance of the quantum channel between Alice and Bob, μ,v_1, v_2 are the average number of signal photons and two decoy photons, respectively, the proportion of single photon in the signal states is described as p_1, and the lower bound is p_1^{L,v_1,v_2}, the probability of signal pulses chosen by the client is defined as p_μ.

In the RBSP protocol with two decoy states, for the computation size S, if an error occurs in the prepared qubits, the whole preparation protocol will fail. We assume that the preparation protocol is a repeatable Bernoulli experiment. If the error probability of each prepared qubit is e_0, the probability of successful qubit preparation in each experiment will be $(1 - e_0)^S$. After repeating k times, the probability of successful qubit preparation will be $1 - (1 - (1 - e_0)^S)^k$. In the n-level concatenated code, the probability of successful qubit preparation is $(1 - e_n)^S$. If the probability of successful qubit preparation is the same in both coding and non-coding cases, i.e. $1 - (1 - (1 - e_0)^S)^k = (1 - e_n)^S$, then the repetition number k can be calculated as follows:

$$k = \ln[1 - (1 - e_n)^S] / \ln[1 - (1 - e_0)^S], \tag{8}$$

where e_n can be estimated according to Eq. (5). In other words, if Alice wants to use the non-coding protocol to achieve the same probability as encoding, the number of required data pulses is k times that of the previous case, i.e. kN^d. Let R describe the ratio of the number of pulses for concatenation codes to that of the non-coding case at the same probability. The ratio R is a resource consumption function with the number of levels n, which can be used to estimate the optimal level of concatenated codes. The resource consumption ratio $R(n)$ is shown as follows:

$$\begin{aligned} R(n) &= \frac{N_n}{kN^d} = \frac{N^d + N_n^a}{kN^d} \leq \frac{N^{L,v_1,v_2} + N_n^a}{kN^{L,v_1,v_2}} \\ &= \frac{\ln\left[1 - \left(1 - e_0\right)^S\right]}{\ln\left[1 - \left(1 - e_n\right)^S\right]}\left(\frac{(7^n - 1)CS}{6TN^{L,v_1,v_2}} + 1\right), \end{aligned} \tag{9}$$

where N^{L,v_1,v_2} is the lower bound of the number of required pulses in the RBSP with two decoy states, which can be estimated according to Eq. (2). When the computation scale S is fixed, we can obtain the ratio $R(n) \sim (7/2)^n$ according to Eq. (5) and Eq. (9). Hence, the ratio R has an exponential growth trend for a large number of levels n. For a high-level concatenation code, a large number of qubits are used for repeated encoding. In our concatenation code, we not only need to consider improving the probability of successful qubit preparation but also reducing resource consumption. Hence, we only take into account the optimal level in the low-level concatenated code ($0 \leq n \leq 4$). When the resource consumption ratio $R(n)$ reaches its minimum, it indicates that the utilization efficiency of pulses is at its maximum compared to the non-coding case, which implies that the performance of concatenated code is optimal at this point.

In summary, according to Eq. (7), note that the number of required data pulses is unchanged with the increasing number of levels, and the number of required ancilla pulses grows exponentially with the increasing number of levels in the concatenation code. Combined with Eq. (9), we can estimate the resource consumption ratio $R(n)$, and further determine the optimal level n.

5. Performance evaluation

The original UBQC protocol introduced measurement-based quantum computation (MBQC) on brickwork states, enabling clients with limited quantum resources to delegate computations. However, it lacks explicit error correction, making it vulnerable to noise in realistic settings. Hence, UBQC protocols face significant challenges in balancing fault tolerance, resource overhead, and client's simplicity. In the section, we use tables instead of diagrams to evaluate and compare the relevant schemes. The performance trade-offs of different quantum error-correcting schemes within FT-BQC protocols are compared in Table 1.

Our protocol demonstrates that the [[7,1,3]] concatenated code achieves an optimal balance between practicality and fault tolerance for noisy intermediate-scale UBQC. In Table 1, note that our protocol enables remote preparation of logical qubits with minimal

Table 1
Comparison of Fault-Tolerant BQC Protocols.

Protocol	Method	Client's Requirements	Ancilla Overhead	Threshold	Feasibility
Original UBQC [3]	Toric code	SPP	Linear in S	High (≈1%–10%)	Low
Topological BQC [47]	RHG codes	SPP	Constant in S	High (≈0.43%)	Low
Chien's FT- UBQC [48]	[[7,1,3]] codes	LQP	Constant in S	Moderate (≈0.01%)	Low
Zhao's FT-UBQC [7]	[7,1,3] codes	WCP	Constant in S	Moderate (≈0.01%)	Low
Kapourniotis's BQC [49]	Repetition code	SPP	Constant in S	High (≈10%)	High
Our Protocol	2-level [[7,1,3]]	WCP	Constant in S	High (≈1%)	High

SPP: Single-Photon Preparation; **LQP:** Logical Qubit Preparation; **WCP:** Weak Coherent Pulses; **S:** Computation Size.

Table 2
Simulation parameters for our Protocol.

α	t_S	η_S	μ	v_1	v_2	p_μ	p_{v_1}	p_{v_2}	S	ϵ
0.2	0.45	0.1	0.6	0.125	0	0.9	0.05	0.05	1000	10^{10}

client-side overhead–requiring only weak coherent pulses–while suppressing error propagation through its hierarchical structure. In contrast, existing fault-tolerant protocol face inherent limitations within UBQC frameworks:

(1) Broadbent et al.'s [3] original UBQC[3] required a large number of swap gates for quantum error correction coding and fault-tolerant computation.
(2) Topological BQC [47] schemes leveraging 3D RHG lattice codes face impractical preparation demands, as their non-planar connectivity conflicts with MBQC's brickwork state geometry.
(3) Chien et al.'s [48] FT-UBQC imposes prohibitive client-side requirements, including logical qubit preparation and quantum memory, which violate UBQC's core design principle of client simplicity.
(4) Zhao et al.'s [7] FT-UBQC required many CNOT gates for fault-tolerant quantum error correcting coding and computation.
(5) Kapourniotis et al.'s [49] verification-based BQC lacks robustness against high-weight errors due to its reliance on post hoc statistical checks rather than embedded error correction.

In the section, we present the performance and simulation results of our protocols to demonstrate the advantages of quantum error-correcting codes in fault-tolerant blind quantum computation.

Simulation settings: Our computer uses Matlab with an Intel(R) Core(TM) i7-6700HQ CPU with 12.0GB RAM and Win 10 pro-OS to carry out these simulations. The overall transmittance T (including fiber transmittance and detection efficiency) is calculated as follows:

$$T = t_s \cdot \eta_s \cdot 10^{-\alpha L/10} \tag{10}$$

Table 2 (please refer to the data in [50] and [40]) summarizes our simulation parameters, where the parameter α is the loss coefficient measured in dB/km, L is the length of the fiber in km, t_s is denoted as the internal transmittance of optical components on Bob's side, and η_s is the detector efficiency on Bob's side. The mean photon number of signal states and two decoy states are represented as μ, v_1, v_2, respectively. The chosen probability of signal states and two decoy states are denoted as p_μ, p_{v_1}, p_{v_2}, respectively. The computation scale is S. The required blindness of the UBQC is denoted as ϵ. Let the error probability of each qubit is $e_0 = 0.01$. Based on 4, we calculate the number of ancilla qubits, $C = 1774$.

We assume that Alice has a laser transmitter with frequency $f = 1MHZ$, and Bob has a full-fledged quantum computer. According to the setting in Table 2, one can derive the preparation efficiency of our protocols, that is the number of qubits generated per second. Combined with Eq. (7), the upper bound of the efficiency in the concatenation code is estimated as.

$$\begin{aligned} E &= S \cdot f / N_n \\ &\leq S \cdot f / \left(N^{L,v_1,v_2} + N_n^a\right) \end{aligned} \tag{11}$$

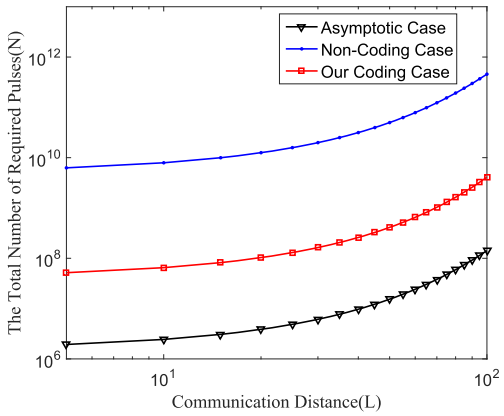

Fig. 8. The total number of required pulses N with the same probability of successful qubit preparation vs. the communication distance between Alice and Bob.

A. simulation 1: Performance evaluation with [[7,1,3]] codes

Purpose of simulation 1: In Protocol 1, we utilize the cluster state to prepare quantum error-correcting codes for correcting error qubits. In order to show the advantages of our approach, we perform simulations to evaluate the quantum resource consumption and preparation efficiency.

Discussions of simulation 1: Fig. 8 shows the relationship between the total number of required pulses and the communication distance under the condition of having the same probability of successful qubit preparation. The red line represents the simulation results of our Protocol 1 with our coding case, and the green line represents the non-coding case. The black line represents the simulation results in the asymptotic case which is the infinite data size and with near-perfect qubits preparation [6,50].

From Fig. 8, one can see that the total number of required pulses for the coding case in Protocol 1 is less than that of the non-coding case in RBSP with two decoy states under the condition of having the same probability of successful qubits preparation. Since the initial error probability of each qubit is e_0, the probability of successful qubit preparation with size S is $(1-e_0)^S$. In quantum error-correcting code preparation, the error probability of each encoded logical qubit is e_0^2, and the probability of successful encoded qubit preparation is $(1-e_0^2)^S$. In order to obtain the same probability of successful qubit preparation, the non-coding case needs to be repeated k times. As the communication distance increases, the value of N grows, which implies that the channel loss and qubit error rate have a significant influence on N. For long-distance communication, the advantages of our Protocol 1 are more apparent than the non-coding case, which means that the number of required pulses N is closer to the asymptotic case.

Fig. 9 illustrates the correlation between the preparation efficiency E and the communication distance L under the condition of having the same probability of successful qubit preparation. The red line in the figure represents the efficiency curves of our Protocol 1 in the coding case, while the green and black lines represent the non-coding case and the

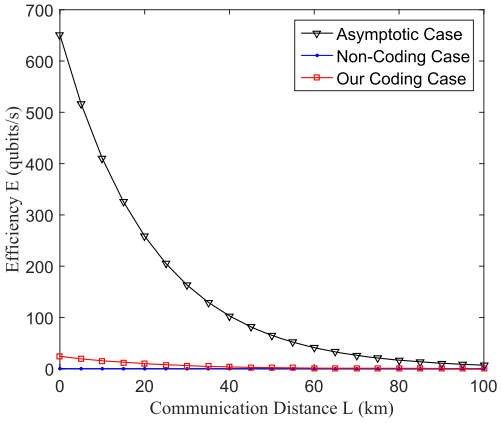

Fig. 9. The preparation efficiency E with the same probability of successful qubit preparation vs. the communication distance L.

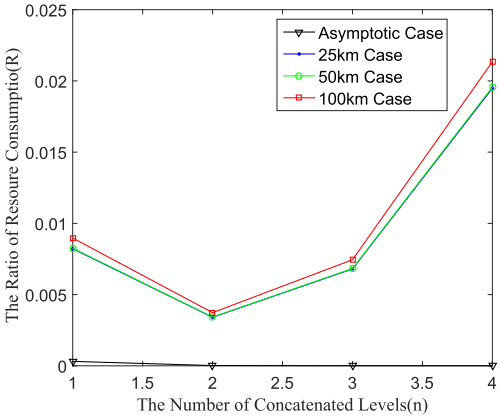

Fig. 10. The ratio of resource consumption ($R(n)$) vs. the number of levels (n) in the concatenation code.

asymptotic case (infinite data size and with near-perfect qubits preparation), respectively.

From Fig. 9, one can observe that the efficiency of preparing qubits gradually decreases with increasing communication distance. Compared with the non-coding case, the preparation efficiency of qubits for the coding case in our Protocol 1 is closer to the asymptotic limit under the same condition. As the error rate decreases from e_0 to e_0^2, more pulses are used as ancilla qubits to prepare quantum error-correcting codes, which will lead to a drop in efficiency.

Summary of simulation 1: We show that the performance with [[7,1,3]] code is better than the non-coding case in terms of both quantum resources and preparation efficiency. Hence, our Protocol 1 is practical for preparing the required qubits.

B. simulation 2: Discussions on concatenation levels

Purpose of simulation 2: In Protocol 2, we employ the quantum error-correcting codes as "*logical units*" to facilitate fault-tolerant blind quantum computation. However, the propagation and accumulation of logical qubit errors may result in a relatively high fault-tolerant threshold, Therefore, we propose to use the concatenated [[7,1,3]] code to reduce the error rate of encoded logical qubits, thereby improving the fault-tolerant threshold. The naive implementation of concatenated codes will result in a large number of ancilla qubits. To enhance fault tolerance with limited quantum resources, we perform simulation 2 to identify the optimal level of concatenation.

Discussions of simulation 2: Fig. 10 illustrates the relationship between the ratio of resource consumption ($R(n)$) and the number of levels (n) in the concatenation code. The blue, green, and red lines represent the simulation results of the resource consumption ratio for communication distances of 25km, 50km, and 100km, respectively. The black line corresponds to the simulation results of the asymptotic case, which is the infinite data size and with near-perfect qubits preparation.

According to Eq. (9), the resource consumption ratio $R(n)$ is the proportion of the number of pulses for concatenated [[7,1,3]] codes in that of the non-coding case. In order to identify the optimal level, one can set the partial derivative $\partial R(n)/\partial n = 0$ to solve for the extreme value. Nevertheless, the partial derivative is too complicated that it is difficult to solve its analytical solution. Instead, we use simulation to estimate the optimal level n, as it is shown in Fig. 10. Observe that when the optimal number of levels n is about 2, the resource consumption ratio $R(n)$ reaches the minimum value. This implies the 2-level concatenated code only needs relatively fewer quantum resources than other levels in meeting the requirements of qubit error rate. As a result, the 2-level concatenated [[7,1,3]] code enables quantum computation with limited resources to achieve better error-correcting performance.

Summary of simulation 2: When the concatenated code is used in Protocol 2, we discuss the effect of the concatenation levels on the quantum resource consumption under the condition of having the same error rate of qubits. By analyzing the ratio of resource consumption $R(n)$, one can determine that the optimal concatenation level is n = 2, which can minimize resource consumption while still maintaining a high level of fault tolerance in UBQC.

C. simulation 3: Comparison with concatenated [[7,1,3]] codes

Purpose of simulation 3: In Protocol 3, we propose to use the concatenated codes to perform fault-tolerant blind quantum computation. To further evaluate the performance of concatenated [[7,1,3]] code, we perform simulation 3 to compare the quantum resource consumption and preparation efficiency of the concatenation codes at different levels.

Discussions of simulation 3: Fig. 11 depicts the relationship between the total number of required pulses N and the communication distance (L) under the condition of having the same probability of successful qubit preparation. The green, red, yellow, cyan, and blue lines represent the simulation results of the 1,2,3,4-level concatenation code and non-encoding case, respectively. The black line represents the simulation results in the asymptotic case which is the infinite data size and with near-perfect qubits preparation.

From Fig. 11, the number of required pulses N in each case is an increasing trend with the communication distance L, which indicates that both the channel loss and qubit error rate have a significant influence on N. Compared with the non-coding case under the same condition, the number of required pulses N for encoding is closer to the asymptotic case. The reason is that the error probability of the encoded logical qubits prepared by the concatenated circuit is much lower than that of the non-coding case. In order to obtain the same error probability of prepared qubits, the preparation protocol for the non-coding case needs to be repeated k times, which results in wasting a vast number of pulses. Further, one can see that the number of required pulses in the 2-level concatenation code is less than other levels at the same probability of successful qubit preparation, which shows the 2-level concatenation code can provide superior error-correcting performance using limited quantum resources.

Fig. 12 and 13 depict the relationship between the preparation efficiency E and the communication distance L, while maintaining the same probability of successful qubit preparation. The green, red, yellow, cyan, and blue lines represent the simulation results of the 1,2,3,4-level concatenation codes and the non-coding case, respectively. The black line corresponds to the simulation results in the asymptotic case which is the infinite data size and with near-perfect qubits preparation.

In Fig. 12, the preparation efficiency E in the asymptotic case is much higher than that of the actual case. Since the fluctuation of finite data and imperfect preparation of required qubits are inevitable in real UBQC, a large number of pulses are used as ancilla qubits to deal with

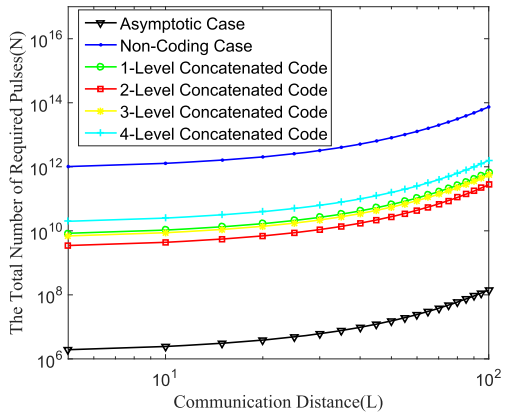

Fig. 11. The total number of required pulses N with the same probability of successful qubit preparation vs. the communication distance (L).

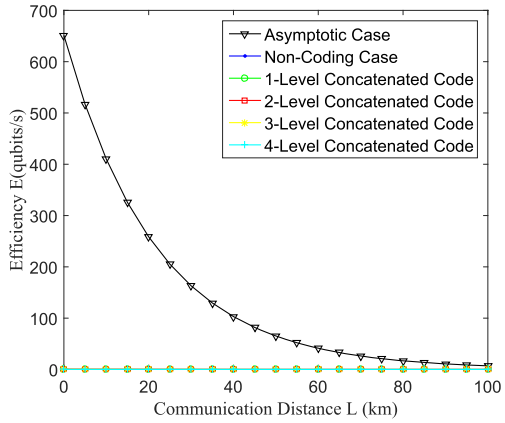

Fig. 12. The preparation efficiency E with the same probability of successful qubit preparation vs. the communication distance L.

the impacts of fluctuation and qubit errors, which leads to a reduction in efficiency. In order to better show our simulation results, we give a partial enlargement of Fig. 12, as shown in Fig. 13. Not that the preparation efficiency E has a sharp decline trend as the communication distance increases. Under the condition of having the same probability of successful qubit preparation, we demonstrate the preparation efficiency of the non-coding case and the concatenation codes with different levels. Although the high-level concatenation code can obtain a lower error rate, it requires massive quantum resource consumption, which results in lower efficiency. Hence, it is not applicable in reality. In UBQC with limited quantum resources, the 2-level concatenation code is capable of delivering a better preparation efficiency in Fig. 13.

Summary of simulation 3: Our Protocol 3 demonstrates that the 2-level concatenation code can be utilized to improve both the preparation efficiency and quantum resource utilization. In addition, Protocol 3 yields significant improvement in the error-correcting performance for fault-tolerant blind quantum computation. These findings highlight the potential of the 2-level concatenation code as an effective method for enhancing the performance and efficiency of UBQC.

6. Conclusions

In the paper, we first propose a protocol to realize quantum error-correcting codes on cluster state. A fault-tolerant blind quantum computation with quantum error-correcting codes is proposed to address errors in quantum computation. To improve the error-correcting performance, the stabilizer codes are used for multi-level concatenating to improve

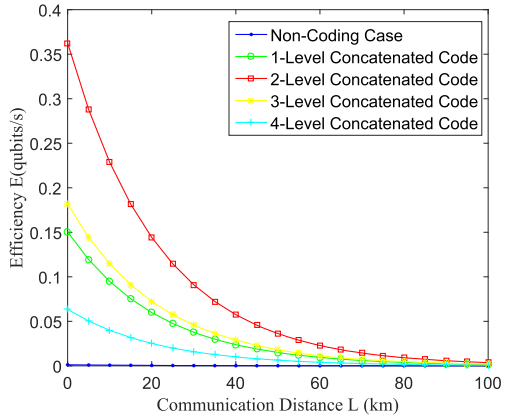

Fig. 13. The preparation efficiency E with the same probability of successful qubit preparation vs. the communication distance L. It is the partial enlargement of Fig. 12.

the fault-tolerant threshold of qubits. To reduce quantum resource consumption, we also analyze the number of required pulses in each level of concatenation code. Since a large number of qubits are required for repetitive encoding in the high-level concatenation codes, the low-level concatenated codes ($0 \leq n \leq 4$) are only considered in this paper.

We show that the optimal level in the concatenation codes is about 2, and the resource consumption ratio reaches the minimum value, which means that the number of required pulses in the 2-level concatenation codes is less than other levels under the same probability of successful qubit preparation. From the perspective of reducing quantum resources, our simulation results show that the 2-level concatenated code can achieve a better performance than other levels with the increase of communication distance.

To further improve the practical performance of UBQC, our future works include designing high-dimensional q-ary stabilizer codes, entanglement-assisted quantum error-correcting codes, and other codes under different propagation rules. The question regarding these codes points to a conceptually distinct and potentially fruitful direction for extending fault-tolerant UBQC. In summary, exploring this synergy between correction and blindness constitutes a compelling and challenging avenue for future work.

CRediT authorship contribution statement

Qiang Zhao: Writing – review & editing, Writing – original draft, Formal analysis, Data curation, Conceptualization; **John C.S. Lui:** Writing – review & editing, Supervision, Methodology.

Data availability

Data will be made available on request.

Declaration of competing interest

Dear Editor,

This is a manuscript by Qiang Zhao, John C.S. Lui entitled “Applying Quantum Error-Correcting Codes for Fault-Tolerant Blind Quantum cloud Computation”. It is submitted to be considered for publication as an “Original Research Article” in “Quantum error correction codes” and “Quantum computing ”. Three versions of it has been uploaded to arXiv (https://arxiv.org/abs/2301.01960). But neither the entire paper nor any part of its content has been published or accepted anywhere.

All authors have approved the manuscript and agreed with its submission to Future Generation Computer System for Special Collection on Advances in Quantum Computing: Methods, Algorithms, and Systems Vol II. The authors declare that they have no competing interests.

This work fits the scope “Quantum error correction code and quantum computing” of Special Collection on Advances in Quantum Computing. We believe the work may be of particular interest to the readers of Future Generation Computer System.

Best regards,
Sincerely yours,
Qiang Zhao

References

[1] A.M. Childs, Secure assisted quantum computation, Quantum Info. Comput. 5 (6) (2005) 456–466.
[2] P. Arrighi, L. Salvail, Blind quantum computation, Int. J. Quant. Inf. 4 (05) (2006) 883–898.
[3] A. Broadbent, J. Fitzsimons, E. Kashefi, Universal blind quantum computation, in: Foundations of Computer Science, 2009. FOCS'09. 50Th Annual IEEE Symposium on, IEEE, 2009, pp. 517–526.
[4] V. Dunjko, E. Kashefi, A. Leverrier, Blind quantum computing with weak coherent pulses, Phys. Rev. Lett. 108 (20) (2012).
[5] Q. Zhao, Q. Li, Blind Quantum Computation with Two Decoy States, Springer International Publishing, 2017.
[6] Q. Zhao, Q. Li, Finite-data-size study on practical universal blind quantum computation, Quantum Inf. Process. 17 (7) (2018) 171.
[7] Q. Zhao, Q. Li, H. Mao, X. Wen, Q. Han, M. Li, Fault-tolerant quantum error correction code preparation in UBQC, Quantum Inf. Process. 19 (8) (2020) 236.
[8] A. Cojocaru, L. Colisson, E. Kashefi, P. Wallden, On the possibility of classical client blind quantum computing, Cryptography 5 (1) (2021) 3.
[9] R.-T. Shan, X. Chen, K.-G. Yuan, Multi-party blind quantum computation protocol with mutual authentication in network, Sci. China Inf. Sci. 64 (6) (2021) 1–14.
[10] B. Polacchi, D. Leichtle, L. Limongi, G. Carvacho, G. Milani, N. Spagnolo, M. Kaplan, F. Sciarrino, E. Kashefi, Multi-client distributed blind quantum computation with the qline architecture, Nat. Commun. 14 (1) (2023) 7743.
[11] B. Polacchi, D. Leichtle, G. Carvacho, G. Milani, N. Spagnolo, M. Kaplan, E. Kashefi, F. Sciarrino, Experimental verifiable multiclient blind quantum computing on a qline architecture, Phys. Rev. Lett. 134 (20) (2025) 200603.
[12] P. Drmota, D.P. Nadlinger, D. Main, B.C. Nichol, E.M. Ainley, D. Leichtle, A. Mantri, E. Kashefi, R. Srinivas, G. Araneda, et al., Verifiable blind quantum computing with trapped ions and single photons, Phys. Rev. Lett. 132 (15) (2024) 150604.
[13] S. Ma, C. Zhu, X. Liu, H. Li, S. Li, Universal blind quantum computation with improved brickwork states, Phys. Rev. A 109 (1) (2024) 012606.
[14] Y. Lee, D. Chung, Partial blind quantum computation, arXiv:2503.10007 (2025).
[15] Z. Cai, R. Babbush, S.C. Benjamin, S. Endo, W.J. Huggins, Y. Li, J.R. McClean, T.E. O'Brien, Quantum error mitigation, Rev. Mod. Phys. 95 (4) (2023) 045005.
[16] H. Jnane, J. Steinberg, Z. Cai, H.C. Nguyen, B. Koczor, Quantum error mitigated classical shadows, PRX Quantum. 5 (1) (2024) 010324.
[17] W.-J. Liu, Z.-X. Li, W.-B. Li, Q. Yang, Public verifiable measurement-only blind quantum computation based on entanglement witnesses, Quantum Inf. Process. 22 (3) (2023) 137.
[18] Q. Zhao, H. Mao, Y. Qiao, A.A.A. El-Latif, Q. Li, A remote quantum error-Correcting code preparation protocol on cluster states, Mathematics 11 (14) (2023) 3035.
[19] D. Bhatnagar, M. Steinberg, D. Elkouss, C.G. Almudever, S. Feld, Low-depth flag-style syndrome extraction for small quantum error-correction codes, in: 2023IEEE International Conference on Quantum Computing and Engineering (QCE), 1, IEEE, 2023, pp. 63–69.
[20] I. Dinur, M.-H. Hsieh, T.-C. Lin, T. Vidick, Good quantum LDPC codes with linear time decoders, in: Proceedings of the 55Th Annual ACM Symposium on Theory of Computing, 2023, pp. 905–918.
[21] Y. Wang, S. Simsek, T.M. Gatterman, J.A. Gerber, K. Gilmore, D. Gresh, N. Hewitt, C.V. Horst, M. Matheny, T. Mengle, et al., Fault-tolerant one-bit addition with the smallest interesting color code, Sci. Adv. 10 (29) (2024) eado9024.
[22] G.Q. AI, et al., Quantum error correction below the surface code threshold, Nature 638 (8052) (2024) 920.
[23] G. Baranes, I.W. Wang, F. Machado, A. Suleymanzade, P.-J. Stas, Y.-C. Wei, S.F. Yelin, J. Borregaard, M.D. Lukin, Designing fault-Tolerant blind quantum computation, arXiv:2505.21621 (2025).
[24] D. Ruiz, J. Guillaud, A. Leverrier, M. Mirrahimi, C. Vuillot, LDPC-Cat codes for low-overhead quantum computing in 2D, Nat. Commun. 16 (1) (2025) 1040.
[25] B. Pato, T. Tansuwannont, K.R. Brown, Concatenated steane code with single-flag syndrome checks, Phys. Rev. A 110 (3) (2024) 032411.
[26] T. Morimae, K. Fujii, Blind topological measurement-based quantum computation, Nat. Commun. 3 (1) (2012) 1036.
[27] Z. Qu, K. Wang, M. Zheng, Secure quantum fog computing model based on blind quantum computation, J. Ambient. Intell. Humaniz. Comput. 13 (8) (2022) 3807–3817.
[28] J. Guillaud, M. Mirrahimi, Error rates and resource overheads of repetition cat qubits, Phys. Rev. A 103 (4) (2021) 042413.
[29] C. Chamberland, K. Noh, P. Arrangoiz-Arriola, E.T. Campbell, C.T. Hann, J. Iverson, H. Putterman, T.C. Bohdanowicz, S.T. Flammia, A. Keller, Building a fault-tolerant quantum computer using concatenated cat codes, PRX Quantum. 3 (1) (2022) 010329.
[30] C. Chamberland, T. Jochym-O'Connor, R. Laflamme, Overhead analysis of universal concatenated quantum codes, Phys. Rev. A 95 (2) (2017) 022313.
[31] P. Webster, M. Vasmer, T.R. Scruby, S.D. Bartlett, Universal fault-tolerant quantum computing with stabilizer codes, Phys. Rev. Res. 4 (1) (2022) 013092.
[32] Y. Ouyang, Avoiding coherent errors with rotated concatenated stabilizer codes, npj Quantum. Inf. 7 (1) (2021) 1–7.
[33] G. Luo, M.F. Ezerman, M. Grassl, S. Ling, Constructing quantum error-correcting codes that require a variable amount of entanglement, Quantum Inf. Process. 23 (1) (2023) 4.
[34] M. Grassl, F. Huber, A. Winter, Entropic proofs of singleton bounds for quantum error-correcting codes, IEEE Trans. Inf. Theory 68 (6) (2022) 3942–3950.
[35] Y. Li, S. Zhu, Linear codes of larger lengths with galois hulls of arbitrary dimensions and related entanglement-assisted quantum error-correcting codes, Discrete Math. 347 (2) (2024) 113760.
[36] S. Krinner, N. Lacroix, A. Remm, A. Di Paolo, E. Genois, C. Leroux, C. Hellings, S. Lazar, F. Swiadek, J. Herrmann, Realizing repeated quantum error correction in a distance-three surface code, Nature 605 (7911) (2022) 669–674.
[37] J. Zhang, Y.-C. Wu, G.-P. Guo, The concatenation of the gottesman-Kitaev-Preskill code with the XZZX surface code, arXiv:2207.04383 (2022).
[38] A. Broadbent, J. Fitzsimons, E. Kashefi, Measurement-based and universal blind quantum computation, in: International School on Formal Methods for the Design of Computer, Communication and Software Systems, Springer, 2010, pp. 43–86.
[39] Y.-F. Jiang, K. Wei, L. Huang, K. Xu, Q.-C. Sun, Y.-Z. Zhang, W. Zhang, H. Li, L. You, Z. Wang, H.-K. Lo, F. Xu, Q. Zhang, J.-W. Pan, Remote blind state preparation with weak coherent pulses in the field, Phys. Rev. Lett. 123 (10) (2019) 100503.
[40] K. Xu, H.-k. Lo, Blind quantum computing with decoy states, arXiv:1508.07910 (2015).
[41] J. Preskill, Fault-tolerant quantum computation, in: Introduction to Quantum Computation and Information, World Scientific, 1998, pp. 213–269.
[42] M.A. Nielsen, I. Chuang, Quantum computation and quantum information, 2002.
[43] D. Gottesman, Stabilizer codes and quantum error correction, arxiv:9705052 (1997).
[44] A. Rodriguez-Blanco, H.N. Nguyen, K.B. Whaley, Fault-tolerant correction-ready encoding of the [[7, 1, 3]] steane code on a 2D grid, arXiv:2504.01083 (2025).
[45] R. Raussendorf, D.E. Browne, H.J. Briegel, Measurement-based quantum computation on cluster states, Phys. Rev. A 68 (2) (2003).
[46] D.E. Deutsch, A. Barenco, A. Ekert, Universality in quantum computation, Proc. R. Soc. Lond. Ser. A: Math. Phys. Sci. 449 (1937) (1995) 669–677.
[47] T. Morimae, K. Fujii, Blind topological measurement-based quantum computation, Nat. Commun. 3 (2012) 1036.
[48] C.-H. Chien, R. Van Meter, S.-Y. Kuo, Fault-tolerant operations for universal blind quantum computation, ACM J. Emerg.Technol. Comput. Sys. 12 (2015) 9.
[49] T. Kapourniotis, E. Kashefi, D. Leichtle, L. Music, H. Ollivier, Unifying quantum verification and error-detection: theory and tools for optimisations, Quantum Sci. Technol. 9 (3) (2024) 035036.
[50] X. Ma, B. Qi, Y. Zhao, H.-K. Lo, Practical decoy state for quantum key distribution, Phys. Rev. A 72 (1) (2005).