

Commun. Theor. Phys. **74** (2022) 095103 (15pp) https://doi.org/10.1088/1572-9494/ac68d8

A prototype of quantum von Neumann architecture

Dong-Sheng Wang

CAS Key Laboratory of Theoretical Physics, Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China

E-mail: wds@itp.ac.cn

Received 13 January 2022, revised 20 April 2022
Accepted for publication 21 April 2022
Published 29 August 2022

Abstract

A modern computer system, based on the von Neumann architecture, is a complicated system with several interactive modular parts. It requires a thorough understanding of the physics of information storage, processing, protection, readout, etc. Quantum computing, as the most generic usage of quantum information, follows a hybrid architecture so far, namely, quantum algorithms are stored and controlled classically, and mainly the executions of them are quantum, leading to the so-called quantum processing units. Such a quantum–classical hybrid is constrained by its classical ingredients, and cannot reveal the computational power of a fully quantum computer system as conceived from the beginning of the field. Recently, the nature of quantum information has been further recognized, such as the no-programming and no-control theorems, and the unifying understandings of quantum algorithms and computing models. As a result, in this work, we propose a model of a universal quantum computer system, the quantum version of the von Neumann architecture. It uses ebits (i.e. Bell states) as elements of the quantum memory unit, and qubits as elements of the quantum control unit and processing unit. As a digital quantum system, its global configurations can be viewed as tensor-network states. Its universality is proved by the capability to execute quantum algorithms based on a program composition scheme via a universal quantum gate teleportation. It is also protected by the uncertainty principle, the fundamental law of quantum information, making it quantum-secure and distinct from the classical case. In particular, we introduce a few variants of quantum circuits, including the tailed, nested, and topological ones, to characterize the roles of quantum memory and control, which could also be of independent interest in other contexts. In all, our primary study demonstrates the manifold power of quantum information and paves the way for the creation of quantum computer systems in the near future.

Keywords: quantum computation, quantum channel, von Neumann architecture

1. Introduction

In quantum computing, we often apply sequences of unitary operations on multi-qubit states, followed by measurements. This is described in the quantum circuit model, the most popular model for universal quantum computing [1]. Being universal is not only vital to prove its own consistency [2–4], but also to demonstrate its power relative to conventional computing. However, the quantum circuit model is not complete so far in the sense that it does not provide a *quantum computer system*, the quantum version of the von Neumann architecture of modern computers [5].

A modern computer system contains at least five modular components: the input, output, memory unit, control unit, and computing unit (also known as a central processing unit). In particular, the memory contains stored programs, which enables the power of modern classical computers to automate the execution of algorithms. The current paradigm for quantum computing is a quantum–classical hybrid: the quantum circuit model mainly serves as a quantum central processing unit, which is usually classically controlled with quantum algorithms stored as classical programs. There are excessive amounts of classical ingredients which appear inevitable.

There have been persistent efforts to go beyond the scope of the circuit model, even starting from the beginning of quantum information. In the setting of a quantum Turing machine, it was pursued if quantum computing can be fully quantum [6–8],

i.e. with all components including the read-write head, the address of qubits, programs, control, halt signal, etc being quantum. It was then discovered that programs cannot be made quantum in the sense that once a program is stored as a quantum state, it cannot be read out deterministically [9]. An obstruction for the quantum control over arbitrary quantum operations is also revealed lately [10–13]. These studies together with other no-go theorems, e.g. [14–23], illustrate the sharp distinction between quantum information and classical ones.

The difficulty to formalize a universal quantum computer system is linked to a vital issue of quantum physics. It concerns if there is a so-called quantum–classical boundary, and how or where to draw such a boundary [24]. From the modern theory of quantum decoherence [25], classicality arises if the coherence of a quantum system is lost or delocalized into another system. As the elements of quantum information, qubits are considered to be the extensions of bits and probabilistic bits (or pbits), in the sense that a qubit is a superposition of bit values, and it leads to pbits if it is measured. This heuristic indicates that, instead of being puzzled by the nature of quantumness, the central issue for a proper model of a quantum computer system is to unfold the quantum advantages for various information processing tasks.

In this work, we propose a prototypical model of a universal quantum computer system. The central ingredient is a stored-program scheme based on the quantum channel-state duality [26, 27]. The stored quantum programs can be composed together, processed into other ones, and executed to realize quantum algorithms. Our model is not only universal and modular but also quantum-secure, in the sense that it is protected by the uncertainty principle, which has played vital roles in quantum communication and cryptography [28].

Our work is made possible based on a few recent progressions. First, the essence of some no-go theorems becomes clearer, including the no-programming [29–31], no-control [12], and the incompatibility between transversality and universality of logical gates [32–39]. The linearity of quantum operations and fundamental constraints by the uncertainty principle are revealed. Meanwhile, a scheme using Choi states as stored quantum programs is proposed [40], which turns out to be the proper way to bypass the accuracy constraint by the uncertainty principle. Also, we recently presented a physical understanding of various universal computing models [41]. In particular, we pointed out the relation between the uncertainty principle and logical gates, and the relation between quantum algorithms and quantum combs [42–46]. These progresses, but not limited to, enable the formation of a universal quantum computer system with less classical ingredients.

1.1. Overview of our model

Here we provide an overview of the model of a quantum computer system (QCS), which shares similarities with the classical case but also shows key distinctions. As for the classical case, there are five modular components: the input, output, memory unit, control unit (CU), and central processing unit (CPU). See figure 1. We treat the CU as a separate part from the CPU. We only focus on the functionality of these components, e.g. how they work and relate with each other. We do not study devices or hardware in this paper which should be specified for a practical QCS. This will be further discussed in section 7.

Figure 1. The five primary components of a computer system: the input, output, memory unit, control unit, and central processing unit. Their relations in terms of control and information flows are not shown explicitly here. More details can be found in computer-science textbooks, e.g. [47].

A modern CPU has a few components such as its own memory and control, but here we treat it as an arithmetic unit and study the quantum version, the quantum processing unit (QPU). A QPU contains at least two parts: the qubits required to perform quantum circuits, and the devices that are needed to realize quantum gates on the qubits and qubits from the memory. The quantum memory stores data and programs, both are in terms of Choi states [26, 27]. Quantum programs are stored as many copies of Choi states of quantum operations. Quantum data are stored as many copies of Choi states of preparation circuits for states. In our model, a computation is carried out on a part of memory under the control of a quantum control unit (QCU), which contains a collection of qubits. A large program is formed by the composition of small programs from the memory. The initial input state is 'injected' into the program by measurements. The solution to a given problem is obtained by measurements of observable on the final state.

Compared with classical cases, there are two major differences. First, due to the uncertainty principle, quantum data can be made secure, termed as 'quantum-secure' in this work, hence cannot be cloned or estimated efficiently. Second, quantum measurements are interactive with random outcomes. After a computation, the stored programs are consumed but can be restored. A quantum eavesdropper (Eve) or virus can destroy the programs by local measurements without knowing the programs, or, if powerful, can do joint measurements on a few copies to estimate a program state, for which the accuracy is limited by the uncertainty principle. On the contrary, classical data can be cloned if not encrypted, and classical measurements (e.g. read-write operations) are usually deterministic. This not only harvests quantum evolution as a computational resource but also harvests quantum memory and quantum measurements as resources.

Using stored quantum programs provides advantages especially when this cannot be efficiently done classically, but also leads to a few challenges. First, in order to execute a stored program or algorithm exactly and deterministically, the input to the algorithm is prepared in a heralded way. The output of an algorithm is required to be expectation values of observables, instead of being the final state itself. It appears that quantum computing occurs in Hilbert spaces, but eventually, the information carried by quantum states has to be

read out by measurements, i.e. we have to convert qubits into bits to obtain the solution to a given problem. This leads to our refinement of universality to be algorithmic (see section 3). Also due to the requirement of fault-tolerance, there will be the potential overhead of quantum error correction on the quantum memory (see section 5). This also makes a quest for the finding of fully or partially self-correcting qubits [48]. Our study proves that a QCS, as the analog of the classical ones, can be established in principle.

This work contains the following parts. In section 2 we survey the primary tools including quantum channels, quantum combs, quantum error correction, the quantum circuit model and quantum algorithms, and a few no-go theorems and their relations with the uncertainty principle. In section 3 we explain the main ingredients for our model of QCS, including the stored quantum programs and data, an extension of quantum circuits by stored programs, which are defined as tailed quantum circuits, a definition of algorithmic universality to describe the execution of quantum algorithms, and a description of the quantum control unit. We further discuss the features and requirements of the model in section 4, and its relation with some other models. In section 5 we show that QCS can be made fault-tolerant, hence completing the formalization of universal QCS. Finally, in section 6 we briefly discuss a few extensions of our study and conclude with more problems relating to QCS in section 7.

2. Preliminary

In this section, we review the primary background to make our presentation self-consistent. Along the way, we clarify facts and draw connections that are relevant to our study of QCS.

2.1. Quantum operations

We consider finite-dimensional Hilbert spaces. A pure state $|\psi\rangle$ is an element of a Hilbert space $\mathcal{H}$ with the normalization $\langle\psi|\psi\rangle = 1$ and any global phase is unphysical. A mixed state or density operator $\rho \in \mathcal{D}(\mathcal{H})$ is a nonnegative semidefinite operator acting on $\mathcal{H}$ with $\mathrm{tr}\rho = 1$, for $\mathcal{D}(\mathcal{H}) \subset \mathcal{B}(\mathcal{H})$ as the convex set of all mixed states, $\mathcal{B}(\mathcal{H})$ as the space of bounded linear operators acting on $\mathcal{H}$. A simple way to distinguish mixed states from pure states is by the purity

$$\mathrm{tr}(\rho^2) \leqslant 1, \tag{1}$$

which only equals to 1 for pure states. The purity is preserved under unitary evolution, which is described by unitary operators $U \in \mathcal{U}(\mathcal{H})$ in the unitary group $\mathcal{U}(\mathcal{H})$ acting on $\mathcal{H}$. More general evolution is described as completely positive, trace-preserving (CPTP) maps [49] of the form

$$\mathcal{E}(\rho) = \sum_i K_i \rho K_i^\dagger, \tag{2}$$

for $\forall\, \rho \in \mathcal{D}(\mathcal{H})$, also known as quantum channels, and K_i known as Kraus operators. This is also called a Kraus operator-sum representation, which is not unique due to an isometric degree of freedom. The minimal number of Kraus operators is the rank of the channel. A positive operator-valued measure (POVM) is a set $\{F_i\}$ with $F_i \geqslant 0$, $\sum_i F_i = \mathbb{1}$. It can be constructed from Kraus operators with $F_i = K_i^\dagger K_i$. A quantum instrument $\{\Phi_i\}$ is a set of CP maps Φ_i so that the sum of them $\Phi = \sum_i \Phi_i$ is TP, and the set of indices $[i]$ is fixed. A quantum channel $\mathcal{E}$: $\mathcal{D}(\mathcal{H}_1) \to \mathcal{D}(\mathcal{H}_2)$ does not need to preserve the dimensions of Hilbert spaces, but for simplicity, our presentation is made for the dimension-preserving case without loss of generality. We use $\mathcal{E}$: $\mathcal{D}(\mathcal{H})$ to represent a channel $\mathcal{E}$ that acts on $\mathcal{D}(\mathcal{H})$.

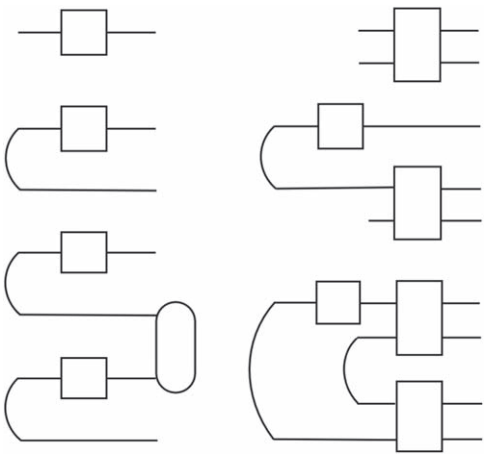

Figure 2. Quantum circuit diagrams (from top to bottom, left to right) for a quantum channel, Choi state, composition of two Choi states, unitary dilation of a channel, initial-state injection scheme on a Choi state, and superchannel acting on a Choi state. The curved wires are ebits, boxes are quantum operations, their meanings shall be clear from the main text.

Any quantum channel $\mathcal{E}$: $\mathcal{D}(\mathcal{H})$ can be represented as its dual state

$$\omega_\mathcal{E} := \mathcal{E} \otimes \mathbb{1}(\omega), \tag{3}$$

usually known as a Choi state [26, 27], for $\omega := |\omega\rangle\langle\omega|$, and $|\omega\rangle := \sum_i |ii\rangle / \sqrt{d}$, $d = \dim(\mathcal{H})$. The state $|\omega\rangle$ is a Bell state, also known as an ebit. Kraus operators can be found from the eigenvalue decomposition of $\omega_\mathcal{E}$. Choi states are bipartite and for clarity, we label them as site A and site B in order. The partial trace of a Choi state is constrained as $\mathrm{tr}_\mathrm{A}\omega_\mathcal{E} = \mathbb{1}/d$, and $\mathrm{tr}_\mathrm{B}\omega_\mathcal{E} = \mathcal{E}(\mathbb{1})/d$. Given $\omega_\mathcal{E}$, the action of the channel can be obtained as

$$\mathcal{E}(\rho) = d\ \mathrm{tr}_\mathrm{B}[\omega_\mathcal{E}(\mathbb{1} \otimes \rho^t)], \tag{4}$$

for ρ^t as the transpose of a state $\rho \in \mathcal{D}(\mathcal{H})$.

We see that ω is the dual state of $\mathbb{1}$. There is a concise graphical way to illustrate this map, see figure 2. For an operator $A \in \mathcal{B}(\mathcal{H})$, define its vectorization as

$$|\omega_A\rangle := A \otimes \mathbb{1}|\omega\rangle, \tag{5}$$

which has the property $|\omega_A\rangle = \mathbb{1} \otimes A^t|\omega\rangle$. The vectorization or duality is just to bend over the input towards the bottom of the output wire. This can be generalized if the given system is multi-partite. For n-partite operators, $|\omega\rangle^{\otimes n}$ is needed and it

holds

$$|\omega_A\rangle = A \otimes \mathbb{1}|\omega\rangle^{\otimes n} = \mathbb{1} \otimes \tilde{A}|\omega\rangle^{\otimes n}, \tag{6}$$

for $\tilde{A} = RA^tR$, R is the unitary operation that reverses the order of the subsystems. For the bipartite case, R is the swap operation. The relation above can be seen as to shuffle the operator A along the wires from the top to the bottom. Alternatively, an n-partite operator $A \in \mathcal{B}(\mathcal{H})$ can be just treated as a single partite, and we can use a high-dimensional Bell state $|\omega\rangle \in \mathcal{H} \otimes \mathcal{H}$ to define the vectorization. In this case, $\tilde{A} = A^t$. Despite the two choices, we still view the Choi state of an operator as a unique definition.

The general actions on Choi states $\omega_{\mathcal{E}} \in \mathcal{C}(\mathcal{H} \otimes \mathcal{H}) \subset \mathcal{D}(\mathcal{H} \otimes \mathcal{H})$ are found to be superchannels [42–44], which again can be represented by their dual states. To describe them, we use the unitary dilation representation of them which is more appropriate for the quantum circuit model. For a channel $\mathcal{E}$: $\mathcal{D}(\mathcal{H})$, from dilation, it can be realized by a unitary U with

$$\mathcal{E}(\rho) = \mathrm{tr}_a \mathcal{U}(\rho \otimes |0\rangle\langle 0|), \tag{7}$$

for $\mathcal{U}$ as the superoperator form of a unitary U, the trace over an ancilla at an initial state $|0\rangle$, which realizes Kraus operators as $K_i = \langle i|U|0\rangle$, with $\{|i\rangle\}$ as states of the ancilla. In order to tell channels from superchannels, we use a hat on the symbols for superchannels. A superchannel $\hat{\mathcal{S}}$: $\mathcal{C}(\mathcal{H} \otimes \mathcal{H})$ can be realized as

$$\hat{\mathcal{S}}(\mathcal{E})(\rho) = \mathrm{tr}_a \mathcal{V}\, \mathcal{E}\, \mathcal{U}(\rho \otimes |0\rangle\langle 0|), \tag{8}$$

for $\rho \in \mathcal{D}(\mathcal{H})$, $\mathcal{E}$: $\mathcal{D}(\mathcal{H})$, $\mathcal{U}$ and $\mathcal{V}$ are unitary, and a is an ancilla. Note that the dimension of V can be larger than U, while here we find no need to provide the details of the ancilla. The Choi state $\omega_{\mathcal{E}}$ can be made explicit by bending over the input wires, see figure 2, with

$$\hat{\mathcal{S}}(\mathcal{E})(\rho) = \mathrm{tr}_{\bar{\mathrm{A}}} \mathcal{V} \otimes \tilde{\mathcal{U}}(\omega_{\mathcal{E}} \otimes \omega)(\mathbb{1} \otimes \rho^t \otimes |0\rangle\langle 0|), \tag{9}$$

where the support of each operator shall be easy to see hence omitted for simplicity. The trace is over the subsystems except for the top one, A. The unitary $\tilde{\mathcal{U}}$ is the transpose of U conjugated by a swap. This formalism for superchannels includes channels as a special case with no channel $\mathcal{E}$ as input and no $\tilde{\mathcal{U}}$. We see that in order to change a channel to another one, we have to use both a pre- and post-unitary operations, with a memory wire connecting them. More generally, quantum n-combs are defined when $(n-1)$ channels are taken as input sandwiched between unitary operations [42–46]. As such, states, channels, and superchannels are also called 0-combs, 1-combs, and 2-combs, respectively.

In quantum computing, an important type of quantum operation is quantum error correction (QEC), which corrects errors that occur on quantum error-correction codes (QECC). A quantum code is often defined by an encoding isometry V: $\mathcal{H}_L \to \mathcal{H}_P$, with $V^\dagger V = 1$, or by the projector $P = VV^\dagger$ on the code space, $\mathcal{C} \subset \mathcal{H}_P$. A set of error operators $\{E_i\}$ acting on a code P is correctable when

$$PE_i^\dagger E_j P = c_{ij} P, \tag{10}$$

and $[c_{ij}] := \rho_P$ can be viewed as a state. The correction or recovery scheme, $\mathcal{R}$, is defined as a set $\{R_k\}$ with $R_k = \frac{1}{\sqrt{d_k}} P F_k^\dagger$ for d_k as eigenvalues of ρ_P, and F_k satisfies $PF_k^\dagger F_\ell P = d_k \delta_{k\ell} P$. When the error operators form a channel, $\mathcal{N}$, the logical information is perfectly recovered $\mathcal{R}\mathcal{N}(|\psi\rangle) = |\psi\rangle$, $\forall |\psi\rangle \in \mathcal{C}$. In addition, the condition for error-detection by P is $PE_iP = e_iP$, weaker than the QEC condition.

We see that QEC is to find the inverse of a map $\mathcal{N}$, which does not exist in general but is possible when its action on a subspace $\mathcal{C}$ is concerned. The QEC condition (10) is actually more powerful: any set of operators $\{A_i\}$ with each A_i as a linear combination from $\{E_i\}$ can also be corrected by $\mathcal{R}$. This is often known as the linear Kraus-span property as the error operators are considered as Kraus operators for channels. For codes with tensor-product form $\mathcal{H}_P = \otimes_n \mathcal{H}_n$, we only need to consider a spanning set of errors for each subsystem n, as others are linear combinations of them. For multi-qubit codes, we often consider Pauli bit-flip error X and phase-flip error Z for each qubit, with Y as a product of them, and that is sufficient to characterize the primary error-correction features of a code, such as code distance and threshold. A QECC is usually denoted as $C = [[n, k, d]]$, which uses n physical qubits to encode k logical qubits, and has a distance $d = 2t + 1$, namely, can correct errors that act on up to t physical qubits at unknown sites. There are also approximate QECC which we do not focus on in this paper.

2.2. Quantum circuit model

There are a few universal quantum computing models [41] and here we employ the quantum circuit model (QCM) in our study. In the setting of QCM, universality means that any unitary operator $U \in SU(2^n)$ can be efficiently approximated by $\tilde{U}$ to an arbitrary accuracy ϵ. The circuit size of $\tilde{U}$ shall be polynomial of $\log \frac{1}{\epsilon}$.

In QCM, a unitary U is realized as a sequence of gates that are available, and an algorithm is realized by acting U on an input state, followed by a measurement process for readout. A gate is a unitary operation that can be turned on and off by external control. A set of gates is called a universal gate set if the product of gates from it can approximate any unitary efficiently. The two well-known examples are the set $\{H, T, \mathrm{CX}\}$ and the set $\{H, \mathrm{CCX}\}$ for

$$H = \frac{1}{\sqrt{2}}(X + Z),\ T = Z^{\frac{1}{4}},\ CX = P_0 \otimes \mathbb{1} + P_1 \otimes X, \tag{11}$$

and CCX as the Toffoli gate $\mathrm{CCX} = P_0 \otimes 1 + P_1 \otimes \mathrm{CX}$, CX usually denoted as CNOT, and X, Z as Pauli matrices, $P_{0,1} = \frac{\mathbb{1} \pm Z}{2}$. The Toffoli gate is known to be universal for classical computation. We do not need to consider nonunitary gates since, due to the dilation theorem, any nonunitary quantum channel can be realized by a unitary operator, together with final measurements on ancilla. When QEC rounds are required, they can also be realized by unitary operations followed by measurements.

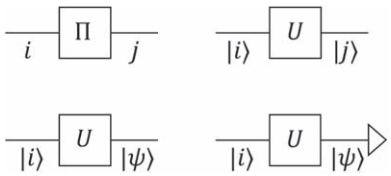

Figure 3. Quantum algorithm and special cases (from left to right, top to bottom): classical one by permutation Π from bits to bits (e.g. a bit-string i to j), quantum one by unitary U from bits $|i\rangle$ to bits $|j\rangle$ or a nontrivial entangled state $|\psi\rangle$, and with the final bit values read out by measurement. This also applies to quantum meta-algorithms, i.e. combs. The classical circuit-design algorithm is not shown here, see figure 4.

In quantum circuits, quantum gates are causally ordered in the sense that the space and time location of a gate is classical and controlled by a classical computer or system. Also the direction of time, namely, the information flow, of a circuit is fixed, which is from the given input to the desired output. This agrees with the intuition from classical algorithms, which can be generalized for the quantum case (see section 3).

Quantum algorithms can be described relative to a universal computing model, such as adiabatic algorithms and quantum walks, but they can be translated into the circuit model. A quantum algorithm, Q, is usually described by a quantum circuit, together with a proper initial state and readout scheme. The readout contains the solution S to a given problem, P, which is the input of a classical circuit-design algorithm, A. Namely, A: $P \mapsto [Q]$ for $[Q]$ as a classical description of the quantum algorithm Q: $|0\rangle \mapsto S$, with S as measurement outcome and $|0\rangle$ as an initial state required by Q. See figures 3 and 4. This framework applies to many quantum algorithms, including quantum phase estimation, quantum simulation, etc [1]. The classical algorithm A is an essential part and often difficult to find, and it could be limited if $[Q]$ cannot be efficiently described.

The solution to a problem is encoded in the expectation value

$$o_f := \mathrm{tr}(\mathcal{O}\rho_f) \tag{12}$$

of observable $\mathcal{O}$ on the final state ρ_f. The measurement scheme of $\mathcal{O}$ is required to be efficient. This usually requires running many rounds of the circuit to estimate the observable. Sometimes the output is just the final state ρ_f without a measurement, which can be subsequently given into another quantum algorithm, yet eventually, measurement is required to convert quantum states into classical values. Quantum algorithms are generically probabilistic, namely, within a given accuracy ϵ, the approximate solution o_f is obtained with a high probability p that can be efficiently boosted towards 1. We emphasize here that a quantum algorithm is a generalization of a probabilistic algorithm, or in other words, it unifies classical and probabilistic ones. The output of the quantum algorithm is of the form $\mathrm{tr}(\mathcal{O}\rho_f)$, which needs both quantum state generation and quantum measurement.

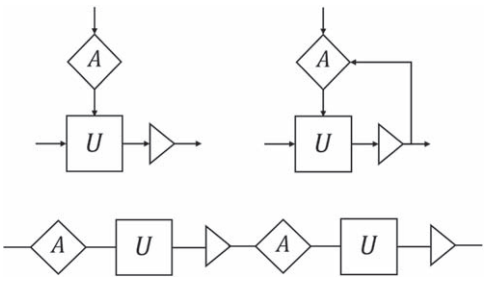

Figure 4. Structures of quantum algorithms. The basic structure (top-left) has a classical algorithm A that designs the quantum algorithm Q, or labelled by the unitary circuit U. It extends to the iterative classical-quantum algorithms (top-right), which can be 'stretched' into a linear flow (bottom), and the most general forms in terms of quantum combs, see figure 8.

2.3. Quantum no-go theorems

Although quantum information is an extension of the classical case, there are significant differences that are apparently revealed by various so-called no-go theorems. The goal of this section is to show that a few fundamental no-go theorems are equivalent, and the underlying physics is the uncertainty principle.

The no-cloning theorem [14, 15] is well known and it states that an unknown quantum state cannot be cloned by any quantum operations, e.g. from $|\psi\rangle$ to $|\psi\rangle|\psi\rangle$. In general, it applies to $n \to m$ cloning process

$$U|\psi\rangle^n|\chi\rangle = |\psi\rangle^m, \tag{13}$$

for $m > n$, U and $|\chi\rangle$ are independent from the input $|\psi\rangle$. It would violate the linearity of quantum operations. It was shown to be equivalent to a quantum estimation problem [19], namely, a perfect quantum cloning machine will estimate an unknown state perfectly, and a perfect quantum estimation machine will also yield many copies of an unknown state. The quantum estimation task can be realized by a unitary operation with

$$U|\psi\rangle^n|\chi\rangle = |\psi'\rangle|[\psi]\rangle, \tag{14}$$

for $|[\psi]\rangle$ as a bit-string encoding of $|\psi\rangle$ or the estimated parameters in it. The states $|[\psi]\rangle$ are orthogonal for any pair of input states. For pure states, the set of states that can be perfectly cloned or estimated are orthogonal with each other.

The no-programming theorem [9] is also due to orthogonality. If there is a quantum operation U that realizes

$$U|d\rangle|P_G\rangle = G|d\rangle|P'_G\rangle, \tag{15}$$

for any data state $|d\rangle$ and any program state $|P_G\rangle$ that encodes the unitary operator G serving as a program, then the set of programmable states $\{|P_G\rangle\}$ have to be orthogonal. Orthogonality implies classicality since an orthogonal set of states can be viewed as a basis of a Hilbert space. Any superposition of basis states is not programmable. Note that the output is product states. The no-programming has also been extended to POVM [20] and the physics is similar.

It is easy to see its relation with the cloning and estimation task. If $|P_G\rangle$ can be cloned or estimated perfectly, it would lead to a perfect universal programming machine. On

the contrary, the state $|P'_G\rangle$ is independent of $|d\rangle$ so it can be recycled [31] to recover $|P_G\rangle$, which would lead to the execution of G an arbitrary number of times, which is a cloning of G or state $G|d\rangle$.

Furthermore, it is well understood that quantum estimation can be done probabilistically or approximately, and a probabilistic scheme can also be treated as an approximate scheme if the outcomes are mixed together. In quantum metrology, it has been established that an arbitrary parameter held by a quantum state or evolution can only be estimated approximately, and the error is lowered bounded due to the uncertainty principle and also the extension of it via quantum Fisher information [50]. Although there are many variants of the uncertainty principle based on different operations, the essential fact it established from the Cauchy—Schwarz inequality

$$|\langle x|y\rangle|^2 \leqslant |\langle x|x\rangle|\,|\langle y|y\rangle|, \tag{16}$$

for $|x, y\rangle$ from an inner-product space is that the uncertainty of two non-commuting operators on a quantum system is lower bounded by the degree of their non-commutativeness. Optimal schemes for universal quantum cloning, estimation, and programming are known [29, 31, 51, 52], and the common fact is that, given n copies of the unknown parameter, state, or gate, the accuracy is no better than the scaling $1/n^2$, which is achievable using multipartite entangled states. Without entanglement, the accuracy bound reduces to the so-called short-noise limit $1/n$. The accuracy cannot be exponential (e.g. 2^{-n}) as that would converge to the perfect case, hence violating the no-go theorems.

There is also a no-go theorem in the setting of QECC [23], which states that the transversal logical gates on a finite-dimensional quantum error-detection code form a finite group. We can see the connection with the above no-go theorems from the error-detection condition and output state form. An encoding operation can be defined by a unitary operator U: $|\ell\rangle \to |\psi_\ell\rangle$ that maps logical states $|\ell\rangle \in \mathcal{H}_L$ to the encoded states $|\psi_\ell\rangle \in \mathcal{H}_P$. For a transversal partition $\mathcal{H}_P = \otimes_n \mathcal{H}_n$, any transversal unitary operator takes the form

$$U = \otimes_n U_n. \tag{17}$$

Now suppose the states $|\psi_\ell\rangle$ are product states $\otimes_n |\psi_n\rangle$ instead of being entangled. If the error-detection condition is required, then all 1-local states are fixed, which is then impossible to encode logical states $|\ell\rangle$. If the states $|\psi_\ell\rangle$ are product states, while the error-detection is only for classical errors, e.g. for bit-flip errors on any site but not for phase-flip errors, then we find orthogonal logical states are mapped to orthogonal 1-local states for all sites. Here we notice the orthogonality and this actually leads to classical error-detection codes. Now if we allow entangled states $|\psi_\ell\rangle$, the error-detection will require the set of transversal logical gates to be a finite group. This is a feature that is not present in other no-go theorems we mentioned above. If the error-detection condition is dropped, then $SU(2^n)$-covariant codes exist [32–39] which apparently allow any transversal logical gates. However, the accuracy of such codes is limited by the uncertainty principle, and for n transversal parts, the accuracy is upper bounded by $1/n^2$ [36, 37]. Indeed, the encoding operation can be viewed as an estimation scheme of $|\ell\rangle$, and the transversal operations are the parallel black-box calls. The optimal entangled resource states for estimation are also found to be optimal for $SU(2^n)$-covariant codes [38].

Another notable no-go theorem is the no-control over unknown quantum operations [10, 11], which forbids the process

$$U_2(\mathbb{1}\otimes\mathbb{1}\otimes U)U_1 = U_3 \otimes CU, \tag{18}$$

for fixed unitary $U_{1,2,3}$ acting on the tri-partite system, and any unknown U acting on the third subsystem, and CU as the controlled-U on the second and third subsystems, up to an arbitrary global phase on U. It turns out it is of a different category from the three no-go theorems above. It not only violates the linearity of quantum operations, but also the meaningless of global phases of quantum states or operations. The above process would lead to the perfect distinction between a gate U and $-U$, which is impossible in quantum theory. This has been shown as a topological obstruction from the Borsuk–Ulam theorem [12], and there is no approximate version of the no-control theorem. It also forbids the process $|\psi_1\rangle|\psi_2\rangle \mapsto |\psi_1\rangle \oplus |\psi_2\rangle$ since it maps global phases of one state to relative phases between the two states. In general, this is a 'no-packing' of unknown quantum operators, and such packings are not valid operations on Hilbert spaces.

It turns out there is an easy scheme to overcome the no-control theorem [10–13], which is to know at least one eigenstate of the gate U. Given the black-box access of U, the CU can be realized as

$$CU \otimes \mathbb{1}_a = \mathrm{CSWAP}(\mathbb{1}\otimes U_a)\mathrm{CSWAP}, \tag{19}$$

for a as an ancilla and $\mathbb{1}_a$ as the completely-mixed state of it, CSWAP as the controlled-SWAP gate, and the input state for the target of CU is the known eigenstate of U. In addition, another case is when the gate CU itself is given with U unknown, which applies to situations that U has trivial action on a few levels or modes that belongs to the whole Hilbert space [13]. These levels or modes are eigenstates of CU with eigenvalues 1.

3. Ingredients

3.1. Stored quantum programs and data

Stored programs and data are important components of the von Neumann architecture of computers. Despite the no-programming theorem, here we present an efficient scheme of stored quantum programs and data, based on the recent study [40], and this is the starting point for our model of QCS. The quantum memory unit in QCS contains many programs and data, each of which are prepared with many copies made and stored with definite addresses in the memory unit. The query of quantum memory and computation with them is efficiently controlled by the quantum control unit. For simplicity, we focus on the unitary evolution of pure states, which can be quite straightforwardly extended to the non-unitary evolution of mixed states.

For a pure state $|\psi\rangle \in \mathcal{H}$, a simple scheme is just to prepare $|\psi\rangle$ and store it as quantum data. While here we introduce a different scheme that will be employed in our model. We define its computational preparation circuit as U with $|\psi\rangle = U|0\rangle$, for $|0\rangle$ as a computational basis state of $\mathcal{H}$. The U is not unique but can always be chosen properly. We define quantum data of $|\psi\rangle$ as the Choi state $|\omega_U\rangle$ of its preparation circuit U. For an n-qubit state, it needs $2n$ qubits to store it as quantum data. In general, this storage can be made arbitrarily accurate if the accuracy of $|\omega\rangle$ and U can be guaranteed. On the contrary, we can store a state $|\psi\rangle$ directly as classical data, denoted as $[\psi]$. For generic n-qubit states, this is to convert all the amplitudes $\psi_i = \langle i|\psi\rangle$ into bits, which cannot be efficient with respect to n. Only special types of states such as stabilizer states [53] can be described efficiently using bit strings on classical computers.

A quantum program or algorithm is defined by a unitary operator, while often also requiring special initial states as input and special measurements as a readout scheme. For simplicity, we treat the input as a part of data, and readout as a separate stage. Namely, a quantum program merely refers to a unitary operator U. Similar to quantum data, a stored quantum program $|\omega_U\rangle$ is the Choi state of U.

From the relation (4), the action $U|\psi\rangle$ can be realized by the measurement of $|\psi\rangle$ on site B of $|\omega_U\rangle$ [40]. Define a binary projective measurement $\{P_0, P_1\}$ with

$$P_0^t = |\psi\rangle\langle\psi|, \quad P_1 = \mathbb{1} - P_0, \tag{20}$$

for P_0^t as the transpose of P_0, and the measurement outcomes 0 and 1 are recorded. When the outcome is 0, the state on site A is $U|\psi\rangle$. When the outcome is 1, the state on site A is $\mathbb{1} - U|\psi\rangle\langle\psi|U^\dagger$. This is enough for the readout of the computational result $\mathrm{tr}(\mathcal{O}\rho_f)$, which is encoded as an observable $\mathcal{O}$ on site A, for $\rho_f = U|\psi\rangle\langle\psi|U^\dagger$ and the value $\mathrm{tr}(\mathcal{O})$ is efficiently computable as required by the readout scheme (see section 3.3 for more discussions).

When the input $|\psi\rangle$ is stored as the quantum data of its preparation circuit, we need the composition of Choi states. Given two Choi states $|\omega_{U_1}\rangle$ and $|\omega_{U_2}\rangle$, the symmetry-based quantum gate teleportation [40] leads to the state $|\omega_{U_2U_1}\rangle$ or $|\omega_{U_2^tU_1}\rangle$ in a heralded way. Namely, a qubit ancilla is used to encode the outcomes of Bell measurement being trivial (with no byproduct) and non-trivial (with Pauli byproducts), recorded by 0 and 1. The case of being 0 requires no further action but only teleports U_2^t, while the case of being 1 can teleport U_2 but need a correction rotation $U_{2;\mathrm{adj}}$ as the adjoint representation of U_2. As a unitary matrix can be decomposed as a product of two symmetric unitary matrices, a program can be stored by two Choi states and the composition

$$U_{\mathrm{UQT}}|\omega_{U_1}\rangle|\omega_{U_2}\rangle = |\omega_{U_2U_1}\rangle \tag{21}$$

will avoid the transpose (of U_2) and becomes deterministic. In this work, we name the symmetry-based quantum gate teleportation as an $SU(d)$-covariant or universal quantum teleportation (UQT) since it can teleport any unitary gate, which eventually guarantees the universality of our stored-program scheme. The UQT serves as the composition operations and is denoted by a rounded box in the circuit, see figure 2 (or by distinguished wires in figure 7 or circles in figure 10).

We see that after the execution, the quantum program is destroyed, i.e. overwritten by trivial bits. Even with many copies of a quantum program, it can only be used for a finite amount of runs. This seems to be a drawback of quantum data, but it turns out to be the opposite. Quantum programs are secure and expensive, which stand as key distinctions from classical programs and memory, which can be perfectly cloned and reused forever, if no encryption or subscription is required (also see section 4). After a quantum program is executed and overwritten by trivial bits, it shall be restored by downloading it.

A program shall not only be stored in memory, but also shall be downloadable from the internet. Although the internet is an extra object of a computer system, here we present a scheme to download quantum programs from an anticipated quantum internet [54]. The downloading operation for the classical case contains the cloning of classical data and overwriting some amount of bits in the memory. For the quantum case, this cannot be done since quantum data cannot be cloned. Quantum programs are provided by quantum software vendors, who wish to keep the programs unknown to the agent. The quantum internet requires the communication of qubits. However, Holevo's bound asserts that a qubit can communicate at most a single bit [55], consistent with the uncertainty principle and the no-cloning theorem. For a quantum program U, flying qubits such as photons can be prepared as the entangled state $|\omega_U\rangle$ itself in principle, but the agent cannot use them to recover the quantum program U efficiently in its quantum memory.

Fortunately, quantum cryptography [28] shows that bits can be securely communicated using qubits. Meanwhile, a quantum program U can be described by its classical information efficiently, if it is given as a sequence of gates composing a quantum circuit, $\prod_\ell U_\ell$. Namely, given a universal gate set, the type of each gate (e.g. H, T, CX) can be encoded by two bits. The space-time location of an elementary gate can be encoded by bits efficiently. Therefore, the classical information of U contains the bits for the types and space-time locations of the gates. For a circuit that is efficient with respect to the number n of qubits and accuracy ϵ, the bit-string description of the gate sequence of U is efficient. Note that although it can be used to construct U, yet as a single matrix, U cannot be stored efficiently by bits in general. The bit-string description of U can be encrypted by a quantum software vendor, and securely communicated using qubits (or even using the post-quantum encryption [56]). After receiving the bit-string description, the agent, which is a quantum computer and is protected by encryption, can apply the gate sequence to restore the quantum program. The quantum program remains unknown to the agent, hence the program is securely downloaded and the quantum program is secure. In all, we demonstrated the viability of stored quantum programs, and we also remark that practical security is far more complicated than the primary scheme we presented here.

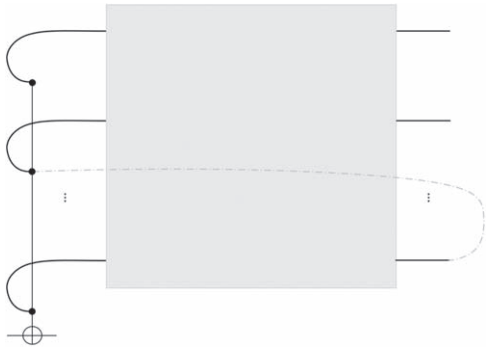

Figure 5. Schematics of a tailed quantum circuit. The shaded box is a unitary operator. The curved wires contain the tails, which are arranged on the left side for the input. The multiple-controlled NOT gate (i.e. an n-fold Toffoli gate) is for the initial-state injection. The dotted gray wire is an example of a contraction on a head-tail pair.

3.2. Tailed quantum circuits

In accord with the stored quantum programs, we need to introduce a modification of the standard quantum circuit model, termed as tailed quantum circuit model. Recall that a quantum program $|\omega_U\rangle$ is a bipartite state, and we name the subsystem acted upon by U as 'head', and the other as 'tail'. A tailed quantum circuit is a quantum circuit U of a sequence of quantum gates that act on a few qubits and the heads of a few ebits, while each tail is left unchanged. Without qubit input, a tailed quantum circuit is just a program state $|\omega_U\rangle$. The input for a program is injected into the circuit by making measurements on a few tails. The readout is specified by quantum measurement and supported on a few heads. The input is always carried by tails, output is carried by heads. See figure 5 for an example.

Due to ebits, we can apply some novel operations with tailed quantum circuits. Given a collection of elementary-tailed quantum circuits, they can be connected in series and parallel to form larger programs. The elementary tailed quantum circuits are those for elementary gates from universal gate sets. Familiar gates are Pauli gates X, Y, Z, and H, T, CZ, CX, CCX, CCZ, etc. We focus on H, T, and CZ as examples, see figures 6 and 7 for an example with simplified notations. They are all symmetric matrices, so they can act on either the upper or the bottom wire, but as a convention, we choose the upper one as the head. Using the UQT defined in section 3.1, smaller tailed circuits can be composed into larger ones, just as the composition of gates in a usual quantum circuit.

A large program $U \in SU(d)$ can be stored with a 'bold' high-dimensional tail of dimension d, or with a qubit tail for each qubit in the system when d is converted to 2^n for a number n. When there are multiple tails and heads, contraction (or fusion) of a pair can be made by a Bell measurement on them. A pair can be of any form, head-head, tail-tail, or head-tail. Cares are needed to properly choose the time flow after the input and output measurements are being made to avoid backward flow in time or closed time loops. The application of the contraction will be further discussed in section 6.

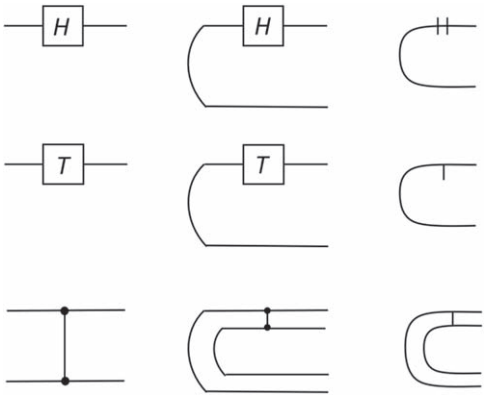

Figure 6. The elementary quantum gates H, T, and CZ (left), their Choi states (middle), and simplified notations (right).

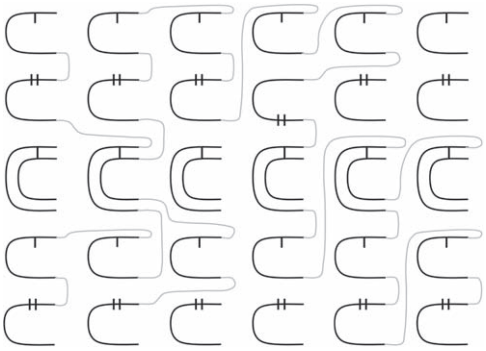

Figure 7. An example of a quantum circuit realized by composition of elementary quantum programs for H, T, and CZ stored in the quantum memory. The compositions are denoted by gray wires.

As quantum measurement outcome is random, a tailed quantum circuit can be employed to sample a collection of circuits. For each ebit, a measurement of Pauli Z on its tail injects either state $|0\rangle$ or $|1\rangle$ with equal probability, or in other words, a Pauli X byproduct that cannot be corrected. For the contraction of a pair of head and tail, the Bell measurement yields a connected wire but with Pauli byproduct X, Y, or Z that cannot be corrected either. The measurement outcomes are recorded so can be used to sample circuits with different initial states and final measurements. However, this also means that an initial state, e.g. a computational state $|\vec{0}\rangle = |00\cdots 0\rangle$, cannot be prepared deterministically. This is indeed true since a qubit state cannot be obtained deterministically from an ebit. To execute an algorithm on a tailed quantum circuit, we need a different input-injection scheme which is studied in the next subsection.

3.3. Algorithmic universality

Here we define a so-called algorithmic universality relative to tailed quantum circuits and show how to run quantum algorithms with tailed quantum circuits. Basically, it needs initialization, unitary evolution, and the readout measurement. The unitary operation is given as the program state. Suppose the input is required to be $|\vec{1}\rangle$, which is easier to illustrate than $|\vec{0}\rangle$. We define a binary measurement $P = \{P_0, P_1\}$ with

$P_0 = \mathbb{1} - P_1$ and $P_1 = |\vec{1}\rangle\langle\vec{1}|$. A qubit ancilla initialized at $|0\rangle$ is needed to realize it. For n qubit tails, a n-fold Toffoli gate [57] is needed to copy the AND of all qubit values to the ancilla (see figure 5). The Z measurement on the ancilla realizes P: 0 for P_0, and 1 for P_1. The n-fold Toffoli gate can be decomposed into a product of a polynomial number of elementary gates. In particular, it can be decomposed as a cascade of $n-1$ Toffoli gates with $n-1$ qubit ancilla. Furthermore, it is easy to see the probability for P_1 is 2^{-n}, which is tiny. In other words, most of the time P_0 is realized. In this case, the output is $o_f' := \mathrm{tr}(\mathcal{O}) - \langle\psi_f|\mathcal{O}|\psi_f\rangle$ for $o_f = \langle\psi_f|\mathcal{O}|\psi_f\rangle$ as the true desired output, with $|\psi_f\rangle = U|\vec{1}\rangle$. The value of o_f' can be estimated by running the algorithm multiple times, as usually to be the case for quantum algorithms. Given that $\mathrm{tr}(\mathcal{O})$ is easy to compute, then $o_f = \mathrm{tr}(\mathcal{O}) - o_f'$ is obtained with high probability.

The above demonstrates that any quantum algorithm that can be realized on a usual quantum circuit can also be realized on a tailed quantum circuit. This proves the universality of the tailed quantum circuit model, which we term here as an *algorithmic universality* since it is defined in the setting of quantum algorithms. In passing, probably a better term could be observational universality since it is due to the ability to compute observable values or a weak universality based on the weak operator topology on a Hilbert space [58]. Furthermore, the algorithmic universality is actually more complete than the usual notion of universality, which does not take account of the cost of readout explicitly. For instance, some readout schemes cannot be done efficiently (with respect to ϵ), such as the estimation of unknown gates [29] and an approximate stored-program scheme [31], which actually reduce the universality to a quasi universality [35]. The algorithmic universality guarantees the universality of a model to realize quantum algorithms.

The study above also extends to more general quantum algorithms. Just as quantum combs are able to describe general quantum operations, it is natural to see that they also describe more general types of quantum algorithms [41]. A quantum comb takes a set of quantum objects $\{Q_n\}$ as input but uses quantum operations to change them into a desired output, with a quantum adversary as a resource. This forms a quantum meta-algorithm that designs a quantum algorithm by another quantum algorithm (the comb). The input $\{Q_n\}$ can be given as unitary oracles or known as black boxes. If they are given as stored programs $\{|\omega_{Q_n}\rangle\}$, we can use the composition scheme to connect them and form a comb. That is to say, we can implement a quantum comb using a stored-program scheme, see figure 8. Each unitary operator in the comb can be decomposed as a product of two, then an input object Q_n surrounded by two unitary operators is a block for a superchannel $\hat{\mathcal{S}}_n$. Now for each $|\omega_{Q_n}\rangle$ we first apply a superchannel $\hat{\mathcal{S}}_n$ on it, and then we apply the composition of them in sequence to form the comb. In addition, the classical-quantum hybrid algorithms (see figure 4), which feed measurement results from Q to A, can be viewed as a special case of classical combs with the sequence of Qs as its input and the measurements and A as the comb.

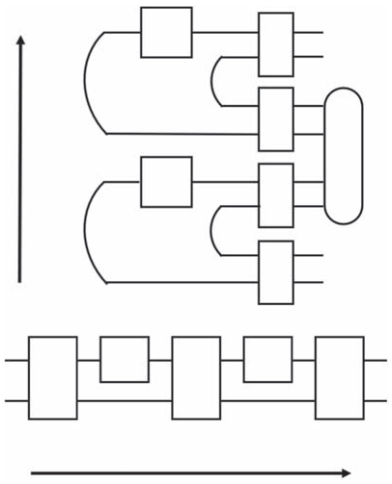

Figure 8. A quantum comb (bottom) and its realization by the composition on Choi states (top). The information flows towards the right for the circuit of a comb, while flows upwards for the composition.

3.4. Quantum control unit

The control unit (CU) is an important component of a computer system. In general, it interacts with all other components of a computer system, but usually, it does not carry the final solution to the given problems. It controls or monitors the procedure and progress of information processing during computation. For instance, it needs to guarantee that instructions from any algorithm are executed in the right order. Also, it needs to guarantee that data and programs are read and written at the right place in the memory. Besides universality, there is also a feature of modularity such that CU and CPU are independent components of a computer system. Here, we lay out the basic principle to design a quantum control unit (QCU), without going into details of the so-called control bus and data bus [47]. We find there are freedoms that can be explored for its design.

We first analyze the idea of control and quantum control from the viewpoint of algorithms. A control scheme is a procedure that aims to achieve a goal by applying a set of control items on a system. It can be understood algorithmically, but with different goals and costs from algorithms. Quantum control often uses a set of quantum operators to control a scheme to achieve an operational goal [59]. The control operators may be supported by the target system itself, or not. It is straightforward to see that a general quantum control can be described as a quantum comb, with the adversary as the control. The controller becomes entangled with the target system in general, but it shall not contain final solutions. For instance, a common task in quantum control is to use external fields to control the states of atoms or electrons, which is often semi-classical since there is no entanglement between the fields and the systems. There are also controls with feedback. The simplest example is the CNOT gate, which is an entangling gate. In terms of Pauli operators, the target Pauli is also copied to the control qubit. Another example is using measurement on the target, and the outcome

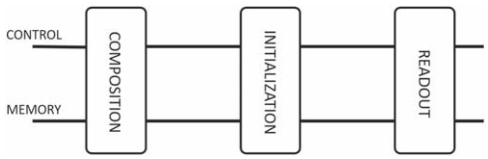

Figure 9. The schematics to show the role of quantum control in a computation.

is fed back to the control, and this basically behaves as an iterative algorithm.

In our model, the QCU is qubits and their interactions with other components. This includes the control of the composition of programs, the initial-state injection measurement, and readout measurement, see figure 9. We find there is a freedom to choose for the nature of the control. The minimal type is classical, namely, any quantum algorithm can be monitored classically. The classical control does not carry quantum information. On the other hand, the control can be as quantum as possible, and this is described as quantum combs. As a result, the control scheme could be an inevitable part of a quantum algorithm, and the design of quantum algorithms would involve a significant part of the controller. Actually, some quantum algorithms can be understood in this way, such as the quantum switch [60] and the linear combination of unitary operations [61, 62].

Furthermore, whenever the controller participates in the computation in a nontrivial way, the corresponding comb can be stored as a set of programs again. Then another level of control is required to realize this comb by composition. This eventually reduces the control to classical ones. Such a reduction of the control sequences or levels is similar to phenomena in other topics, such as algorithms and quantum measurements. For algorithms, we can always add a pre-algorithm that designs the algorithm, no matter whether it is classical or quantum. For quantum measurement, there requires a cut to tell how a special value of an observable is obtained from a set of possible values, as originally studied by von Neumman [24].

Therefore, we require a minimal or basic set of functions of a QCU. We shall make the QCU modular so it can be used for all quantum targets easily. The basic operation is the quantum control of a quantum operation, which is harder than classical control of a quantum operation, but easier than general entangling quantum operations. As the study of the no-control over unknown quantum operations in section 2.3 reveals that, an unknown quantum operation as a grey box can be controlled by qubits. Using qubits as control is a resource that can be explored, just as the quantum algorithms mentioned above demonstrated. With bits, each run of a quantum algorithm is a unitary U applying on a fixed initial state $|\psi_i\rangle$ and a fixed readout o_f. With qubits, there could be a superposition of quantum algorithms by applying different compositions and measurements conditioned on the control qubits, i.e. different unitary, initial states, and readout can be realized in parallel and interfere.

4. Features

Our model of QCS requires both qubits and ebits (i.e. Bell states), Bell measurement and its generalization, and both unitary evolution and measurement play vital roles. The input does not have to be fixed at the beginning. The readout scheme avoids the problem in the setting of the no-programming theorem, which requires input as a separate state and the output is a state instead of measurement outcomes. A nontrivial POVM readout can be simulated by a unitary and a simple projective measurement from the dilation theorem. Quantum control in general requires multiple-controlled gates, which can be decomposed into elementary gates but that complicates the control process. The usage of ebits as quantum memory signifies the distinction between quantum information and classical information. It is protected by the uncertainty principle. Below we discuss our model in more detail to reveal its relations with other computing models or schemes, and its overall features, requirements, and limitations.

First, we clarify the notion of a computer system. The universal computing model mainly refers to the framework to process information, while a computer system mainly refers to the architecture to divide and combine various aspects of information processing. A quantum computer system is a particular quantum system that is designed for computing tasks. Together with our formulation of universality, we emphasize the difference between quantum computing and a usual quantum evolution that occurs in nature every day. Quantum computing is usually formulated just as instrumentalists do: prepare an initial state, let it evolve, and then observe via measurement. This is an analog description. Instead, our study shows that quantum computing is more and needs to be almost fully digital. The states need to be encoded as qubits, programs encoded as ebits, and the final solution o_f is carried by an observable, which can in principle be converted to bit strings such as using a quantum amplitude estimation algorithm [63].

As has been mentioned, the current QCS is a quantum–classical hybrid, known as a quantum random-access machine [64], which contains both classical and quantum registers. The quantum data on the quantum registers would not be entangled with the control or program. In the original sense [6–8], our model of QCS is also not fully quantum, despite the usage of quantum control and quantum programs. For instance, we do not need a halt qubit to signal the end of a computation. Instead, a computation or an algorithm ends with measurements. Actually, the vital thing is not whether it is fully quantum or not; instead, it shall be its computational power and physical flexibility, such as being local and modular. As we have discussed for the quantum control, we can always shift the boundary between the quantum and classical parts. Also, we currently do not require the ability of the quantum query of all quantum data in superposition, like that in the scheme [65] for a quantum random-access memory.

The QPU in our model is described by the quantum circuit model, which can be replaced by other universal models, such as quantum Turing machine and quantum cellular automata, which are the quantum versions of their classical analogs [41]. Meanwhile, if we view the composition as a whole, it prepares

matrix-product states or tensor-network states, which are universal forms of quantum states [66–68]. This is linked with a local model of the quantum Turing machine [69] which prepares matrix-product states assisted by a quantum adversary. But here it is the quantum adversary (or edge) that carries the logical information.

Our model can be viewed as an extension of quantum communication and cryptography by the universal computation on ebits and Choi states, besides qubits. The computation with Choi states introduces input by measurements, and mainly concerns the final expectation value o_f of observable as output. If the input is treated as an initial state and carried by a separate system from the program, one has to use a highly entangled optimal program state, e.g. a generalized Choi state and global covariant measurement for the retrieval operation [29, 31, 51], which is constrained by the uncertainty principle. That is to say, our stored programs are protected by the uncertainty principle, namely, if an Eve or virus tries to obtain a stored program, a global measurement on many copies of it has to be applied with limited accuracy.

Our model can also be seen as an extension of measurement-based quantum computing (MBQC) [70]. In MBQC, the resource state does not contain the program; instead, the program is the measurement base. Each measurement is one-local and it is based on the $U(1)$ symmetry of teleportation [71]. The measurements are adaptive in order to avoid the byproduct of teleportation. The original universal blind quantum computing [72] is based on MBQC, which achieves blindness or security via pbits to hide measurement bases. The security relies on the computational difficulty of searching for the right program over a large set of possible ones. In fusion-based quantum computing [73], Bell measurements are used to grow graph states or stabilizer states, but there is no stored quantum programs. In our setting, we use two-local covariant measurements as an extension of Bell measurements, which do not contain the programs; instead, the programs are 'pre-stored' as quantum states. Our model has a close connection with valence-bond solids (VBS) [66] (see section 6 for more details). As a result, we establish our model of QCS to be digital, universal, modular, and quantum-secure, and can also be made fault-tolerant.

5. Fault tolerance

In this section, we show how the QCS is consistent with the requirement of fault tolerance. The quantum fault-tolerance or threshold theorem states that universal quantum computation on logical qubits can be realized if the physical error rate for each logical qubit is below a threshold that is determined by the QEC [1]. The QEC itself could also be noisy, whose effect is to reduce the threshold value. Physical gates to realize logical ones may also be imperfect, which are treated as perfect ones followed by noises. The fault tolerance is implicitly required by universality since between any two logical gates, U and V, QEC is needed to ensure the identity gate $\mathbb{1}$ to form $U\mathbb{1}V$. For a QCS, it also needs to replace qubits with logical qubits for all its components, including the memory, control, gates, measurements, etc. We find that the main issue is the fault tolerance of the composition operations.

For the quantum memory, an ebit is replaced by a logical ebit, which can be efficiently prepared. If a code is defined by an isometry V, the logical ebit can be obtained as

$$|\omega\rangle_L = V \otimes V|\omega\rangle, \tag{22}$$

for $|\omega\rangle$ as the encoded ebit. A logical stored program state $|\omega_U\rangle_L$ is obtained by applying a logical gate U on $|\omega\rangle_L$. For a code $[[n, k, d]]$, logical gates U commute with its projector P, $[U, P] = 0$. For different codes, the form of U varies significantly. The composition of two logical program states $|\omega_U\rangle_L$ and $|\omega_V\rangle_L$ is done by the logical version of the composition scheme, which is a UQT on $2n$ physical qubits. A qubit ancilla is needed for a composition, which in principle can use a different code.

In the composition scheme, logical gates need to be symmetric to avoid the transposition of gates. This is a nontrivial requirement on logical gates. We find a concise scheme that satisfies the requirement. First, notice that the tensor product of symmetric gates is still symmetric. So it is easy to teleport transversal logical gates that are symmetric. It turns out the elementary gates are all symmetric such as logical H, T, and CZ. Then we can use the scheme of code switching [74] to combine the transversal logical gates from different codes. For instance, if there are two codes C_1 and C_2 that can be fault-tolerantly switched into each other by measuring their stabilizers, then their transversal logical gates can be combined together, even achieving universality [75]. Then we can apply code switching and composition on them to form large programs.

For composition, a gate $U_{\text{adj}} \in SO(N^2 - 1)$ for $N = 2^n$ is needed to teleport a logical gate U. Such gates U_{adj} can be efficiently done. However, they are not logical gates in general for the given codes. Actually, a sequence of compositions can be viewed as a code concatenation procedure, with the given codes as the inner codes, and the outer code being an $SU(N)$ VBS edge code [35]. Now the fault-tolerance of the composition is the fault-tolerance of the concatenation procedure, and this is also a common issue when just preparing a code. For a code defined by an encoding isometry V, there could be noises during the encoding itself. It could be hard to do QEC before the code is prepared, and the threshold theorem implicitly assumes that this can be done. For codes that the code projector P is composed of a product of smaller projectors $P = \prod_n P_n$, e.g. for stabilizer codes [53] and codes defined by frustration-free Hamiltonians, each projector P_n is prepared gradually, hence can be used for QEC during the encoding or concatenation.

Despite the similarity, there is also an important difference between the composition and code concatenation. In order to run a quantum algorithm, the VBS outer code does not have to be prepared firsthand. Instead, we can do an initial-state injection first, and then composite the first logical gate, then the second logical gate, and so on. The VBS outer code only appears in the time direction. Actually, this is similar to MBQC [70] and it reduces the idler time and QEC

cost. To ensure fault tolerance, QEC is performed on the inner codes before and after the teleportation of each logical gate. Therefore, the fault tolerance is mainly determined by the inner codes for logical qubits and ebits.

More generally, logical gates can be of higher depths. Ideally, we shall employ codes with some non-transversal high-depth logical gates as symmetric matrices, which currently we do not know of. This is left as an interesting task for the future. Nevertheless, there is a method to enforce fault tolerance by adapting the QEC dynamically during the time period of logical gates to account for the effects of the gate on the code. The issue is also encountered for the braiding of non-Abelian anyons [76]. Suppose a logical gate $U_L = \prod_i U_i$ is a sequence of symmetric non-logical gates U_i (note all elementary gates and their tensor products are symmetric). Each U_i only affects a code C locally, and it defines a new code C_i, which is stored separately. QEC can be performed for each C_i. When the gate sequence U_i is applied one after another, it induces a sequence of codes $C_{[i]}$, which is for gates $U_{[i]} = U_i \cdots U_2 U_1$. The support of $U_{[i]}$ will be maximal in the middle of U_L, and QEC becomes challenging but can be performed in principle.

6. Extensions

Before we conclude, we present a few direct extensions of our model. These are based on the key features or elements in our model, e.g. the usage of ebits, composition, and contraction operations. The extensions here can be used in our model, in principle, while their properties and applications remain intriguing. They could also be of independent interest for other tasks such as quantum simulation, quantum communication, and quantum error-correction codes.

As for the circuit model, a quantum circuit can be imprinted on a geometric structure such as graphs or regular lattices. This also applies to the composition of programs, which can be extended from the one-dimensional flow to high-dimensional or structured flows. For instance, we showed that two qubit-gates are connected in parallel through a CZ gate, which requires the $SU(2)$-covariant gate teleportation. This can be extended to the $SU(4)$-covariant case by treating CZ and qubit gates on an equal footing. It can lead to interesting tensor-network states, also known as PEPS. See figure 10 for an example. Other Lie groups can also be used for the composition by treating the wires as representations. If the program states are arranged regularly, it can yield states with SPT orders [77]. A special class of states is valence-bond solids [66] which have global Lie-group symmetry and weak SPT order. Indeed, if we start from empty program states, i.e. ebits, and use the projectors onto certain representations, this yields valence-bond solids, which, in this setting, shall be viewed as a herald of our composition network. It has been well established that valence-bond solids can be used for MBQC [78], with a program served by a sequence of measurement bases, which is classical. On the contrary, in our scheme here the program is quantum and has been pre-stored in the ebits, as we have discussed in section 4.

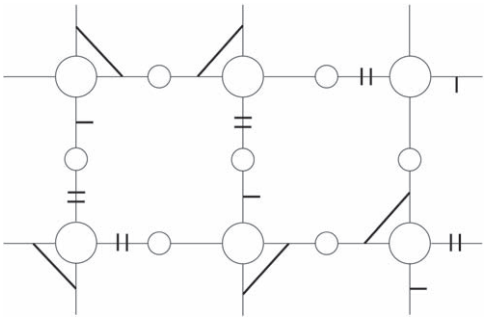

Figure 10. An example of a portion of a high-dimensional composition of quantum programs. The big circles are $SU(4)$ compositions, the small circles are $SU(2)$ compositions. We have stretched the wires for H and T programs to be lines. The inclined bold segments are CZ gates.

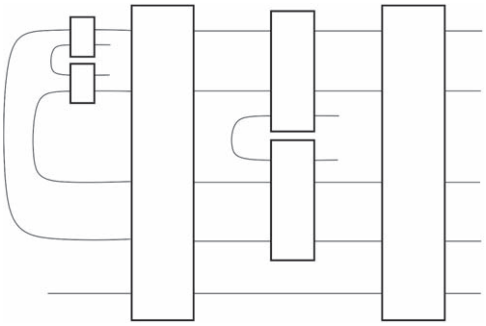

Figure 11. An example of a higher-order quantum circuit. The boxes are unitary gates. The dimension of each wire could be different.

A generalization of the tailed quantum circuits is to use higher-order quantum operations [42–44], which is based on an iterative usage of the channel-state duality. For instance, a superchannel can also be converted into a Choi state and then acted upon by a channel of one higher order. They can all be properly represented by unitary quantum circuits. See figure 11 for an example. Recall that a channel is a 1-comb. One central difference between nth-order operations and n-combs is that the dimension of the former likely grows exponentially faster than the latter. That is, it is much harder to climb up the hierarchy by increasing the order than by adding more input channels to a comb. Despite this, higher-order quantum operations lead to interesting circuits with concatenated or nested structures, with each unitary being a 'nest.' Nests on different levels of the hierarchy can be composed together leading to higher-order quantum combs. These nested circuits involve nonlocal gates acting on many wires and may be used to describe novel quantum dynamics.

Using local gates and also the contraction operation, we introduce another type of circuit, called topological quantum circuits, as an extension of the tailed quantum circuits. Intuitively, an ebit together with a contraction forms a closed loop. A tailed quantum circuit with many tails and many contractions can be viewed as a complex of loops. These loops are linked or knotted together due to their interactions, i.e. quantum gates. From knot theory [79], a knot or link is formed by vertices (or crossings) and lines connecting them. A vertex contains a top wire and a bottom wire. Now we map

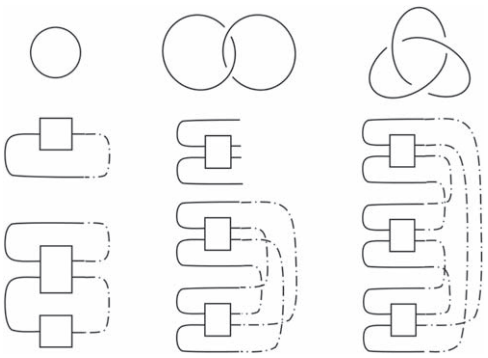

Figure 12. Primary examples of topological quantum circuits. From left to right, top to bottom: a circle, its circuit, and a circuit for two separate circles; a link, the circuit for a vertex, and the circuit for the link; a knot, and the circuit for the knot. The boxes are unitary gates. The dashed lines are contractions. Additional qubit wires for each box are not shown explicitly.

a vertex to a type of circuit with two tails and two heads, together with possible qubit wires, and the gate in the circuit is not fixed. As a convention, we assign the top (bottom) wires as tails (heads). A segment between two vertices is mapped to a contraction. We can see primary examples in figure 12, while more complicated topological quantum circuits can also be constructed following the roles. Without additional qubits, each diagram is an overlap between a product of Choi states and a product of Bell states. If there are un-contracted qubits, a topological circuit, in general, prepares a multi-qubit entangled state, without direct interactions among the qubits. These entangled states are not appearing in the form of matrix-product states, and their properties are worth separate investigations.

7. Discussion and conclusion

In this work, we present a primary model of a universal quantum computer system. Our study extends the current formation of quantum–classical hybrid computer systems, and it shows that there are proper quantum advantages for information storage, processing, protection, etc. In the meantime, a modern computer system is far more complicated than the original von Neumann architecture, and this provides challenging opportunities for further development of quantum computer systems.

Our study highlights the role of the uncertainty principle, revealing the fundamental difference between classical and quantum information. The nature of quantum information is intriguing [80, 81]. Our model indicates that ebits (i.e. Bell states) are the elements for memory, while qubits are the elements for computation. On the contrary, classical bits are the elements for both classical computation and memory, while pbits are only used for computation. It also appears that ebits are the analog of bits, which is an echo of the fact that quantum teleportation is the analog of the one-time pad [82]. Yet Bell states are entangled and nonlocal, which is not the case for bits. Furthermore, qubits are combinations of bits and pbits, and somehow are analog signals since the amplitudes in superposition are carried by qubits instead of being digitized into bit strings. In the so-called classical regime, the uncertainty principle also applies to the Fourier transform, which is often used to analyze signals. For the dynamics, we know that quantum computing generalizes permutation and stochastic operations to unitary operations followed by quantum measurements. However, there is no need to convert qubits or pbits into bits, and simulate unitary or stochastic operations by permutations. Quantum computing is still digital in the sense that any task can be decomposed into elementary gates and projective measurements acting on qubits and ebits.

A key feature of our model is that the stored quantum programs are consumed after a computation. In order for the computer system to be reusable, the programs have to be restored by downloading from the internet, if renewing the hardware of the quantum memory were not advisable. Actually, the internet and communication network have become an indispensable part of the modern computing network. Therefore, an extension of our model is to include a quantum internet toward distributed quantum computing [54]. A major merit of distributed computing is that it distributes or breaks a computational task into communication among several parts, each with fewer requirements on its computational power. As a whole, it can also be viewed as a restricted QCS, e.g. direct operations on two QCS are not available. This can be described by the local quantum Turing machine model [69], which has a close connection with matrix-product states and tensor-network states. This points to a viable connection between tensor-network states and quantum communication.

We mentioned that we do not study QCS from the perspective of hardware, which requests various physical and engineering tasks. For instance, the quantum memory in our model is a collection of quantum states. In the sense of hardware, quantum memory refers to a hardware or device that can store quantum states, more precisely, store the quantum systems that carry quantum information. Devices for input and output are diverse and also require sophisticated information processing techniques, but in our study, they are basically bit strings generated by quantum measurements. QCS requires the understanding of physical problems including, but not limited to, dissipation, efficiency, energy cost, the role of fundamental laws, mechanism of almost all devices, etc. More broadly, like a giant artificial quantum system, QCS is a subject to reveal the interplay between quantum physics and modern technologies based on control, system, and information theory.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (Grant No. 12 047 503 & No. 12 105 343). Previous conversations with I Affleck, P Hayden, R Laflamme, H Nautrup, T Shi, Y Wang, J Watrous, Y Wu, Y Yang, S Yi, and G Zhu are acknowledged.

References

[1] Nielsen M A and Chuang I L 2000 *Quantum Computation and Quantum Information* (Cambridge: Cambridge University Press)

[2] Deutsch D 1985 Quantum theory, the Church-Turing principle and the universal quantum computer *Proc. R. Soc.* A **400** 97–117

[3] Bernstein E and Vazirani U 1997 Quantum complexity theory *SIAM J. Comput.* **26** 1411–73

[4] Yao A C-C 1993 *Proc., 34th Annual Symp. on Foundations of Computer Science* (IEEE) pp 352–61

[5] von Neumann J 1958 *The Computer and the Brain* (New Haven, CT: Yale University Press)

[6] Myers J M 1997 Can a universal quantum computer be fully quantum? *Phys. Rev. Lett.* **78** 1823–4

[7] Ozawa M 1998 Quantum nondemolition monitoring of universal quantum computers *Phys. Rev. Lett.* **80** 631–4

[8] Shi Y 2002 Remarks on universal quantum computer *Phys. Lett.* A **293** 277–82

[9] Nielsen M A and Chuang I L 1997 Programmable quantum gate arrays *Phys. Rev. Lett.* **79** 321–4

[10] Araujo M, Feix A, Costa F and Brukner C 2014 Quantum circuits cannot control unknown operations *New J. Phys.* **16** 093026

[11] Thompson J, Modi K, Vedral V and Gu M 2018 Quantum plug n' play: modular computation in the quantum regime *New J. Phys.* **20** 013004

[12] Gavorova Z, Seidel M and Touati Y 2020 Topological obstructions to implementing controlled unknown unitaries arXiv:2011.10031

[13] Vanrietvelde A and Chiribella G 2021 Universal control of quantum processes using sector-preserving channels *Quant. Infor. Comput.* **21** 1320

[14] Dieks D 1982 Communication by EPR devices *Phys. Lett.* A **92** 271

[15] Wootters W K and Zurek W H 1982 A single quantum cannot be cloned *Nature* **299** 802–3

[16] Barnum H, Caves C M, Fuchs C A, Jozsa R and Schumacher B 1996 Noncommuting mixed states cannot be broadcast *Phys. Rev. Lett.* **76** 2818–21

[17] Mayers D 1997 Unconditionally secure quantum bit commitment is impossible *Phys. Rev. Lett.* **78** 3414–7

[18] Lo H-K and Chau H F 1997 Is quantum bit commitment really possible? *Phys. Rev. Lett.* **78** 3410–3

[19] Bruss D, Ekert A and Macchiavello C 1998 Optimal universal quantum cloning and state estimation *Phys. Rev. Lett.* **81** 2598–601

[20] D'Ariano G M and Perinotti P 2005 Efficient universal programmable quantum measurements *Phys. Rev. Lett.* **94** 090401

[21] Zeng B, Cross A and Chuang I L 2011 Transversality versus universality for additive quantum codes *IEEE Trans. Inf.* **57** 6272–84

[22] Chen X, Chung H, Cross A W, Zeng B and Chuang I L 2008 Subsystem stabilizer codes cannot have a universal set of transversal gates for even one encoded qudit *Phys. Rev.* A **78** 012353

[23] Eastin B and Knill E 2009 Restrictions on transversal encoded quantum gate sets *Phys. Rev. Lett.* **102** 110502

[24] von Neumann J 1955 *Mathematical Foundations of Quantum Mechanics* (Princeton, NJ: Princeton University Press)

[25] Zurek W H 2003 Decoherence, einselection, and the quantum origins of the classical *Rev. Mod. Phys.* **75** 715–75

[26] Choi M-D 1975 Positive linear maps on complex matrices *Linear Algebra Appl.* **290** 285–90

[27] Jamiołkowski A 1972 Linear transformations which preserve trace and positive semidefiniteness of operators *Rep. Math. Phys.* **3** 275

[28] Bennett C H and Brassard G 1984 Quantum cryptography: Public key distribution and coin tossing *Proc. IEEE Int. Conf. on Computers, Systems and Signal Processing (Bangalore, India)* (New York: IEEE) pp 175–9

[29] Bisio A, Chiribella G, D'Ariano G M, Facchini S and Perinotti P 2010 Optimal quantum learning of a unitary transformation *Phys. Rev.* A **81** 032324

[30] Kubicki A M, Palazuelos C and Pérez-García D 2019 Resource quantification for the no-programing theorem *Phys. Rev. Lett.* **122** 080505

[31] Yang Y, Renner R and Chiribella G 2020 Optimal universal programming of unitary gates *Phys. Rev. Lett.* **125** 210501

[32] Hayden P, Nezami S, Popescu S and Salton G 2021 Error correction of quantum reference frame information *PRX Quantum* **2** 010326

[33] Faist P, Nezami S, Albert V V, Salton G, Pastawski F, Hayden P and Preskill J 2020 Continuous symmetries and approximate quantum error correction *Phys. Rev.* X **10** 041018

[34] Woods M P and Alhambra Á M 2020 Continuous groups of transversal gates for quantum error correcting codes from finite clock reference frames *Quantum* **4** 245

[35] Wang D-S, Zhu G, Okay C and Laflamme R 2020 Quasi-exact quantum computation *Phys. Rev. Res.* **2** 033116

[36] Kubica A and Demkowicz-Dobrzański R 2021 Using quantum metrological bounds in quantum error correction: a simple proof of the approximate eastin-knill theorem *Phys. Rev. Lett.* **126** 150503

[37] Zhou S, Liu Z-W and Jiang L 2021 New perspectives on covariant quantum error correction *Quantum* **5** 521

[38] Yang Y, Mo Y, Renes J M, Chiribella G and Woods M P 2020 Covariant quantum error correcting codes via reference frames arXiv:2007.09154

[39] Wang D-S, Wang Y-J, Cao N, Zeng B and Laflamme R 2022 Theory of quasi-exact fault-tolerant quantum computing and valence-bond-solid codes *New J. Phys.* **24** 023019

[40] Wang D-S 2020 Choi states, symmetry-based quantum gate teleportation, and stored-program quantum computing *Phys. Rev.* A **101** 052311

[41] Wang D-S 2021 A comparative study of universal quantum computing models: towards a physical unification *Quantum Eng.* **3**

[42] Chiribella G, D'Ariano G M and Perinotti P 2008a Transforming quantum operations: quantum supermaps *Europhys. Lett.* **83** 30004

[43] Chiribella G, D'Ariano G M and Perinotti P 2008b Quantum circuit architecture *Phys. Rev. Lett.* **101** 060401

[44] Chiribella G, D'Ariano G M and Perinotti P 2009 Theoretical framework for quantum networks *Phys. Rev.* A **80** 022339

[45] Gutoski G and Watrous J 2007 Toward a general theory of quantum games *Proc. 39th ACM Symp. on Theory of Computing* 565574

[46] Jenčová A 2011 Generalized channels: channels for convex subsets of the state space *J. Math. Phys.* **53** 012201

[47] Nisan N and Schocken S 2007 *The Elements of Computing Systems: Building a Modern Computer from First Principles* (Cambridge, MA: MIT Press)

[48] Brown B J, Loss D, Pachos J K, Self C N and Wootton J R 2016 Quantum memories at finite temperature *Rev. Mod. Phys.* **88** 045005

[49] Kraus K 1983 States, effects, and operations: fundamental notions of quantum theory *Lecture Notes in Physics* vol 190 (Berlin: Springer)

[50] Helstrom C W 1976 *Quantum Detection and Estimation Theory* (New York: Academic)

[51] Derka R, Buz˘ek V and Ekert A K 1998 Universal algorithm for optimal estimation of quantum states from finite ensembles via realizable generalized measurement *Phys. Rev. Lett.* **80** 1571–5
[52] Fan H, Wang Y-N, Jing L, Yue J-D, Shi H-D, Zhang Y-L and Mu L-Z 2014 Quantum cloning machines and the applications *Phys. Rep.* **544** 241–322
[53] Gottesman D 1998 Theory of fault-tolerant quantum computation *Phys. Rev.* A **57** 127–37
[54] Wehner S, Elkouss D and Hanson R 2018 Quantum internet: a vision for the road ahead *Science* **362** 303
[55] Holevo A S 1982 *Probabilistic and Statistical Aspect of Quantum Theory* (Amsterdam: North-Holland)
[56] Bernstein D and Lange T 2017 Post-quantum cryptography *Nature* **549** 188–94
[57] Barenco A, Bennett C H, Cleve R, DiVincenzo D P, Margolus N, Shor P, Sleator T, Smolin J A and Weinfurter H 1995 Elementary gates for quantum computation *Phys. Rev.* A **52** 3457
[58] Wang D-S 2015 Weak, strong, and uniform quantum simulations *Phys. Rev.* A **91** 012334
[59] Brif C, Chakrabarti R and Rabitz H 2010 Control of quantum phenomena: past, present and future *New J. Phys.* **12** 075008
[60] Chiribella G, D'Ariano G M, Perinotti P and Valiron B 2013 Quantum computations without definite causal structure *Phys. Rev.* A **88** 022318
[61] Long G L 2011 Duality quantum computing and duality quantum information processing *Int. J. Theor. Phys.* **50** 1305
[62] Childs A M and Wiebe N 2012 Hamiltonian simulation using linear combinations of unitary operations *Quantum Inf. Comput.* **12** 901
[63] Brassard G, Hoyer P, Mosca M and Tapp A 2002 Quantum amplitude amplification and estimation *Contem. Mathemat.* **305** 53–74
[64] Knill E 1996 Conventions for quantum pseudocode *LANL Report LAUR-96-2724*
[65] Giovannetti V, Lloyd S and Maccone L 2008 Quantum random access memory *Phys. Rev. Lett.* **100** 160501
[66] Affleck I, Kennedy T, Lieb E H and Tasaki H 1987 Rigorous results on valence-bond ground states in antiferromagnets *Phys. Rev. Lett.* **59** 799–802
[67] Perez-Garcia D, Verstraete F, Wolf M and Cirac J 2007 Matrix product state representations *Quantum Inf. Comput.* **7** 401–30
[68] Schollwöck U 2011 The density-matrix renormalization group in the age of matrix product states *Ann. Phys.* **326** 96–192
[69] Wang D-S 2020 A local model of quantum Turing machines *Quantum Inf. Comput.* **20** 0213–29
[70] Raussendorf R and Briegel H J 2001 A one-way quantum computer *Phys. Rev. Lett.* **86** 5188–91
[71] Wang D-S 2019 Quantum computation by teleportation and symmetry *Int. J. Mod. Phys.* B **33** 1930004
[72] Broadbent A, Fitzsimons J and Kashefi E 2009 Universal blind quantum computation *Proc. 50th Annual Symp. on Foundations of Computer Science* vol 2009 (Los Alamitos, CA: IEEE Computer Society) pp 517–27
[73] Bartolucci S *et al* 2021 Fusion-based quantum computation arXiv:2101.09310
[74] Paetznick A and Reichardt B W 2013 Universal fault-tolerant quantum computation with only transversal gates and error correction *Phys. Rev. Lett.* **111** 090505
[75] Bombín H 2015 Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes *New J. Phys.* **17** 083002
[76] Nayak C, Simon S H, Stern A, Freedman M and Sarma S D 2008 Non-abelian anyons and topological quantum computation *Rev. Mod. Phys.* **80** 1083
[77] Childs A M, Gosset D and Webb Z 2013 Universal computation by multiparticle quantum walk *Science* **339** 791
[78] Stephen D T, Wang D-S, Prakash A, Wei T-C and Raussendorf R 2017 Computational power of symmetry-protected topological phases *Phys. Rev. Lett.* **119** 010504
[79] Rolfsen D 1976 *Knots and Links* (AMS Chelsea Publishing)
[80] Hayden P and Penington G 2020 Approximate quantum error correction revisited: introducing the alpha-bit *Commun. Math. Phys.* **374** 369–432
[81] Harrow A 2004 Coherent communication of classical messages *Phys. Rev. Lett.* **92** 097902
[82] Gisin N, Ribordy G, Tittel W and Zbinden H 2002 Quantum cryptography *Rev. Mod. Phys.* **74** 145–95