Improved real-space parallelizable matrix-product state compression and its application to unitary quantum dynamics simulation

Rong-Yang Sun [1,2,3] Tomonori Shirakawa [1,2,3,4] and Seiji Yunoki [1,2,4,5]

[1]*Computational Materials Science Research Team, RIKEN Center for Computational Science (R-CCS), Kobe, Hyogo 650-0047, Japan*
[2]*Quantum Computational Science Research Team, RIKEN Center for Quantum Computing (RQC), Wako, Saitama 351-0198, Japan*
[3]*RIKEN Interdisciplinary Theoretical and Mathematical Sciences Program (iTHEMS), Wako, Saitama 351-0198, Japan*
[4]*Computational Condensed Matter Physics Laboratory, RIKEN Cluster for Pioneering Research (CPR), Saitama 351-0198, Japan*
[5]*Computational Quantum Matter Research Team, RIKEN Center for Emergent Matter Science (CEMS), Wako, Saitama 351-0198, Japan*

Towards the efficient simulation of near-term quantum devices using tensor network states, we introduce an improved real-space parallelizable matrix-product state (MPS) compression method. This method enables efficient compression of all virtual bonds in constant time, irrespective of the system size, with controlled accuracy, while it maintains the stability of the wave-function norm without necessitating sequential renormalization procedures. In addition, we introduce a parallel regauging technique to partially restore the deviated canonical form, thereby improving the accuracy of the simulation in subsequent steps. We further apply this method to simulate unitary quantum dynamics and introduce an improved parallel time-evolving block-decimation (pTEBD) algorithm. We employ the improved pTEBD algorithm for extensive simulations of typical one- and two-dimensional quantum circuits, involving over 1000 qubits. The obtained numerical results unequivocally demonstrate that the improved pTEBD algorithm achieves the same level of simulation precision as the current state-of-the-art MPS algorithm but in polynomially shorter time, exhibiting nearly perfect weak scaling performance on a modern supercomputer.

I. INTRODUCTION

Owing to the recent rapid advancement of synthetic quantum devices [1–4], quantum computing is becoming a competitive candidate for next-generation computing. It utilizes these controllable quantum systems for quantum information processing and to address challenging problems beyond the capability of classical computing [5]. Due to the principles of quantum mechanics, a quantum computer maintaining a quantum state composed of N qubits can access a total of 2^N dimensions in its working space, i.e., the Hilbert space, hence providing ample computational resources for addressing problems with comparable exponential complexity, such as the quantum many-body problem [6]. An ideal quantum computer can encompass the entire Hilbert space, thereby offering indisputable exponential resources compared to a classical computer, which thus establishes the quantum advantage. However, currently available quantum computers are hindered by strong decoherence noise, allowing only a small fraction of the Hilbert space to be explored [7]. These noisy quantum computers are referred to as a noisy intermediate-scale quantum (NISQ) [8] device. The central challenge in near-term development of quantum computing is determining efficient strategies for utilizing NISQ devices to achieve a practical quantum advantage in meaningful computational tasks [6,9–11].

In parallel, to validate the attainment of practical quantum advantage and to establish a reliable and readily accessible testing environment for quantum algorithm development, extensive research has been focused on developing classical simulations of quantum computing. For exact simulations, two primary methods are employed: the state-vector method and the tensor contraction method. The state-vector method can simulate quantum circuits with any circuit depth (in polynomial time with respect to the circuit depth) and provides the complete amplitudes of the wave function. However, simulating a quantum circuit with more than 50 qubits using currently available classical computers is considered impossible [12,13]. On the other hand, the tensor contraction method [14–18] can simulate quantum circuits with up to around 100 qubits, but the arrangement of gates and the circuit depth are severely restricted, allowing only a limited portion of the wave-function amplitudes to be obtained [19]. Therefore, to efficiently simulate current NISQ devices with over 400 qubits [20], it is essential to develop more specialized simulation algorithms. These algorithms should consider the limitation that NISQ devices can only generate limited quantum entanglement.

Inspired by their great successes in the study of quantum many-body problems [21,22], tensor network states, particularly the matrix-product state (MPS) [23] and the projected entangled-pair state (PEPS) [24], have been employed to simulate quantum computing and have achieved accurate results in faithfully simulating NISQ devices [7,25,26]. In this paper, our focus will be on MPS-based simulation algorithms. Presently, MPS-based methods can perform full-amplitude approximate simulations for quantum circuits with over 100 qubits and moderate circuit depth, benefiting from its efficient

representation of quantum entanglement [27]. Nevertheless, restricted by the real-space sequential nature of these algorithms [7,25], efficiently simulating a NISQ device with hundreds, or even thousands, of qubits remains elusive.

In general, an MPS-based algorithm for quantum circuit simulations includes two main procedures. One involves applying quantum gates, and the other involves compressing the MPS to a computable size. Regarding the former, we can parallelly apply multiple gates as long as they do not have spacetime overlap. For the later, it has been realized that even if all the MPS virtual bonds are compressed in parallel, the error induced by this parallel compression is still manageable [27]. These observations imply the feasibility of developing a real-space parallelizable algorithm to simulate quantum circuits. In fact, in the study of the time evolution of a quantum many-body system, a similar parallel scheme has been proposed and has obtained promising results [28] (see also Ref. [29] for an alternative parallel approach to simulating Hamiltonian dynamics and Ref. [30] for a parallel algorithm of searching the ground state). However, a crucial issue that remains in this scheme is the fast-decaying wave-function norm caused by parallel MPS compression, which might result in serious numerical instability. This instability necessitates the renormalization of the wave function after simulating a certain period of time evolution [28]. Being a sequential procedure, this additional wave-function renormalization significantly diminishes parallelism and hampers the efficient utilization of the parallel computing environment.

Here we introduce an improved real-space parallelizable MPS compression (IPMC) method that can stabilize the wave-function norm without compromising parallelism. For an MPS in the canonical form, we prove that the wave-function norm, after undergoing the IPMC, stabilizes within an interval with two bounds uniformly converging to 1 from both sides, and the convergence improves as the compression error decreases. Moreover, we integrate an additional parallel regauging procedure into the IPMC to generate a better starting point for the next simulation step, thereby enhancing the simulation accuracy. Based on this IPMC method, we propose the parallel time-evolving block-decimation (pTEBD) algorithm to simulate unitary quantum dynamics in a fully parallelizable manner. We benchmark the pTEBD algorithm by extensively simulating typical random and parametrized quantum circuits on both one-dimensional (1D) and 2D qubit arrays. These numerical results demonstrate that the pTEBD algorithm achieves a comparable simulation precision to the previous sequential MPS algorithm proposed in Ref. [7] and can attain the same accuracy in polynominally shorter time. Meanwhile, the wave-function norm stabilizes to approach 1, instead of decaying exponentially to zero, throughout the entire simulation period with a circuit depth exceeding 1000. Consequently, we successfully achieve nearly perfect weak scaling performance in the pTEBD simulation, involving over 1000 qubits with over 250 computational nodes on the Supercomputer Fugaku installed at RIKEN.

The rest of the paper is organized as follows. First, we briefly summarize the MPS representation of quantum many-body states, the properties of the canonical form, and the time-evolving block-decimation (TEBD) algorithm (also denoted as the simple update algorithm) in Sec. II.

Next, we systematically describe the real-space parallelizable MPS compression method in Sec. III, where we introduce a wave-function norm stabilization method with a proof of its bounding theorem, explore the parallel regauging technique through the trivial simple update, and finally explain the IPMC method. In Sec. IV, we propose the pTEBD algorithm and demonstrate its accuracy, numerical stability, and performance through extensive simulations of random and parametrized quantum circuits in one and two dimensions. Finally, in Sec. V, we summarize the results of this paper and briefly discuss their impact on the development of near-term quantum computing. The details of the quantum circuits used for the simulations are described in Appendix A. An additional result regarding the accuracy of the pTEBD algorithm is provided in Appendix B. Furthermore, the pTEBD algorithm is applied to the simulation of the quantum Fourier transformation in Appendix C.

II. MPS REPRESENTATION OF QUANTUM MANY-BODY STATES

A. Matrix-product state

The Hilbert space $\mathcal{H}$ for an N-site quantum many-body system composed of qubits is described by the tensor product of N local Hilbert spaces $\mathcal{H}_i$ (spanned by orthogonal states $|0\rangle_i$ and $|1\rangle_i$ with the dimension $d = 2$) located on each qubit site i, i.e.,

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_N . \tag{1}$$

Therefore, the dimension of $\mathcal{H}$ is $\dim(\mathcal{H}) = 2^N$, increasing exponentially with N. Any quantum many-body state $|\Psi\rangle$ living on $\mathcal{H}$ can be represented using a complete many-body basis formed by a direct product of local states $|\sigma_i\rangle_i$ with $\sigma_i = 0$ or 1 on each $\mathcal{H}_i$, i.e., $|\sigma_1 \cdots \sigma_N\rangle = |\sigma_1\rangle_1 \otimes |\sigma_2\rangle_2 \otimes \cdots \otimes |\sigma_N\rangle_N$. With this basis, the quantum state $|\Psi\rangle$ is represented as

$$|\Psi\rangle = \sum_{\sigma_1,\ldots,\sigma_N} \Psi_{\sigma_1,\ldots,\sigma_N} |\sigma_1 \cdots \sigma_N\rangle , \tag{2}$$

where the coefficients $\Psi_{\sigma_1,\ldots,\sigma_N}$ can be regarded as a rank-N tensor. Note that, without specifying, the normalization condition $\langle\Psi|\Psi\rangle = 1$ is always maintained.

The MPS representation of $|\Psi\rangle$ is constructed by a tensor network decomposition, i.e., decomposing a tensor to a set of tensors (a tensor network) such that the contraction of these tensors restores the original tensor, as

$$\Psi_{\sigma_1,\ldots,\sigma_N} = \text{Tr}(M^{[1]\sigma_1} M^{[2]\sigma_2} \cdots M^{[N]\sigma_N}), \tag{3}$$

where $M^{[i]\sigma_i}$ is a rank-3 tensor on site i and it is simply a matrix when the index σ_i is fixed. The row and column of the matrix $M^{[i]\sigma_i}$ are associated with the virtual spaces $\mathcal{V}_{i-1}$ and $\mathcal{V}_i$, respectively, emerged from the decomposition. The dimension of $\mathcal{V}_i$, $\chi_i = \dim(\mathcal{V}_i)$, is the bond dimension of bond i connecting sites i and $i + 1$. Especially, bond N (also labeled as bond 0) connects sites N and 1. Without losing generality, we can always assign $\chi_0 = \chi_N = 1$ in the decomposition [23]. The corresponding MPS is usually named an *open* MPS. In the rest of the paper, we will only consider the open MPS.

B. Canonical form

Although the decomposition in Eq. (3) can be carried out in various ways, the canonical decomposition [31], known for its additional advantageous properties, is commonly employed. The canonical decomposition is represented as

$$\Psi_{\sigma_1,\ldots,\sigma_N} = \mathrm{Tr}(\Lambda^{[0]}\Gamma^{[1]\sigma_1}\Lambda^{[1]}\Gamma^{[2]\sigma_2}\cdots\Lambda^{[N-1]}\Gamma^{[N]\sigma_N}\Lambda^{[N]}), \quad (4)$$

where an additional diagonal real matrix $\Lambda^{[i]}$ is attached to bond i with $\Lambda^{[0]} = \Lambda^{[N]} \equiv (1)$ for the open MPS, and $\Gamma^{[i]\sigma_i}$ is a rank-3 tensor on site i. Here we assume that the diagonal elements of each $\Lambda^{[i]}$ are in descending order. The form $\{\Lambda^{[i]}, \Gamma^{[j]}\}$ in Eq. (4) is commonly known as the Vidal form. Additionally, the MPS is considered to be in the canonical form when $\Gamma^{[i]}$ and $\Lambda^{[i]}$ satisfy the canonical conditions:

$$A^{[i]\sigma_i} \equiv \Lambda^{[i-1]}\Gamma^{[i]\sigma_i} \quad \text{with} \quad \sum_{\sigma_i} \bar{A}^{[i]\sigma_i}A^{[i]\sigma_i} = I_{\mathcal{V}_i}$$

$$B^{[i]\sigma_i} \equiv \Gamma^{[i]\sigma_i}\Lambda^{[i]} \quad \text{with} \quad \sum_{\sigma_i} B^{[i]\sigma_i}\bar{B}^{[i]\sigma_i} = I_{\mathcal{V}_{i-1}}, \quad (5)$$

where $\bar{A}^{[i]}$ ($\bar{B}^{[i]}$) denotes the complex conjugate transpose of $A^{[i]}$ ($B^{[i]}$) and $I_{\mathcal{V}_i}$ is the identity matrix on $\mathcal{V}_i$.

Here we concisely list main properties of the canonical form. First, we can straightforwardly show that under the complete orthogonal basis

$$\left\{ |L_{\alpha_i}\rangle = \sum_{\sigma_1,\ldots,\sigma_i} \left(\Lambda^{[0]}\Gamma^{[1]\sigma_1}\cdots\Lambda^{[i-1]}\Gamma^{[i]\sigma_i}\right)_{\alpha_i} |\sigma_1\cdots\sigma_i\rangle \right\}, \quad (6)$$

the reduced density matrix $\rho^{[1\cdots i]}$ of the subsystem $\{1, 2, \ldots, i\}$ is simply represented as

$$\rho^{[1\cdots i]} = (\Lambda^{[i]})^2 \quad \text{with} \quad \sum_{\alpha_i} \left(\Lambda^{[i]}_{\alpha_i\alpha_i}\right)^2 = 1. \quad (7)$$

Note that the elements of the diagonal matrix $\Lambda^{[i]}$ correspond to the singular value spectrum $\{\lambda_{\alpha_i}\}$ obtained through the singular value decomposition (SVD) of the matrix $\Psi_{\sigma_1\cdots\sigma_i,\sigma_{i+1}\cdots\sigma_N}$, where the rows and columns are formed by grouping indices $\{\sigma_1, \ldots, \sigma_i\}$ and $\{\sigma_{i+1}, \ldots, \sigma_N\}$, respectively. Therefore, $(\Lambda^{[i]})^2$ is also the reduced density matrix of the subsystem $\{i + 1, i + 2, \ldots, N\}$.

Using these properties, the information of quantum entanglement for $|\Psi\rangle$ can be directly extracted from the canonical form. Naturally, the bond i provides a bipartition of the system into two subsystems $\{1, \ldots, i\}$ and $\{i + 1, \ldots, N\}$. Accordingly, the entanglement entropy $S_v^{[i]}$ for this bipartition is given by

$$S_v^{[i]} = -\mathrm{Tr}(\rho^{[1\cdots i]}\ln\rho^{[1\cdots i]}) = -\sum_{\alpha_i} \lambda_{\alpha_i}^2 \ln\lambda_{\alpha_i}^2, \quad (8)$$

and the entanglement spectrum level ξ_{α_i} is given by $\xi_{\alpha_i} = -\ln\lambda_{\alpha_i}^2$. Therefore, truncating the bond dimension χ_i at bond i on the basis of $\{\lambda_{\alpha_i}\}$ allows for the retention of a significant portion of the entanglement information between these two subsystems. For practical construction of the canonical form for a general quantum state and additional insights into its useful properties, one can find more comprehensive reviews [21–23].

C. TEBD algorithm

Based on the canonical form, the unitary dynamics of a quantum state can be efficiently simulated using the TEBD algorithm [31]. In general, a unitary quantum dynamic process is described by applying a unitary operator $\hat{U}$ to the state $|\Psi\rangle$. However, $\hat{U}$ can be decomposed into many two-site local unitary operators $\hat{U}_{i,i+1}^\alpha$ acting on sites i and $i + 1$, where α is the index for these local operators [32]. Therefore, we can focus on simulating $|\Psi\rangle \to \hat{U}_{i,i+1}^\alpha|\Psi\rangle$ at a specific moment. The simulation of this process using the TEBD algorithm is referred to as a TEBD procedure and will be outlined below.

A TEBD procedure consists of three steps. Starting from the canonical form of $|\Psi\rangle$, we contract the involved local tensors to construct the local two-site wave function

$$\Psi_{i,i+1} = \Lambda^{[i-1]}\Gamma^{[i]}\Lambda^{[i]}\Gamma^{[i+1]}\Lambda^{[i+1]}. \quad (9)$$

Next, we update $\Psi_{i,i+1}$ to $\Psi'_{i,i+1}$ by appending $\hat{U}_{i,i+1}^\alpha$ to $\Psi_{i,i+1}$. Subsequently, we perform the SVD of $\Psi'_{i,i+1} = U^{[i]}S\bar{V}^{[i+1]}$ and update the canonical form as

$$\Gamma^{[i]} \leftarrow (\Lambda^{[i-1]})^{-1}U^{[i]},$$

$$\Lambda^{[i]} \leftarrow S,$$

$$\Gamma^{[i+1]} \leftarrow \bar{V}^{[i+1]}(\Lambda^{[i+1]})^{-1} \quad (10)$$

with $\Lambda^{[i-1]}$ and $\Lambda^{[i+1]}$ intact.

Ideally, the TEBD procedure can be parallelized perfectly in the real-space to evaluate multiple local unitary operators $\{\hat{U}_{i,i+1}^\alpha\}$ simultaneously as long as these unitary operators do not overlap with each other, automatically maintaining the canonical form. However, the bond dimensions in general increase exponentially. The TEBD algorithm has also been applied to simulate imaginary-time evolution using an MPS [33,34] and a PEPS [35] for ground state searches, and has also become a fundamental technique in the research field of tensor network states under the name of the simple update (SU) method. Specifically, in the case where $\hat{U}_{i,i+1}^\alpha$ is an identity operator, the TEBD or SU procedure does not alter the underlying quantum state $|\Psi\rangle$ and is thus denoted as the trivial simple update (tSU) step. The repeated application of tSU steps to an MPS or a PEPS, known as the tSU algorithm, has been developed to construct its canonical form for an MPS or its quasicanonical form for a PEPS [36–38].

D. Computational complexities of MPS calculations

Before concluding this section, let us discuss the spacial and time complexities of generic MPS calculations. For an MPS with N sites and a bond dimension χ, the spacial complexity, i.e., the memory footprint, is $\mathcal{O}(N\chi^2)$, increasing polynomially with both N and χ. Here, for simplicity, we assume the same bond dimension for each matrix and maintain this convention throughout the rest of the paper. Typically, the time complexity of local MPS operations in MPS-based algorithms is $\mathcal{O}(\chi^n)$, where, for instance, $n = 3$ in the TEBD algorithm and the density-matrix renormalization group (DMRG) algorithm [39,40]. In addition, the time cost of real-space sequential algorithms for simulating a system with N sites linearly increases with N, as the *sweep procedure* is usually required in these algorithms [7,23]. Therefore, the complexities of MPS calculations are determined by the

bond dimension χ and the system size N for real-space sequential algorithms. These complexity analyses suggest that compressing an MPS to another MPS with a smaller χ while maintaining high fidelity is essential for accelerating MPS calculations. Notice that we omit the dimension d_{loc} of the local Hilbert space in the complexity analysis since it is in general small (in our case, $d_{loc} = 2$) compared with χ and N.

III. REAL-SPACE PARALLELIZABLE MPS COMPRESSION

In this section, we explore the real-space parallelizable MPS compression. After introducing the concept of MPS compression, we assess the accuracy of straightforward parallel MPS compression (i.e., truncating all the virtual bonds simultaneously) and show its comparablity with a typical sequential compression method. Then, we propose a method to stabilize the wave-function norm in parallel MPS compression and provide a proof for its upper and lower bounds. Next, we discuss the recovery of the canonical form after parallel MPS compression by additional parallel tSU steps. Combining these insights, we ultimately propose an improved parallel MPS compression method that is not only numerically more stable but also more accurate for simulations in subsequent steps.

A. MPS compression

For an MPS representing the state $|\Psi\rangle$ with bond dimension χ, we can construct another MPS

$$|\Psi_T\rangle = \sum_{\sigma_1,\ldots,\sigma_N} \mathrm{Tr}\big(M'^{[1]\sigma_1} M'^{[2]\sigma_2} \cdots M'^{[N]\sigma_N}\big)|\sigma_1 \cdots \sigma_N\rangle,$$

(11)

with bond dimension $\chi' \leqslant \chi$ to approximate $|\Psi\rangle$. Here $|\Psi_T\rangle$ represents the compressed (or truncated) MPS representation of the state $|\Psi\rangle$.

It is meaningless to compress an MPS without considering accuracy since our goal is to construct an MPS that shares the largest similarity with the original uncompressed one. The similarity of two MPSs is determined by the wave-function fidelity between $|\Psi\rangle$ and $|\Psi_T\rangle$, i.e.,

$$\mathcal{F}_2 = |\langle\Psi|\Psi_T\rangle|^2.$$

(12)

Given that $|\Psi\rangle$ and $|\Psi_T\rangle$ are normalized, $\mathcal{F}_2$ should always be in the region $0 \leqslant \mathcal{F}_2 \leqslant 1$. The accuracy of MPS compression is characterized by $\mathcal{F}_2$. For a given χ', our aim is to construct $|\Psi_T\rangle$ with the maximal $\mathcal{F}_2$. Furthermore, in this section, we assume that the MPS representing $|\Psi\rangle$ has been brought into the canonical form before being compressed.

There are two typical approaches to performing MPS compression. One involves locally and sequentially modifying each of the MPS tensors (*sequential MPS compression*). Representative algorithms in this approach include the sequential SVD truncation and the sequential local variational optimization (for details, see Ref. [23]). In this case, $\mathcal{F}_2$ in Eq. (12) can be evaluated from the singular values kept in $|\Psi_T\rangle$. Due to the sequential nature of this approach, the time complexity of these algorithms always linearly increases with the system size N. The other approach involves compressing all MPS tensors simultaneously (*parallel MPS compression*). Although

this real-space parallelizable approach has a promising constant time complexity with increasing the system size N, it has been rarely discussed in the literature [27,28,41]. In this parallel approach, it is not straightforward to build the connection between the fidelities in each local tensor compression and the global fidelity in Eq. (12).

B. Fundamental theorem on parallel MPS compression

A straightforward algorithm to perform parallel MPS compression (to the best of our knowledge, it is also the only existing algorithm) involves inserting the local projector $P_i = \sum_{\alpha=1}^{\chi'} |\alpha\rangle_{\mathcal{V}_i}{}_{\mathcal{V}_i}\langle\alpha|$, where $\{|\alpha\rangle_{\mathcal{V}_i}\}$ are orthogonal (but not complete) bases in $\mathcal{V}_i$, into each bond in Eq. (4) to truncate it to dimension χ'. The resulting compressed MPS is denoted as $|\tilde{\Psi}_T\rangle$. It is important to note that $|\tilde{\Psi}_T\rangle$ is generally not normalized, while $|\Psi\rangle$ is always normalized. Therefore, to obtain the normalized state $|\Psi_T\rangle$, additional renormalization is necessary: $|\Psi_T\rangle = |\tilde{\Psi}_T\rangle/|\langle\tilde{\Psi}_T|\tilde{\Psi}_T\rangle|^{1/2}$.

The accuracy of the above parallel MPS compression algorithm is bounded by the following fundamental theorem [22,27]:

Theorem 1.

$$||\Psi\rangle - |\tilde{\Psi}_T\rangle|^2 \leqslant 2\epsilon(\chi')$$

(13)

and

$$\mathcal{F}_2 \geqslant 1 - 2\epsilon(\chi'),$$

(14)

where $\epsilon(\chi') = \sum_{i=1}^{N-1} \epsilon_i(\chi')$ is defined as the global truncation error, and $\epsilon_i(\chi') = \sum_{\alpha=\chi'+1}^{\chi} (\Lambda_{\alpha\alpha}^{[i]})^2$ is the local truncation error associated with bond i. An illustration of Eq. (13) is shown in Fig. 1.

Combining Eq. (14) with the fact that $0 \leqslant \epsilon_i \leqslant 1$, we can derive a looser bound for $\mathcal{F}_2$ as

$$\mathcal{F}_2 \geqslant 1 - 2\sum_{i=1}^{N-1} \sqrt{\epsilon_i(\chi')}.$$

(15)

Note that the right-hand side of Eq. (15) can be negative, while the wave-function fidelity $\mathcal{F}_2$ must be positive. The bound in Eq. (15) has the same formulation as the exact fidelity lower bound for the sequential SVD truncation formulated in Ref. [7]. Although $\epsilon_i(\chi_i)$ may slightly deviates from the corresponding local truncation error in the sequential SVD compression, we argue that the parallel MPS compression algorithm could have the same order of accuracy as the sequential SVD compression algorithm. This argument will be numerically confirmed in Sec. IV B 1. Also notice that, in general, the canonical form is broken after a parallel MPS compression, which could impact on the subsequent simulation steps.

C. Wave-function norm stabilization in parallel MPS compression

Although the norm of the wave function does not impact the calculation of any physical observables, it strongly influences numerical stability in actual numerical simulations, as it may vanish during the calculation process. Hence, we have to carefully consider the possible norm deviation induced by the parallel MPS compression.
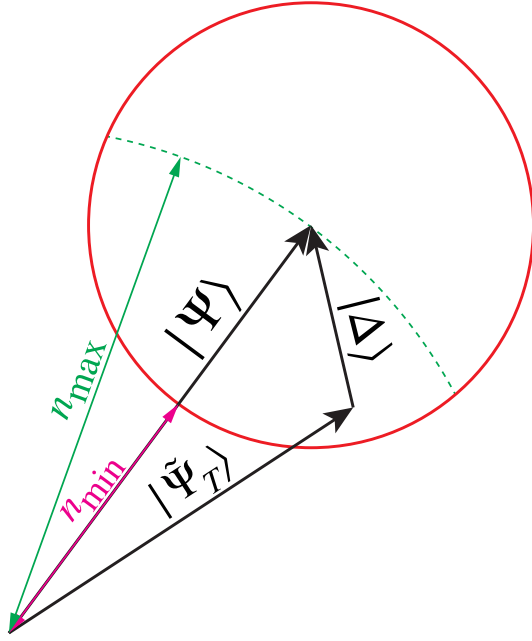
FIG. 1. Schematic plot illustrating the relation between the untruncated state $|\Psi\rangle$ and the state after the parallel MPS compression, $|\tilde{\Psi}_T\rangle$, defined in Sec. III B. The red circle with a radius of $\sqrt{2\epsilon(\chi')}$ indicates the boundary of $|\tilde{\Psi}_T\rangle$ guaranteed by Theorem 1. $n_{\mathrm{max(min)}}$ represents the maximum (minimum) of the wave-function norm of $|\tilde{\Psi}_T\rangle$.

1. Decay of the wave-function norm after parallel MPS compression

To evaluate the wave-function norm deviation after the parallel MPS compression, we first derive its bounds in Lemma 1. Its proof is based on Theorem 1 and the properties of the canonical form.

Lemma 1. The wave-function norm $n = ||\tilde{\Psi}_T\rangle|$ after the parallel MPS compression is bounded by

$$1 - \sqrt{2\epsilon(\chi')} \leqslant n \leqslant 1. \tag{16}$$

Proof. According to the triangle inequality, we have $1 - ||\Delta\rangle| \leqslant n$, where $|\Delta\rangle = |\Psi\rangle - |\tilde{\Psi}_T\rangle$, as schematically illustrated in Fig. 1. Then, we can derive the lower bound in Eq. (16) by using Theorem 1 as

$$1 - \sqrt{2\epsilon(\chi')} \leqslant 1 - ||\Delta\rangle| \leqslant n. \tag{17}$$

As for the upper bound in Eq. (16), from the direct calculation, we have

$$n^2 = \sum_{\sigma_1,...,\sigma_N} \mathrm{Tr}\big(A^{[1]\sigma_1}P_1 \cdots P_{N-2}A^{[N-1]\sigma_{N-1}}P_{N-1}A^{[N]\sigma_N}$$
$$\times \bar{A}^{[N]\sigma_N}P_{N-1}\bar{A}^{[N-1]\sigma_{N-1}}P_{N-2}\cdots P_1\bar{A}^{[1]\sigma_1}\big). \tag{18}$$

Here we adopt the left canonical form described in Eq. (5). By defining the quantum map

$$\mathcal{E}^{[i]}(X) = \sum_{\sigma_i} A^{[i]\sigma_i}X\bar{A}^{[i]\sigma_i}, \tag{19}$$

where X is a $(\mathcal{V}_i \times \mathcal{V}_i)$ matrix, Eq. (18) can be reformulated as

$$n^2 = \mathrm{Tr}[\mathcal{E}^{[1]}(P_1 \cdots \mathcal{E}^{[N-2]}(P_{N-2}$$
$$\times \mathcal{E}^{[N-1]}(P_{N-1}\rho^{[N]}P_{N-1})P_{N-2})\cdots P_1)] \tag{20}$$

and $\rho^{[N]} \equiv \sum_{\sigma_N} A^{[N]\sigma_N}\bar{A}^{[N]\sigma_N}$ is obviously a positive matrix. Introducing the following recursive relation:

$$Y^{[i]} = \mathcal{E}^{[i]}(P_i Y^{[i+1]} P_i) \quad \text{with} \quad Y^{[N]} = \rho^{[N]} \tag{21}$$

for $i = 1, 2, \ldots, N-1$, Eq. (20) can be simply given as $n^2 = \mathrm{Tr}(Y^{[1]})$.

Next, we define another map

$$\mathcal{P}_i(X) = P_i X P_i. \tag{22}$$

Because $\mathcal{E}^{[i]}(X)$ is a completely positive trace preserving map [42] and $\mathcal{P}_i(X)$ is a positive map, $Y^{[i]}$ is a positive matrix. We can also readily show that

$$\mathrm{Tr}(X) = \mathrm{Tr}(\mathcal{P}_i(X)) + \mathrm{Tr}(\bar{\mathcal{P}}_i(X)), \tag{23}$$

where $\bar{\mathcal{P}}_i(X) = (I - P_i)X(I - P_i)$ is also a positive map. Since $\mathrm{Tr}(\bar{\mathcal{P}}_i(X)) \geqslant 0$ for a positive matrix X, $\mathrm{Tr}(\mathcal{P}_i(X)) \leqslant \mathrm{Tr}(X)$. Noticing that $Y^{[i]}$ is a positive matrix, we can finally obtain the following inequality:

$$\mathrm{Tr}(Y^{[i]}) \leqslant \mathrm{Tr}(Y^{[i+1]}). \tag{24}$$

Using this inequality, we can derive the upper bound in Eq. (16) as

$$n = \sqrt{\mathrm{Tr}(Y^{[1]})} \leqslant \sqrt{\mathrm{Tr}(Y^{[N]})} = 1. \tag{25}$$

∎

Lemma 1 shows that the wave-function norm necessarily decreases, except for the ideal case, i.e., without truncating any bonds, after a parallel MPS compression process. This implies that if we alternatively perform TEBD procedures and parallel MPS compression processes, as in the case of Ref. [28], then the wave-function norm will monotonically decay, ultimately leading to serious numerical instability. This norm vanishing will be numerically demonstrated later in Sec. IV B 2. To avoid the numerical instability, naively, we can renormalize the wave function after a parallel MPS compression process. However, the time cost of calculating the wave-function norm increases linearly with the system size N, offsetting the benefits of parallel MPS compression. Therefore, it is highly desired to develop an efficient method that can suppress the norm vanishing during repeated parallel MPS compression processes without any sequential procedure.

2. Parallelizable wave-function norm stabilization

To stabilize the wave-function norm, a straightforward possibility is to locally multiply $\Lambda^{[i]}$ by a factor $\nu_i = [1 - \epsilon_i(\chi')]^{-1/2}$ after a parallel MPS compression process. Consequently, the wave-function norm changes to

$$n^* = n \prod_{i=1}^{N-1} \nu_i. \tag{26}$$

Now we shall prove that n^*, denoted as stabilized norm, is bounded in an interval that includes 1, and the two boundaries of this interval uniformly converge to 1 as $\epsilon(\chi')$ approaches 0.

Theorem 2. The stabilized norm n^* is bounded by

$$n^*_{\text{lower}} = (1 - \sqrt{2\epsilon(\chi')}) \prod_{i=1}^{N-1} \nu_i, \tag{27}$$

$$n^*_{\text{upper}} = \prod_{i=1}^{N-1} \nu_i, \tag{28}$$

with $n^*_{\text{lower}} \leqslant 1$ and $n^*_{\text{upper}} \geqslant 1$. Moreover, they uniformly converge to 1 as $\epsilon(\chi') \to 0$.

Proof. Equations (27) and (28) follow directly from Eq. (16) in Lemma 1. Let us now prove that these quantities satisfy $n^*_{\text{lower}} \leqslant 1$ and $n^*_{\text{upper}} \geqslant 1$. Noticing that $\epsilon(\chi') \geqslant 0$ and $1 - 2\epsilon(\chi') = (1 + \sqrt{2\epsilon(\chi')})(1 - \sqrt{2\epsilon(\chi')})$, we have the following inequalities:

$$1 - \sqrt{2\epsilon(\chi')} \leqslant 1 - 2\epsilon(\chi') \leqslant 1 - \epsilon(\chi'), \tag{29}$$

when $2\epsilon(\chi') \leqslant 1$. On the other hand, since $0 \leqslant \epsilon_i(\chi') \leqslant 1$, we can show that

$$\prod_{i=1}^{N-1} \nu_i^{-1} \geqslant \prod_{i=1}^{N-1} [1 - \epsilon_i(\chi')] \geqslant 1 - \epsilon(\chi'). \tag{30}$$

Combine Eqs. (29) and (30), we can obtain that $1 - \sqrt{2\epsilon(\chi')} \leqslant \prod_{i=1}^{N-1} \nu_i^{-1}$ and thus $n^*_{\text{lower}} = (1 - \sqrt{2\epsilon(\chi')}) \prod_{i=1}^{N-1} \nu_i \leqslant 1$. This is trivial when $2\epsilon(\chi') \geqslant 1$. It is also straightforward to show that $n^*_{\text{upper}} \geqslant 1$ because $\epsilon_i(\chi') \leqslant 1$.

Since both quantities $1 - \sqrt{2\epsilon(\chi')}$ and $\prod_{i=1}^{N-1} \nu_i$ converge to 1 as $\epsilon(\chi') \to 0$, the two boundaries n^*_{lower} and n^*_{upper} converge to 1 from both sides. ∎

Theorem 2 implies that the local renormalization method can indeed effectively mitigate the decay of the wave-function norm during the parallel MPS compression. While the proof provided above assumes that the uncompressed MPS is in its canonical form, we argue that this wave-function norm stabilization method remains applicable even when the uncompressed MPS deviates slightly from the canonical form. The numerical demonstration supporting this assertion will be presented in Sec. IV B 2.

D. Parallel recovery of the canonical form through parallel tSU

Because of its many advantages, almost all MPS-based algorithms developed previously employ the canonical form. Therefore, it is important to maintain an MPS in the canonical form or close to the canonical form also in a parallel MPS algorithm. In this respect, it is well known that the right-left sweeps of tSU, which is a sequential procedure, can converge an arbitrary MPS to its canonical form. However, how the canonical form is recovered with a parallel procedure has been rarely investigated thus far. In this section, we first prove that at most $\frac{N}{2}$ ($\frac{N-1}{2}$) parallel tSU steps can converge an arbitrary MPS to its canonical form when N is even (odd), and then we discuss how it approaches the canonical form with fewer operations.

Similarly to the TEBD algorithm, which is exactly parallelizable in the case of no truncation [31], we can define a single parallel trivial simple update (PtSU) step as applying tSU on all the odd MPS bonds and then on all the even MPS bonds. First, we prove the convergence of an MPS to its canonical form through PtSU steps with the following theorem.

Theorem 3. For an MPS with N sites, at most $\frac{N}{2}$ ($\frac{N-1}{2}$) PtSU steps are required to convert it to its canonical form when N is even (odd).

Proof. Before presenting the proof, let us first introduce an equivalent definition of the canonical form. We define the following left tensor $L^{[i]}$ and right tensor $R^{[i]}$ for bond i, respectively:

$$L^{[i]\sigma_1\cdots\sigma_i} \equiv \Lambda^{[0]}\Gamma^{[1]\sigma_1}\cdots\Lambda^{[i-1]}\Gamma^{[i]\sigma_i}, \tag{31}$$

$$R^{[i]\sigma_{i+1}\cdots\sigma_N} \equiv \Gamma^{[i+1]\sigma_{i+1}}\Lambda^{[i+1]}\cdots\Gamma^{[N]\sigma_N}\Lambda^{[N]}. \tag{32}$$

Then, asserting that an MPS composed of $\{\Gamma^{[i]\sigma_i}, \Lambda^{[j]}\}$ is in the canonical form, satisfying the conditions in Eq. (5), is equivalent to saying that $L^{[i]\sigma_1\cdots\sigma_i}$ and $R^{[i]\sigma_{i+1}\cdots\sigma_N}$ satisfy

$$\sum_{\sigma_1,\ldots,\sigma_i} \bar{L}^{[i]\sigma_1\cdots\sigma_i} L^{[i]\sigma_1\cdots\sigma_i} = I_{\mathcal{V}_i}, \tag{33}$$

$$\sum_{\sigma_{i+1},\ldots,\sigma_N} R^{[i]\sigma_{i+1}\cdots\sigma_N} \bar{R}^{[i]\sigma_{i+1}\cdots\sigma_N} = I_{\mathcal{V}_i}, \tag{34}$$

for $\forall i \in [0, N]$, where $L^{[0]} \equiv (1)$ and $R^{[N]} \equiv (1)$. The proof of this equivalence follows from straightforward calculations.

Using this equivalent definition of the canonical form, we can prove the theorem inductively. As the starting point, $\bar{L}^{[0]}L^{[0]} = (1) = I_{\mathcal{V}_0}$ is trivially satisfied. Then, if we assume that $\sum_{\sigma_1,\ldots,\sigma_{i-1}} \bar{L}^{[i-1]\sigma_1\cdots\sigma_{i-1}} L^{[i-1]\sigma_1\cdots\sigma_{i-1}} = I_{\mathcal{V}_{i-1}}$ is satisfied, then one additional tSU on sites i and $i+1$ can realize $\sum_{\sigma_1,\ldots,\sigma_i} \bar{L}^{[i]\sigma_1\cdots\sigma_i} L^{[i]\sigma_1\cdots\sigma_i} = I_{\mathcal{V}_i}$. This is simply because, after this tSU, $\Lambda^{[i-1]}\Gamma^{[i]\sigma_i} = A^{[i]\sigma_i}$ satisfies $\sum_{\sigma_i} \bar{A}^{[i]\sigma_i}A^{[i]\sigma_i} = I_{\mathcal{V}_i}$ and thus

$$\sum_{\sigma_1,\ldots,\sigma_i} \bar{L}^{[i]\sigma_1\cdots\sigma_i} L^{[i]\sigma_1\cdots\sigma_i}$$
$$= \sum_{\sigma_1,\ldots,\sigma_i} \bar{A}^{[i]\sigma_i} \bar{L}^{[i-1]\sigma_1\cdots\sigma_{i-1}} L^{[i-1]\sigma_1\cdots\sigma_{i-1}} A^{[i]\sigma_i} = I_{\mathcal{V}_i}. \tag{35}$$

At the same time, by definition, any tSU involving sites in $\{1, \ldots, i\}$ does not change the left tensors $\{L^{[k]\sigma_1\cdots\sigma_k}\}_{k=0}^{i-1}$. Therefore, $\bar{L}^{[N-1]}L^{[N-1]} = I_{\mathcal{V}_{N-1}}$ is established after $\frac{N}{2}$ ($\frac{N-1}{2}$) PtSU steps for even (odd) N. Note that Eq. (33) is satisfied for $i = N$ because of the wave-function normalization condition.

Similarly, we can prove that the same PtSU steps also establish Eq. (34) for $\forall i \in [1, N-1]$. For $i = 0$, Eq. (34) is automatically satisfied because of the wave-function normalization condition. Finally, since this proof is inductive, it provides an upper bound on how many PtSU steps are required to restore the canonical form. ∎

Theorem 3 shows that the PtSU, a real-space parallelizable algorithm, can recover the canonical form with only locally operating tensors, implying the potential of its utilization on modern supercomputers where communication between neighboring nodes is highly optimized [43]. Moreover, we can simply skip applying the identity gate in the tSU procedure to further optimize the performance in practice.

Now, we explore the behavior of approaching the canonical form by applying the PtSU. To measure how an MPS approaches its canonical form with PtSU steps, we first define
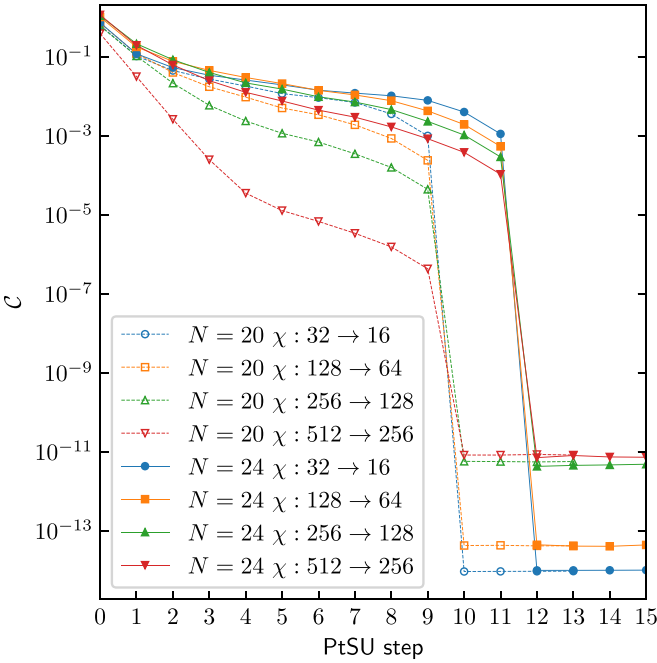
FIG. 2. Semilog plot of $\mathcal{C}$ as a function of PtSU steps for $N = 20$ and 24. The initial MPS is prepared by simultaneously truncating all the virtual bonds of a canonicalized random MPS to half of the bond dimension and manually normalizing the norm of the resulting MPS to 1. The results are averaged over 100 initial MPSs with different sets of random parameters.

the following distance to the canonical form:

$$\mathcal{C} = \frac{1}{2N} \sum_{i=1}^{N} \left\| \bar{A}^{[i]} A^{[i]} - I_{\mathcal{V}_i} \right\| + \left\| B^{[i]} \bar{B}^{[i]} - I_{\mathcal{V}_{i-1}} \right\|, \quad (36)$$

where $|| \cdot ||$ represents the Frobenius norm of the matrix. This quantity is zero if and only if an MPS is in the canonical form. Note that, in Eq. (36), we assume that the norm of the MPS is 1. Notice that a similar quantity has been proposed in a recent study to examine the regauging of general tensor networks, where the nuclear norm is considered [44].

We investigate how $\mathcal{C}$ changes with the application of PtSU steps. To closely mimic the actual calculation task, we prepare the initial MPS as follows: We first construct a canonicalized random MPS with the bond dimension χ. Each tensor element in this MPS is initialized to a complex number $z = a + ib$ with a and b chosen randomly from $[-1, 1]$, and then its canonical form is constructed with the norm properly normalized to 1. Subsequently, we truncate all the virtual bonds to $\chi/2$, i.e., disregarding the $\chi/2$ smallest elements in each $\Lambda^{[i]}$, and renormalize the wave-function norm to 1. We then perform PtSU steps on this MPS and calculate $\mathcal{C}$ after each PtSU step. The results are summarized in Fig. 2. Here we choose $N = 20$ and 24 and examine the cases of $\chi = 32, 128, 256,$ and 512.

In Fig. 2, we observe that $\mathcal{C}$ rapidly and monotonically decreases with PtSU steps, especially at the beginning of several PtSU steps. Additionally, the canonical form is attained more rapidly when the bond dimension is larger. This implies that we can drive the MPS closer to the canonical form by appending a few PtSU steps after the parallel MPS compression.

Also notice that the canonical form is exactly restored after $N/2$ PtSU steps, justifying Theorem 3.

E. Improved parallel MPS compression

Inspired by the new insights from Theorems 2 and 3, we propose the IPMC method: starting from an MPS in the Vidal form, which is usually the resulting MPS of the previous TEBD evolution and has the bond dimension $d_{\text{loc}} \chi$ (d_{loc} is the dimension of the local Hilbert space, i.e., $d_{\text{loc}} = 2$ for the qubit case), we perform a parallel MPS compression process (discussed in Sec. III B) followed by a norm stabilization step (discussed in Sec. III C 2) to compress the bond dimension back to χ. Then, we perform g PtSU steps to partially recover the canonical form. This parallel MPS compression method is fully real-space parallelizable, requiring only communication between neighboring sites [also see Fig. 3(d)]. In the next section, we will apply this method to simulate unitary quantum dynamics and examine its efficiency through extensive numerical experiments.

IV. APPLICATION TO THE SIMULATION OF UNITARY QUANTUM DYNAMICS

The time evolution operator $\hat{\mathcal{U}}(t)$, which describes the unitary quantum dynamic process of evolving a state from time 0 to t, $|\Psi(t)\rangle = \hat{\mathcal{U}}(t)|\Psi(0)\rangle$, can always be decomposed into a quantum circuit composed of layers of local unitary operators

$$\hat{\mathcal{U}}(t) = \prod_{d=1}^{D} \prod_{i=1}^{m(d)} \hat{U}_i^d, \quad (37)$$

where $\hat{U}_i^d$ is the ith local unitary operator in the dth layer, $m(d)$ is the number of local unitary operators in the dth layer, and D is the total number of layers (i.e., depth) of the quantum circuit [32]. Notice that there exist other simulation approaches when $\hat{\mathcal{U}}(t)$ is associated with a Hamiltonian [29,41].

The above decomposition is, in general, approximate and, moreover, it is not unique. Therefore, for the convenience of applying MPS-based algorithms, one usually constructs $\{\hat{U}_i^d\}$ only involving adjacent qubits in a 1D path going through all the qubits in the system [see Fig. 3(b) as an example and also Appendix A]. Therefore, in the rest of this section, we discuss the simulations of unitary quantum dynamics by considering their 1D quantum circuit representations.

A. Parallel time-evolving block decimation algorithm

A unitary quantum dynamic process represented by a 1D quantum circuit with N qubits can be efficiently simulated using a sequential TEBD-type [7] or time-dependent DMRG-type [25] algorithm. The time cost of these two algorithms is always proportional to N. Here, incorporating the IPMC method introduced above, we propose a pTEBD algorithm that can reduce the time complexity to a constant with increasing N.

The pTEBD algorithm along with the IPMC is illustrated in Fig. 3. The initial state $|\Psi(0)\rangle$ can be represented as a canonicalized MPS with bond dimension χ_0 [see Fig. 3(a)]. To simulate quantum circuits for a circuit-based quantum computer, generally one chooses $|\Psi(0)\rangle = |00 \cdots 0\rangle$ with $\chi_0 = 1$. In the pTEBD algorithm, to simulate one layer of gates represented
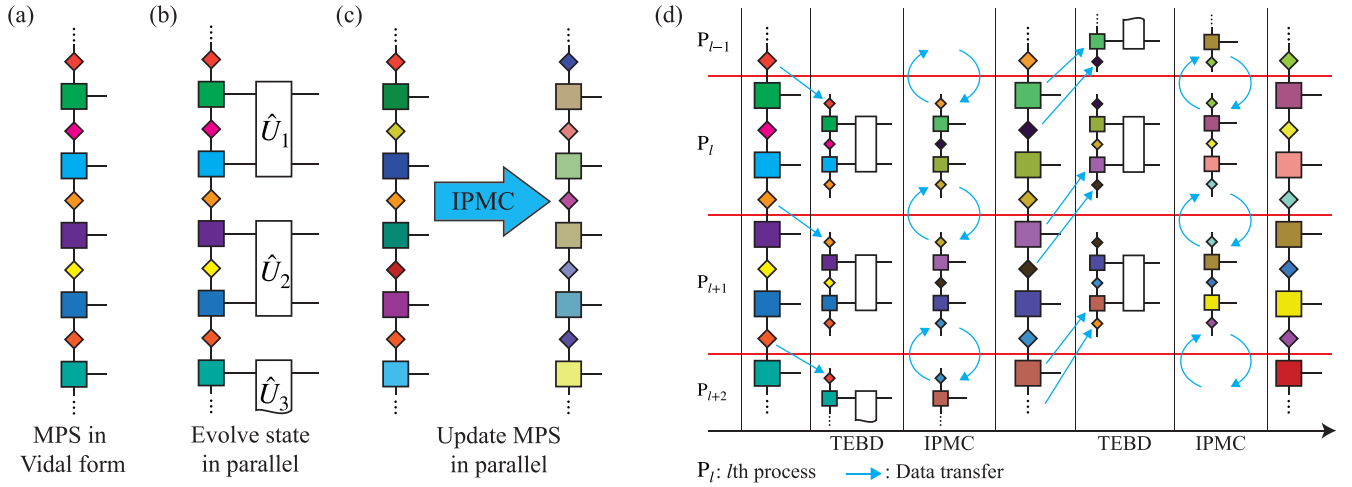
FIG. 3. Illustration of the pTEBD algorithm for a quantum circuit simulation. Starting from an MPS in (a), which is allowed to be slightly deviated from the canonical form, the state is evolved using the standard TEBD procedure but with all gates in the same layer treated parallelly in (b), and then truncate the state using the IPMC (see Sec. III E) in (c) to update the MPS, which is subsequently used to simulate the quantum circuit with the next layer of gates. A possible realization of the distribution of data to each MPI process P_l and the data transfer among MPI processes in the pTEBD algorithm is schematically shown in (d). As multiple PtSU steps may be executed in the IPMC, the data transfer is consequently required several times, as indicated here simply by two arrows pointing forward and backward between neighboring MPI processes. Note also that single-qubit gates can be treated exactly without increasing the bond dimension by any MPS-based simulations, including the pTEBD algorithm.

by $\prod_i^{m(d)} \hat{U}_i^d$ in Eq. (37), the standard TEBD procedure is utilized to simulate each gate operation in parallel [see Fig. 3(b)]. The bond dimension of the resulting MPS increases to $\tilde{\chi}$, which can be larger than the maximum bond dimension χ predetermined in the algorithm. Then, the IPMC is performed to truncate the increased bond dimension back to χ [see Fig. 3(c)] and it returns to the first step to simulate the next layer of gates in the quantum circuit. As indicated in Fig. 3(d), the data transfer in the pTEBD algorithm is strictly local.

After completing the simulation of all D layers of gates in the quantum circuit, the resulting MPS approximates the final state of the unitary quantum dynamic process $|\Psi(t)\rangle$. We can then use standard MPS techniques to exactly evaluate the expectation values of observables [23] or perfectly sample bit strings from this state [45] to mimic quantum devices after restoring the canonical form of the obtained MPS representing $|\Psi(t)\rangle$. The flow chart of the pTEBD algorithm is provided in Algorithm 1.

ALGORITHM 1. pTEBD.

Input: $|\Psi(d=0)\rangle = \{\Gamma^{[p]}, \Lambda^{[q]}\}, \{\hat{U}_i^d\}, \chi, g$
for $m = 1, \ldots, D$ **do**
    $\{\tilde{\Gamma}^{[p]}, \tilde{\Lambda}^{[q]}\} \leftarrow$ **TEBD**$(\{\Gamma^{[p]}, \Lambda^{[q]}\}, \{\hat{U}_i^m\})$ ▷ In parallel
    if $\tilde{\chi} > \chi$ **then**
        $\{\Gamma^{[p]}, \Lambda^{[q]}\} \leftarrow$ **IPMC**$(\{\tilde{\Gamma}^{[p]}, \tilde{\Lambda}^{[q]}\}, \chi, g)$
    else
        $\{\Gamma^{[p]}, \Lambda^{[q]}\} \leftarrow \{\tilde{\Gamma}^{[p]}, \tilde{\Lambda}^{[q]}\}$
    end if
end for
$\{\Gamma^{[p]}, \Lambda^{[q]}\} \leftarrow$ **Canonicalize**$(\{\Gamma^{[p]}, \Lambda^{[q]}\})$
$|\Psi(d=D)\rangle \leftarrow \{\Gamma^{[p]}, \Lambda^{[q]}\}$
Do measurements or sampling on $|\Psi(d=D)\rangle$

B. Accuracy, numerical stability, and performance of pTEBD

To comprehensively benchmark the accuracy, numerical stability, and performance of the pTEBD algorithm, we perform intensive simulations of typical 1D and 2D quantum circuits and compare the results with those obtained by sequential MPS simulations. For this purpose, we consider both random quantum circuits (RQCs) on 1D and 2D qubit arrays (RQC-1D and RQC-2D, respectively) and parametrized quantum circuits (PQCs), which are widely adopted in the variational quantum eigensolver [46], on 1D and 2D qubit arrays (PQC-1D and PQC-2D, respectively). These four types of quantum circuits cover representative cases, providing concrete demonstrations for the practical feasibility of the pTEBD algorithm. The details of each circuit are described in Appendix A. Note also that in the sequential algorithm described in Ref. [7], which we employ as the benchmark algorithm, referred to as the *sequential MPS algorithm*, in the following numerical experiments, an MPS is always in the canonical form and normalized to one, which is generally not the case in the pTEBD algorithm. We also provide an additional benchmark test of the pTEBD algorithm applied to simulate the quantum Fourier transformation algorithm in Appendix C.

1. Accuracy

First, we assess the simulation accuracy of the pTEBD algorithm and demonstrate its ability to achieve precision comparable to that of the sequential MPS algorithm. Here the simulation precision is determined by the wave-function fidelity

$$\mathcal{F} = |\langle \Psi_{\text{exact}} | \Psi(\chi) \rangle|^2, \tag{38}$$

where $|\Psi_{\text{exact}}\rangle$ represents the state obtained by the exact simulation and $|\Psi(\chi)\rangle$ is the state obtained using an MPS with
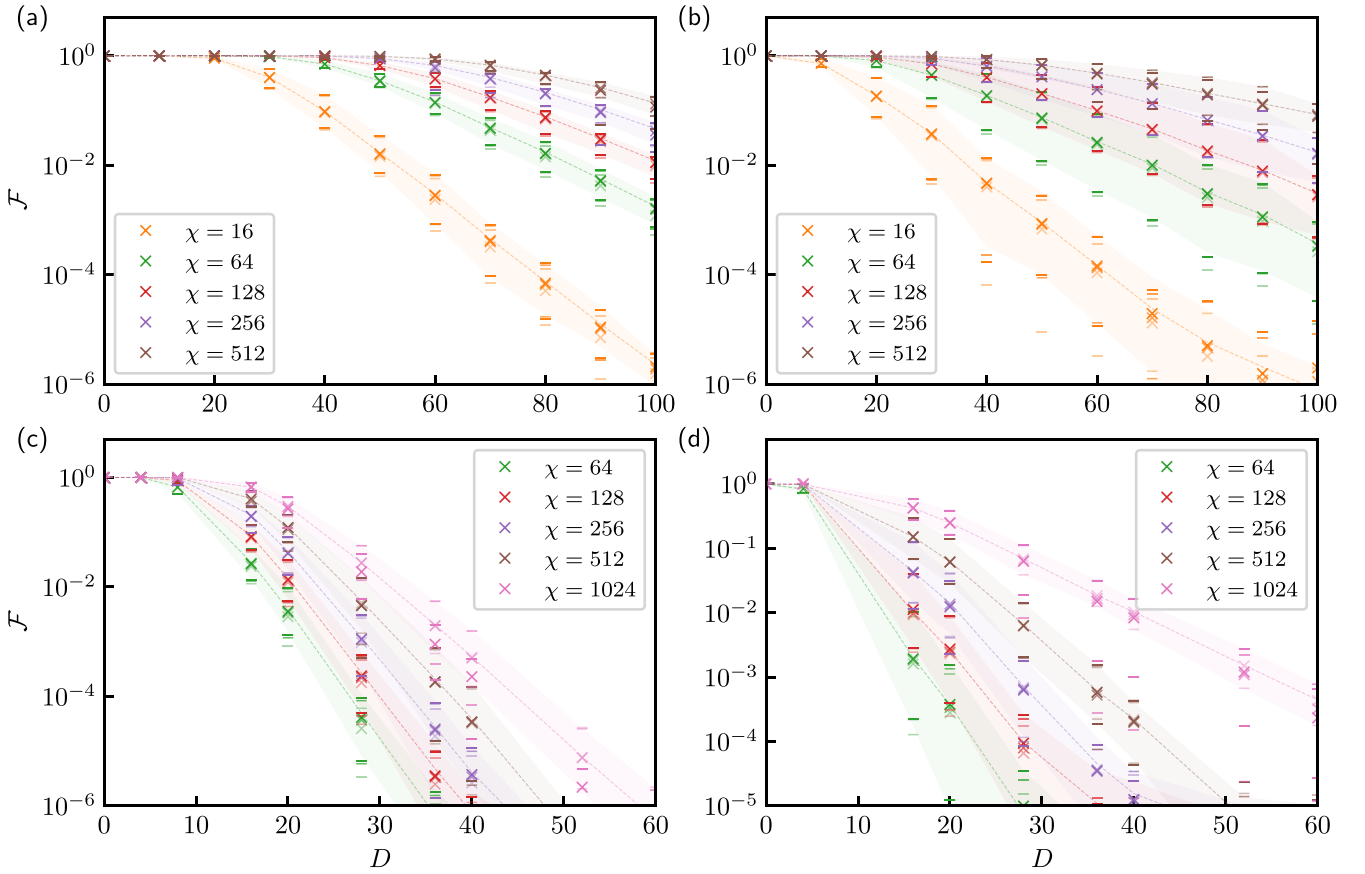
FIG. 4. Wave-function fidelity $\mathcal{F}$ as a function of physical circuit depth D (for the definition, see Appendix A) obtained using the sequential MPS algorithm and the pTEBD algorithm with a fixed MPS bond dimension χ: (a) RQC-1D with $N = 25$, (b) PQC-1D with $N = 24$, (c) RQC-2D with $L_x = 5$ and $L_y = 5$ (i.e., $N = 25$), and (d) PQC-2D with $L_x = 4$ and $L_y = 6$ (i.e., $N = 24$). The dashed lines (crosses) represent the average fidelity $\bar{\mathcal{F}}_{\mathrm{MPS(pTEBD)}}$ over 10 simulations of the same quantum circuit but with different sets of random parameters obtained using the sequential MPS (pTEBD) algorithm. The shades (bars) indicate the minimal and maximal fidelities among these 10 simulations using the sequential MPS (pTEBD) algorithm. The color intensities of crosses and bars represent the pTEBD results with $g = 0, 1$, and 2 (from light to dark).

the fixed bond dimension χ. We perform the simulations for quantum circuits with up to 100 (60) physical circuit layers, i.e., before recompiling a quantum circuit to fit an MPS 1D path, in the 1D (2D) cases. The definitions of the *physical* and *compiled* circuit depth can be found in Appendix A. Each simulation is repeated 10 times with different sets of random parameters to evaluate the average fidelity, denoted as $\bar{\mathcal{F}}_{\mathrm{MPS(pTEBD)}}$ in the sequential MPS (pTEBD) simulations. These results are summarized in Fig. 4.

The simulations for all these four types of quantum circuits show consistent results: for a given bond dimension χ, the accuracy of the pTEBD simulation is comparable to the sequential MPS simulation. In most cases, we find that $\bar{\mathcal{F}}_{\mathrm{pTEBD}} \approx \bar{\mathcal{F}}_{\mathrm{MPS}}$ and $\bar{\mathcal{F}}_{\mathrm{pTEBD}}$ improves with an increase in the number g of PtSU steps, while we observe that the PtSU steps have the opposite effect in some rare cases. Moreover, in cases with relatively small χ (for example, $\chi = 16$ and 64), we notice that $\bar{\mathcal{F}}_{\mathrm{pTEBD}}$ significantly increases with applying more PtSU steps. Since $\mathcal{F}$ is highly sensitive to the random parameters used in each quantum gate in the quantum circuit, we also shown its minimum and maximum among the 10 simulations with different sets of random parameters.

These extremes obtained in the sequential MPS simulations and the pTEBD simulations are consistent with each other. In some cases [see Fig. 4(b)], the maximum value obtained in the pTEBD simulation can be larger than that obtained in the sequential MPS simulation. To present more clearly the fidelities obtained using these two algorithms, we also show the results for a quantum circuit with the same single set of random parameters in Appendix B.

For the unitary quantum dynamics described by a quantum circuit with local gates, the quantum entanglement propagates only within a finite range of space after applying each single layer of gates. Therefore, in the pTEBD algorithm, $\Lambda^{[i-1]}$ and $\Lambda^{[i+1]}$ are still expected to be a good approximation for the environments of each local two sites i and $i + 1$, although the canonical form deviates globally in general. This is probably the reason for the high precision of the pTEBD algorithm and the improvement with additional PtSU steps that suppress the deviation of the canonical form.

2. Numerical stability

After establishing the accuracy of the pTEBD algorithm, let us now demonstrate the crucial role of the wave-function
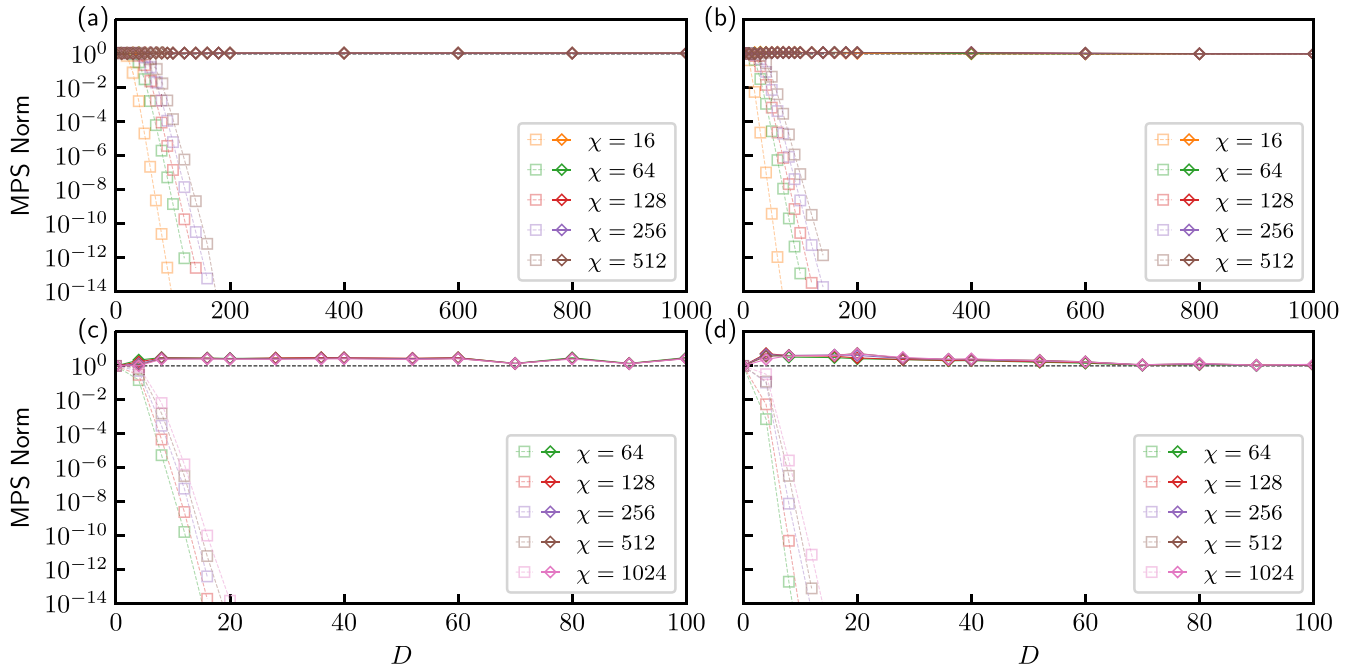
FIG. 5. Wave-function norm of the MPS as a function of physical circuit depth D (for the definition, see Appendix A) obtained using the pTEBD algorithm with a fixed MPS bond dimension χ: (a) RQC-1D with $N = 101$, (b) PQC-1D with $N = 100$, (c) RQC-2D with $L_x = L_y = 12$ (i.e., $N = 144$), and (d) PQC-2D with $L_x = L_y = 12$ (i.e., $N = 144$). The open diamonds (squares) represent the averaged results of the wave-function norm n^* (n) over 10 simulations of the same quantum circuit but with different sets of random parameters, evaluated with (without) the wave-function norm stabilization procedure. Black dashed lines indicate the norm equal to 1. Here we set $g = 0$, i.e., no PtSU since it does not affect the wave-function norm. Note that the results obtained using the pTEBD algorithm with the wave-function norm stabilization procedure are almost completely overlapped to each other in this scale.

norm stabilization procedure introduced in Sec. III C 2 in maintaining numerical stability during pTEBD simulations. To this end, we systematically evaluate the wave-function norm in the simulations of 1D (2D) quantum circuits, containing up to 1000 (100) physical circuit layers, with and without the wave-function norm stabilization procedure. The results are summarized in Fig. 5.

In cases without the wave-function norm stabilization procedure (see open squares in Fig. 5), the wave-function norm monotonically and exponentially decays with increasing D. While it slightly increases with the bond dimension χ, a tiny value, as small as 10^{-14}, is reached after simulating a quantum circuit with around 200 and 20 physical circuit layers both in 1D and 2D cases, respectively. This indicates that a renormalization procedure, involving sequential calculations and thus breaking the real-space parallelism, must be employed to prevent the exponentially decaying wave-function norm and stabilize the simulation.

In sharp contrast, in cases with the wave-function norm stabilization procedure (see open diamonds in Fig. 5), the wave-function norm consistently remains close to one, resisting decay with increasing D, even when χ is small. This demonstrates that, in the simulation of unitary quantum dynamics, the numerical stability of the pTEBD algorithm can be maintained by the IPMC, particularly the wave-function norm stabilization procedure, even though the MPS deviates from its canonical form. Hence, no sequential procedure is required throughout the entire pTEBD simulation, implying

the achievement of scalability, i.e., perfect weak scaling, as will be examined in the next section.

3. Performance and week scaling

Finally, let us assess the performance of the pTEBD algorithm from two perspectives. On the one hand, we examine the simulation elapsed time per circuit layer, which is defined as the total simulation elapsed time divided by *compiled* circuit depth (for more details, see Appendix A), versus the system size N, evaluating its performance in weak scaling. We always distribute tensors of each four adjacent sites in an MPS on one computational node. When N is not divisible by 4, the remaining tensors are distributed on the last node. On the other hand, we compare the elapsed time required to achieve a given simulation precision, measured by wave-function fidelity $\mathcal{F}$, in both sequential MPS and pTEBD simulations.

As illustrated in Fig. 6, for a fixed bond dimension χ, the pTEBD algorithm consistently exhibits nearly constant simulation times for quantum circuits with various system sizes, achieving excellent weak scaling performance. This is attributed to the absence of any real-space sequential procedures and the lack of a need for global data communication in the pTEBD algorithm [also see Fig. 3(d)]. In contrast, the elapsed time per circuit layer in the sequential MPS algorithm increases linearly with the system size due to its unavoidable sweep procedure. We also expect good strong scaling since, compared with the previous approach
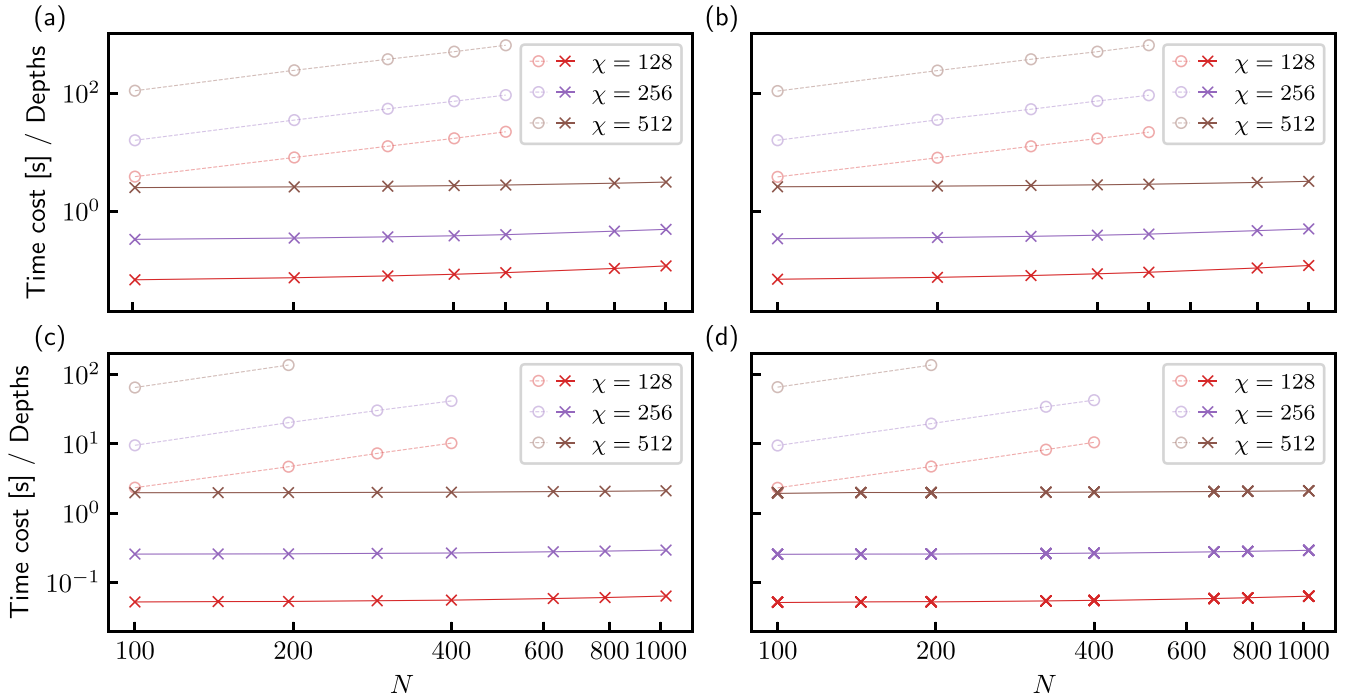
FIG. 6. Elapsed time per circuit layer (averaged over 10 simulations with different sets of random parameters) versus the system size N ($L_x = L_y$ in 2D cases) obtained using the sequential MPS algorithm (open circles) and the pTEBD algorithm (crosses) with a fixed MPS bond dimension χ: (a) RQC-1D with 300 physical circuit layers, (b) PQC-1D with 300 physical circuit layers, (c) RQC-2D with 100 physical circuit layers, and (d) PQC-2D with 100 physical circuit layers. Note that the number of circuit layers in (c) and (d) is the total number of layers of the quantum circuit after recompiling it to fit an MPS 1D path, i.e., the compiled circuit depth (for more details, see Appendix A). We set $g = 0$ for the pTEBD simulations since the performance showing the almost perfect weak scaling does not depend on g.

in Ref. [28], the IPMC does not increase the computational complexity.

It might be more insightful to compare the performance between these two algorithms by studying their elapsed time to achieve the same simulation precision, a measure similarly adopted in a recent study of a 2D tensor network algorithm [47]. This measure demonstrates the practical speedup of the pTEBD algorithm over its sequential counterpart. For this purpose, we reinterpret the fidelity results shown in Fig. 4 and replot these results as a function of the elapsed time of the simulation in Fig. 7. Remarkably, in all cases, the pTEBD simulation reaches the same $\mathcal{F}$ in polynomially shorter time than the corresponding sequential MPS simulation (as also expected from Fig. 6), indicating its efficient utilization of the parallel computing environment.

V. SUMMARY AND DISCUSSION

In summary, we have proposed an improved parallel MPS compression method that can accurately compress the dimensions of all the virtual bonds in a constant time, regardless of the system size. Simultaneously, it stabilizes the wavefunction norm of the MPS, converging to a value around 1 that is bounded from both sides. Although both the accuracy and the norm stabilization are mathematically proved under the assumption of the canonical form, we have demonstrated its feasibility in simulating unitary quantum dynamics, where the canonical form slightly deviates globally. Moreover, we have numerically shown that the deviated canonical form

resulting from the parallel truncation procedure in the parallel MPS compression can be gradually restored through the subsequent PtSU steps. Additionally, we have provided a proof that at most $\frac{N}{2}$ PtSU steps can drive any MPS to its canonical form.

Utilizing the IPMC method, we have proposed a fully real-space parallelizable pTEBD algorithm for efficiently simulating unitary quantum dynamics. Furthermore, we systematically benchmarked the pTEBD algorithm by simulating typical 1D and 2D quantum circuits. Our results demonstrate that the pTEBD algorithm achieves nearly perfect weak scaling performance even for very deep quantum circuits with hundreds of circuit layers. Additionally, it achieves the same simulation precision in polynomially shorter time compared to the sequential MPS algorithm.

While existing MPS-based quantum computing simulation methods have been recognized as highly efficient for simulating NISQ devices, their applications to quantum circuits with hundreds of qubits and circuit layers remains challenging due to the inherent limitation of linear time complexity with the system size. In contrast, the pTEBD algorithm proposed in this study can harness the abundant computing resources offered by current supercomputing systems to address these demanding tasks, providing a practical way for exploring quantum computing on large NISQ devices.

As we have demonstrated, the pTEBD algorithm exhibits the same simulation capability as other sequential MPS algorithms. However, due to the inherent limitation of MPS expressibility, simulating quantum circuits with higher-
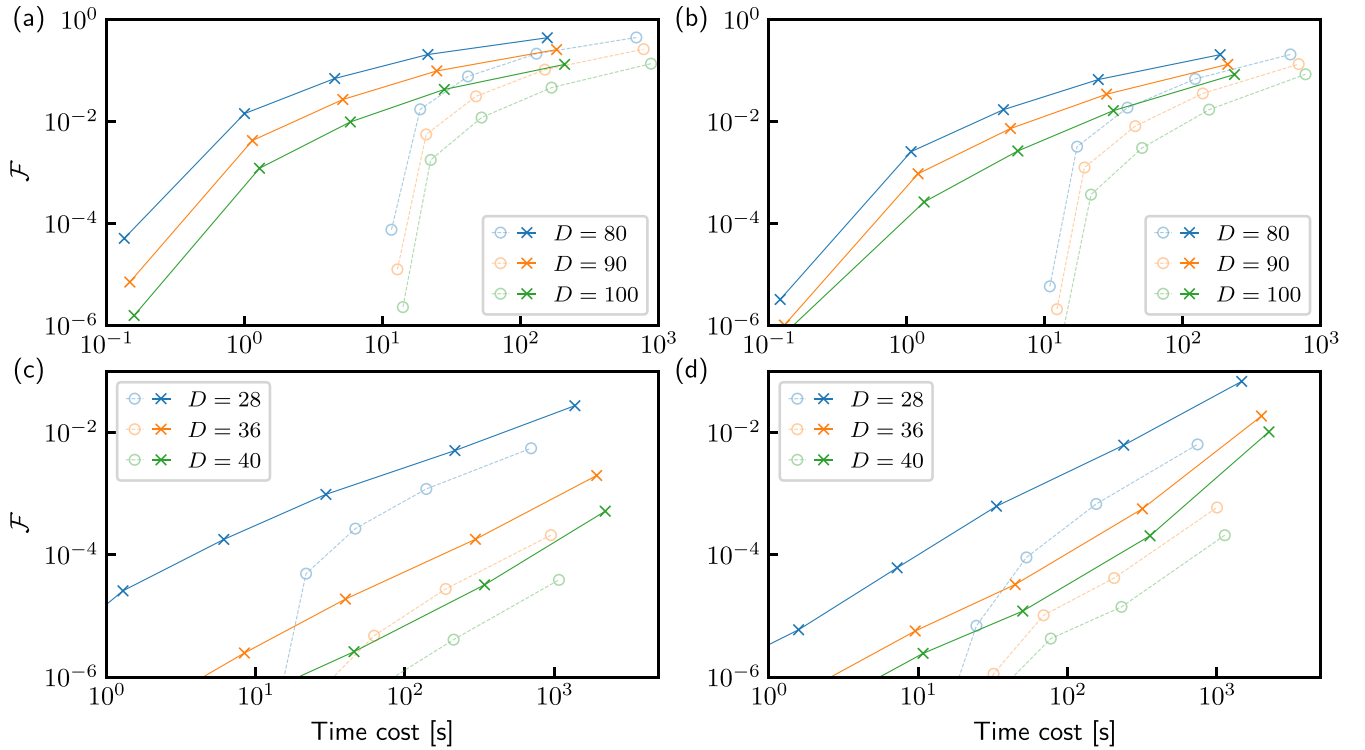
FIG. 7. Wave-function fidelity $\mathcal{F}$ versus elapsed time obtained using the sequential MPS algorithm (open circles) and the pTEBD algorithm (crosses) with a fixed physical circuit depth D (for the definition, see Appendix A): (a) RQC-1D with $N = 25$, (b) PQC-1D with $N = 24$, (c) RQC-2D with $L_x = L_y = 5$ (i.e., $N = 25$), and (d) PQC-2D with $L_x = 4$ and $L_y = 6$ (i.e., $N = 24$). These results are averaged over 10 simulations of the same quantum circuit but with different sets of random parameters. We set $g = 0$ for the pTEBD simulations. Note that the elapsed time increases simply because of the increase of bond dimensions.

dimensional structures and/or rapidly accumulating large entanglement remains challenging [2,9,48]. Therefore, it is of great interest to extend the present parallel approach to recently developed approximate contraction algorithms for more complex tensor networks [49,50] or to integrate the pTEBD algorithm with other levels of parallelization, such as those utilized in quantum chemistry simulations [51], which are feasible on current supercomputers.

APPENDIX A: DETAILS OF THE QUANTUM CIRCUITS ADOPTED IN THE BENCHMARKING SIMULATIONS

To benchmark the pTEBD algorithm, we select RQCs and PQCs, the latter being commonly used in variational quantum algorithms, as the simulation tasks. RQCs have been widely employed in benchmarking NISQ devices [2,53] and quantum computing simulations [19]. PQCs, being more featured circuits, can reflect the practical performance of devices and simulations for more realistic computational tasks. Therefore, these benchmarks allow for a thorough evaluation of the pTEBD algorithm's performance in the most realistic and representative cases. In this Appendix, we provide explanations for the construction of these quantum circuits. More detailed features and applications of these quantum circuits can be found in the corresponding references cited below.

1. Physical circuit depth vs compiled circuit depth

Physical circuit depth is defined as the total number of physical circuit layers. Typically, for a given quantum circuit featuring repeating structural units, as observed in the cases of RQCs and PQCs considered in this study, a physical circuit layer is determined by a single type of repeating unit. For instance, in the cases of RQCs, a single physical circuit layer is formed by applying single-qubit random gates to each qubit, followed by the application of two-qubit entangling gates with a specific covering pattern. The specific definition of the
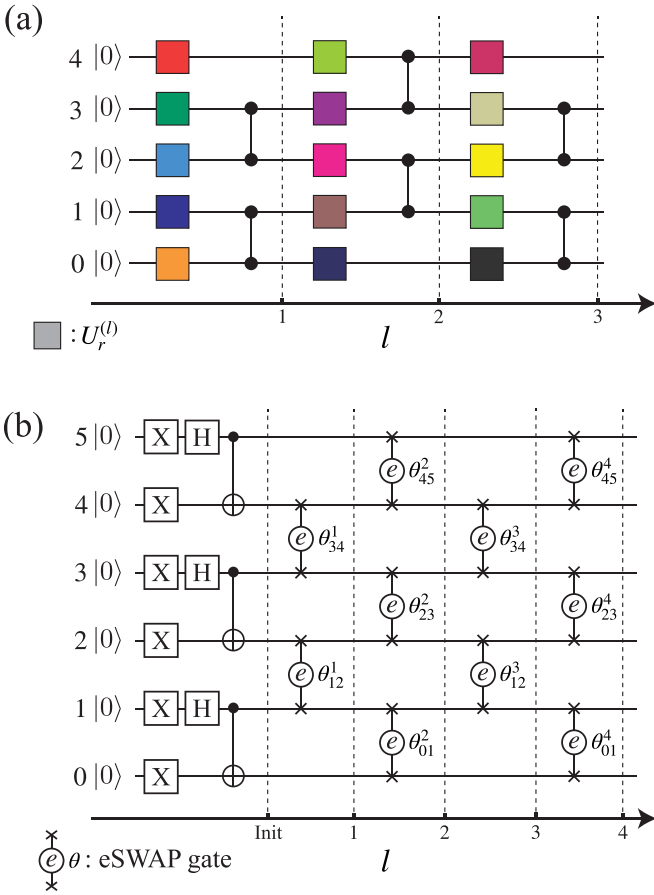
FIG. 8. Schematic figures for (a) RQC-1D and (b) PQC-1D. In (a), the colored squares represent single-qubit random gates. In (b), the first layer of the quantum circuit is dedicated to initializing the state as a product of singlet dimers. Dashed lines indicate each physical circuit layer.

physical circuit layer in each quantum circuit adopted in this study will be provided in the following subsections.

To enable a classical simulation of a quantum circuit with an MPS representation using a TEBD-type algorithm, a physical circuit layer must be recompiled to one (or several) compiled circuit layer(s), wherein all two-qubit gates exclusively apply to neighboring qubits aligned on the MPS 1D path. In this study, for 1D quantum circuits, one physical circuit layer is directly mapped to one compiled circuit layer. However, for 2D quantum circuits, a portion of physical circuit layers (details provided below) must be mapped to several compiled circuit layers, thereby increasing the effective circuit layers in the simulation. The total number of compiled circuit layers is denoted as the compiled circuit depth. As a single-qubit gate can always be contracted exactly into a nearby two-qubit gate during the recompiling procedure in the quantum circuit, we only consider two-qubit gates in defining and counting compiled circuit layers.

2. One-dimensional random quantum circuit

The circuit structure of RQC-1D is illustrated in Fig. 8(a). Here we adopt the same construction as in Ref. [7]. The single-qubit random gate at qubit r in the lth circuit layer,

denoted by squares in the figure, is defined as

$$
\begin{aligned}
U_r^{(l)} = \exp \big[-i\theta_r^{(l)} \big(& \sigma_r^x \sin \alpha_r^{(l)} \cos \phi_r^{(l)} \\
& + \sigma_r^y \sin \alpha_r^{(l)} \sin \phi_r^{(l)} + \sigma_r^z \cos \alpha_r^{(l)} \big) \big],
\end{aligned} \tag{A1}
$$

where σ_r^x, σ_r^y, and σ_r^z are Pauli matrices, and $\alpha_r^{(l)}$, $\theta_r^{(l)}$, and $\phi_r^{(l)}$ are uniformly distributed real random numbers. After applying $U_r^{(l)}$ to all qubits in the lth circuit layer, the quantum entanglement is generated by subsequently applying a layer of controlled-Z (CZ) gates, completing one physical layer of the circuit construction. The CZ gates apply to $(0, 1), (2, 3), \ldots, (N - 3, N - 2)$ qubit pairs for odd physical layers and on $(1, 2), (3, 4), \ldots, (N - 2, N - 1)$ qubit pairs for even physical layers. The total number N of qubits is restricted to odd, and the total number D of physical circuit layers is restricted to even. Note that these RQCs (including the RQC-2D introduced below) are different from those used in Google's quantum supremacy experiment [2], in which fSim gates are adopted to generate larger entanglement. More detailed discussion of these two kinds of RQCs can be found in Refs. [7,54].

3. One-dimensional parametrized quantum circuit

For a more practical setting, we also consider the Hamiltonian variational *ansatz* (HVA) [55], which has been widely studied in quantum computing for quantum many-body systems, and apply it to the nearest-neighbor spin $S = 1/2$ Heisenberg model. Initializing the qubit register to form a product of singlet dimers on $(0, 1), (2, 3), \ldots, (N - 2, N - 1)$ qubit pairs [see Fig. 8(b)], a layer of eSWAP gates [56–62], $U_{ij}^{(l)}(\theta_{ij}^{(l)}) = \exp(-i\theta_{ij}^{(l)} P_{ij}/2)$ with P_{ij} being the SWAP gate acting at qubits i and j, is applied to $(1, 2), (3, 4), \ldots, (N - 3, N - 2)$ [$(0, 1), (2, 3), \ldots, (N - 2, N - 1)$] qubit pairs for the lth physical layer with odd (even) l. In using the HVA for variational quantum algorithms [61,62], $\{\theta_{ij}^{(l)}\}$ is a set of variational parameters to be optimized. Here we assign uniformly distributed real random numbers to these parameters $\{\theta_{ij}^{(l)}\}$ in our benchmark simulations. The resulting circuit structure of PQC-1D is illustrated in Fig. 8(b). The total number N of qubits and the total number D of physical circuit layers are both restricted to even.

4. Two-dimensional random quantum circuit

RQC-2D is the direct extension of RQC-1D on a 2D square lattice with the number of qubits $N = L_x \times L_y$. The physical circuit layers are applied in an ABCDABCD $\cdots$ pattern, as illustrated in Fig. 9, where each rectangle indicates the qubits on which a CZ gate acts. Additionally, single-qubit random gates $\{U_r^{(l)}(\theta_{ij}^{(l)})\}$ are applied to all qubits in each physical circuit layer, similar to the case of RQC-1D. The number of physical circuit layers is restricted to a multiple of 4. However, there are no specific restrictions on L_x and L_y. In cases when L_x (L_y) is odd, no two-qubit gates are applied on qubits at the right and left (top and bottom) edges in physical circuit layers C and D (A and B), respectively, as shown in Fig. 9 for $L_x = L_y = 5$.

In contrast to 1D quantum circuits, where qubits naturally align in a 1D path suitable for the MPS representation, a
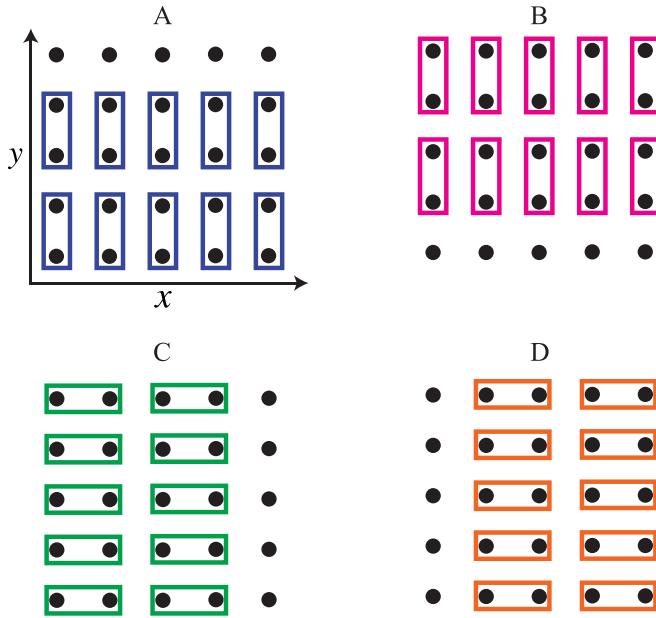
FIG. 9. Locations of two-qubit gates, indicated by rectangles, in four distinct physical circuit layers A, B, C, and D for both RQC-2D and PQC-2D. Solid dots represent qubits arranged in a 2D square lattice. The two-qubit gates consist of CZ gates for RQC-2D and eSWAP gates for PQC-2D. Additionally, single-qubit random gates are applied to all qubits in each physical circuit layer for RQC-2D, similar to the case of RQC-1D shown in Fig. 8(a). For PQC-2D, the first physical layer A is dedicated to forming a product of singlet dimers, as in the case of PQC-1D shown in Fig. 8(b). In this example, we set $L_x = 5$ and $L_y = 5$.

mapping of qubits for 2D quantum circuits to a 1D path suitable for the MPS representation must be determined. Here we choose the path shown in Fig. 10(a) to ensure that two-qubit gates in physical circuit layers A and B act on neighboring qubits in the sense of the MPS 1D path. Under this mapping, the two-qubit gates in physical circuit layers C and D become long-distance gates, necessitating a recompilation of these two physical circuit layers for MPS-based simulations to ensure that gates apply only to neighboring qubits on the MPS 1D path. As an example, the recompiling procedure for the $L_x = L_y = 4$ case is illustrated in Fig. 10(b). Note that the circuit depth increases from 2 (physical circuit depth) to $3(L_y - 1) + 2$ (compiled circuit depth) after this recompiling. The physical circuit depths and compiled circuit depths for several 2D quantum circuits studied in Figs. 4–7 are provided in Table I. Although we focus on the square lattice, an extension to other lattices such as the Bristlecone lattice used in Ref. [2] is straightforward.

5. Two-dimensional parametrized quantum circuit

Similar to RQC-2D, PQC-2D is the extension of PQC-1D on a 2D square lattice with the number of qubits $N = L_x \times L_y$. A slight difference from RQC-2D is found in the first physical circuit layer A, where we prepare a product of singlet dimers, similar to PQC-1D. Beyond this initial physical circuit layer A, the subsequent physical circuit layers incorporate eSWAP gates in a BCDABCDA··· pattern with parameters $\{\theta_{ij}^{(l)}\}$ (see
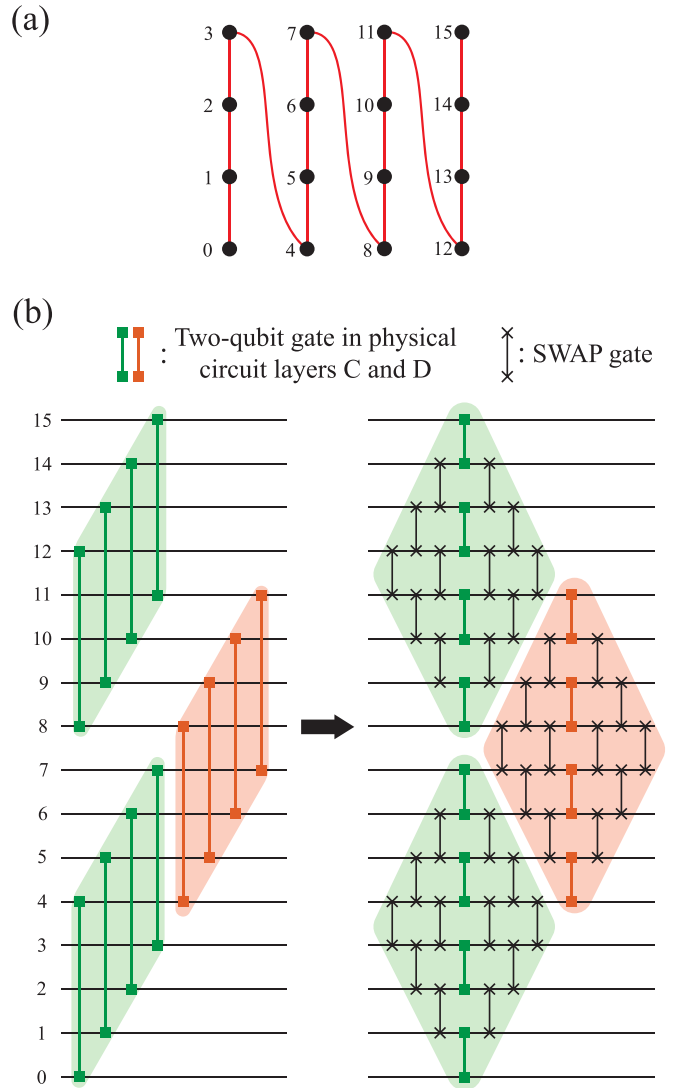

FIG. 10. (a) Mapping of qubits for a 2D quantum circuit to a 1D path suitable for its MPS representation. Solid dots represent qubits arranged in a 2D square lattice with $L_x = L_y = 4$. The numbers beside solid dots indicate the indices of qubits. (b) Arrangement of two-qubit gates in physical circuit layers C and D for a 2D quantum circuit with $L_x = L_y = 4$ (left-hand side) and the equivalent recompiled quantum circuit containing only two-qubit gates applying to neighboring qubits but with many additional SWAP gates. The number of each qubit register corresponds to that in (a). The shades of the same color indicate the correspondence between gates involved before and after the recompiling procedure for each physical circuit layer. Although we consider the $L_x = L_y = 4$ case here as an example, it is straightforward to extend to other cases.

Fig. 9). Similar to RQC-2D, we employ an MPS 1D path as depicted in Fig. 10(a). Consequently, two-qubits gates in physical circuit layers C and D must be recompiled to ensure that all gates apply only to neighboring qubits, as illustrated in Fig. 10(b). Due to the specific configuration, the value of L_y in PQC-2D is restricted to an even number. As in the case of RQC-2D, the total number D of physical circuit layers is a multiple of 4.

TABLE I. Physical and compiled circuit depths for several 2D quantum circuits studied in Figs. 4–7. Here a 2D quantum circuit is arranged in a square lattice with the number of qubits $N = L_x \times L_y$.

N	L_x	L_y	Physical circuit depth	Compiled circuit depth
24	4	6	28	133
24	4	6	60	285
25	5	5	28	112
25	5	5	60	192
144	12	12	40	370
144	12	12	100	925
1024	32	32	100	2425

TABLE II. Physical and compiled circuit depths for QFT circuits with various numbers (N) of qubits.

N	Physical circuit depth	Compiled circuit depth
16	136	946
20	210	1482
24	300	2219
28	406	3034
32	528	3882
48	1176	8430
64	2080	14 670
80	3240	22 514
96	4656	31 493

APPENDIX B: SIMULATION ACCURACY OF THE pTEBD ALGORITHM FOR A SINGLE INSTANCE OF RANDOM PARAMETERS

For a fair comparison of the wave-function fidelity $\mathcal{F}$ obtained by the pTEBD algorithm and the sequential MPS algorithm, here we compare the results for a specific set of random parameters rather than averaging over 10 simulations with different sets as in Fig. 4. This particular set is chosen among the 10 different sets used in Fig. 4. The corresponding results are denoted as $\mathcal{F}_{\mathrm{pTEBD}}$ and $\mathcal{F}_{\mathrm{MPS}}$ obtained by the pTEBD and sequential MPS algorithms, respectively.

As shown in Fig. 11, in most cases, we observe behavior consistent with that found in Fig. 4: $\mathcal{F}_{\mathrm{pTEBD}} \approx \mathcal{F}_{\mathrm{MPS}}$, and

$\mathcal{F}_{\mathrm{pTEBD}}$ increases with the number g of PtSU steps. Additionally, we note that $\mathcal{F}_{\mathrm{pTEBD}}$ outperforms $\mathcal{F}_{\mathrm{MPS}}$ in some cases with small bond dimensions and larger circuit depths, while $\mathcal{F}_{\mathrm{pTEBD}}$ decreases with more appended PtSU steps in certain cases. In practice, we can adjust g to achieve optimal performance in the pTEBD simulation.

APPENDIX C: ADDITIONAL BENCHMARK TEST OF PTEBD

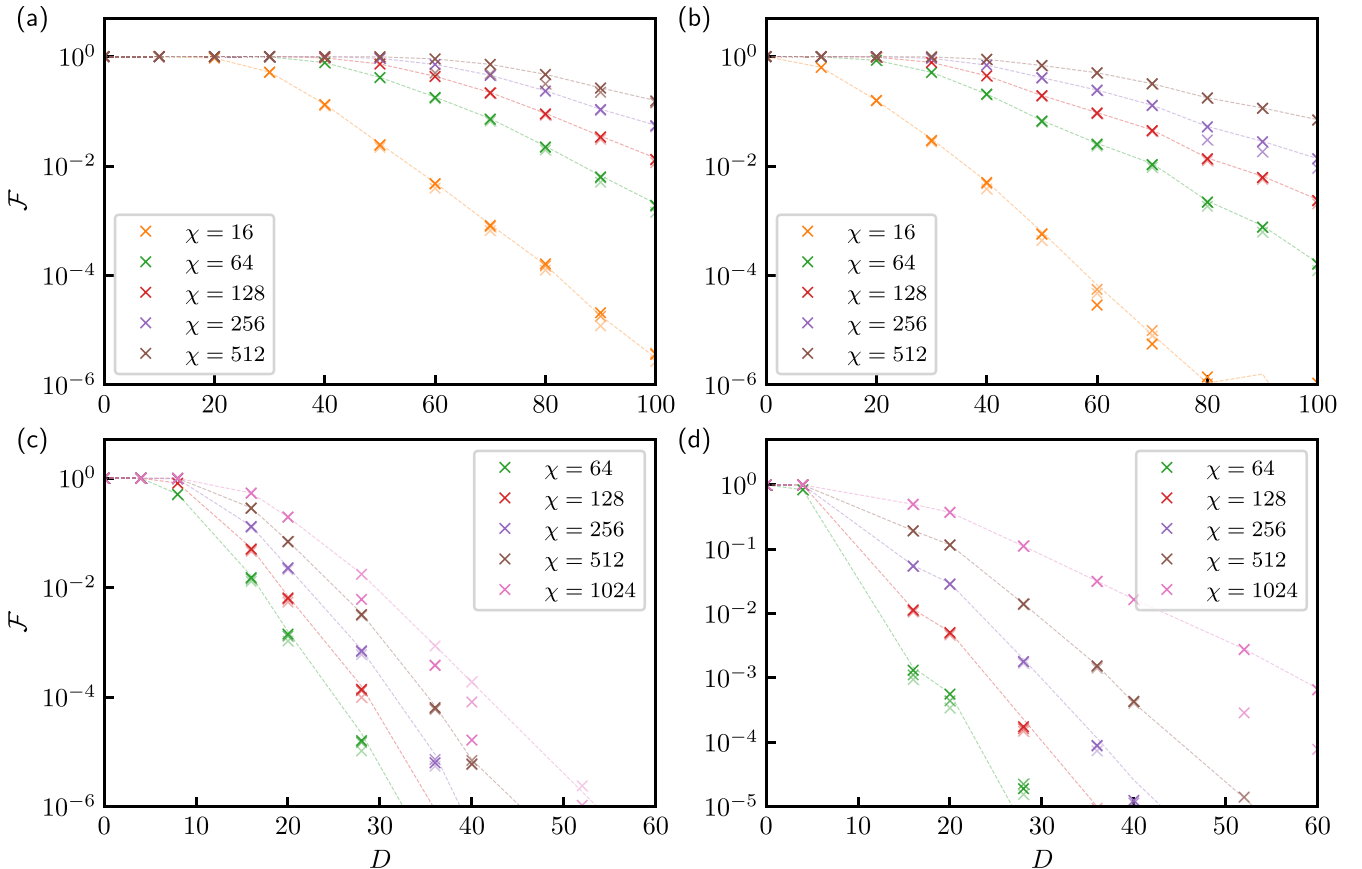Quantum Fourier transformation (QFT) [63] is a fundamental algorithm in fault-tolerant quantum computing

FIG. 11. Same as Fig. 4 but showing the results for a single set of random parameters, selected from the 10 different sets of random parameters used for averaging in Fig. 4.
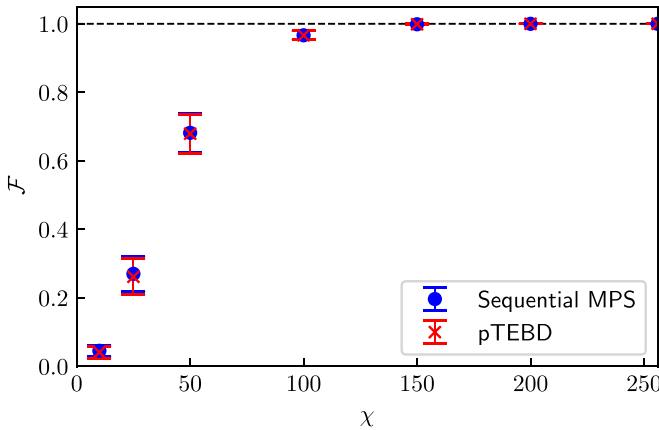
FIG. 12. Wave-function fidelity $\mathcal{F}$ as a function of MPS bond dimension χ in simulations of the QFT circuit with $N = 16$ using the sequential MPS algorithm and the pTEBD algorithm. We set $g = 0$ for the pTEBD simulations. The results for each χ are obtained from the average over 10 different random initial states.
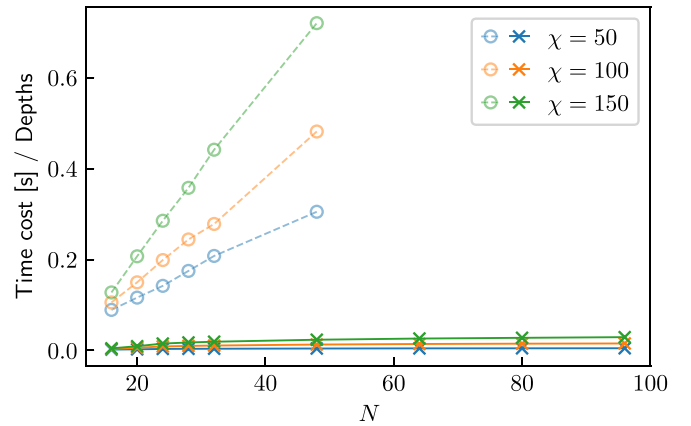

FIG. 13. Elapsed time per circuit layer versus the number N of qubits obtained using the sequential MPS algorithm (open circles) and the pTEBD algorithm (crosses), with a fixed MPS bond dimension χ in the simulations of QFT circuits. For the pTEBD simulations, we set $g = 0$ and distribute 16 qubits to each node.

(FTQC) and is typically integrated into other FTQC algorithms, such as Shor's algorithm [64] and the quantum phase estimation algorithm [65]. In this Appendix, we simulate the QFT applied to a random state using both the sequential MPS algorithm and the pTEBD algorithm to assess the accuracy and scalability of the pTEBD algorithm in the simulation of more irregularly structured quantum circuits compared with those studied in Sec. IV B.

We initialize a random MPS with bond dimension $\chi_0 = 10$ and apply a QFT circuit to this random state. The QFT circuit is generated and further compiled to a qubits layout with linear connectivity using Qiskit [66]. For the compilation, the `transpile()` function is used with the single-qubit rotation gate $U(\theta, \phi, \lambda)$ and the controlled-NOT gate CX as the basis gates, and with the optimization level 3. The physical circuit depths and compiled circuit depths of the QFT circuits for different system sizes are summarized in Table II. We then employ the sequential MPS algorithm and the pTEBD algorithm to simulate these compiled circuits.

First, we examine the simulation accuracy. For this purpose, we fix the number of qubits $N = 16$ with the corresponding compiled circuit depth being 946. The results of the simulation accuracy, quantified by the wave-function fidelity $\mathcal{F}$ in Eq. (38), are shown in Fig. 12. We observe that

the wave-function fidelity in the pTEBD simulation always approximates very accurately the wave-function fidelity obtained from the sequential MPS simulation, even in the cases with relatively small bond dimensions. This indicates that the accuracy of the pTEBD algorithm is comparable to that of the sequential MPS algorithm, as in the cases for more regularly structured quantum circuits before recompilation, studied in Fig. 4.

Furthermore, we compare the performance of these two algorithms in simulating QFT circuits. Figure 13 shows the elapsed time per compiled circuit layer against the number N of qubits. The elapsed time per layer increases linearly with N in the sequential MPS simulations, while it remains approximately constant, especially for larger N, in the pTEBD simulations. The very slowly increase in elapsed time per layer in the pTEBD simulations might be due to the increase in internode communications, as we always distribute 16 qubits to each node in these simulations. This performance analysis demonstrates that the pTEBD algorithm can also achieve good weak scaling even when simulating more general, unstructured circuits, consistent with the results for RQC-2D and PQC-2D in Figs. 6(c) and 6(d), where the quantum circuits after recompilation are not perfectly regularly structured.

[1] C. Gross and I. Bloch, Quantum simulations with ultracold atoms in optical lattices, Science **357**, 995 (2017).

[2] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[3] F. Schäfer, T. Fukuhara, S. Sugawa, Y. Takasu, and Y. Takahashi, Tools for quantum simulation with ultracold atoms in optical lattices, Nat. Rev. Phys. **2**, 411 (2020).

[4] I. Pogorelov, T. Feldker, C. D. Marciniak, L. Postler, G. Jacob, O. Krieglsteiner, V. Podlesnic, M. Meth, V. Negnevitsky, M. Stadler, B. Höfer, C. Wächter, K. Lakhmanskiy,

R. Blatt, P. Schindler, and T. Monz, Compact ion-trap quantum computing demonstrator, PRX Quantum **2**, 020343 (2021).

[5] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, UK, 2010).

[6] A. J. Daley, I. Bloch, C. Kokail, S. Flannigan, N. Pearson, M. Troyer, and P. Zoller, Practical quantum advantage in quantum simulation, Nature (London) **607**, 667 (2022).

[7] Y. Zhou, E. M. Stoudenmire, and X. Waintal, What limits the simulation of quantum computers? Phys. Rev. X **10**, 041038 (2020).

[8] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[9] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. Van Den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme *et al.*, Evidence for the utility of quantum computing before fault tolerance, Nature (London) **618**, 500 (2023).

[10] Y. Alexeev, M. Amsler, P. Baity, M. A. Barroca, S. Bassini, T. Battelle, D. Camps, D. Casanova, Y. J. Choi, F. T. Chong, C. Chung, C. Codella, A. D. Corcoles, J. Cruise, A. Di Meglio, J. Dubois, I. Duran, T. Eckl, S. Economou, S. Eidenbenz *et al.*, Quantum-centric supercomputing for materials science: A perspective on challengesand future directions, Future Generation Comput. Syst. **160**, 666 (2024).

[11] J. Robledo-Moreno, M. Motta, H. Haas, A. Javadi-Abhari, P. Jurcevic, W. Kirby, S. Martiel, K. Sharma, S. Sharma, T. Shirakawa, I. Sitdikov, R.-Y. Sun, K. J. Sung, M. Takita, M. C. Tran, S. Yunoki, and A. Mezzacapo, Chemistry beyond exact solutions on a quantum-centric supercomputer, arXiv:2405.05068.

[12] T. Häner and D. S. Steiger, 0.5 petabyte simulation of a 45-qubit quantum circuit, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '17)* (Association for Computing Machinery, New York, NY, 2017).

[13] H. De Raedt, F. Jin, D. Willsch, M. Willsch, N. Yoshioka, N. Ito, S. Yuan, and K. Michielsen, Massively parallel quantum computer simulator, eleven years later, Comput. Phys. Commun. **237**, 47 (2019).

[14] C. Guo, Y. Liu, M. Xiong, S. Xue, X. Fu, A. Huang, X. Qiang, P. Xu, J. Liu, S. Zheng, H.-L. Huang, M. Deng, D. Poletti, W.-S. Bao, and J. Wu, General-purpose quantum circuit simulator with projected entangled-pair states and the quantum supremacy frontier, Phys. Rev. Lett. **123**, 190501 (2019).

[15] J. Gray and S. Kourtis, Hyper-optimized tensor network contraction, Quantum **5**, 410 (2021).

[16] C. Guo, Y. Zhao, and H.-L. Huang, Verifying random quantum circuits with arbitrary geometry using tensor network states algorithm, Phys. Rev. Lett. **126**, 070502 (2021).

[17] F. Pan and P. Zhang, Simulation of quantum circuits using the big-batch tensor network method, Phys. Rev. Lett. **128**, 030501 (2022).

[18] Y. Liu, Y. Chen, C. Guo, J. Song, X. Shi, L. Gan, W. Wu, W. Wu, H. Fu, X. Liu, D. Chen, Z. Zhao, G. Yang, and J. Gao, Verifying quantum advantage experiments with multiple amplitude tensor network contraction, Phys. Rev. Lett. **132**, 030601 (2024).

[19] Y. Liu, X. Liu, F. Li, H. Fu, Y. Yang, J. Song, P. Zhao, Z. Wang, D. Peng, H. Chen, C. Guo, H. Huang, W. Wu, and D. Chen, Closing the "quantum supremacy" gap: Achieving real-time simulation of a random quantum circuit using a new sunway supercomputer, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '21)* (Association for Computing Machinery, New York, NY, 2021).

[20] H. Collins and C. Nay, IBM unveils 400 qubit-plus quantum processor and next-generation IBM quantum system two [press release], https://newsroom.ibm.com/2022-11-09-IBM-Unveils-400-Qubit-Plus-Quantum-Processor-and-Next-Generation-IBM-Quantum-System-Two.

[21] R. Orús, A practical introduction to tensor networks: Matrix product states and projected entangled pair states, Ann. Phys. **349**, 117 (2014).

[22] J. I. Cirac, D. Pérez-García, N. Schuch, and F. Verstraete, Matrix product states and projected entangled pair states: Concepts, symmetries, theorems, Rev. Mod. Phys. **93**, 045003 (2021).

[23] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys. **326**, 96 (2011).

[24] V. M. F. Verstraete and J. I. Cirac, Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems, Adv. Phys. **57**, 143 (2008).

[25] T. Ayral, T. Louvet, Y. Zhou, C. Lambert, E. M. Stoudenmire, and X. Waintal, Density-matrix renormalization group algorithm for simulating quantum circuits with a finite fidelity, PRX Quantum **4**, 020304 (2023).

[26] J. Tindall, M. Fishman, M. Stoudenmire, and D. Sels, Efficient tensor network simulation of IBM's kicked Ising experiment, PRX Quantum **5**, 010308 (2024).

[27] F. Verstraete and J. I. Cirac, Matrix product states represent ground states faithfully, Phys. Rev. B **73**, 094423 (2006).

[28] M. Urbanek and P. Soldán, Parallel implementation of the time-evolving block decimation algorithm for the Bose-Hubbard model, Comput. Phys. Commun. **199**, 170 (2016).

[29] P. Secular, N. Gourianov, M. Lubasch, S. Dolgov, S. R. Clark, and D. Jaksch, Parallel time-dependent variational principle algorithm for matrix product states, Phys. Rev. B **101**, 235123 (2020).

[30] E. M. Stoudenmire and S. R. White, Real-space parallel density matrix renormalization group, Phys. Rev. B **87**, 155137 (2013).

[31] G. Vidal, Efficient classical simulation of slightly entangled quantum computations, Phys. Rev. Lett. **91**, 147902 (2003).

[32] T. Shirakawa, H. Ueda, and S. Yunoki, Automatic quantum circuit encoding of a given arbitrary quantum state, arXiv:2112.14524.

[33] G. Vidal, Efficient simulation of one-dimensional quantum many-body systems, Phys. Rev. Lett. **93**, 040502 (2004).

[34] R. Orús and G. Vidal, Infinite time-evolving block decimation algorithm beyond unitary evolution, Phys. Rev. B **78**, 155117 (2008).

[35] H. C. Jiang, Z. Y. Weng, and T. Xiang, Accurate determination of tensor network state of quantum lattice models in two dimensions, Phys. Rev. Lett. **101**, 090603 (2008).

[36] H. Kalis, D. Klagges, R. Orús, and K. P. Schmidt, Fate of the cluster state on the square lattice in a magnetic field, Phys. Rev. A **86**, 022317 (2012).

[37] S.-J. Ran, W. Li, B. Xi, Z. Zhang, and G. Su, Optimized decimation of tensor networks with super-orthogonalization for two-dimensional quantum lattice models, Phys. Rev. B **86**, 134429 (2012).

[38] R. Alkabetz and I. Arad, Tensor networks contraction and the belief propagation algorithm, Phys. Rev. Res. **3**, 023073 (2021).

[39] S. R. White, Density matrix formulation for quantum renormalization groups, Phys. Rev. Lett. **69**, 2863 (1992).

[40] U. Schollwöck, The density-matrix renormalization group, Rev. Mod. Phys. **77**, 259 (2005).

[41] S. Paeckel, T. Köhler, A. Swoboda, S. R. Manmana, U. Schollwöck, and C. Hubig, Time-evolution methods for matrix-product states, Ann. Phys. **411**, 167998 (2019).

[42] M.-D. Choi, Completely positive linear maps on complex matrices, Linear Algebra Appl. **10**, 285 (1975).

[43] Y. Ajima, T. Kawashima, T. Okamoto, N. Shida, K. Hirai, T. Shimizu, S. Hiramoto, Y. Ikeda, T. Yoshikawa, K. Uchida, and T. Inoue, The Tofu interconnect D, in *Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER)* (IEEE, Los Alamitos, CA, 2018), pp. 646–654.

[44] J. Tindall and M. T. Fishman, Gauging tensor networks with belief propagation, SciPost Phys. **15**, 222 (2023).

[45] A. J. Ferris and G. Vidal, Perfect sampling with unitary tensor networks, Phys. Rev. B **85**, 165146 (2012).

[46] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson, The variational quantum eigensolver: A review of methods and best practices, Phys. Rep. **986**, 1 (2022).

[47] D. Adachi, T. Okubo, and S. Todo, Anisotropic tensor renormalization group, Phys. Rev. B **102**, 054432 (2020).

[48] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, Characterizing quantum supremacy in near-term devices, Nat. Phys. **14**, 595 (2018).

[49] F. Pan, P. Zhou, S. Li, and P. Zhang, Contracting arbitrary tensor networks: General approximate algorithm and applications in graphical models and quantum circuit simulations, Phys. Rev. Lett. **125**, 060503 (2020).

[50] J. Gray and G. Kin-Lic Chan, Hyperoptimized approximate contraction of tensor networks with arbitrary geometry, Phys. Rev. X **14**, 011009 (2024).

[51] H. Shang, L. Shen, Y. Fan, Z. Xu, C. Guo, J. Liu, W. Zhou, H. Ma, R. Lin, Y. Yang *et al.*, Large-scale simulation of quantum computational chemistry on a new sunway supercomputer, in *International Conference for High Performance Computing, Networking, Storage and Analysis (SC22)* (IEEE, Los Alamitos, CA, 2022), pp. 1–14.

[52] GraceQ/tensor, https://github.com/gracequantum/tensor.

[53] Y. Wu, W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen, T.-H. Chung, H. Deng, Y. Du, D. Fan, M. Gong, C. Guo, C. Guo, S. Guo, L. Han, L. Hong, H.-L. Huang, Y.-H. Huo, L.

Li, N. Li *et al.*, Strong quantum computational advantage using a superconducting quantum processor, Phys. Rev. Lett. **127**, 180501 (2021).

[54] F. Pan, K. Chen, and P. Zhang, Solving the sampling problem of the sycamore quantum circuits, Phys. Rev. Lett. **129**, 090502 (2022).

[55] D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, Phys. Rev. A **92**, 042303 (2015).

[56] D. Loss and D. P. DiVincenzo, Quantum computation with quantum dots, Phys. Rev. A **57**, 120 (1998).

[57] D. P. DiVincenzo, D. Bacon, J. Kempe, G. Burkard, and K. B. Whaley, Universal quantum computation with the exchange interaction, Nature (London) **408**, 339 (2000).

[58] R. Brunner, Y.-S. Shin, T. Obata, M. Pioro-Ladrière, T. Kubo, K. Yoshida, T. Taniyama, Y. Tokura, and S. Tarucha, Two-qubit gate of combined single-spin rotation and interdot spin exchange in a double quantum dot, Phys. Rev. Lett. **107**, 146801 (2011).

[59] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nat. Phys. **10**, 631 (2014).

[60] H.-K. Lau and M. B. Plenio, Universal quantum computing with arbitrary continuous-variable encoding, Phys. Rev. Lett. **117**, 100501 (2016).

[61] K. Seki, T. Shirakawa, and S. Yunoki, Symmetry-adapted variational quantum eigensolver, Phys. Rev. A **101**, 052340 (2020).

[62] R.-Y. Sun, T. Shirakawa, and S. Yunoki, Efficient variational quantum circuit structure for correlated topological phases, Phys. Rev. B **108**, 075127 (2023).

[63] D. Coppersmith, An approximate Fourier transform useful in quantum factoring, arXiv:quant-ph/0201067.

[64] P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE, Los Alamitos, CA, 1994), pp. 124–134.

[65] A. Y. Kitaev, Quantum measurements and the Abelian stabilizer problem, arXiv:quant-ph/9511026.

[66] Qiskit, https://www.qiskit.org/.