The Journal of Chemical Physics

RESEARCH ARTICLE | AUGUST 06 2024

# Optimal tree tensor network operators for tensor network simulations: Applications to open quantum systems

Special Collection: Algorithms and Software for Open Quantum System Dynamics

Weitang Li ; Jiajun Ren ; Hengrui Yang ; Haobin Wang ; Zhigang Shuai

*J. Chem. Phys.* 161, 054116 (2024)
https://doi.org/10.1063/5.0218773

AIP Publishing

The Journal of Chemical Physics | ARTICLE | pubs.aip.org/aip/jcp

# Optimal tree tensor network operators for tensor network simulations: Applications to open quantum systems

Cite as: J. Chem. Phys. **161**, 054116 (2024); doi: 10.1063/5.0218773
Submitted: 13 May 2024 • Accepted: 17 July 2024 •
Published Online: 6 August 2024

Weitang Li,[1,2,a)] Jiajun Ren,[3] Hengrui Yang,[4] Haobin Wang,[5] and Zhigang Shuai[1,4]

## AFFILIATIONS

[1] School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, People's Republic of China
[2] Tencent Quantum Lab, Tencent, Shenzhen 518057, People's Republic of China
[3] MOE Key Laboratory of Theoretical and Computational Photochemistry, College of Chemistry, Beijing Normal University, Beijing 100875, People's Republic of China
[4] MOE Key Laboratory of Organic OptoElectronics and Molecular Engineering, Department of Chemistry, Tsinghua University, 100084 Beijing, People's Republic of China
[5] Department of Chemistry, University of Colorado Denver, Denver, Colorado 80217-3364, USA

**Note:** This paper is part of the JCP Special Topic on Algorithms and Software for Open Quantum System Dynamics.
[a)] **Author to whom correspondence should be addressed:** liwt31@gmail.com

## ABSTRACT

Tree tensor network states (TTNS) decompose the system wavefunction to the product of low-rank tensors based on the tree topology, serving as the foundation of the multi-layer multi-configuration time-dependent Hartree method. In this work, we present an algorithm that automatically constructs the optimal and exact tree tensor network operators (TTNO) for any sum-of-product symbolic quantum operator. The construction is based on the minimum vertex cover of a bipartite graph. With the optimal TTNO, we simulate open quantum systems, such as spin relaxation dynamics in the spin-boson model and charge transport in molecular junctions. In these simulations, the environment is treated as discrete modes and its wavefunction is evolved on equal footing with the system. We employ the Cole–Davidson spectral density to model the glassy phonon environment and incorporate temperature effects via thermo-field dynamics. Our results show that the computational cost scales linearly with the number of discretized modes, demonstrating the efficiency of our approach.

## I. INTRODUCTION

Tensor network algorithms have emerged as a highly effective numerical approach for studying open quantum systems. These algorithms decompose the combined system and bath wavefunction into a product of low-rank tensors, providing both efficiency and flexibility. The multilayer multiconfiguration time-dependent Hartree method[1–3] (ML-MCTDH), which utilizes tree tensor network states (TTNS) ansatz,[4–6] has been successfully employed to investigate a wide range of problems. These include non-adiabatic chemical reactions in the condensed phase,[7–9] charge and heat transfer through molecular junctions,[10–12] relaxation dynamics of a spin coupled to various types of baths,[13,14] and energy transfer in biological molecules.[15] TTNS has also found applications in *ab initio* quantum chemistry,[5,16,17] computation of vibrational eigenstates,[18] and solving hierarchical equations of motion,[19,20] among others.[21,22] Another branch of tensor network algorithms is time-dependent density matrix renormalization group (TD-DMRG), which is based on matrix product states (MPS).[23,24] MPS is a specific form of tensor networks where the low-rank tensors are arranged in a one-dimensional chain. Originally designed to address problems related to the ground state or low-lying excited states,[25–27] DMRG has recently been extended to the time-dependent domain.[24,28] This extension has enabled the simulation of ultrafast exciton dynamics in organic materials,[29–31] spectra of molecular aggregates,[32–34] transport properties of organic semiconductors and thermoelectric materials,[35,36] and the simulation of open quantum systems.[37–40]

Matrix product operators (MPOs) have significantly contributed to the success of MPS.[23,41] Similar to MPS, MPO is a low-rank decomposition of quantum operators. When the input operators are in a sum-of-product (SOP) form, the analytical form of the MPO can be obtained without any approximation. The form of MPO for a given operator is not unique, and it is desirable to construct the most compact MPO to minimize the subsequent computational cost in DMRG algorithms. In the following, we refer to the most compact MPO as the "optimal" MPO. Typically, the construction of the MPO for the Hamiltonian is required for the ground state search or time evolution. The MPO of interested physical observables are also frequently constructed for efficient evaluation of the expectation values. One of the earliest and most straightforward methods for constructing MPOs is through manual design. This approach is commonly used in practical MPS algorithms, particularly for constructing MPOs for *ab initio* electronic structure Hamiltonians.[41,42] However, this method is labor-intensive and prone to errors if the MPOs of many different types of operators are required. To alleviate the need for manual MPO design for different operators, various methods for automated MPO construction based on the input operator have been proposed. A popular approach involves naively constructing an MPO that is far from optimal initially, and then, compressing it through singular value decomposition (SVD).[43,44] Recently, we proposed another method based on bipartite graph theory.[45] The method translates from the symbolic operator strings to the optimal MPO without any numerical error and has become the cornerstone of a series of recent TD-DMRG applications.[34–36,46]

The concept of MPS/MPO can be generalized to TTNS/TTNO.[47] The manual construction of TTNO is more complex than MPO due to greater flexibility of the TTN structure.[20] Developing a general algorithm for automatic TTNO construction also presents a significant challenge. Analog to the construction of MPO, numerical SVD compression can be used for TTNO construction.[48,49] However, this numerical SVD compression is typically expensive. Recently, an algorithm for the symbolic construction of TTNO based on state diagrams is proposed,[50] yet the resulting TTNO is not optimal. Due to these challenges, several TTNS studies have opted to use a direct sum-of-product Hamiltonian instead of TTNO,[18,19,51,52] which results in a higher computational scaling than using optimal TTNO.[24]

In this work, we extend our former bipartite graph theory approach for MPO construction to TTNO construction. Our algorithm efficiently constructs the optimal TTNO for any operator in the sum-of-product form. For the spin-boson model,[53] our algorithm generates a TTNO whose $M_{\rm O}$ is constant, meaning it is independent of the number of modes in the model. Combined with the projector splitting method for time evolution, we show that the computational cost for the simulation scales linearly with the number of modes. We also analyze the computational scaling with respect to the dimension of node indices in TTNS for different TTN topologies, ranging from MPS to binary and ternary trees. We showcase the capabilities of our algorithm by studying the spin relaxation dynamics of the spin-boson model and charge transport in a molecular junction. We also consider the finite temperature effect through thermo-field dynamics. Thanks to the automatic TTNO construction, the programming cost for the inclusion of the temperature effect is negligible.

## II. ALGORITHM IMPLEMENTATION

In this section, we will first recap the concepts of TTNS and TTNO in Sec. II A and then describe the time evolution algorithm based on the projector splitting integrator and TTNO in Sec. II B. Finally, we will describe our algorithm for automatic TTNO construction based on bipartite graph theory in Sec. II C. All the algorithms discussed in this section have been implemented in the latest version of the open-source package RENORMALIZER. The core advantage of our implementation, compared to other TTNS packages, such as the Heidelberg MCTDH package[54] and QuTree,[55] is its use of TTNO and its Python-based nature. Python is a high-level scripting language known for its readability and ease of use. This combination of Python's accessibility and TTNO's efficiency makes our implementation a powerful platform for TTNS-based simulations.

### A. TTNS and TTNO

Suppose a quantum system has $N$ degrees of freedom, and for each degree of freedom, the corresponding primitive basis is $|\sigma_i\rangle$, the TTNS ansatz represents the wavefunction of this many-body system as the result of contracting low-rank tensors. The TTNS ansatz can be expressed as

$$|\Psi\rangle = \sum_{\{a\},\{\sigma\}} A[1]^{\sigma_1}_{\Lambda_1,a_1} A[2]^{\sigma_2}_{\Lambda_2,a_2} \cdots A[N]^{\sigma_N}_{\Lambda_N,a_N} |\sigma_1\sigma_2\cdots\sigma_N\rangle. \tag{1}$$

Here, $A[i]$ represents the low-rank tensors, with indices $\sigma_i$, $a_i$, and $\Lambda_i$. $\sigma_i$ is called the physical index since it is associated with a physical degree of freedom, whereas $a_i$ and $\Lambda_i$ are virtual indices. The contraction between $A[i]$ is performed according to a tree topology. The index $\Lambda_i$ is a collective index that determines the connection topology of the TTNS structure. $\Lambda_i$ connects to child nodes, and $a_i$ connects to the parent node. For instance, $\Lambda_i = \{a_{i-1}\}$ means an MPS. For a perfect binary tree with $2^M - 1$ tree nodes, $\Lambda_i = \{a_{2i-1}, a_{2i}\}$ for $i \le 2^{M-1}$ and $\Lambda_i = \emptyset$ otherwise. A schematic diagram for TTNS is shown in Fig. 1(a). In this paper, we denote the dimension of $a_i$ in a TTNS as $M_{\rm S}$ and the dimension of $\sigma_i$ as $d$. $M_{\rm S}$ and $d$ are called virtual and physical bond dimensions in the MPS language, respectively, and $M_{\rm S}$ represents the number of single-particle functions in the MCTDH language. Larsson recently provided a thorough description for the two sets of languages.[6] In principle, the dimension of $a_i$ and $\sigma_i$ can be different for different nodes in the tree. For simplicity, we assume the indices for different nodes share the same dimension unless otherwise specified. Thus, the size of $A[i]$ is $M_{\rm S}^k d$, where $k$ is the number of nodes connected to the $i$th node.

In Eq. (1), we have assumed that each node in the tree is associated with a physical degree of freedom. However, TTNS can also include entirely "virtual" nodes, as seen in ML-MCTDH or three-legged tree tensor network states,[16] which is not associated with any physical degree of freedom. For these virtual nodes, we can assign an auxiliary physical degree of freedom. This auxiliary degree of freedom has a Hilbert space of dimension 1, and the only permissible operator is the identity operator. By adopting this approach, the need for special treatment of these nodes is eliminated, and both the notation and the implementation are simplified.

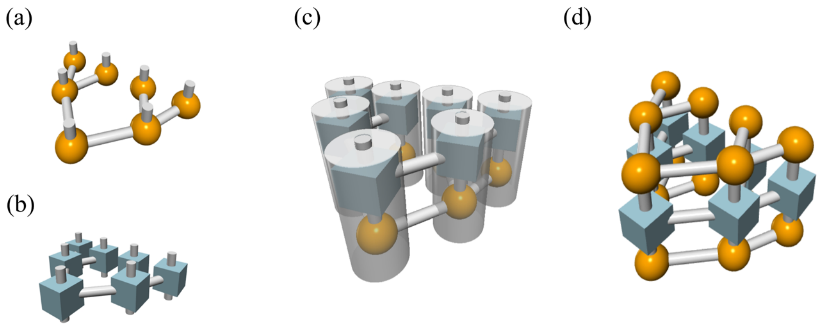

**FIG. 1.** Schematic diagram of TTNS and TTNO using a perfect binary tree as an example. Panels (a) and (b) show the schematic diagram for TTNS and TTNO, respectively. Each ball represents a node in TTNS, and each box represents a node in TTNO. The vertical cylinders represent physical indices. (c) The tensor network contraction diagram for applying a quantum operator represented by TTNO to a quantum state represented by TTNS. The tensors represented by the gray cylinders are contracted together to form a new TTNS. (d) The tensor network contraction diagram when calculating $\langle\Psi|\hat{O}|\Psi\rangle$, the expectation value of a physical observable.

Just as TTNS, a quantum operator $\hat{O}$ can be expressed as a TTNO,

$$\hat{O} = \sum_{\{a\},\{\sigma\},\{\sigma'\}} W[1]^{\sigma_1',\sigma_1}_{\Lambda_1,a_1} \cdots W[N]^{\sigma_N',\sigma_N}_{\Lambda_N,a_N} |\sigma_1' \cdots \sigma_N'\rangle\langle\sigma_1 \cdots \sigma_N|. \quad (2)$$

Each tensor $W[i]$ in Eq. (2) is a numeric tensor, expanded in the basis of $|\sigma_i\rangle$. The dimension of $a_i$ in a TTNO is denoted as $M_\mathrm{O}$, and the size of $W[i]$ is $M_\mathrm{O}^k d^2$. TTNO can also be expressed in a symbolic form

$$\hat{O} = \sum_{\{a\}} \hat{W}[1]_{\Lambda_1,a_1} \cdots \hat{W}[N]_{\Lambda_N,a_N}, \quad (3)$$

where $\hat{W}[i]$ is a tensor whose elements are symbolic operators. The size of $\hat{W}[i]$ is $M_\mathrm{O}^k$. A schematic diagram for TTNO is shown in Fig. 1(b). For later convenience, we define $\hat{W}[\sim i]$ based on the recurring relation,

$$\hat{W}[\sim i]_{a_i} = \sum_{\Lambda_i} \hat{W}[i]_{\Lambda_i,a_i} \prod_{j\in \mathrm{child}(i)} \hat{W}[\sim j]_{a_j}. \quad (4)$$

Here, $\mathrm{child}(i)$ refers to the indices for all the direct children of the $i$th tree node and $\Lambda_i = \{a_j | j \in \mathrm{child}(i)\}$. It should be noted that for leaf nodes, we have $\hat{W}[\sim i]_{a_i} = \hat{W}[i]_{\emptyset,a_i}$.

Similar to MPO/MPS, a TTNO can be applied to a TTNS through tensor contractions, resulting in a new TTNS, as shown in Fig. 1(c). This feature provides significant flexibility for manipulating TTNS and lays the foundation of a whole class of time evolution methods based on propagation and compression.[24,28,56] Another immediate advantage of using TTNO is the efficient computation of the physical observable $\langle\Psi|\hat{O}|\Psi\rangle$ through tensor network contraction. To achieve this, we stack $|\Psi\rangle$, $|\hat{O}\rangle$, and $\langle\Psi|$ in a three-layer manner, as shown in Fig. 1(d). The contraction process then begins from the leaves and moves inward toward the root. The computation cost scales polynomially with $M_\mathrm{S}$, $M_\mathrm{O}$, and $d$.

## B. Time evolution based on projector splitting

Based on time-dependent variational principle,[57] the formal solution of the time-dependent Schrödinger equation for tensor networks can be expressed as[58]

$$\frac{\partial|\Psi\rangle}{\partial t} = -i\hat{P}\hat{H}|\Psi\rangle. \quad (5)$$

Here, $\hat{P} = \sum_i^N \hat{P}_i^+ - \sum_i^{N-1} \hat{P}_i^-$ is a projection operator to the tangent space of the TTNS manifold. The exact form of $\hat{P}$ is provided in the literature.[6,59] At a short time step $\delta t$, the formal solution of the Schrödinger equation is

$$|\Psi\rangle(t_0 + \delta t) = e^{-i\hat{P}\hat{H}\delta t}|\Psi\rangle(t_0). \quad (6)$$

The projector splitting integrator decomposes the formal time evolution operator into the product of operators,

$$e^{-i\hat{P}\hat{H}\delta t} = \prod_i^{N-1} \{e^{-i\hat{P}_i^+\hat{H}\delta t} e^{i\hat{P}_i^-\hat{H}\delta t}\} e^{-i\hat{P}_N^+\hat{H}\delta t} + \mathcal{O}(\delta t^2). \quad (7)$$

Each term in the product corresponds to the time evolution of a local tensor in the TTNS. The complete time evolution consists of a sweep over the tree tensor network. For example, $e^{-i\hat{P}_i^+\hat{H}\delta t}$ is implemented as a local time evolution of $A[i]$ based on the effective Hamiltonian $\hat{H}_i^{\mathrm{eff}}$,

$$A[i](t_0 + \delta t) = e^{-i\hat{H}_i^{\mathrm{eff}}\delta t} A[i](t_0). \quad (8)$$

A schematic illustration of $\hat{H}_i^{\mathrm{eff}}A[i]$ is shown in Fig. 2(a). In Fig. 2, $A[i]$ is shown as the red ball, and $\hat{H}_i^{\mathrm{eff}}$ corresponds to the rest of the diagram. The term $e^{-i\hat{H}_i^{\mathrm{eff}}\delta t}A[i]$ can then be computed using a Krylov matrix exponential solver.

The computational bottleneck for the time evolution is the contraction between $A[i]$ and $\hat{H}_i^{\mathrm{eff}}$. During the contraction, the tensors in the gray containers shown in Fig. 2(b) are first contracted together, resulting in the contraction pattern shown in Fig. 2(c). The construction of the environment tensor is not the bottleneck of the

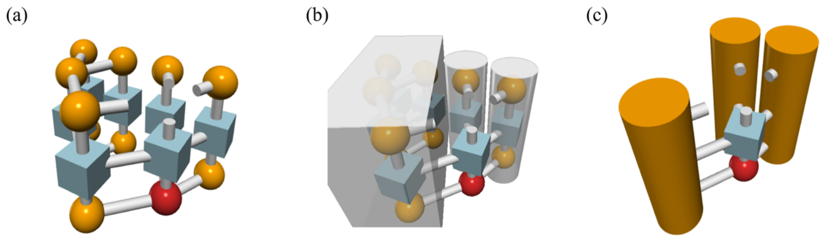

FIG. 2. Schematic diagram for the calculation of $\hat{H}_i^{\text{eff}} A[i]$ using TTNS and TTNO. (a) The red ball represents $A[i]$, and the rest of the diagram constitutes $\hat{H}_i^{\text{eff}}$. (b) Tensors indicated by the gray containers are contracted to construct the effective environment in $\hat{H}_i^{\text{eff}}$. Panel (c) shows the actual tensors involved in the computation of $\hat{H}_i^{\text{eff}} A[i]$, and the orange cylinders are environment tensors.

computation. During the sweep process, the environment tensors from the previous step can be employed to calculate the environment tensors required for the next step. The computation of the environment tensor has a lower scaling than computing $\hat{H}_i^{\text{eff}} A[i]$, which involves the contraction between $k$ environment tensors, $W[i]$, and $A[i]$. The size of the environment tensor is $M_{\text{S}} \times M_{\text{O}} \times M_{\text{S}} = M_{\text{S}}^2 M_{\text{O}}$. Recall that the size of $A[i]$ is $M_{\text{S}}^k d$ and the size of $W[i]$ is $M_{\text{O}}^k d^2$. The contraction between one of the environment tensor and $A[i]$ has computational cost $\mathcal{O}(M_{\text{S}}^{k+1} M_{\text{O}} d)$, and the resulting tensor is of size $M_{\text{S}}^k M_{\text{O}} d$. The tensor is subsequently contracted with $W[i]$ at a computational cost of $\mathcal{O}(M_{\text{S}}^k M_{\text{O}}^k d^2)$. The resulting tensor of size $M_{\text{S}}^k M_{\text{O}}^{k-1} d$ is then contracted with another environment tensor at a computational cost of $\mathcal{O}(M_{\text{S}}^{k+1} M_{\text{O}}^{k-1} d)$. The resulting tensor has a smaller size of $M_{\text{S}}^k M_{\text{O}}^{k-2} d$, and the rest of the contraction with other environment tensors has less computational cost. In all the cases discussed in this work, $M_{\text{O}}$ is less than 10, and $M_{\text{S}}$ ranges from 20 to 96. Meanwhile, most of $d$ is less than or equals to 10. Thus, we can assume $M_{\text{O}} \approx d \ll M_{\text{S}}$ and the overall computational scaling for the contraction of $\hat{H}_i^{\text{eff}} A[i]$ is $\mathcal{O}(M_{\text{S}}^{k+1} M_{\text{O}}^{k-1} d)$. The complete time evolution also requires the application of $e^{-i\hat{P}_i^- \hat{H}\,\delta t}$, which is similar to the application of $e^{i\hat{P}_i^+ \hat{H}\,\delta t}$ but with a reduced cost. In our implementation, we use a second-order symmetric Trotter decomposition of $e^{-i\hat{P}\hat{H}\,\delta t}$. For the complete algorithm of the time evolution, please refer to the cited papers.[59–61]

## C. Automatic construction of TTNO

In this section, we describe our algorithm for the automatic construction of TTNO based on bipartite graph theory. Our algorithm is developed based on the algorithm for the automatic construction of MPO.[45] Any quantum operator in the SOP form can be expressed as

$$\hat{O} = \sum_{o=1}^{K} \gamma_o \prod_{i=1}^{N} \hat{z}_i^{(o)}, \tag{9}$$

where $K$ is the number of terms, $N$ is the number of degrees of freedom, $\gamma_o$ stands for the coefficient for the $o$th term and $\hat{z}_i^{(o)}$ is the elementary operator for the $i$th degree of freedom for the $o$th term. It should be noted that $\hat{z}_i^{(o)}$ can be an identity operator. Tensor networks usually impose the commutation condition $[\hat{z}_i, \hat{z}_j] = 0$ for $i \neq j$. In Sec. III C, we discuss how to comply with the anti-commutation property for fermion operators. Table I presents the tabular form of $\hat{O}$, where each row represents a term in $\hat{O}$. We refer to Table I as the SOP table in the following discussion. The coefficients $\gamma_0$ can be stored in another table, which has the same number of rows as Table I but has only one column.

TABLE I. Tabular form of the SOP operator of Eq. (9). Each row represents a term in the operator, and each column corresponds to one of the degrees of freedom.

| | Degree of freedom | | | | |
|---|---|---|---|---|---|
| Term index | 1 | ... | $i$ | ... | $N$ |
| 1 | $\hat{z}_1^{(1)}$ | ... | $\hat{z}_i^{(1)}$ | ... | $\hat{z}_N^{(1)}$ |
| ... | ... | ... | ... | ... | ... |
| $o$ | $\hat{z}_1^{(o)}$ | ... | $\hat{z}_i^{(o)}$ | ... | $\hat{z}_N^{(o)}$ |
| ... | ... | ... | ... | ... | ... |
| $K$ | $\hat{z}_1^{(K)}$ | ... | $\hat{z}_i^{(K)}$ | ... | $\hat{z}_N^{(K)}$ |

In tree tensor networks, cutting an arbitrary edge will divide the tree into two parts. Suppose by cutting the $i$th edge, which connects the $i$th node to its parent, the degrees of freedom of the entire system are divided into two parts, namely, $L$ and $R$. It is then possible to express $\hat{O}$ as

$$\hat{O} = \sum_{o_i=1}^{K} \gamma_{o_i} \cdot \hat{O}_{i,L}^{(o_i)} \otimes \hat{O}_{i,R}^{(o_i)}, \tag{10}$$

where

$$\begin{aligned} \hat{O}_{i,L}^{(o_i)} &= \prod_{j \in \Omega^{i,L}} \hat{z}_j^{(o_i)}, \\ \hat{O}_{i,R}^{(o_i)} &= \prod_{j \in \Omega^{i,R}} \hat{z}_j^{(o_i)}. \end{aligned} \tag{11}$$

Here, $\Omega^{i,L(R)}$ stands for the set of degrees of freedom that fall into the $L(R)$ part after cutting the $i$th edge. Based on Eq. (10), $M_{\text{O}}$ for the TTNO at the $i$th edge is $K$, which showcases a straightforward construction of TTNO that is not optimal.

The key to the construction of optimal TTNO is to exploit redundancy in $\{\hat{O}^{(o_i)}_{i,L(R)} | o_i \in [1, K]\}$. The unique operators in $\{\hat{O}^{(o_i)}_{i,L(R)} | o_i \in [1, K]\}$ are denoted as $\mathcal{U}^{i,L(R)}$, and $\hat{O}$ can be written as

$$\hat{O} = \sum_{jk} \Gamma_{jk} \mathcal{U}^{i,L}_{j} \otimes \mathcal{U}^{i,R}_{k}, \tag{12}$$

where Γ is the coefficient matrix on the basis of $\mathcal{U}^{i,L} \otimes \mathcal{U}^{i,R}$. A natural next step is to perform Schmidt decomposition to reduce the dimension and obtain a more efficient TTNO, similar to the idea of TTNS. The computational scaling of this approach is $\mathcal{O}(K^3)$ since Γ is approximately of size $K \times K$. The $M_\mathrm{O}$ at the $i$th edge equals the rank of Γ. Additional truncation based on singular values can be performed to compress the TTNO.[62]

In addition to the Schmidt decomposition method, better scaling can be obtained by leveraging the sparse nature of Γ using the bipartite graph theory. Γ is sparse since it only contains $K$ non-zero elements. A bipartite graph is a type of graph where all vertices can be divided into two distinct sets, and every edge connects a vertex in one set to a vertex in the other set. In this context, $\mathcal{U}^{i,L}$ and $\mathcal{U}^{i,R}$ are considered as the two sets of vertices in the bipartite graph, and their interactions represented by nonzero elements in Γ are considered as the graph edges. Our goal is to find the minimum number of vertices that can connect to all edges in the graph, a problem known as the minimum vertex cover problem in graph theory. The minimum $M_\mathrm{O}$ at the $i$th edge equals the number of these vertices, by constructing complementary operators[63] for the vertices. This minimum number of vertices can be efficiently found in polynomial time with respect to the number of vertices and edges. Specifically, since there are $K$ vertices and $K$ edges, the computational scaling is $\mathcal{O}(K^{\frac{3}{2}})$ using the Hopcroft–Karp algorithm.[64] This is an improvement over the $K^3$ scaling using SVD. Solving the minimum vertex cover problem for the $i$th edge determines the construction of optimal TTNO for the $i$th node by constructing complementary operators according to the solution of the minimum vertex cover problem. The entire TTNO is constructed by iterating over all of the nodes in the tree. For the simulation of open quantum systems, such as the spin-boson model and the transport model studied in this paper, the number of terms in the Hamiltonian $K$ is proportional to the number of nodes and edges in the tree tensor network $N$. Therefore, the overall scaling for the construction of the complete TTNO is $N^{\frac{3}{2}} \times N = N^{\frac{5}{2}}$.

The construction of MPO/TTNO based on the bipartite graph approach does not rely on the coefficients of the SOP operator. In other words, with different but finite $\gamma_o$, the MPO/TTNO has the same $M_\mathrm{O}$ for the same operator. The approach of manual MPO/TTNO design also shares this feature. However, the same is not true for the SVD compression approach. If $\gamma_o$ exhibits a particular pattern, it is possible to obtain a more compact MPO/TTNO based on SVD compression without numerical compression error. Consider, for example, that Γ is a $2 \times 2$ matrix and all matrix elements are 1. Thus, the minimum number of vertices to connect to all edges is 2. Meanwhile, SVD over Γ yields only one non-zero singular value. Such a pattern is hardly encountered in chemical models in practice. However, for lattice models encountered in physics, most Hamiltonian terms share the same coefficient. In such cases, SVD or QR over Γ is preferred for obtaining the optimal MPO/TTNO.

**TABLE II.** SOP table of the Hamiltonian of the spin-boson model in Eq. (13) with $N_b = 2$, as a special case of Table I.

| | Spin | $v_1$ | $v_2$ |
|---|---|---|---|
| 1 | $\hat{\sigma}^x$ | $\hat{I}$ | $\hat{I}$ |
| 2 | $\hat{I}$ | $\hat{p}_1^2$ | $\hat{I}$ |
| 3 | $\hat{I}$ | $\hat{q}_1^2$ | $\hat{I}$ |
| 4 | $\hat{I}$ | $\hat{I}$ | $\hat{p}_2^2$ |
| 5 | $\hat{I}$ | $\hat{I}$ | $\hat{q}_2^2$ |
| 6 | $\hat{\sigma}^z$ | $\hat{q}_1$ | $\hat{I}$ |
| 7 | $\hat{\sigma}^z$ | $\hat{I}$ | $\hat{q}_2$ |

To illustrate the concept of the SOP table, bipartite graph, and minimum vertex cover, let us consider a spin-boson model where a spin is coupled with a set of vibrational modes. The Hamiltonian for the spin-boson model is given by

$$\hat{H} = \Delta\hat{\sigma}^x + \frac{1}{2}\sum_{i=1}^{N_b}(\hat{p}_i^2 + \omega_i^2\hat{q}_i^2) + \hat{\sigma}^z\sum_{i=1}^{N_b} c_i\hat{q}_i, \tag{13}$$

which follows the sum-of-product form. Here, $\hat{p}$ and $\hat{q}$ are the momentum and coordinate operators of the vibration modes, respectively. $\omega$ is the vibration frequency. $c$ is the coupling constant. $\hat{\sigma}^x$ and $\hat{\sigma}^z$ are Pauli matrices for the spin. Δ is the tunneling constant. For ease of demonstration, we limit $N_b = 2$, and generalization to more complex cases is straightforward. The two vibration modes are denoted as $v_1$ and $v_2$. We consider a simple tree, where the spin node is the root and the vibration nodes are the two leaves.

We can first construct the SOP table for Eq. (13) as presented in Table II. Next, assume that spin and $v_1$ are in the $L$ part while $v_2$ is in the $R$ part, as indicated by the vertical line presented in Table II. The operators in $\Omega^L$ and $\Omega^R$ are redundant. For example, the unique operators in $\Omega^R$, denoted as $\mathcal{U}^R$, are $\{\hat{I}, \hat{p}_2^2, \hat{q}_2^2, \hat{q}_2\}$. Meanwhile, $\Omega^L$ has six unique operators. Then, Γ is a $6 \times 4$ matrix with seven nonzero elements. SVD over Γ yields three nonzero singular values, which means the minimum $M_\mathrm{O}$ of TTNO is 3.

Figure 3 shows the bipartite graph representation of Table II. The vertices on the left(right) side are $\mathcal{U}^{L(R)}$, and each edge represents a term presented in Table II. For optimal TTNO, it is necessary to select a minimum set of vertices in the graph that covers all edges. Such a task can be solved efficiently using the Hopcroft–Karp algorithm. By selecting $\hat{I} \otimes \hat{I}$, $\hat{\sigma}^z \otimes \hat{I}$ in the $L$ set and $\hat{I}$ in the $R$ set, i.e., operators in round-corner boxes, all the edges are covered. Complementary operators are then constructed based on the edge connections. For example, since $\hat{I} \otimes \hat{I}$ is connected to $\hat{p}_2^2$ and $\hat{q}_2^2$, these two are combined to form a complementary operator $\frac{1}{2}\hat{p}_2^2 + \frac{1}{2}\omega_2^2\hat{q}_2^2$ for $\hat{I} \otimes \hat{I}$. This allows us to determine the TTNO tensor $\hat{W}[v_2]$ for the node representing $v_2$. The tensor has only one virtual index, and its symbolic form is

$$\hat{W}[v_2] = \begin{bmatrix} \hat{I} & \frac{1}{2}\hat{p}_2^2 + \frac{1}{2}\omega_2^2\hat{q}_2^2 & c_2\hat{q}_2 \end{bmatrix}. \tag{14}$$

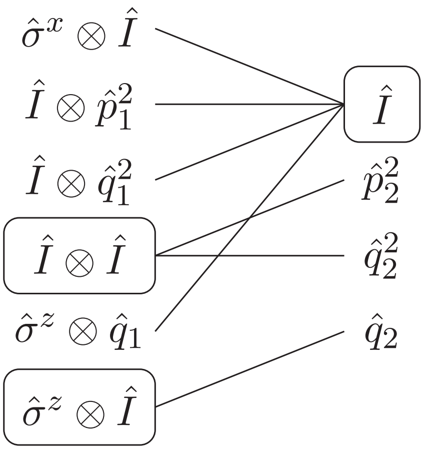
$\hat{\sigma}^x \otimes \hat{I}$
$\hat{I} \otimes \hat{p}_1^2$
$\hat{I} \otimes \hat{q}_1^2$
$\hat{I} \otimes \hat{I}$
$\hat{\sigma}^z \otimes \hat{q}_1$
$\hat{\sigma}^z \otimes \hat{I}$
$\hat{I}$
$\hat{p}_2^2$
$\hat{q}_2^2$
$\hat{q}_2$

**FIG. 3.** Bipartite graph for SOP operators taking the Hamiltonian of the spin-boson model with $N_b = 2$ as an example. The operators acting on the spin and $v_1$ are grouped on the left and the operators acting on $v_2$ are grouped on the right. Each edge represents a term in the Hamiltonian, and the operators in boxes are a minimal set of vertices that covers all edges.

In order to determine the whole TTNO, we additionally bipartite the system by cutting the edge between the spin node and the $v_1$ node. Then, the left part of the system contains $v_1$ and the right part contains the spin and $v_2$. Similarly, $\hat{W}[v_1]$ is determined as

$$\hat{W}[v_1] = \begin{bmatrix} \hat{I} & \frac{1}{2}\hat{p}_1^2 + \frac{1}{2}\omega_1^2\hat{q}_1^2 & c_1\hat{q}_1 \end{bmatrix}. \tag{15}$$

Since the spin node is the root node, its symbolic TTNO tensor $\hat{W}[\text{spin}]$ can be determined from $\hat{W}[\sim v_1] = \hat{W}[v_1]$, $\hat{W}[\sim v_2] = \hat{W}[v_2]$, and the overall SOP Hamiltonian $\hat{H}$. $\hat{W}[\text{spin}]$ is then,

$$\hat{W} = \begin{bmatrix} \Delta\hat{\sigma}^x & I & \hat{\sigma}^z \\ I & 0 & 0 \\ \hat{\sigma}^z & 0 & 0 \end{bmatrix}. \tag{16}$$

In the specific case of the spin-boson model, it can be shown that $M_{\text{O}} = 3$ regardless of the number of discretized modes. Consequently, for fixed $M_{\text{S}}$ and $d$, the computational cost for updating one of the tensors $A[i]$ is independent of the number of modes. Therefore, the total computational time for a single time evolution step scales linearly with the number of nodes in the tree or the number of modes in the system.

Finally, we outline the general procedure to construct TTNO using the bipartite graph theory step by step. The algorithm visits all nodes in the post-order, meaning that if the $i$th node is visited, all its children have already been visited and $\hat{W}[\sim j]$ for all the children have been constructed. For the $i$th node in the tree in the post-order, the following steps should be performed:

1. Construct the SOP table of the target operator $\hat{O}$. Consider elementary operators from all the children, as provided by $\hat{W}[\sim j]$, at the current node, as well as the rest of the system. If the current node does not have children, identity operators can be taken as placeholders for the elementary operators from the children.
2. Divide the SOP table into two parts. The left part contains elementary operators from all children $\hat{z}^{(o)}_{\text{child}}$ and the current node $\hat{z}^{(o)}_{\text{node}}$. The right part contains all operators from the rest of the system $\hat{z}^{(o)}_{\text{others}}$. Identify the unique operators $\mathcal{U}^{L(R)}$ and use them to construct the bipartite graph. Then, solve the minimum vertex cover problem using the Hopcroft–Karp algorithm.
3. For each operator in $\mathcal{U}^L$, if $\mathcal{U}^L_k$ is included in the solution of the minimum vertex cover problem, append $\mathcal{U}^L_k$ as a new row of $\hat{W}[\sim i]$. For each operator in $\mathcal{U}^R$, if $\mathcal{U}^R_k$ is included in the solution of the minimum vertex cover problem, add the operators in $\mathcal{U}^L$ that are connected to $\mathcal{U}^R_k$ and append the summed operator as a new row of $\hat{W}[\sim i]$. It should be noted that if an edge is connected to both $\mathcal{U}^L$ and $\mathcal{U}^R$, it should only be included once.
4. Using $\hat{W}[\sim i]$ and $\hat{W}[\sim j]$ for all the child nodes of the $i$th node, construct $\hat{W}[i]$ based on the recurring relation in Eq. (4). $\hat{W}[\sim i]$ are then considered as elementary operators for the parent of the $i$th node.
5. Proceed to the next node following the post-order sequence.

## III. NUMERICAL RESULTS

In this section, we present numerical results based on our TTNS algorithm with optimal TTNO. First, we will numerically demonstrate the computational scaling of our algorithm, highlighting how the computational cost scales linearly with the number of degrees of freedom in the spin-boson model. Subsequently, we will showcase two example simulations of open quantum systems to illustrate the versatility and efficiency of our algorithm.

### A. Computational scaling

We first show the computational scaling of our algorithm through numerical simulation for the spin relaxation dynamics of the spin-boson model. The Hamiltonian of the model is presented in Eq. (13). The coupling between the spin and the vibrational environment is specified by the spectral density function,

$$J(\omega) = \frac{\pi}{2}\sum_j \frac{c_j^2}{\omega_j}\delta(\omega - \omega_j). \tag{17}$$

In this section, we focus on the sub-Ohmic spectral density function,

$$J_{\text{Ohmic}}(\omega) = \frac{\pi}{2}\alpha\omega^s\omega_c^{1-s}e^{-\omega/\omega_c}. \tag{18}$$

Here, $\alpha$ is the dimensionless Kondo parameter that controls the strength of the system–bath coupling. $\omega_c$ is the characteristic frequency of the bath. $s$ controls the shape of the spectral density. A value of $0 < s < 1$ corresponds to sub-Ohmic spectral density, $s = 1$ to Ohmic spectral density, and $s > 1$ to super-Ohmic spectral density. For our purposes, we set $s = 0.5$ and $\omega_c = 20\Delta$. $\alpha$ is set to 0.05 unless otherwise specified. The vibration modes are discretized based on a particular density of states $\rho(\omega)$,

$$\int_0^{\omega_j} \rho(\omega)d\omega = j, \quad j = 1, \ldots, N_b, \tag{19}$$

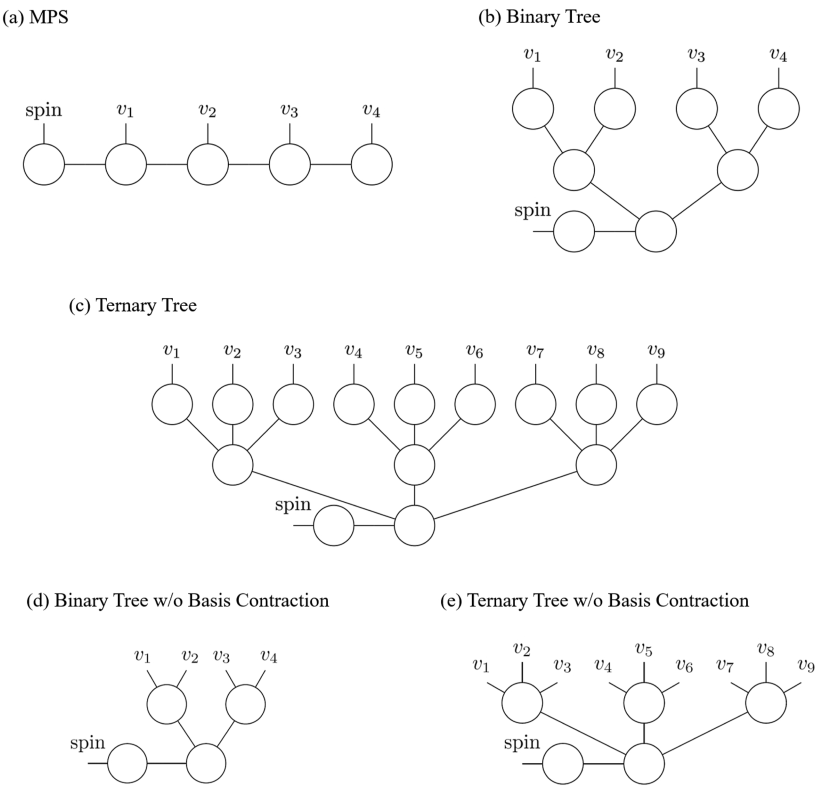

**FIG. 4.** Tree tensor network structures used in this work. (a) Linear chain topology, which is identical to MPS. (b) Binary tree topology. (c) Ternary tree topology. (d) Binary tree topology without primitive basis contraction. (e) Ternary tree topology without primitive basis contraction.

where $N_b$ is the number of bath modes. The density of states $\rho(\omega)$ is defined as

$$\rho(\omega) = \frac{N_b + 1}{\omega_c} e^{-\omega/\omega_c}. \tag{20}$$

We use 10 harmonic oscillator eigenbasis for all the vibration modes ($d = 10$). This setup is chosen for the scaling benchmark to facilitate direct comparison with the results from the literature.[13,24,65] Benchmark results with $\alpha$ from 0.05 to 1.0 are shown in Fig. 12 in the Appendix, aligning perfectly with previous reports.

We next describe the TTNS tree structure employed for the simulation. Our algorithm and its implementation can work with any tree structure, regardless of tree depth, and the tree structure we have used here is chosen for its simplicity. It is important to note that these structures may not necessarily be the optimal tree structure for the models that we are studying. We first look at the MPS structure, where all degrees of freedom are arranged in a linear chain. The spin is positioned at the start of the chain, followed by the vibration modes in ascending order of vibration frequency. A visual representation of the MPS structure is shown in Fig. 4(a). We then describe two types of tree topologies that are inspired by ML-MCTDH. These are shown in Figs. 4(b) and 4(c). In both types of trees, the vibration degrees of freedom are first grouped to form either a binary or ternary tree. The spin is then attached to the root of the tree. In both trees, only the leaf node has physical indices. In Figs. 4(b) and 4(c), each primitive mode is first contracted to $M_\text{S}$ states before being connected to the next layer. The actual tree structure employed in our simulation is a bit more subtle. More specifically, the contraction of the primitive modes shown in Figs. 4(b) and 4(c) applies when $M_\text{S} < d$. When $M_\text{S} > d$, such contraction is unnecessary and two or three primitive modes are directly linked to the leaf node, as shown in Figs. 4(d) and 4(e). In the tree structure, the number of layers is approximately $\log N_b$. For all the structures considered, the number of nodes is on the same order as $N_b$ or the number of degrees of freedom in the model.

We first show the wall time required for one step of TDVP-PS time evolution vs the number of modes $N_b$ in the model shown in Fig. 5. We use the binary tree shown in Fig. 4(b), with $M_\text{S}$ set to 20 and the time evolution step size at $\delta t\Delta = 0.05$. The simulation is run for 10 steps, and the average wall time per step is reported. The computation is carried out on a single core of Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50 GHz (the same hereinafter). Simulations up to $N_b = 8096$ reveal that the wall time required to perform a single step of the time evolution scales linearly with the number of bath modes $N_b$ in the model. This favorable scaling is realized by constructing the Hamiltonian as TTNO with a constant $M_\text{O}$ of 3. When $N_b = 8096$, the average wall time per step is ~800 s. Since simulation to $t\Delta = 40$ requires 800 steps, a complete simulation with $N_b = 8096$

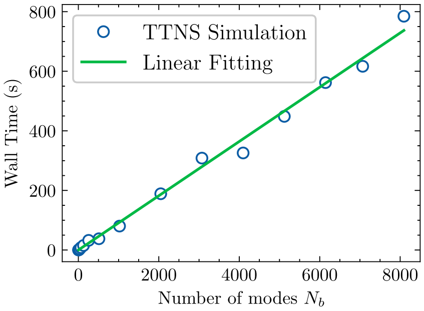

**FIG. 5.** Wall time for one step of the TDVP-PS time evolution based on the spin-boson model as a function of the number of modes $N_b$. $M_S = 20$, $d = 10$, and $\delta t\Delta = 0.05$. The solid line is a linear fitting of the collected data.

would take roughly one week. The mode combination technique in ML-MCTDH is not employed in this case.

Next, we examine how different tree structures, as well as the values of $M_S$ and $d$, affect the computational cost. In Fig. 6(a), we show the computational wall time vs $N_b$ with different tree structures. The MPS tree structure and the binary tree structure have almost identical computational costs. This is because, with $M_S > d$, each leaf node has 2 physical indices in the binary tree. Thus, both MPS and the binary tree have approximately the same number of nodes. In addition, in both MPS and binary tree, most nodes have three indices (virtual and physical combined). This is why the two different tree structures have almost the same computational cost. In the ternary tree, most nodes have 4 physical and virtual indices, leading to a significantly higher computational cost compared to the other two cases. In Fig. 6(b), we show the computational wall time vs $M_S$. As analyzed earlier in Sec. II B, the computational cost scales as $\mathcal{O}(M_S^{k+1})$ with respect to $M_S$ where $k$ is the number of nodes connected to each node. Clearly, for MPS $k = 2$, for the binary tree $k = 3$, and for the ternary tree $k = 4$. The computational scaling is consistent with our analysis shown in Fig. 6. In Fig. 6(c), we show the computational wall time vs the number of primitive basis $d$. According to our analysis in Sec. II B, the computational cost scales as $\mathcal{O}(d)$ when $d$ is small and $\mathcal{O}(d^2)$ when $d$ is large. However, this analysis assumes that each node in the TTNS has a physical index with dimension $d$, which is true for MPS but not true for the tree networks employed here. For binary and ternary trees, when $d$ is large, the $d$ primitive basis is first contracted to the $M_S$ states at the leaf node. Thus, up to a reasonably large $d$, the computational cost with respect to these leaf nodes is negligible compared to the computation of the body nodes, which have more indices than the leaf nodes. For the cases considered in this work, TTNO and MPO have similar bond dimensions. The key advantage of using TTNO is that they can reduce the computing scaling over the number of primitive basis $d$ to a constant. For comparison, the MPS/MPO approach has $\mathcal{O}(d^2)$ scaling. This significant improvement in scaling justifies the development and use of TTNS/TTNO approaches, especially when $d$ is large, despite their increased coding complexity. This advantage helps explain why MPS is predominantly used in electronic structure calculations, while TTNS is more commonly employed in quantum dynamics simulations. These findings are consistent with previous reports.[6,16–18]

## B. Spin relaxation dynamics of the spin-boson model

We next apply our algorithm to the spin relaxation dynamics of the spin-boson model using the Cole–Davidson spectral density,[66]

$$J_{\mathrm{CD}}(\omega) = \eta \frac{\sin \beta_{\mathrm{CD}} \theta_{\mathrm{CD}}}{\left[1 + (\omega^2/\omega_c^2)\right]^{\beta_{\mathrm{CD}}/2}},$$
$$\theta_{\mathrm{CD}} = \arctan \frac{\omega}{\omega_c}, \tag{21}$$

where $\beta_{\mathrm{CD}} < 1$ is the fractional stretching exponent. The Cole–Davidson spectral density function exhibits distinct behavior at low frequency and high frequency limits. When $\omega \ll \omega_c$, $J_{\mathrm{CD}} \approx \eta\beta_{\mathrm{CD}}\frac{\omega}{\omega_c}$. When $\omega \gg \omega_c$, $J_{\mathrm{CD}} \approx \eta \sin\left(\frac{\pi\beta_{\mathrm{CD}}}{2}\right)\left(\frac{\omega_c}{\omega}\right)^{\beta_{\mathrm{CD}}}$, which shows slow power law decay. Despite the complicated form of $J_{\mathrm{CD}}$, the reorganization energy $\lambda$ is simply related to $\eta$ by $\lambda = 2\eta$. The key difference between $J_{\mathrm{CD}}$ and $J_{\mathrm{Ohmic}}$ is that $J_{\mathrm{CD}}$ has a long tail at the high-frequency region, particularly when $\beta_{\mathrm{CD}}$ is small. The contribution from the high frequency part can be taken into account through a Born–Oppenheimer type approximation, resulting in a modified $\Delta$,

$$\Delta^{\mathrm{eff}} = \Delta \exp\left\{-\frac{2}{\pi}\int_{\omega_q}^{\infty} \frac{J_{\mathrm{CD}}(\omega)}{\omega^2} d\omega\right\}, \tag{22}$$

where $\omega_q$ is the cutoff frequency during mode discretization. The density of states for mode discretization is chosen as

$$\rho(\omega) \propto \frac{J_{\mathrm{CD}}(\omega)}{\omega}. \tag{23}$$

The other setups, unless otherwise specified, are the same as those in Sec. III A. A binary tree as shown in Fig. 4(b) is used for TTNS topology. We set $M_S = 20$ and $N_b = 1000$. For low-frequency modes, $d$ ranges from dozens to hundreds, and for high-frequency modes, $d$ is chosen as 4. We first validate our algorithm by reproducing the previous ML-MCTDH results, shown in Fig. 13 in the Appendix.

The finite temperature dynamics of the spin relaxation is obtained through thermo-field dynamics.[67–70] In this method, the finite temperature density matrix of the bath environment is transformed into a pure state by introducing an auxiliary space $Q$, analogous to the physical space $P$. The finite temperature dynamics is then reduced to the zero temperature dynamics with a transformed Hamiltonian,

$$\hat{\tilde{H}} = \Delta\hat{\sigma}^x + \frac{1}{2}\sum_i \left(\hat{p}_i^2 + \omega_i^2\hat{q}_i^2\right) - \frac{1}{2}\sum_i \left(\hat{\tilde{p}}_i^2 + \omega_i^2\hat{\tilde{q}}_i^2\right)$$
$$+ \hat{\sigma}^z\sum_i c_i \cosh\theta_i\hat{q}_i + \hat{\sigma}^z\sum_i c_i \sinh\theta_i\hat{\tilde{q}}_i, \tag{24}$$

where operators with a tilde "~" are operators in the $Q$ space and $\theta_i = \operatorname{arctanh}\exp\left\{-\frac{\omega_i}{2k_BT}\right\}$. The initial thermal state of the bath is represented by $|0\rangle_P|0\rangle_Q$ in the harmonic oscillator eigenbasis.

We next provide more details on the automatically constructed TTNO for the spin-boson model. The tree tensor network follows a binary tree topology, as shown in Fig. 4(b) and reproduced in Fig. 7. The tree only shows 4 vibration modes for simplicity, but our actual simulation uses 1000 modes. For reference, each node is labeled with an index.

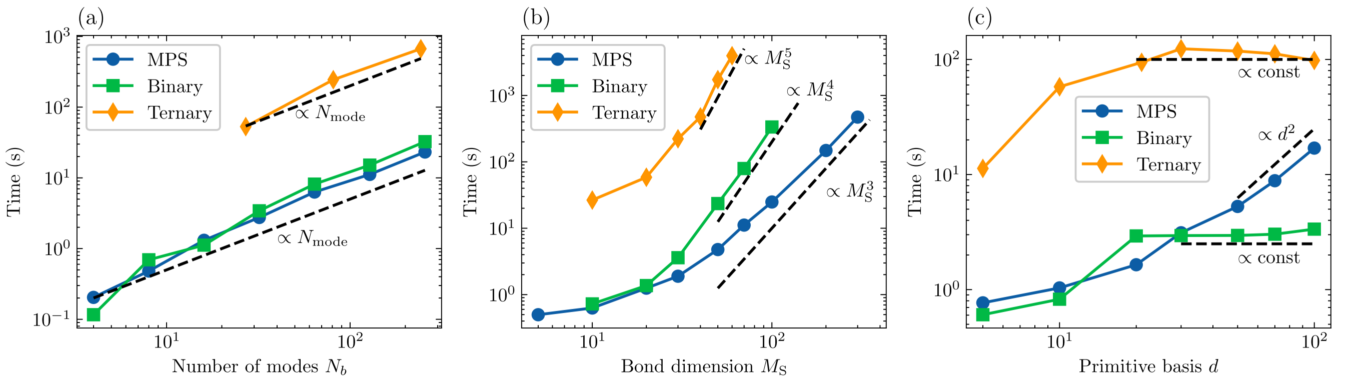

**FIG. 6.** Computational cost of single evolution step using MPS, binary TTNS, and ternary TTNS. All the calculations are performed on a single intel xeon(R) platinum 8255C CPU @ 2.50 GHz core. (a) $M_S = 20$, $d = 10$, and $N_b$ varies. (b) For MPS and binary TTNS $N_b = 16$ and for ternary TTNS $N_b = 27$; $d = 10$ and $M_S$ varies. (c) For MPS and binary TTNS $N_b = 16$ and for ternary TTNS $N_b = 27$; $M_S = 20$ and $d$ varies. The black dashed lines are guides to indicate the polynomial scaling.

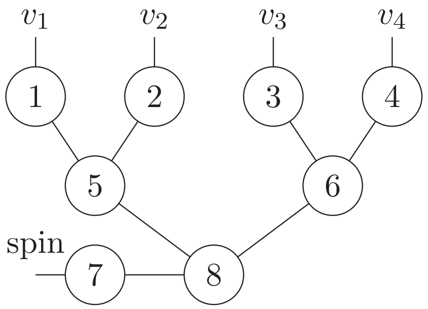

**FIG. 7.** Schematic diagram for the tree nodes in the spin-boson model. Each node is labeled with an index.

We then show the symbolic tensors $\hat{W}[i]$ for the corresponding TTNO. For the leaf nodes ($i = 1, 2, 3, 4$), $\hat{W}[i]$ has only one index,

$$\hat{W}[i] = \begin{bmatrix} \hat{I}_i \\ \frac{1}{2}\hat{p}_i^2 + \frac{1}{2}\omega_i^2\hat{q}_i^2 \\ c_i\hat{q}_i \end{bmatrix}, \quad i = 1, 2, 3, 4. \tag{25}$$

For the body nodes $(i = 5, 6)$, $\hat{W}[i]$ has three indices and the shape is $(3, 3, 3)$. The first two indices connect to the children, and the last index connects to the parent. Since the body nodes are not associated with any physical degree of freedom, the possible matrix elements for $\hat{W}[i]$ are 0 and 1, and $\hat{W}[i]$ is constructed as

$$\hat{W}[i]_{jk} = \begin{bmatrix} \delta_{j1}\delta_{k1} \\ \delta_{j1}\delta_{k2} + \delta_{j2}\delta_{k1} \\ \delta_{j1}\delta_{k3} + \delta_{j3}\delta_{k1} \end{bmatrix}, \quad i = 5, 6. \tag{26}$$

Here, $j$ and $k$ are indices to the children, and the index to the parent is shown as the vector. The vector elements correspond to the identity operator, the vibration energy, and the vibration coordinate for the vibrations, respectively.

The local TTNO tensor $\hat{W}$ for the spin node is

$$\hat{W}[7] = \begin{bmatrix} \hat{\sigma}^z \\ \hat{\sigma}^x \\ \hat{I} \end{bmatrix}. \tag{27}$$

The final virtual node with $i = 8$ has three indices $jkl$, connecting to nodes 5, 6, and 7, respectively. $\hat{W}[8]$ is then

$$\hat{W}[8]_{jkl} = \delta_{j1}\delta_{k1}\delta_{l2} + \delta_{j1}\delta_{k3}\delta_{l1} + \delta_{j3}\delta_{k1}\delta_{l1} + \delta_{j1}\delta_{k2}\delta_{l3} + \delta_{j2}\delta_{k1}\delta_{l3}. \tag{28}$$

Regarding the treatment of the auxiliary space for finite temperature dynamics simulation, the transformed Hamiltonian Eq. (21) has the same form as the spin-boson Hamiltonian in Eq. (13), despite that the number of vibration modes is doubled. Consequently, the constructed TTNO at finite temperature is identical to the TTNO at zero temperature. Because the computational cost scales linearly with the number of modes in the spin-boson model, we conclude that the finite temperature computation is twice as costly as zero temperature computation.

Figure 8 shows the simulated spin relaxation dynamics using the Cole–Davidson spectral density. In Fig. 8(a), we show the influence of the characteristic frequency $\omega_c$ on the spin relaxation dynamics, with $\eta/\Delta = 10$ and $\beta_{\rm CD} = 0.5$. Notably, when the vibration frequency and tunneling constant are comparable (i.e., $\omega_c/\Delta = 1$), the spin relaxation exhibits the most incoherent behavior compared to other cases. In addition, due to the strong system–bath coupling, $\langle\hat{\sigma}^z\rangle$ demonstrates localization in both the adiabatic and the intermediate regimes. We then explore the impact of temperature on the dynamics shown in Fig. 8(b), with $\eta/\Delta = 1$, $\beta_{\rm CD} = 0.25$, and $\omega_c/\Delta = 1$. As the temperature increases, the spin dynamics become increasingly incoherent. For this figure, we employ $N_b = 500$, including both $P$ and $Q$ space and $M_S = 96$ due to the strong entanglement at finite temperature. Finally, we study the effect of $\beta_{\rm CD}$ on the spin relaxation dynamics. As shown in Fig. 8(c), the dynamics of the spin becomes more incoherent with higher values of $\beta_{\rm CD}$. The other parameters are $\eta/\Delta = 2.5$ and $\omega_c/\Delta = 1$. We note that a lower value of $\beta_{\rm CD}$ enhances the contribution of high-frequency vibration modes. Up to $t\Delta = 20$, localization is not observed for the values of $\beta_{\rm CD}$ considered.

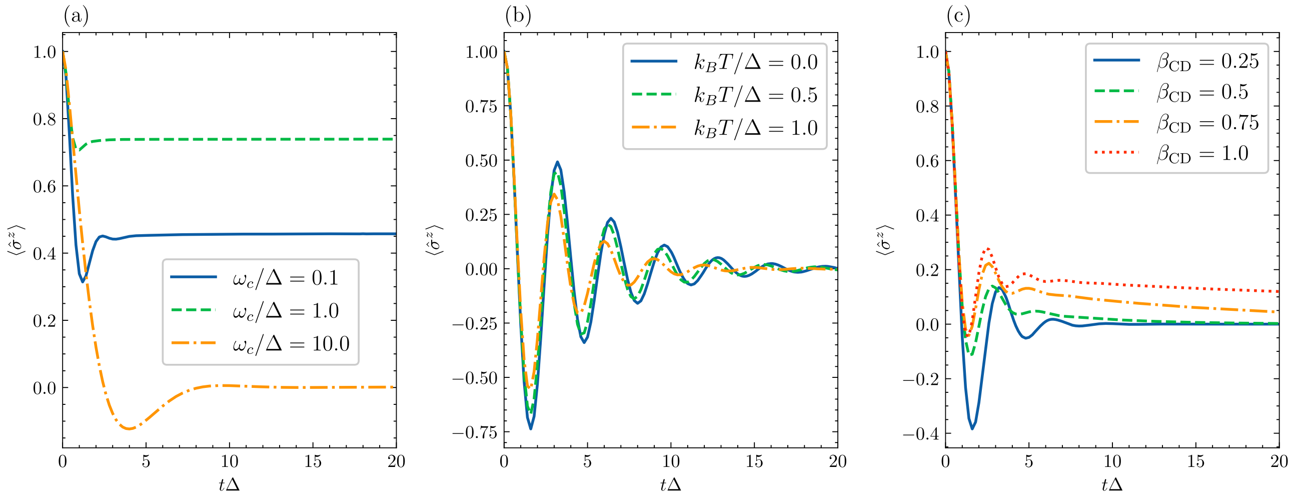

**FIG. 8.** Spin relaxation dynamics based on the Cole–Davidson spectral density. (a)$\eta/\Delta = 10$, $\beta_{\rm CD} = 0.5$, and $k_BT/\Delta = 0$. (b) $\eta/\Delta = 1$, $\beta_{\rm CD} = 0.25$, and $\omega_c/\Delta = 1$. (c) $\eta/\Delta = 2.5$, $\omega_c/\Delta = 1$, and $k_BT/\Delta = 0$.

### C. Charge transport in a molecular junction

We then move on to study quantum charge transport in a molecule junction with vibrational coupling. Single molecular junction has been widely used to investigate nonequilibrium many-body quantum phenomena at the nanoscale.[71–74] The model for vibrationally coupled electron transport through a single-molecule junction can usually be divided into three parts.[12,40,75,76] The first part is the electronic part, which includes one discrete electronic state at the molecular bridge and two identical electronic continua describing the left and the right metal leads,

$$\hat H_{\rm el} = E_d\hat d^\dagger\hat d + \sum_{\zeta=L,R}\sum_{k_\zeta} E_{k_\zeta}\hat c^\dagger_{k_\zeta}\hat c_{k_\zeta} + \sum_{\zeta=L,R}\sum_{k_\zeta} V_{dk_\zeta}\left(\hat d^\dagger\hat c_{k_\zeta} + \hat c^\dagger_{k_\zeta}\hat d\right). \tag{29}$$

Here, $\hat d^\dagger(\hat d)$ and $\hat c^\dagger(\hat c)$ are fermionic creation (annihilation) operators for the electronic state on the molecular bridge and the metal leads, respectively. The electronic states in the left (right) lead state are indexed with $k_L$ and $k_R$, respectively. It should be noted that the partition between the left and right leads should not be confused with the partition in the bipartite graph in Sec. II C. $E_d$ is the site energy of the molecular bridge and is set to 0 throughout this section. $E_{k_\zeta}$ is the energy of the lead states, and $V_{dk_\zeta}$ is the molecule-lead coupling strength, which will be provided later. Since $\hat d^\dagger(\hat d)$ and $\hat c^\dagger(\hat c)$ have to follow the anti-commutation property of fermionic operators, they are transformed to spin operators through the Jordan–Wigner transformation,[77]

$$\begin{aligned}\hat a^\dagger_k &= \prod_{j=1}^{k-1}\hat\sigma^z_j\hat\sigma^+_k,\\ \hat a_k &= \prod_{j=1}^{k-1}\hat\sigma^z_j\hat\sigma^-_k,\end{aligned} \tag{30}$$

where $\hat\sigma^+ = \frac{1}{2}(\hat\sigma^x - i\hat\sigma^y)$ and $\hat\sigma^- = \frac{1}{2}(\hat\sigma^x + i\hat\sigma^y)$ are Pauli ladder operators.

The molecular bridge is then coupled to a phonon bath,

$$\begin{aligned}\hat H_{\rm el-nuc} &= \hat d^\dagger\hat d\sum_j 2c_j\hat q_j,\\ \hat H_{\rm nuc} &= \frac{1}{2}\sum_j\left(\hat p_j^2 + \omega_j^2\hat q_j^2\right).\end{aligned} \tag{31}$$

The total Hamiltonian is then written as

$$\hat H = \hat H_{\rm el} + \hat H_{\rm el-nuc} + \hat H_{\rm nuc}. \tag{32}$$

The primary physical observable for quantum transport through a molecular junction is the electronic current for a given source-drain bias voltage. The electronic current operator for each lead reads

$$\hat I_\zeta = i[\hat H, \hat N_\zeta] = i\sum_{k_\zeta} V_{dk_\zeta}\left(\hat d^\dagger\hat c_{k_\zeta} - \hat c^\dagger_{k_\zeta}\hat d\right), \quad \zeta = L, R, \tag{33}$$

where $\hat N_\zeta = \sum_{k_\zeta}\hat c^\dagger_{k_\zeta}\hat c_{k_\zeta}$ is the occupation number operator for each lead. The overall time-dependent current $I(t)$ is calculated with

$$I(t) = \frac{1}{2}\left[\langle\hat I_R(t)\rangle - \langle\hat I_L(t)\rangle\right]. \tag{34}$$

The source-drain bias voltage is considered through different initial lead states based on the grand-canonical ensemble. More specifically, the initial density matrix for lead $\zeta$ is

$$\rho_\zeta = \exp\left\{-\sum_{k_\zeta}(E_{k_\zeta} - \mu_\zeta)\hat c^\dagger_{k_\zeta}\hat c_{k_\zeta}/k_BT\right\}, \tag{35}$$

where $\mu_\zeta$ is the chemical potential for lead $\zeta$, given by

$$\mu_{L/R} = \pm V/2, \tag{36}$$

where $V$ is the source-drain bias voltage. Furthermore, we consider two different initial states of the molecular bridge: occupied and unoccupied. In both cases, the oscillator bath is in equilibrium with the state of the molecule bridge. When the molecular bridge is occupied, the coordinate operator $\hat{q}$ of the phonon modes is replaced with another displaced coordinate operator $\hat{Q}$ according to $\hat{q} = \hat{Q} - 2c_j/\omega_j^2$.

Next, we turn to the parameters in the charge transport Hamiltonian. The electronic energies $E_{k_\zeta}$ and molecule-lead coupling strengths $V_{dk_\zeta}$ are defined through energy-dependent level width functions,

$$\Gamma_\zeta(E) = 2\pi \sum_{k_\zeta} |V_{dk_\zeta}|^2 \delta(E - E_{k_\zeta}), \tag{37}$$

which is analogous to the spectral density function for the phonon bath defined in Eq. (17). In this work, we choose a tight-binding model for $\Gamma_\zeta$,

$$\Gamma(E) = \begin{cases} \dfrac{\alpha_e^2}{\beta_e^2}\sqrt{4\beta_e^2 - E^2}, & |E| < 2|\beta_e|, \\ 0, & |E| > 2|\beta_e|, \end{cases} \tag{38}$$

where $\beta_e$ and $\alpha_e$ are nearest-neighbor coupling between two lead sites and between the lead site and the bridge state, respectively. $\Gamma$ is then discretized according to Eq. (19) to produce the lead site energy $E_{k_\zeta}$ and the coupling constant $V_{dk_\zeta}$. In this work, we use a constant density of state, leading to an equidistant discretization of the interval $[-2\beta_e, 2\beta_e]$. We set $\alpha_e = 0.2$ eV, $\beta_e = 1$ eV, and a bias voltage of 0.1 V. For each lead, 160 electronic states are discretized. The phonon bath is modeled using the Cole–Davidson spectral density, and the prescription to obtain the discrete modes is the same as that described in Sec. III B. We set $\eta = 1000$ cm$^{-1}$, $\omega_c = 500$ cm$^{-1}$, and $\beta_{\text{CD}} = 0.5$. The number of bath modes $N_b$ is 1000. The overall tree structure for the model is based on the binary tree shown in Fig. 4(b). Two binary trees are constructed for lead states with $E_{k_\zeta} - \mu_\zeta < E_d$ and $E_{k_\zeta} - \mu_\zeta > E_d$, respectively.[40,78] The two trees are then grouped together for an overall binary tree. Another binary tree is constructed for the phonon bath. The two binary trees are then attached to the root, which represents the molecular bridge.

Similar to the study of the spin-boson model, the finite temperature time-dependent current is calculated through thermo-field dynamics.[40,67] The transformed Hamiltonian for the phonon bath and the electron–phonon coupling is the same as Eq. (24). Meanwhile, the electronic part is transformed as follows:

$$\begin{aligned} \hat{\hat{H}}_{\text{el}} = {} & E_d \hat{d}^\dagger \hat{d} + \sum_{\zeta=L,R} \sum_{k_\zeta} E_{k_\zeta} \left( \hat{c}_{k_\zeta}^\dagger \hat{c}_{k_\zeta} - \hat{\hat{c}}_{k_\zeta}^\dagger \hat{\hat{c}}_{k_\zeta} \right) \\ & + \sum_{\zeta=L,R} \sum_{k_\zeta} V_{dk_\zeta} \Big( \cos\theta_{k_\zeta} \hat{d}^\dagger \hat{c}_{k_\zeta} + \cos\theta_{k_\zeta} \hat{c}_{k_\zeta}^\dagger \hat{d} \\ & + \sin\theta_{k_\zeta} \hat{d}^\dagger \hat{\hat{c}}_{k_\zeta}^\dagger + \sin\theta_{k_\zeta} \hat{\hat{c}}_{k_\zeta} \hat{d} \Big), \end{aligned} \tag{39}$$

where $\theta_{k_\zeta} = \arctan \exp\left\{ -\frac{E_{k_\zeta} - \mu_\zeta}{2k_B T} \right\}$. Similarly, the transformation over the current operator follows

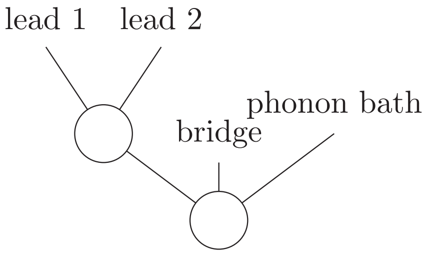

**FIG. 9.** Schematic diagram for the tree structure of the molecular junction model.

$$\begin{aligned} \hat{I}_\zeta = {} & i \sum_{k_\zeta} V_{dk_\zeta} \Big( \cos\theta_{k_\zeta} \hat{d}^\dagger \hat{c}_{k_\zeta} - \cos\theta_{k_\zeta} \hat{c}_{k_\zeta}^\dagger \hat{d} \\ & + \sin\theta_{k_\zeta} \hat{\hat{c}}_{k_\zeta} \hat{d} - \sin\theta_{k_\zeta} \hat{d}^\dagger \hat{\hat{c}}_{k_\zeta}^\dagger \Big), \quad \zeta = L, R. \end{aligned} \tag{40}$$

In this section, the time-dependent currents are calculated with 300 K, unless otherwise specified.

The overall formalism of the quantum transport problem appears to be much more complicated than the spin-boson model. However, thanks to the automatic construction of TTNO, both the programming effort and the computational cost for the simulation are of the same order as the simulation of spin-boson model. For example, the overall Python script for the simulation of the quantum transport problem contains ~300 lines of code, whereas the script for the simulation of the spin-boson model at zero temperature has around 200 lines. These scripts handle tasks such as the determination of model parameters through the discretization of the spectral density function, construction of the SOP Hamiltonian and current operators, specification of the TTN tree structure, construction of the TTNOs, time evolution, the calculation of physical observables, and various logging outputs. In these scripts, we only rely on library functions and classes that are general and applicable to any other physical model. In addition, the maximum $M_{\text{O}}$ across all the edges for the TTNO of the transformed Hamiltonian is 5. This ensures linear scaling with respect to both the number of electron modes and the number of phonon modes in the model.

We next provide more details on the automatically constructed TTNO for the molecular junction model. First of all, the overall tree structure is shown in Fig. 9. Two binary trees are constructed for lead states with $E_{k_\zeta} - \mu_\zeta < E_d$ and $E_{k_\zeta} - \mu_\zeta > E_d$, respectively. The two trees are then grouped together for an overall binary tree. Another binary tree is constructed for the phonon bath. The two binary trees are then attached to the root, which represents the molecular bridge.

The combined system of the bridge and the phonon bath is similar to a spin-boson model. As a result, the TTNO structure of the phonon bath tree is the same for both the molecular junction model and the spin-boson model. Then, the maximum bond dimension $M_{\text{O}}$ in this part of the tree is 3.

For the electronic lead, the maximum bond dimension $M_{\text{O}}$ is 5. Since each electronic mode has a physical bond dimension of only 2, two modes are attached directly to one node without primitive basis contraction. In Fig. 10, we show a schematic diagram of the tree tensor network corresponding to the electronic leads. Here, for ease of demonstration, only 8 lead electronic modes are shown.

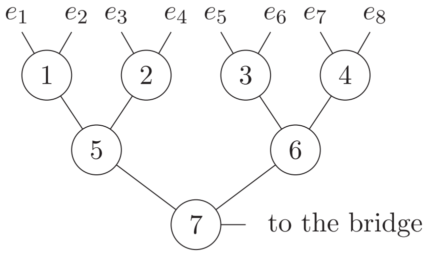

**FIG. 10.** Schematic diagram for the tree nodes of the lead part in the molecular junction model. $e_i$ represents the $i$th degree of freedom of the two leads. Each node is labeled with an index. The molecular bridge and the phonon bath are omitted.

The TTNO for the molecular junction is more complex than the TTNO for the spin-boson model. For nodes 1 and 4, the symbolic TTNO tensor has one index and the dimension is 4,

$$\hat{W}[i] = \begin{bmatrix} \hat{I}_{2i-1}\hat{I}_{2i} \\ E_{2i-1}\hat{\sigma}^{+}_{2i-1}\hat{\sigma}^{-}_{2i-1} + E_{2i}\hat{\sigma}^{+}_{2i}\hat{\sigma}^{-}_{2i} \\ V_{2i-1}\hat{\sigma}^{+}_{2i-1}\hat{\sigma}^{z}_{2i} + V_{2i}\hat{I}_{2i-1}\hat{\sigma}^{+}_{2i} \\ V_{2i-1}\hat{\sigma}^{-}_{2i-1}\hat{\sigma}^{z}_{2i} + V_{2i}\hat{I}_{2i-1}\hat{\sigma}^{-}_{2i} \end{bmatrix}, \quad i = 1, 4. \tag{41}$$

It should be noted that here spin operators are employed due to Jordan–Wigner transformation.

For nodes 2 and 3, the symbolic TTNO tensor has one index and the dimension is 5, which is the maximum bond dimension in the whole TTNO tree,

$$\hat{W}[i] = \begin{bmatrix} \hat{I}_{2i-1}\hat{I}_{2i} \\ \hat{\sigma}^{z}_{2i-1}\hat{\sigma}^{z}_{2i} \\ E_{2i-1}\hat{\sigma}^{+}_{2i-1}\hat{\sigma}^{-}_{2i-1} + E_{2i}\hat{\sigma}^{+}_{2i}\hat{\sigma}^{-}_{2i} \\ V_{2i-1}\hat{\sigma}^{+}_{2i-1}\hat{\sigma}^{z}_{2i} + V_{2i}\hat{I}_{2i-1}\hat{\sigma}^{+}_{2i} \\ V_{2i-1}\hat{\sigma}^{-}_{2i-1}\hat{\sigma}^{z}_{2i} + V_{2i}\hat{I}_{2i-1}\hat{\sigma}^{-}_{2i} \end{bmatrix}, \quad i = 2, 3. \tag{42}$$

The additional element in Eq. (42) compared to Eq. (41) is $\hat{\sigma}^{z}_{2i-1}\hat{\sigma}^{z}_{2i}$, which is the result of Jordan–Wigner transformation. The nodes at the "boundary" of the tree, such as nodes 1 and 4, do not have this term.

The nodes 5 and 6 are three-indexed tensors and the shapes are (4, 5, 4) and (5, 4, 4), respectively. Taking the fifth node as an example, the local TTNO tensor is

$$\hat{W}[5]_{jk} = \begin{bmatrix} \delta_{j1}\delta_{k1} \\ \delta_{j1}\delta_{k3} + \delta_{j2}\delta_{k1} \\ \delta_{j1}\delta_{k4} + \delta_{j3}\delta_{k2} \\ \delta_{j1}\delta_{k5} + \delta_{j4}\delta_{k2} \end{bmatrix}. \tag{43}$$

The elements represent the identity operator, lead site energy, creation operator from the lead, and annihilation operator from the lead, respectively.

The seventh node, or the root node of the lead part, has 3 indices and the shape is (4, 4, 4). The local TTNO tensor is

$$\hat{W}[7]_{jk} = \begin{bmatrix} \delta_{j1}\delta_{k1} \\ \delta_{j1}\delta_{k2} + \delta_{j2}\delta_{k1} \\ \delta_{j1}\delta_{k3} + \delta_{j3}\delta_{k1} \\ \delta_{j1}\delta_{k4} + \delta_{j4}\delta_{k1} \end{bmatrix}. \tag{44}$$

The nature of the elements is the same as Eq. (43), and the role of the seventh node is to merge the operators from two sub-trees.

Finally, the symbolic TTNO tensor for the bridge node is

$$\hat{W}[\text{bridge}] = \begin{bmatrix} E_d\hat{\sigma}^{+}_{d}\sigma^{-}_{d} & \hat{I}_d & 2\hat{\sigma}^{+}_{d}\sigma^{-}_{d} \\ \hat{I}_d & 0 & 0 \\ \hat{\sigma}^{-}_{d} & 0 & 0 \\ \hat{\sigma}^{+}_{d} & 0 & 0 \end{bmatrix}. \tag{45}$$

As in the spin-boson model, the finite temperature effect in the molecular junction model is taken into account via thermo-field

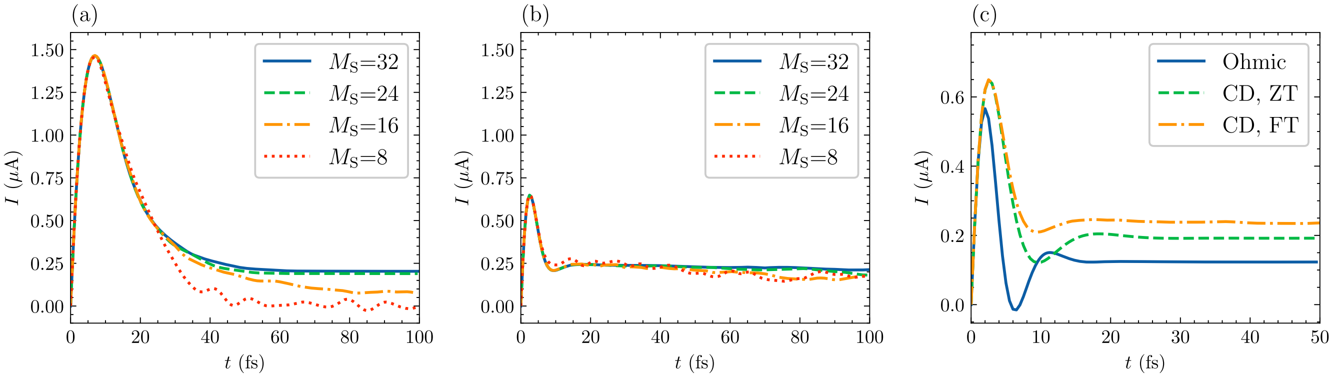

**FIG. 11.** Time-dependent current for the charge transport in a molecular junction. The Cole–Davidson spectral density function is employed, and the temperature is set to 300 K. In panels (a) and (b), the initial state of the molecular bridge is unoccupied and occupied, respectively. In panel (c), the finite temperature (FT) result is compared to zero temperature (ZT) result with both Cole–Davidson (CD) spectral density and Ohmic spectral density.

dynamicss.[67] The transformed Hamiltonian is structurally similar to the original Hamiltonian. Thus, the structure of the constructed TTNO is the same.

The simulated time-dependent current for the charge transport model is shown in Fig. 11. Figure 11(a) shows the convergence with respect to $M_S$ when the molecular bridge is initially unoccupied. The regularization technique from the Refs. 12 and 75 is used when $M_S = 24$ and $M_S = 32$ to achieve a converged current. Figure 11(b) shows the convergence with respect to $M_S$ when the molecular bridge is initially occupied. In both cases, $M_S = 32$ is sufficient to produce a converged outcome. Interestingly, an initially unoccupied bridge leads to a larger transient current, which is consistent with previous studies. Despite the different initial states, the steady current shown in Figs. 11(a) and 11(b) appears to be the same. We further compare the finite-temperature current with zero-temperature current and zero-temperature current with Ohmic spectral density, as shown in Fig. 11(c). For the Ohmic spectral density, we use the same characteristic frequency $\omega_c$ and reorganization energy $\lambda$ as the Cole–Davidson spectral density. From Fig. 11(c), it appears that using Cole–Davidson spectral density leads to a higher steady current. Furthermore, increasing temperature also results in a higher steady current.

## IV. CONCLUSION AND OUTLOOK

In this work, we present an efficient implementation of the TTNS algorithm, which is included as a module of the open-source Python package RENORMALIZER. Our implementation features the automatic construction of the optimal TTNO based on bipartite graph theory. For the models studied in this work, including the spin-boson model and the charge transport model, the TTNO of the Hamiltonian has constant $M_O$. Consequently, our algorithm scales linearly with the number of modes in the model, given fixed $M_S$. It is important to note that our algorithm for the construction of TTNO is completely general. It can be applied to any SOP operator and any tensor network with a tree topology. Therefore, the application of our program is not limited to the simulation of open quantum systems. It can also be applied to the simulation of other physical models, the propagation of stochastic Schrodinger equation,[79] and the simulation of quantum circuits.[80] While our program has broad applications, there is still plenty of room for further software improvement. For example, the efficiency of our program can be improved through massive parallelism on both CPU and GPU[81] or mixed-precision computation. In our previous work, we have shown that using a GPU can accelerate the time evolution of TD-DMRG by dozens of times.[39] Furthermore, just-in-time compilation techniques, which have been widely used to accelerate neural network training and the simulation of quantum circuits, could potentially be used to speed up the computational bottleneck of TTNS algorithms, such as the contraction of $\hat{H}_i^{\mathrm{eff}} A[i]$.

## ACKNOWLEDGMENTS

Weitang Li is supported by the Young Elite Scientists Sponsorship Program by CAST, Grant No. 2023QNRC001. Zhigang Shuai is supported by the National Natural Science Foundation of China (Grant No. T2350009), the Guangdong Provincial Natural Science Foundation (Grant No. 2024A1515011185), and the Shenzhen city "Pengcheng Peacock" Talent Program. Jiajun Ren is supported by the National Natural Science Foundation of China (Grant No. 22273005).

## AUTHOR DECLARATIONS

### Conflict of Interest

The authors have no conflicts to disclose.

### Author Contributions

**Weitang Li**: Conceptualization (lead); Data curation (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Resources (equal); Software (equal); Supervision (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Jiajun Ren**: Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Methodology (equal); Software (equal); Validation (equal); Writing – review & editing (equal). **Hengrui Yang**: Methodology (equal); Writing – review & editing (equal). **Haobin Wang**: Conceptualization (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Project administration (equal); Supervision (equal). **Zhigang Shuai**: Conceptualization (equal); Funding acquisition (equal); Investigation (equal); Project administration (equal); Resources (equal); Supervision (equal).

## DATA AVAILABILITY

The data that support the findings of this study are available from https://github.com/liwt31/li2024optimal.

## APPENDIX. BENCHMARK WITH ML-MCTDH

In this section, we benchmark our implementation by reproducing the existing results from ML-MCTDH, demonstrating the accuracy of our implementation. First, we consider the spin-boson model with the sub-Ohmic spectral density function as specified by Eq. (18). Here, $s = 0.5$ and $\omega_c = 20\Delta$ are employed as the parameters for the spectral density. We employ $N_b = 1000$, $M_S = 20$, and $d = 10$ as the parameters for the TTNS calculation. The other setups are the same as those shown in Fig. 5. We vary the coupling strength $\alpha$ from the weak coupling regime ($\alpha = 0.05$) to the strong coupling regime ($\alpha = 1.0$). In Fig. 12, we compare the dynamics calculated using our program to the results by ML-MCTDH.[65] The black solid lines indicate the results of ML-MCTDH, and the colored dashed lines indicate the results of our TTNS program. Across the weak to strong coupling regimes, the results show excellent agreement.

The second case involves the spin-boson model with the Cole–Davidson spectral density function as specified by Eq. (21). The computational setup is the same as those shown in Fig. 8, but with different model parameters. The details of the parameters are shown in the caption of Fig. 13. In Fig. 13, the black solid lines represent the results of ML-MCTDH, and the colored dashed lines represent the results of our TTNS program. Once again, the results

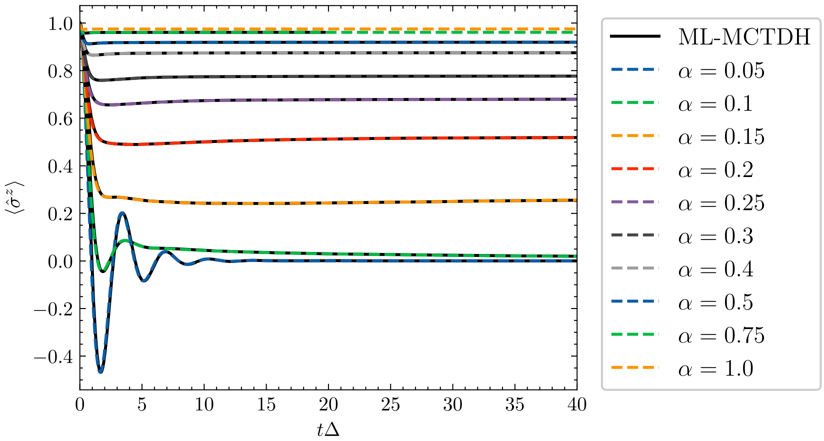

**FIG. 12.** Spin relaxation dynamics of the spin-boson model with sub-Ohmic spectral density from weak coupling $\alpha$ = 0.05 to strong coupling $\alpha$ = 1.0. The colored dashed lines indicate the results of our TTNS implementaion with $M_S$ = 20, $d$ = 10. The black solid lines indicate the results of ML-MCTDH with $\alpha$ from 0.05 to 0.75.

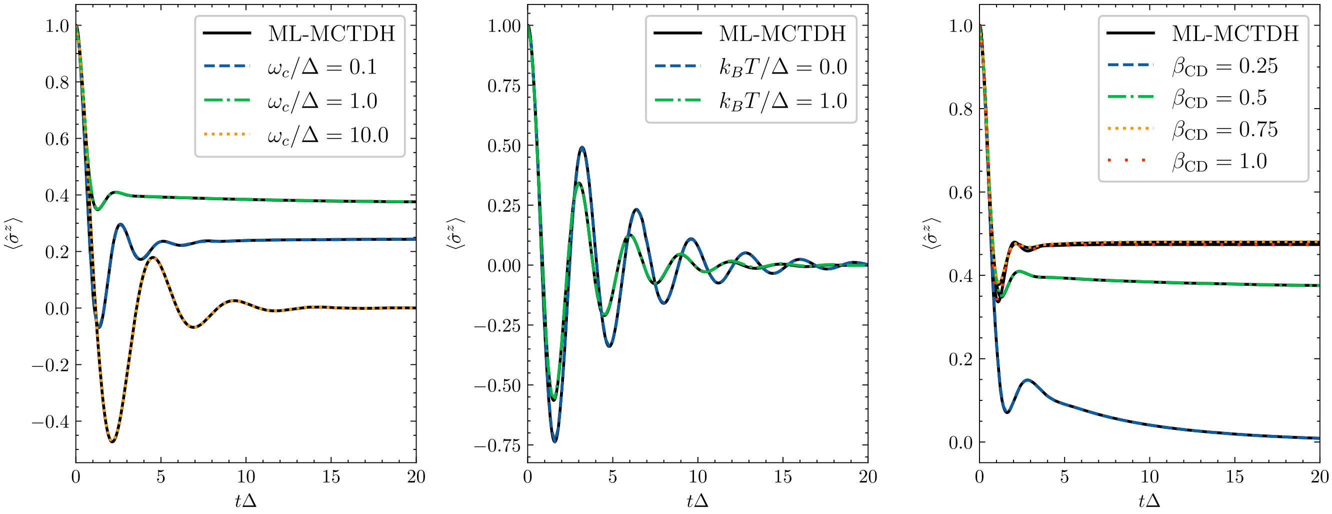

**FIG. 13.** Spin relaxation dynamics based on the Cole–Davidson spectral density. (a)$\eta/\Delta$ = 5, $\beta_{\mathrm{CD}}$ = 0.5, and $k_BT/\Delta$ = 0. (b) $\eta/\Delta$ = 1, $\beta_{\mathrm{CD}}$ = 0.25, and $\omega_c/\Delta$ = 1. (c) $\eta/\Delta$ = 5, $\omega_c/\Delta$ = 1, and $k_BT/\Delta$ = 0.

are in excellent agreement, further validating the accuracy of our algorithm.

## REFERENCES

[1] H.-D. Meyer, U. Manthe, and L. S. Cederbaum, "The multi-configurational time-dependent Hartree approach," Chem. Phys. Lett. **165**, 73–78 (1990).

[2] M. H. Beck, A. Jäckle, G. A. Worth, and H.-D. Meyer, "The multiconfiguration time-dependent Hartree (MCTDH) method: A highly efficient algorithm for propagating wavepackets," Phys. Rep. **324**, 1–105 (2000).

[3] H. Wang and M. Thoss, "Multilayer formulation of the multiconfiguration time-dependent Hartree theory," J. Chem. Phys. **119**, 1289–1299 (2003).

[4] Y.-Y. Shi, L.-M. Duan, and G. Vidal, "Classical simulation of quantum many-body systems with a tree tensor network," Phys. Rev. A **74**, 022320 (2006).

[5] N. Nakatani and G. K. Chan, "Efficient tree tensor network states (TTNS) for quantum chemistry: Generalizations of the density matrix renormalization group algorithm," J. Chem. Phys. **138**, 134113 (2013).

[6] H. R. Larsson, "A tensor network view of multilayer multiconfiguration time-dependent Hartree methods," Mol. Phys. **122**, e2306881 (2024).

[7] M. Thoss and H. Wang, "Quantum dynamical simulation of ultrafast molecular processes in the condensed phase," Chem. Phys. **322**, 210–222 (2006).

[8] H. Wang and M. Thoss, "Quantum dynamical simulation of electron-transfer reactions in an anharmonic environment," J. Phys. Chem. A **111**, 10369–10375 (2007).

[9] T. Westermann, R. Brodbeck, A. B. Rozhenko, W. Schoeller, and U. Manthe, "Photodissociation of methyl iodide embedded in a host-guest complex: A full dimensional (189D) quantum dynamics study of $CH_3I$@resorc[4]arene," J. Chem. Phys. **135**, 184102 (2011).

[10] K. A. Velizhanin, H. Wang, and M. Thoss, "Heat transport through model molecular junctions: A multilayer multiconfiguration time-dependent Hartree approach," Chem. Phys. Lett. **460**, 325–330 (2008).

[11] H. Wang, I. Pshenichnyuk, R. Härtle, and M. Thoss, "Numerically exact, time-dependent treatment of vibrationally coupled electron transport in single-molecule junctions," J. Chem. Phys. **135**, 244506 (2011).

[12] H. Wang and M. Thoss, "Numerically exact, time-dependent study of correlated electron transport in model molecular junctions," J. Chem. Phys. **138**, 134704 (2013).

[13] H. Wang and M. Thoss, "From coherent motion to localization: Dynamics of the spin-boson model at zero temperature," New J. Phys. **10**, 115005 (2008).

[14] H. Wang and J. Shao, "Dynamics of a two-level system coupled to a bath of spins," J. Chem. Phys. **137**, 22A504 (2012).

[15] J. Schulze, M. F. Shibl, M. J. Al-Marri, and O. Kühn, "Multi-layer multi-configuration time-dependent Hartree (ML-MCTDH) approach to the correlated exciton-vibrational dynamics in the FMO complex," J. Chem. Phys. **144**, 185101 (2016).

[16] K. Gunst, F. Verstraete, S. Wouters, O. Legeza, and D. Van Neck, "T3NS: Three-legged tree tensor network states," J. Chem. Theory Comput. **14**, 2026–2033 (2018).

[17] Z. Li, "Expressibility of comb tensor network states (CTNS) for the P-cluster and the FeMo-cofactor of nitrogenase," Electron. Struct. **3**, 014001 (2021).

[18] H. R. Larsson, "Computing vibrational eigenstates with tree tensor network states (TTNS)," J. Chem. Phys. **151**, 204102 (2019).

[19] Y. Yan, M. Xu, T. Li, and Q. Shi, "Efficient propagation of the hierarchical equations of motion using the Tucker and hierarchical Tucker tensors," J. Chem. Phys. **154**, 194104 (2021).

[20] Y. Ke, "Tree tensor network state approach for solving hierarchical equations of motion," J. Chem. Phys. **158**, 211102 (2023).

[21] C. Guo, A. Weichselbaum, J. von Delft, and M. Vojta, "Critical and strong-coupling phases in one- and two-bath spin-boson models," Phys. Rev. Lett. **108**, 160401 (2012).

[22] F. A. Schröder, D. H. Turban, A. J. Musser, N. D. Hine, and A. W. Chin, "Tensor network simulation of multi-environmental open quantum dynamics via machine learning and entanglement renormalisation," Nat. Commun. **10**, 1062 (2019).

[23] U. Schollwöck, "The density-matrix renormalization group in the age of matrix product states," Ann. Phys. **326**, 96–192 (2011).

[24] J. Ren, W. Li, T. Jiang, Y. Wang, and Z. Shuai, "Time-dependent density matrix renormalization group method for quantum dynamics in complex systems," WIREs Comput. Mol. Sci. **12**, e1614 (2022).

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

[26] S. R. White, "Density-matrix algorithms for quantum renormalization groups," Phys. Rev. B **48**, 10345 (1993).

[27] Z. Shuai, J. Brédas, A. Saxena, and A. Bishop, "Linear and nonlinear optical response of polyenes: A density matrix renormalization group study," J. Chem. Phys. **109**, 2549–2555 (1998).

[28] 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).

[29] Y. Yao, X. Xie, and H. Ma, "Ultrafast long-range charge separation in organic photovoltaics: Promotion by off-diagonal vibronic couplings and entropy increase," J. Phys. Chem. Lett. **7**, 4830–4835 (2016).

[30] J. R. Mannouch, W. Barford, and S. Al-Assam, "Ultra-fast relaxation, decoherence, and localization of photoexcited states in $\pi$-conjugated polymers," J. Chem. Phys. **148**, 034901 (2018).

[31] X. Xie, Y. Liu, Y. Yao, U. Schollwöck, C. Liu, and H. Ma, "Time-dependent density matrix renormalization group quantum dynamics for realistic chemical systems," J. Chem. Phys. **151**, 224101 (2019).

[32] J. Ren, Z. Shuai, and G. Kin-Lic Chan, "Time-dependent density matrix renormalization group algorithms for nearly exact absorption and fluorescence spectra of molecular aggregates at both zero and finite temperature," J. Chem. Theory Comput. **14**, 5027–5039 (2018).

[33] A. Baiardi and M. Reiher, "Large-scale quantum dynamics with matrix product states," J. Chem. Theory Comput. **15**, 3481–3498 (2019).

[34] Y. Wang, J. Ren, and Z. Shuai, "Minimizing non-radiative decay in molecular aggregates through control of excitonic coupling," Nat. Commun. **14**, 5056 (2023).

[35] W. Li, J. Ren, and Z. Shuai, "A general charge transport picture for organic semiconductors with nonlocal electron–phonon couplings," Nat. Commun. **12**, 4260 (2021).

[36] Y. Ge, W. Li, J. Ren, and Z. Shuai, "Computational method for evaluating the thermoelectric power factor for organic materials modeled by the Holstein model: A time-dependent density matrix renormalization group formalism," J. Chem. Theory Comput. **18**, 6437–6446 (2022).

[37] A. Chin, J. Prior, R. Rosenbach, F. Caycedo-Soler, S. Huelga, and M. B. Plenio, "The role of non-equilibrium vibrational structures in electronic coherence and recoherence in pigment–protein complexes," Nat. Phys. **9**, 113 (2013).

[38] R. Borrelli and M. F. Gelin, "Simulation of quantum dynamics of excitonic systems at finite temperature: An efficient method based on thermo field dynamics," Sci. Rep. **7**, 9127 (2017).

[39] W. Li, J. Ren, and Z. Shuai, "Numerical assessment for accuracy and GPU acceleration of TD-DMRG time evolution schemes," J. Chem. Phys. **152**, 024127 (2020).

[40] H. Yang, W. Li, J. Ren, and Z. Shuai, "Time-dependent density matrix renormalization group method for quantum transport with phonon coupling in molecular junction," J. Chem. Theory Comput. **19**, 6070–6081 (2023).

[41] G. K.-L. Chan, A. Keselman, N. Nakatani, Z. Li, and S. R. White, "Matrix product operators, matrix product states, and *ab initio* density matrix renormalization group algorithms," J. Chem. Phys. **145**, 014102 (2016).

[42] S. Keller, M. Dolfi, M. Troyer, and M. Reiher, "An efficient matrix product operator representation of the quantum chemical Hamiltonian," J. Chem. Phys. **143**, 244118 (2015).

[43] E. M. Stoudenmire and S. R. White, "Sliced basis density matrix renormalization group for electronic structure," Phys. Rev. Lett. **119**, 046401 (2017).

[44] C. Hubig, I. McCulloch, and U. Schollwöck, "Generic construction of efficient matrix product operators," Phys. Rev. B **95**, 035129 (2017).

[45] J. Ren, W. Li, T. Jiang, and Z. Shuai, "A general automatic method for optimal construction of matrix product operators using bipartite graph theory," J. Chem. Phys. **153**, 084118 (2020).

[46] T. Jiang, J. Ren, and Z. Shuai, "Unified definition of exciton coherence length for exciton–phonon coupled molecular aggregates," J. Phys. Chem. Lett. **14**, 4541–4547 (2023).

[47] S. Szalay, M. Pfeffer, V. Murg, G. Barcza, F. Verstraete, R. Schneider, and Ö. Legeza, "Tensor product methods and entanglement optimization for *ab initio* quantum chemistry," Int. J. Quantum Chem. **115**, 1342–1391 (2015).

[48] F. Otto, "Multi-layer Potfit: An accurate potential representation for efficient high-dimensional quantum dynamics," J. Chem. Phys. **140**, 014106 (2014).

[49] D. Sulz, C. Lubich, G. Ceruti, I. Lesanovsky, and F. Carollo, "Numerical simulation of long-range open quantum many-body dynamics with tree tensor networks," Phys. Rev. A **109**, 022420 (2024).

[50] R. M. Milbradt, Q. Huang, and C. B. Mendl, "State diagrams to determine tree tensor network operators," SciPost Phys. Core **7**, 036 (2024).

[51] U. Manthe, "A multilayer multiconfigurational time-dependent Hartree approach for quantum dynamics on general potential energy surfaces," J. Chem. Phys. **128**, 164116 (2008).

[52] V. Murg, F. Verstraete, Ö. Legeza, and R. M. Noack, "Simulating strongly correlated quantum systems with tree tensor networks," Phys. Rev. B **82**, 205105 (2010).

[53] A. J. Leggett, S. Chakravarty, A. T. Dorsey, M. P. Fisher, A. Garg, and W. Zwerger, "Dynamics of the dissipative two-state system," Rev. Mod. Phys. **59**, 1 (1987).

[54] G. A. Worth, M. H. Beck, A. Jäckle, O. Vendrell, and H.-D. Meyer, The MCTDH package, Version 8.2, (2000). H.-D. Meyer, Version 8.3 (2002), Version 8.4 (2007). O. Vendrell and H.-D. Meyer Version 8.5 (2013). Versions 8.5 and 8.6 contains the

ML-MCTDH algorithm. Current versions: 8.4.23, 8.5.16, and 8.6.2 (May 2022), See http://mctdh.uni-hd.de/

[55] R. Ellerbrock, K. G. Johnson, S. Seritan, H. Hoppe, J. Zhang, T. Lenzen, T. Weike, U. Manthe, and T. J. Martínez, "QuTree: A tree tensor network package," J. Chem. Phys. **160**, 112501 (2024).

[56] J. J. García-Ripoll, "Time evolution of matrix product states," New J. Phys. **8**, 305 (2006).

[57] J. Broeckhove, L. Lathouwers, E. Kesteloot, and P. Van Leuven, "On the equivalence of time-dependent variational principles," Chem. Phys. Lett. **149**, 547–550 (1988).

[58] J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, and F. Verstraete, "Unifying time evolution and optimization with matrix product states," Phys. Rev. B **94**, 165116 (2016).

[59] D. Bauernfeind and M. Aichhorn, "Time dependent variational principle for tree tensor networks," SciPost Phys. **8**, 024 (2020).

[60] L. P. Lindoy, B. Kloss, and D. R. Reichman, "Time evolution of ML-MCTDH wavefunctions. I. Gauge conditions, basis functions, and singularities," J. Chem. Phys. **155**, 174108 (2021).

[61] L. P. Lindoy, B. Kloss, and D. R. Reichman, "Time evolution of ML-MCTDH wavefunctions. II. Application of the projector splitting integrator," J. Chem. Phys. **155**, 174109 (2021).

[62] H. Zhai, H. R. Larsson, S. Lee, Z.-H. Cui, T. Zhu, C. Sun, L. Peng, R. Peng, K. Liao, J. Tölle *et al.*, "Block2: A comprehensive open source framework to develop and apply state-of-the-art DMRG algorithms in electronic structure and beyond," J. Chem. Phys. **159**, 234801 (2023).

[63] T. Xiang, "Density-matrix renormalization-group method in momentum space," Phys. Rev. B **53**, R10445 (1996).

[64] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," SIAM J. Comput. **2**, 225–231 (1973).

[65] H. Wang and M. Thoss, "From coherent motion to localization: II. Dynamics of the spin-boson model with sub-Ohmic spectral density at zero temperature," Chem. Phys. **370**, 78–86 (2010).

[66] Y. Zhou, J. Shao, and H. Wang, "Dynamics of electron transfer in complex glassy environment modeled by the Cole–Davidson spectral density," Mol. Phys. **110**, 581–594 (2012).

[67] Y. Takahashi and H. Umezawa, "Thermo field dynamics," Int. J. Mod. Phys. B **10**, 1755–1805 (1996).

[68] R. Borrelli and M. F. Gelin, "Quantum electron-vibrational dynamics at finite temperature: Thermo field dynamics approach," J. Chem. Phys. **145**, 224101 (2016).

[69] R. Borrelli and M. F. Gelin, "Finite temperature quantum dynamics of complex systems: Integrating thermo-field theories and tensor-train methods," WIREs Comput. Mol. Sci. **11**, e1539 (2021).

[70] E. W. Fischer and P. Saalfrank, "A thermofield-based multilayer multiconfigurational time-dependent Hartree approach to non-adiabatic quantum dynamics at finite temperature," J. Chem. Phys. **155**, 134109 (2021).

[71] M. A. Reed, C. Zhou, C. Muller, T. Burgin, and J. Tour, "Conductance of a molecular junction," Science **278**, 252–254 (1997).

[72] F. Chen, J. Hihath, Z. Huang, X. Li, and N. Tao, "Measurement of single-molecule conductance," Annu. Rev. Phys. Chem. **58**, 535–564 (2007).

[73] M. Galperin, M. A. Ratner, A. Nitzan, and A. Troisi, "Nuclear coupling and polarization in molecular transport junctions: Beyond tunneling to function," Science **319**, 1056–1060 (2008).

[74] D. Xiang, X. Wang, C. Jia, T. Lee, and X. Guo, "Molecular-scale electronics: From concept to function," Chem. Rev. **116**, 4318–4440 (2016).

[75] H. Wang and M. Thoss, "Multilayer multiconfiguration time-dependent Hartree study of vibrationally coupled electron transport using the scattering-state representation," J. Phys. Chem. A **117**, 7431–7441 (2013).

[76] H. Wang and M. Thoss, "Employing an interaction picture to remove artificial correlations in multilayer multiconfiguration time-dependent Hartree simulations," J. Chem. Phys. **145**, 164105 (2016).

[77] P. Jordan and E. Wigner, "Über das paulische äquivalenzverbot," Z. Phys. **47**, 631–651 (1928).

[78] M. M. Rams and M. Zwolak, "Breaking the entanglement barrier: Tensor network simulation of quantum transport," Phys. Rev. Lett. **124**, 137701 (2020).

[79] X. Gao, J. Ren, A. Eisfeld, and Z. Shuai, "Non-Markovian stochastic Schrödinger equation: Matrix-product-state approach to the hierarchy of pure states," Phys. Rev. A **105**, L030202 (2022).

[80] P. Seitz, I. Medina, E. Cruz, Q. Huang, and C. B. Mendl, "Simulating quantum circuits using tree tensor networks," Quantum **7**, 964 (2023).

[81] 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).

Published under an exclusive license by AIP Publishing