

RESEARCH ARTICLE | SEPTEMBER 08 2025

Tree tensor network hierarchical equations of motion based on time-dependent variational principle for efficient open quantum dynamics in structured thermal environments

Xinxian Chen  ; Ignacio Franco  



J. Chem. Phys. 163, 104109 (2025)

<https://doi.org/10.1063/5.0278591>

 CHORUS



Articles You May Be Interested In

Bexcitonics: Quasiparticle approach to open quantum dynamics

J. Chem. Phys. (May 2024)

A low-temperature quantum Fokker–Planck equation that improves the numerical stability of the hierarchical equations of motion for the Brownian oscillator spectral density

J. Chem. Phys. (February 2022)

Modeling of excitation dynamics in large-size molecular systems: Hierarchical equations with compartmentalization

J. Chem. Phys. (October 2024)



The Journal of Chemical Physics

Special Topics Open for Submissions

[Learn More](#)

Tree tensor network hierarchical equations of motion based on time-dependent variational principle for efficient open quantum dynamics in structured thermal environments

Cite as: J. Chem. Phys. 163, 104109 (2025); doi: 10.1063/5.0278591

Submitted: 30 April 2025 • Accepted: 17 July 2025 •

Published Online: 8 September 2025



Xinxian Chen¹  and Ignacio Franco^{1,2,3,a)} 

AFFILIATIONS

¹ Department of Chemistry, University of Rochester, Rochester, New York 14627, USA

² Department of Physics and Astronomy, University of Rochester, Rochester, New York 14627, USA

³ The Institute of Optics, University of Rochester, Rochester, New York 14627, USA

^{a)} Author to whom correspondence should be addressed: ignacio.franco@rochester.edu

ABSTRACT

We introduce an efficient method, TTN-HEOM, for exactly calculating the open quantum dynamics for driven quantum systems interacting with highly structured bosonic baths by combining the tree tensor network (TTN) decomposition scheme with the hexcitonic generalization of the numerically exact hierarchical equations of motion (HEOM). The method yields a series of quantum master equations for all core tensors in the TTN that efficiently and accurately capture the open quantum dynamics for non-Markovian environments to all orders in the system-bath interaction. These master equations are constructed based on the time-dependent Dirac-Frenkel variational principle, which isolates the optimal dynamics for the core tensors given the TTN ansatz. The dynamics converges to the HEOM when increasing the rank of the core tensors, a limit in which the TTN ansatz becomes exact. We introduce TENSQ, tensor equations for non-Markovian structured open systems, as a general-purpose Python code to propagate the TTN-HEOM dynamics. We implement three general propagators for the coupled master equations: two fixed-rank methods that require a constant memory footprint during the dynamics and one adaptive-rank method with a variable memory footprint controlled by the target level of computational error. We exemplify the utility of these methods by simulating a two-level system coupled to a structured bath containing one Drude-Lorentz component and eight Brownian oscillators, which is beyond what can presently be computed using the standard HEOM. Our results show that the TTN-HEOM is capable of simulating both dephasing and relaxation dynamics of driven quantum systems interacting with structured baths, even those of chemical complexity, with an affordable computational cost.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0278591>

I. INTRODUCTION

Computation of the dynamics of open quantum systems with high precision is a central challenge in physics, chemistry, and quantum information science.^{1–5} It is necessary to capture the decoherence and eventual thermalization of quantum systems due to interactions with a quantum thermal environment. From a molecular science perspective, such open quantum dynamics is central in our elementary description of photophysics, photochemistry, multidimensional optical spectroscopies,^{6,7} coherent control,^{8–11} and charge and energy transfer.^{12–14} From a quantum information science perspective, correctly capturing such open quantum dynamics

is needed to understand the evolution and decay of coherence and entanglement in qubits, simulate the operation of digital and analog quantum processors, design quantum control strategies, and develop strategies to minimize decoherence effects in next-generation quantum devices.^{15–23}

In open quantum dynamics,^{1,15,24} it is customary to divide the Hamiltonian of the quantum universe, $H = H_S + H_B + H_{SB}$, into a system, H_S , which corresponds to the degrees of freedom (DoFs) of interest, an environment or bath, H_B , and their interaction, H_{SB} . The state of the system is completely described by its reduced density matrix $\rho_S(t) = \text{Tr}_B[\rho(t)]$ obtained by tracing out the bath DoFs from the density matrix of the composite quantum system $\rho(t)$. The

dynamics of the system $\rho_S(t)$ can be obtained by either following the dynamics of both the system and bath and then tracing over the bath or by solving so-called quantum master equations satisfied by $\rho_S(t)$ that implicitly capture the influence of the environment on the system DoFs. The former approach is preferred^{18,25–29} but is often computationally impractical, as the bath can be macroscopic. For this reason, there has been impressive progress in formulating increasingly accurate quantum master equations and developing computational techniques to propagate them.^{4,30–36}

Of particular interest are numerically “exact” master equations, such as the hierarchical equations of motion (HEOM),^{35,37–39} as they can be used to model a large class of problems of interest in chemistry and quantum information science with an accuracy that can be assumed. This contrasts with common strategies based on the Born–Markov approximation, such as the Lindblad⁴⁰ and Redfield master equations,^{30,41} that are only valid for quantum systems weakly coupled to an environment that has a short memory time with respect to the system’s dynamics—conditions that are often violated by chemically and physically relevant systems. The HEOM approach is complemented by other numerically exact methods to capture the open quantum dynamics that do not have a master equation form, such as the quasi-adiabatic propagator path integral (QuAPI).^{42–46}

The HEOM is based on decomposing the dynamics of the bath correlation function (BCF) into a series of K complex exponential functions or *features*. The influence of the thermal environment on the system is captured by introducing an infinite hierarchy of auxiliary density matrices (ADMs) that evolve as the system decoheres and reaches thermal equilibrium. The dimensionality $M \times M$ of each ADM is the same as the reduced density matrix of the M -level system. This HEOM dynamics was recently shown to be identical to the system in interaction with K fictitious bosonic quasiparticles called *bexcitons* that are born, oscillate, and decay during the dynamics in such a way that the system has the correct dynamics.³⁹

This hierarchy can be truncated to a given order N for each feature, which is referred to as the depth of the HEOM. The number of required ADMs is N^K , resulting in the overall space complexity of HEOM $\mathcal{O}(M^2 N^K)$. Therefore, the computational cost of the HEOM increases exponentially with the number of features, K . This number of features grows with the complexity of the chemical environment or when low-temperature corrections are needed in the dynamics. For this reason, to date, using the HEOM, we are able to investigate illustrative model problems with simple environment models with only a few features in the BCF decomposition, such as a single Drude–Lorentz or Brownian oscillator model.^{35,37,47} However, the HEOM becomes intractable for realistically highly structured chemical environments.

To reduce this curse of dimensionality, one strategy is to employ the filtering technique that removes ADMs that are almost zero,⁴⁸ thus enabling HEOM simulations with more complex environments with larger K . While helpful, the strategy is still insufficient to model chemically realistic problems; furthermore, it is not applicable to environments at low temperatures.⁴⁹ A second strategy is to capture the time dynamics of the BCF more efficiently, thus reducing the number of required features.^{50–57} More general strategies to curb this curse of dimensionality are needed to apply this numerically exact HEOM method to chemically complex systems and environments.

In this paper, we introduce a tree tensor network (TTN) decomposition of the HEOM, TTN-HEOM, that enables efficient simulation of open quantum dynamics in structured thermal environments, even those of chemical complexity. Our approach is based on the recent *bexcitonic* generalization of the HEOM, which recovers all HEOM variants and is, therefore, of general applicability to the HEOM family of quantum master equations. The method further admits arbitrary time-dependence in the system Hamiltonian as needed to investigate driven dynamics of qubits and molecular systems in the presence of a thermal environment. In addition, we develop a general-purpose Python-based computational implementation of the TTN-HEOM that we name *TENSO*. For computational efficiency, our implementation takes advantage of NumPy⁵⁸ and PyTorch,⁵⁹ which contain a series of libraries specifically designed and optimized to deal with tensor manipulation on CPUs and GPUs.

Tensor network decompositions are the basis of highly successful simulation strategies^{60,61} in many-body science. In unitary quantum dynamics, tensor trains (or matrix product states) have been successfully used to enable wavefunction propagation in high-dimensions.^{25,62–67} In turn, TTN decompositions of the multi-DoF wavefunction propagated by the time-dependent Schrödinger equation are the basis of widely employed methods such as the multi-layer multi-configurational time-dependent Hartree method (ML-MCTDH)^{60,68} and related strategies.^{69,70} For open quantum systems, tensor network techniques have also been proposed to accelerate simulations. This includes efforts in approximate methods such as Lindblad master equations^{71,72} and strategies where thermal effects are included just at the initial time through purification.^{28,73} They have also been used in numerically exact approaches, such as the thermalized time-evolving density operator with orthogonal polynomials algorithm,⁷⁴ path-integral process tensor methods,^{75–77} and also the HEOM.^{38,78–84} Overall, the current view that has emerged from these efforts is that tensor network strategies can be successful in curbing the curse of dimensionality in quantum dynamics by efficiently encoding the entanglement among DoFs.

In this paper, we advance a rigorous and practical TTN decomposition of the HEOM by taking inspiration from ideas and methods in the MCTDH literature. Specifically, we arrange the collection of all ADMs in HEOM into an extended density operator (EDO) containing both the physical DoFs and the DoFs arising from the decomposition of the BCF into features. We then use a TTN decomposition to express this high-order EDO tensor as a contraction of low-order *core* tensors. We provide a rigorous derivation of the quantum master equation satisfied by each core tensor in the TTN by invoking the Dirac–Frenkel time-dependent variational principle (TDVP), which is also used in the MCTDH.^{29,85} These coupled master equations guarantee that the TTN decomposition of the EDO remains as accurate as possible during the dynamics. To propagate the dynamics, we use strategies used in MCTDH and adapt them and generalize them to the EDO and non-unitary dynamics of the HEOM. Specifically, we implement and test the projector-splitting^{86–89} and direct-integration with regularization^{26,90,91} and demonstrate that they provide stable propagation of TTN-HEOM.

We exemplify our method and computational implementation in a two-level molecule coupled to a highly structured thermal environment with a spectral density extracted from experiments,⁹² a

system that is challenging to model using standard HEOM due to the large memory requirements. To this end, we decompose the spectral density in terms of Drude–Lorentz features to represent the solvent and underdamped Brownian oscillators to represent intramolecular vibrations.^{92,93} This approach is more efficient to invoke in HEOM compared to the discretization approach, where finitely many discrete vibrational modes are used for modeling the bath, as they do not capture the dissipative nature in open quantum dynamics.³⁷

Compared to recent advances in combining tensor network techniques into HEOM, our proposed method and computational implementation admit both tensor trees and tensor trains and, therefore, build upon but generalize initial efforts using tensor train strategies.^{78–80,82,83}

Compared to recent efforts to introduce TTN into the HEOM,^{38,84} we have been able to develop and implement three numerically stable propagation strategies based on projector-splitting (PS) and direct integration. PS is a form of Trotterization where a single-step in the overall propagation is split into stepwise propagation of each core tensor in the TTN. We implemented two versions of PS: a constant rank PS1 method where memory requirements are fixed and an adaptive rank PS2 method controlled by the target level of computational error.

In addition, we introduce a direct integration method that arises from our derivation of the quantum master equation of the core tensors through TDVP. The method simultaneously integrates the master equations for all core tensors. As such, it admits any numerical integration scheme, such as high-order Runge–Kutta schemes⁹⁴ and, therefore, can be parallelized and has better scaling with the integration time step Δt with respect to PS. However, the scheme requires regularization,^{26,90,91} which can add a small error to the dynamics.

We implemented the TTN-HEOM with these numerical propagation strategies into a Python package called Tensor Equations for Non-Markovian Structured Open systems (TENS0). We discuss the merits and limitations of these numerical propagation strategies in TTN-HEOM using TENS0. Overall, our developments provide a TTN-HEOM method and computational implementation of full functionality that enables investigations into the dissipative dynamics of quantum systems immersed in highly structured thermal bosonic environments.

The paper is organized as follows. We first summarize the bexcitonic generalization of the HEOM (Sec. II A). Then, we introduce its TTN decomposition (Sec. II B) and isolate the equations of motion satisfied by the core tensors (Sec. II C). Next, we introduce the three propagation methods (Sec. II D) and discuss the computational implementation of the TTN-HEOM (Sec. II E). In Sec. III, we exemplify the utility of the method by simulating dissipative quantum dynamics due to interactions of quantum systems with highly structured thermal baths using the three propagation methods and different TTN topologies. We summarize our main findings in Sec. IV.

II. THEORY

A. Hierarchical equations of motion and bexcitonic picture

HEOM is capable of following the dissipative dynamics of general driven quantum systems coupled to multiple independent

thermal baths through system operators that do not need to commute.³⁵ For clarity in presentation, and without loss of generality, we consider coupling to one thermal harmonic bath with Hamiltonian

$$H_B = \sum_j \left(\frac{p_j^2}{2m_j} + \frac{m_j \omega_j^2 x_j^2}{2} \right), \quad (1)$$

where x_j and p_j are the position and momentum operators of the j th harmonic mode of effective mass m_j and frequency ω_j . The system–bath coupling $H_{SB} = Q_S \otimes X_B$ is linear to a system operator Q_S and a collective bath coordinate

$$X_B = \sum_j c_j x_j, \quad (2)$$

where c_j quantifies the coupling strength between the j th bath mode and the system operator.

While the dynamics of the density matrix of the composite system $\rho(t)$ is unitary, the dynamics of the system's density matrix $\rho_S(t) = \text{Tr}_B(\rho(t))$ is non-unitary and satisfies³⁴

$$\tilde{\rho}_S(t) = \mathcal{T} \tilde{\mathcal{F}}(t, 0) \rho_S(0), \quad (3)$$

where $\mathcal{T}$ is the time-ordering operator

$$\tilde{\mathcal{F}}(t, 0) = e^{-\int_0^t ds \tilde{Q}_S^\times(s) \int_0^s du (C(s-u) \tilde{Q}_S(u))^\times}, \quad (4)$$

and $C(t) = \text{Tr}(\tilde{X}_B(t) \tilde{X}_B(0) \rho_B^{\text{eq}})$ is the BCF. Throughout we use atomic units where $\hbar = 1$ and the notation $A^>B = AB$ and $A^<B = BA^\dagger$ for the ordering of matrix multiplications, and $A^\times = A^> - A^<$ for the commutator super-operator generated from A .⁹⁵ In writing Eq. (3), we have adopted the interaction picture of $H_0(t) = H_S(t) + H_B$, where $\tilde{O}(t) = \left(\mathcal{T} e^{-i \int_0^t H_0(t') dt'} \right)^\dagger O(t) \mathcal{T} e^{-i \int_0^t H_0(t') dt'}$. Equation (4) provides a formal solution to the open quantum dynamics at all temperatures and to all orders in the system–bath interaction. As seen, $C(t)$ contains all the information needed to capture the influence of the bath on $\rho_S(t)$.

The BCF is related to the bath spectral density $J(\omega) = \sum_j |c_j|^2 \delta(\omega - \omega_j) / (2m_j \omega_j)$ through^{12,96}

$$C(t) = \int_0^\infty J(\omega) (\coth(\omega/(2k_B T)) \cos(\omega t) - i \sin(\omega t)) d\omega. \quad (5)$$

The integral can be resolved by using the residue theorem through the analytical continuation of $J(\omega)$ and Matsubara⁹⁷ or Padé⁹⁸ expansion of the thermal $\coth(\omega/(2k_B T))$ component. That analysis shows that $C(t)$ and its complex conjugate $C^*(t)$ can always be decomposed in terms of a series of complex exponential functions as

$$C(t) = \sum_{k=1}^K c_k e^{\gamma_k t} \quad \text{and} \quad C^*(t) = \sum_{k=1}^K \bar{c}_k e^{\gamma_k^* t}, \quad (6)$$

where c_k , $\bar{c}_k$, and γ_k are complex numbers. Other numerical methods can also be used to fit the BCF into the form in Eq. (6).^{50–57} Each k in the series Eq. (6) defines a *feature* of the bath. This decomposition of $C(t)$ into K features can capture any physical dynamics, including exponential decay, oscillations, and their combination.³⁹

The HEOM results from introducing this decomposition of the BCF into the exact dynamical map in Eq. (3) and calculat-

ing the time-derivatives. What this shows is that the influence of the thermal environment on the dynamics of the system is exactly captured through a collection of auxiliary $M \times M$ density matrices (ADMs) $\{\rho_{\vec{n}}(t)\}$ with the same dimensionality of $\rho_S(t)$. Here, $\vec{n}$ is a K -dimensional index $\vec{n} = (n_1, \dots, n_k, \dots, n_K)$ with $n_k = 0, 1, 2, \dots$, and the series runs *ad infinitum*. We arrange these ADMs as a vector of matrices that we call the *extended density operator (EDO)*

$$|\Omega(t)\rangle = \sum_{\vec{n}} \rho_{\vec{n}}(t) |\vec{n}\rangle, \quad (7)$$

in a basis $\{|\vec{n}\rangle \equiv |n_1\rangle \otimes \dots \otimes |n_k\rangle \otimes \dots \otimes |n_K\rangle\}$ such that $\rho_{\vec{n}}(t) = \langle \vec{n} | \Omega(t) \rangle$. The physical system's density matrix $\rho_S(t) = \rho_{\vec{0}}(t)$ is located at $\vec{n} = \vec{0} \equiv (0, \dots, 0)$.

In this context, we find that the exact quantum dynamics for this EDO $|\Omega(t)\rangle$ is³⁹

$$\frac{d}{dt} |\Omega(t)\rangle = \left(-iH_S^\times(t) + \sum_{k=1}^K \mathcal{D}_k \right) |\Omega(t)\rangle, \quad (8)$$

with

$$\mathcal{D}_k = \gamma_k \hat{\alpha}_k^\dagger \hat{\alpha}_k + (c_k Q_S^\geq - \bar{c}_k Q_S^\leq) \hat{z}_k^{-1} \hat{\alpha}_k^\dagger - Q_S^\times \hat{\alpha}_k \hat{z}_k, \quad (9)$$

and initial conditions

$$|\Omega(0)\rangle = \rho_S(0) |\vec{0}\rangle, \quad (10)$$

where $\rho_S(0)$ is the initial state of the system. The first term in Eq. (8) is the unitary dynamics, while $\mathcal{D}_k$ captures the dissipation due to the k th feature of the bath. Here, the bosonic creation $\hat{\alpha}_k^\dagger$ and annihilation $\hat{\alpha}_k$ operators ($[\hat{\alpha}_k, \hat{\alpha}_{k'}^\dagger] = \delta_{kk'}$) associated with the k th bath feature connect the different ADMs as

$$\hat{\alpha}_k^\dagger |n_k\rangle = \sqrt{n_k + 1} |n_k + 1\rangle, \quad \hat{\alpha}_k |n_k\rangle = \sqrt{n_k} |n_k - 1\rangle. \quad (11)$$

In turn, the $\hat{z}_k$ is any invertible operator that satisfies $[\hat{z}_k, \hat{\alpha}_k^\dagger \hat{\alpha}_k] = 0$, which we refer to as the metric. Equation (8) defines a class of exact quantum master equations, as there is a choice in the representation of $|\vec{n}\rangle$ (position $|\vec{x}\rangle$, momentum $|\vec{p}\rangle$, or number $|\vec{n}\rangle$) and the metric $\hat{z}_k$. The standard HEOM^{37,48,99} is a specific case of Eq. (8) obtained when the number representation and $\hat{z}_k = i(\hat{\alpha}_k^\dagger \hat{\alpha}_k)^{-1/2}$ are chosen.

As discussed in Ref. 39, based on Eq. (8), the open quantum dynamics can be interpreted as the system interacting with a collection of fictitious bosonic quasiparticles that we call *bexcitons*. For this, we associate $|\vec{n}\rangle$ with the creation of bexcitons with respect to vacuum $|\vec{0}\rangle$. Specifically, we associate a bexciton of label k , a k -bexciton, for each feature of the bath k . The state $|\vec{n}\rangle$ corresponds to a situation in which n_k k -bexcitons have been created for each k . In this picture, $\hat{\alpha}_k^\dagger$ creates and $\hat{\alpha}_k$ destroys a k -bexciton. The commutation relation between $\hat{\alpha}_k$ and $\hat{\alpha}_k^\dagger$ dictates that bexcitons are bosons. While the bath can be macroscopic, only K effective bexcitons are needed to capture the relevant component that influences the system. Therefore, the bexcitons offer a coarse-grained, but still exact, view of the correlated non-Markovian system-bath dynamics to all orders in H_{SB} . The dissipators $\{\mathcal{D}_k\}$ in Eq. (8) describe the bexcitonic dynamics and their interaction with the system. As the composite system evolves toward a stationary state, bexcitons are created and destroyed. Each version of Eq. (8) constitutes an

exact map of the open quantum dynamics to the system-bexciton dynamics. While the system's dynamics is common to all maps, the bexcitonic one is not. For this reason, the bexcitons are unphysical quasiparticles, and bexcitonic properties should be viewed as a way to monitor the open quantum dynamics and its numerical convergence.

B. Tree tensor network decomposition

The computational challenge of the HEOM is that the number of bexcitons K needed to accurately describe the dynamics increases as the complexity of the spectral density grows and with decreasing temperature as needed to appropriately decompose $C(t)$ [see Eq. (5)]. Furthermore, the ladder of states for each n_k needs to be truncated at a given $(N_k - 1)$ that defines the *depth* of the k -bexciton, a quantity that needs to be increased until convergence. The overall space complexity of Eq. (8) for an M -state system and K bath features all truncated at a depth of $N_k = \mathcal{O}(N)$ is $\mathcal{O}(M^2 N^K)$ and, therefore, shows exponential growth with the number of bath features K . This is the reason why the HEOM computations have been limited to relatively simple models of the bath.⁴⁷ Our hypothesis is that the HEOM has a lot of redundancy in state-space that can be efficiently compressed through a tensor network strategy and used to curb this curse of dimensionality.

In the same way that the density matrix of the system $\rho_S(t)$ has matrix elements $[\rho_S]_{ij} = \langle i | \rho_S(t) | j \rangle$, where $\{|i\rangle\}$ is a basis that spans the Hilbert space of the system, the EDO has tensor elements

$$[\Omega(t)]_{ijn_1 \dots n_K} = \langle i | \langle n_1 \dots n_K | \Omega(t) \rangle | j \rangle, \quad (12)$$

where $\{|n_k\rangle\}_{n_k=0}^{N_k-1}$ is the number basis that spans the space of the k -bexciton truncated at the level of $(N_k - 1)$. The bexcitonic dynamics Eq. (8) for this extended tensor can be written as $\frac{d}{dt} \Omega(t) = \mathcal{L}(t) \Omega(t)$, where $\mathcal{L}(t)$ is the tensor representation of the superoperator that generates the dynamics $(-iH_S^\times(t) + \sum_{k=1}^K \mathcal{D}_k)$ in the given basis. Because the basis is a tensor product of individual elements $|i\rangle \otimes |j\rangle \otimes |n_1\rangle \otimes \dots \otimes |n_K\rangle$, then from Eq. (8) we can define local operators h_m^κ ($\kappa = >, <, 1, \dots, K$) such that

$$\sum_{m=1}^{5K+2} h_m^>(t) \otimes h_m^<(t) \otimes h_m^{(1)} \otimes \dots \otimes h_m^{(K)} \equiv \mathcal{L}(t). \quad (13)$$

The label m runs over the individual terms in Eq. (8). Each $\mathcal{D}_k$ in Eq. (8) gives five terms, and the system Liouvillian $-iH_S^\times(t)$ gives two more. Each term consists of a component $h_m^>(t)$ that acts on basis $\{|i\rangle\}$, a component $h_m^<(t)$ that acts on $\{|j\rangle\}$, and components $h_m^{(k)}$ that act on $\{|n_k\rangle\}$.

The extended density tensor is high dimensional and can be compressed through a TTN, which contains a collection of many low-order *core tensors* with a given contraction-ordering that can be topologically described by a tree graph (see Fig. 1). The TTN may contain core tensors with different tensor orders. We want to decompose the high-order tensor into a series of low-order tensors, as the operations between high-order tensors are computationally expensive. For instance, the space complexity of a D -order tensor $A_{a_1 \dots a_D}$ with R as the range of all indices a_d is $\mathcal{O}(R^D)$. Since the space-complexity of a tensor grows as a power of its order D , we naturally want the order of each core tensor in a TTN to be as small as possible. However, one cannot use only order-2 tensors in

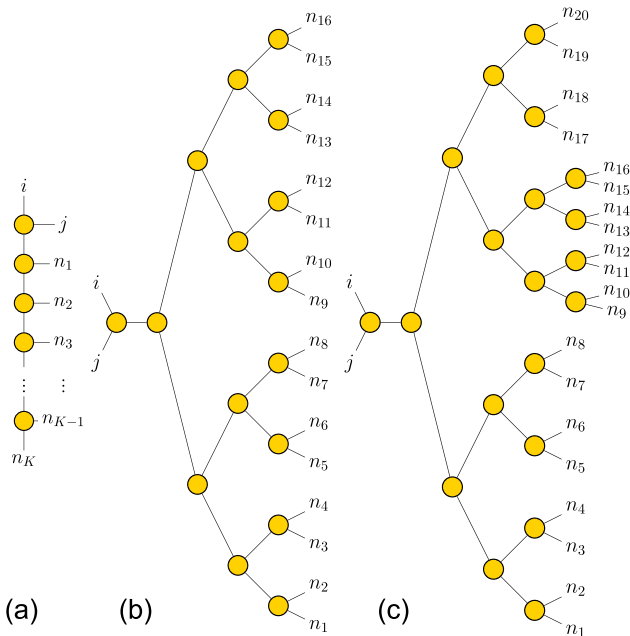


FIG. 1. Topological structure of (a) a tensor train with K bexcitons for the extended density operator (EDO), (b) a perfectly balanced tensor tree with 16 bexcitons, and (c) a balanced tensor tree with 20 bexcitons. TTN-HEOM admits all these topologies with core tensors of variable order; the figure focuses on order-3.

TTN for such decomposition in the presence of a thermal environment, as it cannot give a tree, including a train for $K > 0$. Therefore, the minimal non-trivial order for the core tensors is 3. For this reason, below we focus on the TTN-HEOM, where all core tensors are of order-3. The generalization of the TTN to arbitrary order for each of the core tensors is included in the [supplementary material](#).

No matter what the topology of the TTN is, the number of order-3 core tensors in the decomposition will be K , and the number of indices for contractions in the TTN will be $K - 1$. The simplest example is a tensor train [Fig. 1(a)], which can be formulated as

$$\Omega_{ijn_1 \dots n_K} = \sum_{a_1 a_2 \dots a_{K-1}}^{R_1 R_2 \dots R_{K-1}} A_{ija_1}^{(0)} U_{a_1 n_1 a_2}^{(1)} U_{a_2 n_2 a_3}^{(2)} \dots U_{a_{K-1} n_{K-1} n_K}^{(K-1)}. \quad (14)$$

Here, $\{a_s\}_{s=1}^{K-1}$ are the introduced indices for contractions with $a_s = 0, \dots, R_s - 1$, with R_s being the *rank* of index a_s . In turn, Figs. 1(b) and 1(c) show tensor trees for a 16-bexciton and 20-bexciton EDO, respectively. In Fig. 1, each node represents a core tensor, while each bond represents an index in $\{i, j\} \cup \{a_s\}_{s=1}^{K-1} \cup \{n_k\}_{k=1}^K$. The bonds $\{\alpha, \beta, \gamma\}$ attached to a node indicate that the tensor $U^{(s)}$ represented by that node will have indices α, β, γ as $U_{\alpha\beta\gamma}^{(s)}$.

Notice that the arrangement of indices α, β, γ in tensor $U^{(s)}$ is not reflected in Fig. 1. As a convention (and without loss of generality), for $U^{(s)}$ we always place the index a_s in the first index position α .

Generally, the core tensors are isolated through hierarchical Tucker decomposition,^{100,101} which repeatedly applies singular value

decompositions (SVDs). The final result of such a decomposition on a high-order EDO gives a TTN that can be formally represented as

$$\begin{aligned} [\Omega(t)]_{ijn_1 \dots n_K} &= \sum_{a_1 \dots a_{K-1}}^{R_1 \dots R_{K-1}} A_{ija_1}^{(0)} U_{a_1 \beta_1 \gamma_1}^{(1)} \dots U_{a_{K-1} \beta_{K-1} \gamma_{K-1}}^{(K-1)} \\ &\equiv [\text{Con}(A^{(0)}(t), U^{(1)}(t), \dots, U^{(K-1)}(t))]_{ijn_1 \dots n_K}, \end{aligned} \quad (15)$$

where Con represents contractions among all core time-dependent tensors $A^{(0)}(t), U^{(1)}(t), \dots, U^{(K-1)}(t)$ in the TTN. For each $U^{(s)}(t)$, its second and third indices $\beta_s, \gamma_s \in \{a_s\}_{s=1}^{K-1} \cup \{n_k\}_{k=1}^K$ ($s = 1, \dots, K - 1$). From the SVD, the $U^{(s)}(t)$ are semi-unitary core tensors in the sense that the matrix $[U^{(s)}(t)]_{a_s \beta \gamma}$ reshaped from the tensor $[U^{(s)}(t)]_{a_s \beta \gamma}$ satisfies

$$\sum_{\beta \gamma} [U^{(s)}(t)]_{a'_s \beta \gamma}^* [U^{(s)}(t)]_{a_s \beta \gamma} = \delta_{a'_s a_s}. \quad (16)$$

By contrast, there is only one *root* core tensor $A^{(0)}$ in the TTN, which does not need to be semi-unitary. As a design principle, we choose to include the system's indices i, j , and the index a_1 in the root tensor, such that the influence of all bexcitons is captured through compressed index a_1 . In this way, the unitary component of the system is exact and not compressed, while the influence of the bexcitons is compactly captured by the TTN.

For a given set of ranks $\{R_s\}$ such that $R_s = \mathcal{O}(R)$ and a bexciton depth N such that $N_k = \mathcal{O}(N)$, the space complexity of the TTN is $\mathcal{O}(M^2 R + KNR(N + R))$,¹⁰¹ which no longer grows exponentially with the number of bath features K . The smaller the rank that can be used, the more efficient the compression of the TTN.

It is useful to define the *height* L of a core tensor $U^{(s)}$ as the number of bonds on the path between $U^{(s)}$ and the root. For instance, the root $A^{(0)}$ is of height 0, and in our TTN ansatz Eq. (15), the $U^{(1)}$ is always of height 1. We sort the label s of the core tensor $U^{(s)}$ according to its height $L(U^{(s)})$ without loss of generality. That is, for two core tensors $U^{(r)}$ and $U^{(s)}$, if $L(U^{(r)}) < L(U^{(s)})$, then $r < s$.

C. Master equations for a tree tensor network

To develop the master equations for the TTN, we invoke the Dirac-Frenkel TDVP^{29,85} and adapt it to $\Omega(t)$ as

$$\sum_{ijn_1 \dots n_K} [\delta\Omega(t)]_{ijn_1 \dots n_K}^* \left[\left(\mathcal{L}(t) - \frac{d}{dt} \right) \Omega(t) \right]_{ijn_1 \dots n_K} = 0, \quad (17)$$

where $\delta\Omega(t)$ denotes a small variation of $\Omega(t)$. By doing so, it yields optimal dynamics for the core tensors that capture the dynamics of $\Omega(t)$ in a space with reduced dimensionality that changes dynamically during the quantum evolution. To guarantee that the TTN decomposition remains, we further require that Eq. (16) holds during the propagation by demanding

$$\sum_{\beta \gamma} [U^{(s)}(t)]_{a'_s \beta \gamma}^* \left[\frac{d}{dt} U^{(s)}(t) \right]_{a_s \beta \gamma} = 0, \quad (18)$$

for all t , a condition that is referred to as the gauge condition.⁶⁵ From Eqs. (17) and (18), we can systematically develop equations

of motion for the semi-unitary and root core tensors for an arbitrary tensor tree. Appendix A derives the equations of motion for order-3 tensors, and the supplementary material offers their generalization using graph notation to a general TTN containing tensors with arbitrary order.

With the TTN decomposition in Eq. (15) and the gauge condition in Eq. (18), the master equation of the root tensor $A^{(0)}(t)$ depends on $f_m^{(1)}(t)$ (defined later) as

$$\frac{d}{dt} A_{i'j'a_1}^{(0)} = \sum_m \sum_{ij a_1} [h_m^{>}]_{i'i} [h_m^{<}]_{j'j} [f_m^{(1)}]_{a_1 a_1} A_{ij a_1}^{(0)}. \quad (19)$$

This is the simplest of the equations because the semi-unitary properties of the $U^{(s)}(t)$ and the gauge condition cancel out the terms involving direct contraction between the $U^{(s)}(t)$ and its time-derivatives. In turn, the master equations of the semi-unitary tensors $U^{(s)}(t)$ are

$$\begin{aligned} \sum_{a'_s} D_{a'_s a'_s}^{(s)} \frac{d}{dt} U_{a'_s a'_s}^{(s)} &= \sum_m \sum_{a'_s a_s \beta \gamma} [D_m^{(s)}]_{a'_s a'_s} \\ &\times \left([F_m^{(s2)}]_{\beta' \beta} [F_m^{(s3)}]_{\gamma' \gamma} U_{a'_s \beta \gamma}^{(s)} - U_{a_s \beta' \gamma'}^{(s)} [f_m^{(s)}]_{a_s a'_s} \right). \end{aligned} \quad (20)$$

Note that in our notation, $F_m^{(s2)}(t)$ always contracts with the second index of $U^{(s)}(t)$, while $F_m^{(s3)}(t)$ contracts with the third one. The definition of matrix $F_m^{(sk)}(t)$ for $\kappa = 2$ and 3 depends on the location of the tensor $U^{(s)}(t)$ in the TTN. $F_m^{(sk)}(t) \equiv f_m^{(u)}(t)$ ($u > s$) if the κ th index corresponds to a contracted index or bond in the TTN. That is, when the κ th index in $U^{(s)}(t)$ is an a_u in Eq. (15). In turn, $F_m^{(sk)}(t) \equiv h_m^{(k)}$ [cf. Eq. (13)] when the κ th index also occurs in the original EDO tensor $\Omega(t)$ and, therefore, corresponds to an open bond in the TTN. That is, when the κ th index in $U^{(s)}(t)$ is an n_k in Eq. (15).

The matrices $f_m^{(s)}(t)$ ($s = 1, \dots, K-1$) are defined as

$$[f_m^{(s)}]_{a'_s a_s} \equiv \sum_{\beta' \beta \gamma \gamma'} U_{a'_s \beta' \gamma'}^{(s)*} [F_m^{(s2)}]_{\beta' \beta} [F_m^{(s3)}]_{\gamma' \gamma} U_{a_s \beta \gamma}^{(s)}. \quad (21)$$

Notice that this definition is recursive, as $f_m^{(s)}(t)$ depends on $F_m^{(sk)}(t) = f_m^{(u)}(t)$ ($u > s$) if the κ th index corresponds to a contracted bond. The matrices $D^{(s)}(t)$ and matrices $D_m^{(s)}(t)$ for given label m in Eq. (13) are also defined recursively. For $s = 1$,

$$\begin{aligned} D_{a'_1 a_1}^{(1)} &\equiv \sum_{ij} A_{ij a_1}^{(0)} A_{ij a_1}^{(0)*}, \\ [D_m^{(1)}]_{a'_1 a_1} &\equiv \sum_{i'j'j} [h_m^{>}]_{i'i'} [h_m^{<}]_{j'j} A_{i'j'j}^{(0)} A_{ij a_1}^{(0)*}. \end{aligned} \quad (22)$$

For $s > 1$, there is a bond in the TTN corresponding to a_s in Eq. (15) that contracts tensors $U^{(s)}(t)$ and $U^{(r)}(t)$ ($r < s$). If $U^{(r)}(t)$ has a_s as its third index in Eq. (15), then

$$\begin{aligned} D_{a'_s a_s}^{(s)} &\equiv \sum_{a'_r a_r \varepsilon} U_{a'_r \varepsilon a'_s}^{(r)} D_{a'_r a_r}^{(r)} U_{a_r \varepsilon a_s}^{(r)*}, \\ [D_m^{(s)}]_{a'_s a_s} &\equiv \sum_{a'_r a_r \varepsilon' \varepsilon} [F_m^{(r2)}]_{\varepsilon \varepsilon'} U_{a'_r \varepsilon' a'_s}^{(r)} [D_m^{(r)}]_{a'_r a_r} U_{a_r \varepsilon a_s}^{(r)*}. \end{aligned} \quad (23)$$

In turn, if $U^{(r)}$ has a_s as the second index in Eq. (15), then

$$\begin{aligned} D_{a'_s a_s}^{(s)} &\equiv \sum_{a'_r a_r \varepsilon} U_{a'_r \varepsilon a'_s}^{(r)} D_{a'_r a_r}^{(r)} U_{a_r \varepsilon a_s}^{(r)*}, \\ [D_m^{(s)}]_{a'_s a_s} &\equiv \sum_{a'_r a_r \varepsilon' \varepsilon} [F_m^{(r3)}]_{\varepsilon \varepsilon'} U_{a'_r \varepsilon' a'_s}^{(r)} [D_m^{(r)}]_{a'_r a_r} U_{a_r \varepsilon a_s}^{(r)*}. \end{aligned} \quad (24)$$

In Sec. II E, we discuss the order in which these terms $[f_m^{(s)}(t)]$, $D^{(s)}(t)$ and $D_m^{(s)}(t)$ need to be evaluated. As an explicit example, Appendix B details the TTN-HEOM scheme with $K = 4$.

From Eq. (10), the initial condition for the EDO is $\Omega_{ijn_1 \dots n_K}(0) = [\rho_S(0)]_{ij} \delta_{0n_1} \dots \delta_{0n_K}$, where $\rho_S(0)$ is the initial state of the system. In addition, we need to determine the initial conditions for the core tensors, $A^{(0)}(t=0)$ and $U^{(s)}(t=0)$, in the TTN. Except for the minimal rank case when all $R_s = 1$, this choice is not unique. We choose

$$A_{ij a_1}^{(0)}(0) = [\rho_S]_{ij} \delta_{0a_1}, \quad (25)$$

for the root tensor. In turn, $U_{a_s \beta \gamma}^{(s)}(0)$ for given $a_s = 0, 1, \dots, R_s - 1$ is filled as $U_{0\beta\gamma}^{(s)} = \delta_{0\beta} \delta_{0\gamma}$, $U_{1\beta\gamma}^{(s)} = \delta_{1\beta} \delta_{0\gamma}$, $U_{2\beta\gamma}^{(s)} = \delta_{0\beta} \delta_{1\gamma}$, $U_{3\beta\gamma}^{(s)} = \delta_{2\beta} \delta_{0\gamma}$, $U_{4\beta\gamma}^{(s)} = \delta_{1\beta} \delta_{1\gamma}$, $U_{5\beta\gamma}^{(s)} = \delta_{0\beta} \delta_{2\gamma}$, $\dots$. More explicitly, in each page of tensor $U_{a_s \beta \gamma}^{(s)}(0)$, the matrix $U_{a_s}^{(s)}(0)$ for $a_s = 0, 1, \dots, R_s - 1$ is chosen as

$$\begin{aligned} U_0^{(s)} &= \begin{pmatrix} 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, & U_1^{(s)} &= \begin{pmatrix} 0 & 0 & 0 & \dots \\ 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \\ U_2^{(s)} &= \begin{pmatrix} 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, & U_3^{(s)} &= \begin{pmatrix} 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ 1 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \\ U_4^{(s)} &= \begin{pmatrix} 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, & U_5^{(s)} &= \begin{pmatrix} 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \end{aligned}$$

and so on. The choice of $U_{0\beta\gamma}^{(s)}$ satisfies the correct initial condition for the EDO. In addition, the choice of $U_{a_s \beta \gamma}^{(s)}$ further guarantees the semi-unitary property of $U^{(s)}(0)$ at the initial time. This choice is sufficient and is one of the simplest possible, as all non-zero elements are 1 and there is only one non-zero element in each $U_{a_s}^{(s)}(0)$. Furthermore, it locally balances all bonds associated with a node across the TTN.

D. Propagation methods

1. Direct integration

The main idea of direct integration is to simultaneously integrate the non-linear coupled series of ordinary differential equations (ODEs) in Eqs. (19) and (20) using standard integration techniques. The advantage of this strategy is that it enables coupling the TTN-HEOM to well-developed ODE solvers based on Runge–Kutta⁹⁴ and other schemes that allow for large integration time steps, adaptive time steps, and even parallelization.

To isolate the exact derivatives for the semi-unitary tensors, we need to multiply both sides of Eq. (20) by $(D^{(s)}(t))^{-1}$. The challenge of this direct integration strategy is that this inverse does not always exist. In particular, for initially separable states, as required by the HEOM, $D^{(s)}(t)$ is singular. To see this, consider $[D^{(1)}(0)]_{a_1 a'_1} = \sum_{ij} A_{ija_1}(0) A_{ija'_1}^*(0) = (\text{Tr} \rho_S^2(0)) \delta_{0a_1} \delta_{0a'_1}$ from Eq. (25). This leads to a singular matrix $D^{(1)}(0)$. The remaining $D^{(s)}(0)$ are also singular according to the recursive relation [Eq. (23)] and the semi-unitary properties of $U^{(s)}$ [Eq. (16)].

To make progress, we introduce the pseudo-inverse¹⁰² $(D^{(s)})^+$ of a matrix $D^{(s)}$. The pseudo-inverse is a well-known generalization of the inverse of a matrix. It can be constructed from the SVD of $D^{(s)} = U\sigma V^\dagger$ where U, V are unitary matrices that are readily invertible as $U^\dagger = U^{-1}$. The pseudo-inverse $(D^{(s)})^+ = V\sigma^+ U^\dagger$, where σ^+ is the pseudo-inverse of the diagonal matrix σ obtained by replacing the nonzero singular values σ_b with their multiplicative inverses σ_b^{-1} . Since the SVD always exists, then $(D^{(s)})^+$ can also always be defined.

Multiplying both sides of Eq. (20) by $(D^{(s)})^+$ yields

$$\begin{aligned} \sum_{a'_s} \mathcal{P}_{a'_s a''_s}^{(s)} \frac{d}{dt} U_{a'_s \beta' \gamma'}^{(s)} \\ = \sum_m \sum_{a'_s a''_s \beta \gamma} [C_m^{(s)}]_{a'_s a''_s} \\ \times \left([F_m^{(s2)}]_{\beta' \beta} [F_m^{(s3)}]_{\gamma' \gamma} U_{a'_s \beta \gamma}^{(s)} - U_{a'_s \beta' \gamma'}^{(s)} [f_m^{(s)}]_{a_s a'_s} \right). \end{aligned} \quad (26)$$

Here,

$$\mathcal{P}_{a'_s a''_s}^{(s)} \equiv \sum_{a_s} D_{a'_s a_s}^{(s)} (D^{(s)})_{a_s a''_s}^+, \quad (27)$$

and

$$[C_m^{(s)}]_{a'_s a''_s} \equiv \sum_{a_s} [D_m^{(s)}]_{a'_s a_s} (D^{(s)})_{a_s a''_s}^+. \quad (28)$$

If matrix $D^{(s)}$ is invertible, then $(D^{(s)})^+ = (D^{(s)})^{-1}$ and $\mathcal{P}_{a'_s a''_s}^{(s)} = \delta_{a'_s a''_s}$ becomes an identity matrix. In turn, when $D^{(s)}$ is singular, then $\mathcal{P}^{(s)}$ is a projector to the column space of the matrix $D^{(s)}$ as

$$\mathcal{P}^{(s)}(t) = D^{(s)}(t) (D^{(s)}(t))^+ = \sum_{\substack{b \text{ s.t.} \\ \sigma_b(t) > 0}} u_b(t) u_b^\dagger(t), \quad (29)$$

where $u_b(t)$ is the b th column of $U(t)$ and $\sigma_b(t)$ is the b th singular value of matrix $D^{(s)}(t)$. Hence, a singular $D^{(s)}(t)$ will introduce a loss of information if we let $\frac{d}{dt} U^{(s)}(t) \approx \mathcal{P}^{(s)}(t) \frac{d}{dt} U^{(s)}(t)$.

To handle this singularity issue, we invoke the regularization technique developed in MCTDH and adapt it to this TTN-HEOM.^{26,90,91} In one simple strategy, all singular values σ_b in $D^{(s)}(t)$ that are smaller than a threshold ϵ are replaced by ϵ . That is, if the SVD of $D_{a'_s a_s}^{(s)} = \sum_b U_{a'_s b} \sigma_b V_{a_s b}^*$, the regularization approximate $D_{a'_s a_s}^{(s)} \approx \sum_b U_{a'_s b} \max(\sigma_b, \epsilon) V_{a_s b}^*$. This makes $D^{(s)}(t)$ invertible but introduces an error of $\mathcal{O}(\epsilon)$ in $D^{(s)}(t)$.

Our choice of regularization described below reduces the introduced error in $D^{(s)}(t)$ from $\mathcal{O}(\epsilon)$ to $\mathcal{O}(\epsilon^2)$, increasing the stability and overall accuracy of this propagation scheme. To do so, what is needed is to regularize both $D^{(s)}(t)$ and every $D_m^{(s)}(t)$. Starting from $s = 1$, we perform the SVD for $A_{ija_1}^{(0)} = \sum_{b_1} W_{ijb_1}^{(1)} \sigma_{b_1}^{(1)} V_{a_1 b_1}^{(1)*}$ and define

$$[\tilde{D}_m^{(1)}]_{a_1 b_1} \equiv \sum_{i' j' j} [h_m^{(1)}]_{i' i} [h_m^{(1)}]_{j' j} A_{i' j' a_1}^{(0)} W_{ijb_1}^{(1)*}. \quad (30)$$

Here, $\tilde{D}_m^{(1)}(t)$, $\sigma^{(1)}(t)$, and $V^{(1)}(t)$ are all time-dependent quantities. Furthermore, together with the core tensor $U^{(1)}(t)$, we define a non-semi-unitary tensor $A^{(1)}(t)$ such that

$$A_{b_1 \beta \gamma}^{(1)} \equiv \sum_{a_1} \sigma_{b_1}^{(1)} V_{a_1 b_1}^{(1)*} U_{a_1 \beta \gamma}^{(1)}. \quad (31)$$

For $s > 1$, the construction of $\tilde{D}_m^{(s)}$, as well as $W^{(s)}$, $\sigma^{(s)}$, $V^{(s)}$, and $A^{(s)}$, is performed by a recursive process over the TTN structure that is similar to the definition for $D^{(s)}$ and $D_m^{(s)}$ [cf. Eqs. (22)–(24)].

For $s > 1$, there is a bond in the TTN corresponding to a_s in Eq. (15) that contracts tensors $U^{(s)}(t)$ and $U^{(r)}(t)$ ($r < s$). In this recursive argument, $\tilde{D}_m^{(r)}(t)$ and $A^{(r)}(t)$ have already been determined from the previous step in the recursion. If $U^{(r)}(t)$ has a_s as its second index in Eq. (15), then the SVD of $A^{(r)}$ is

$$A_{b_r a_s \epsilon}^{(r)} = \sum_{b_s} W_{b_r b_s \epsilon}^{(s)} \sigma_{b_s}^{(s)} V_{a_s b_s}^{(s)*}. \quad (32)$$

Then, $\tilde{D}_m^{(s)}(t)$ is defined as

$$[\tilde{D}_m^{(s)}]_{a_s b_s} \equiv \sum_{a_r b_r \epsilon} [F_m^{(r3)}]_{\epsilon \epsilon'} U_{a_r a_s \epsilon'}^{(r)} [\tilde{D}_m^{(r)}]_{a_r b_r} W_{b_r b_s \epsilon}^{(s)*}. \quad (33)$$

In this case,

$$D_{a_s a'_s}^{(s)} = \sum_{b_r \epsilon} A_{b_r a_s \epsilon}^{(r)} A_{b_r a'_s \epsilon}^{(r)*}. \quad (34)$$

In turn, if $U^{(r)}(t)$ has a_s as its third index in Eq. (15), the SVD of $A^{(r)}$ is

$$A_{b_r \epsilon a_s}^{(r)} = \sum_{b_s} W_{b_r b_s \epsilon}^{(s)} \sigma_{b_s}^{(s)} V_{a_s b_s}^{(s)*}, \quad (35)$$

and

$$[\tilde{D}_m^{(s)}]_{a_s b_s} \equiv \sum_{a_r b_r \epsilon} [F_m^{(r2)}]_{\epsilon \epsilon'} U_{a_r \epsilon' a_s}^{(r)} [\tilde{D}_m^{(r)}]_{a_r b_r} W_{b_r b_s \epsilon}^{(s)*}. \quad (36)$$

In this case,

$$D_{a_s a'_s}^{(s)} = \sum_{b_s, \epsilon} A_{b_s \epsilon a_s}^{(r)} A_{b_s \epsilon a'_s}^{(r)*} \quad (37)$$

For both cases, the definition of $A^{(s)}(t)$ continues as

$$A_{b_s \beta \gamma}^{(s)} \equiv \sum_{a_s} \sigma_{b_s}^{(s)} V_{a_s b_s}^{(s)*} U_{a_s \beta \gamma}^{(s)}. \quad (38)$$

Substitute Eq. (32) into Eq. (34) and Eq. (35) into Eq. (37), and $D^{(s)}$ becomes

$$D_{a_s a'_s}^{(s)} = \sum_{b_s} V_{a_s b_s}^{(s)*} \left(\sigma_{b_s}^{(s)} \right)^2 V_{a'_s b_s}^{(s)}. \quad (39)$$

Furthermore, comparing Eq. (23) with Eq. (33) and Eq. (24) with Eq. (36), we have

$$[D_m^{(s)}]_{a_s a'_s} = \sum_{b_s} [\bar{D}_m^{(s)}]_{a_s b_s} \sigma_{b_s}^{(s)} V_{a'_s b_s}^{(s)}. \quad (40)$$

From Eqs. (39) and (40), Eq. (28) becomes

$$[C_m^{(s)}]_{a_s a'_s} = \sum_{b_s} [\bar{D}_m^{(s)}]_{a_s b_s} \left(\sigma_{b_s}^{(s)} \right)^{-1} V_{a'_s b_s}^{(s)}. \quad (41)$$

The regularization is to replace $\sigma_{b_s}^{(s)}$ that are less than ϵ by ϵ . Hence, the equation of motion Eq. (26) becomes

$$\begin{aligned} \frac{d}{dt} U_{a''_s \beta' \gamma'}^{(s)} &\approx \sum_m \sum_{b_s, a'_s, \beta \gamma} [\bar{D}_m^{(s)}]_{a'_s b_s} \left(\max(\sigma_{b_s}^{(s)}, \epsilon) \right)^{-1} V_{a''_s b_s}^{(s)} \\ &\times \left([F_m^{(s2)}]_{\beta' \beta} [F_m^{(s3)}]_{\gamma' \gamma} U_{a'_s \beta \gamma}^{(s)} - U_{a_s \beta' \gamma'}^{(s)} [f_m^{(s)}]_{a_s a'_s} \right), \end{aligned} \quad (42)$$

and the multiplicative inverse is now always achievable. The key aspect of this regularization is that it introduces an error of $\mathcal{O}(\epsilon^2)$ in $D^{(s)}(t)$ [cf. Eq. (39)].

2. Projector-splitting propagator

Another branch of the propagation method is based on the so-called projector-splitting (PS) technique. These techniques avoid the errors introduced by regularization but introduce Trotterization errors inherent to the approach. In a PS algorithm, instead of propagating all the core tensors at the same time as in the direct integration, the dynamics of each tensor is propagated individually and sequentially.

The details of this algorithm, and proof of its validity, are discussed in the studies of tensor train and tensor tree^{86–89,103,104} in the context of time-evolution of the matrix product state for a wavefunction. The generalization of a one-site version of this algorithm (PS1) to HEOM with a tree tensor network can be found in Ref. 84. Here, we outline these algorithms and how they are used in TTN-HEOM and generalize the two-site version (PS1) of this algorithm to the TTN-HEOM. PS1 is a static-rank method fixed memory algorithm where the rank is constant during propagation. In turn, the PS2 that we generalize is a dynamic-rank method that updates the rank in the TTN to achieve a target propagation accuracy.

The formal solution of the master equation $\frac{d}{dt} \Omega(t) = \mathcal{L}(t) \Omega(t)$ is $\Omega(t + \Delta) = e^{\Delta \mathcal{L}(t)} \Omega(t)$ for a small time step Δ . In a Trotterization scheme in PS, $\mathcal{L}(t)$ is split into $\mathcal{L}(t) = \sum_{i=1}^{I_{\max}} \mathcal{P}_i \mathcal{L}(t)$. The Trotter propagator is $\Omega(t + \Delta) \approx e^{\Delta \mathcal{P}_{I_{\max}} \mathcal{L}(t)} \dots e^{\Delta \mathcal{P}_1 \mathcal{L}(t)} \Omega(t)$ to first order in Δ , or $\Omega(t + \Delta) \approx e^{\frac{\Delta}{2} \mathcal{P}_1 \mathcal{L}(t)} \dots e^{\frac{\Delta}{2} \mathcal{P}_{I_{\max}} \mathcal{L}(t)} e^{\frac{\Delta}{2} \mathcal{P}_{I_{\max}} \mathcal{L}(t)} \dots e^{\frac{\Delta}{2} \mathcal{P}_1 \mathcal{L}(t)} \Omega(t)$ to second order in Δ . We employ the second Trotter, where each time step is divided into a forward step in the splitting of $\mathcal{L}$, $e^{\frac{\Delta}{2} \mathcal{P}_{I_{\max}} \mathcal{L}(t)} \dots e^{\frac{\Delta}{2} \mathcal{P}_1 \mathcal{L}(t)}$, followed by a backward step in such splitting $e^{\frac{\Delta}{2} \mathcal{P}_1 \mathcal{L}(t)} \dots e^{\frac{\Delta}{2} \mathcal{P}_{I_{\max}} \mathcal{L}(t)}$. We denote each $e^{\tau \mathcal{P}_i \mathcal{L}(t)}$ as a split-step with a time τ .

In the PS method, at each split-step, the TTN is transformed such that the propagation is always on the root tensor, for which the dynamics does not have the singularity issue. [cf. Eq. (19)]. For this, one needs to construct $f_m^{(s)}$ [cf. Eq. (21)] during the propagation. Suppose at a split-step that the root is now located at $A^{(r)}$; then the master equation is

$$\frac{d}{dt} A_{\alpha' \beta' \gamma'}^{(r)} = \sum_m \sum_{\alpha \beta \gamma} [F_m^{(r1)}]_{\alpha' \alpha} [F_m^{(r2)}]_{\beta' \beta} [F_m^{(r3)}]_{\gamma' \gamma} A_{\alpha \beta \gamma}^{(r)}, \quad (43)$$

where $F_m^{(rk)}$ acts on the k th index of $A^{(r)}$. The value of $F_m^{(rk)}$ is $h_m^{>}$ if the k th index of $A^{(r)}$ is i , $h_m^{<}$ if it is j , $h_m^{(k)}$ if it is n_k , and $f_m^{(s)}$ if it is a_s . The transformation of the TTN moves the location of the root tensor from its current location to one adjacent semi-unitary tensor via additional SVD. In one step of the PS propagator, we start from the original root and travel over every core tensor in the TTN. Once all core tensors in the TTN have been updated, the algorithm then returns the root tensor to its original position and proceeds to take another dynamical time step.

In the rest of this section, the Einstein summation convention is assumed.

a. PS1 algorithm. The PS1 algorithm we implemented is in a second-order Trotter propagator form. The key of the algorithm is to find a round-trip path over the whole tensor tree such that each contracted *bond* in the tree is traveled exactly two times. This can be performed by the depth-first-search algorithm¹⁰⁵ from the root $A^{(0)}$. We first travel over the tree: start from the root $A^{(0)}$, go past every closed bond twice, and return to the origin $A^{(0)}$. The forward path is a sequence $P = (A^{(0)}, U^{(1)}, \dots, U^{(K-1)}, \dots, U^{(1)}, A^{(0)})$. We propagate the whole TTN by $\Delta/2$ when we travel along the forward path. After that we use the reversed sequence of the forward path P as the backward path to propagate another $\Delta/2$ to finish one step of propagation for the whole TTN. We use $P[i]$ to represent the core tensor at the i th location of P , and the range of $i = 1, \dots, 2K - 1$ for the order-3 TTN as in Eq. (15).

Suppose that in the TTN the root tensor is $A^{(r)}$ and one of its neighbors is $U^{(s)}$. The one-site move function $\text{move1}(r, s, \tau)$ with a split-step time τ is shown in Algorithm 1.

This algorithm generates $f_m^{(r)}$, a semi-unitary $U^{(r)}$ such that

$$U_{\alpha \beta \alpha'}^{(r)*} U_{\alpha \beta \alpha'}^{(r)} = \delta_{\alpha' \alpha}, \quad (44)$$

and the new root tensor $A^{(s)}$. For the specific case when the time step $\tau = 0$ (which is needed below), it is equivalent to skipping the propagation at line 1 in Algorithm 1.

ALGORITHM 1. One-site move function move1.

```

// Without loss of generality, suppose that the indices in
A(r) and U(s) are Aαβas(r) and Uasγe(s).
1 Perform the SVD of Aαβas(r) = WαβbσbVasb*.
2 Let Uαβar(r) ← Wαβar.
3 Let [fm(r)]ar'ar ← Uαβar(r)*[Fm(r1)]α'α[Fm(r2)]β'βUαβar(r).
4 Let Maras ← σarVasar*.
5 Propagate M by τ using the master equation
  d/dt Mar'as' = [fm(r)]ar'ar[fm(r)]ar'arMaras.
6 Let Aarγe(s) ← MarasUasγe(s).
7 Delete A(r), U(s) and fm(r).

```

The iterative PS1 algorithm for the forward step in the splitting of $\mathcal{L}$ is shown in Algorithm 2, and the backward one in Algorithm 3.

b. PS2 algorithm. In the two-site PS algorithm (PS2), it is possible to dynamically update the rank, i.e., the dimensionality of the dynamical space, in the TTN by constructing 4-order tensors, propagating them, and then decomposing them back to the 3-order tensor structure. The forward steps and backward steps for PS2 are similar to those in PS1, but PS2 implements a two-site move of the root tensor in the split steps in addition to the one-site move.

Suppose that in the TTN the root tensor is $A^{(r)}$ and one of its neighbor $U^{(s)}$. The two-site move function $\text{move2}(r, s, \tau)$ with a split-time τ is showed in Algorithm 4.

In the SVD at line 2 in Algorithm 4, to control the rank of a_r after the move, in practice we use a truncated version of SVD such that the range of b is only for those $\sigma_b > \epsilon'$ as

$$M_{\alpha\beta\gamma e} \approx \sum_{\substack{b \text{ s.t.} \\ \sigma_b \geq \epsilon'}} W_{\alpha\beta b} \sigma_b V_{\gamma e b}^*, \quad (45)$$

where ϵ' is a parameter that controls the error in the truncated SVD. Similarly to move1 , for the specific case of time step $\tau = 0$, it is equivalent to skipping the propagation at line 2 in Algorithm 4.

ALGORITHM 2. Forward step of PS1.

```

1 for i ← 1, 2, ..., 2K - 2 do
2   Suppose P[i] is A(r), and P[i + 1] is U(s).
   // Compare the heights of A(r) and U(s).
3   if L(A(r)) < L(U(s)) then
4     Call move1(r, s, 0) to get U(r) and A(s).
5   else
6     Propagate A(r) by Δ/2 using Eq. (43).
7     Call move1(r, s, -Δ/2) to get U(r) and A(s).
8   end if
9   Let P[i] ← U(r), and P[i + 1] ← A(s).
10 end for
11 Propagate A(0) by Δ/2 using Eq. (43).

```

ALGORITHM 3. Backward step of PS1.

```

1 Propagate A(0) by Δ/2 using Eq. (43).
2 for i ← 2K - 1, 2K - 2, ..., 2 do
3   Suppose P[i] is A(r), and P[i - 1] is U(s).
4   if L(A(r)) < L(U(s)) then
5     Propagate A(r) by Δ/2 using Eq. (43).
6     Call move1(r, s, -Δ/2) to get U(r) and A(s).
7   else
8     Call move1(r, s, 0) to get U(r) and A(s).
9   end if
10  Let P[i] ← U(r), and P[i - 1] ← A(s).
11 end for

```

ALGORITHM 4. Two-site move function move2.

```

// Without loss of generality, suppose the indices in
A(r) and U(s) are Aαβas(r) and Uasγe(s).
1 Let Mαβγe ← Aαβas(r)Uasγe(s).
2 Propagate M by τ using the master equation
  d/dt Mαβγe = [Fm(r1)]α'α[Fm(r2)]β'β[Fm(s2)]γ'γ[Fm(s3)]ε'εMαβγe.
3 Perform the SVD of Mαβγe = WαβbσbVγe b*.
4 Let Uαβar(r) ← Wαβar.
5 Let [fm(r)]ar'ar ← Uαβar(r)*[Fm(r1)]α'α[Fm(r2)]β'βUαβar(r).
6 Let Aarγe(s) ← σarVγe ar*.
7 Delete A(r), U(s) and fm(r).

```

ALGORITHM 5. Forward step of PS2.

```

1 for i ← 1, 2, ..., 2K - 2 do
2   Suppose P[i] is A(r), and P[i + 1] is U(s).
3   if L(A(r)) < L(U(s)) then
4     Call move1(r, s, 0) to get U(r) and A(s).
5   else
6     Call move2(r, s, Δ/2) to get U(r) and A(s).
7     Propagate A(s) by -Δ/2 using Eq. (43).
8   end if
9   Let P[i] ← U(r), and P[i + 1] ← A(s).
10 end for
11 Propagate A(0) by Δ/2 use Eq. (43).

```

The iterative PS2 algorithm for the forward step in the splitting of $\mathcal{L}$ is shown in Algorithm 5, and the backward one is shown in Algorithm 6.

The adaptive rank is dictated by our criterion in Eq. (45) in PS2. This criterion is useful for the bulk of the dynamics. However, for trees that contain core tensors with all three bonds connected, the criterion needs modification at initial times. This is because the tensor rank of $A^{(r)}$ is one for the initial HEOM state Eq. (25). In Algorithm 4, if $A^{(r)}$ is of tensor rank one, then together with the semi-unitary property of $U^{(s)}$, the number of non-zero σ_b in the SVD

ALGORITHM 6. Backward step of PS2.

```

1 Propagate  $A^{(0)}$  by  $\frac{\Delta}{2}$  use Eq. (43).
2 for  $i \leftarrow 2K - 1, 2K - 2, \dots, 2$  do
3   Suppose  $P[i]$  is  $A^{(r)}$ , and  $P[i - 1]$  is  $U^{(s)}$ .
4   if  $L(A^{(r)}) < L(U^{(s)})$  then
5     Propagate  $A^{(r)}$  by  $-\frac{\Delta}{2}$  using Eq. (43).
6     Call move2( $r, s, \frac{\Delta}{2}$ ) to get  $U^{(r)}$  and  $A^{(s)}$ .
7   else
8     Call move1( $r, s, 0$ ) to get  $U^{(r)}$  and  $A^{(s)}$ .
9   end if
10  Let  $P[i] \leftarrow U^{(r)}$ , and  $P[i - 1] \leftarrow A^{(s)}$ .
11 end for

```

step Eq. (45) is at most 1, resulting in the new rank R_r to be fixed at 1, no matter whether the propagation of time τ is performed.

To address this challenge, for all times, in practice the range of b is chosen to be twice the number that satisfies Eq. (45), where the order of the SVD is chosen such that $\sigma_1 > \sigma_2 > \dots \geq 0$.

3. Remarks

In the same way that for wavefunction propagation there is no one propagation scheme that is better in all physical problems,^{106,107} we expect that for the TTN-HEOM the three proposed methods—direct integration, PS1, and PS2—will have specific regimes in which they have favorable properties. The advantage of PS propagators over direct integration is that it avoids the regularization error controlled by ϵ . The disadvantage of the PS propagator is that it requires sequential SVDs during the dynamics, which makes the algorithm more difficult to parallelize.^{103,104} Furthermore, since it individually propagates components of the tensor tree with a given time step Δ , its propagation error is of $\mathcal{O}(\Delta^3)$, which is comparable to Trotter error. By contrast, for a Runge–Kutta method of order n , the direct integration can compute with the integration error $\mathcal{O}(\Delta^n)$.

Both direct integration and PS1 are limited by the assumption that the complete dynamics can be described by the TTN with a given rank. By contrast, in PS2, the ranks are variable during the dynamics from a truncated SVD controlled by an error of ϵ' , which makes it possible to change the size of the TTN accordingly.

E. Implementation considerations and capabilities

We implemented the TTN-HEOM in a Python package, Tensor Equations for Non-Markovian Structured Open (TENS0) systems, using the popular NumPy⁵⁸ and PyTorch⁵⁹ libraries for the tensor data structure and tensor operations, as well as `torchdiffeq`¹⁰⁸ for integrating the quantum master equations for tensors.¹⁰⁹ These packages offer high-level protocols for ease of programming that are compatible with various computational platforms such as CPUs and GPUs of different architectures. Details of the implementation will be provided in a subsequent publication; however, here we describe some of its key elements for TTN-HEOM.

TENS0 admits system Hamiltonians with any level structure and arbitrary time dependence, making it of utility to investigate driven open quantum systems. The TENS0 implementation admits arbitrary order for the core tensors and arbitrary tree structure. As

such, it goes beyond the order-3 tensor equations discussed in Secs. II B–II D, and beyond tensor-train approaches to the HEOM. In the [supplementary material](#), we detail this generalization using the language of graphs.

The system–bath coupling can include any number of terms $H_{SB} = \sum_d Q_S^{(d)} \otimes X_B^{(d)}$, and the $\{Q_S^{(d)}\}$ do not need to commute. Therefore, TENS0 can be used to investigate a system that is coupled to two or more environments through non-commuting operators, something that is computationally challenging to adopt in path integral-based transfer tensor strategies.

This package currently implements the three propagation strategies discussed in Sec. II D: direct integration of the quantum master equations with fixed ranks and the stepwise projector-splitting propagator, including PS1 with fixed ranks and PS2 for variable ranks during propagation.

To run a simulation using TENS0, in the input one needs to specify the parameters c_k , $\tilde{c}_k$, γ_k in the decomposition of the BCF Eq. (6). Any decomposition compatible with Eq. (6) can be used. For common spectral density models, including Drude–Lorentz and underdamped Brownian oscillator, we have implemented a helper function to obtain these parameters in Eq. (6) using either a Padé⁹⁸ or Matsubara⁹⁷ expansion for the thermal factor $\coth(\omega/2k_B T)$. The helper function gives both the high-temperature terms from the model spectral densities and an arbitrary order of low-temperature correction terms from the expansion from the thermal factor. Our code is also compatible with other BCF decomposition strategies that yield the form in Eq. (6).^{50–57}

To make use of the flexibility of the TTN, in the input one can also specify the topology of a TTN with the open bonds corresponding to all system and exciton indices. The topology that is chosen for the TTN will automatically determine the quantum master equations for the core tensors. The code admits as input a list of the nodes in the TTN and their connectivity to either open bonds or to other nodes. We implemented templates for automatically generating the train topology as exemplified in Fig. 1(a) and the balanced tree topology as exemplified in Figs. 1(b) and 1(c). However, TENS0 admits as input any type of TTN with open ends $i, j, n_1, \dots, n_K$, with any one of the core tensors specified as the root initially.

Each tree structure has a unique version of the quantum master equations (19)–(24). These equations have common quantities that, for computational efficiency, must be evaluated in a specified order to avoid duplication of efforts. The order in which the $\{f_m^{(s)}(t)\}$ are computed is based on the structure of the tree. We compute the $\{f_m^{(s)}\}$ from $s = K - 1$ to 1. This can be seen in Eq. (21), which shows that $f_m^{(s)}$ only depends on $f_m^{(u)}$ with $u > s$. That is, the computation of $f_m^{(s)}$ proceeds from the leaves of the TTN (the nodes with open bonds $n_1, \dots, n_K$) to the root.

Furthermore, in the direct integration propagation method with regularization, we need to evaluate $\{\bar{D}_m^{(s)}(t)\}$, as well as $\{\sigma^{(s)}(t)\}$ and $\{V^{(s)}(t)\}$, for integrating Eq. (42). In this case, we proceed from the root to the leaves. That is, we go from $s = 1$ to $K - 1$ to evaluate all $\bar{D}_m^{(s)}(t)$, $\sigma^{(s)}(t)$, and $V^{(s)}(t)$ for Eq. (42). This can be seen from Eqs. (30)–(36), as $\bar{D}_m^{(s)}$, $\sigma^{(s)}(t)$, and $V^{(s)}(t)$ only depend on $\bar{D}_m^{(r)}$, $\sigma^{(r)}(t)$, and $V^{(r)}(t)$ with $r < s$. In this way, we provide a systematic sequential procedure to travel through the TTN and construct the needed $f_m^{(s)}$, $\bar{D}_m^{(s)}$. For the direct integration, this needs to

be performed whenever the derivative of the core tensors is computed. For PS, this needs to be performed before each forward step in the algorithm.

Each propagation step runs over the whole TTN of $\Omega(t)$ that includes the dynamical information of the system and the collection of bexcitons. The reduced density operator of the system $\rho_s(t)$ is calculated from the $\Omega(t)$ for output times as

$$[\rho_s(t)]_{ij} = \sum_{n_1 \dots n_K} [\text{Con}(A^{(0)}(t), U^{(1)}(t), \dots, U^{(K-1)}(t))]_{ijn_1 \dots n_K} \times \delta_{0n_1} \dots \delta_{0n_K}. \quad (46)$$

In practice, the expression of TTN in Eq. (15) is substituted in Eq. (46). To avoid reconstructing the full high-order EDO, the contractions are first performed for the n_k indices and then from a_{K-1} to a_1 .

Specifically, this is performed by constructing a series of vectors $\mathbf{t}^{(s)}$ from $s = K - 1$ to 1. The recursive definition of $\mathbf{t}^{(s)}$ is

$$\mathbf{t}_a^{(s)} \equiv \sum_{\beta\gamma} [\mathfrak{T}_m^{(s2)}]_{\beta} [\mathfrak{T}_m^{(s2)}]_{\gamma} U_{a,\beta\gamma}^{(s)}. \quad (47)$$

Here, the definition of vector $\mathfrak{T}^{(sk)}(t)$ for $\kappa = 2$ and 3 depends on the location of the tensor $U^{(s)}(t)$ in the TTN. $\mathfrak{T}^{(sk)}(t) \equiv \mathbf{t}^{(u)}(t)$ ($u > s$) if the κ th index corresponds to a contracted index or bond in the TTN. That is, when the κ th index in $U^{(s)}(t)$ is an a_u in Eq. (15). In turn, $\mathfrak{T}_{n_k}^{(sk)}(t) \equiv \delta_{0n_k}$ when the κ th index is an n_k in Eq. (15). That is, when it corresponds to an open bond in the TTN. Notice that this definition is recursive, as $\mathbf{t}_m^{(s)}(t)$ depends on $\mathfrak{T}^{(sk)}(t) = \mathbf{t}^{(u)}(t)$ ($u > s$) if the κ th index corresponds to a contracted bond [cf. Eq. (21)]. After the construction of $\mathbf{t}^{(s)}(t)$, the reduced density operator of the system $\rho_s(t)$ is calculated as

$$[\rho_s(t)]_{ij} = \sum_{a_1} [A^{(0)}(t)]_{ija_1} [\mathbf{t}^{(1)}(t)]_{a_1}. \quad (48)$$

In this way, the explicit evaluation of the full high-order EDO tensor is avoided.

The HEOM is a numerically exact method for a given decomposition of the BCF Eq. (5). However, by construction it does not guarantee positivity of the reduced density operator of the system and, in fact, negativities can occur when employing inaccurate BCFs. Since the TTN-HEOM is a decomposed version of HEOM, it can be numerically exact, but its overall accuracy will also be limited by the quality of the spectral density that is employed in the model.

In contrast to HEOM, TTN-HEOM can be efficiently employed with highly structured spectral density and with low-temperature corrections, as needed to perform computations in chemically realistic systems. Our efforts complement a tensor train implementation of the HEOM¹¹⁰ and a recent ML-MCTDH software package with HEOM capabilities.¹¹¹

III. NUMERICAL EXAMPLE

A. Model

To illustrate the TTN-HEOM, we consider a two-electronic surface molecular system described by a two-level model coupled to

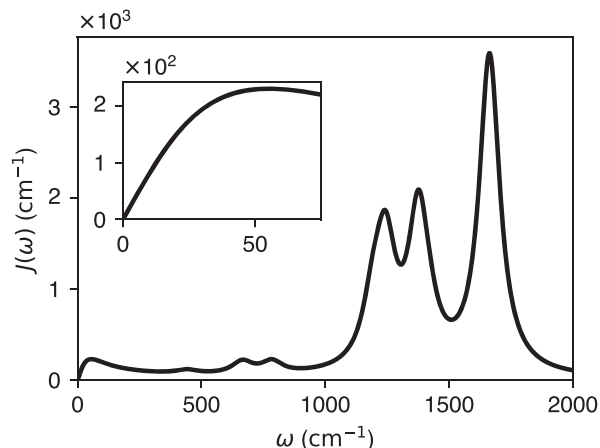


FIG. 2. Bath spectral density $J(\omega)$ describing electron–nuclear interactions for thymine nucleotide in room temperature water⁹² with broadenings $\gamma_b = 50 \text{ cm}^{-1}$ for each Brownian oscillator.

a structured thermal bath. In the Hamiltonian, the electronic system is

$$H_S = \frac{E}{2}(|1\rangle\langle 1| - |0\rangle\langle 0|) + V(|1\rangle\langle 0| + |0\rangle\langle 1|), \quad (49)$$

where $|0\rangle$ and $|1\rangle$ denote two diabatic electronic states, E is the energy level difference between them, and V is their electronic coupling. In turn, the system is coupled to the bath via

$$Q_S = \frac{1}{2}(|1\rangle\langle 1| - |0\rangle\langle 0|). \quad (50)$$

That is, the bath is assumed to introduce energy fluctuations between $|0\rangle$ and $|1\rangle$. As an initial state, we take the system to be in a pure superposition state of the form $|\psi_S\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$.

To characterize the system–bath interaction, in numerical exact simulation it is common to use simple model spectral densities, such as the Drude–Lorentz or the Brownian oscillator. As a computationally challenging example, in this paper we adopt the realistic spectral density⁹² shown in Fig. 2 recently extracted from resonance Raman experiments for thymine nucleotide in room temperature

TABLE I. Parameters in the spectral density Eq. (51) characterize the bath for thymine nucleotide in water at 300 K. Parameters are taken from Ref. 92.

b	$\omega_b' \text{ (cm}^{-1}\text{)}$	$\lambda_b \text{ (cm}^{-1}\text{)}$	$\gamma_b \text{ (cm}^{-1}\text{)}$
0	...	715.73	54.45
1	1663	330.0	50
2	1416	25.6	50
3	1376	186.0	50
4	1243	161.7	50
5	1193	77.3	50
6	784	26.5	50
7	665	32.0	50
8	442	14.9	50

water. This spectral density consists of one Drude–Lorentz component at low-frequencies that describes the solvent and 8 Brownian oscillators at higher frequencies that represent the interaction of the electronic system with intramolecular vibrations. That is, the bath spectral density is

$$J(\omega) = J_{\text{DL}}(\omega) + \sum_{b=1}^8 J_{\text{B}}^{(b)}(\omega), \quad (51)$$

with $J_{\text{DL}}(\omega) = \frac{2\lambda_0}{\pi} \frac{\gamma_0 \omega}{\omega^2 + \gamma_0^2}$ and $J_{\text{B}}^{(b)}(\omega) = \frac{4\lambda_b}{\pi} \frac{\gamma_b \omega_b^2 \omega}{(\omega^2 - \omega_b^2)^2 + 4\gamma_b^2 \omega^2}$. Here, λ_0 is the reorganization energy of the solvent and γ_0^{-1} its relaxation time. In turn, λ_b is the reorganization energy of the b th vibrational mode, ω_b its natural frequency, γ_b^{-1} its lifetime, and $\omega_b' = \sqrt{\omega_b^2 - \gamma_b^2} > 0$ its effective frequency under damping. Spectral density parameters are listed in Table I.

B. Tensor tree and bexcitonic choices

For the TTN, we use either a balanced binary tree or a tensor train, both of them containing order-3 core tensors only. The balanced tree structure, in particular, minimizes the average distance between the index of each bexciton n_k and the indices of the system i, j . To obtain the correct thermal state, we include 3 low temperature correction terms from the Padé expansion to evaluate Eq. (6). This results in 20 overall bexcitons in the HEOM, and the resulting balanced tensor tree structure is shown in Fig. 1(c). Since the index k for different terms in the BCF decomposition Eq. (6) is arbitrary, the correspondence of n_k to different parts of the spectral density is not unique. Here, we choose n_1 to correspond to the high-temperature Drude–Lorentz, n_2 – n_{17} to the high-temperature Brownian oscillators, and n_{18} – n_{20} to the overall low temperature corrections. The Brownian oscillators are sorted in descending order of their frequencies. Each Brownian oscillator requires two bexcitons to be described, while the Drude–Lorentz feature requires just one. As a metric in Eq. (9), we employ $\hat{z}_k = i\sqrt{\text{Rec}_k}$.

C. Open quantum dynamics of the model

To test the performance of TTN-HEOM under different system settings, we set $V = 1000 \text{ cm}^{-1}$ and change the energy gap E from 0 to 5000 cm^{-1} . We monitor the dynamics through the population of states $|0\rangle$, $[\rho_S]_{00}$, and the purity $\text{Tr}(\rho_S^2)$, which is a basis-independent measure of coherence (i.e., purity = 1 for a pure system, <1 for mixed states, and 1/2 for a maximally mixed two-level system).

Figure 3 shows the converged purity and $[\rho_S]_{00}(t)$ dynamics for the two-surface molecules with varying E . The system undergoes an initial decay of purity due to interaction with the bath until it reaches a minimum around 0.5. Subsequently, the purity recovers as the system relaxes to thermal equilibrium. For early times, the decay of purity is Gaussian and independent of the details of the system Hamiltonian. In agreement with the theory of early decoherence time scales,^{112,113} this segment of the dynamics just depends on the initial-time quantum and thermal fluctuations of the operators coupling the system and bath. The subsequent purity oscillations are due to the population transfer between $|0\rangle$ and $|1\rangle$, which are beyond the short-time limit. For longer times $t > 100 \text{ fs}$, the purity and population oscillate as the system relaxes to thermal equilibrium. These

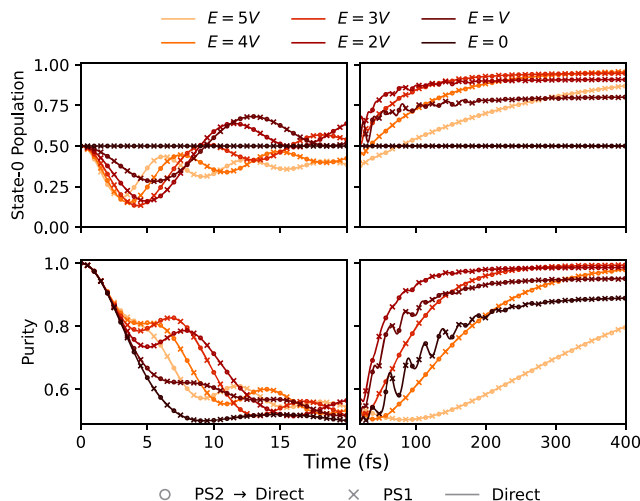


FIG. 3. TTN-HEOM dynamics captured by different propagation methods for the two-level system in Eq. (49) interacting with a highly structured thermal environment (Fig. 2) using the balanced tree in Fig. 1(c). Different colors denote varying model parameters. The direct (solid line) and PS1 (crosses) integration use a rank of 60 for all tensors. The mixed propagator (PS2 → direct, circles) starts with the PS2 propagator with an initial rank 3 and switches to direct integration when the adaptive rank grows beyond 60. Note the convergence and stability of the dynamics for all model parameters and integrators.

deviations from exponential dynamics are clear signatures of non-Markovian open quantum dynamics that persist even for long times for this highly structured bath.

These results demonstrate that the TTN-HEOM can capture the numerically exact open quantum dynamics of systems interacting with highly structured thermal environments. We further note that the TTN-HEOM and HEOM yield identical results. While HEOM computations for highly structured environments like those in Fig. 3 are not tractable, we numerically illustrate in Fig. S4 in the supplementary material the coincidence between TTN-HEOM and HEOM using only the Drude–Lorentz component in the bath spectral density.

D. Propagator choice

Figure 3 shows that using TENS0 we can obtain identical dynamics with the three implemented propagation strategies. For the PS1 and direct integration, converged results are achieved with a moderate rank $R = 60$ for all R_k and a depth $N = 20$ for all N_k . For the direct integration, the integration of all core tensors is calculated simultaneously with a regularizing parameter $\epsilon = 10^{-4}$ using the RK4(5) method, which allows for adaptive time step h during the propagation. This method is of $\mathcal{O}(h^4)$ with an error estimator of order $\mathcal{O}(h^5)$ used to determine the integration time step h . For the PS1 and PS2 methods, a fixed timestep Δ of 0.1 fs is applied for splitting the propagation as described in Algorithms 2, 3, 5, and 6, while the integration of each low-order tensor is calculated by RK4(5). Furthermore, in the PS2 method the truncated SVD is performed with an $\epsilon' = 10^{-7}$. Here, in all RK4(5) integrators, the relative error tolerance is 10^{-5} and the absolute error tolerance is 10^{-7} .

The direct integration strategy offers a practical approach for propagating the bulk of the dynamics. However, it is numerically challenging in the initial stage (<2 fs), requiring extremely small time steps (<0.0001 fs). This is because we start from an initially separable system–bath state, which requires regularization to remove the singularity issues in evaluating Eq. (20). This regularization introduces a small artificial error when this singularity occurs and affects the stability and accuracy of the numerical integration. Once this initial stage is overcome, the matrix $D^{(s)}$ becomes numerically invertible, as all the eigenvalues λ_i in matrix $D^{(s)}$ are greater than zero. That is, if the regularization constant ϵ satisfies $\epsilon \leq \min_i \sqrt{\lambda_i}$, then the regularization scheme Eq. (42) becomes a numerically exact method to calculate the inverse of $D^{(s)}$ in Eq. (20).

PS1 is a robust strategy to propagate the TTN-HEOM and a common choice for tensor network methods. The main challenge is that it incurs Trotterization errors of $\mathcal{O}(\Delta^3)$ in addition to the integration errors within each split-step in $\mathcal{O}(h^4)$ with the actual integration time $h \leq \frac{\Delta}{2}$. As direct integration, PS1 requires a list of initial ranks to capture the entanglement between different core tensors, and the convergence with rank requires performing repeated calculations.

In turn, in PS2 the ranks change adaptively during propagation starting from an initial given rank for each contracted bond. The algorithm has the advantage of adapting the ranks as needed to accurately capture the dynamics and, thus, has variable memory requirements. Specifically, the ranks change such that the error introduced in the SVD Eq. (45) is consistent with the control parameter ϵ' . These ranks change in a non-uniform fashion, as the rank of some bonds can be larger than others. Our PS2 propagation starts with minimal ranks. Therefore, these ranks initially grow using PS2 but, eventually, as the dynamics progress, can also decrease. Overall, for a TTN with order- M core tensors, the PS2 contains the propagation of $2M - 2$ tensors, which is of higher-order computational complexity.

These three methods can be combined on-the-fly to construct strategies that leverage their strengths and overcome their limitations. There is significant flexibility in combining them, as they only require the state of the TTN at the specific propagation time. For instance, one straightforward PS2 $\rightarrow$ direct strategy is to use the PS2 at initial times followed by direct integration. This mixed strategy has the benefit of determining the proper requirement for the ranks for each bond from the early time dynamics and avoiding the initial singularity in TTN-HEOM that limits the direct propagation strategy. Once the required computational resource requested by PS2 exceeds a threshold level, one switches to the direct integration method that has the advantage of allowing integration with higher order adaptive time step methods compared to the order of Trotterization errors in PS1 and PS2.

The results from this mixed PS2 $\rightarrow$ direct strategy approach are also shown in Fig. 3. In this case, all core tensors in the TTN start with a rank of 3 and use the PS2 propagator until the maximum rank in the TTN reaches 60. After that, the remaining dynamics are propagated using direct integration. We find that, compared to the direct integration with regularization and PS1 with all ranks to be the same, the mixed PS2 $\rightarrow$ direct strategy can achieve the converged results with reduced computational resources. For instance, Fig. 4 shows the computation time of each of the propagation strategies in Fig. 3 for

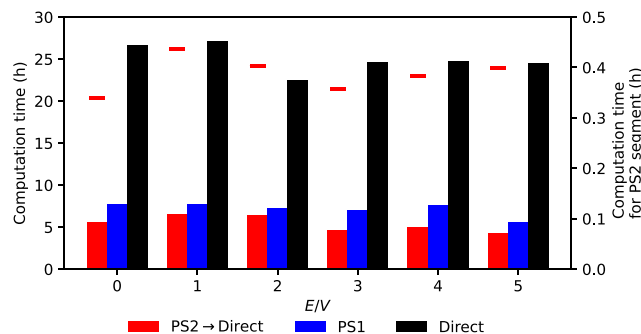


FIG. 4. Computation time of each propagation strategy in Fig. 3 on 8 cores of an Intel Xeon Gold 6330 Processor. Each bar shows the computation time of different propagation strategies for the initial 100 fs of dynamics (left axis). The dashes mark the computation time for the PS2 segment (~ 2 fs) of the PS2 $\rightarrow$ direct method (right axis).

the initial 100 fs of dynamics (run on 8 cores of an Intel Xeon Gold 6330 Processor). By adopting the PS2 for the initial propagation, the mixed strategy with direct integration achieves the best performance while still providing numerically converged results. The reason for this is that the effective simulation space identified by PS2, where at least one bond has a rank of 60, is smaller than the one used for PS1 and direct, where all bonds have a rank of 60.

For the chosen propagation parameters, PS1 is actually faster than direct integration with the same size of ranks in the TTN. This is because the main computational effort in the direct integration is in evaluating the $\{f_m^{(s)}\}$ and $\{\tilde{D}_m^{(s)}\}$, which requires a series of sequential SVDs that are not parallelized. This is similar to the sequential SVDs required in the PS1. However, the difference is that, while in PS1 the $f_m^{(s)}$ are evaluated only once during Trotterization time Δ , in the direct integrator the $f_m^{(s)}$ and $\tilde{D}_m^{(s)}$ are constructed whenever the derivatives of all core tensors are requested by the high-order numerical integrator, and this occurs several times per integration time step $h < \Delta$. Therefore, the direct integrator is slower than the PS1 for the same size of TTN, as it contains more matrix construction steps. However, if we use the mixed strategy where the most part (>2 fs) of the dynamics is obtained by the direct integration, it actually becomes faster than PS1. This is because the direct integrator inherits the non-uniform ranks from the initial PS2 segment that reduces the overall simulation space.

Figure 5 shows the effect of adjusting the rank threshold in the mixed propagation strategy. The other parameters are the same as in Fig. 3. The figure shows that for the first 100 fs convergence is achieved with a rank limit of 40, while longer dynamics require a rank of 60. As expected, increasing the accuracy of the TTN by increasing the ranks achieves longer converged dynamics. The maximum rank needed for convergence depends on the system Hamiltonian, revealing that a larger system energy gap demands a higher rank for convergence. We hypothesize that as the gap increases, the influence of the bath becomes increasingly Markovian, as revealed by the reduction of the partial purity oscillations for long times. However, this Markovian limit is more challenging to capture for this numerically exact non-Markovian TTN-HEOM method.

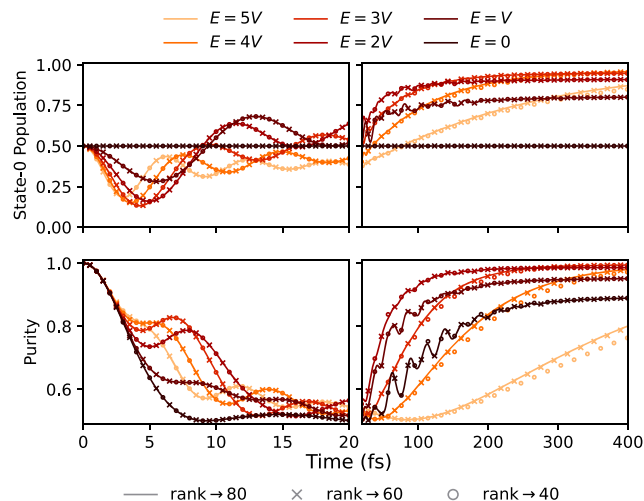


FIG. 5. TTN-HEOM dynamics for varying maximum rank thresholds for the system and TTN in Fig. 3 using the mixed propagator (PS2 $\rightarrow$ direct). The propagator starts with PS2 with an initial rank 3 and switches to direct when one of the adaptive ranks grows beyond 80 (solid lines), 60 (crosses), and 40 (circles).

E. Tree structure choice

Different TTN structures are expected to influence the computational cost of the TTN-HEOM dynamics and the effectiveness of the method to compress the open quantum dynamics. However, given a tree structure, it is challenging to optimize the ranks for efficient computation. The PS2 strategy has the advantage of automatically adapting the compression in different components of the tree to satisfy the criteria of Eq. (45).

To investigate the influence of the tree structure on the computational resources requested by the PS2 $\rightarrow$ direct propagation, we

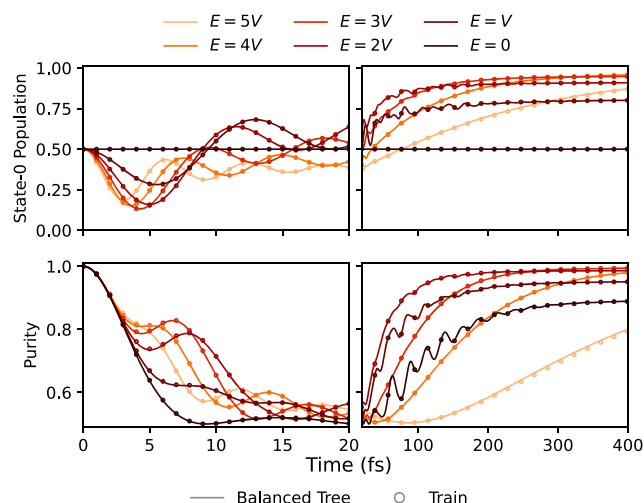


FIG. 6. Balanced tensor tree vs tensor train TTN-HEOM dynamics for the system in Fig. 3 using the mixed strategy (PS2 $\rightarrow$ direct) with maximum rank threshold 80. Note that the dynamics is independent of the tree structure.

performed TTN-HEOM simulations for the same model as in Fig. 3 using this mixed propagation strategy using a tensor tree and tensor train. The tensor train scheme is the one shown in Fig. 1(a) with $K = 20$, and the tree scheme is the one shown in Fig. 1(c). Figure 6 shows that, as expected, the open quantum dynamics is independent of the TTN employed.

Figure 7 shows the growth of the maximal rank and size (overall number of core tensor elements) of the TTN with these two TTN structures for the dynamics in Fig. 6. The maximal rank increases during the PS2 propagation until it satisfies the threshold and changes to the direct method of fixed rank. Overall, the size of the TTN in the tensor train [Fig. 1(a)] grows faster than that in the tree scheme [Fig. 1(c)] when applying the same error tolerance in the propagation algorithm. This is because in the HEOM, the primary entanglements in the tensor occur between the system and each bath feature. By employing the tensor tree with the system DOFs at the root and the bath features at the leaves, one minimizes the average distance in the TTN between the system and each bath feature to $\mathcal{O}(\log K)$. In turn, for the tensor train, the average distance is $\mathcal{O}(K)$. This offers an example where balanced tensor trees are better suited for TTN-HEOM from a computational cost perspective, which may be because it keeps strongly correlated parts in the TTN closer to one another.⁷⁰ Our TTN includes all possible tensor tree topological structures, with the balanced TTN and the tensor train being two extreme particular cases. We expect that the “optimal” TTN structure should sit in between these two extreme cases.

Finally, we point out that simply storing the needed EDO directly in these simulations is just not possible using present-day and foreseeable computational resources. Table II lists the size of

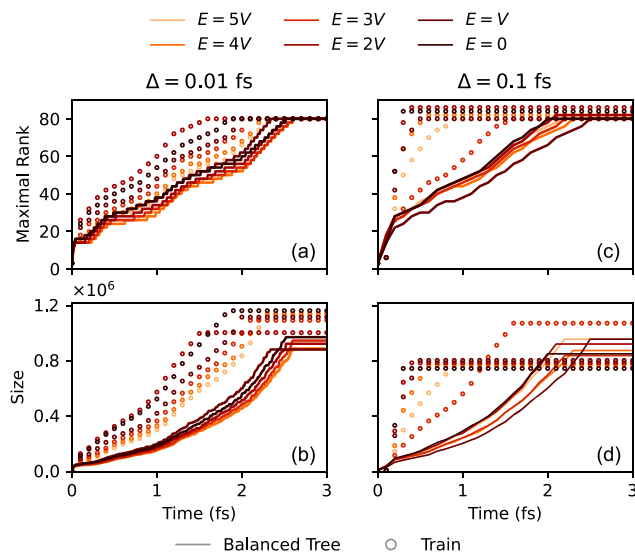


FIG. 7. Maximal rank and size of the TTN during PS2 $\rightarrow$ direct propagation with the maximum rank threshold of 80 for a tensor train (circles) and a tensor tree (solid lines) for the dynamics in Fig. 1(c). TTN size refers to the overall number of core tensor elements. In (a) and (b), the splitting time step Δ is 0.01 fs, while in (c) and (d), it is 0.1 fs. Note that the tensor train size grows faster than the tensor tree scheme under the same SVD tolerance (10^{-7}) in Eq. (45). Although the rank required from PS2 grows rapidly, switching to direct integration with fixed rank does not introduce appreciable errors.

TABLE II. Size (overall number of core tensor elements) of TTN for a bath described by 20 features and a depth of 20 for each N_k . The memory usage of a dense high-order EDO tensor in HEOM is also shown for comparison.

	Balanced tree			Train			Dense HEOM
Rank	40	60	80	40	60	80	...
Size ($\times 10^6$)	0.7	2.2	4.9	0.6	1.3	2.3	4.2×10^{20}

a TTN with fixed rank and compares it to the size of the EDO in conventional HEOM with 20 features and a depth of 20 for each feature. Storing a dense tensor with that size of 4.2×10^{26} would require 6.7 ronnabytes (10^{27}) of memory! Here, the estimation of the dense HEOM reflects the whole uncompressed Hilbert space dimension. In reality, from practical experience, one can use other numerical techniques, such as filtering out near-zero elements,⁴⁸ or use the standard truncation by total depth instead of the depth of each bexciton to shrink the active dynamical space in HEOM more aggressively. The depth of 20 is a conservative choice for this case study, and the greatest lower bound of such truncation depths needs to be determined by actual computations or experiences on a specific physical model. Therefore, either tensor networks or other techniques are necessary for practical simulation based on current classical computers.

IV. CONCLUSION

In conclusion, we introduced TTN-HEOM, a numerically exact quantum master equation method based on the bexcitonic hierarchical equations of motion (HEOM) and a tree tensor network (TTN) decomposition. TTN-HEOM is designed to capture the dynamics of driven open quantum systems interacting with structured bosonic thermal environments, even those of chemical complexity.

The specific advances of this paper are as follows: (1) we introduced a tensor network decomposition of the HEOM based on the bexcitonic generalization. As such, the proposed TTN decomposition applies to all HEOM variants that can be cast into the general bexcitonic form—including the standard HEOM with and without scaling and the collective bath coordinate method—and admits a representation of the bexcitons in number, position, or momentum basis. (2) We showed that the bexcitonic equations of motion can naturally be expressed in sum-of-product form, that the bexcitonic density operator can be decomposed by a tree tensor network, and that a useful set of coupled master equations can be developed for the low-order tensors from Dirac-Frenkel's TDVP. Our developments are analogous to the ML-MCTDH, as the three main design principles (sum-of-product dynamical generator, tree tensor network decomposition, and Dirac-Frenkel's TDVP) are identical. However, while ML-MCTDH is designed for unitary dynamics, the TTN-HEOM is designed for thermal dissipative dynamics. (3) We implemented the TTN-HEOM into a general-purpose code, TENS0, which stands for Tensor Equations for Non-Markovian Structured Open systems. TENS0 admits arbitrary tensor tree structures, including tensor trains and balanced tensor trees, and arbitrary orders for the core tensors. TENS0 includes three numerically stable

propagation strategies (fixed-rank direct integration and PS1 and adaptive rank PS2) for the decomposed master equations based on TDVP. The direct propagation offers adaptive time steps and the use of integration routines with errors that are high-order in the integration time step. In turn, the PS1 and PS2 are based on second-order Trotterization. Specifically, PS1 conserves the size of the TTN by keeping all ranks constant during the dynamics. In turn, PS2 is an adaptive-rank method that updates the size of the TTN according to the error in a truncated SVD step. These strategies can be mixed at will in practical simulations. TENS0 also includes common decompositions of the bath correlation functions, including the low-temperature corrections, for common environmental spectral densities such as the Drude-Lorentz and Brownian oscillator models.

Our TTN-HEOM method can capture both the early times and the asymptotic dissipative dynamics of general quantum systems immersed in thermal environments. This contrasts with some other tensor network techniques, such as the TD-DMRG,⁶¹ ML-MCTDH,²⁶ and T-TEDOPA,⁷⁴ which are unitary in nature and can only mimic the dissipative dynamics for a finite amount of time by explicitly capturing the dynamics of a finite discretized version of the bath.

Our TTN-HEOM method admits arbitrary tree tensor network structures, tensors with orders that vary across nodes, and variable ranks for the core tensors during propagation. This contrasts with recent advances in combining tensor network techniques into HEOM with specific tensor train decompositions.^{78–80,82,83}

With respect to the choice of master equation formalism, the TENS0 package implements propagation strategies based on the general sum-of-product master equation generator Eq. (13). This contrasts with other approaches where the generator of the dynamics is equivalently expressed as a hierarchical sum-of-product form¹¹¹ or strategies where the generator is decomposed as a matrix product operator (MPO)³⁸ or a tree tensor network operator (TTNO) with the same network structure as the extended density operator⁸⁴ (see also Refs. 114 and 115 for a discussion on how to optimize this strategy). An advantage of the TENS0 framework is that it is straightforwardly adaptable to any dynamical master equation method that admits a sum-of-product type of generator. Using it, we thus avoid repeatedly implementing tensor network techniques for other quantum master equations with sum-of-product generators, such as the Lindblad equation and the time-dependent Schrödinger equation.

We demonstrated the self-consistency and utility of TTN-HEOM and TENS0 by capturing the open quantum dynamics of a two-level molecule interacting with a structured thermal environment with a spectral density composed of one Drude-Lorentz and 8 Brownian Oscillator features. Because of computational cost, such a model is well beyond the applicability of standard versions of the HEOM. We show that the dynamics is independent of the tree structure and propagation method, demonstrating the self-consistency of TENS0. Overall, by providing a systematic approach for propagating exact quantum master equations, TTN-HEOM facilitates precise numerical simulations from simulating open quantum dynamics coupled with realistic chemical thermal environments.

We expect that the TTN-HEOM and TENS0 will be useful to understand and emulate the operation of realistic quantum

devices, to engineer quantum environments that enhance molecular function, to isolate molecular qubits with enhanced coherence properties as needed for quantum technologies, to understand elementary steps in photosynthesis and photovoltaics, and to test quantum control strategies in the presence of quantum environments. Future prospects include investigating other combinations for the mixed propagation strategy, implementing an adaptive one-site algorithm^{111,116} as an alternative choice of PS2, and the potential use of auto-differentiation techniques^{59,117} for propagation.

SUPPLEMENTARY MATERIAL

See the [supplementary material](#) for the generalization of the TTN theory and algorithms to general tree topology and a numerical illustration of the coincidence between TTN-HEOM and HEOM for a Drude–Lorentz bath.

ACKNOWLEDGMENTS

This material is based on work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, Quantum Information Science Research in Chemical Sciences, Geosciences, and Biosciences Program under Award No. DE-SC0025334.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Xinxian Chen: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Project administration (equal); Resources (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Ignacio Franco:** Conceptualization (equal); 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).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

APPENDIX A: SKETCH OF DERIVATIONS OF EQS. (19) AND (20)

The derivation of the two master equations (19) and (20) for the TTN in Eq. (15) follows from (i) the TDVP (Eq. (17)), (ii)

the constraint that all $U^{(s)}(t)$ ($0 < s < K$) are semi-unitary tensors [Eq. (16)], and (iii) the gauge condition [Eq. (18)].

We rewrite the TTN decomposition as

$$\Omega_{ijn_1 \dots n_K}(t) = \sum_{a_1} A_{ija_1}^{(0)} \mathcal{U}_{a_1 n_1 n_2 \dots n_K}^{(1)} \quad (\text{A1})$$

Here, $\mathcal{U}^{(1)}$ is the branch of the TTN that contracts with the root $A^{(0)}$ and contains $U^{(1)}$ as

$$\mathcal{U}_{a_1 n_1 \dots n_K}^{(1)} = [\text{Con}(U^{(1)}(t), \dots, U^{(K-1)}(t))]_{a_1 n_1 \dots n_K} \quad (\text{A2})$$

More generally, we define

$$\mathcal{U}_{a_s n_{\ell_1} \dots n_{\ell_{K_s}}}^{(s)} = [\text{Con}(U^{(s)}(t), \dots)]_{a_s n_{\ell_1} \dots n_{\ell_{K_s}}}, \quad (\text{A3})$$

as the branch of TTN that breaks the bond a_s and includes $U^{(s)}$, where $\{\ell_1, \dots, \ell_{K_s}\}$ is a subset of $\{k\}_{k=1}^K$, corresponding to the part of indices n_k that show up in the branch $\mathcal{U}^{(s)}$. In turn, $\mathcal{A}_{ijn_{\ell_{K_s}+1} \dots n_{\ell_K} a_s}^{(s)}$ is another half of the branch of TTN that breaks the bond a_s and includes $A^{(0)}$ and $\{\ell_{K_s+1}, \dots, \ell_K\} = \{k\}_{k=1}^K \setminus \{\ell_1, \dots, \ell_{K_s}\}$, corresponding to the complement part of the indices n_k that show up in the branch $\mathcal{A}^{(s)}$.

In this way, the original EDO $\Omega_{ijn_1 \dots n_K}(t)$ is

$$\begin{aligned} \Omega_{ijn_1 \dots n_K}(t) &= \sum_{a_s} \mathcal{A}_{ijn_{\ell_{K_s}+1} \dots n_{\ell_K} a_s}^{(s)} \mathcal{U}_{a_s n_{\ell_1} \dots n_{\ell_{K_s}}}^{(s)} \\ &\equiv \sum_{a_s} \mathcal{A}_{ia_s}^{(s)} \mathcal{U}_{a_s j}^{(s)}. \end{aligned} \quad (\text{A4})$$

Here, $\mathbf{i}$ and $\mathbf{j}$ are the multi-indices with $\mathbf{i} = \{i, j, n_{\ell_{K_s}+1}, \dots, n_{\ell_K}\}$ and $\mathbf{j} = \{n_{\ell_1}, \dots, n_{\ell_{K_s}}\}$. Notice that $\mathcal{A}^{(1)} = A^{(0)}$ as shown in Eq. (A1).

Because the TTN can be considered as a series of singular value decompositions, and all $U^{(1)}$ are semi-unitary, we have

$$\sum_{\mathbf{j}} \mathcal{U}_{a_s \mathbf{j}}^{(s)*} \mathcal{U}_{a_s \mathbf{j}}^{(s)} = \delta_{a_s' a_s}. \quad (\text{A5})$$

Proof. This can be proved with an induction on the tensor tree.

- (1) Suppose the core tensor $U_{a_s n_k n_l}^{(s)}$ has the indices n_k and n_l on its second and third positions in the TTN decomposition Eq. (15). Then $\mathcal{U}^{(s)} = U^{(s)}$, and Eq. (A5) is an instance of Eq. (16).
- (2) $U_{a_s a_u n_l}^{(s)}$ in the TTN decomposition Eq. (15). Suppose $\mathcal{U}^{(u)}$ has indices $\mathcal{U}_{a_u \mathbf{j}_1}^{(u)}$ where $\mathbf{j}_1 = \mathbf{j} \setminus \{n_l\}$, and

$$\begin{aligned} \sum_{\mathbf{j}} \mathcal{U}_{a_s \mathbf{j}}^{(s)*} \mathcal{U}_{a_s \mathbf{j}}^{(s)} &= \sum_{a_u a_u} \sum_{\mathbf{j}_1 n_l} \mathcal{U}_{a_s \mathbf{j}_1}^{(u)*} U_{a_s' a_u n_l}^{(s)*} U_{a_s a_u n_l}^{(s)} \mathcal{U}_{a_u \mathbf{j}_1}^{(u)} \\ &= \sum_{a_u a_u n_l} \delta_{a_s' a_u} U_{a_s' a_u n_l}^{(s)*} U_{a_s a_u n_l}^{(s)} \\ &= \sum_{a_u n_l} U_{a_s' a_u n_l}^{(s)*} U_{a_s a_u n_l}^{(s)} = \delta_{a_s' a_s}. \end{aligned} \quad (\text{A6})$$

The second-to-last equal sign is due to the inductive hypothesis, and the last one is from Eq. (16).

- (3) $U_{a,n_k a_v}^{(s)}$. This case is similar to (2).
 (4) $U_{a,a_u a_v}^{(s)}$. Assume $U^{(u)}$ has indices $U_{a,j_1}^{(u)}$, and $U^{(v)}$ has indices $U_{a_v,j_2}^{(v)}$, with $j_2 = j \setminus j_1$. Therefore,

$$\begin{aligned} \sum_j U_{a,j}^{(s)*} U_{a,j}^{(s)} &= \sum_{a'_u a_u} \sum_{a'_v a_v} \sum_{j_1 j_2} U_{a'_u j_1}^{(v)*} U_{a'_u j_1}^{(u)*} U_{a'_u a'_v}^{(s)*} \\ &\quad \times U_{a_u a_v}^{(s)} U_{a_u j_1}^{(u)} U_{a_v j_2}^{(v)} \\ &= \sum_{a'_u a_u} \sum_{a'_v a_v} \delta_{a'_u a_u} \delta_{a'_v a_v} U_{a'_u a'_v}^{(s)*} U_{a_u a_v}^{(s)} \\ &= \sum_{a_u a_v} U_{a'_u a'_v}^{(s)*} U_{a_u a_v}^{(s)} = \delta_{a'_u a'_v}. \end{aligned} \quad (A7)$$

The second-to-last equal sign is due to the inductive hypothesis, and the last one is from Eq. (16).

From (1)–(4), Eq. (A5) is proved for all s . $\square$

Similarly, the gauge condition can also be generalized to the branch as

$$\sum_j U_{a,j}^{(s)*} \left[\frac{d}{dt} U^{(s)} \right]_{a,j} = 0. \quad (A8)$$

To derive Eq. (19), we plug Eq. (A4) into the TDVP Eq. (17),

$$\delta(\mathcal{A}^{(s)} U^{(s)})^* \mathcal{L} \mathcal{A}^{(s)} U^{(s)} = \delta(\mathcal{A}^{(s)} U^{(s)})^* \frac{\partial}{\partial t} (\mathcal{A}^{(s)} U^{(s)}). \quad (A9)$$

That is,

$$\begin{aligned} \delta \mathcal{A}^{(s)*} U^{(s)*} \mathcal{L} \mathcal{A}^{(s)} U^{(s)} + \mathcal{A}^{(s)*} \delta U^{(s)*} \mathcal{L} \mathcal{A}^{(s)} U^{(s)} \\ = \delta \mathcal{A}^{(s)*} U^{(s)*} \frac{\partial}{\partial t} \mathcal{A}^{(s)} U^{(s)} + \mathcal{A}^{(s)*} \delta U^{(s)*} \frac{\partial}{\partial t} \mathcal{A}^{(s)} U^{(s)} \\ + \delta \mathcal{A}^{(s)*} U^{(s)*} \mathcal{A}^{(s)} \frac{\partial}{\partial t} U^{(s)} + \mathcal{A}^{(s)*} \delta U^{(s)*} \mathcal{A}^{(s)} \frac{\partial}{\partial t} U^{(s)}. \end{aligned} \quad (A10)$$

Since the variation of $\delta \mathcal{A}^{(s)*}$ and $\delta U^{(s)*}$ is independent and arbitrary, therefore,

$$\begin{aligned} \sum_{j'ij_a} U_{a,j'}^{(s)*} \mathcal{L}_{i'j',ij} U_{a,j}^{(s)} \mathcal{A}_{ia_s}^{(s)} \\ = \sum_{j_a} U_{a,j}^{(s)*} \frac{\partial}{\partial t} U_{a,j}^{(s)} \mathcal{A}_{i'a_s}^{(s)} + \sum_{j_a} U_{a,j}^{(s)*} U_{a,j}^{(s)} \frac{\partial}{\partial t} \mathcal{A}_{i'a_s}^{(s)}, \end{aligned} \quad (A11)$$

and

$$\begin{aligned} \sum_{i'ij_a} \mathcal{A}_{i'a_s}^{(s)*} \mathcal{L}_{i'j',ij} \mathcal{A}_{ia_s}^{(s)} U_{a,j}^{(s)} \\ = \sum_{ia_s} \mathcal{A}_{ia_s}^{(s)*} \frac{\partial}{\partial t} \mathcal{A}_{ia_s}^{(s)} U_{a,j}^{(s)} + \sum_{ia_s} \mathcal{A}_{ia_s}^{(s)*} \mathcal{A}_{ia_s}^{(s)} \frac{\partial}{\partial t} U_{a,j}^{(s)}. \end{aligned} \quad (A12)$$

From Eq. (A11) and the properties of the branch $U^{(s)}$ in Eqs. (A5) and (A8), we have

$$\frac{\partial}{\partial t} \mathcal{A}_{i'a_s}^{(s)} = \sum_{j'ij_a} U_{a,j'}^{(s)*} \mathcal{L}_{i'j',ij} U_{a,j}^{(s)} \mathcal{A}_{ia_s}^{(s)}, \quad (A13)$$

and when $s = 1$, it becomes

$$\frac{\partial}{\partial t} A_{i'j'a_1}^{(0)} = \sum_{ij_{a_1}} \sum_{\substack{n'_1 \dots n'_K \\ n_1 \dots n_K}} U_{a'_1 n'_1 \dots n'_K}^{(1)*} \mathcal{L}_{i'j',n'_1 \dots n'_K} U_{a_1 n_1 \dots n_K}^{(1)} A_{ij_{a_1}}^{(0)}. \quad (A14)$$

Furthermore, note that

$$\begin{aligned} \sum_{\substack{n'_1 \dots n'_K \\ n_1 \dots n_K}} U_{a'_1 n'_1 \dots n'_K}^{(1)*} \mathcal{L}_{i'j',n'_1 \dots n'_K} U_{a_1 n_1 \dots n_K}^{(1)} \\ = \sum_m [h_m^>]_{i'i} [h_m^<]_{j'j} [f_m^{(1)}]_{a'_1 a_1}. \end{aligned} \quad (A15)$$

Hence, we obtain Eq. (19).

On the other hand, plug Eq. (A13) into Eq. (A12), and then we obtain

$$\begin{aligned} \sum_{i'ij_a} \mathcal{A}_{i'a_s}^{(s)*} \mathcal{L}_{i'j',ij} \mathcal{A}_{ia_s}^{(s)} U_{a,j}^{(s)} \\ = \sum_{i''j''ij_a a'_s} \mathcal{A}_{ia'_s}^{(s)*} U_{a,j}^{(s)} \mathcal{L}_{ij,i''j''} U_{a'_s j''}^{(s)} \mathcal{A}_{i''a'_s}^{(s)} U_{a,j}^{(s)} \\ + \sum_{ia_s} \mathcal{A}_{ia_s}^{(s)*} \mathcal{A}_{ia_s}^{(s)} \frac{\partial}{\partial t} U_{a,j}^{(s)}. \end{aligned} \quad (A16)$$

To obtain the master equation of $U^{(s)}$, similar to the derivation in Eq. (A5), suppose $U^{(s)}$ looks like $U_{a,a_u a_v}^{(s)}$ in the TTN decomposition Eq. (15) [other cases $U_{a,n_k a_v}^{(s)}$, $U_{a,a_u n_l}^{(s)}$, and $U_{a,n_k n_l}^{(s)}$ are similar]; then the master equation of $U^{(s)}$ can be obtained by multiplying $U^{(u)*}$ and $U^{(v)*}$ on both sides of Eq. (A16). Suppose $U_{a_u j_1}^{(u)}$ and $U_{a_v j_2}^{(v)}$ with $j_2 = j \setminus j_1$, then

$$\begin{aligned} \sum_{i'j'ij_a} U_{a'_u j'_1}^{(v)*} U_{a'_u j'_1}^{(u)*} \mathcal{A}_{i'a'_s}^{(s)*} \mathcal{L}_{i'j',ij} \mathcal{A}_{ia_s}^{(s)} U_{a,j}^{(s)} \\ = \sum_m \sum_{a_s a_u a_v} [D_m^{(s)}]_{a_s a'_s} [f_m^{(u)}]_{a'_u a_u} [f_m^{(v)}]_{a'_v a_v} U_{a_s a_u a_v}^{(s)}, \end{aligned} \quad (A17)$$

$$\begin{aligned} \sum_{i''j''ij_a a'_s} U_{a'_v j'_2}^{(v)*} U_{a'_v j'_2}^{(u)*} \mathcal{A}_{ia'_s}^{(s)*} U_{a,j}^{(s)} \mathcal{L}_{ij,i''j''} U_{a'_s j''}^{(s)} \mathcal{A}_{i''a'_s}^{(s)} U_{a,j}^{(s)} \\ = \sum_m \sum_{a'_s a_s} [D_m^{(s)}]_{a'_s a'_s} U_{a_s a'_u a'_v}^{(s)} [f_m^{(s)}]_{a_s a'_s}, \end{aligned} \quad (A18)$$

and

$$\sum_{ij_a} U_{a_v j_2}^{(v)*} U_{a_u j_1}^{(u)*} \mathcal{A}_{ia'_s}^{(s)*} \mathcal{A}_{ia_s}^{(s)} \frac{\partial}{\partial t} U_{a,j}^{(s)} = \sum_{a_s} D_{a_s a'_s}^{(s)} \frac{d}{dt} U_{a_s a_u a_v}^{(s)}. \quad (A19)$$

Here, we have used the definition of $f_m^{(s)}$, $D_m^{(s)}$, and $D^{(s)}$ introduced in Eqs. (21)–(24) to simplify the equation. Plugging Eqs. (A17)–(A19) into Eq. (A16), we have

$$\begin{aligned} \sum_{a'_s} D_{a'_s a'_s}^{(s)} \frac{d}{dt} U_{a'_s a'_u a'_v}^{(s)} \\ = \sum_m \sum_{a'_s a_u a_v} [D_m^{(s)}]_{a'_s a'_s} [f_m^{(u)}]_{a'_u a_u} [f_m^{(v)}]_{a'_v a_v} U_{a'_s a_u a_v}^{(s)} \\ - \sum_m \sum_{a'_s a_s} [D_m^{(s)}]_{a'_s a'_s} U_{a_s a'_u a'_v}^{(s)} [f_m^{(s)}]_{a_s a'_s}. \end{aligned} \quad (A20)$$

Similarly, for $U_{a_s a_u n_l}^{(s)}$,

$$\begin{aligned} \sum_{a'_s} D_{a'_s a''}^{(s)} \frac{d}{dt} U_{a'_s a_u n_l}^{(s)} \\ = \sum_m \sum_{a'_s a_u n_l} [D_m^{(s)}]_{a'_s a''} [f_m^{(u)}]_{a'_u a_u} [h_m^{(l)}]_{n'_l n_l} U_{a'_s a_u n_l}^{(s)} \\ - \sum_m \sum_{a'_s a_s} [D_m^{(s)}]_{a'_s a''} U_{a'_s a_u n_l}^{(s)} [f_m^{(s)}]_{a_s a'_s}, \end{aligned} \quad (A21)$$

for $U_{a_s n_k a_v}^{(s)}$,

$$\begin{aligned} \sum_{a'_s} D_{a'_s a''}^{(s)} \frac{d}{dt} U_{a'_s n'_k a'_v}^{(s)} \\ = \sum_m \sum_{a'_s a_v} [D_m^{(s)}]_{a'_s a''} [h_m^{(k)}]_{n'_k a_u} [f_m^{(v)}]_{a'_v a_v} U_{a'_s n'_k a'_v}^{(s)} \\ - \sum_m \sum_{a'_s a_s} [D_m^{(s)}]_{a'_s a''} U_{a'_s n'_k a'_v}^{(s)} [f_m^{(s)}]_{a_s a'_s}, \end{aligned} \quad (A22)$$

and for $U_{a_s n_k n_l}^{(s)}$,

$$\begin{aligned} \sum_{a'_s} D_{a'_s a''}^{(s)} \frac{d}{dt} U_{a'_s n'_k n'_l}^{(s)} \\ = \sum_m \sum_{a'_s n_k n_l} [D_m^{(s)}]_{a'_s a''} [h_m^{(k)}]_{n'_k n_k} [h_m^{(l)}]_{n'_l n_l} U_{a'_s n_k n_l}^{(s)} \\ - \sum_m \sum_{a'_s a_s} [D_m^{(s)}]_{a'_s a''} U_{a'_s n'_k n'_l}^{(s)} [f_m^{(s)}]_{a_s a'_s}. \end{aligned} \quad (A23)$$

Hence, Eq. (20) is derived.

APPENDIX B: EXAMPLE OF TTN-HEOM FOR AN OPEN QUANTUM DYNAMICS WITH 4 BEXCITONS

To better demonstrate that the TTN scheme yields a closed set of master equations, here we consider an example of tensor tree decomposition for $\Omega(t)$ with $K = 4$ bexcitons,

$$\begin{aligned} \Omega_{ij n_1 n_2 n_3 n_4}(t) \\ = \sum_{a_1 a_2 a_3}^{R_1 R_2 R_3} A_{ij a_1}^{(0)}(t) U_{a_1 a_2 a_3}^{(1)}(t) U_{a_2 n_1 n_2}^{(2)}(t) U_{a_3 n_3 n_4}^{(3)}(t), \end{aligned} \quad (B1)$$

where $A^{(0)}$ is the root. Inserting Eq. (B1) into Eq. (17) and taking into account the gauge condition Eq. (18) yields

$$\frac{d}{dt} A_{i'j'a_1}^{(0)} = \sum_m \sum_{ij a_1} [h_m^{>}]_{i'i} [h_m^{<}]_{j'j} [f_m^{(1)}]_{a'_1 a_1} A_{ij a_1}^{(0)}, \quad (B2)$$

for the root tensor. The $f_m^{(s)}$ is defined as

$$[f_m^{(3)}]_{a'_3 a_3} \equiv \sum_{n'_3 n'_4 n_3 n_4} U_{a'_3 n'_3 n'_4}^{(4)*} [h_m^{(3)}]_{n'_3 n_3} [h_m^{(4)}]_{n'_4 n_4} U_{a_3 n_3 n_4}^{(4)}, \quad (B3)$$

$$[f_m^{(2)}]_{a'_2 a_2} \equiv \sum_{n'_1 n'_2 n_1 n_2} U_{a'_2 n'_1 n'_2}^{(3)*} [h_m^{(1)}]_{n'_1 n_1} [h_m^{(2)}]_{n'_2 n_2} U_{a_2 n_1 n_2}^{(3)}, \quad (B4)$$

$$[f_m^{(1)}]_{a'_1 a_1} \equiv \sum_{a'_3 a'_2 a_3 a_2} U_{a'_1 a'_2 a_3}^{(2)*} [f_m^{(3)}]_{a'_3 a_3} [f_m^{(2)}]_{a'_2 a_2} U_{a_1 a_2 a_3}^{(2)}, \quad (B5)$$

and capture the bexciton influence on the system. Note that $f_m^{(1)}$ depends on $f_m^{(2)}$ and $f_m^{(3)}$, while the latter depends on the dissipators $h_m^{(k)}$. These quantities effectively extract the relevant bath dynamics that influences the system in a compressed fashion. In turn, the semi-unitary tensors capture the active space of the bexcitons that influence the system's dynamics. The equations of motion for them are given by

$$\begin{aligned} \sum_{a'_1} [D^{(1)}]_{a'_1 a''} \frac{d}{dt} U_{a'_1 a'_2 a'_3}^{(2)} \\ = \sum_m \sum_{a'_1 a_1 a_2 a_3} [D_m^{(1)}]_{a'_1 a''} \\ \times ([f_m^{(2)}]_{a'_2 a_2} [f_m^{(3)}]_{a'_3 a_3} U_{a'_1 a_2 a_3}^{(2)} - U_{a_1 a'_2 a'_3}^{(2)} [f_m^{(1)}]_{a_1 a'_1}), \end{aligned} \quad (B6)$$

$$\begin{aligned} \sum_{a'_2} [D^{(2)}]_{a'_2 a''} \frac{d}{dt} U_{a'_2 n'_1 n'_2}^{(3)} \\ = \sum_m \sum_{a'_2 a_2 n_1 n_2} [D_m^{(2)}]_{a'_2 a''} \\ \times ([h_m^{(1)}]_{n'_1 n_1} [h_m^{(2)}]_{n'_2 n_2} U_{a'_2 n_1 n_2}^{(3)} - U_{a_2 n'_1 n'_2}^{(3)} [f_m^{(2)}]_{a_2 a'_2}), \end{aligned} \quad (B7)$$

$$\begin{aligned} \sum_{a'_3} [D^{(3)}]_{a'_3 a''} \frac{d}{dt} U_{a'_3 n'_3 n'_4}^{(4)} \\ = \sum_m \sum_{a'_3 a_3 n_3 n_4} [D_m^{(3)}]_{a'_3 a''} \\ \times ([h_m^{(3)}]_{n'_3 n_3} [h_m^{(4)}]_{n'_4 n_4} U_{a'_3 n_3 n_4}^{(4)} - U_{a_3 n'_3 n'_4}^{(4)} [f_m^{(3)}]_{a_3 a'_3}). \end{aligned} \quad (B8)$$

Here, the quantities $D_m^{(s)}$ and $D^{(s)}$ are defined by

$$[D^{(1)}]_{a'_1 a_1} \equiv \sum_{ij} A_{ij a'_1}^{(0)} A_{ij a_1}^{(1)*}, \quad (B9)$$

$$[D^{(2)}]_{a'_2 a_2} \equiv \sum_{a'_1 a_1 a_3} U_{a'_1 a'_2 a_3}^{(2)} [D^{(1)}]_{a'_1 a_1} U_{a_1 a_2 a_3}^{(2)*}, \quad (B10)$$

$$[D^{(3)}]_{a'_3 a_3} \equiv \sum_{a'_1 a_1 a_2} U_{a'_1 a'_2 a_3}^{(2)} [D^{(1)}]_{a'_1 a_1} U_{a_1 a_2 a_3}^{(2)*}, \quad (B11)$$

and

$$[D_m^{(1)}]_{a'_1 a_1} \equiv \sum_{i'j'ij} [h_m^{>}]_{i'i'} [h_m^{<}]_{j'j} A_{i'j'a'_1}^{(0)} A_{ij a_1}^{(1)*}, \quad (B12)$$

$$[D_m^{(2)}]_{a'_2 a_2} \equiv \sum_{a'_3 a_3 a_1 a_3} [f_m^{(3)}]_{a'_3 a_3} U_{a'_1 a'_2 a_3}^{(2)} [D_m^{(1)}]_{a'_1 a_1} U_{a_1 a_2 a_3}^{(2)*}, \quad (B13)$$

$$[D_m^{(3)}]_{a'_3 a_3} \equiv \sum_{a'_1 a'_2 a_1 a_2} [f_m^{(2)}]_{a'_2 a_2} U_{a'_1 a'_2 a_3}^{(2)} [D_m^{(1)}]_{a'_1 a_1} U_{a_1 a_2 a_3}^{(2)*}. \quad (B14)$$

REFERENCES

- ¹H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, 2007).
- ²I. de Vega and D. Alonso, *Rev. Mod. Phys.* **89**, 015001 (2017).
- ³H. Weimer, A. Kshetrimayum, and R. Orús, *Rev. Mod. Phys.* **93**, 015008 (2021).
- ⁴E. Mulvihill and E. Geva, *J. Phys. Chem. B* **125**, 9834 (2021).
- ⁵G. T. Landi, D. Poletti, and G. Schaller, *Rev. Mod. Phys.* **94**, 045006 (2022).
- ⁶S. Mukamel, *Principles of Nonlinear Optical Spectroscopy* (Oxford University Press, 1995).
- ⁷S. Biswas, J. Kim, X. Zhang, and G. D. Scholes, *Chem. Rev.* **122**, 4257 (2022).
- ⁸M. Shapiro and P. Brumer, *Quantum Control of Molecular Processes* (John Wiley & Sons, 2011).
- ⁹S. A. Rice and M. Zhao, *Optical Control of Molecular Dynamics* (John Wiley & Sons, 2000).
- ¹⁰C. Bäuerle, D. C. Glatzli, T. Meunier, F. Portier, P. Roche, P. Roulleau, S. Takada, and X. Waintal, *Rep. Prog. Phys.* **81**, 056503 (2018).
- ¹¹C. Heide, P. D. Keathley, and M. F. Kling, *Nat. Rev. Phys.* **6**, 648 (2024).
- ¹²V. May and O. Kühn, *Charge and Energy Transfer Dynamics in Molecular Systems* (John Wiley & Sons, 2011).
- ¹³Y.-C. Cheng and G. R. Fleming, *Annu. Rev. Phys. Chem.* **60**, 241 (2009).
- ¹⁴F. Ortmann, F. Bechstedt, and K. Hannebald, *Phys. Rev. B* **79**, 235206 (2009).
- ¹⁵M. Schlosshauer, *Decoherence: And the Quantum-To-Classical Transition* (Springer, 2007).
- ¹⁶M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2010).
- ¹⁷B. Gu and I. Franco, *J. Chem. Phys.* **149**, 174115 (2018).
- ¹⁸W. Hu, B. Gu, and I. Franco, *J. Chem. Phys.* **148**, 134304 (2018).
- ¹⁹C. W. Kim, J. M. Nichol, A. N. Jordan, and I. Franco, *PRX Quantum* **3**, 040308 (2022).
- ²⁰C. W. Kim and I. Franco, *J. Chem. Phys.* **160**, 214111 (2024).
- ²¹R. Korol, X. Chen, and I. Franco, *J. Phys. Chem. A* **129**, 3587 (2025).
- ²²J. Cao, R. J. Cogdell, D. F. Coker, H.-G. Duan, J. Hauer, U. Kleinekathöfer, T. L. C. Jansen, T. Mančal, R. J. D. Miller, J. P. Ogilvie, V. I. Prokhorenko, T. Renger, H.-S. Tan, R. Tempelaar, M. Thorwart, E. Thyrtahaug, S. Westenhoff, and D. Zigmantas, *Sci. Adv.* **6**, eaaz4888 (2020).
- ²³A. Chiesa, P. Santini, E. Garlatti, F. Luis, and S. Carretta, *Rep. Prog. Phys.* **87**, 034501 (2024).
- ²⁴E. Joos, H. D. Zeh, C. Kiefer, D. Giulini, J. Kupsch, and I.-O. Stamatescu, *Decoherence and the Appearance of a Classical World in Quantum Theory* (Springer, 2003).
- ²⁵N. Lyu, M. B. Soley, and V. S. Batista, *J. Chem. Theory Comput.* **18**, 3327 (2022).
- ²⁶H. Wang and H.-D. Meyer, *J. Phys. Chem. A* **125**, 3077 (2021).
- ²⁷W. Popp, D. Brey, R. Binder, and I. Burghardt, *Annu. Rev. Phys. Chem.* **72**, 591 (2021).
- ²⁸J. Ren, Z. Shuai, and G. K.-L. Chan, *J. Chem. Theory Comput.* **14**, 5027 (2018).
- ²⁹H.-D. Meyer, U. Manthe, and L. S. Cederbaum, *Chem. Phys. Lett.* **165**, 73 (1990).
- ³⁰A. G. Redfield, *IBM J. Res. Dev.* **1**, 19 (1957).
- ³¹M. O. Scully and W. E. Lamb, *Phys. Rev.* **159**, 208 (1967).
- ³²B. R. Mollow and M. M. Miller, *Ann. Phys.* **52**, 464 (1969).
- ³³Y. Tanimura and R. Kubo, *J. Phys. Soc. Jpn.* **58**, 101 (1989).
- ³⁴A. Ishizaki and G. R. Fleming, *J. Chem. Phys.* **130**, 234111 (2009).
- ³⁵T. Ikeda and G. D. Scholes, *J. Chem. Phys.* **152**, 204101 (2020).
- ³⁶N. Anto-Sztrikacs, A. Nazir, and D. Segal, *PRX Quantum* **4**, 020307 (2023).
- ³⁷Y. Tanimura, *J. Chem. Phys.* **153**, 020901 (2020).
- ³⁸Y. Yan, M. Xu, T. Li, and Q. Shi, *J. Chem. Phys.* **154**, 194104 (2021).
- ³⁹X. Chen and I. Franco, *J. Chem. Phys.* **160**, 204116 (2024).
- ⁴⁰G. Lindblad, *Commun. Math. Phys.* **48**, 119 (1976).
- ⁴¹A. Redfield, "The theory of relaxation processes," in *Advances in Magnetic Resonance*, Advances in Magnetic and Optical Resonance Vol. 1 (Academic Press, 1965), pp. 1–32.
- ⁴²M. Topaler and N. Makri, *Chem. Phys. Lett.* **210**, 285 (1993).
- ⁴³N. Makri and D. E. Makarov, *J. Chem. Phys.* **102**, 4600 (1995).
- ⁴⁴N. Makri and D. E. Makarov, *J. Chem. Phys.* **102**, 4611 (1995).
- ⁴⁵S. Kundu and N. Makri, *J. Chem. Phys.* **158**, 224801 (2023).
- ⁴⁶A. Bose, *J. Chem. Phys.* **158**, 204113 (2023).
- ⁴⁷L. P. Lindoy, A. Mandal, and D. R. Reichman, *Nat. Commun.* **14**, 2733 (2023).
- ⁴⁸Q. Shi, L. Chen, G. Nan, R.-X. Xu, and Y. Yan, *J. Chem. Phys.* **130**, 084105 (2009).
- ⁴⁹X. Dan, M. Xu, J. T. Stockburger, J. Ankerhold, and Q. Shi, *Phys. Rev. B* **107**, 195429 (2023).
- ⁵⁰B. Le Dé, A. Jaouadi, E. Mangaud, A. W. Chin, and M. Desouter-Lecomte, *J. Chem. Phys.* **160**, 244102 (2024).
- ⁵¹M. Xu, Y. Yan, Q. Shi, J. Ankerhold, and J. T. Stockburger, *Phys. Rev. Lett.* **129**, 230601 (2022).
- ⁵²Y. Nakatsukasa, O. Sète, and L. N. Trefethen, *SIAM J. Sci. Comput.* **40**, A1494 (2018).
- ⁵³H. Takahashi, S. Rudge, C. Kaspar, M. Thoss, and R. Borrelli, *J. Chem. Phys.* **160**, 204105 (2024).
- ⁵⁴M. Lednev, F. J. García-Vidal, and J. Feist, *Phys. Rev. Lett.* **132**, 106902 (2024).
- ⁵⁵N. Lambert, S. Ahmed, M. Cirio, and F. Nori, *Nat. Commun.* **10**, 3721 (2019).
- ⁵⁶D. Potts and M. Tasche, *Linear Algebra Appl.* **439**, 1024 (2013).
- ⁵⁷Z.-H. Chen, Y. Wang, X. Zheng, R.-X. Xu, and Y. Yan, *J. Chem. Phys.* **156**, 221102 (2022).
- ⁵⁸C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, *Nature* **585**, 357 (2020).
- ⁵⁹J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. K. Luk, B. Maher, Y. Pan, C. Puhersch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, S. Zhang, M. Suo, P. Tillet, X. Zhao, E. Wang, K. Zhou, R. Zou, X. Wang, A. Mathews, W. Wen, G. Chanan, P. Wu, and S. Chintala, in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (ACM, 2024), Vol. 2, pp. 929–947.
- ⁶⁰H. Wang and M. Thoss, *J. Chem. Phys.* **119**, 1289 (2003).
- ⁶¹U. Schollwöck, *Ann. Phys.* **326**, 96 (2011).
- ⁶²M. A. Cazalilla and J. B. Marston, *Phys. Rev. Lett.* **88**, 256403 (2002).
- ⁶³H. G. Luo, T. Xiang, and X. Q. Wang, *Phys. Rev. Lett.* **91**, 049701 (2003).
- ⁶⁴A. E. Feiguin and S. R. White, *Phys. Rev. B* **72**, 020404 (2005).
- ⁶⁵J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pižorn, H. Verschelde, and F. Verstraete, *Phys. Rev. Lett.* **107**, 070601 (2011).
- ⁶⁶S. Keller, M. Dolfi, M. Troyer, and M. Reiher, *J. Chem. Phys.* **143**, 244118 (2015).
- ⁶⁷S. M. Greene and V. S. Batista, *J. Chem. Theory Comput.* **13**, 4034 (2017).
- ⁶⁸M. F. X. Dorfner, D. Brey, I. Burghardt, and F. Ortmann, *J. Chem. Theory Comput.* **20**, 8767 (2024).
- ⁶⁹Ö. Legeza, T. Rohwedder, R. Schneider, and S. Szalay, "Tensor product approximation (DMRG) and coupled cluster method in quantum chemistry," in *Many-Electron Approaches in Physics, Chemistry and Mathematics: A Multidisciplinary View*, edited by V. Bach and L. Delle Site (Springer, 2014), pp. 53–76.
- ⁷⁰V. Murg, F. Verstraete, R. Schneider, P. R. Nagy, and Ö. Legeza, *J. Chem. Theory Comput.* **11**, 1027 (2015).
- ⁷¹A. H. Werner, D. Jaschke, P. Silvi, M. Kliesch, T. Calarco, J. Eisert, and S. Montangero, *Phys. Rev. Lett.* **116**, 237201 (2016).
- ⁷²A. D. Somoza, O. Marty, J. Lim, S. F. Huelga, and M. B. Plenio, *Phys. Rev. Lett.* **123**, 100502 (2019).
- ⁷³W. Li, J. Ren, and Z. Shuai, *J. Phys. Chem. Lett.* **11**, 4930 (2020).
- ⁷⁴D. Tamascelli, A. Smirne, J. Lim, S. F. Huelga, and M. B. Plenio, *Phys. Rev. Lett.* **123**, 090402 (2019).
- ⁷⁵A. Strathearn, P. Kirton, D. Kilda, J. Keeling, and B. W. Lovett, *Nat. Commun.* **9**, 3322 (2018).
- ⁷⁶A. Bose and P. L. Walters, *J. Chem. Phys.* **156**, 024101 (2022).
- ⁷⁷S. Kundu and N. Makri, *J. Phys. Chem. Lett.* **11**, 8783 (2020).
- ⁷⁸Q. Shi, Y. Xu, Y. Yan, and M. Xu, *J. Chem. Phys.* **148**, 174102 (2018).

- ⁷⁹R. Borrelli, *J. Chem. Phys.* **150**, 234102 (2019).
- ⁸⁰R. Borrelli and S. Dolgov, *J. Phys. Chem. B* **125**, 5397 (2021).
- ⁸¹T. Li, Y. Yan, and Q. Shi, *J. Chem. Phys.* **156**, 064107 (2022).
- ⁸²Y. Ke, R. Borrelli, and M. Thoss, *J. Chem. Phys.* **156**, 194102 (2022).
- ⁸³E. Mangaud, A. Jaouadi, A. Chin, and M. Desouter-Lecomte, *Eur. Phys. J.: Spec. Top.* **232**, 1847 (2023).
- ⁸⁴Y. Ke, *J. Chem. Phys.* **158**, 211102 (2023).
- ⁸⁵A. Raab, *Chem. Phys. Lett.* **319**, 674 (2000).
- ⁸⁶J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, and F. Verstraete, *Phys. Rev. B* **94**, 165116 (2016).
- ⁸⁷B. Kloss, I. Burghardt, and C. Lubich, *J. Chem. Phys.* **146**, 174107 (2017).
- ⁸⁸C. Lubich, I. V. Oseledets, and B. Vandereycken, *SIAM J. Numer. Anal.* **53**, 917 (2015).
- ⁸⁹C. Lubich, B. Vandereycken, and H. Walach, *SIAM J. Numer. Anal.* **56**, 1273 (2018).
- ⁹⁰H. Wang and H.-D. Meyer, *J. Chem. Phys.* **149**, 044119 (2018).
- ⁹¹H.-D. Meyer and H. Wang, *J. Chem. Phys.* **148**, 124105 (2018).
- ⁹²I. Gustin, C. W. Kim, D. W. McCamant, and I. Franco, *Proc. Natl. Acad. Sci. U. S. A.* **120**, e2309987120 (2023).
- ⁹³N. Lorenzoni, N. Cho, J. Lim, D. Tamascelli, S. F. Huelga, and M. B. Plenio, *Phys. Rev. Lett.* **132**, 100403 (2024).
- ⁹⁴J. R. Dormand and P. J. Prince, *J. Comput. Appl. Math.* **6**, 19 (1980).
- ⁹⁵T. Takagahara, E. Hanamura, and R. Kubo, *J. Phys. Soc. Jpn.* **43**, 802 (1977).
- ⁹⁶H. B. Callen and T. A. Welton, *Phys. Rev.* **83**, 34 (1951).
- ⁹⁷A. Ishizaki and Y. Tanimura, *J. Phys. Soc. Jpn.* **74**, 3131 (2005).
- ⁹⁸J. Hu, R.-X. Xu, and Y. Yan, *J. Chem. Phys.* **133**, 101106 (2010).
- ⁹⁹H. Liu, L. Zhu, S. Bai, and Q. Shi, *J. Chem. Phys.* **140**, 134106 (2014).
- ¹⁰⁰L. Grasedyck, *SIAM J. Matrix Anal. Appl.* **31**, 2029 (2010).
- ¹⁰¹L. Grasedyck and W. Hackbusch, *Comput. Methods Appl. Math.* **11**, 291 (2011).
- ¹⁰²R. Penrose, *Math. Proc. Camb. Philos. Soc.* **51**, 406 (1955).
- ¹⁰³L. P. Lindoy, B. Kloss, and D. R. Reichman, *J. Chem. Phys.* **155**, 174108 (2021).
- ¹⁰⁴L. P. Lindoy, B. Kloss, and D. R. Reichman, *J. Chem. Phys.* **155**, 174109 (2021).
- ¹⁰⁵R. Tarjan, *SIAM J. Comput.* **1**, 146 (1972).
- ¹⁰⁶A. Gómez Pueyo, M. A. L. Marques, A. Rubio, and A. Castro, *J. Chem. Theory Comput.* **14**, 3040 (2018).
- ¹⁰⁷C. Leforestier, R. H. Bisseling, C. Cerjan, M. D. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karrlein, H.-D. Meyer, N. Lipkin, O. Roncero, and R. Kosloff, *J. Comput. Phys.* **94**, 59 (1991).
- ¹⁰⁸R. T. Q. Chen (2018). “torchdiffq,” GitHub. <https://github.com/rtqichen/torchdiffq>
- ¹⁰⁹X. Chen and I. Franco (2025). “TENSO,” GitHub. <https://github.com/ifgroup/pytenso>
- ¹¹⁰W. Guan, P. Bao, J. Peng, Z. Lan, and Q. Shi, *J. Chem. Phys.* **161**, 122501 (2024).
- ¹¹¹L. P. Lindoy, D. Rodrigo-Albert, Y. Rath, and I. Rungger, “pyTTN: An open source toolbox for open and closed system quantum dynamics simulations using tree tensor networks,” *arXiv:2503.15460* (2025).
- ¹¹²B. Gu and I. Franco, *J. Phys. Chem. Lett.* **8**, 4289 (2017).
- ¹¹³B. Gu and I. Franco, *J. Phys. Chem. Lett.* **9**, 773 (2018).
- ¹¹⁴W. Li, J. Ren, H. Yang, H. Wang, and Z. Shuai, *J. Chem. Phys.* **161**, 054116 (2024).
- ¹¹⁵H. Çakır, R. M. Milbradt, and C. B. Mendl, “Optimal symbolic construction of matrix product operators and tree tensor network operators,” *arXiv:2502.18630* (2025).
- ¹¹⁶A. J. Dunnett and A. W. Chin, *Phys. Rev. B* **104**, 214302 (2021).
- ¹¹⁷H.-J. Liao, J.-G. Liu, L. Wang, and T. Xiang, *Phys. Rev. X* **9**, 031041 (2019).