

OQuPy: A Python package to efficiently simulate non-Markovian open quantum systems with process tensors

Gerald E. Fux^{1, a)} Piper Fowler-Wright^{2, b)} Joel Beckles² Eoin P. Butler^{3, 4} Paul R. Eastham^{3, 4}
 Dominic Gribben⁵ Jonathan Keeling² Dainius Kilda⁶ Peter Kirton⁷ Ewen D.C. Lawrence⁷ Brendon
 W. Lovett² Eoin O'Neill^{3, 4} Aidan Strathearn⁸ and Roosmarijn de Wit²

¹⁾*The Abdus Salam International Center for Theoretical Physics (ICTP), Strada Costiera 11, 34151 Trieste, Italy*

²⁾*SUPA, School of Physics and Astronomy, University of St Andrews, St Andrews, KY16 9SS, United Kingdom*

³⁾*School of Physics, Trinity College Dublin, Dublin 2, Ireland*

⁴⁾*Trinity Quantum Alliance, Unit 16, Trinity Technology and Enterprise Centre, Pearse Street, Dublin 2, Ireland*

⁵⁾*Institute for Physics, Johannes Gutenberg University of Mainz, D-55099 Mainz, Germany*

⁶⁾*Max-Planck-Institut für Quantenoptik, Hans-Kopfermann-Strasse 1, D-85748 Garching, Germany*

⁷⁾*Department of Physics and SUPA, University of Strathclyde, Glasgow, G4 0NG, United Kingdom*

⁸⁾*Institute for Photonics and Advanced Sensing (IPAS) and School of Physical Sciences, University of Adelaide, Adelaide, South Australia 5005, Australia*

(Dated: 28 August 2024)

Non-Markovian dynamics arising from the strong coupling of a system to a structured environment is essential in many applications of quantum mechanics and emerging technologies. Deriving an accurate description of general quantum dynamics including memory effects is however a demanding task, prohibitive to standard analytical or direct numerical approaches. We present a major release of our open source software package, OQuPy (Open Quantum System in Python), which provides several recently developed numerical methods that address this challenging task. It utilizes the process tensor approach to open quantum systems in which a single map, the process tensor, captures all possible effects of an environment on the system. The representation of the process tensor in a tensor network form allows an exact yet highly efficient description of non-Markovian open quantum systems (NM-OQS). The OQuPy package provides methods to (1) compute the dynamics and multi-time correlations of quantum systems coupled to single and multiple environments, (2) optimize control protocols for NM-OQS, (3) simulate interacting chains of NM-OQS, and (4) compute the mean-field dynamics of an ensemble of NM-OQS coupled to a common central system. Our aim is to provide an easily accessible and extensible tool for researchers of open quantum systems in fields such as quantum chemistry, quantum sensing, and quantum information.

I. INTRODUCTION

The theory of open quantum systems is one of the key ingredients for making quantum mechanics more applicable for experimental applications. It describes the dynamics of microscopic systems governed by the rules of quantum mechanics, but unlike the theory of closed systems, open systems additionally incorporate the unavoidable interactions with the environment. Many physical scenarios allow for simple time-local effective equations of motion^{1,2}. Such a description is called *Markovian*, which means that the future evolution of the system only depends on its present state, and not on its history. This is typically a good approximate description when the environment is featureless or the interaction between the system and environment is weak. However, there are

numerous applications where such a Markovian description fails. Solid state based quantum devices, such as quantum dots and nitrogen-vacancy color centers in diamonds, for example, often interact strongly with the lattice vibrational modes, surrounding charges, or magnetic spins³. In many such cases a non-Markovian treatment is necessary^{3–8}. Non-Markovian descriptions of open quantum systems are also an important ingredient for the development of better light-harvesting devices⁹ and understanding the role of quantum mechanics in biological systems¹⁰, such as the Fenna-Matthews-Olson complex^{11,12}. Apart from these particular applications, the general theory of non-Markovian open quantum systems is also of importance for fundamental research, such as the study of strong-coupling quantum thermodynamics^{13–19} and the development of theoretical tools for the characterization of quantum devices²⁰.

In recent years much progress has been made in the development of numerical methods to treat non-Markovian open quantum systems. A particularly versatile class of recently developed methods^{26–30} is based on an opera-

^{a)}Electronic mail: gerald.e.fux@gmail.com

^{b)}Electronic mail: piperfw@gmail.com

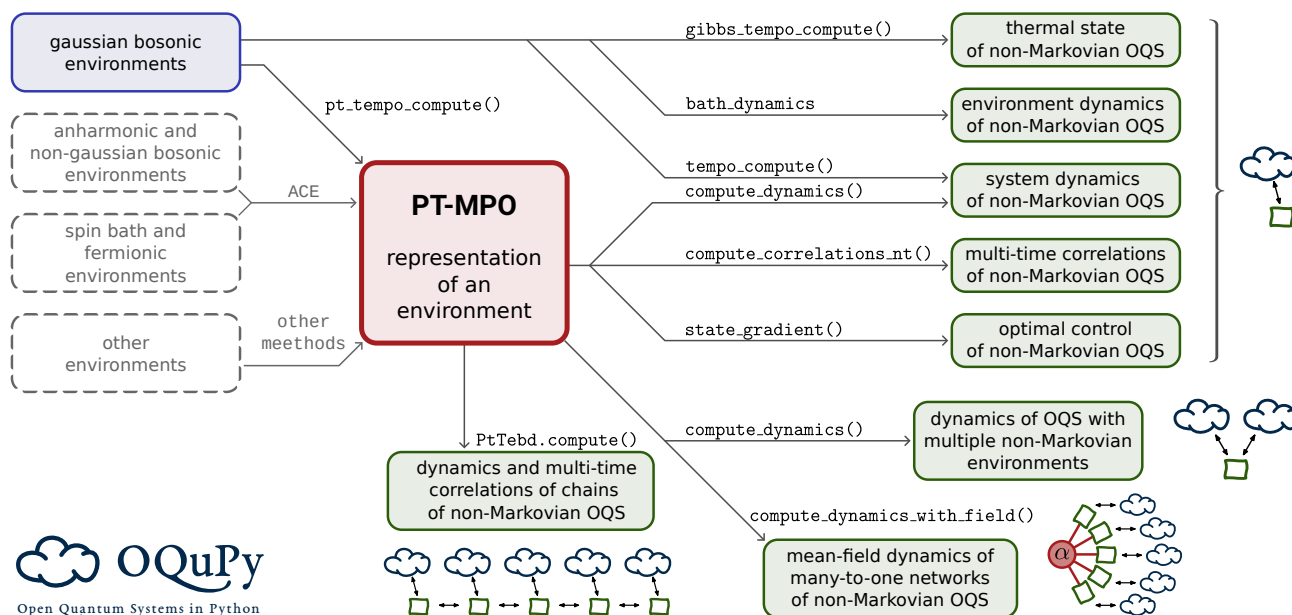


FIG. 1. Schematic of the workflow for various types of use cases in OQuPy. Each green box corresponds to a particular type of scenario of interest, most of which require a PT-MPO representation of the relevant environment (red box). The construction of PT-MPOs is available in OQuPy for Gaussian bosonic environments (blue box). One can import PT-MPOs created with other methods external to OQuPy (gray dashed boxes), such as the automated compression of environments (ACE) method²¹. For each type of use case scenario we display a sketch of the interactions among the open quantum systems (green squares), the environments (blue clouds), and possibly a common central system (red circle).

tional approach to open quantum systems, known as the *process tensor framework*^{31,32}, which is well suited for the challenges of non-Markovianity. Related to this approach are several other recently developed tensor network methods^{22–25} that build on the Quasi Adiabatic Path Integral (QUAPI)^{33–35}. As is clear from the other papers in this special issue, there are now a wide range of open source libraries that allow one to study quantum dynamics. Our particular focus is on methods centred around the process tensor. While libraries such as PathSum³⁵ and QuantumDynamics.jl³⁶ use tensor network approaches, they do not have the same focus on the process tensor. Other approaches to calculating the process tensor can however be found in the ACE^{21,37} library—as discussed in³⁷, that library broadens the range of environments for which process tensors can be calculated, but in the case of Gaussian baths, the methods we discuss here are generally more efficient. Moreover, as discussed further below, process tensors can in principle be calculated in another library such as ACE and then used within OQuPy.

The purpose of the OQuPy Python package is to give easy access to several of the recently developed methods^{22–30} that combine the conceptional advantages of the process tensor framework, with the numerical efficiency of tensor networks. In this article we give an overview of the toolset available in OQuPy. This includes numerically exact methods (i.e. employing only numerically

well controlled approximations) for the non-Markovian dynamics and multi-time correlations of

- quantum systems coupled to a single environment^{22,23,38},
- quantum systems coupled to multiple environments²⁷,
- interacting chains of non-Markovian open quantum systems²⁸, and
- ensembles of open many-body systems with many-to-one coupling to some common central system²⁹.

Furthermore, OQuPy implements methods to

- optimize control protocols for non-Markovian open quantum systems^{26,30},
- compute the dynamics of a non-Markovian environment²⁴, and
- obtain the thermal state of a quantum system strongly coupled to a structured environment²⁵.

At the heart of most of these methods lies the numerically efficient representation of the involved environment through a so-called *Process Tensor in Matrix Product Operator form* (PT-MPO), which we will introduce in Section II below. In short, the PT-MPO representation of an environment allows the systematical removal of negligible correlations between the system and environment, leading to a numerically efficient characterization of their

interaction. The overall workflow for most of the methods in OQuPy is thus separated into two stages: first, the computation of the PT-MPO representation for the environment of interest, and second, the application of the PT-MPO in one of the scenarios listed above. We sketch this workflow diagrammatically in Fig 1.

Outline.—In this article we give an overview of the underlying methodology and practical application of the OQuPy package. We begin in Section II by introducing the process tensor approach to open quantum systems. In particular, we present the construction of PT-MPOs using a *process tensor adoption* of the so-called *time evolving matrix product operator* method (PT-TEMPO) and comment on its computational parameters in Section II A. In Section III we give a brief introduction to the various methods built around PT-MPOs and showcase their application in the OQuPy package. This includes methods for the dynamics of non-Markovian open quantum systems in Section III A, multi-time correlations of non-Markovian open quantum systems in Section III B, optimization of control protocols in Section III C, and the dynamics of open quantum systems with multiple environments in Section III D. Also, we briefly introduce methods to address many-body open quantum systems such as chains of open quantum systems in Section III E, and many-to-one networks of open quantum systems in Section III F. Section IV discusses other related methods that are also included in OQuPy, such as a method for the computation of the environment dynamics (Section IV A), an alternative method for the system dynamics (Section IV B), and a method to directly compute the thermal state of a non-Markovian open quantum system (Section IV C). Finally we conclude with a brief discussion on possible future directions for further development of the OQuPy package in Section V.

II. PROCESS TENSOR FRAMEWORK

The process tensor framework introduced by Pollock *et al.*³¹ is an operational approach to fully characterize general open quantum systems and differs fundamentally from the canonical approach based on dynamical maps and their master equations. The dynamical map describes the evolution of the reduced system state for each initial state and hence, from an operational point of view, corresponds to prepare–evolve–measure type experiments. Given that an experiment can only access the system, but has no direct control over the environment, one may be led to believe that such a collection of dynamical maps gives a full description of the operationally accessible quantities of the open quantum system. This is, however, not true. Not all properties of an open system can be determined in a prepare–evolve–measure type experiment. In many experiments the observed quantities are related to multi-time correlations, or, one is interested in measuring the state of the system after multiple interactions with additional ancillary

systems. In fact, many scenarios of interest, such as the optimization of control protocols of an open quantum system, or the interaction among multiple open quantum systems, necessarily require the correct treatment of multi-time correlations. However, the dynamical map—even if known exactly from an exact master equation—cannot encode the necessary information to correctly describe multi-time correlations³².

The process tensor, on the other hand, not only encodes the outcome of prepare–evolve–measure type experiments, but also encodes the outcome of all possible experiments with intermediate interventions on the system, and thus also multi-time correlations. It is closely related to quantum combs³⁹, generalized influence functionals³⁸, and process matrices⁴⁰. The process tensor stands in a one-to-one relation to the outcome of all possible experiments one can perform on an open quantum system and is hence an operationally well defined and complete characterization of general open quantum systems. While the dynamical map is the map from all possible initial system states to the resulting final system state, the process tensor is the map from all possible control sequences to the final system state. Consider, for example, an open quantum system where one prepares the system in a certain state at time t_0 , applies a projective measurement in the X basis at time t_1 , and another projective measurement in the Z basis at time t_2 . In such a case the dynamical map correctly encodes the expectation value of the first measurement (i.e. the X measurement). However, the expectation value of the Z measurement, as well as the correlations between the two measurement outcomes will generally not be correctly encoded in the dynamical map, even if the dynamical map is known exactly. The process tensor, on the other hand, is the map that gives the state of the system at the final time t_2 for any sequence of control operations at time t_0 and t_1 , which in the above example are the state preparation and the X measurement.

Let us now define control operations and the process tensor a little more formally (see Ref. 31 for a fully formal definition). A control operation in this context is any physical operation on the system, such as the preparation of the system state, a measurement of the system, the application of a unitary, or the application of a unitary on the system and some ancilla system. Any such physical control operation can be described by a superoperator (i.e. a linear operator that acts on density matrices) that has the mathematical property of complete positivity (called a CP-map)⁴¹. A control sequence is hence a sequence of CP-maps $\{\mathcal{A}_n\}_{n=0\dots N-1}$ applied to the system at times $\{t_n\}_{n=0\dots N-1}$. Let us write the overall dynamics of the system and environment between time t and t' as an overall unitary operator $\hat{U}(t', t)$, and let us consider an overall initial state ρ_0 , which generally need not be separable between system and environment. When we include the control sequence $\{\mathcal{A}_n\}_{n=0\dots N-1}$ at the times $\{t_n\}_{n=0\dots N-1}$ then the final reduced density

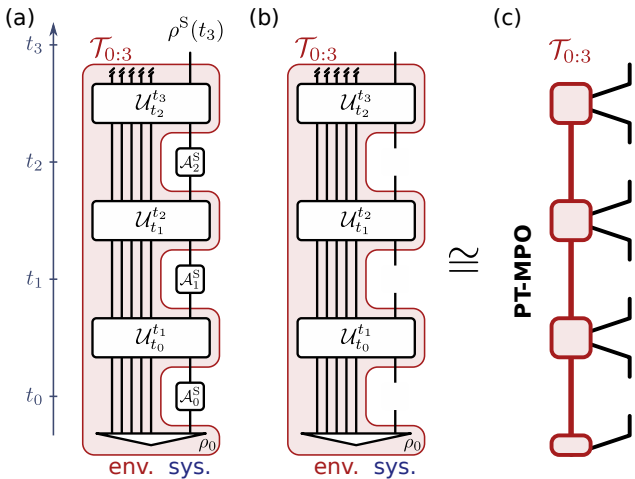


FIG. 2. (a) Quantum circuit for the state at t_3 of an open quantum system (sys.) coupled to an environment (env.) with control operations $\mathcal{A}_n$ at times t_0, t_1 , and t_2 . (b) The process tensor $\mathcal{T}_{0:3}$ of an open quantum system for times $t_0, \dots, t_3$. (c) The process tensor in matrix product operator form (PT-MPO).

matrix of the system at time t_N is

$$\rho^S(t_N) = \text{Tr}_E \left[\mathcal{U}_{t_{N-1}}^{t_N} \mathcal{A}_{N-1} \dots \mathcal{U}_{t_1}^{t_2} \mathcal{A}_1 \mathcal{U}_{t_0}^{t_1} \mathcal{A}_0 \rho_0 \right], \quad (1)$$

where $\mathcal{U}_t' \rho := \hat{U}(t', t) \rho \hat{U}^\dagger(t', t)$ is the superoperator of the unitary evolution, and Tr_E denotes the partial trace over the environment. The process tensor $\mathcal{T}_{0:N}$ for times $\{t_n\}_{n=0 \dots N}$ is then defined to be the map from such sequences of CP-maps to the corresponding final state of the system, i.e.

$$\rho^S(t_N) =: \mathcal{T}_{0:N} [\mathcal{A}_{N-1}, \dots, \mathcal{A}_1, \mathcal{A}_0]. \quad (2)$$

Note that the discrete number N of control operations does not imply that the evolution was discrete in time. It only means that we (for simplicity) consider a discrete number of interventions on the system. The dynamical map, for example, can be obtained from a process tensor with $N = 1$, because dynamical maps only describe the outcome of an experiment at *one* point in time after an initial system preparation (even if that one point in time is variable).

In Fig. 2 we show the construction of the process tensor for $N = 3$. It provides the final state of a generic open quantum system at time t_3 given control operations at times t_0, t_1 , and t_2 . The environment can be of any dimension and is here represented with five lines. This figure can be read as a quantum circuit or, equivalently, as a tensor network in Liouville space^{42,43}. In Fig. 2(a) the control sequence $[\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2]$ is inserted, yielding the final system state at time t_3 . Removing the control sequence and keeping the slots open in Fig. 2(b) thus leaves the tensor marked by the red shaded region, which can then be identified with the process tensor $\mathcal{T}_{0:3}$.

Note that the environment Liouville space does not contribute to the dimensionality of the tensor, for which instead only the external legs (corresponding to the system dimension) count.

Process tensor in matrix product operator form (PT-MPO).

The process tensor is a tensor of dimension d^{4N+2} , where d is the Hilbert space dimension of the system. That is, it grows exponentially with the number of possible interventions N . In many cases, however, it is possible to systematically discard negligible correlations and express the process tensor as a tensor network^{42,43} in a matrix product operator form^{43,44} (PT-MPO) as shown in Fig. 2(c). Here, the single high-rank process tensor is represented as an array of rank-4 tensors connected through indices that are called *bonds*. The necessary bond dimension of a PT-MPO reflects the degree of non-Markovianity in the interaction⁴⁵, but does not necessarily scale with the correlation time or the dimension of the environment Hilbert space. We note that the representation of process tensors in an MPO form is of practical use only when the system Hilbert space dimension d is small, which is why most examples presented below consider two-level systems. However, all methods discussed in this article are equally applicable to few-level systems, albeit generally requiring additional computational resources.

Various methods exist for the construction of PT-MPOs for different types of environments^{21,38,46–58}. The most straightforward approach is to start with a tensor network that explicitly represents the evolution of the environment and then perform a contraction sequence to yield a PT-MPO^{46,47,49,50}. For linearly coupled Gaussian environments one can even directly construct a tensor network that yields a PT-MPO without the need to explicitly model the environment as a tensor network^{38,48,52–56}. Other approaches^{21,51} allow the construction of PT-MPOs for any environment that can be approximated by a finite set of independent degrees of freedom.

All of these methods yield a PT-MPO that fully encodes the influence of an environment on the open system and can then be equally used in various ways (as shown in Fig. 1). As of version 0.5, the OQuPy package implements a single method for the creation of PT-MPOs, namely PT-TEMPO (the process tensor adoption of the time evolving matrix product operator method). This method allows the computation of PT-MPOs of open quantum systems coupled to Gaussian bosonic environments, such as photon and phonon fields. We stress, however, that for all applications of PT-MPOs presented in Section III below the origin of the PT-MPO or the method with which it has been created are irrelevant. PT-MPOs can be created for other types of environments with other methods and other software, and can then be imported into OQuPy and applied in the exact same way.

as PT-MPOs constructed with PT-TEMPO.

A. Construction of PT-MPOs with PT-TEMPO

The time evolving matrix product operator (TEMPO) method is a tensor network method for simulating the reduced dynamics of open quantum systems with Gaussian bosonic environments^{23,48}. In Ref. 38 Jørgensen and Pollock suggest a modification of the TEMPO tensor network contraction scheme to obtain a PT-MPO. In the following we outline the construction of the TEMPO tensor network and the adopted contraction sequence.

The most general form of the total Hamiltonian we consider for PT-TEMPO is (in units of $\hbar = 1$)

$$\hat{H}(t) = \hat{H}_S(t) + \underbrace{\hat{S} \sum_k \left(g_k \hat{b}_k + g_k^* \hat{b}_k^\dagger \right)}_{\hat{H}_I} + \underbrace{\sum_k \omega_k \hat{b}_k^\dagger \hat{b}_k}_{\hat{H}_E}, \quad (3)$$

where $\hat{H}_S(t)$ is an arbitrary (possibly time dependent) system Hamiltonian, $\hat{S}$ is the system operator that couples to the field, and $\hat{b}_k$ ($\hat{b}_k^\dagger$) are bosonic creation (annihilation) environment operators for the k^{th} mode. Also, we assume that the total initial system and environment state is separable, i.e. $\rho_0 = \rho_0^S \otimes \rho_0^E$, where the environment is a Gaussian (e.g. thermal) state with respect to $\hat{H}_E$. The environment interaction is fully characterized by the system coupling operator $\hat{S}$ and the spectral density $J(\omega)$, which is defined as

$$J(\omega) = \sum_k |g_k|^2 \delta(\omega - \omega_k). \quad (4)$$

In Section III and IV we will often consider $J(\omega)$ to take the power-law form⁵⁹

$$J(\omega) = \alpha \omega^\zeta \omega_c^{1-\zeta} \exp\left(-\frac{\omega}{\omega_c}\right), \quad (5)$$

where α is a dimensionless coupling constant, ζ is the power-law exponent, and ω_c is a cutoff frequency. For $\zeta = 1$ the environment is called “Ohmic”, for $\zeta < 1$ and $\zeta > 1$ it is said to be “sub-Ohmic” and “super-Ohmic”, respectively.

The derivation of the TEMPO tensor network²³ starts with a symmetrized second order Trotter splitting between system and environment interaction of the evolution into N small time steps ($\delta t = t_N/N$). Tracing out the environment degrees of freedom leads to a discretized version of the Feynman-Vernon influence functional⁶⁰. The resulting calculation can be represented as a tensor network of the form shown in Fig 3(a). The tensors b_k are constructed from the coupling operator $\hat{S}$ and the environment’s auto-correlation function. For thermal environments the auto-correlation function is determined by the spectral density $J(\omega)$ and the initial bath temperature T alone. These tensors quantify how the system

evolution at a time step t_n is influenced through the environment by the system’s history from time step t_{n-k} . The $\mathcal{V}(n)$ and $\mathcal{V}'(n)$ superoperators are the pure system propagators for the first and second half of the n^{th} time step. In many cases the two-time auto-correlation function of the environment interaction almost vanishes after some time τ_{cut} . In such a case the influence tensors b_k become identities for $k\delta t > \tau_{\text{cut}}$ and can be omitted from the tensor network.

To see how this tensor network links to the process tensor we include additional interventions $\mathcal{A}_n$ to the picture, which we draw as blue squares in Fig. 3(a). The tensor network in Fig. 3(a) now represents the final state $\rho^S(t_N)$ for any intervention sequence $\{\mathcal{A}_n\}_{n=0\dots N-1}$. We know from Ref. 31 that the process tensor is the unique multi-linear map with this property. We can thus conclude that the tensor network within the red shaded area in Fig. 3(a) (excluding the interventions $\mathcal{A}_0, \mathcal{A}_1, \dots$) is a representation of the desired process tensor $\mathcal{T}_{0:N}$.

As we will see in section III, it is often advantageous to consider the slightly smaller part of this tensor network shown as the dotted area in Figs. 3, which excludes the initial state ρ_0^S and the system propagators $\mathcal{V}_n$ and $\mathcal{V}'_n$. This corresponds to the process tensor $\tilde{\mathcal{T}}_{0:N}$ for an initially uncorrelated state and a total Hamiltonian $\tilde{H} = \tilde{H}_I + \tilde{H}_E$, without any contribution from the system Hamiltonian. From this perspective the final state is then

$$\rho^S(t_N) = \mathcal{T}_{0:N}[\mathcal{A}_{N-1}, \dots, \mathcal{A}_1, \mathcal{A}_0] \quad (6)$$

$$= \mathcal{V}'_{N-1} \tilde{\mathcal{T}}_{0:N}[\tilde{\mathcal{A}}_{N-1}, \dots, \tilde{\mathcal{A}}_1, \tilde{\mathcal{A}}_0], \quad (7)$$

with $\tilde{\mathcal{A}}_n = \mathcal{V}_n \circ \mathcal{A}_n \circ \mathcal{V}'_{n-1}$ and $\tilde{\mathcal{A}}_0 : \rho^S \mapsto \mathcal{A}_0[\mathcal{V}_0(\rho_0^S)]$. This means that the evolution due to the system part of the Hamiltonian can be realized as a discrete set of interventions when the time steps δt are chosen small enough such that the higher order terms of a second-order Trotterization can be neglected.

To compute $\tilde{\mathcal{T}}_{0:N}$ from the tensor network, Jørgensen and Pollock suggest to contract the network column by column with appropriate singular value decomposition (SVD) sweeps during the process³⁸. The computation time of this algorithm scales as $O(N \chi^3 d^6 K_{\text{max}})$, where χ is the bond dimension of the PT-MPO, d is the Hilbert space dimension of the system, and $K_{\text{max}} = \tau_{\text{cut}}/\delta t$.

One improvement that can be made to this tensor network is implementing a so-called *degeneracy checking* on the internal legs of b_k tensors^{23,61,62}. This is where identical rows of the b_k tensors are grouped together, reducing the bond dimensions of the internal legs of the tensor network. The east-west legs can be grouped together by equal energy differences in the coupling operator eigenvalues, as only the energy difference affects the result of the influence functional. The north-south legs require additionally degeneracy in the sum of eigenvalues as well as the difference. For evenly spaced eigenvalues, like collective spin operators, this gives a best case reduction of the scaling of the computation time from $O(d^6)$ to

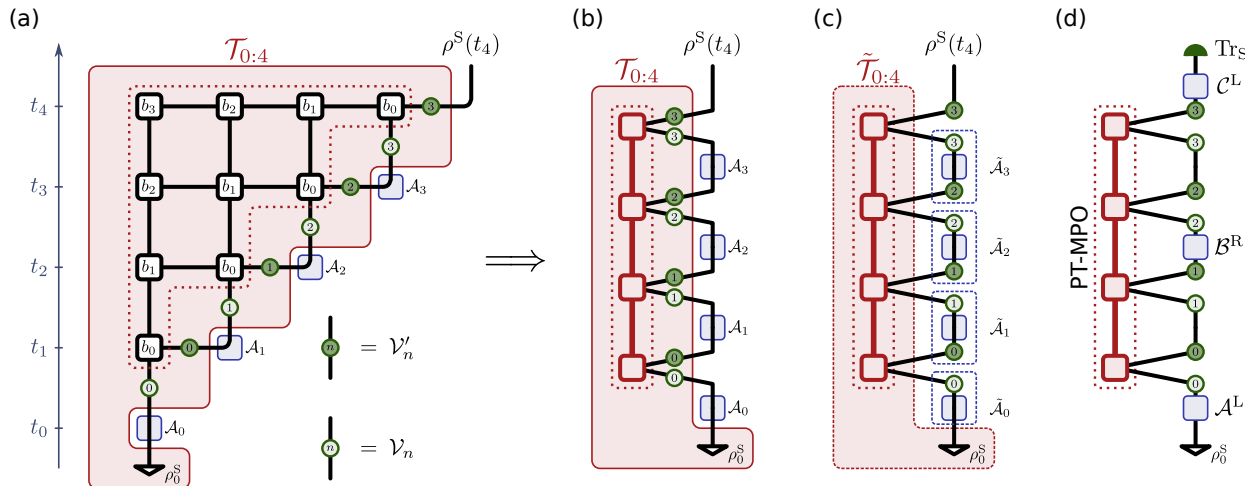


FIG. 3. (a) The TEMPO tensor network for four time steps with influence tensors b_k , system propagators $\mathcal{V}_n^{(r)}$, a set of control operations $\mathcal{A}_n$, and initial system state ρ_0^S . (b) A PT-MPO $\mathcal{T}_{0:4}$ applied to a set of control operations. (c) A PT-MPO $\tilde{\mathcal{T}}_{0:4}$ applied to a set of control operations and system propagators. (d) A PT-MPO applied to system propagators, three control operations and the trace operation for the computation of three-time correlation $\text{Tr}[\hat{C}(t_4)\hat{A}(t_0)\rho_0\hat{B}(t_2)]$ as explained in Section III B.

$O(d^3)$. In general, the reduction in numerical effort is related to the number of distinct eigenvalue differences in the system-bath coupling operator⁶¹.

Construction of PT-MPOs in OQuPy

The PT-MPO construction with PT-TEMPO is handled by the `pt_tempo_compute()` function in OQuPy. This takes, as its leading argument, a `Bath` object containing both the spectral density $J(\omega)$ of the environment and the operator $\hat{S}$ coupling it to the system. For each time step, influence tensors for the `Bath` are calculated and subsequently contracted to produce the PT-MPO. The `ProcessTensor` object returned by `pt_tempo_compute()` contains the PT-MPO and can be held in memory or written to disk for later use. When a process tensor is stored on disk it can be loaded in a “lazy way”, i.e. OQuPy will automatically load only the parts of the process tensor that are needed at any point in time to reduce memory requirements. Note that we use a monospace font throughout this article for expressions (classes, functions, and variable names) that are part of the OQuPy software package.

We now explain the computational parameters relevant to the PT-TEMPO method in OQuPy. PT-TEMPO is a numerically exact method in the sense that no approximations are required in its derivation. Error then only arises in its numerical implementation, controlled by a set of computational parameters. The calculated dynamics can, in principle (i.e., with unlimited computational resource), be made as accurate as desired by tuning those parameters.

There are three main computational parameters to be decided. The first is a memory cut-off $\text{tcut} \hat{=} \tau_{\text{cut}}$ for the PT-MPO, set physically by the period of time τ_{cut} over which non-Markovian effects persist via the environment. This may be estimated by inspecting the decay of the environment auto-correlation function. In practice, a suitable value is verified by repeating the calculation at longer tcut and checking the result is unchanged.

Second, a time step length $\text{dt} \hat{=} \delta t$ must be specified. This must be short enough so as to avoid Trotter error in the discretization of the path integral for the system dynamics that is the TEMPO tensor network. At the same time, dt fixes the grid upon which the calculation is performed, so also should be short enough to provide satisfactory resolution of the dynamics.

A precision `epsrel` completes the set of core computational parameters. This sets the relative cut-off for singular value truncation when constructing the PT-MPO. It must therefore be sufficiently small so as to avoid error resulting from the discard of physical correlations. Note that PT-TEMPO runs at a fixed precision ϵ_{rel} such that in the construction of the PT-MPO singular values smaller than ϵ_{rel} relative to the largest singular value are discarded. This is in contrast to other MPO methods where instead the bond dimension of the tensor is fixed and the precision varies.

Similar to the `tcut` parameter, to verify the suitability of chosen values of dt and `epsrel`, one may repeat the computation with more stringent (smaller) values and check the convergence of results. Smaller dt and `epsrel` (or longer `tcut`) typically imply a more resource intense calculation, so a balance should be sought between the accuracy of the dynamics and computational cost.

A practical approach to establishing convergence with

respect to `tcut`, `dt` and `epsrel` (note convergence under one parameter is generally dependent on the values of the others) is to firstly determine a suitable value of `tcut`, since this is set physically by the properties of the environment. For this purpose a utility function `plot_correlations_with_parameters()` is provided. This allows one to inspect the auto-correlation function of a given environment (encoded in a `Bath` object) and hence read off a value for `tcut` beyond which correlations are zero to a desired accuracy. In addition, `guess_tempo_parameters()` estimates from the environment and (optionally) system a reasonable set (`tcut`, `dt`, `epsrel`) which may be used as a good starting point.

Having determined a value of `tcut` and with rough values for `dt` and `epsrel`, one can then look to decrease `dt` until either 1., a satisfactory accuracy is obtained in observables of interest or 2., the error becomes non-monotonic with `dt`. The latter suggests a higher precision is required and so one should decrease `epsrel` and restart the process of decreasing `dt`. Finally, after termination condition 1., one should repeat the calculation with a smaller `epsrel` to check the results do not change. The utility of the PT-TEMPO method is in the realization that results of sufficient accuracy can be obtained at a manageable computational cost for a wide range of problems. In OQuPy, the parameters `tcut`, `dt` and `epsrel` are encapsulated in a `TempoParameters` object and passed as an argument to `pt_tempo_compute()` at the stage of creating the PT-MPO. Further discussion of numerical convergence and examples are provided in the “Computational parameters” tutorial found in the online documentation of OQuPy⁶³.

As an example, we consider a spin-boson model with a coupling operator $\hat{S}_z$ (spin operator in Z-direction) and a spectral density of the form in Eq.(5) with $\alpha = 0.1$, $\zeta = 1.0$, $\omega_c = 1.0 \text{ ps}^{-1}$, at temperature $T = 1.0 \text{ ps}^{-1}$ ($= 7.64 \text{ K}$). Here and in the following, we have set $\hbar = k_B = 1$ and give all frequencies in units of ps^{-1} . This choice of units is motivated by the fact that most physical examples discussed in Section III take place at this time scale. We note, however, that there is no handling of units in OQuPy. Hence, all input and outputs of frequency and time are without unit and need to be interpreted by the user with respect to some (arbitrary) characteristic frequency.

In Fig. 4(a) we plot the computation time taken by `pt_tempo_compute` function with the parameters `tcut` = 2.0, `dt` = 0.0625 ps, and `epsrel` = 6.1×10^{-5} for 32 time steps. We see that the computation time to generate the process tensor scales with the system size and that it can be improved by the use of the previously discussed degeneracy checking.

III. PT-MPO USE CASES

In this section we discuss a number of use cases that are accommodated by the current version (v0.5) of the

OQuPy package⁶⁴. In addition to the calculation of dynamics for a quantum system coupled to one or more structured environments^{23,24}, these include calculations of multi-time correlation functions⁶⁵, optimization of control protocols^{26,30}, and the dynamics of *many-body* open quantum systems described by chains²⁸ or many-to-one networks of open quantum systems²⁹. The range of these examples demonstrate the versatility of the process tensor approach for extracting static and dynamic quantities from models of open quantum systems.

All of the use cases presented in this section apply pre-computed PT-MPO representations of environments for specific tasks. We will always obtain such a PT-MPO by employing the PT-TEMPO method for Gaussian bosonic environments described in the previous Section II A. We note again, however, that the methods presented in this section are agnostic to the origin of a PT-MPO, and as such can also be readily applied to PT-MPOs for other types of environments that have been computed with other methods and software.

The applicability of the methods presented in this section depends to a large extent on the complexity of the PT-MPOs that enter into the computation. This complexity is typically reflected in the bond dimension of PT-MPOs. The main restriction of the methods presented in this section is thus that for the environment of interest a sufficiently accurate PT-MPO with a manageable bond dimension ($\chi \lesssim 1000$) exists, and that it can be obtained with one of the known methods. Although this is certainly not always the case, the literature on known methods to obtain PT-MPOs^{21,38,46–58} demonstrates that this is feasible for a large range of different highly relevant non-Markovian environments.

Several of the scenarios considered in this section may also be simulated using other numerical methods. While it would be desirable to know in advance which numerical method performs best for a given scenario, such a judgment is difficult because the performance of a method will not only depend on the type of problem but generally even on the parameters of the problem. A systematic and fair comparison among the many known methods is thus hard to achieve and outside the scope of this work.

However, when one is directly or indirectly interested in multiple sets of multi-time correlations then the PT-MPO approach has a clear advantage over other approaches. This is because the process tensor is the minimal amount of information that encodes all multi-time correlations. Any method that does not compute a process tensor thus needs to be reiterated for each new set of multi-time correlation. An example scenario that relies on the knowledge of all multi-time correlations is the optimization of control protocols (section III C). This is because the different system propagators at each time step can be understood as a high-order multi-time correlation probe of the environment. Also, in a chain of open systems (section III E), the interaction of neighboring sites effectively acts as a mutual high-order multi-time correlation probe of the sites and its environments.

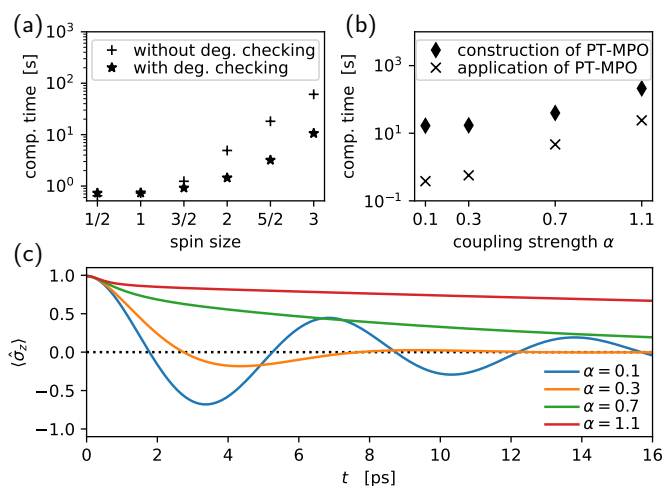


FIG. 4. (a) Computation times for the construction of a PT-MPO using PT-TEMPO with/without degeneracy checking for different system sizes, with parameters described in the main text of Section II A. (b) Computation times for the construction and application of a PT-MPO for different coupling strengths of a biased spin-boson model described in the main text of Section III A. (c) Dynamics of the biased spin-boson model for different coupling strengths.

The PT-MPO approaches to such scenarios can thus be expected to be generally more efficient than alternative methods.

For every use case we outline the inner workings of the numerical method and show the results of an example, forgoing further discussion of the resulting physics. For more details on the methods and physical discussion of results, we refer the reader to the research articles of the individual methods^{22–30}. For more details on their implementation using OQuPy, we refer the reader to the online documentation of OQuPy⁶³. The online documentation includes tutorials for several of the use cases presented in this section, as well as a complete reference for all modules in OQuPy.

All computation times quoted in this paper have been determined by running OQuPy on a *single* core of an Intel i7 (8th Gen) processor. We note that for most of the methods presented in this article the overwhelming majority of computation time is spent in LAPACK (linear algebra package) routines and hence it is possible to speed up the computation by using multiple cores and a suitable LAPACK implementation. In addition, the PT-TEBD method (explained in section III E below) also supports both explicit multithreading and multiprocessing.

A. Dynamics of non-Markovian open quantum systems

Given a PT-MPO, constructed using the `pt_tempo_compute()` function or otherwise, single-time dynamics are computed in OQuPy following the introduction of a `System` object. This describes

the unitary evolution of the system (i.e., the system Hamiltonian) as well as coupling to any additional Markovian baths (Lindblad operators and rates). The PT-MPO and `System` object are passed together with an argument prescribing the initial state of the system ρ_0^S to the `compute_dynamics()` function, which performs the tensor network contractions required to yield the dynamics. A `Control` object describing a set of control operations or interventions in the system dynamics may also be provided at this stage.

The result returned by `compute_dynamics()` is a `Dynamics` object. This contains a list of reduced density matrices of the system for each time step from which, for example, the time dependent expectation values of system observables may readily be derived. Note that the same approach can be used in the case of coupling to multiple non-Markovian environments described by separate PT-MPOs; see Section III D.

To provide an example, we consider the quench dynamics of a simple spin-boson model with a Ohmic spectral density at different coupling strengths. For this, the overall Hamiltonian is of the form of Eq. (3), with $H^S = \frac{\Omega}{2}\hat{\sigma}_x$, coupling operator $\hat{S} = \frac{1}{2}\hat{\sigma}_z$. We assume the spectral density $J(\omega)$ in the form of Eq. (5), where $\Omega = 1.0 \text{ ps}^{-1}$, $\zeta = 1.0$, $\omega_c = 5.0 \text{ ps}^{-1}$, and temperature $T = 0$. Figure 4(c) shows the dynamics of $\langle \sigma_z(t) \rangle$, starting in the spin-up state $|\uparrow\rangle$, for different coupling strengths α . For each coupling strength α we first use the `pt_tempo_compute()` function to generate a PT-MPO, and then apply the `compute_dynamics()` function to obtain the dynamics. In Fig. 4(b) we show the computation times for those calculations using `tcut = 4.0 ps`, `dt = 0.0625 ps`, `epsrel =  $6.1 \times 10^{-5}$` for $N = 256$ time steps. The maximal memory requirement for these calculations was 4.5 GiB.

In Fig. 4(b) we can observe clearly that *applying* a PT-MPO takes much less time than *constructing* a PT-MPO. This is because during the creation of the PT-MPO, the PT-TEMPO method identifies and removes negligible correlations by performing truncated SVDs, which constitutes the bulk of the overall computation. This process of removing negligible correlations is essential to the general method of using PT-MPOs, as it compresses the otherwise exponential complexity of a non-Markovian quantum process to an MPO with a finite bond dimension. This reduced complexity of the PT-MPO is particularly useful when one is interested in a large set of different multi-time correlations or the dynamics for many different system Hamiltonians, because in such cases the difficult creation of the PT-MPO has to be performed only once, while the repeated application of it can be performed much more quickly. This fact will be used extensively in the following two use cases on multi-time correlations and the optimization of control protocols.

B. Multi-time correlations of non-Markovian open quantum systems

The calculation of multi-time correlations for open quantum systems often involves application of the quantum regression theorem, which relies on making a Markovian approximation². However, there exist many scenarios—from exciton dynamics in quantum dots to natural light-harvesting—where such an approximation is not sufficient^{3,8,10,12}. Moreover, as mentioned in the introduction, even if a Markovian description correctly captures the reduced dynamics of the system, an accurate description of multi-time observables is not guaranteed^{32,66}. The PT-MPO, on the other hand, not only encodes the evolution of the density matrix but also all multi-time correlations. These are accessed by simply inserting the operators at the corresponding times as control operations, depicted in Fig 3(d).

Figure 3(d) shows the three-time correlation

$$R = \text{Tr} [\hat{C}(\tau_3) \hat{A}(\tau_1) \rho_0 \hat{B}(\tau_2)] \quad (8)$$

with $\tau_1 = t_0$, $\tau_2 = t_2$, and $\tau_3 = t_4$, where $\hat{X}(\tau) = U^\dagger(\tau) \hat{X} \hat{U}(\tau)$ denotes system operators in the Heisenberg picture with respect to the *total* Hamiltonian. Also, ρ_0 is the total initial state and Tr denotes the trace over the total (i.e. system *and* environment) Hilbert space. The control operations that need to be inserted into the process tensor at the corresponding times are the left and right acting superoperators⁶⁷ $\mathcal{A}^L[\rho] = \hat{A}\rho$, $\mathcal{B}^R[\rho] = \rho\hat{B}$, and $\mathcal{C}^L[\rho] = \hat{C}\rho$. After the last control operation we trace over the system Hilbert space. Inserting a left or right acting system operator and tracing over the system is equivalent to computing the expectation value of that operator. Therefore, multi-time correlations for various final times can be computed in a single run of the simulation. To vary any of the earlier time arguments, e.g. for t_1 time steps, the simulation is repeated i times, shifting the position of the earlier control operations accordingly. As explained in Section III A, since the process tensor is independent of the system propagators and control sequence, the same PT-MPO can be applied repeatedly, which drastically decreases the necessary computation time.

We note that the set of multi-time expectations that can be calculated this way require that the set of operators acting to the left and those acting to the right are separately time ordered. There is no restriction on how the times of left-acting and right-acting operators are related, however one cannot calculate arbitrary out-of-time-order-correlations⁶⁸. Also, we note that the convergence of multi-time correlations with an increased accuracy of the PT-MPO needs to be checked separately from convergence of single-time observables. This is because SVD truncations in the PT-MPO tend to affect the accuracy of multi-time correlations more compared to single-time observables.

In OQuPy, multi-time correlations are computed with the function `compute_correlations_nt()`. This requires a `System` and `ProcessTensor` object, together with three lists specifying the operators, the times at which they should be applied and whether each operator should be applied to the left or right of the (total) density matrix.

To demonstrate the utility of the code, we apply it to a three-level system and simulate a linear absorption and 2D electronic spectroscopy (2DES) measurement⁶⁵. 2DES is a non-linear spectroscopy technique that uses a series of laser pulses to probe energy and charge transport on a femtosecond timescale. The resulting signal is commonly visualized as a 2D spectrum that correlates the excitation and detection frequencies excited by the lasers, see Fig. 5(b). Because of its sensitivity to complex phases, 2DES is particularly useful for investigating the coherent dynamics of energy excitations in real time. For a more detailed overview, we refer the reader to Refs. 69 and 70.

As shown in Fig. 5(a), our model is a three-level system with a linear coupling to a phonon bath. The total Hamiltonian has the form of Eq. (3), where the system Hamiltonian and coupling operator are

$$\hat{H}_S = (\epsilon + \lambda)(|1\rangle\langle 1| + |2\rangle\langle 2|) + \Omega(|1\rangle\langle 2| + \text{H.c.}), \quad (9)$$

and $\hat{S} = (|1\rangle\langle 1| - |2\rangle\langle 2|)$. This Hamiltonian describes an excitonic dimer embedded in a vibrational environment with ground state $|0\rangle$ and monomer states $|1\rangle, |2\rangle$. Each monomer state has a bare energy $\epsilon + \lambda$, where λ represents the bath reorganization energy (defined below). The coupling Ω creates two delocalized exciton states $|\pm\rangle = \frac{1}{\sqrt{2}}(|1\rangle \pm |2\rangle)$ with energies $E_\pm = \epsilon + \lambda \pm \Omega$. Since the dynamics of the type of molecules studied with 2DES generally take place on a picosecond timescale¹², we set $\epsilon = 5.0 \text{ ps}^{-1}$ and $\Omega = 2.0 \text{ ps}^{-1}$. The spectral density $J(\omega)$ is given by Eq. (5) with $\zeta = 1.0$, $\alpha = 0.1$ and $\omega_c = 3.04 \text{ ps}^{-1}$, and leads to the reorganization energy $\lambda = \int_0^\infty \frac{1}{\omega} J(\omega) d\omega = 2 \alpha \omega_c$.

A 2DES signal can be modeled by a series of four-time correlation functions that represent all possible light-matter interaction pathways. Here we will consider the following four correlation functions:

$$\begin{aligned} R_1 &= \text{Tr} [\hat{V}(\tau_4) \hat{V}(\tau_1) \rho_0 \hat{V}(\tau_2) \hat{V}(\tau_3)] \\ R_2 &= \text{Tr} [\hat{V}(\tau_4) \hat{V}(\tau_2) \rho_0 \hat{V}(\tau_1) \hat{V}(\tau_3)] \\ R_3 &= \text{Tr} [\hat{V}(\tau_4) \hat{V}(\tau_3) \rho_0 \hat{V}(\tau_1) \hat{V}(\tau_2)] \\ R_4 &= \text{Tr} [\hat{V}(\tau_4) \hat{V}(\tau_3) \hat{V}(\tau_2) \hat{V}(\tau_1) \rho_0], \end{aligned} \quad (10)$$

where $\hat{V} = |0\rangle\langle 2| + |2\rangle\langle 0|$. To plot a 2D spectrum, each correlation function is Fourier transformed with respect to the first ($\tau_2 - \tau_1$) and last ($\tau_4 - \tau_3$) time delay, corresponding to an excitation (ω_{exc}) and detection (ω_{detec}) frequency axis. We then sum together the real parts of each pathway to obtain the total spectrum, which is shown in Fig. 5(b). Here for simplicity, we have set the second time delay $\tau_3 - \tau_2$ to zero.

This is the author's peer-reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

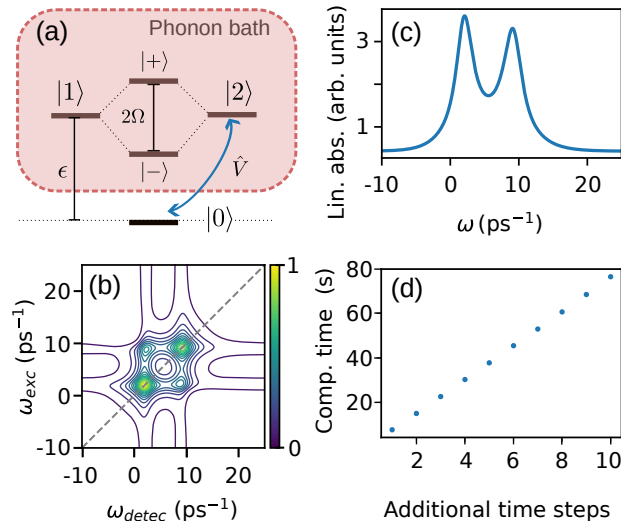


FIG. 5. (a) Sketch of model employed in the multi-time correlations use case in Eq. (9); $\epsilon = 5.0 \text{ ps}^{-1}$, $\Omega = 2.0 \text{ ps}^{-1}$. (b) Corresponding 2D spectrum calculated from the four-time correlation functions in Eq. (10) at $T = 13.0 \text{ ps}^{-1}$ ($= 100 \text{ K}$). (c) Corresponding linear absorption spectrum at $T = 13.0 \text{ ps}^{-1}$. (d) Scaling of the computation time for calculating the four-time correlation function $R_4(\tau_1, \tau_4)$ when adding additional time steps to τ_1 . Note that the same PT-MPO was used for each data point, which took 44 s to compute with $\text{tcut} = 50.0 \text{ ps}$, $\text{dt} = 0.1 \text{ ps}$ and $\text{epsrel} = 10^{-6}$.

Simulating a linear absorption spectrum for our model requires computing the two-time correlation function $\text{Tr}[\hat{V}(\tau_2)\hat{V}(\tau_1)\rho_0]$. As with the two-dimensional case, the spectrum is given by the Fourier transform of this function with respect to $\tau_2 - \tau_1$, and is shown in Fig. 5(c). The spectrum contains two peaks corresponding to the $|+\rangle$ and $|-\rangle$ states excited by the dipole operator $\hat{V}$. The 2D spectrum (Fig. 5(b)) contains two diagonal peaks, reflecting the same transition frequencies observed in the linear absorption spectrum. Furthermore, two off-diagonal peaks that cross-correlate the two frequencies can be observed; these signify the presence of the electronic coupling Ω between the excited states.

To illustrate how this computation scales with the range of time steps, we first consider the correlation function $R_4(\tau_4)$ in Eq. (10). As a function of the final time argument τ_4 only, the simulation runs once and took 8 s to compute. If we additionally vary any of the earlier time arguments, e.g. τ_1 in $R_4(\tau_1, \tau_4)$ over i time steps, the simulation is repeated i times. As shown in Fig. 5(d), the computation time therefore scales linearly with the number of additional time steps. The 2D spectrum in Fig. 5(b) was computed for 40 time steps each in τ_1 and τ_4 and consists of the four correlation functions in Eq. (10), giving a total computation time of 21 minutes. The maximal memory requirement for these calculations was 3.8 GiB.

C. Optimization of control protocols

As mentioned in the end of Section III A, the fact that the PT-MPO does not depend on the system Hamiltonian allows one to efficiently determine optimal control protocols for non-Markovian open quantum systems^{26,30}. Optimal control involves defining an objective function to quantify the success of a given protocol, and then maximizing this function over the set of controls. A common scenario is for the protocols to correspond to different time-dependent system Hamiltonians, and for the value of the objective function to be determined by the final state. The optimization can be done by first computing the PT-MPO of the given environment interaction, and then repeatedly applying different time-dependent system Hamiltonians. This has the advantage that each trial system Hamiltonian can be applied with minimal computational effort to the same pre-computed PT-MPO.

A further advantage of the PT-MPO approach is that it provides a natural and efficient way to compute the gradient of the objective function with respect to the parameterization of the system Hamiltonian. This allows one to use the gradient in the optimization process, drastically reducing the computation time required. In this section, we outline how OQuPy is used to calculate the gradient of an objective function with respect to a set of parameters.

The basis for the gradient calculation is the observation that the process tensor is a multi-linear map from the set of system propagators—or, more generally, system operations—to the final state. This implies that the derivative of the final state with respect to a particular system propagator is the diagram for the final state, such as Fig. 3(c), with that propagator omitted. It can be constructed by combining two partial diagrams, one obtained by contracting the network starting from the bottom, and one starting from the top, in each case stopping just before the omitted propagator. Furthermore, the partial diagrams required to compute the derivatives with respect to all the propagators, i.e. the gradient, are computed in the course of one forwards-in-time propagation, from bottom to top, and one backwards-in-time propagation, from top to bottom. The forward propagation would, in any case, be needed to compute the objective function, so the additional cost of this process is that of the backwards propagation and contractions of the partial diagrams.

In OQuPy this calculation is focused on the problem of computing the gradient of an objective function, defined in terms of the final state, $Z(\rho_f)$, with respect to some parameterization of the time-dependent system Hamiltonian. We define a set of parameters $\{c_\alpha\}$, with $\alpha = 0, \dots, M$ which appear in the system Hamiltonian, and take the values $c_\alpha(t_n) = c_\alpha^n$ at the n^{th} time step. The derivative of the objective function with respect to

c_α^n is given by the chain rule,

$$\frac{\partial Z}{\partial c_\alpha^n} = \underbrace{\sum_{i,j,k}^{d_{H_S}^2} \frac{\partial Z}{\partial \rho_f^i} \frac{\partial \rho_f^i}{\partial \mathcal{V}_n^{jk}} \frac{\partial \mathcal{V}_n^{jk}}{\partial c_\alpha^n}}_{\text{state_gradient()}} , \quad (11)$$

where ρ_f is the final state and $\mathcal{V}_n$ the propagator(s) for the time step n . The second factor in Eq. (11) can be computed for all n , as discussed above, using one forwards and one backwards propagation, and combined with the remaining terms to compute the gradient, $\frac{\partial Z}{\partial c_\alpha^n}$ for all n, α .

There are three steps in computing the gradient of an objective function of a final state with respect to a set of parameters in OQuPy. The first step is to define a system via the `ParameterizedSystem` object. This object inherits from `BaseSystem` and represents a Hamiltonian which depends on M parameters $\{c_0, c_1, \dots\}$. Instantiation requires the user to supply a function which takes values for these parameters and returns the Hamiltonian operator. The `ParameterizedSystem` class then handles the calculation of the propagators $\mathcal{V}_n$ and propagator derivatives $\frac{\partial \mathcal{V}_n}{\partial c_\alpha^n}$ during the gradient calculation via the `get_propagators()` and `get_propagator_derivatives()` methods. Alternatively, the user can provide a function which returns the propagator derivatives.

The simplest calculation one can do with a `ParameterizedSystem` object is to compute the dynamics. This can be done by calling the routine `compute_gradient_and_dynamics` and providing values of the parameters, c_α^n for all time steps. An additional detail is that the calculations use a second-order Trotter splitting, so that, as shown in Fig. 3, propagation over a full time step involves two propagators, each over half a time step, with one, the pre-propagator $\mathcal{V}_n$ applied to the corresponding input leg of the PT-MPO, and one, the post-propagator $\mathcal{V}'_n$, to the output leg. The computation of the dynamics thus requires the values of the parameters on both halves of every time step. The results are accurate up to second order in Δt provided the parameters are continuous within each full time step, i.e. the difference in the Hamiltonian between the first and second halves of each step is $O(\Delta t)$. Note there is no requirement for continuity between full time steps, so that discontinuous Hamiltonians can be treated correctly.

To compute the gradient of an objective function, the `ParameterizedSystem` object, along with the PT-MPO, initial state, and values for the control parameters, can be passed to the function `state_gradient()`. The objective function can be specified in two ways. For the case of a linear objective function, $Z = \sum_{ij} (A)_{ij} (\rho_f)_{ij}$, the first factor in Eq. (11) is $\frac{\partial Z}{\partial \rho_f} = A$, which can be provided to `state_gradient()` as an array. An example arises for state transfer where we seek to reach a pure target

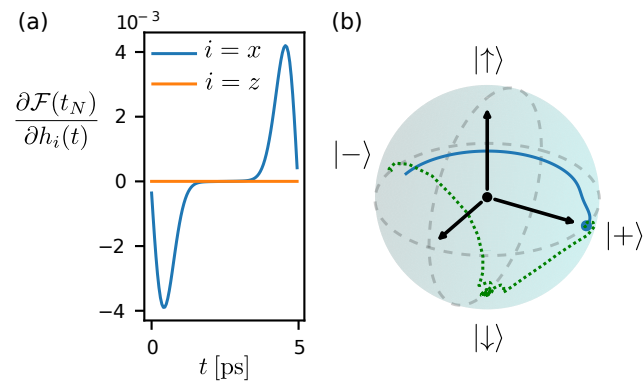


FIG. 6. (a) Derivative of the fidelity at the final time t_N with respect to the parameters at each time-step $h_x(t)$ and $h_z(t)$ for initial controls. (b) Dynamics of states on Bloch sphere under initial (blue, solid) and optimized (green, dotted) controls.

state, $\rho_t = |\sigma\rangle\langle\sigma|$, at the final time. The fidelity $\mathcal{F} = \langle\sigma|\rho_f|\sigma\rangle$ is then a linear objective function with $A = \rho_t^T$. For nonlinear objective functions, the user passes a function to `state_gradient()`. This will be called with the computed final state as an argument and should return $\frac{\partial Z}{\partial \rho_f}$ evaluated for that state.

The `state_gradient()` function handles the forwards and backwards propagations as well as the application of the chain rule. It first passes the arguments to the `compute_gradient_and_dynamics()` function, yielding the tensors $\frac{\partial Z}{\partial \rho_f^i} \frac{\partial \rho_f^i}{\partial \mathcal{V}_n^{jk}}$. These tensors are then passed to the `chain_rule()` function, where they are combined with the propagators and propagator derivatives for each half time step. The result is a list of derivatives of the objective function with respect to the control parameters at each half time step. The final dictionary returned by `state_gradient()` function includes a `Dynamics` object, the gradient tensors $\frac{\partial Z}{\partial \rho_f} \frac{\partial \rho_f}{\partial \mathcal{V}_n^{jk}}$, and the derivatives at each half time step $\frac{\partial Z}{\partial c_\alpha^n}$.

Fig. 6(a) shows an example of the gradient computation for a two-level system coupled to a Gaussian bosonic bath, with a super-Ohmic spectral density $J(\omega)$ given by Eq. (5) with $\zeta = 3.0$, $\alpha = 0.126$, and $\omega_c = 3.04 \text{ ps}^{-1}$. We consider an initial state $\rho_0 = |+\rangle\langle+|$, and take the objective function to be the fidelity $\mathcal{F}(\rho_f, \rho_t)$ to a target state $\rho_t = |-\rangle\langle-|$. The system Hamiltonian and coupling operator are $\hat{H}_S = h_x(t)\hat{\sigma}_x + h_z(t)\hat{\sigma}_z$ and $\hat{S} = \frac{\hat{\sigma}_z}{2}$, and the dynamics is simulated with $N = 100$ time steps. The gradient is shown for the case of control fields $h_x(t) = 0$ and $h_z(t) = \pi/t_N$, which are the optimal controls in the absence of the environment. The dynamics of the state on the Bloch sphere under this set of controls is depicted by the blue solid line in Fig. 6(a).

The `state_gradient()` function can be used within a numerical optimization routine to determine optimal control protocols. The green dotted curve in 6(b) shows the trajectory for optimal controls determined in

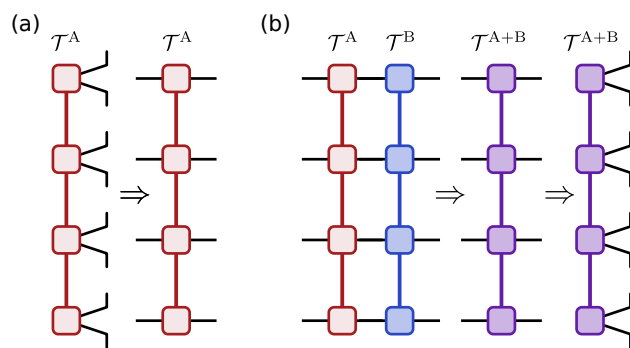


FIG. 7. Tensor network sketch of the addition of two PT-MPOs. (a) Suitable reshaping of a PT-MPO. (b) Contraction of two PT-MPOs to a single PT-MPO that represents the corresponding joint environment.

this way, using the L-BFGS algorithm implemented in the `scipy.optimize.minimize()` function⁷¹. This optimization was done using bounds $|h_x| \leq h_x^{\max} = 5.0\pi \text{ ps}^{-1}$ and $|h_z| \leq h_z^{\max} = 1.0\pi \text{ ps}^{-1}$, which restrict the speed of the unitary evolution.

For this optimal control, the fidelity of the final state with respect to the target state is 0.9991. From the dynamics of the Bloch vector it can be seen that the increase in fidelity over the unoptimized solution arises because the state is transferred to an intermediate state $|\downarrow\rangle$, which is not subject to decoherence⁷². A different mechanism can appear with smaller bounds on the control fields, which prohibit access to that decoherence-free subspace. In that case, it has been shown that for larger process durations there can be an increase in the fidelity³⁰ due to information previously lost to the environment being restored by the optimization. The optimization utilizes information back-flow from the environment by maximizing the non-Markovianity of the map.

D Open system dynamics with multiple environments

It can sometimes be sensible, and even necessary, to express the external influences on a system in terms of multiple environments. A typical case in physics is that of an optically active system that is also strongly coupled to its vibrational environment. For example, in systems studied for light harvesting and energy transfer the molecular vibrational degrees of freedom can play a key role^{73–77}. Strong coupling to vibrations has also been shown to have a distinct effect on the optical properties of even relatively simple systems^{27,78}; the combined effect of multiple environments in this way is termed *non-additive*. Another example of non-additivity occurs when considering the combined effect of the leads and vibrational environments of molecular nanjunctions^{79,80}; here using an additive treatment can even lead to a violation of the Carnot bound on efficiency⁸¹.

Typically PT-MPOs containing information on mul-

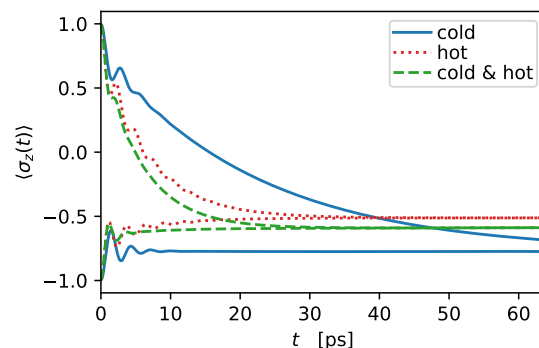


FIG. 8. Evolution of a spin coupled to a cold bath, a hot bath, and to both baths simultaneously. We show the dynamics for two different initial states, $|\uparrow\rangle$ and $|\downarrow\rangle$.

iple environments can be much more costly to construct and store; this is especially true when the environments couple via non-commuting system operators. Fortunately, we can avoid this cost by making use of the additive property of PT-MPOs, namely that two PT-MPOs can be contracted together to yield a single process tensor with the effect of both environments. This allows us to construct one PT-MPO for each environment and store them separately, only combining them when necessary. This step corresponds to a standard MPO-MPO multiplication as depicted in Figure 7. However, combining the entirety of each environment's PT-MPO typically offers little to no improvement over constructing the single large object in the first place. We can avoid this in certain calculations by only combining the tensors necessary at each step. For example, when computing an observable at a single time step t_n we can combine the PT-MPOs one time step at a time absorbing the tensor of each previous time step until we reach t_n . This avoids the extra memory requirements in combining PT-MPOs and the computational overhead in compressing the resulting object.

As a demonstration we consider a two-level system coupled to both a hot and cold bath of bosons. The total Hamiltonian is analogous to Eq. (3), but with two interaction and two environment parts H_a^I , H_a^E , and H_b^I , H_b^E , with bosonic environment annihilation operators $\hat{a}_k$ and $\hat{b}_k$, respectively. We choose the system Hamiltonian $\hat{H}_S = \frac{\epsilon}{2}\hat{\sigma}_z + \frac{\Omega}{2}\hat{\sigma}_x$, with $\epsilon = 2.0 \text{ ps}^{-1}$, and $\Omega = 1.0 \text{ ps}^{-1}$. Both environments have the same Ohmic spectral density $J(\omega)$ given by Eq. (5), with $\zeta = 1.0$, $\alpha = 0.16$, and $\omega_c = 1.0 \text{ ps}^{-1}$. The environments are distinguished by the spin component to which they couple and their temperature. The cold bath, at temperature $T_{\text{cold}} = 0.8 \text{ ps}^{-1}$ ($= 6.11 \text{ K}$), couples via $\hat{S}_a = \hat{\sigma}_z/2$, while the hot bath, at temperature $T_{\text{hot}} = 1.6 \text{ ps}^{-1}$ ($= 12.22 \text{ K}$), couples via $\hat{S}_b = \hat{\sigma}_x/2$. The dynamics of this model for initial states spin-up and spin-down are plotted in Figure 8 and the combined effect of the baths, when compared with their separate impact, can be seen clearly.

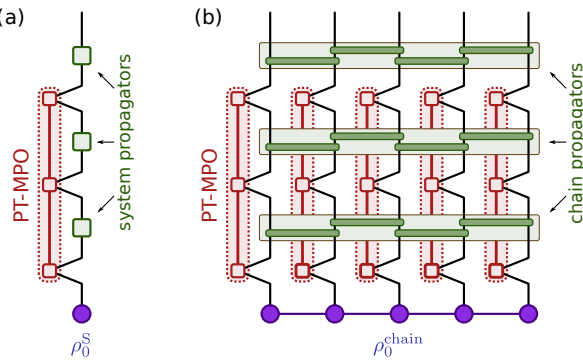


FIG. 9. Tensor network for the evolution of (a) a single system and (b) a chain of systems coupled to their local environment. Panel (a) shows the state of a system after three time steps evolved by system propagators under the influence of an environment (represented by an PT-MPO), starting from the initial state ρ_0^S . Panel (b) shows the state of a chain of systems after three time steps under the influences of locally coupled environments, starting from an initial state represented as an MPS in Liouville space. The dark green rectangles are Trotterized chain propagators.

The combination of multiple PT-MPOs corresponds to standard tensor network operations but, as mentioned, this is often very demanding; in the above example the PT-MPOs had a maximum bond dimensions of $\chi_{\text{cold}} = 35$ and $\chi_{\text{hot}} = 41$. OQuPy is instead able to make use of the more efficient calculation of dynamics introduced earlier in this section. Doing so requires simply providing the `compute_dynamics()` function with a list of PT-MPOs of the multiple environments and, provided they are compatible, the dynamics step-by-step will be computed. The construction of each PT-MPO for Fig. 8 took 40 s (cold) and 51 s (hot) and the dynamics for combined simulation took 8 s compared with 3 s and 5 s for the separate cold and hot bath dynamics respectively. The maximal memory requirement for these calculations was 220 MiB.

In this section we highlighted the simplicity with which the dynamics of a system coupled to multiple environments can be computed with OQuPy. The implementation allows for great flexibility in considering the effect of arbitrary combinations of PT-MPOs on the dynamics of the system. The above example demonstrates this with two PT-MPOs for Gaussian bosonic baths, but—like all methods in this section—is also readily applicable to combinations of PT-MPOs that represent environments of different nature created with other methods and other software.

E. Chains of open quantum systems

The methods and examples considered so far always involved a single (small) system coupled to one or more environments. There is, however, a range of interest-

ing physical scenarios where the system of interest is a many-body system that couples to one or more environments^{82–86}. Such scenarios are of importance for fundamental research, such as the study of strong coupling quantum thermodynamics^{13–16,18,19}, as well as technological and biological applications^{10,11,17,87–90}. The method presented in this section enables the computation of the dynamics and multi-time correlations of chains of general open quantum systems²⁸. It is a combination of the time evolving block decimation (TEBD) method⁹¹ in Liouville space with PT-MPOs on each site to include the influence of locally coupled environments (see Fig. 9). We call this tensor network method *PT-MPO augmented TEBD* (or short *PT-TEBD*) because the TEBD method is augmented with one additional leg for each environment connecting the site with its PT-MPO. This augmented leg encodes the correlations of the site with its environment. Figure 9(b) shows the tensor network for a first order Trotterized PT-MPO augmented TEBD tensor network. This tensor network is suitable for simulating a chain of locally interacting systems where each site may couple strongly to its individual environment. For this method the environment-system coupling must be local, i.e scenarios with environments that couple directly to other environments or multiple sites are outside of the scope of this method.

The PT-TEBD method has three convergence parameters. These are (1) the Trotterization time step δt (which should be the same as the time step chosen for the PT-MPO), (2) the Trotterization order O_{TEBD} (only 1st and 2nd are currently implemented), and (3) the relative cutoff ϵ_{TEBD} for the SVD truncation along the spatial direction of the chain. The method is limited to chains whose state can be well approximated by an matrix product state (MPS) of some finite bond dimension ξ , as well as environments whose process tensor can be well approximated by an MPO of bond dimension χ . The computational complexity is then dominated by performing the singular SVD involved in compressing the spatial MPS after the application of the system propagators. In the worst case the largest matrices involved are of the dimension $(\xi\chi d^2) \times (\xi\chi d^2)$, where d is the Hilbert space dimension of a single site. In many cases this can be reduced to $(\eta d^2) \times (\eta d^2)$ with $\xi \lesssim \eta \leq \xi\chi$ by a careful choice of contraction and SVD order (see appendix A of Ref. 28). The overall simulation of an N site chain for K time steps thus takes $O(NK\eta^3 d^6)$ operations. This algorithm is—like the canonical TEBD algorithm—well suited for parallel computing, since each pair of neighboring sites can be evolved separately.

To demonstrate the capabilities of this method we present the dynamics of an anisotropic XY-model where the first K of N sites each couple strongly to an environment (see sketch in Fig. 10). The chain Hamiltonian is

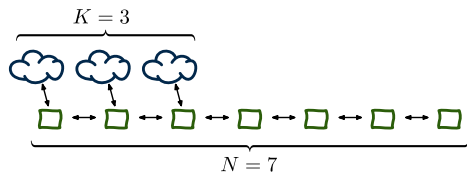


FIG. 10. Sketch of an interacting chain of systems (green squares) with length $N = 7$ coupling to local environments (blue clouds) on the first $K = 3$ sites.

of the form

$$\hat{H}_{XY} = \sum_{n=1}^N \epsilon \hat{s}_n^z + \sum_{n=1}^{N-1} [(J - \eta) \hat{s}_n^x \hat{s}_{n+1}^x + (J + \eta) \hat{s}_n^y \hat{s}_{n+1}^y], \quad (12)$$

with spin-1/2 operators $\hat{s}_n^\gamma = \hat{\sigma}_n^\gamma/2$ at site n , onsite energy $\epsilon = 1.0 \text{ ps}^{-1}$, coupling strength $J = 1.0 \text{ ps}^{-1}$ with anisotropy $\eta = 0.04 \text{ ps}^{-1}$. In OQuPy this chain Hamiltonian is encoded in a `SystemChain` object. As an environment we choose a bosonic bath with an Ohmic spectral density $\zeta = 1$, $\alpha = 0.32$, $\omega_c = 1.0 \text{ ps}^{-1}$ (see Eq. (5)) and coupling operator $\hat{S} = \hat{\sigma}_z/2$. The corresponding PT-MPO has been computed using the PT-TEMPO algorithm with convergence parameters `tcut` = 4.0 ps, `dt` = 2^{-4} ps, and `epsrel` = 2^{-16} . For the PT-TEBD algorithm we use $\epsilon_{\text{TEBD}} = 2^{-16}$ and $O_{\text{TEBD}} = 2$. The parameters are collected in a `PtTebdParameters` object and passed into a `PtTebd` object together with the above `SystemChain` instance and a list of process tensors. We choose the state $|\uparrow\downarrow\downarrow\cdots\downarrow\rangle$ as the initial spin chain state, and are interested in the time evolution of $\langle \uparrow | \rho^{(n)}(t) | \uparrow \rangle$ of each spin n . The initial state is encoded in an `AugmentedMPS` object, which for product states takes a list of initial density matrices of each site. It is however also possible to specify a delocalized initial state, by passing an initial MPS through a list of rank-3 tensors.

Figure 11(a) shows the dynamics of a spin chain of length $N = 7$ for $K = 0, \dots, 3$ attached environments. We can observe in Fig. 11(a) that the spreading of the initial excitation is slowed down with every additional environment coupled to the chain. Although the PT-TEBD algorithm does not allow for direct interaction among the environments, it still captures the correlations among the environments that build up through the chain. These correlations are reflected in the bond dimension of the augmented MPS. Figure 11(b) shows the maximal bond dimension of each bond, for different K . While we can observe significant growth of bond dimensions with a growing number of environments K , their values are still well below the possible maximum that is given by the system Hilbert space dimension ($d = 2$) and the bond dimensions of the PT-MPOs ($\chi = 11$). For $K = 3$ this would give a maximum at bond 2 with $(\xi d^2)^2 = 1936$ which is an order of magnitude larger than the observed value of 167. The maximal memory requirement for these

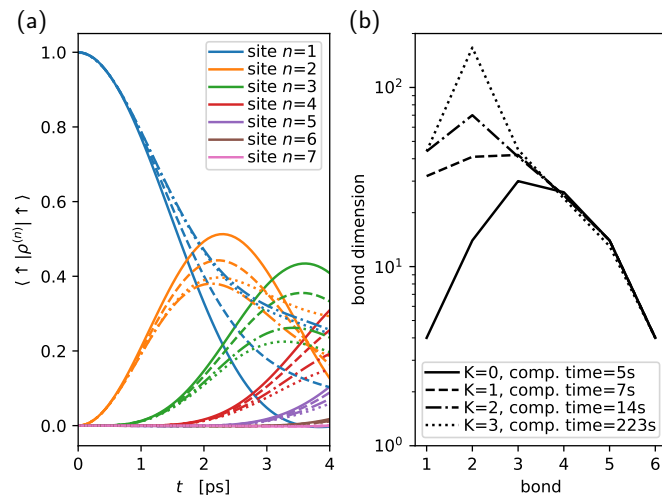


FIG. 11. OQuPy results for the anisotropic XY-model with $N = 7$ and $K = 0 \dots 3$. (a) Evolution of the chain after the quench at $t = 0$. (b) Maximal bond dimension of the augmented MPS during the entire evolution at different bonds. The different line styles in panel (a) correspond to the different values of K as shown in the legend of panel (b).

calculations was 1.1 GiB.

We note that a few alternative methods for chains of non-Markovian open quantum systems exist in the literature^{35,92–100}. While a quantitative comparison with these methods would need to be done case by case, qualitatively the PT-TEBD approach stands out in two ways. First, it approaches the challenging many-body problem sequentially by systematically reducing the numerical complexity originating from system-environment correlations through compression of the PT-MPO, before integrating them into the full many-body problem. Second, the method is agnostic about the origin of the process tensor, i.e. it is applicable to any environment, given its PT-MPO.

Because the simulation of a chain of non-Markovian open quantum systems is generally a very challenging task, we have in the above example chosen an environment with a small PT-MPO bond dimension such that the computation can still be performed on a single core in a matter of minutes. In its original publication²⁸ the PT-TEBD method has been applied to a 21 site long XYZ-Heisenberg spin chain with strongly coupled bosonic environment on every site (with $\chi = 37$) using only moderate computational resources, while a similar simulation of the same example would not be feasible with any of the alternative methods listed above. Finally, we note that because PT-TEBD also yields correct multi-time correlations it can potentially be helpful for studying the thermalization of strongly coupled open many-body quantum systems²⁸.

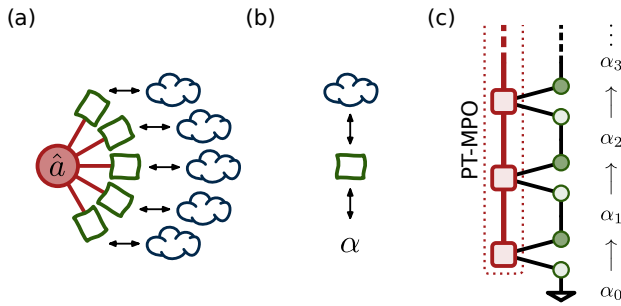


FIG. 12. (a) Schematic for the many-to-one network of a central boson model: each emitter (green squares) couples to a single bosonic mode $\hat{a}$ (red circle) as well as a harmonic environment (blue clouds). (b) Mean-field theory reduction to a single emitter interacting with a classical field $\alpha = \langle \hat{a} \rangle$. (c) Tensor network for the PT-MPO method with concurrent dynamics of the cavity field. The field α_i at $t = t_i$ is used in Eq. (16) to construct the system propagators at that time. Once the molecular system has been evolved to $t = t_{i+1}$, the resulting expectation $\langle \hat{\sigma}^- \rangle_{i+i}$ is used to integrate the field $\alpha_i \rightarrow \alpha_{i+1}$ according to Eq. (17).

F. Mean-field open quantum systems

A second type of many-body open quantum system that can be addressed in OQuPy are models with many-to-one or star-like topologies, see Fig. 12(a). We discuss this scenario by an example of a central boson model as was recently used to study the non-Markovian dynamics of organic polaritons²⁹. Here, a large number N of organic emitters couple to independent structured environments as well as a common photon mode (with bosonic operator $\hat{a}$). The emitters are modeled as two-level systems (Pauli matrices $\hat{\sigma}_n^\alpha$) and the light-matter dynamics is governed by

$$\hat{H}_S = \omega_c \hat{a}^\dagger \hat{a} + \sum_{n=1}^N \left[\frac{\omega_0}{2} \hat{\sigma}_n^z + \frac{\Omega}{2\sqrt{N}} (\hat{a} \hat{\sigma}_n^+ + \hat{a}^\dagger \hat{\sigma}_n^-) \right], \quad (13)$$

where ω_c and ω_0 are the cavity and emitter frequencies, and Ω is the collective light-matter coupling strength. Each emitter is also coupled to a harmonic bath,

$$\hat{H}_{IE}^{(n)} = \sum_j \left[\nu_j \hat{b}_{j,n}^\dagger \hat{b}_{j,n} + \frac{\xi_j}{2} (\hat{b}_{j,n} + \hat{b}_{j,n}^\dagger) \hat{\sigma}_n^z \right]. \quad (14)$$

This captures the local vibrational environment of organic molecules, which may typically be a low frequency continuum of modes described by, for example, an Ohmic spectral density $J(\nu)$. In addition, there are cavity losses with rate κ and individual pumping $\Gamma_\uparrow$ and dissipation $\Gamma_\downarrow$ of the emitters; these can be included as incoherent processes under the Markovian approximation.

In order to solve the many-molecule-cavity dynamics, we use mean-field theory to reduce the dimension of the

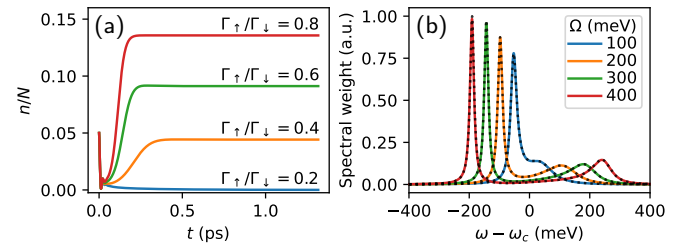


FIG. 13. (a) Scaled photon number $n/N = |\langle \hat{a} \rangle|^2/N$ below ($\Gamma_\uparrow = 0.2\Gamma_\downarrow$) and above ($\Gamma_\uparrow \geq 0.4\Gamma_\downarrow$) a lasing transition at $\Omega = 200$ meV, $T = 300$ K and $\Delta = \omega_c - \omega_0 = -20$ meV²⁹. Dynamics were calculated from an initial state with the spin down and a small number of photons $n_0/N = 0.05$. (b) Spectral weight (absorption) when $\Gamma_\uparrow = 0$. At each light-matter coupling, an analytic result¹⁰¹ is shown as a dotted line, and the result²⁹ of a mean-field PT-TEMPO calculation as a solid line. The PT-TEMPO numerical parameters used were $t_{\text{cut}} = 0.1$ ps, $dt = 4.0 \times 10^{-4}$ ps, $\text{epsrel} = 5.0 \times 10^{-12}$.

problem. In this approach, one neglects correlations in the many-body state, i.e. one assumes a product

$$\rho = \rho_a \otimes \bigotimes_{n=1}^N \rho_n, \quad (15)$$

where ρ_a and ρ_n are reduced density matrices for the cavity and n^{th} emitter, respectively. This reduces the problem to the coupled dynamics of the molecular mean-field Hamiltonian²⁹

$$\hat{H}_{\text{MF}} = \frac{\omega_0}{2} \hat{\sigma}^z + \frac{\Omega}{2\sqrt{N}} (\langle \hat{a} \rangle \hat{\sigma}^+ + \langle \hat{a} \rangle^* \hat{\sigma}^-), \quad (16)$$

combined with evolution of the field expectation

$$\partial_t \langle \hat{a} \rangle = -(i\omega_c + \kappa) \langle \hat{a} \rangle - i \frac{\Omega\sqrt{N}}{2} \langle \hat{\sigma}^- \rangle. \quad (17)$$

Here $\langle \hat{\sigma}^- \rangle = \langle \hat{\sigma}_n^- \rangle$ is the average of any of the identical spins.

Thus, by propagating a *single* spin with $\hat{H}_{\text{MF}}$ subject to the vibrational environment and individual pump and dissipation described above, one can effectively simulate the N -molecule system using the TEMPO or PT-TEMPO method provided that at each time step the field $\langle \hat{a} \rangle$ is evolved according to Eq. (17), as schematically depicted in Fig. 12(c).

Figure 13(a) illustrates this method in calculating photon number dynamics for different pump ratios $\Gamma_\uparrow/\Gamma_\downarrow$. This can be used, for example, to determine the dependence on the threshold for organic lasing on cavity detuning $\omega_c - \omega_0$ and light-matter coupling strength Ω (see Ref. 29 for details). The maximal memory requirement for these calculations was 1.6 GiB.

To verify the numerics, we calculated the absorption spectrum (see Sec. III B) for the system without pumping

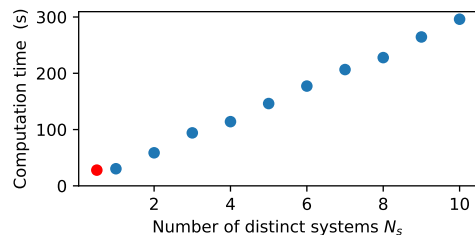


FIG. 14. Scaling of mean-field dynamics computation with number of distinct systems (emitters). A comparable calculation without mean-field integration is included in red: here a time-dependent Hamiltonian $\hat{H}(t) = \hat{H}_{\text{MF}}(f(t))$ was prescribed with $f(t) := \langle \hat{a} \rangle$ the solution for the field determined from the $N_s = 1$ mean-field dynamics computation.

($\Gamma_{\uparrow} = 0$), for which an analytical result is known¹⁰¹. Figure 13(b) shows excellent agreement between this result and that of a PT-TEMPO with mean-field calculation.

Note that in the above we took the emitters to be identical. The mean-field approach is, however, not limited to this case and so can describe e.g. multiple molecular species in a single cavity¹⁰². The cost is a separate PT-TEMPO (or TEMPO) computation for each distinct ρ_n . Figure 14 shows how the cost of calculating dynamics with mean-field evolution scales linearly with the number of separate systems ρ_n . If the systems have different environments, then one must also consider the resource cost of calculating separate process tensors for the systems beforehand, although that can be done in parallel and moreover does not change the complexity of the dynamics calculation.

For reference Fig. 14 includes a PT-TEMPO dynamics computation for a single time-dependent system *without* coupling to a mean field (red point). That this matches computation time for the corresponding $N_s = 1$ mean-field calculation demonstrates that integration of the mean-field via Eq. (17) adds negligible overhead to the overall computation.

Mean-field dynamics is accessible in OQuPy from an instance of the `MeanFieldSystem` class. This comprises one or more `TimeDependentSystemWithField` objects and a callable `field_eom`. The former describes systems (emitters) with different mean-field Hamiltonians (such as Eq. (16)), and the latter specifies the field equation of motion (e.g. Eq. (17)).

In this section we discussed a central boson model, for which the mean-field approximation can be shown¹⁰³ to be exact as $N \rightarrow \infty$. The same approach can be applied in other situations where mean-field theory is known to either be exact, or become a good approximation, in this limit. This includes central spin models¹⁰⁴ and models with all-to-all interactions¹⁰⁵ that arise, for example, in the context of waveguide and cavity-QED with cold atoms^{106,107}.

Note while the currently OQuPy is setup to address the central boson model specifically, i.e., the case of many independent satellite sites coupling to a common *bosonic*

mode, it is not limited to the many-body Hamiltonian Eq. (13): the satellite sites can be any system represented by a finite dimensional density matrix and interact with the cavity mode in any way; different realizations will yield different mean-field Hamiltonians Eq. (16) and field equations of motion Eq. (17). The generalization to the case where the central site is itself finite dimensional, such as a central spin model, is straightforward. Moreover, the same concept of mean-field decoupling can also be applied in other contexts, such as in lattice problems. In those cases mean-field theory is an approximation (due to finite connectivity), but can be a useful computational tool. This case is not currently implemented in OQuPy, but can be added by analogy to the existing approaches.

IV. OTHER RELATED METHODS

The methods and use cases presented in the previous Section III, all start from a pre-computed PT-MPO and are thus applicable to any type of environment for which such an PT-MPO can be obtained, regardless of the nature of the environment or the method with which the PT-MPO has been computed. In this section we give a brief introduction to further useful methods that are available in OQuPy, but are more specific as they require that the environment is a linearly coupled Gaussian bosonic bath.

A. Environment dynamics

A key step in PT construction, and many approaches to simulating open quantum systems, is tracing over the environment degrees of freedom. However, for non-Markovian processes these degrees of freedom by definition play a non-trivial role. In tracing them out we lose key insight into the interplay between system and environment. This could involve, for example, any engineered effect of the system on the environment such as with quantum thermal machines^{17,18}—where distinct thermodynamic effects beyond weak coupling can be observed^{108,109}—or the formation of system-environment bound states such as polarons^{110–113}.

The loss of access to environment information has previously put process tensor approaches at a disadvantage to methods such as the reaction coordinate or chain mappings^{114–116} which capture non-Markovian effects by augmenting the system with certain environmental degrees of freedom; here direct insight into environment dynamics can be gained. However, for the widely applicable case of linearly coupled bosons in a Gaussian initial state the bath correlations can be directly calculated from system correlation functions via an integral transform²⁴. This relation is general and can be used with any technique capable of generating system correlations. The PT-MPO approach is naturally placed to efficiently compute the

many-system correlations necessary for the calculation, as outlined in Section III B.

To demonstrate this approach we shall now compute the change in energy of the environment resolved by mode frequency for the biased spin-boson model with an Ohmic environment. The total Hamiltonian is thus again of the form of Eq (3), with the system Hamiltonian $\hat{H}_S = \frac{\epsilon}{2}\hat{\sigma}_z + \frac{\Omega}{2}\hat{\sigma}_x$. This could describe, for example, a semiconductor quantum dot driven by a laser with strength $\Omega = 1.0 \text{ ps}^{-1}$ and bias $\epsilon = 2.0 \text{ ps}^{-1}$ coupled to the phonon modes in the medium. The phonons are described by a bath at temperature $T = 1.0 \text{ ps}^{-1}$ ($= 7.64 \text{ K}$) with a spectral density $J(\omega)$ given by Eq. (5), where $\zeta = 1.0$, $\alpha = 0.05$, and $\omega_c = 10.0 \text{ ps}^{-1}$. The initial state of the system is $\rho_0^S = |\downarrow\rangle\langle\downarrow|$.

The environment is taken to be a continuum such that the coupling between the system and any single mode is infinitesimal. As such it makes no sense to investigate the change in energy of single modes. Instead we compute the change in energy over a range of modes with bandwidth δ . This is given by

$$\Delta Q(\omega, t) = \sum_k \int_{\omega-\delta/2}^{\omega+\delta/2} d\nu \delta(\nu - \omega_k) \omega_k \Delta n_k(t), \quad (18)$$

where $\Delta n_k(t) = \langle \hat{b}_k^\dagger(t) \hat{b}_k(t) \rangle - \langle \hat{b}_k^\dagger(0) \hat{b}_k(0) \rangle$ is the change in occupation of environment mode k from the initial state to time t ; for the results presented here we fix $\delta = 0.1 \text{ ps}^{-1}$. We plot this as a function of t and ω in Fig. 15(b) and see that most bath modes show a modest increase in energy, this is associated with the heating that arises from the interaction being quenched at $t = 0$. A notable departure from this trend occurs in the vicinity of modes resonant to the laser bias frequency ϵ which instead, after an initial heating, show a reduction in energy. This can be understood as the bath providing the additional energy lacking in the laser drive for the transition to be driven resonantly. This is highlighted in the upper panel where we focus on the change in energy in the vicinity of modes resonant with ϵ . Also visible here are long-lived oscillations in the heat exchanged, this feature is present in all frequency bands and is a result of the relaxation time of these bands increasing as the width of the band is reduced.

In OQuPy two-time bath correlation functions can be calculated using the `bath_dynamics` module. Currently this module consists of a single class, `TwoTimeBathCorrelations`, which has the capability of calculating any correlation function of the form

$$\langle \hat{b}_{k_2}^{(\dagger)}(t_2) \hat{b}_{k_1}^{(\dagger)}(t_1) \rangle, \quad (19)$$

where $t_2 \geq t_1$. All that is required is a `BaseSystem`, a `Bath`, and a `BaseProcessTensor`. The necessary system correlations, if not provided, are determined and computed automatically. A single set of these system correlations can be used to calculate a variety of bath correlation functions by simply adjusting the integral transform applied. By defining the two-time bath correlations

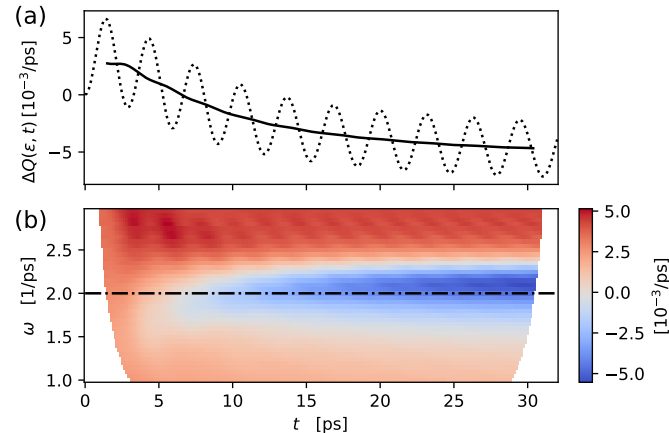


FIG. 15. Change of energy in environment modes after the quench (by coupling system and environment) at $t = 0$. (a) The change of energy in the mode interval $[\epsilon - \delta/2, \epsilon + \delta/2]$. The dotted line shows the raw result, while the solid line shows the period-averaged result. (b) The period-averaged change of energy in modes, $\Delta Q(\omega, t)$, as defined in Eq. (18). The lack of data at early and late times is a result of the period-averaging.

as a class, any system correlations computed for a given bath correlation can be stored within the class object and re-used in subsequent bath correlation calculations for a much faster computation.

Here we have highlighted how OQuPy can be used to calculate two-time bath correlation functions. In contrast to the other use cases, this relies on a derived relation assuming a specific form of interaction and bath; namely a linear coupling to a Gaussian bath of bosons. Although restrictive, this form of environment is ubiquitous in literature in the form of spin-boson models^{59,117}. In future it would be interesting to expand the `bath_dynamics` class to handle more general bath correlations. For example, computation of out-of-time-order correlators¹¹⁸, in particular one mixed between system and bath, could give a deeper insight into how information on the system is scrambled by a non-Markovian environment. In Ref. 24 a recipe is provided that gives the transform of system correlations necessary to compute this along with any other bath correlations desired.

B. Time evolving matrix product operator method

In previous sections we have seen how process tensors can be useful for studying many situations which commonly arise when simulating open quantum systems. An alternative approach for the case of Gaussian bosonic environments is to contract the tensor network shown in Fig. 3(a) in a different way to directly obtain the system dynamics instead of creating a PT-MPO. This is the original time evolving matrix product operator method (TEMPO) algorithm²³ as mentioned in Sec. II A. This loses the advantages of the process tensor approach for

being able to efficiently solve many problems which are related by having the same bath Hamiltonian. It can, however, be useful in cases where only a single simulation is required.

The key difference between TEMPO and PT-TEMPO is the order of contractions of the tensor network. In TEMPO we contract the tensor network row by row, giving an MPS after each contraction step. This MPS then gives access to the state of the system which can be recorded to calculate dynamics. As with PT-TEMPO, a memory cut-off can be used to limit how many history points are tracked. This means the MPS will grow in size until it reaches the cut-off where it will reach its maximum size. After this growth phase, a propagator can be defined as all subsequent layers are identical (for time independent Hamiltonians). After application of the propagator the MPS will grow one extra site to the right, and we contract the leftmost tensor to maintain the size of the MPS. This process can then be repeated until the desired time step is reached.

The TEMPO method can be accessed in OQuPy through the `tempo_compute()` function. This function takes a `Bath` (with $J(\omega)$, T , and $\hat{S}$) and `System` or `TimeDependentSystem` object (with $\hat{H}_S$) and returns a `Dynamics` object which encodes the evolution of the reduced system density matrix.

C Gibbs TEMPO

Although the functionality of OQuPy is primarily concerned with non-equilibrium systems, the tensor network methods that have been introduced can also be used to calculate equilibrium properties of open quantum systems. Specifically, if a system and Gaussian environment are collectively in a Gibbs state, i.e. in thermal equilibrium at some temperature $T = 1/\beta$, the TEMPO method discussed in the previous section can be used, with minor alterations, to calculate the reduced Gibbs state of the system (without resorting to weak coupling, or separability assumptions). The Gibbs TEMPO method²⁵ achieves this by performing an “evolution” along the imaginary time axis from time $t = 0$ to time $t = i\beta$.

The main difference between Gibbs TEMPO and standard TEMPO is that here the influence functional is written in terms of operators rather than superoperators, such that the legs of the influence tensors have dimension d , equal to that of the dimension of the system Hilbert space, rather than d^2 . If the goal is to simply calculate a thermal steady state of an open quantum system, this makes Gibbs TEMPO the more efficient option over standard TEMPO. Another difference is that the environment correlation functions do not decay on the imaginary time axis and hence no memory cutoff `tcut` can be used. Also, there is no initial state required as an input to Gibbs TEMPO.

In OQuPy Gibbs TEMPO is accessed through the `gibbs_tempo_compute()` function, which takes a

`System`, a `Bath`, and a `GibbsParameters` object, and returns the reduced Gibbs state in the form of a $d \times d$ array. The `GibbsParameters` class takes the number of imaginary time steps `num_steps` and a relative singular value cut-off threshold `epsrel`. The `Bath` object must have been initialized using a spectral density (i.e. either a `CustomSD` or `PowerLawSD` object).

V. FUTURE DIRECTIONS

The versatility of version 0.5 of OQuPy is the result of continuous extension of its functionality in the past couple of years. We intend to continue active development and maintenance of this open source project, and for this also welcome contributions from outside the current collaborations. We hence close in this section with a brief discussion of potential further extensions of OQuPy that we believe could be useful for research in fields related to non-Markovian open quantum systems.

As discussed above, the process tensor formalism is not restricted to a particular form of the environment. To fully leverage this fact in OQuPy, a useful extension would thus be to implement methods that yield PT-MPOs for environments other than Gaussian bosonic environments^{21,38,46–58}. A particularly versatile method would be the *automated compression of environments* (ACE) method²¹, which allows an efficient computation of PT-MPOs for a range of different environments, including anharmonic non-Gaussian bosonic, fermionic, and spin environments.

Beside additional methods for the creation of PT-MPOs, there are a series of methods for manipulating existing PT-MPOs that would be of practical value to develop, such as methods to combine, cut, and coarse-grain PT-MPOs. This would grant further flexibility to the construction and integration of process tensors in use cases such as above. Also, to facilitate the modeling of realistic environments, which may be highly structured, one can consider coarse-graining schemes¹¹⁹ for spectral densities. In OQuPy these could be implemented as an interface in which one inputs a complex spectral density—derived from experimental data or otherwise—and an effective, simplified spectral density is used to construct a corresponding `Correlations` object that is numerically tractable.

Another set of tools that would greatly enrich the functionality of OQuPy are methods that create and employ time translational invariant PT-MPOs, which encode an PT-MPO of arbitrary length in a single time translational unit cell^{55,56}, in analogy to infinite MPS in space¹²⁰. Such time translational invariant process tensors could be particularly useful for the direct study of non-equilibrium steady states of non-Markovian open quantum systems, with possibly interesting applications in the field of quantum thermodynamics.

Finally, we emphasize that the methods implemented in OQuPy and described in this article have been de-

veloped and employed in first studies only recently^{22–30}, with great scope for their application in many other problems that have previously been inaccessible. The primary purpose of OQuPy thus is, and continues to be, the low-barrier access to those numerical methods and to foster research in fields such as quantum chemistry, quantum technology, and quantum thermodynamics.

Acknowledgments

We thank the Unitary Fund community for their support. GEF acknowledges support from EPSRC (EP/L015110/1) and from ERC under grant agreement n.101053159 (RAVE). JB acknowledges support from the Laidlaw Foundation (Leadership and Research Programme scholarship). EPB acknowledges support from the Irish Research Council (GOIPG/2019/1871). DG acknowledges support from the QuantERA II Programme that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n.101017733 ("QuSiED"). PRE acknowledges support from Science Foundation Ireland (21-FFP-10142). EDCL acknowledges support from EPSRC (EP/T517938/1). RdW acknowledges support from EPSRC (EP/W524505/1). BWL and JK acknowledge support from EPSRC (EP/T014032/1).

Author contributions

The software project and writing of this article has been led by GEF and co-led by PFW. The TEMPO and PT-TEMPO method has been implemented by GEF and extended by EDCL and PK, building on precursor code by AS, DK, and PK. Multi-time correlations functionality has been contributed by RdW. The gradient functionality has been contributed by EPB, PRE, and EON. Multiple environment and environment dynamics functionality has been contributed by DG. The PT-TEBD method has been implemented by GEF, building on precursor code by DK. Mean-field functionality has been implemented by PFW, and extended by JB. Gibbs TEMPO has been implemented by AS. BWL and JK supervised and coordinated the scientific development. All code contributors have written the corresponding sections in the article.

Data availability

The data that support the findings of this study are openly available on Zenodo at <https://doi.org/10.5281/zenodo.4428316>, reference number 7243607.

¹C. Gardiner, P. Zoller, and P. Zoller, *Quantum Noise: A Handbook of Markovian and Non-Markovian Quantum Stochastic*

- Methods with Applications to Quantum Optics*, Springer Series in Synergetics (Springer, Berlin, 2000).
- ²H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, Oxford, 2002).
- ³L. Chirolli and G. Burkard, "Decoherence in solid-state qubits," *Adv. Phys.* **57**, 225–285 (2008).
- ⁴I. Wilson-Rae and A. Imamoglu, "Quantum dot cavity-QED in the presence of strong electron-phonon interactions," *Phys. Rev. B* **65**, 235311 (2002).
- ⁵C. Galland, A. Högele, H. E. Türeci, and A. Imamoglu, "Non-Markovian Decoherence of Localized Nanotube Excitons by Acoustic Phonons," *Phys. Rev. Lett.* **101**, 067402 (2008).
- ⁶A. J. Ramsay, T. M. Godden, S. J. Boyle, E. M. Gauger, A. Nazir, B. W. Lovett, A. M. Fox, and M. S. Skolnick, "Phonon-induced rabi-frequency renormalization of optically driven single InGaAs/GaAs quantum dots," *Phys. Rev. Lett.* **105**, 177402 (2010).
- ⁷C. Roy and S. Hughes, "Phonon-Dressed Mollow Triplet in the Regime of Cavity Quantum Electrodynamics: Excitation-Induced Dephasing and Nonperturbative Cavity Feeding Effects," *Phys. Rev. Lett.* **106**, 247403 (2011).
- ⁸I. de Vega and D. Alonso, "Dynamics of non-Markovian open quantum systems," *Rev. Mod. Phys.* **89**, 015001 (2017).
- ⁹L. Meng, Y. Zhang, X. Wan, C. Li, X. Zhang, Y. Wang, X. Ke, Z. Xiao, L. Ding, R. Xia, H.-L. Yip, Y. Cao, and Y. Chen, "Organic and solution-processed tandem solar cells with 17.3% efficiency," *Science* **361**, 1094–1098 (2018).
- ¹⁰J. 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. Thyryhaug, S. Westenhoff, and D. Zigmantas, "Quantum biology revisited," *Sci. Adv.* **6**, eaaz4888 (2020).
- ¹¹G. S. Engel, T. R. Calhoun, E. L. Read, T. K. Ahn, T. Mančal, Y. C. Cheng, R. E. Blankenship, and G. R. Fleming, "Evidence for wavelike energy transfer through quantum coherence in photosynthetic systems," *Nature* **446**, 782–786 (2007).
- ¹²L. Wang, M. A. Allodi, and G. S. Engel, "Quantum coherences reveal excited-state dynamics in biophysical systems," *Nat. Rev. Chem.* **3**, 477 (2019).
- ¹³L. Nicolin and D. Segal, "Quantum fluctuation theorem for heat exchange in the strong coupling regime," *Phys. Rev. B* **84**, 161414(R) (2011).
- ¹⁴M. Horodecki and J. Oppenheim, "Fundamental limitations for quantum and nanoscale thermodynamics," *Nat. Commun.* **4**, 2059 (2013).
- ¹⁵S. Vinjanampathy and J. Anders, "Quantum thermodynamics," *Contemp. Phys.* **57**, 545–579 (2016).
- ¹⁶U. Seifert, "First and Second Law of Thermodynamics at Strong Coupling," *Phys. Rev. Lett.* **116**, 020601 (2016).
- ¹⁷M. T. Mitchison, "Quantum thermal absorption machines: refrigerators, engines and clocks," *Contemp. Phys.* **60**, 164–187 (2019).
- ¹⁸F. Binder, L. A. Correa, C. Gogolin, J. Anders, and G. Adesso, *Thermodynamics in the Quantum Regime: Fundamental Aspects and New Directions*, Vol. 195 (Springer, Cham, 2018).
- ¹⁹P. Talkner and P. Hänggi, "Colloquium: Statistical mechanics and thermodynamics at strong coupling: Quantum and classical," *Rev. Mod. Phys.* **92**, 041002 (2020).
- ²⁰G. White, F. Pollock, L. Hollenberg, K. Modi, and C. Hill, "Non-Markovian Quantum Process Tomography," *PRX Quantum* **3**, 020344 (2022).
- ²¹M. Cygorek, M. Cosacchi, A. Vagov, V. M. Axt, B. W. Lovett, J. Keeling, and E. M. Gauger, "Simulation of open quantum systems by automated compression of arbitrary environments," *Nat. Phys.* **18**, 662–668 (2022).
- ²²A. Strathearn, B. W. Lovett, and P. Kirton, "Efficient real-time path integrals for non-Markovian spin-boson models," *New J. Phys.* **19**, 093009 (2017).

- 23 A. Strathearn, P. Kirton, D. Kilda, J. Keeling, and B. W. Lovett, "Efficient non-Markovian quantum dynamics using time-evolving matrix product operators," *Nat. Commun.* **9**, 3322 (2018).
- 24 D. Gribben, A. Strathearn, G. E. Fux, P. Kirton, and B. W. Lovett, "Using the environment to understand non-Markovian open quantum systems," *Quantum* **6**, 847 (2022).
- 25 Y.-F. Chiu, A. Strathearn, and J. Keeling, "Numerical evaluation and robustness of the quantum mean-force gibbs state," *Phys. Rev. A* **106**, 012204 (2022).
- 26 G. E. Fux, E. P. Butler, P. R. Eastham, B. W. Lovett, and J. Keeling, "Efficient Exploration of Hamiltonian Parameter Space for Optimal Control of Non-Markovian Open Quantum Systems," *Phys. Rev. Lett.* **126**, 200401 (2021).
- 27 D. Gribben, D. M. Rouse, J. Iles-Smith, A. Strathearn, H. Maguire, P. Kirton, A. Nazir, E. M. Gauger, and B. W. Lovett, "Exact Dynamics of Nonadditive Environments in Non-Markovian Open Quantum Systems," *PRX Quantum* **3**, 010321 (2022).
- 28 G. E. Fux, D. Kilda, B. W. Lovett, and J. Keeling, "Tensor network simulation of chains of non-markovian open quantum systems," *Phys. Rev. Res.* **5**, 033078 (2023).
- 29 P. Fowler-Wright, B. W. Lovett, and J. Keeling, "Efficient Many-Body Non-Markovian Dynamics of Organic Polaritons," *Phys. Rev. Lett.* **129**, 173001 (2022).
- 30 E. P. Butler, G. E. Fux, C. Ortega-Taberner, B. W. Lovett, J. Keeling, and P. R. Eastham, "Optimizing Performance of Quantum Operations with Non-Markovian Decoherence: The Tortoise or the Hare?" *Phys. Rev. Lett.* **132**, 060401 (2024).
- 31 F. A. Pollock, C. Rodríguez-Rosario, T. Frauenheim, M. Paternostro, and K. Modi, "Non-Markovian quantum processes: Complete framework and efficient characterization," *Phys. Rev. A* **97**, 012127 (2018).
- 32 S. Milz and K. Modi, "Quantum Stochastic Processes and Quantum non-Markovian Phenomena," *PRX Quantum* **2**, 030201 (2021).
- 33 N. Makri and D. E. Makarov, "Tensor propagator for iterative quantum time evolution of reduced density matrices. I. Theory," *J. Chem. Phys.* **102**, 4600–4610 (1995).
- 34 N. Makri and D. E. Makarov, "Tensor propagator for iterative quantum time evolution of reduced density matrices. II. Numerical methodology," *J. Chem. Phys.* **102**, 4611–4618 (1995).
- 35 S. Kundu and N. Makri, "PathSum: A C++ and Fortran suite of fully quantum mechanical real-time path integral methods for (multi-)system + bath dynamics," *J. Chem. Phys.* **158**, 224801 (2023).
- 36 A. Bose, "Quantumdynamics.jl: A modular approach to simulations of dynamics of open quantum systems," *The Journal of Chemical Physics* **158** (2023), 10.1063/5.0151483.
- 37 M. Cygorek and E. M. Gauger, "Ace: A general-purpose non-markovian open quantum systems simulation toolkit based on process tensors," (2024), arXiv:2405.19319 [quant-ph].
- 38 M. R. Jørgensen and F. A. Pollock, "Exploiting the causal tensor network structure of quantum processes to efficiently simulate non-Markovian path integrals," *Phys. Rev. Lett.* **123**, 240602 (2019).
- 39 G. Chiribella, G. M. D'Ariano, and P. Perinotti, "Quantum circuit architecture," *Phys. Rev. Lett.* **101**, 060401 (2008).
- 40 O. Oreshkov, F. Costa, and Č. Brukner, "Quantum correlations with no causal order," *Nat. Commun.* **3**, 1092 (2012).
- 41 M. A. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2010).
- 42 U. Schollwöck, "The density-matrix renormalization group in the age of matrix product states," *Ann. Phys. (N. Y.)* **326**, 96–192 (2011).
- 43 R. Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states," *Ann. Phys. (N. Y.)* **349**, 117–158 (2014).
- 44 F. Verstraete, J. J. García-Ripoll, and J. I. Cirac, "Matrix Product Density Operators: Simulation of Finite-Temperature and Dissipative Systems," *Phys. Rev. Lett.* **93**, 207204 (2004).
- 45 F. A. Pollock, C. Rodríguez-Rosario, T. Frauenheim, M. Paternostro, and K. Modi, "Operational Markov Condition for Quantum Processes," *Phys. Rev. Lett.* **120**, 040405 (2018).
- 46 M. C. Bañuls, M. B. Hastings, F. Verstraete, and J. I. Cirac, "Matrix Product States for Dynamical Simulation of Infinite Chains," *Phys. Rev. Lett.* **102**, 240603 (2009).
- 47 A. Müller-Hermes, J. I. Cirac, and M. C. Bañuls, "Tensor network techniques for the computation of dynamical observables in one-dimensional quantum spin systems," *New J. Phys.* **14**, 075003 (2012).
- 48 A. Strathearn, *Modelling Non-Markovian Quantum Systems Using Tensor Networks*, Springer Theses (Springer International Publishing, Cham, 2020).
- 49 A. Lerose, M. Sonner, and D. A. Abanin, "Influence matrix approach to many-body floquet dynamics," *Phys. Rev. X* **11**, 021040 (2021).
- 50 M. Sonner, A. Lerose, and D. A. Abanin, "Influence functional of many-body systems: Temporal entanglement and matrix-product state representation," *Ann. Phys. (N. Y.)* **435**, 168677 (2021).
- 51 E. Ye and G. K.-L. Chan, "Constructing tensor network influence functionals for general quantum dynamics," *J. Chem. Phys.* **155**, 044104 (2021).
- 52 J. Thoenness, M. Sonner, A. Lerose, and D. A. Abanin, "Efficient method for quantum impurity problems out of equilibrium," *Phys. Rev. B* **107**, L201115 (2023).
- 53 J. Thoenness, A. Lerose, and D. A. Abanin, "Nonequilibrium quantum impurity problems via matrix-product states in the temporal domain," *Phys. Rev. B* **107**, 195101 (2023).
- 54 N. Ng, G. Park, A. J. Millis, G. K.-L. Chan, and D. R. Reichman, "Real-time evolution of Anderson impurity models via tensor network influence functionals," *Phys. Rev. B* **107**, 125103 (2023).
- 55 M. Cygorek, J. Keeling, B. W. Lovett, and E. M. Gauger, "Sub-linear scaling in non-Markovian open quantum systems simulations," *Phys. Rev. X* **14**, 011010 (2024).
- 56 V. Link, H.-H. Tu, and W. T. Strunz, "Open quantum system dynamics from infinite tensor network contraction," *Phys. Rev. Lett.* **132**, 200403 (2024).
- 57 B. Kloss, J. Thoenness, M. Sonner, A. Lerose, M. T. Fishman, E. M. Stoudenmire, O. Parcollet, A. Georges, and D. A. Abanin, "Equilibrium quantum impurity problems via matrix product state encoding of the retarded action," *Phys. Rev. B* **108**, 205110 (2023).
- 58 N. Ng, G. Park, A. J. Millis, G. K.-L. Chan, and D. R. Reichman, "Real-time evolution of anderson impurity models via tensor network influence functionals," *Phys. Rev. B* **107**, 125103 (2023).
- 59 A. J. Leggett, S. Chakravarty, A. T. Dorsey, M. P. A. Fisher, A. Garg, and W. Zwerger, "Dynamics of the dissipative two-state system," *Rev. Mod. Phys.* **59**, 1–85 (1987).
- 60 R. P. Feynman and F. L. Vernon, "The theory of a general quantum system interacting with a linear dissipative system," *Ann. Phys. (N. Y.)* **24**, 118–173 (1963).
- 61 M. Cygorek, A. M. Barth, F. Ungar, A. Vagov, and V. M. Axt, "Nonlinear cavity feeding and unconventional photon statistics in solid-state cavity qed revealed by many-level real-time path-integral calculations," *Phys. Rev. B* **96**, 201201 (2017).
- 62 Y. Minoguchi, P. Kirton, and P. Rabl, "Environment-induced rabi oscillations in the optomechanical boson-boson model," (2019), arXiv:1904.02164 [quant-ph].
- 63 "OQuPy (open quantum systems in python) online documentation [https://oqupy.readthedocs.io/]," (2024).
- 64 The TEMPO collaboration, "OQuPy: A Python 3 package to efficiently compute non-Markovian open quantum systems." (2022).

- ⁶⁵R. de Wit, J. Keeling, B. W. Lovett, and A. W. Chin, "Process tensor approaches to modeling two-dimensional spectroscopy," (2024), arXiv:2402.15454 [quant-ph].
- ⁶⁶G. Guarnieri, A. Smirne, and B. Vacchini, "Quantum regression theorem and non-Markovianity of quantum dynamics," *Phys. Rev. A* **90**, 022110 (2014).
- ⁶⁷Although these inserted "control operations" are not CP-maps, this procedure yields nonetheless correct and numerically exact results.
- ⁶⁸B. Swingle, "Unscrambling the physics of out-of-time-order correlators," *Nat. Phys.* **14**, 988–990 (2018).
- ⁶⁹M. Cho, "Coherent Two-Dimensional Optical Spectroscopy," *Chem. Rev.* **108**, 1331–1418 (2008).
- ⁷⁰E. Collini, "2D Electronic Spectroscopic Techniques for Quantum Technology Applications," *J. Phys. Chem. C* **125**, 13096–13108 (2021).
- ⁷¹P. Virtanen *et al.* (SciPy 1.0 Contributors), "SciPy 1.0: Fundamental algorithms for scientific computing in python," *Nat. Methods* **17**, 261–272 (2020).
- ⁷²D. Basilewitsch, H. Yuan, and C. P. Koch, "Optimally controlled quantum discrimination and estimation," *Phys. Rev. Res.* **2**, 033396 (2020).
- ⁷³G. J. Hedley, A. Ruseckas, and I. D. Samuel, "Light harvesting for organic photovoltaics," *Chem. Rev.* **117**, 796–837 (2017).
- ⁷⁴A. Adronov and J. M. Fréchet, "Light-harvesting dendrimers," *Chem. Commun.* **18**, 1701–1710 (2000).
- ⁷⁵C. Felip-León, S. Díaz-Oltra, F. Galindo, and J. F. Miravet, "Chameleonic, light harvesting photonic gels based on orthogonal molecular fibrillization," *Chem. Mater.* **28**, 7964–7972 (2016).
- ⁷⁶E. Burzurí, J. O. Island, R. Díaz-Torres, A. Fursina, A. González-Campo, O. Roubeau, S. J. Teat, N. Aliaga-Alcalde, E. Ruiz, and H. S. van der Zant, "Sequential electron transport and vibrational excitations in an organic molecule coupled to few-layer graphene electrodes," *ACS Nano* **10**, 2521–2527 (2016).
- ⁷⁷G. D. Scholes, G. R. Fleming, A. Olaya-Castro, and R. van Grondelle, "Lessons from nature about solar light harvesting," *Nature Chem.* **3**, 763–774 (2011).
- ⁷⁸H. Maguire, J. Iles-Smith, and A. Nazir, "Environmental nonadditivity and Franck-Condon physics in nonequilibrium quantum systems," *Phys. Rev. Lett.* **123**, 093601 (2019).
- ⁷⁹J. O. Thomas, B. Limburg, J. K. Sowa, K. Willick, J. Baugh, G. A. D. Briggs, E. M. Gauger, H. L. Anderson, and J. A. Mol, "Understanding resonant charge transport through weakly coupled single-molecule junctions," *Nat. Commun.* **10**, 1–9 (2019).
- ⁸⁰J. K. Sowa, N. Lambert, T. Seideman, and E. M. Gauger, "Beyond Marcus theory and the Landauer-Büttiker approach in molecular junctions. II. A self-consistent Born approach," *J. Chem. Phys.* **152**, 064103 (2020).
- ⁸¹C. McConnell and A. Nazir, "Strong coupling in thermoelectric nanojunctions: A reaction coordinate framework," *New J. Phys.* **24**, 025002 (2022).
- ⁸²Y. Blanter and M. Büttiker, "Shot noise in mesoscopic conductors," *Phys. Rep.* **336**, 1–166 (2000).
- ⁸³N. Agraït, A. L. Yeyati, and J. M. van Ruitenbeek, "Quantum properties of atomic-sized conductors," *Phys. Rep.* **377**, 81–279 (2003).
- ⁸⁴F. Giazotto, T. T. Heikkilä, A. Luukanen, A. M. Savin, and J. P. Pekola, "Opportunities for mesoscopics in thermometry and refrigeration: Physics and applications," *Rev. Mod. Phys.* **78**, 217–274 (2006).
- ⁸⁵M. D. Losego, M. E. Grady, N. R. Sottos, D. G. Cahill, and P. V. Braun, "Effects of chemical bonding on heat transport across interfaces," *Nat. Mater.* **11**, 502–506 (2012).
- ⁸⁶J. R. Widawsky, W. Chen, H. Vázquez, T. Kim, R. Breslow, M. S. Hybertsen, and L. Venkataraman, "Length-Dependent Thermopower of Highly Conducting Au–C Bonded Single Molecule Junctions," *Nano Lett.* **13**, 2889–2894 (2013).
- ⁸⁷S. Bose, "Quantum Communication through an Unmodulated Spin Chain," *Phys. Rev. Lett.* **91**, 207901 (2003).
- ⁸⁸A. Wójcik, T. Łuczak, P. Kurzyński, A. Grudka, T. Gdala, and M. Bednarska, "Unmodulated spin chains as universal quantum wires," *Phys. Rev. A* **72**, 034303 (2005).
- ⁸⁹N. Lambert, Y. N. Chen, Y. C. Cheng, C. M. Li, G. Y. Chen, and F. Nori, "Quantum biology," *Nat. Phys.* **9**, 10–18 (2013).
- ⁹⁰H. N. Motlagh, J. O. Wrabl, J. Li, and V. J. Hilser, "The ensemble nature of allostery," *Nature* **508**, 331–339 (2014).
- ⁹¹G. Vidal, "Efficient Simulation of One-Dimensional Quantum Many-Body Systems," *Phys. Rev. Lett.* **93**, 040502 (2004).
- ⁹²N. Makri, "Communication: Modular path integral: Quantum dynamics via sequential necklace linking," *J. Chem. Phys.* **148**, 101101 (2018).
- ⁹³N. Makri, "Modular path integral methodology for real-time quantum dynamics," *J. Chem. Phys.* **149**, 214108 (2018).
- ⁹⁴S. Suzuki, H. Oshiyama, and N. Shibata, "Quantum annealing of pure and random ising chains coupled to a bosonic environment," *J. Phys. Soc. Japan* **88**, 061003 (2019).
- ⁹⁵A. Purkayastha, G. Guarnieri, S. Campbell, J. Prior, and J. Goold, "Periodically refreshed baths to simulate open quantum many-body dynamics," *Phys. Rev. B* **104**, 045417 (2021).
- ⁹⁶N. Makri, "Small matrix modular path integral: iterative quantum dynamics in space and time," *Phys. Chem. Chem. Phys.* **23**, 12537–12540 (2021).
- ⁹⁷S. Kundu and N. Makri, "Efficient matrix factorisation of the modular path integral for extended systems," *Mol. Phys.* **119**, e1797200 (2021).
- ⁹⁸S. Kundu and N. Makri, "Exciton-Vibration Dynamics in J-Aggregates of a Perylene Bisimide from Real-Time Path Integral Calculations," *J. Phys. Chem. C* **125**, 201–210 (2021).
- ⁹⁹S. Flannigan, F. Damanet, and A. J. Daley, "Many-Body Quantum State Diffusion for Non-Markovian Dynamics in Strongly Interacting Systems," *Phys. Rev. Lett.* **128**, 063601 (2022).
- ¹⁰⁰A. Bose and P. L. Walters, "A multisite decomposition of the tensor network path integrals," *J. Chem. Phys.* **156**, 024101 (2022).
- ¹⁰¹P. Kirtan and J. Keeling, "Thermalization and breakdown of thermalization in photon condensates," *Phys. Rev. A* **91**, 033826 (2025).
- ¹⁰²K. Georgiou, R. Jayaprakash, A. Othonos, and D. G. Lidzey, "Ultralong-Range Polariton-Assisted Energy Transfer in Organic Microcavities," *Angew. Chem. Int. Ed.* **60**, 16661–16667 (2021).
- ¹⁰³F. Carollo and I. Lesanovsky, "Exactness of Mean-Field Equations for Open Dicke Models with an Application to Pattern Retrieval Dynamics," *Phys. Rev. Lett.* **126**, 230601 (2021).
- ¹⁰⁴N. Makri, "The Linear Response Approximation and Its Lowest Order Corrections: An Influence Functional Approach," *J. Phys. Chem. B* **103**, 2823–2829 (1999).
- ¹⁰⁵E. Fiorelli, M. Müller, I. Lesanovsky, and F. Carollo, "Mean-field dynamics of open quantum systems with collective operator-valued rates: validity and application," *New J. Phys.* **25**, 083010 (2023).
- ¹⁰⁶M. Lewenstein, A. Sanpera, V. Ahufinger, B. Damski, A. Sen(De), and U. Sen, "Ultracold atomic gases in optical lattices: Mimicking condensed matter physics and beyond," *Adv. Phys.* **56**, 243–379 (2003).
- ¹⁰⁷D. E. Chang, J. S. Douglas, A. González-Tudela, C.-L. Hung, and H. J. Kimble, "Colloquium: Quantum matter built from nanoscopic lattices of atoms and photons," *Rev. Mod. Phys.* **90**, 031002 (2018).
- ¹⁰⁸M. Brenes, J. J. Mendoza-Arenas, A. Purkayastha, M. T. Mitchison, S. R. Clark, and J. Goold, "Tensor-Network Method to Simulate Strongly Interacting Quantum Thermal Machines," *Phys. Rev. X* **10**, 031040 (2020).
- ¹⁰⁹P. Strasberg, G. Schaller, N. Lambert, and T. Brandes, "Nonequilibrium thermodynamics in the strong coupling and non-Markovian regime based on a reaction coordinate mapping," *New J. Phys.* **18**, 073007 (2016).

This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.
PLEASE CITE THIS ARTICLE AS DOI: 10.1063/5.0225367

- ¹¹⁰G. D. Mahan, *Many-Particle Physics*, Physics of Solids and Liquids (Springer, New York, 2000).
- ¹¹¹R. Silbey and R. A. Harris, "Variational calculation of the dynamics of a two level system interacting with a bath," J. Chem. Phys. **80**, 2615–2617 (1984).
- ¹¹²R. A. Harris and R. Silbey, "Variational calculation of the tunneling system interacting with a heat bath. II. Dynamics of an asymmetric tunneling system," J. Chem. Phys. **83**, 1069–1074 (1985).
- ¹¹³J. T. Devreese and A. S. Alexandrov, "Fröhlich polaron and bipolaron: recent developments," Rep. Prog. Phys. **72**, 066501 (2009).
- ¹¹⁴J. Iles-Smith, N. Lambert, and A. Nazir, "Environmental dynamics, correlations, and the emergence of noncanonical equilibrium states in open quantum systems," Phys. Rev. A **90**, 032114 (2014).
- ¹¹⁵A. W. Chin, Á. Rivas, S. F. Huelga, and M. B. Plenio, "Exact mapping between system-reservoir quantum models and semi-infinite discrete chains using orthogonal polynomials," J. Math. Phys. **51**, 092109 (2010).
- ¹¹⁶J. Prior, A. W. Chin, S. F. Huelga, and M. B. Plenio, "Efficient Simulation of Strong System-Environment Interactions," Phys. Rev. Lett. **105**, 050404 (2010).
- ¹¹⁷U. Weiss, *Quantum dissipative systems*, Vol. 13 (World scientific, 2012).
- ¹¹⁸S. Xu and B. Swingle, "Scrambling dynamics and out-of-time-ordered correlators in quantum many-body systems," PRX Quantum **5**, 010201 (2024).
- ¹¹⁹N. Lorenzoni, N. Cho, J. Lim, D. Tamascelli, S. F. Huelga, and M. B. Plenio, "Systematic Coarse Graining of Environments for the Nonperturbative Simulation of Open Quantum Systems," Phys. Rev. Lett. **132**, 100403 (2024).
- ¹²⁰G. Vidal, "Classical Simulation of Infinite-Size Quantum Lattice Systems in One Spatial Dimension," Phys. Rev. Lett. **98**, 070201 (2007).