Weierstraß-Institut
für Angewandte Analysis und Stochastik

Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

Large-scale stochastic simulation of open quantum systems

Aaron Sander[1], Maximilian Fröhlich[2], Martin Eigel[2], Jens Eisert[3,4], Patrick Gelß[4,5], Michael Hintermüller[2,6], Richard M. Milbradt[1], Robert Wille[1,7,8], Christian B. Mendl[1]

submitted: February 13, 2025

[1] Technical University of Munich
Arcisstr. 21
80333 Munich, Germany
E-Mail: aaron.sander@tum.de
richard.milbradt@tum.de
christian.mendl@tum.de
robert.wille@tum.de

[2] Weierstrass Institute
Mohrenstr. 39
10117 Berlin, Germany
E-Mail: maximilian.froehlich@wias-berlin.de
martin.eigel@wias-berlin.de
michael.hintermueller@wias-berlin.de

[3] Freie Universität Berlin
Arnimallee 14
14195 Berlin, Germany
E-Mail: jense@zedat.fu-berlin.de

[4] Helmholtz-Zentrum Berlin
für Materialien und Energie
Hahn-Meitner-Platz 1
14109 Berlin, Germany

[5] Zuse Institute Berlin
Takustraße 7
14195 Berlin, Germany
E-Mail: gelss@zib.de

[6] Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin, Germany
E-Mail: hint@math.hu-berlin.de

[7] Software Competence Center
Hagenberg GmbH (SCCH)
Softwarepark 32a
4232 Hagenberg im Mühlkreis, Austria

[8] Quantum Software
Company GmbH
Münchener Str. 34
85748 Garching bei München, Germany

No. 3175
Berlin 2025

2020 *Mathematics Subject Classification.* 65N75, 65C30, 60H35, 65J10, 65C05.

Key words and phrases. Lindblad master equation, quantum simulation, Monte Carlo methods, tensor networks, matrix product states, time-dependent variational principle.

This work was funded by the European Union's Horizon 2020 Quantum Flagship innovation program Millenion (grant agreement No. 101114305) for joint work of FU Berlin and TU Munich, and the Einstein Research Unit on Quantum Devices for the Weierstrass Institute, the Zuse Institute and the FU Berlin. The full acknowledgements of all parties is written in the acknowledgements section.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Large-scale stochastic simulation of open quantum systems

Aaron Sander, Maximilian Fröhlich, Martin Eigel, Jens Eisert, Patrick Gelß, Michael Hintermüller, Richard M. Milbradt, Robert Wille, Christian B. Mendl

Abstract

Understanding the precise interaction mechanisms between quantum systems and their environment is crucial for advancing stable quantum technologies, designing reliable experimental frameworks, and building accurate models of real-world phenomena. However, simulating open quantum systems, which feature complex non-unitary dynamics, poses significant computational challenges that require innovative methods to overcome. In this work, we introduce the *tensor jump method* (TJM), a scalable, embarrassingly parallel algorithm for stochastically simulating large-scale open quantum systems, specifically Markovian dynamics captured by Lindbladians. This method is built on three core principles where, in particular, we extend the *Monte Carlo wave function* (MCWF) method to matrix product states, use a dynamic *time-dependent variational principle* (TDVP) to significantly reduce errors during time evolution, and introduce what we call a *sampling MPS* to drastically reduce the dependence on the simulation's time step size. We demonstrate that this method scales more effectively than previous methods and ensures convergence to the Lindbladian solution independent of system size, which we show both rigorously and numerically. Finally, we provide evidence of its utility by simulating Lindbladian dynamics of XXX Heisenberg models up to a thousand spins using a consumer-grade CPU. This work represents a significant step forward in the simulation of large-scale open quantum systems, with the potential to enable discoveries across various domains of quantum physics, particularly those where the environment plays a fundamental role, and to both dequantize and facilitate the development of more stable quantum hardware.

1 Introduction

Classical simulations are currently one of the most useful tools for comparing quantum theory with experimental results, as well as for providing benchmarks for the practical performance of quantum computers. Simulation results allow us to gain a deeper understanding of the underlying intricate physical mechanisms at work in complex quantum systems, as well as to build and scale more stable, reliable quantum computers and simulators. Quantum systems are notoriously difficult to simulate due to the exponential growth of parameters needed to represent larger systems. This growth leads to both increasing memory requirements to store exponentially many complex values and longer computational times to perform the operations needed to simulate their dynamics.

Tensor network methods, particularly *matrix product states* (MPS) [1, 2, 3], are famously at the forefront of classical tools for quantum simulation [4, 5] as they facilitate the reduction of memory usage and computational time compared to a full state vector representation, especially in the case of manageable entanglement growth [6, 7, 8, 9]. Rather than representing quantum systems, i.e., quantum states, as state vectors that grow exponentially in dimension with system size, MPSs enable the system to be split into a tensor train [3] with each tensor representing a local system. This makes it possible to not only store the state of the quantum system with substantially fewer parameters, but

DOI 10.20347/WIAS.PREPRINT.3175

also reduces many key components needed to simulate dynamics to local rather than global operations. This ansatz captures common correlation and entanglement patterns that present in quantum many-body systems well [10, 2]. This property has led to the development of the *time-evolving block decimation* (TEBD) algorithm [11] which is a critical simulation method for MPS dynamics. The caveat is that for increasingly entangled states, especially those generated during time evolution, the bond dimension between these tensors grows, with the upper bound becoming exponential as the system gets larger. The ability to truncate the bonds to much lower dimensions has been proven to be a successful approximation method that enables TEBD and MPS to be reliable as an accurate simulation method for local Hamiltonians [12], even for highly-entangled states. However, truncation can break symmetries, violate energy conservation, and poorly approximate long-range interactions on fixed bond dimension manifolds. The *time-dependent variational principle* (TDVP) [13, 14, 15, 16, 14] addresses these issues by evolving quantum states in time through provably optimal MPS approximations with lower bond dimensions. TDVP thus has the advantage of preserving energy conservation and symmetries by avoiding TEBD's truncation errors, as well as not being confined to local interactions.

Substantial progress has been made in simulating quantum systems, such as finding ground states and closed-system time evolution [3, 1, 6, 7, 12, 2, 8, 9, 13, 11, 10, 14, 15, 16, 14] and, more recently, quantum circuits [17, 18, 19]. However, simulating *open* quantum systems remains comparatively underexplored [20, 21, 22, 23, 24], even though environmental effects such as relaxation, dephasing, and thermal excitations are critical to understanding and mitigating noise in quantum technologies. Importantly, these interactions lead to unstable quantum computing architectures; in fact, quantum noise is the core antagonist in quantum technologies. In order to build better quantum devices, we need to precisely understand these noise processes to mitigate the error that they cause.

Lindblad master equations [25, 26] (or *Lindbladians*) have long been the standard theoretical framework for modeling dissipative dynamics. This approach is the most general form of a completely positive and trace-preserving quantum dynamical semi-group. It evolves the density matrix $\rho(t) \in \mathbb{C}^{d^L \times d^L}$ directly, where d is the physical dimension of one site and L is the number of sites. Storing $\rho(t)$ as a dense matrix requires $\mathcal{O}(d^{2L})$ computer memory and becomes intractable with growing system size. An alternative is the *Monte Carlo wave function method* (MCWF) [27, 28, 29, 30, 31, 32], which stochastically simulates individual quantum trajectories of pure states to approximate the corresponding density matrix. Although this approach partially circumvents the complexity of large density matrices, the computational cost still grows exponentially with system size as it relies on state vectors $|\Psi(t)\rangle \in \mathbb{C}^{d^L}$.

Early work towards achieving scalable approaches has focused on using matrix product operators to simulate open quantum systems [20, 21]. However, this can lead to a situation in which such tensor networks are no longer positive semi-definite and hence do not represent valid quantum states. Tensor network approaches that are not based on unravelings can be easily overburdened by the undecidability of checking positivity [33]. Alternative approaches use tensor networks that are positive semi-definite by construction [22]. In the literature, one also discriminates approaches that simulate time evolution from those that study stationary states [34]. It is known that the bond dimension can grow large during time evolution, while the stationary state only requires small bond dimensions [35].

Other forms of stochastic unravelings of open quantum systems using tensor network states have been explored in several works [36, 37, 38, 39, 40], e.g., using quantum state diffusion. Recent studies have also investigated MPS-based quantum jump methods for Hubbard-Holstein models with strong electron-phonon coupling [41] and photo-induced dynamics under vibrational baths [42], where they have used global Krylov methods and TDVP in combination with local subspace expansions to reduce the needed dimension of the bosonic Hilbert space. Our approach builds most directly on the seminal ideas of Ref. [37], yet surpasses that method substantially by (i) employing a *dynamic time-dependent variational principle* to tackle scalability and accuracy constraints for larger systems, and (ii) incorporating *higher-order Trotterization*, which leads to a clear separation of unitary and non-unitary time evolution of the state vector, resulting in a highly efficient evolution on an adaptive manifold while preserving the stability of the *dynamic time-dependent variational principle*. Unlike t-DMRG and TEBD approaches [43], which rely on truncations and lack proven convergence guarantees, our framework ensures *provably optimal unitary and non-unitary time evolution* of the state vector with an adaptive bond dimension. Moreover, the quantum jump-based unraveling here draws on foundational work from the 1990s in quantum optics [28, 27, 29, 30, 31, 32], serving not only as a calibration tool for quantum simulators but also as a step toward "dequantizing" challenging regimes where one expects a quantum advantage.

Concretely, we utilize the dynamic TDVP with an adaptive bond dimension, the stochastic *Monte Carlo wave function* (MCWF) approach, and what we term a *sampling MPS* to form a scalable simulation method for open quantum systems called the *tensor jump method* (TJM). We implement the MCWF procedure within a tensor network algorithm that uses single-site as well as two-site integration schemes to adapt the bond dimension before restricting it to a specified manifold. Employing a sampling MPS, derived from the higher-order Trotterization of the time-evolution operator, allows us to retrieve the state at intermediate times with substantially reduced sensitivity to the time-step size. We then show analytically and numerically that the TJM converges independently of system size, providing a scalable and robust new tool for open system simulations. Finally, we demonstrate its utility by extending its application well beyond the current computational state-of-the-art.

2 Simulation of open quantum systems

2.1 Lindbladian master equations

A large variety of processes in quantum physics can be captured by *quantum Markov processes*. On a formal level, they are described by the Lindblad master equation (or Lindbladian) [44, 45]

$$\frac{d}{dt}\rho = -i[H_0, \rho] + \sum_{m=1}^{k} \gamma_m \left(L_m \rho L_m^\dagger - \frac{1}{2}\{L_m^\dagger L_m, \rho\} \right). \tag{1}$$

This gives rise to a dynamical semi-group that generalizes the Schrödinger equation. The quantum state ρ and the Hamiltonian H_0 are Hermitian operators in the space of bounded operators $\mathcal{B}(\mathcal{H})$, acting on the Hilbert space of pure states of a quantum system $\mathcal{H}$. The first term $-i[H_0, \rho]$ corresponds to the unitary (closed/noise-free) time-evolution of ρ, where we have set $\hbar = 1$ to simplify notation. The summation captures the non-unitary dynamics of the system such that $\{L_m\}_{m=1}^k \subset \mathcal{B}(\mathcal{H})$ is a set of (non-unique) jump operators. These can be either Hermitian or non-Hermitian and correspond to noise processes in the system where $\{\gamma_m\}_{m=1}^k \subset \mathbb{R}_+$ is a set of coupling factors corresponding to the strength of each of these processes. These quantum jumps are defined as a sudden transition in the state of the system such as relaxation, dephasing, or excitation, which occur instantaneously, differentiating it from classical processes.

Directly computing the Lindbladian is one of the standard tools for exactly simulating open quantum systems [46]. However, its dependence on the dimension of the operator $\rho \in \mathbb{C}^{d^L \times d^L}$ limits the

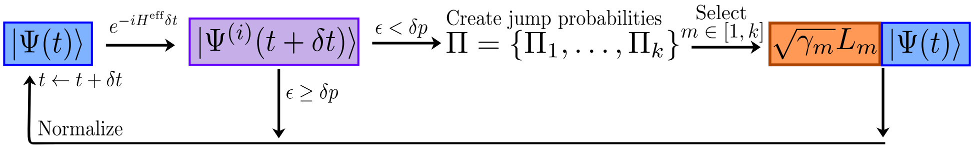
$|\Psi(t)\rangle$
$e^{-iH^{\text{eff}}\delta t}$
$|\Psi^{(i)}(t+\delta t)\rangle$
$\epsilon < \delta p$
Create jump probabilities
$\Pi = \{\Pi_1, \ldots, \Pi_k\}$
Select
$m \in [1,k]$
$\sqrt{\gamma_m} L_m$
$|\Psi(t)\rangle$
$t \leftarrow t + \delta t$
Normalize
$\epsilon \geq \delta p$

Figure 1: This figure shows the algorithm to perform the the Monte Carlo wave function (MCWF) method. An initial state vector $|\Psi(t)\rangle$ is time-evolved with a non-unitary time-evolution operator $e^{-iH^{\text{eff}\delta t}}$. The stochastic factor caused by the dissipation of this evolution $\delta p = 1 - \langle\Psi^{(i)}(t+\delta t)|\Psi^{(i)}(t+\delta t)\rangle$ is compared to a random number $\epsilon \in [0,1]$. If $\epsilon \geq \delta p$, the result of the stochastic process is the dissipated state vector $|\Psi^{(i)}(t+\delta t)\rangle$. Otherwise if $\epsilon < \delta p$, the probability of each jump occurring is calculating for all operators $L_1, \ldots, L_m$. According to this probability distribution, a jump is selected and applied to the state at the original time t such that the result is $L_j\,|\Psi(t)\rangle$. The result of the stochastic process is then normalized and the time updated. This is repeated until a given elapsed time $t = T$.

numerical computation of non-unitary dynamics to small system sizes This highlights the need for substantially more scalable simulation methods of large-scale open quantum systems.

2.2 Monte Carlo wave function method

To decouple the simulation of open quantum systems from the scaling of the density matrix, the Lindbladian can be simulated stochastically using the *Monte Carlo wave function* method (MCWF) [27, 29, 30, 31, 32]. The MCWF breaks the dynamics of an open quantum system into a series of trajectories that represent possible evolutions of a time-dependent pure quantum state vector $|\Psi(t)\rangle \in \mathbb{C}^{d^L}$. Each trajectory corresponds to a non-unitary time evolution of the quantum state followed by stochastic application of quantum jumps, which leads to a sudden, non-continuous shift in its evolution. By averaging over many of these trajectories, the full non-unitary dynamics of the system can be approximated without directly solving the Lindbladian. More formally speaking, it gives rise to a stochastic process in projective Hilbert space that, on average, reflects the quantum dynamical semi-group.

Using the same jump operators as in Eq. (1), a non-Hermitian Hamiltonian can be constructed as

$$H = H_0 + H_D, \tag{2}$$

consisting of the system Hamiltonian H_0 and dissipative Hamiltonian defined as

$$H_D = -\frac{i}{2}\sum_{m=1}^{k} \gamma_m L_m^\dagger L_m. \tag{3}$$

Formally, this non-Hermitian Hamiltonian defines a time-evolution operator

$$U(\delta t) = e^{-iH\delta t}, \tag{4}$$

which can be used to evolve the state vector as

$$|\Psi^{(i)}(t+\delta t)\rangle = e^{-iH\delta t}\,|\Psi(t)\rangle\,, \tag{5}$$

where H acts as an effective Hamiltonian in the time-dependent Schrödinger equation. The superscript (i) denotes the initial time-evolved state vector, which is not yet stochastically adjusted by the jump application process. Additionally, the dissipation caused by this operator does not preserve the norm of the state.

For the purpose of the MCWF derivation, the time evolution can be represented by the first-order approximation of the matrix exponential, where all $\mathcal{O}(\delta t^2)$ terms are dropped in the MCWF method [27]

$$\langle \Psi^{(i)}(t+\delta t)|\Psi^{(i)}(t+\delta t)\rangle = 1 - \delta p(t). \tag{6}$$

The denormalization $\delta p(t)$ can be used as a stochastic factor to determine if any jump has occurred in the given time step where it can be seen as a summation of individual stochastic factors corresponding to the denormalization caused by each noise process

$$\delta p_m(t) = \delta t \gamma_m \langle \Psi(t)|L_m^\dagger L_m|\Psi(t)\rangle \,,\; m = 1, \ldots, k, \tag{7}$$

such that

$$\delta p(t) = \sum_{m=1}^{k} \delta p_m(t). \tag{8}$$

A random number ϵ is then sampled uniformly from the interval $[0,1]$ and compared with the magnitude of δp. If $\epsilon \geq \delta p$, no jump occurs and the initial time-evolved state is normalized before moving onto the next time step

$$|\Psi(t+\delta t)\rangle = \frac{|\Psi^{(i)}(t+\delta t)\rangle}{\sqrt{\langle \Psi^{(i)}(t+\delta t)|\Psi^{(i)}(t+\delta t)\rangle}} = \frac{|\Psi^{(i)}(t+\delta t)\rangle}{\sqrt{1-\delta p(t)}}. \tag{9}$$

If $\epsilon < \delta p$, a probability distribution of the possible jumps at the given time is created by

$$\Pi(t) = \{\Pi_1(t), \ldots, \Pi_k(t)\} \tag{10}$$

with the normalized stochastic factors $\Pi_m(t) = \delta p_m(t)/\delta p(t)$ such that $\sum_{m=1}^{k} \Pi_m(t) = 1$. A jump operator L_m is then selected according to this probability distribution and applied directly to the pre-time-evolved quantum state

$$|\Psi(t+\delta t)\rangle = \frac{\sqrt{\gamma_m} L_m |\Psi(t)\rangle}{\sqrt{\langle \Psi(t)|L_m^\dagger \gamma_m L_m|\Psi(t)\rangle}} = \frac{\sqrt{\gamma_m} L_m |\Psi(t)\rangle}{\sqrt{\delta p_m(t)/\delta t}}, \tag{11}$$

simulating that the dissipative term has caused a jump during this time step. This methodology is then repeated until the desired elapsed time T is reached, resulting in a single quantum trajectory and a final state vector $|\Psi(T)\rangle$. This process, which can be seen in Fig. 1, gives rise to a stochastic process in projective Hilbert space called a piece-wise deterministic stochastic process in the limit $\delta t \to 0$.

This stochastic process is not only a classical Markovian stochastic process: In expectation, it exactly recovers the Markovian quantum process that is described by the dynamical semi-group generated by the master equation in Lindblad form

$$\rho(t) = \lim_{\delta t \to 0} \mathbb{E}\left(|\Psi_j(t)\rangle \langle \Psi_j(t)|\right). \tag{12}$$

For N trajectories, the quantum state $\rho(t)$ at time t is estimated as

$$\bar{\mu}(t) = \frac{1}{N}\sum_{j=1}^{N} |\Psi_j(t)\rangle \langle \Psi_j(t)|\,. \tag{13}$$

Details of this observation can be found in Sec. 9. Rather than directly calculating the quantum state, the trajectories of state vectors can be stored and manipulated individually to calculate relevant expectation values. While this does not solve the exponential scaling with system size, it does provide a computational advantage over exactly calculating the quantum state in form of the density operator $\rho(t)$.

3 Tensor jump method

While the MCWF improves the scalability of simulating open quantum systems in comparison to the exact Lindbladian computation, it still is bounded by the exponential scaling of state vectors. In conjunction with the success of MPS in simulating quantum systems, this provides an opportunity to extend the MCWF to a tensor network-based method which motivates this work. While directly solving the Lindbladian with MPDOs has been explored [22, 47], approaching this problem stochastically has only been examined for relatively small systems relatively small systems [36, 37, 38, 39, 40, 41, 42].

In this section, we introduce the components required for the *tensor jump method* (TJM). We first provide a general overview of the steps needed for the algorithm, without justification or explicit details, before walking the reader through the individual steps afterwards. We present the TJM with the aim to tackle the simulation of open quantum systems by designing every step to reduce the method's error and computational complexity as much as possible to maximize scalability.

3.1 General mindset

The general inspiration for the TJM is to transfer the MCWF to a highly efficient tensor network algorithm in which an MPS structure can be used to represent the trajectories, from which we can calculate the density operator or expectation values of observables. The stochastic time-evolution of one trajectory in the TJM consists of three main elements:

1 A dynamic TDVP $\mathcal{U}[\delta t]$.

2 A dissipative contraction $\mathcal{D}[\delta t]$.

3 A stochastic jump process $\mathcal{J}_\epsilon[\delta t]$.

The steps of the TJM described in this section are depicted in Fig. 2 and defined in Sec. 3.3–3.5.

We begin with an initial state vector $|\mathbf{\Psi}(0)\rangle$ at $t = 0$ encoded as an MPS. We wish to evolve this from some time $t \in [0, T]$ ($T = n\delta t$ ($n > 0$)) according to a Hamiltonian $\boldsymbol{H_0}$, where we use bold font whenever the Hamiltonian is encoded as an MPO, along with some noise processes described by the set of jump operators $\{L_m\}_{m=1}^k$. Using the above elements, we can express the time-evolution operator $U(T)$ of one trajectory of the TJM as

$$U(T) = \prod_{i=0}^{n} \mathcal{F}_{n-i}[\delta t]. \tag{14}$$

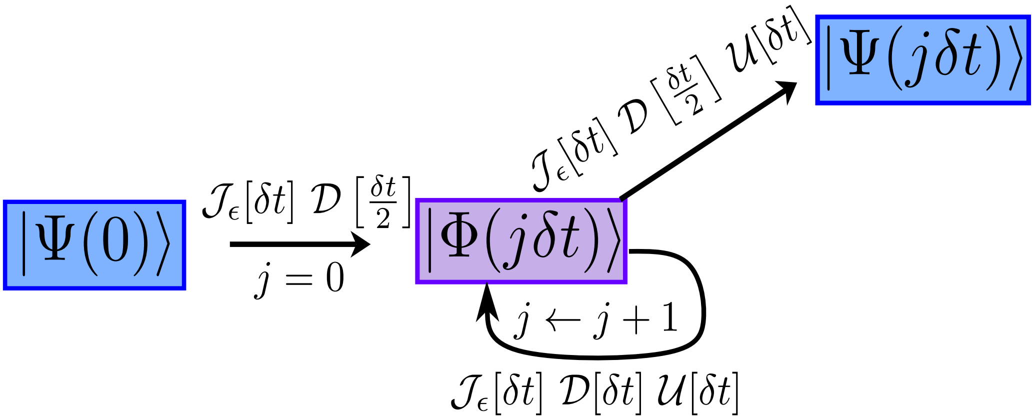
$|\Psi(0)\rangle$
$\mathcal{J}_\epsilon[\delta t]\ \mathcal{D}\left[\frac{\delta t}{2}\right]$
$j=0$
$|\Phi(j\delta t)\rangle$
$\mathcal{J}_\epsilon[\delta t]\ \mathcal{D}\left[\frac{\delta t}{2}\right]\ \mathcal{U}[\delta t]$
$|\Psi(j\delta t)\rangle$
$j \leftarrow j+1$
$\mathcal{J}_\epsilon[\delta t]\ \mathcal{D}[\delta t]\ \mathcal{U}[\delta t]$

Figure 2: This figure depicts the steps of the tensor jump method. First, the sampling MPS $|\Phi(j\delta t)\rangle$ is created from $|\Psi(0)\rangle$ by applying $\mathcal{F}_0[\delta t] = \mathcal{J}_\epsilon[\delta t]\ \mathcal{D}\left[\frac{\delta t}{2}\right]$ $(j=1)$. This is then evolved continuously by repeatedly applying $\mathcal{F}_j[\delta t] = \mathcal{J}_\epsilon[\delta t]\ \mathcal{D}[\delta t]\ \mathcal{U}[\delta t]$ $(j \leftarrow j+1)$. At any time, the quantum state vector $|\Psi(j\delta t)\rangle$ can be sampled by applying $\mathcal{F}_n[\delta t] = \mathcal{J}_\epsilon[\delta t]\ \mathcal{D}\left[\frac{\delta t}{2}\right]\ \mathcal{U}[\delta t]$ to the sampling MPS. The details of the unitary evolution $\mathcal{U}$, the dissipative contraction $\mathcal{D}$, and the tensorized jump process $\mathcal{J}_\epsilon$ can be found throughout Sec. 3. This process is analogous to the MCWF algorithm shown in Fig. 1.

which consists of n subfunctions corresponding to each time step

$$\mathcal{F}_j[\delta t] = \begin{cases} \mathcal{J}_\epsilon[\delta t]\ \mathcal{D}\left[\frac{\delta t}{2}\right]\ \mathcal{U}[\delta t], & j=n, \\ \mathcal{J}_\epsilon[\delta t]\ \mathcal{D}[\delta t]\ \mathcal{U}[\delta t], & 0<j<n, \\ \mathcal{J}_\epsilon[\delta t]\ \mathcal{D}\left[\frac{\delta t}{2}\right], & j=0. \end{cases} \tag{15}$$

This set of subfunctions follows from higher-order Trotterization used to reduce the time step error (see Sec. 3.2). However, this Trotterization causes the unitary evolution to lag behind the dissipative evolution by a half-time step, which is only corrected when the final operator $\mathcal{F}_n[\delta t]$ is applied.

To maintain the reduced time step error while being able to sample at each time step $t = 0, \delta t, 2\delta t, \ldots, T$ during a single simulation run, we introduce what we call a *sampling MPS* (denoted by $\mathbf{\Phi}$ while the quantum state itself is $\mathbf{\Psi}$.) This is initialized by the application of the first subfunction to the quantum state vector

$$|\mathbf{\Phi}(0)\rangle = \mathcal{F}_0[\delta t]\,|\mathbf{\Psi}(0)\rangle\,. \tag{16}$$

We use this to iterate through each successive time step

$$\big|\mathbf{\Phi}\big((j+1)\delta t\big)\big\rangle = \mathcal{F}_j[\delta t]\,|\mathbf{\Phi}(j\delta t)\rangle\,. \tag{17}$$

At any point during the evolution, we can retrieve the quantum state vector $|\mathbf{\Psi}(j\delta t)\rangle$ by applying the final function $\mathcal{F}_n$ to the sampling MPS as

$$|\mathbf{\Psi}(j\delta t)\rangle = \mathcal{F}_n[\delta t]\,|\mathbf{\Phi}(j\delta t)\rangle\,. \tag{18}$$

This allows us to sample at the desired time steps without compromising the reduction in time step error from applying the operators in this order.

We then repeat this time-evolution for N trajectories from which we can reconstruct the density operator in MPO format $\boldsymbol{\rho}(t)$ according to Eq. (13)

$$\boldsymbol{\rho}(t) = \frac{1}{N}\sum_{i=1}^{N} |\boldsymbol{\Psi}_i(t)\rangle \langle \boldsymbol{\Psi}_i(t)| , \tag{19}$$

or, more conveniently, we can calculate the expectation values of observables $\boldsymbol{O}$ (stored as tensors or an MPO denoted by the bold font) for each individual trajectory

$$\langle \boldsymbol{O}(t)\rangle = \frac{1}{N}\sum_{i=1}^{N} \langle \boldsymbol{\Psi}_i(t)| \boldsymbol{O} |\boldsymbol{\Psi}_i(t)\rangle . \tag{20}$$

This results in an embarassingly parallel process since each trajectory is independent and may be discarded after calculating the relevant expectation value.

3.2 Higher-order Trotterization

This section explains the steps to derive the subfunctions from Eq. (15) and provides our justification for doing so. We first define a generic non-Hermitian Hamiltonian created by the system Hamiltonian H_0 and the dissipative Hamiltonian H_D exactly as in Eq. (2)

$$H = H_0 + H_D. \tag{21}$$

From this, we create the non-unitary time-evolution operator that forms the basis of our simulation

$$U(\delta t) = e^{-i(H_0+H_D)\delta t}. \tag{22}$$

In many quantum simulation use cases, including the derivation of the MCWF in Eq. (4), this operator would be split according to the first summands of the matrix exponential definition or Suzuki-Trotter decomposition [48, 49, 50, 11]. However, higher-order splitting methods exist, which exhibit lower error [51, 52, 53, 54]. While this comes at the cost of computational complexity, we show that in this case Strang splitting [54] (or second-order Trotter splitting) can be utilized to reduce the time step error from $\mathcal{O}(\delta t^2) \to \mathcal{O}(\delta t^3)$ with negligible change in computational time.

Applying Strang splitting, the time-evolution operator becomes

$$U^{(i)}(\delta t) = e^{-iH_D\frac{\delta t}{2}} e^{-iH_0\delta t} e^{-iH_D\frac{\delta t}{2}} + \mathcal{O}(\delta t^3), \tag{23}$$

where superscript (i) denotes the initial time-evolution operator, which is not yet stochastically adjusted by the jump application process. These individual terms then define the dissipative operator

$$\mathcal{D}[\delta t] = e^{-iH_D\delta t} \tag{24}$$

and the unitary operator

$$\mathcal{U}[\delta t] = e^{-iH_0\delta t}. \tag{25}$$

For a time-evolution from $t \in [0, T]$ with terminal $T = n\delta t$ for n time steps, the time-evolution operator takes the form

$$\begin{aligned} U^{(i)}(T) &= \left(\mathcal{D}\left[\frac{\delta t}{2}\right] \mathcal{U}[\delta t]\, \mathcal{D}\left[\frac{\delta t}{2}\right]\right)^n \\ &= \mathcal{D}\left[\frac{\delta t}{2}\right] \mathcal{U}[\delta t] \left(\mathcal{D}[\delta t]\, \mathcal{U}[\delta t]\right)^{n-1} \mathcal{D}\left[\frac{\delta t}{2}\right] . \end{aligned} \tag{26}$$

Here, we have combined neighboring half time steps of dissipative operations, which is valid since H_D commutes with itself for any choice of jump operators.

Note that this then takes the same form as the functions in Eq. (15) although without the stochastic operator $\mathcal{J}_\epsilon[j\delta t]$, leading to the initial time-evolution functions

$$\mathcal{F}_j^{(i)}[\delta t] = \begin{cases} \mathcal{D}\left[\frac{\delta t}{2}\right] \mathcal{U}[\delta t], & j = n, \\ \\ \mathcal{D}[\delta t]\, \mathcal{U}[\delta t], & 0 < j < n, \\ \\ \mathcal{D}\left[\frac{\delta t}{2}\right], & j = 0, \end{cases} \tag{27}$$

where

$$\mathcal{F}_j[\delta t] = \mathcal{J}_\epsilon[\delta t]\, \mathcal{F}_j^{(i)}[\delta t],\ \forall j. \tag{28}$$

3.3 Dynamic TDVP

For the unitary time-evolution $\mathcal{U}$, we employ a *dynamic TDVP* method. The simulation begins with the *two-site TDVP* (2TDVP) which allows the bond dimensions of the MPS to grow naturally up to a maximum bond dimension χ_{max}. Once this threshold is reached, we switch to the one-site TDVP (1TDVP), confining the evolution to the current manifold, effectively capping the bond dimension and ensuring computational feasibility for the remainder of the simulation. This dynamic approach enables the necessary entanglement growth in the early stages while controlling the computational cost later on, making optimal use of available resources. In this section, we define a sweep as two half-sweeps, one from $\ell \in [1, \dots, L]$ and back for $\ell \in [L, \dots, 1]$. For a time step δt this leads to two half-sweeps of $\frac{\delta t}{2}$.

3.3.1 1TDVP and 2TDVP as tensor networks

To implement this approach, we express the state vector as a partitioning around one of its site tensors

$$|\mathbf{\Phi}\rangle = |\mathbf{\Phi}^L_{\ell-1}\rangle \otimes M_\ell \otimes |\mathbf{\Phi}^R_{\ell+1}\rangle\,. \tag{29}$$

By fixing the MPS to a mixed canonical form at site ℓ and applying the conjugate transpose of the partitioned single-site map $|\mathbf{\Phi}^L_{\ell-1}\rangle \otimes |\mathbf{\Phi}^R_{\ell+1}\rangle$ to each side of Eq. (83), the forward-evolving ODEs simplify to L local ODEs of the form

$$\frac{d}{dt} M_\ell(t) = -iH_\ell^{\text{eff}} M_\ell(t), \quad \ell = 1, \dots, L, \tag{30}$$

with a local effective Hamiltonian H_ℓ^{eff}, which dictates how to evolve each site tensor M_ℓ of the MPS. This process and the tensor network representation of H_ℓ^{eff} is visualized in Fig. 3.

Once H_ℓ^{eff} is computed, we matricize and exponentiate it using the Lanczos method [55]. After applying the Lanczos method, the site tensor is vectorized and updated according to the solution

$$M_\ell(t + \delta t) = e^{-iH_\ell^{\text{eff}}\delta t} M_\ell(t). \tag{31}$$

A QR decomposition is then applied to shift the orthogonality center from $\ell \mapsto \ell + 1$ resulting in $M_\ell = \tilde{M}_\ell C_\ell$, where $\tilde{M}_\ell$ replaces the previous site tensor and creates an updated MPS $|\tilde{\mathbf{\Phi}}\rangle$. The

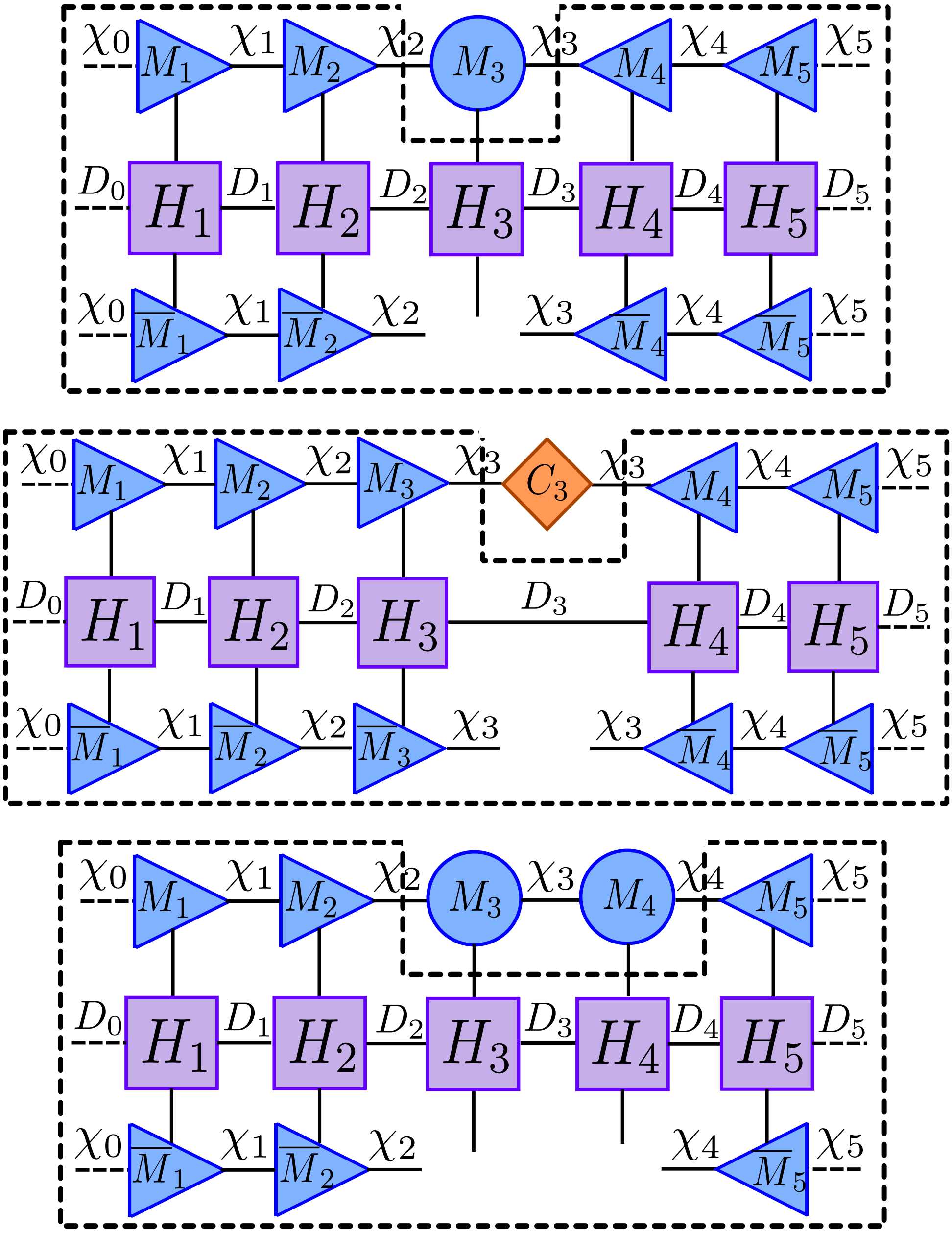
χ_0 M_1 χ_1 M_2 χ_2 M_3 χ_3 M_4 χ_4 M_5 χ_5
D_0 H_1 D_1 H_2 D_2 H_3 D_3 H_4 D_4 H_5 D_5
χ_0 $\overline{M}_1$ χ_1 $\overline{M}_2$ χ_2 χ_3 $\overline{M}_4$ χ_4 $\overline{M}_5$ χ_5
χ_0 M_1 χ_1 M_2 χ_2 M_3 χ_3 C_3 χ_3 M_4 χ_4 M_5 χ_5
D_0 H_1 D_1 H_2 D_2 H_3 D_3 H_4 D_4 H_5 D_5
χ_0 $\overline{M}_1$ χ_1 $\overline{M}_2$ χ_2 $\overline{M}_3$ χ_3 χ_3 $\overline{M}_4$ χ_4 $\overline{M}_5$ χ_5
χ_0 M_1 χ_1 M_2 χ_2 M_3 χ_3 M_4 χ_4 M_5 χ_5
D_0 H_1 D_1 H_2 D_2 H_3 D_3 H_4 D_4 H_5 D_5
χ_0 $\overline{M}_1$ χ_1 $\overline{M}_2$ χ_2 χ_4 $\overline{M}_5$ χ_5

Figure 3: This figure visualizes each step of the various forward- and backward-evolving equations of 1TDVP and 2TDVP indicated by the black dashed line. This shows the reduced practical form of the network caused by the MPS's mixed canonical form. The tensors surrounded by the dashed lines (corresponding to H^{eff}) are contracted, exponentiated with the Lanczos method, then applied to the remaining tensors to update them. **(Top)** 1TDVP forward-evolving and 2TDVP backward-evolving network where H_3^{eff} ($\tilde{H}_{3,4}^{\text{eff}}$) is a degree-6 tensor. **(Middle)** 1TDVP backward-evolving network where $\tilde{H}_3^{\text{eff}}$ is a degree-4 tensor. **(Bottom)** 2TDVP forward-evolving network where $H_{3,4}^{\text{eff}}$ is a degree-8 tensor.

bond tensor C_ℓ then evolves according to the simplified backward-evolving ODE, which is obtained by multiplying the conjugate transpose of $|\mathbf{\Phi}_\ell^L\rangle \otimes |\mathbf{\Phi}_{\ell+1}^R\rangle$ into each side of Eq. (84)

$$\frac{d}{dt}C_\ell(t) = +i\tilde{H}_\ell^{\text{eff}}C_\ell(t), \quad \ell = 1, \ldots, L-1$$

with an effective Hamiltonian $\tilde{H}_\ell^{\text{eff}}$ as shown in Fig. 3. Again, the Lanczos method is used to update the bond tensor

$$C_\ell(t+\delta t) = e^{+i\tilde{H}_\ell^{\text{eff}}\delta t}C_\ell(t), \tag{32}$$

after which it is contracted along its bond dimension with the neighboring site $C_\ell(t+\delta t)M_{\ell+1}(t)$ to continue the sweep.

In the 2TDVP scheme, the process is modified by extending Eq. (77) to sum over two neighboring sites and adjusting Eq. (78) and Eq. (79) to act on both sites ℓ and $\ell+1$

$$K_{\ell,\ell+1} = |\mathbf{\Phi}_{\ell-1}^L\rangle \langle\mathbf{\Phi}_{\ell-1}^L| \otimes I_\ell \otimes I_{\ell+1} \otimes |\mathbf{\Phi}_{\ell+2}^R\rangle \langle\mathbf{\Phi}_{\ell+2}^R|, \tag{33}$$

and

$$F_{\ell,\ell+1} = |\mathbf{\Phi}_{\ell-1}^L\rangle \langle\mathbf{\Phi}_{\ell-1}^L| \otimes I_\ell \otimes |\mathbf{\Phi}_{\ell+1}^R\rangle \langle\mathbf{\Phi}_{\ell+1}^R|. \tag{34}$$

This results in the equations to update two tensors simultaneously

$$M_{\ell,\ell+1}(t+\delta t) = e^{-iH_{\ell,\ell+1}^{\text{eff}}\delta t}M_{\ell,\ell+1}(t), \tag{35}$$

and

$$C_{\ell,\ell+1}(t+\delta t) = e^{+i\tilde{H}_{\ell,\ell+1}^{\text{eff}}\delta t}C_{\ell,\ell+1}(t), \tag{36}$$

where the effective Hamiltonians are defined as the tensor networks in Fig. 3. Note that the 1TDVP forward-evolution and the 2TDVP backward-evolution are functionally the same with a different prefactor in the exponentiation since $C_{\ell,\ell+1} = M_\ell$.

Compared to 1TDVP, this operation requires contraction of the bond between M_ℓ and $M_{\ell+1}$. After the time-evolution of the merged site tensors, a singular value decomposition (SVD) is applied with some threshold s_{max} such that the bond dimension χ_ℓ can be updated and allowed to grow to maintain a low error.

Practically, this results in a DMRG-like [9] algorithm since TDVP reduces to a spatial sweep across sites for all $\ell = 1, \ldots, L$, where at each site (or pair of sites) we alternate between a forward-evolving update to a given site tensor, followed by a backward-evolving update to its bond tensor.

3.3.2 Hybrid strategy

During the 1TDVP and 2TDVP sweeps, we compute the effective Hamiltonians using left and right environments which are updated and reused throughout the evolution [16]. Additionally, to effectively compute the matrix exponential, we apply the Lanczos method with a limited number of iterations [55], which significantly speeds up the computation of the exponential for large matrices, particularly when the bond dimensions grow large. Both of these are essential procedures to ensure that the TJM is scalable.

To summarize, the described algorithm allows us to define the unitary time-evolution operator $\mathcal{U}[\delta t]$ as a piece-wise conditional with error in $\mathcal{O}(\delta t^3)$ since TDVP is a second-order method [54], given by

$$\mathcal{U}[\delta t] = \begin{cases} \prod_{\ell=1}^{L-2} \left(e^{-iH^{\text{eff}}_{\ell,\ell+1}\delta t} e^{+i\tilde{H}^{\text{eff}}_{\ell,\ell+1}\delta t} \right) \; e^{-iH^{\text{eff}}_{L-1,L}\delta t} & \text{if } \forall \ell : \chi_\ell < \chi_{\text{max}}, \\ \prod_{\ell=1}^{L-1} \left(e^{-iH^{\text{eff}}_{\ell}\delta t} e^{+i\tilde{H}^{\text{eff}}_{\ell}\delta t} \right) \; e^{-iH^{\text{eff}}_{L}\delta t} & \text{if } \exists \ell : \chi_\ell \geq \chi_{\text{max}}. \end{cases} \tag{37}$$

3.4 Dissipative contraction

The dissipative operator $\mathcal{D}[\delta t]$ is created by exploiting the structure of the exponentiation of local jump operators in Eq. (24). This operation can be factorized into purely local operations due to the commutativity of the sums of single site operators. As a result, the dissipation term is equivalent to a single contraction of the dissipative operator $\mathcal{D}[\delta t]$ into the current MPS $|\mathbf{\Phi}(t)\rangle$. Additionally, this does not increase the bond dimension when applied to an MPS and the dissipative contraction is exact *without inducing errors*. This contraction is visualized in Fig. 4.

In this work, we focus on single-site jump operators $L_m \in \mathbb{C}^{d^L \times d^L}$ where each L_m is a tensor product of identity matrices $I \in \mathbb{C}^{d\times d}$ and a local non-identity operator $L_m^{[\ell]} \in \mathbb{C}^{d\times d}$ acting on some site $\ell = 1, \ldots, L$. Specifically, for each $m = 1, \ldots, k$, the operator L_m can be written as

$$L_m = I^{\otimes(\ell-1)} \otimes L_m^{[\ell]} \otimes I^{\otimes(L-\ell+1)} = I_{\backslash \ell} \otimes L_m^{[\ell]}, \tag{38}$$

where $L_m^{[\ell]}$ acts on the ℓ^{th} site and $I_{\backslash\ell}$ denotes the identity operator acting on all sites except ℓ. This allows the dissipative Hamiltonian to be localized site-wise

$$\begin{aligned} H_D &= -\frac{i}{2} \sum_{m=1}^{k} \gamma_m L_m^\dagger L_m \\ &= -\frac{i}{2} \sum_{\ell=1}^{L} \Big[\sum_{j \in S(\ell)} \gamma_j (I_{\backslash\ell} \otimes L_j^{[\ell]\dagger} L_j^{[\ell]}) \Big], \end{aligned} \tag{39}$$

where $S(\ell) \subseteq [1, \ldots, k]$ is the set of indices for the jump operators in $\{L_m\}_{m=1}^k$ for which there is a non-identity term at site ℓ. When exponentiated, this results in

$$\begin{aligned} \mathcal{D}[\delta t] &= e^{-i\left(-\frac{i}{2}\sum_{\ell=1}^{L}\left[\sum_{j\in S(\ell)} \gamma_j (I_{\backslash\ell}\otimes L_j^{[\ell]\dagger} L_j^{[\ell]}\right]\right)\delta t} \\ &= \prod_{\ell=1}^{L} e^{-\frac{1}{2}\sum_{j\in S(\ell)} \gamma_j (I_{\backslash\ell}\otimes L_j^{[\ell]\dagger} L_j^{[\ell]})\delta t} \\ &= \prod_{\ell=1}^{L} e^{I_{\backslash\ell}\otimes(-\frac{1}{2}\delta t \sum_{j\in S(\ell)} \gamma_j L_j^{[\ell]\dagger} L_j^{[\ell]})} \\ &= \bigotimes_{\ell=1}^{L} e^{-\frac{1}{2}\delta t \sum_{j\in S(\ell)} \gamma_j L_j^{[\ell]\dagger} L_j^{[\ell]}} = \bigotimes_{\ell=1}^{L} \mathcal{D}_\ell[\delta t]. \end{aligned} \tag{40}$$

The resulting operator corresponds to a factorization of site tensors where $\mathcal{D}_\ell[\delta t] \in \mathbb{C}^{d\times d}$ for $\ell \in [1, \ldots, L]$.

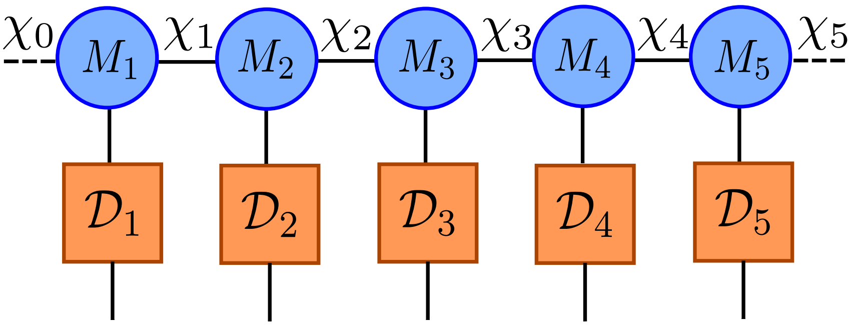

Figure 4: This figure explains the contraction of the factorized dissipative MPO $\mathcal{D}$ (local matrices) into an MPS $|\Psi\rangle$. Each local tensor is equivalent to the exponentiation of the local jump operators according to $\mathcal{D}_\ell = e^{-\frac{1}{2}\delta t \sum_{j\in S(\ell)} L_j^{[\ell]\dagger} L_j^{[\ell]}}$. This process does not change the bond dimension of the MPS and does not require a mixed canonical form.

3.5 Stochastic jump process

Following each $\mathcal{F}_j^{(i)}$, we perform the jump process $\mathcal{J}_\epsilon[\delta t]$ for determining if (and how) jump operators should be applied to the MPS. $\mathcal{J}_\epsilon[\delta t]$ is the value of a stochastic function $\mathcal{J}$ that maps a randomly selected $\epsilon \in [0,1]$ in combination with a time step size δt to an operator. This operator is either the identity operator if no jump occurs or a single site-jump operator $L_m^{[\ell]}, m = 1,\dots,k$ in the case of a jump,. This operator, encoded as a single-site tensor, is then contracted into the sampling MPS $|\mathbf{\Phi}(t)\rangle$.

We apply the jump process $\mathcal{J}_\epsilon[\delta t]$ following each operation defined in Eq. (15). This means that first the initial time-evolved state vector from $t \mapsto t+\delta t$ is created

$$|\mathbf{\Phi}^{(i)}(t+\delta t)\rangle = \mathcal{F}_j^{(i)}[\delta t]\,|\mathbf{\Phi}(t)\rangle\,.$$

From this we create

$$|\mathbf{\Phi}(t+\delta t)\rangle = \mathcal{J}_\epsilon[\delta t]\,|\mathbf{\Phi}^{(i)}(t+\delta t)\rangle\,, \tag{41}$$

where the operator $\mathcal{J}_\epsilon[\delta t]$ acts as described in the following.

First, the overall stochastic factor $\delta p(t)$ is determined as in the MCWF by taking the inner product of this state, where we begin to sweep across the state to maintain a mixed canonical form following the dissipative contraction. This reduces the calculation to contracting the final tensor of the MPS with itself, i.e.,

$$\begin{aligned}\delta p &= 1 - \langle \mathbf{\Phi}^{(i)}(t+\delta t)|\mathbf{\Phi}^{(i)}(t+\delta t)\rangle \\ &= 1 - \sum_{\sigma_L=1}^{d} \sum_{a_{L-1},a_L=1}^{\chi_{L-1},\chi_L} M_L^{\sigma_L,a_{L-1},a_L}\overline{M}_L^{\sigma_L,a_{L-1},a_L}.\end{aligned} \tag{42}$$

In contrast to the MCWF, we do not use a first-order approximation of $e^{-iH\delta t}$ to calculate $\delta p(t)$ since the time-evolution has been carried out by the TDVP projectors and the dissipative contraction. Next, $\epsilon \in [0,1]$ is sampled uniformly, which subsequently leads to two possible paths.

3.5.1 No jump occurs

If $\epsilon \geq \delta p$, we normalize $|\mathbf{\Phi}^{(i)}(t+\delta t)\rangle$. In this case, the dissipative contraction itself represents the noisy interactions from time $t \mapsto t+\delta t$.

3.5.2 A jump occurs

If $\epsilon < \delta p$, we generate the probability distribution of all possible jump operators using the initial time-evolved state

$$\Pi_m(t) = \frac{\delta t \gamma_m}{\delta p(t)} \langle \mathbf{\Phi}^{(i)}(t+\delta t)| L_m^\dagger L_m |\mathbf{\Phi}^{(i)}(t+\delta t)\rangle \tag{43}$$

for $m = 1, \ldots, k$. At any given site ℓ, we can calculate the probability $\Pi_j(t)$ for $j \in S(\ell)$ according to $\langle \mathbf{\Phi}^{(i)}(t+\delta t)| L_j^\dagger L_j |\mathbf{\Phi}^{(i)}(t+\delta t)\rangle$ for the relevant jump operators L_j. When performed in a half-sweep across $\ell = [1, \ldots, L]$ where at each site the MPS is fixed into its mixed canonical form, the probability is calculated by a contraction of the jump operator and the site tensor M_ℓ

$$N_\ell^{\sigma'_\ell, a_\ell, a_{\ell-1}} = \sum_{\sigma_\ell=1}^{d} L_m^{\sigma'_\ell, \sigma_\ell} M_\ell^{\sigma_\ell, a_{\ell-1}, a_\ell}. \tag{44}$$

Note that this is not directly updating the MPS but rather using the current state of its tensors to calculate the stochastic factors. Then, the inner product of this new tensor with itself is evaluated while scaling it accordingly as done in the MCWF

$$\Pi_m(t) = \frac{\delta t \gamma_m}{\delta p(t)} \sum_{\sigma_\ell=1}^{d} \sum_{a_{\ell-1}, a_\ell=1}^{\chi_{\ell-1}, \chi_\ell} N_\ell^{\sigma'_\ell, a_{\ell-1}, a_\ell} \overline{N}_\ell^{\sigma'_\ell, a_\ell, a_{\ell-1}}. \tag{45}$$

This is repeated for $j \in S(\ell)$ until all jump probabilities at site ℓ are calculated. We then move to the next site $\ell \mapsto \ell + 1$ performing the same process until $\ell = L$ and the half-sweep is complete.

This yields the probability distribution $\Pi(t) = \{\Pi_m(t)\}_{m=1}^{k}$ from which we can randomly select a jump operator L_m to apply to $|\mathbf{\Phi}^{(i)}(t+\delta t)\rangle$. This is achieved by multiplying L_m into the relevant site tensor M_ℓ with elements

$$\tilde{M}_\ell^{\sigma_\ell, \alpha_\ell, \alpha_{\ell-1}} := \sqrt{\gamma_m} \sum_{\sigma'_\ell=1}^{d} L_m^{[\ell]\ \sigma'_\ell, \sigma_\ell} M_\ell^{\sigma'_\ell, \alpha_\ell, \alpha_{\ell-1}}.$$

The result is the updated MPS

$$|\mathbf{\Phi}(t+\delta t)\rangle = \sum_{\sigma_1, \ldots, \sigma_L=1}^{d} M_1^{\sigma_1} \ldots \tilde{M}_\ell^{\sigma_\ell} \ldots M_L^{\sigma_L} |\sigma_1, \ldots, \sigma_L\rangle.$$

The state is then normalized before moving onto the next time step. Note that this is a fundamental departure from the MCWF in which the jump is applied to the state at the previous time t.

3.6 Algorithm

With all necessary tensor network methods established, we can now combine the procedures from the previous sections to construct the complete TJM algorithm.

3.6.1 Initialization

The TJM requires the following components:

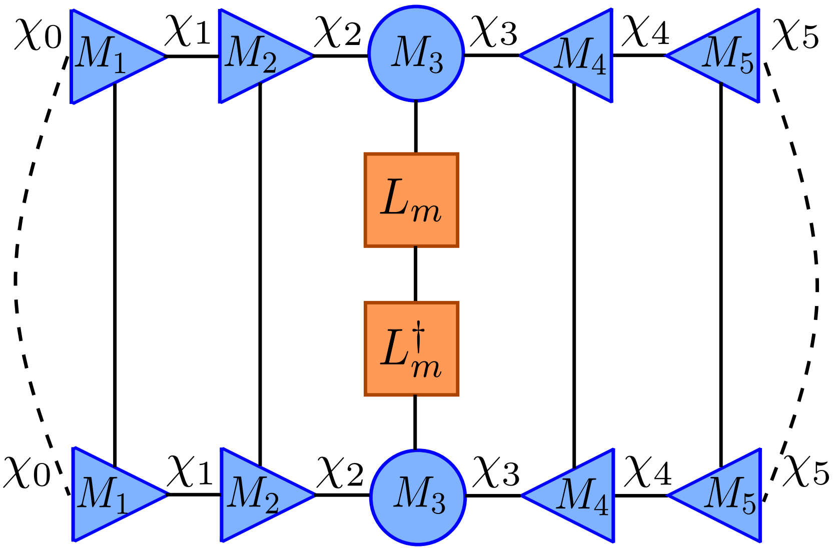

Figure 5: This figure visualizes the tensor network required to calculate δp_m for a given jump operator L_m, which is equivalent to the expectation value of $L_m^\dagger L_m$. The mixed canonical form reduces the computational effort drastically since all site tensors to the left and the right of the targeted site can be ignored as they reduce to identity operations. This leads to performing the calculation of the probability distribution $\Pi(t)$ by sweeping across the network where at each site we place the MPS in mixed canonical form and calculate the probability of jumps at that site $L_j, j \in S(\ell)$.

1 $|\mathbf{\Psi}(0)\rangle$: Initial quantum state vector, represented as an MPS.

2 $\boldsymbol{H_0}$: Hermitian system Hamiltonian, represented as an MPO.

3 $\{L_m\}_{m=1}^k, \{\gamma_m\}_{m=1}^k$: A set of single-site jump operators stored as matrices with their respective coupling factors.

4 δt: Time step size.

5 T: Total evolution time.

6 χ_{max}: Maximum allowed bond dimension.

7 N: Number of trajectories.

3.6.2 Time evolution of the sampling MPS

Once these components are defined, the noisy time evolution from $t \in [0, T]$ is performed by iterating through each time step using the operators described in Eq. (15). We first initialize the sampling MPS for the time evolution. The initial state $|\mathbf{\Psi}(0)\rangle$ is evolved using the operator $\mathcal{F}_0 = \mathcal{J}_\epsilon[\delta t]\, \mathcal{D}\left[\frac{\delta t}{2}\right]$, which includes a half-time step dissipative contraction and a stochastic jump process $\mathcal{J}_\epsilon[\delta t]$

$$|\mathbf{\Phi}(\delta t)\rangle = \mathcal{F}_0\, |\mathbf{\Psi}(0)\rangle\,. \tag{46}$$

The algorithm then evolves the system to each successive time step $t = j\delta t$ using the operators $\mathcal{F}_j = \mathcal{J}_\epsilon[\delta t]\, \mathcal{D}[\delta t]\, \mathcal{U}[\delta t]$ as seen in Eq. (18)

$$\left|\mathbf{\Phi}\big((j+1)\delta t\big)\right\rangle = \mathcal{F}_j\, |\mathbf{\Phi}(j\delta t)\rangle\,. \tag{47}$$

Each iteration involves the following.

- A full time step unitary operation $\mathcal{U}[\delta t]$ using the dynamic TDVP:
    - If any bond dimension $\chi_\ell < \chi_{\text{max}}$, 2TDVP is applied, allowing the bond dimensions to grow dynamically.
    - If the bond dimension has reached χ_{max}, 1TDVP is applied to constrain further growth and maintain computational efficiency.
- A full time step dissipative contraction $\mathcal{D}[\delta t]$ where the non-unitary part of the evolution is applied.
- A jump process to determine whether quantum jumps occur, including normalization of the state.

This process is repeated until the time evolution reaches terminal T.

3.6.3 Retrieving the quantum state

At any point during the time evolution, the quantum state can be retrieved by applying the final function $\mathcal{F}_n = \mathcal{J}_\epsilon[\delta t]\; \mathcal{D}\left[\frac{\delta t}{2}\right]\; \mathcal{U}[\delta t]$ as if the system has evolved to the stopping time

$$|\boldsymbol{\Psi}(\delta t)\rangle = \mathcal{F}_n \, |\boldsymbol{\Phi}(\delta t)\rangle \, . \tag{48}$$

The state is then normalized. This allows for the state to be inspected or observables to be calculated at any intermediate time without compromising the reduction in error from the Strang splitting. The above process is repeated from the beginning for each of the N trajectories, providing access to a compact storage of the density matrix and its evolution as well as the ability to calculate expectation values.

Method	**Time Evolution**	**Storage**	**Exp. Value**
Lindblad	$\mathcal{O}(nd^{6L})$	$\mathcal{O}(d^{2L})$	$\mathcal{O}(d^{6L})$
MCWF	$\mathcal{O}(Nnd^{3L})$	$\mathcal{O}(Nd^{L})$	$\mathcal{O}(Nd^{4L})$
MPDO	$\mathcal{O}(nLd^4 D_H^2 D_s^2)$	$\mathcal{O}(Ld^2 D_s^2)$	$\mathcal{O}(Ld^2 D_s^3)$
TJM	$\mathcal{O}(NnL\chi_{\text{max}}^3[dD + d^2])$	$\mathcal{O}(NLd\chi_{\text{max}}^2)$	$\mathcal{O}(NLdD\chi_{\text{max}}^3)$

Table 1: This table compares the complexities between each method, including the time to generate the time-evolution, store the final data structure, and calculate expectation values with the result. These are dependent on the physical dimension d, system size L, time steps n, trajectories N, MPS bond dimension χ_{max}, system MPO bond dimension D, density matrix bond dimension D_s, and Hamiltonian bond dimension D_H according to the W^{II} algorithm [56, 57]. Note that this assumes that all information is kept, despite the TJM being embarassingly parallel, where the individual trajectories could be used to calculate an expectation value and then be discarded in most practical contexts.

4 Computational complexity and convergence guarantees

While the TJM method proposed here is in practice highly functional and performs well, it can also be equipped with tight bounds concerning the computational and memory complexity as well as with convergence guarantees. This section is devoted to justifying this utility in approximating Lindbladian dynamics by analyzing the mathematical behavior of the TJM based on its convergence and error bounds. Since we assert that the TJM is highly-scalable compared to other methods, this analytical proof serves to lend credence to large-scale results, which may have no other method against which we can benchmark. These important points are discussed here, while substantial additional details are presented in the methods and appendix sections.

4.1 Computational effort

We derive and compare the computational and memory complexity of the exact calculation of the Lindblad equation, the MCWF, a Lindblad MPDO, and the TJM method in detail in Sec. 8.4–8.6 and Sec. 11. The results are summarized in Table 1, showcasing the highly beneficial and favorable scaling of the computational and memory complexities of the TJM method.

4.2 Monte Carlo convergence

The convergence of the TJM is stated in terms of the density matrix standard deviation, which we define as follows.

Definition 1 (Density matrix variance). *Let $\|\cdot\|$ be a matrix norm, and let $X \in \mathbb{C}^{n\times n}$ be a matrix-valued random variable defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where $\mathbb{P}$ is a probability measure. The variance of X with respect to the norm $\|\cdot\|$ is defined as*

$$\mathbb{V}[X] = \mathbb{E}\left[\|X - \mathbb{E}[X]\|^2\right], \tag{49}$$

where $\mathbb{E}[X]$ denotes the expectation of X. The expectation $\mathbb{E}[X]$ is computed entrywise, with each entry being the expectation according to the respective marginal distributions of the entries. Specifically, for each i, j,

$$\mathbb{E}[X]_{i,j} = \mathbb{E}_{\mathbb{P}_{i,j}}[x_{i,j}], \tag{50}$$

where $x_{i,j}$ is the (i,j)-th entry of the matrix X, and $\mathbb{P}_{i,j}$ is the marginal distribution of $x_{i,j}$ induced by $\mathbb{P}$. The expectation value of the norm of a matrix $\mathbb{E}[\|\cdot\|]$ is defined as the multidimensional integral over the function $\|\cdot\| : \mathbb{C}^{n\times n} \to \mathbb{R}$ according to its marginal distributions $\mathbb{P}_{i,j}$.

The standard deviation of X with respect to the norm $\|\cdot\|$ is then defined as

$$\sigma(X) = \sqrt{\mathbb{V}[X]} = \sqrt{\mathbb{E}\left[\|X - \mathbb{E}[X]\|^2\right]}. \tag{51}$$

For the proof of the convergence, we furthermore need additional properties of the density matrix variance, which specifically hold true for the Frobenius norm. They are given in Sec. 10. With this, the convergence of TJM can be proved as follows.

Theorem 2 (Convergence of TJM). *Let $d \in \mathbb{N}$ be the physical dimension and $L \in \mathbb{N}$ be the number of sites in the open quantum system described by the Lindblad master equation Eq.* (1)*. Furthermore, let $\boldsymbol{\rho}_N(t) = \frac{1}{N}\sum_{i=1}^{N} |\boldsymbol{\Psi_i}(t)\rangle\langle\boldsymbol{\Psi_i}(t)|$ be the approximation of the solution $\rho(t)$ of the Lindblad master equation in MPO format at time $t \in [0,T]$ for some ending time $T > 0$ and $N \in \mathbb{N}$ trajectories, where $|\boldsymbol{\Psi_i}(t)\rangle$ is a trajectory sampled according to the TJM in MPS format of full bond dimension. Then, the expectation value of the approximation of the corresponding density matrix $\rho_N(t) \in \mathbb{C}^{d^L\times d^L}$ is given by $\rho(t)$ and there exists a $c > 0$ such that the standard deviation of $\rho_N(t)$ can be upper bounded by*

$$\sigma(\rho_N(t)) \leq \frac{c}{\sqrt{N}} \tag{52}$$

for all matrix norms $\|\cdot\|$ *defined on* $\mathbb{C}^{d^L,d^L}$.

The full proof of Theorem 2 can be found in Sec. 10. For the convenience of the reader, a short sketch of the proof is provided here:

Proof. By the law of large numbers and the equivalence proof in Sec. 9, it follows that for every $N \in \mathbb{N}$ and every time $t \in [0, T]$ we have that $\mathbb{E}[\rho_N(t)] = \rho(t)$. The proof is carried out in state vector and density matrix format since MPSs and MPOs with full bond dimension exactly represent the corresponding vectors and matrices. Thus, we denote the state vector of a trajectory sampled according to the TJM at time t by $|\Psi_i(t)\rangle$. Using Lemma 4 and the fact that each trajectory is independently and identically distributed, we see that the variance of $\mathbb{V}_F[\rho_N(t)]$ decreases linearly with N by

$$\begin{aligned}\mathbb{V}_F[\rho_N(t)] &= \mathbb{V}_F\left[\frac{1}{N}\sum_{i=1}^{N} |\Psi_i(t)\rangle\langle\Psi_i(t)|\right] \quad (53)\\ &= \frac{1}{N^2}\mathbb{V}_F\left[\sum_{i=1}^{N} |\Psi_i(t)\rangle\langle\Psi_i(t)|\right]\\ &= \frac{1}{N^2}\sum_{i=1}^{N}\mathbb{V}_F\left[|\Psi_i(t)\rangle\langle\Psi_i(t)|\right] \quad (54)\\ &= \frac{1}{N}\mathbb{V}_F[|\Psi_1(t)\rangle\langle\Psi_1(t)|] \leq \frac{4}{N}, \quad (55)\end{aligned}$$

where the second to last step follows from the identical distribution of all $|\Psi_i(t)\rangle\langle\Psi_i(t)|$ for $i = 1, \ldots, N$. Hence, the Frobenius norm standard deviation is upper bounded by

$$\sigma_F[\rho_N(t)] = \frac{1}{\sqrt{N}}\sigma_F[|\Psi_1(t)\rangle\langle\Psi_1(t)|] \leq \frac{2}{\sqrt{N}}. \quad (56)$$

By the equivalence of norms on finite vector spaces, there exists $c_1, c_2 \in \mathbb{R}$ such that $c_1\|A\|_F \leq \|A\| \leq c_2\|A\|_F$ for all complex square matrices A and all matrix norms $\|\cdot\|$. Consequently, the convergence rate $\mathcal{O}(1/\sqrt{N})$ also holds true in trace norm and any other relevant matrix norm and is independent of system size. The statement then follows directly. □

4.3 Error measures

The major error sources of the TJM are as follows:

1 the time step error of the *Strang splitting* ($\mathcal{O}(\delta t^3)$) [52],

2 the time step error of the dynamic TDVP ($\mathcal{O}(\delta t^3)$ per time step and $\mathcal{O}(\delta t^2)$ for the whole time-evolution), and

3 the *projection error* of the dynamic TDVP.

Note that for 2TDVP the projection error is exactly zero if we consider Hamiltonians with only nearest neighbor interactions [15, 16] such that the projection error depends on the Hamiltonian structure. If each of the mentioned errors were zero, we would in fact calculate the MCWF from which we know that its stochastic uncertainty decreases with increasing number of trajectories according to the standard Monte Carlo convergence rate as shown in Sec. 4.2.

The projection error of 1TDVP can be calculated as the norm of the difference between the true time evolution vector $\boldsymbol{H_0} \left|\boldsymbol{\Phi}\right\rangle$ and the projected time evolution vector $P_{\mathcal{M}_\chi,\left|\boldsymbol{\Phi}\right\rangle} \boldsymbol{H_0} \left|\boldsymbol{\Phi}\right\rangle$. It depends on the structure of the Hamiltonian and the chosen bond dimensions $\chi \in \mathbb{N}^{L+1}$

$$\epsilon(\chi) = \|(I - P_{\mathcal{M}_\chi,\left|\boldsymbol{\Phi}\right\rangle}) \boldsymbol{H_0} \left|\boldsymbol{\Phi}\right\rangle \|_2. \tag{57}$$

This error can be evaluated as shown in Ref. [58]. It is well-known that the 1TDVP projector solves the minimization problem

$$P_{\mathcal{M}_\chi,\left|\boldsymbol{\Phi}\right\rangle} \boldsymbol{H_0} \left|\boldsymbol{\Phi}\right\rangle = \underset{M \in \mathcal{M}_\chi}{\text{argmin}} \|\boldsymbol{H_0} \left|\boldsymbol{\Phi}\right\rangle - M\|_2. \tag{58}$$

It can thus be noted that TJM uses the computational resources in an optimal way regarding the accuracy in time-evolution [13]. The errors in the dissipative contraction and in the jump application are both zero.

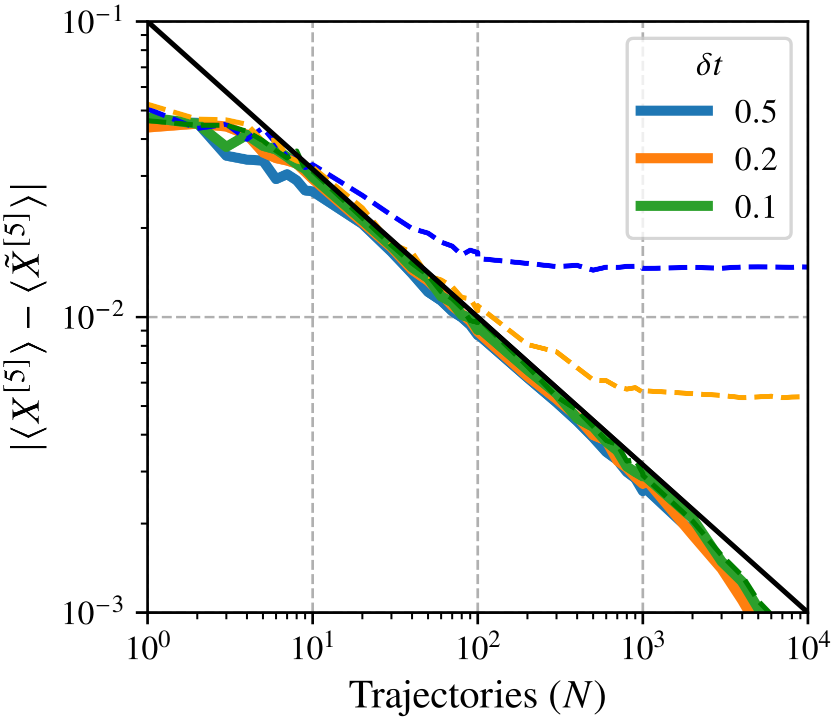

Figure 6: This plot shows the error in a local observable as a function of TJM trajectories at time $Jt = 1$ plotted for various time step sizes δt. Each point on the lines is generated by averaging over the error of 100 batches of N trajectories. The dotted lines correspond to standard first-order Trotterization to serve as a reference for the TJM's second-order splitting. The standard deviation of $\delta t = 0.1$ in the TJM over all samples is plotted as the shaded area. We see that the TJM converges according to the predicted $\sim \frac{C}{\sqrt{N}}$, in this example $C = 0.1$, (denoted by the solid black line) for all times dt. This illustrates that the second-order splitting leads to low time step error such that N dominates δt as a relevant variable in the TJM.

5 Benchmarking

To benchmark the proposed TJM, we consider a 10-site *transverse-field Ising model* (TFIM),

$$H_0 = -J \sum_{i=1}^{L-1} Z^{[i]} Z^{[i+1]} \; - \; g \sum_{j=1}^{L} X^{[j]}, \tag{59}$$

where $X^{[i]}$ and $Z^{[i]}$ are Pauli operators acting on the i-th site of a 1D chain. We evolve this system under a noise model that consists of single-site relaxation and dephasing operators,

$$\sigma_- = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{60}$$

on *all* sites in the lattice. Thus, our set of Lindblad jump operators is given by

$$\{L_m\}_{m=1}^{2L} = \{\sigma_-^{[1]}, \ldots, \sigma_-^{[L]}, \, Z^{[1]}, \ldots, Z^{[L]}\} \tag{61}$$

with coupling factors $\gamma = \gamma_- = \gamma_z = 0.1$.

All simulations reported here were performed on a consumer-grade Intel i5-13600KF CPU (5.1 GHz, 14 cores, 20 threads), using a parallelization scheme in which each TJM trajectory runs on a separate thread. This setup exemplifies how the TJM can handle large-scale open quantum system simulations efficiently even without specialized high-performance hardware. An implementation can be found in the *MQT-YAQS* package available at [59] as part of the *Munich Quantum Toolkit* [60].

5.1 Monte Carlo convergence

We first examine how the TJM converges with respect to the number of trajectories N and the time step size δt. As an exact reference, a direct solution of the Lindblad equation via QuTiP [61, 62] is used.

In Fig. 6, the absolute error in the expectation value of a local X operator at the chain's center is plotted, evaluated at $Jt = 1$ for up to $N = 10^4$ trajectories and for several time step sizes $\delta t \in \{0.1, 0.2, 0.5\}$. Each point in the plot represents an average over 100 independent batches of N trajectories. The dotted lines correspond to a first-order Trotter decomposition of the TJM serving as a baseline. The solid lines correspond to the second-order Trotterization used as a basis for the TJM. The solid black line represents the expected Monte Carlo convergence $\propto 1/\sqrt{N}$ (with prefactor $C = 0.1$).

For the first-order Trotter method (dotted lines), larger step sizes ($\delta t = 0.2$ or 0.5) induce a plateau, indicating that Trotter errors dominate when N becomes large. By contrast, for $\delta t = 0.1$, the first-order method follows the $1/\sqrt{N}$ trend more closely.

In comparison, the second-order TJM approach (solid lines) maintains $\sim 1/\sqrt{N}$ scaling for *all* tested values of δt, confirming that higher-order Trotterization reduces the inherent time step error below the level where it competes with Monte Carlo sampling error. While it is possible that for very large N (beyond those shown here) time discretization errors might again appear, in practice our second-order scheme keeps these systematic errors well below the scale relevant to typical simulation tolerances.

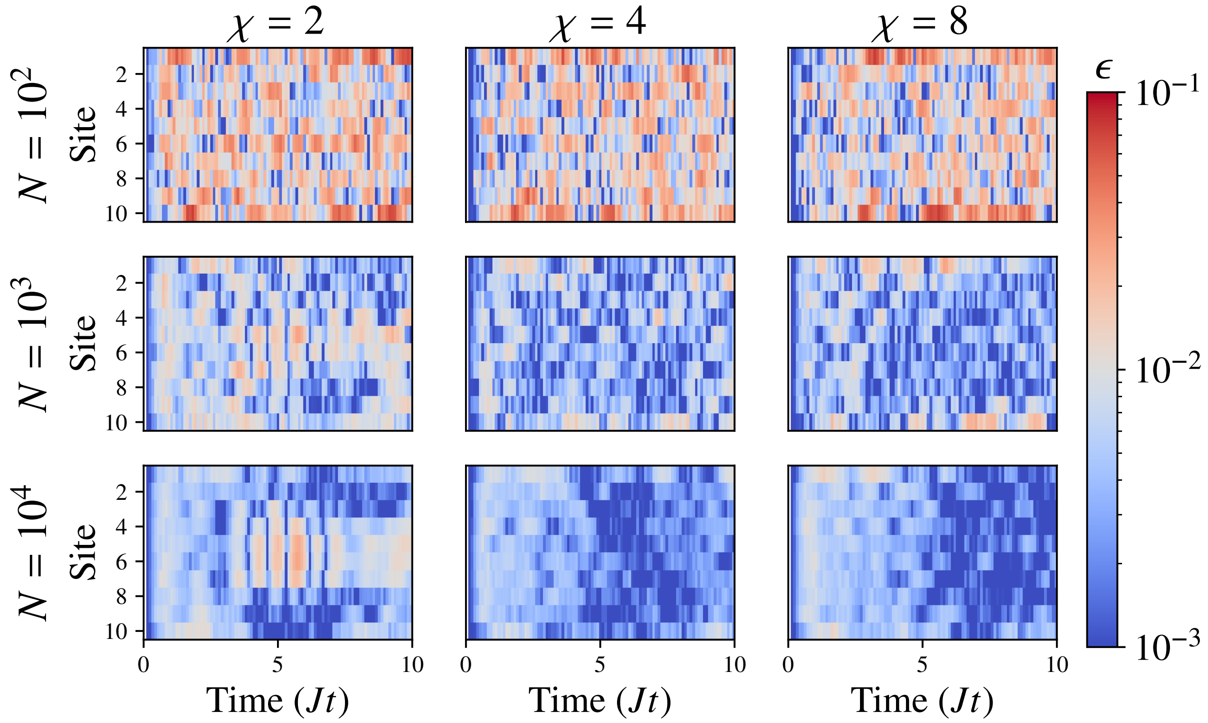

Figure 7: This plot complements Fig. 6 to show the convergence at various times $Jt = [0, 10]$ with $\delta t = 0.1$ according to number of trajectories N and bond dimension χ. Each subplot shows the error in expectation value of a local observable at each site during the time-evolution, calculated by averaging 1000 batches of N trajectories. The color map is centered at a threshold $\epsilon = 10^{-2}$ such that blue areas indicate lower errors and red areas indicate higher errors. We first see that the TJM is independent of time scale and that it shows similar errors over all sites and all times. We also observe that the trajectories have a significantly larger effect compared to the bond dimension of the individual trajectories. While $\chi = 2$ is unable to capture some dynamics even with high trajectories, the convergence of $\chi = 4$ and $\chi = 8$ are effectively identical.

5.2 Effect of bond dimension and elapsed time

Next, we explore how the TJM's accuracy depends on the maximum bond dimension χ of the trajectory MPS and the total evolution time T. We compute

$$\epsilon = \left|\langle X\rangle - \langle\tilde{X}\rangle\right|,$$

where $\langle X\rangle$ is the exact observable (via QuTiP) and $\langle\tilde{X}\rangle$ is the TJM result, for each site at discrete times $Jt \in \{0, 0.1, \ldots, 10\}$ using a time step $\delta t = 0.1$. We vary the bond dimension $\chi \in \{2, 4, 8\}$ and the number of trajectories $N \in \{100, 1000, 10000\}$. The resulting errors, averaged over 1000 batches for each (N, χ), are shown in Fig. 7 using a color map centered at $\epsilon = 10^{-2}$.

First, we note that the error appears consistent across all sites except for slightly increased values at the boundary sites. This indicates that the TJM's local updates do not inherently favor any region of the chain.

Second, while the simulation time T can affect the complexity of the noisy dynamics, the TJM generally maintains similar accuracy at all times. Indeed, times closer to the initial state ($Jt \approx 0$) exhibit very low errors (blue regions) simply because noise has had less time to build up correlations or deviations.

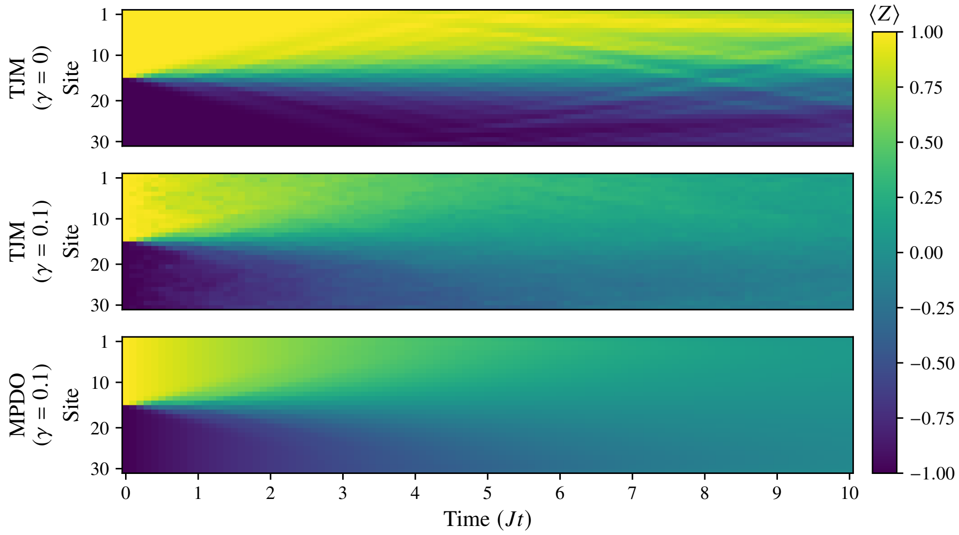

Figure 8: These plots show the results of the evolution of a 30-site noisy XXX Heisenberg model with parameters $J = 1, h = 0.5$ and a domain-wall initial state with wall at site 15. The noise model includes relaxation σ_- and excitation σ_+, both with strength $\gamma_- = \gamma_+ = 0.1$. First, we run the TJM with $\gamma = 0$ to generate a noise-free reference which requires only a single “trajectory". Then, we run it again with $\gamma = 0.1$ using $N = 100$ and bond dimension $\chi = 4$. Finally, this is compared against the time evolution of a Lindbladian using an MPDO with max bond dimension $D = 400$ (which is reached at $t \approx 4$ before truncation). All simulations use a timestep $\delta t = 0.1$. Both TJM runs were executed in roughly 5 minutes, while the MPDO has taken over 24 hours.

Finally, the bond dimension χ plays a comparatively minor role in the overall error. Although $\chi = 2$ sometimes fails to capture certain features (leading to increased error in specific time windows), $\chi = 4$ and $\chi = 8$ give nearly identical results. Interestingly, $\chi = 4$ can even show slightly better convergence than $\chi = 8$ in some patches, which likely is a sampling artifact within the batches.

In summary, these benchmarks confirm that (i) the time step error is effectively minimized by the second-order Trotterization, leaving Monte Carlo sampling as the primary source of error, and (ii) increasing the number of trajectories N is typically more crucial than increasing the bond dimension χ.

6 New Frontiers

In this section, we push the limits of the TJM to large-scale noisy quantum simulations, highlighting its practical utility and the physical insights it can provide. Concretely, we explore a noisy XXX Heisenberg chain described by the Hamiltonian

$$\begin{aligned} H_0 = -J\Big(\sum_{i=1}^{L-1} X^{[i]}X^{[i+1]} \;+\; Y^{[i]}Y^{[i+1]} \;+\; Z^{[i]}Z^{[i+1]}\Big) \\ - h\sum_{j=1}^{L} Z^{[j]}, \end{aligned}$$

subject to relaxation (γ_-) and excitation (γ_+) noise processes. Each simulation begins with a *domain-wall* initial state

$$|\Psi(0)\rangle \;=\; |\sigma_1\,\sigma_2\,\ldots\,\sigma_\ell\,\ldots\,\sigma_L\rangle\,, \quad \sigma_\ell = \begin{cases} 0, & 1 \le \ell < \frac{L}{2}, \\ 1, & \frac{L}{2} \le \ell \le L, \end{cases}$$

such that the top half of the chain is initialized in the spin-down $|0\rangle$ state and the bottom half in the spin-up $|1\rangle$ state, thus forming a sharp "wall". We track the local magnetization $\langle Z\rangle$ at each site as the primary observable of interest.

To demonstrate the scalability of our approach, we simulate system sizes ranging from moderate $L = 30$ to quite large $L = 100$, and then up to $L = 1000$ sites. By examining a wide range of noise strengths and run times, we reveal how noise impacts the evolution of such extended systems—insights that are otherwise out of reach for many conventional methods.

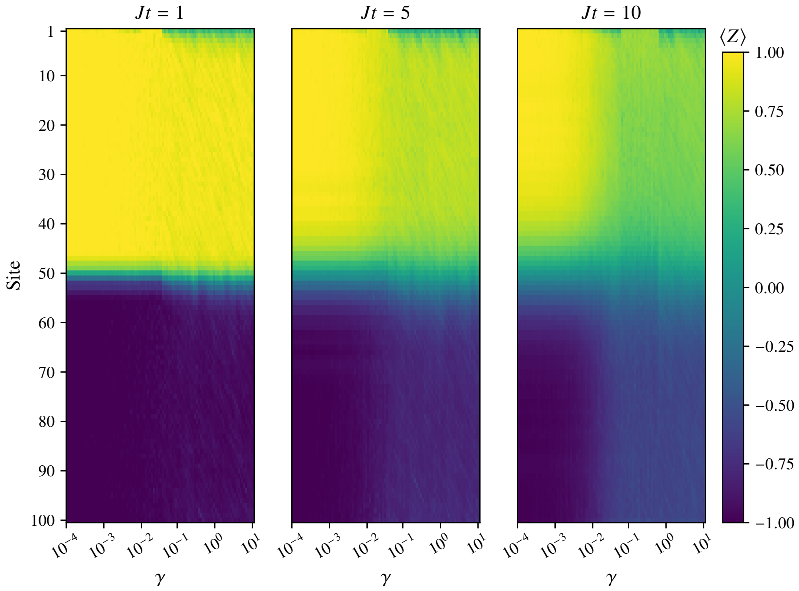

Figure 9: This figure shows the results of the large-scale simulation of the domain-wall state with wall at site 50 evolved according to a noisy 100-site XXX Heisenberg model with parameters $J = 1, h = 0.5$. Each column corresponds to a different elapsed time of the time evolution $Jt = 1, 5, 10$, respectively. Each plot shows the effect of various coupling factors from very weak noise $\gamma = 10^{-4}$ to very strong noise $\gamma = 10$ at the corresponding times. At each time, we see a threshold where noise begins to dominate the state, overwhelming its natural dynamics. These simulations were run with $N = 100$ trajectories, bond dimension $\chi = 4$, and time step $\delta t = 0.5$. This took approximately 2.5 hours to run.

6.1 Comparison with MPDO Lindbladians (30 sites)

In the first test, we use the TJM to simulate a 30-site noisy Heisenberg XXX model $(J = 1,\ h = 0.5)$ with a domain-wall initial state at site 15 and compare our results against a state-of-the-art MPDO Lindbladian approach (implemented via the `LindbladMPO` package [57]). The results are depicted in Fig. 8.

To establish a baseline, we first run a *noise-free* simulation using the standard time-dependent variational principle (TDVP). In practice, this is equivalent to the TJM with $\gamma = 0$ and $N = 1$. We observe that the initial domain wall begins to spread from the central site at early times $(Jt < 1)$, reflecting off boundaries multiple times and broadening further after each reflection.

Next, we investigate a *noisy* scenario by setting $\gamma = 0.1$. Here, we run the TJM with $N = 100$ trajectories, bond dimension $\chi = 4$, and a time step $\delta t = 0.1$. As expected, the noise damps out the coherent domain-wall disturbance over the same time window $(0 \leq Jt \leq 10)$, leading to uniform $\langle Z \rangle = 0$ values across the chain compared to the noise-free case.

We also compare these TJM results to a Lindbladian simulation using an MPDO with bond dimension $D = 400$. We observe that the TJM aligns well with the MPDO dynamics although there is some visible dynamics in the TJM simulation that the MPDO does not capture (seen as vague lines past $Jt = 4$) due to the truncation of the bond dimension. Additionally, whereas both TJM simulations (noise-free and noisy) took only about 5 minutes compute time in total, the MPDO simulation required over 24 hours, yet yielded broadly similar results. This highlights that the TJM can achieve comparable accuracy with substantially reduced computation time.

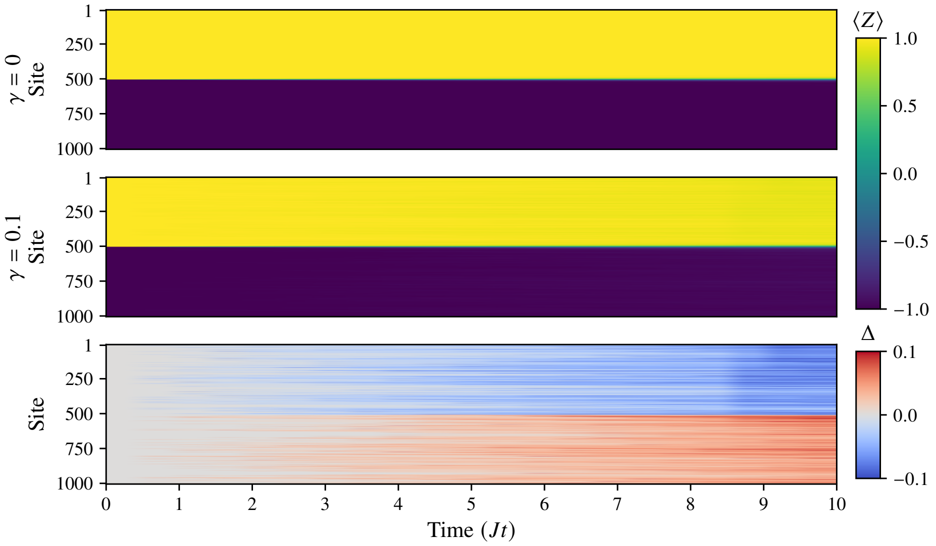

Figure 10: These plots show the results of the evolution of a 1000-site noisy XXX Heisenberg model with parameters $J = 1, h = 0.5$ and a domain-wall initial state with wall at site 500. First, we run the TJM with $\gamma = 0$ to generate a noise-free reference which requires only a single "trajectory". Then, we run it again with $\gamma = 0.1$ using $N = 100$ and bond dimension $\chi = 4$. Next, we run it again with $\gamma = 0.1$ using $N = 100$ and bond dimension $\chi = 4$. Finally, we plot the difference Δ between these two plots. While visually very similar, we see that the TJM is able to capture even relatively minor noise effects, and, as a result, show how this can eventually lead to macroscopic changes.

6.2 Exploring noise in large systems (100 sites)

In the second test, we push the model complexity beyond the regime of most conventional methods by simulating a 100-site noisy XXX Heisenberg chain $(J = 1,\ h = 0.5)$ with a domain-wall initial state at site 50. Here, we vary γ over 100 logarithmically spaced values from 10^{-4} to 10^{1} at times $Jt \in [0, 10]$. We run the TJM with $N = 100$ trajectories, bond dimension $\chi = 4$, and a step size $\delta t = 0.5$. We then extract snapshots of $\langle Z \rangle$ at $Jt = \{1, 5, 10\}$ as illustrated in Fig. 9. The full test took approximately 2.5 hours.

Overall, we observe a clear distinction between a *weak-noise regime* (for smaller γ) in which the domain wall behaves similarly to the noise-free case, and a *strong-noise regime* (larger γ) where noise dominates the system dynamics. At early times $(Jt = 1)$, only $\gamma \gtrsim 5 \times 10^{-2}$ visibly alters the local measurements. By $Jt = 5$ and $Jt = 10$, the threshold beyond which noise washes out the domain wall drops further to $\gamma \gtrsim 10^{-2}$. As we can see this threshold shifting, we expect longer timescales to inevitably also be overcome even by weak noise. Thus, even for a significantly larger system, modest noise can radically suppress the domain-wall features. Nevertheless, we are able to capture these dynamics in a feasible runtime using the TJM.

6.3 Pushing the envelope (1000 sites)

Finally, we demonstrate the scalability of the TJM by simulating a noisy XXX Heisenberg chain of 1000 sites. While our method can handle even larger systems – especially if migrated from consumer-grade to server-grade hardware – we choose 1000 sites here as a reasonable upper bound for our present setup. This test, which took roughly 7.5 hours, maintains the same parameters as before: we run a noise-free simulation ($\gamma = 0$) requiring only a single trajectory, and then a noisy simulation with $\gamma = 0.1$, bond dimension $\chi = 4$, and $N = 100$ trajectories at time step $\delta t = 0.5$.

In Fig. 10, we show the time evolution of the noisy and noise-free systems along with the difference $\Delta = \langle Z_{\text{Noisy}} \rangle - \langle Z_{\text{Noise-Free}} \rangle$. While the overall domain-wall structure appears visually similar at a glance, single-site quantum jumps from relaxation and excitation lead to small, localized “scarring” that accumulates into increasingly macroscopic changes by $Jt = 10$. Notably, although the larger lattice provides more possible sites for noise to act upon, the size may confer partial robustness in early stages of evolution. Consequently, we see that even a modest amount of noise ($\gamma = 0.1$) can subtly alter the state in a way that becomes significant over time. These results underscore that the TJM can efficiently capture open-system dynamics in large spin chains on a consumer-grade CPU, thus opening new frontiers for studying the interplay between coherent dynamics and environmental noise at unprecedented scales.

7 Discussion

All quantum system environments are open to some extent, and hence, having powerful tools available that classically simulate interacting open quantum many-body systems is crucial. While recent years have seen a development towards large-scale, state-of-the-art-simulations for quantum ground state and quantum circuit simulations, the same cannot quite be said for the simulation of open quantum systems. This seems a grave omission, given the important role quantum many-body systems play in notions of quantum simulation [4, 5]. The present work is meant to close this gap, providing a massively scalable algorithm for the simulation of Markovian open quantum systems by means of tensor networks, paving the way for large-scale classical simulations of open quantum systems matching similar tools for equilibrium problems. By bridging the gap between theoretical frameworks and practical applications, this work not only advances the field of open quantum systems but also contributes to the broader goal of realizing robust and scalable quantum technologies in real-world settings. It is our hope that this work can provide important services in the benchmarking and design of state-of-the-art physical platforms of quantum simulators in the laboratory, as well as inspire and facilitate research into large-scale open quantum systems and noisy quantum hardware that was previously infeasible.

8 Methods

This section is devoted to providing the background of key methods and techniques used in this work. We start by presenting core ideas of tensor network methods [9, 16, 63, 2] that this work is building on.

8.1 Matrix product states

Consider a one-dimensional lattice made of $L \in \mathbb{N}$ sites, each corresponding to a local Hilbert space $\mathcal{H}_d$ of dimension $d \in \mathbb{N}$. The Hilbert space of the full lattice is then defined by the iterative tensor product of the L local Hilbert spaces $\mathcal{H} = \bigotimes_{\ell=1}^{L} \mathcal{H}_d$. Elements $|\Psi\rangle$ of this multi-site Hilbert space $\mathcal{H}$ are state vectors defined by

$$|\Psi\rangle = \sum_{\sigma_1,\ldots,\sigma_L=1}^{d} \Psi_{\sigma_1\ldots\sigma_L} |\sigma_1,\ldots,\sigma_L\rangle \tag{62}$$

where $\sigma_\ell \in [1,\ldots,d]$ are the physical dimensions for all $\ell = 1,\ldots,L$ and $\Psi \in \mathbb{C}^{d^L}$ with elements $\Psi_{\sigma_1\ldots\sigma_L} \in \mathbb{C}$.

The vector $|\Psi\rangle \in \mathbb{C}^{d^L}$ can be decomposed into a *matrix product state* (MPS) [64, 11, 9]

$$|\mathbf{\Psi}\rangle = \sum_{\sigma_1,\ldots,\sigma_L=1}^{d} M_1^{\sigma_1} \ldots M_L^{\sigma_L} |\sigma_1,\ldots,\sigma_L\rangle, \tag{63}$$

which is made of L degree-3 tensors

$$M := \{M_\ell \in \mathbb{C}^{d\times\chi_{\ell-1}\times\chi_\ell} \mid \ell = 1,\ldots,L\}, \tag{64}$$

consisting of d matrices corresponding to each index σ_ℓ

$$M_\ell := \{M_\ell^{\sigma_\ell} \in \mathbb{C}^{\chi_{\ell-1}\times\chi_\ell} \mid \sigma_\ell = 1,\ldots,d\}. \tag{65}$$

This structure's complexity is determined by its bond dimensions $\chi_\ell \in \mathbb{N}$ (where $\chi_0 = \chi_L = 1$), which scale with the entanglement entropy of the quantum state it represents. For the rest of this work we denote a quantum state vector living in $\mathcal{H}$ as $|\Psi\rangle$ and we use the bold notation $|\mathbf{\Psi}\rangle$ for the same quantum state vector in an MPS representation.

The MPS representation is non-unique and gauge-invariant for representing a given quantum state such that the individual tensors can be placed in canonical forms which allow many operations to reduce in complexity. These conditions are the left canonical form

$$\sum_{\sigma_\ell=1}^{d} \sum_{a_{\ell-1}=1}^{\chi_{\ell-1}} \overline{M}_\ell^{\sigma_\ell,a_{\ell-1},a_\ell} M_\ell^{\sigma_\ell,a_{\ell-1},a_\ell} = I \in \mathbb{C}^{\chi_\ell\times\chi_\ell}, \tag{66}$$

and the right canonical form

$$\sum_{\sigma_\ell=1}^{d} \sum_{a_\ell=1}^{\chi_\ell} \overline{M}_\ell^{\sigma_\ell,a_{\ell-1},a_\ell} M_\ell^{\sigma_\ell,a_{\ell-1},a_\ell} = I \in \mathbb{C}^{\chi_{\ell-1}\times\chi_{\ell-1}}, \tag{67}$$

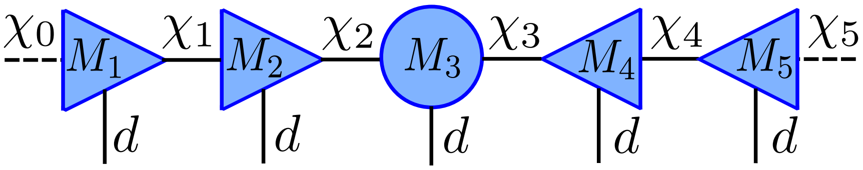

Figure 11: This figure shows a 5-site MPS in mixed canonical form with orthogonality center at site 3. The indices d_j indicate the physical dimensions and χ_j the bond dimensions. A dashed line indicates a dummy index where $\chi_j = 1$. We use right-pointing triangles to denote the left-canonical form and left-pointing triangles for the right-canonical form. A tensor with no required form is shown by a circle.

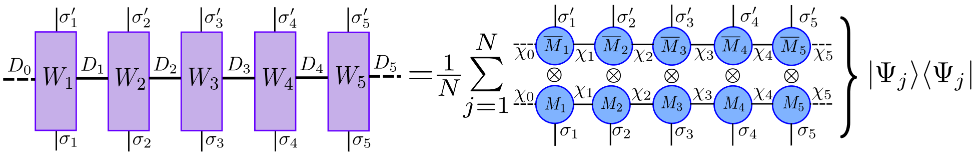
σ'_1 σ'_2 σ'_3 σ'_4 σ'_5
D_0 W_1 D_1 W_2 D_2 W_3 D_3 W_4 D_4 W_5 D_5
σ_1 σ_2 σ_3 σ_4 σ_5
$= \frac{1}{N} \sum_{j=1}^{N}$
χ_0 $\overline{M_1}$ χ_1 $\overline{M_2}$ χ_2 $\overline{M_3}$ χ_3 $\overline{M_4}$ χ_4 $\overline{M_5}$ χ_5
$\otimes$
χ_0 M_1 χ_1 M_2 χ_2 M_3 χ_3 M_4 χ_4 M_5 χ_5
$|\Psi_j\rangle\langle\Psi_j|$

Figure 12: An MPDO can be created by a weighted summation of MPS structures and their conjugates as described in Sec. 8.2. However, this can lead to massive MPDO bond dimensions since the bonds for each state vector $|\Psi_j\rangle$ are combined such that $D_\ell = \prod_{j=1}^{N}(\chi_\ell^2)_j$. This equivalence is critical for the foundations of this work. However, we avoid creating an MPDO directly due to the large bond dimensions.

where $a_\ell \in [1, \ldots, \chi_\ell]$ and $\overline{M}$ is the conjugated tensor. Finally, these can be combined to fix the MPS in a mixed canonical form around an orthogonality center at site tensor j

$$\begin{aligned}\sum_{\sigma_\ell=1}^{d}\sum_{a_{\ell-1}=1}^{\chi_{\ell-1}} \overline{M}_\ell^{\sigma_\ell,a_{\ell-1},a_\ell} M_\ell^{\sigma_\ell,a_{\ell-1},a_\ell} &= I \text{ such that } \ell < j,\\ \sum_{\sigma_\ell=1}^{d}\sum_{a_\ell=1}^{\chi_\ell} \overline{M}_\ell^{\sigma_\ell,a_{\ell-1},a_\ell} M_\ell^{\sigma_\ell,a_{\ell-1},a_\ell} &= I \text{ such that } \ell > j.\end{aligned} \tag{68}$$

A visualization of an MPS in site-canonical form at $j = 3$ can be found in Fig. 11 where the triangular tensors indicate canonical forms and the circular tensor indicates an arbitrary form.

8.2 Matrix product (density) operators

Just as states on a one-dimensional lattice can be represented by MPS, the operators acting upon these states can be represented by a tensor train known as *matrix product operators* (MPO) [65, 66]. Suppose we have a bounded operator $O \in \mathcal{B}(\mathcal{H})$ such that

$$O = \sum_{\sigma_1,\sigma'_1,\ldots,\sigma_L,\sigma'_L=1}^{d} W^{\sigma_1,\sigma'_1,\ldots,\sigma_L,\sigma'_L} \, |\sigma_1,\ldots,\sigma_L\rangle\langle\sigma'_1,\ldots,\sigma'_L| \,, \tag{69}$$

where $W \in \mathbb{C}^{d^{2L}}$ and for each element $W_{\sigma_1,\sigma'_1,\ldots,\sigma_L,\sigma'_L}$ we have $\sigma_\ell, \sigma'_\ell \in \{1,\ldots,d\}$ for all $\ell = 1,\ldots,L$. This coefficient tensor W can then be decomposed into a list of L degree-4 tensors

$$W := \{W_\ell \in \mathbb{C}^{d\times d\times D_{\ell-1}\times D_\ell} \mid \ell = 1,\ldots,L\}, \tag{70}$$

created by degree-2 tensors for the indices $\sigma_\ell, \sigma'_\ell$:

$$W_\ell := \{W_\ell^{\sigma_\ell,\sigma'_\ell} \in \mathbb{C}^{D_{\ell-1}\times D_\ell} \mid \sigma_\ell, \sigma'_\ell = 1,\ldots,d\}. \tag{71}$$

Next, we define multi-indices

$$\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_L), \quad \boldsymbol{\sigma}' = (\sigma_1', \dots, \sigma_L'),$$

and write

$$|\boldsymbol{\sigma}\rangle \equiv |\sigma_1, \dots, \sigma_L\rangle \,, \quad \langle\boldsymbol{\sigma}'| \equiv \langle\sigma_1', \dots, \sigma_L'| \,,$$

as well as

$$W(\boldsymbol{\sigma}, \boldsymbol{\sigma}') \ := \ \prod_{\ell=1}^{L} W_\ell^{\sigma_\ell, \sigma_\ell'}.$$

Then the MPO in bold notation is compactly written as

$$\boldsymbol{O} \ = \ \sum_{\boldsymbol{\sigma}, \boldsymbol{\sigma}'=1}^{d} W^{\boldsymbol{\sigma}, \boldsymbol{\sigma}'} \ |\boldsymbol{\sigma}\rangle\langle\boldsymbol{\sigma}'| \,. \tag{72}$$

Like the MPS, the computational complexity is determined by the bond dimensions $D_\ell \in \mathbb{N}$, which are related to the operator entanglement [67, 68].

Particularly important for this work, density matrices, i.e., mixed quantum states, can be represented in this MPO format, known as *matrix product density operators* (MPDO) [69]. A mixed quantum state $\rho \in \mathcal{B}(\mathcal{H})$ is equivalent to a finite sum of the outer product of $N \in \mathbb{N}$ state vectors $|\Psi_j\rangle$ and weighted by probabilities p_j with $j = 1, \dots, N$ such that

$$\rho = \sum_{j=1}^{N} p_j \, |\Psi_j\rangle \, \langle\Psi_j| \,, \tag{73}$$

and $\sum_j p_j = 1$. If the pure states are represented as MPS according to Eq. (63), this outer product would result in an MPO structure with degree-4 site tensors $W_\ell \in \mathbb{C}^{d \times d \times D_{\ell-1} \times D_\ell}, \ell = 1, \dots, L$ which can be decomposed in a sum of N outer products of MPS given by

$$\{M_{\ell,j} \in \mathbb{C}^{d \times \chi_{\ell-1} \times \chi_\ell} | \ell = 1, \dots, L\}_{j=1}^{N} \tag{74}$$

such that each site W_ℓ is represented by

$$\sum_{j=1}^{N} \overline{M}_{\ell,j} \otimes M_{\ell,j}. \tag{75}$$

The bond dimensions of the MPO are made of the constituent MPS bonds such that $D_{\ell-1} = \sum_{i=1}^{N} \chi_{\ell-1,i}^2$ and $D_\ell = \sum_{i=1}^{N} \chi_{\ell,i}^2$, where $\chi_{\ell,i}$ is the ℓ-th bond dimension of the i-th MPS. This can be seen in Fig. 12.

8.3 Time-dependent variational principle

8.3.1 Overview

Let $|\boldsymbol{\Psi}(t)\rangle \in \mathcal{H}$ for some $t \in \mathbb{R}_+$ be a time-dependent quantum state represented by an MPS with bond dimensions $\chi = \{\chi_0, \dots, \chi_L\}$. Then $|\boldsymbol{\Psi}(t)\rangle$ can be understood as an element of a manifold $\mathcal{M}_\chi \subseteq \mathcal{H}$, solely defined by the set of bond dimensions χ. As the bond dimensions increase, this

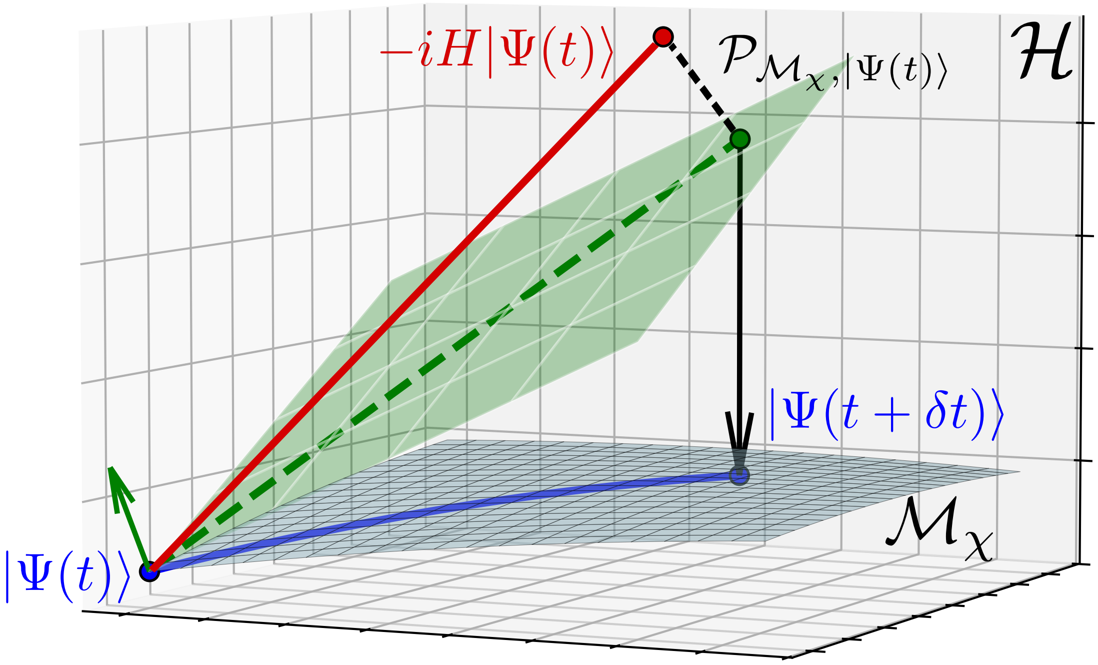
$-iH\,|\Psi(t)\rangle$
$\mathcal{P}_{\mathcal{M}_\chi,|\Psi(t)\rangle}$
$\mathcal{H}$
$|\Psi(t+\delta t)\rangle$
$\mathcal{M}_\chi$
$|\Psi(t)\rangle$

Figure 13: A time-dependent MPS $|\Psi(t)\rangle$ with fixed bond dimensions χ can be viewed as a point on a manifold $\mathcal{M}_\chi \in \mathcal{H}$. The time-evolution caused by $-iH\,|\Psi(t)\rangle$ will generally leave the manfiold, requiring a larger bond dimension to represent. TDVP projects this time-evolution to the tangent space of the original manifold (shown by the projector $\mathcal{P}_{\mathcal{M}_\chi,|\Psi(t)\rangle}$). Each point in the tangent space can be described as a linear combination of partial derivatives of the original point such that by solving a set of site-wise coupled differential equations, we can evolve purely on the original manifold and limit the growth in bond dimension to represent $|\Psi(t+\delta t)\rangle$.

manifold covers a larger part of the Hilbert space such that $\mathcal{M}_\chi \subset \mathcal{M}_{\chi'} \subseteq \mathcal{H}$ [58] for $\chi < \chi'$. Generally, time-evolution of some state with the *time-dependent Schrödinger equation* (TDSE) according to some Hamiltonian $H \in \mathcal{B}(\mathcal{H})$ leads to a growth of the bond dimensions, which severely limits the computational efficiency.

By utilizing the manifold picture of MPS, we get a powerful method known as the *time-dependent variational principle* (TDVP). Rather than allowing the bond dimension to grow, this method projects the Hamiltonian to the tangent space of the current MPS manifold before carrying out the time evolution [15] as visualized in Fig. 13. In a more precise form, TDVP solves the projected TDSE

$$\frac{d}{dt} |\Psi(t)\rangle = -iP_{\mathcal{M}_\chi, |\mathbf{\Psi}(t)\rangle} H |\Psi(t)\rangle , \tag{76}$$

where $P_{\mathcal{M}_\chi, |\mathbf{\Psi}(t)\rangle} \in \mathcal{B}(\mathcal{H})$ is a projector that projects an MPS onto the tangent space of the manifold $\mathcal{M}_\chi$ at the point $|\mathbf{\Psi}(t)\rangle$. TDVP offers significant advantages over other time evolution methods, such as *time-evolving block decimation* (TEBD) [11, 43, 70], by avoiding Trotter and truncation errors while conserving the system's norm and energy [13, 58]. This benefit is transformed into projection error which results in the optimal representation of an MPS at a lower bond dimension (on the smaller manifold) compared to the sub-optimal truncated MPS [16]. The simplest form of TDVP known as 1TDVP leads to $2L-1$ coupled local ordinary differential equations (ODEs) which correspond to a one-site integration scheme similar to the *density matrix renormalization group* (DMRG) method [9]. This can then be extended to an n-site integration scheme [15], allowing for the simultaneous integration of neighboring sites. Higher-order nTDVP reduces projection error, such that if a Hamiltonian has n'-body interactions, no projection error occurs for $n' \leq n$, although at the cost of higher computational complexity [16].

8.3.2 Coupled ordinary differential equations

For 1TDVP, the projection can be expressed as a projector splitting at each site,

$$P_{\mathcal{M}_\chi, |\mathbf{\Phi}\rangle} = \sum_{\ell=1}^{L} K_{\ell, |\mathbf{\Phi}\rangle} - \sum_{\ell=1}^{L-1} G_{\ell, |\mathbf{\Phi}\rangle}, \tag{77}$$

which always depend on the current MPS $|\mathbf{\Phi}(t)\rangle$, where

$$K_{\ell, |\mathbf{\Phi}\rangle} = |\mathbf{\Phi}^L_{\ell-1}\rangle \langle \mathbf{\Phi}^L_{\ell-1}| \otimes I_\ell \otimes |\mathbf{\Phi}^R_{\ell+1}\rangle \langle \mathbf{\Phi}^R_{\ell+1}| , \tag{78}$$

represents the Krylov projectors [71] of the MPS $|\mathbf{\Phi}\rangle$ that fix the orthogonality center at site ℓ, and where $G_{\ell, |\mathbf{\Phi}\rangle}$ are defined as

$$G_{\ell, |\mathbf{\Phi}\rangle} = |\mathbf{\Phi}^L_{\ell}\rangle \langle \mathbf{\Phi}^L_{\ell}| \otimes |\mathbf{\Phi}^R_{\ell+1}\rangle \langle \mathbf{\Phi}^R_{\ell+1}| . \tag{79}$$

We have dropped the time parameter t for ease of notation. Here, the left bipartition of the MPS $|\mathbf{\Phi}\rangle$ around the orthogonality center ℓ is expressed as

$$|\mathbf{\Phi}^L_\ell\rangle = \sum_{\sigma_1, \ldots, \sigma_\ell = 1}^{d} M_1^{\sigma_1} \ldots M_\ell^{\sigma_\ell} |\sigma_1, \ldots, \sigma_\ell\rangle , \tag{80}$$

and the right bipartition is given by

$$|\mathbf{\Phi}^R_\ell\rangle = \sum_{\sigma_\ell, \ldots, \sigma_L = 1}^{d} M_\ell^{\sigma_\ell} \ldots M_L^{\sigma_L} |\sigma_{\ell+1}, \ldots, \sigma_L\rangle , \tag{81}$$

where M_ℓ with $\ell = 1, \dots, L$ are the site tensors of $|\Phi\rangle$. When we substitute the definition of $P_{\mathcal{M}_\chi, |\Phi\rangle}$ into Eq. (76), we derive a set of coupled local *ordinary differential equations* (ODEs) that describe the evolution of the state vector $|\Phi\rangle$, where we use the bold notation $\boldsymbol{H_0}$ to denote the system Hamiltonian in MPO format as

$$\frac{d}{dt}\,|\Phi\rangle = -i\sum_{\ell=1}^{L} K_{\ell,|\Phi\rangle}\boldsymbol{H_0}\,|\Phi\rangle + i\sum_{\ell=1}^{L-1} G_{\ell,|\Phi\rangle}\boldsymbol{H_0}\,|\Phi\rangle\,. \tag{82}$$

This is then split into L forward-evolving terms given by

$$\frac{d}{dt}\,|\Phi\rangle = -iK_{\ell,|\Phi\rangle}\boldsymbol{H_0}\,|\Phi\rangle\,, \tag{83}$$

and $L-1$ backward-evolving terms

$$\frac{d}{dt}\,|\Phi\rangle = +iG_{\ell,|\Phi\rangle}\boldsymbol{H_0}\,|\Phi\rangle\,, \tag{84}$$

8.4 Computational effort of running the simulation

After having presented the basics of tensor network simulations, we now derive original material on the computational effort of running a simulation with the *tensor jump method (TJM)* as developed in this work. We can analyze the total complexity of the TJM by breaking it down into its constituent components. For this, we have a given number of trajectories N and a total number n of time steps to perform (regardless of the size of δt itself). The complexity of calculating N trajectories with n time steps with time step size δt depends on the complexity of the dissipative contraction $\mathcal{D}$, the calculation of the probability distribution of the jump operators $L_m, m = 1, \dots, k$, and the complexity of a one-site TDVP step [72].

The unitary time-evolution $\mathcal{U}$ according to 1TDVP has a complexity of

$$\mathcal{O}(L[d^2D^2\chi_{\text{max}}^2 + dD\chi_{\text{max}}^3 + d^2\chi_{\text{max}}^3]), \tag{85}$$

where D is the maximum bond dimension of the Hamiltonian MPO. Since the majority of time steps is performed with 1TDVP and 2TDVP is not performed with maximum bond dimension, the scaling of 2TDVP is not relevant for the TJM. Additionally, the complexity of 1TDVP and 2TDVP only differs in terms of the local dimension d, which scales quadratic in d for 1TDVP and cubic in 2TDVP. Following this, the dissipative sweep $\mathcal{D}$ scales according to

$$\mathcal{O}(Ld^2\chi_{\text{max}}^2). \tag{86}$$

The complexity of the stochastic process $\mathcal{J}$ is determined by three parts: the calculation of the probability distribution, the sampling of a jump operator (which is in $\mathcal{O}(1)$ and therefore negligible), and its application. This leads to a total complexity of

$$\mathcal{O}(kd\chi_{\text{max}}^3 + d^2\chi_{\text{max}}^2), \tag{87}$$

for k total jump operators $L_m, m = 1, \dots, k$. Putting all this together, we end up with a total complexity of the TJM in

$$\mathcal{O}\Big(Nn\big[L(d^2D^2\chi_{\text{max}}^2 + dD\chi_{\text{max}}^3 + d^2\chi_{\text{max}}^3 + d^2\chi_{\text{max}}^2) + kd\chi_{\text{max}}^3 + d^2\chi_{\text{max}}^2\big]\Big),$$

where the dominant terms come from the TDVP sweep and the calculation of the probability distribution. Since the number of jump operators k is related to the system size L, e.g., a fixed number of jump operators per site, we can rewrite the complexity using $k = \alpha L$ as

$$\mathcal{O}\left(NnL\chi_{\text{max}}^3\big[dD + d^2 + \alpha d\big]\right),$$

where $\alpha \in \mathbb{N}$. However, since we assume jumps only occur rarely, the TDVP sweep dominates the stochastic process such that this reduces to

$$\mathcal{O}\left(NnL\chi_{\text{max}}^3\big[dD + d^2\big]\right), \tag{88}$$

where we assume $d, D \ll \chi_{\text{max}}$.

In comparison, the runtime to solve the Lindblad equation directly scales as $\mathcal{O}(nd^{6L})$ when using the superoperator formalism [46]. The MCWF method, with the same assumptions made for the TJM, requires $\mathcal{O}(Nnd^{3L})$ and a Lindblad MPDO requires $\mathcal{O}(nLd^4D_H^2D_s^2)$ such that D_s is the bond dimension of the MPO representing the density matrix (where we expect $D_s \gg \chi_{\text{max}}$), and D_H is the bond dimension of the Hamiltonian MPO format using the W^{II} algorithm (analogous to χ_{max} and D in the TJM, respectively) [56].

8.5 Resources required for storing the results

For a given maximum bond dimension χ_{max}, the memory complexity to store N MPS trajectories is given by $\mathcal{O}(NLd\chi_{\text{max}}^2)$. Exactly solving the Lindblad equation in matrix format requires storing the density matrix and all other operators in its full dimension. When stored as a complex-valued matrix, ρ has a memory complexity $\mathcal{O}(d^{2L})$ such that it scales exponentially for increasing L. Storing N trajectories $|\Psi\rangle \in \mathbb{C}^{d^L}$ reduces the memory complexity from $\mathcal{O}(d^{2L})$ to $\mathcal{O}(Nd^L)$ compared to the Lindbladian approach. Meanwhile, an MPDO simulation requires $\mathcal{O}(Ld^2D_s^2)$ to store the time-evolved density matrix.

We can compare the TJM and the MCWF complexities to determine the χ_{max} necessary for the TJM to be more compact. This results in the requirement that

$$\chi_{\text{max}} < \sqrt{\frac{d^L}{Ld}}. \tag{89}$$

Since $\sqrt{d^L/(Ld)} \to \infty$ as $L \to \infty$, we see that for large system sizes the TJM is always more compact than the MCWF. When compared against the MPDO, we similarly see that the TJM is more compact if

$$\chi_{\text{max}} < D_s\sqrt{\frac{d}{N}}, \tag{90}$$

where we expect $D_s \gg \chi_{\text{max}}$ for large-scale applications, particularly those with long timescales.

8.6 Resources required for calculating expectation values

Naturally, we can also use the stored states to solve expectation values. If we sample a local observable $O \in \mathcal{B}(\mathcal{H})$ at a time step t, each method has distinct requirements to calculate $\langle O(t)\rangle$. The MPS trajectories from the TJM can be used to calculate the expectation value of an MPO by performing N MPS-MPO-MPS contractions. This results in a complexity of $\mathcal{O}(NL(dD\chi_{\text{max}}^3 + d^2D^2\chi_{\text{max}}^2))$ for an MPO with max bond dimension D [73, 16]. With the assumption that $\chi_{\text{max}} > d^2, D^2$ in most cases this is dominated by the cubic term, resulting in complexity

$$\mathcal{O}(NLdD\chi_{\text{max}}^3). \tag{91}$$

In comparison, using the density matrix $\rho(t)$ from the Lindbladian, calculating $\langle O(t)\rangle = \text{Tr}[\rho(t)O]$ has complexity $\mathcal{O}(d^{6L})$. Replacing the density matrix with the trajectories of the MCWF reduces this to $\mathcal{O}(Nd^{4L})$ while an MPDO requires $\mathcal{O}(Ld^2D_s^3)$.

9 Proof of equivalence of MCWF and Lindblad master equation

Theorem 3 (Equivalence of MCWF and Lindblad master equation)**.** *Given the Lindblad master equation as in Eq. (1) with solution $\rho(t)$, and the MCWF Hamiltonian defined as*

$$H = H_0 - \frac{i\hbar}{2}\sum_{m=1}^{k}\gamma_m L_m^\dagger L_m, \tag{92}$$

consider the following: Let $|\psi_i(t)\rangle$ for $i = 1, \ldots, N$ be state vector trajectories sampled from the initial state $\rho(0)$. The average of the outer products of these sampled pure states at time t is given by

$$\bar{\mu}_N(t) = \frac{1}{N}\sum_{i=1}^{N}|\psi_i(t)\rangle\langle\psi_i(t)|. \tag{93}$$

If the time step δt converges to 0, it holds that

$$\rho(t) = \lim_{N\to\infty}\bar{\mu}_N(t) \qquad \forall t. \tag{94}$$

Proof. For $N \to \infty$, a time-evolved state is described by combining the possibilities of a jump occurring or not such that

$$\begin{aligned}\bar{\mu}(t+\delta t) &= (1-\delta p)\frac{U(\delta t)\bar{\mu}(t)U^\dagger(\delta t)}{\sqrt{1-\delta p}\sqrt{1-\delta p}} + \sum_{m=1}^{k}\delta p_m\frac{L_m\bar{\mu}(t)L_m^\dagger}{\sqrt{\delta p_m}\sqrt{\delta p_m}/(\gamma_m\delta t)} \\ &= \bar{\mu}(t) - iH\delta t\bar{\mu}(t) + \bar{\mu}(t)iH^\dagger\delta t + \delta t\sum_{m=1}^{k}\gamma_m L_m\bar{\mu}(t)L_m^\dagger + \mathcal{O}(\delta t^2),\end{aligned} \tag{95}$$

where we have used the definition of the matrix exponential up to the second summand $U(\delta t) = e^{-i\delta tH} = 1 - iH\delta t + \mathcal{O}(\delta t^2)$. Taking the derivative such that the LHS of the master equation is created, it follows that

$$\begin{aligned}\frac{d}{dt}\bar{\mu}(t) &= \lim_{\delta t\to 0}\frac{\bar{\mu}(t+\delta t)-\bar{\mu}(t)}{\delta t}\\ &= \lim_{\delta t\to 0}\left(-iH\bar{\mu}(t)+\bar{\mu}(t)iH^{\dagger} \quad +\sum_{m=1}^{k}\gamma_m L_m\bar{\mu}(t)L_m^{\dagger}+\mathcal{O}(\delta t)\right)\\ &= -iH\bar{\mu}(t)+\bar{\mu}(t)iH^{\dagger} \quad +\sum_{m=1}^{k}\gamma_m L_m\bar{\mu}(t)L_m^{\dagger}+\lim_{\delta t\to 0}(\mathcal{O}(\delta t))\\ &= -iH\bar{\mu}(t)+\bar{\mu}(t)iH^{\dagger} \quad +\sum_{m=1}^{k}\gamma_m L_m\bar{\mu}(t)L_m^{\dagger}\\ &= -i\left(H_0-\frac{i\hbar}{2}\sum_{m=1}^{k}\gamma_m L_m^{\dagger}L_m\right)\bar{\mu}(t)+\bar{\mu}(t)i\left(H_0+\frac{i\hbar}{2}\sum_{m=1}^{k}\gamma_m L_m^{\dagger}L_m\right)+\sum_{m=1}^{k}\gamma_m L_m\bar{\mu}(t)L_m^{\dagger}\\ &= -i[H_0,\bar{\mu}(t)]-\sum_{m=1}^{k}\gamma_m\left(L_m\bar{\mu}(t)L_m^{\dagger}-\frac{1}{2}\left\{L_m^{\dagger}L_m,\bar{\mu}(t)\right\}\right).\end{aligned} \tag{96}$$

This has the form of the RHS of the master equation such that for a sufficiently small time step δt we have that

$$\rho(t)=\lim_{N\to\infty}\frac{1}{N}\sum_{j=1}^{N}|\psi_j(t)\rangle\langle\psi_j(t)|, \quad \forall t. \tag{97}$$

Since for a fixed $t\in[0,T]$ every sample $|\psi_j(t)\rangle, j=1,\dots,N$ is independent and identically distributed, it follows from the law of large numbers that $\rho(t)=\mathbb{E}[\bar{\mu}_N(t)]$ for all N,t. □

10 Frobenius variance

We present a lemma that is helpful in proving Monte Carlo convergence in the main text.

Lemma 4 (Frobenius variance). *Let $X,Y\in\mathbb{C}^{n\times n}$ be uncorrelated random matrices with $\mathbb{E}[X]=\mathbb{E}[Y]=A$ and the Fobenius norm for a squared complex matrix $A\in\mathbb{C}^{n,n}$ be given as $\|A\|_F=\sqrt{\mathit{Tr}(A^{\dagger}A)}=\sum_{i,j}|a_{i,j}|^2$. Then, the variance according to the Frobenius norm is given by*

$$\mathbb{V}_F[X]=\mathbb{E}\left[\|X-\mathbb{E}[X]\|_F^2\right], \tag{98}$$

and it holds true that

- *i)* $\mathbb{V}_F(X+Y)=\mathbb{V}_F(X)+\mathbb{V}_F(Y)$,
- *ii) for any scalar* $a\in\mathbb{R}$, $\mathbb{V}_F(aX)=a^2\mathbb{V}_F(X)$.

The proof of Lemma 4 is straightforward and is presented subsequently.

Definition 5 (Density matrix variance and standard deviation). *Let $\|\cdot\|$ be a matrix norm, and let $X\in\mathbb{C}^{n\times n}$ be a matrix-valued random variable defined on a probability space $(\Omega,\mathcal{F},\mathbb{P})$, where $\mathbb{P}$ is a probability measure. The variance of X with respect to the norm $\|\cdot\|$ is defined as*

$$\mathbb{V}[X]=\mathbb{E}\left[\|X-\mathbb{E}[X]\|^2\right], \tag{99}$$

where $\mathbb{E}[X]$ denotes the expectation of X. The expectation $\mathbb{E}[X]$ is computed entrywise with each entry being the expectation according to the respective marginal distributions of the entries. Specifically, for each $i, j \in \{1, \ldots, n\}$,

$$\mathbb{E}[X]_{i,j} = \mathbb{E}_{\mathbb{P}_{i,j}}[x_{i,j}], \tag{100}$$

where $x_{i,j}$ is the (i, j)-th entry of the matrix X, and $\mathbb{P}_{i,j}$ is the marginal distribution of $x_{i,j}$ induced by $\mathbb{P}$. The expectation value of the norm of a matrix $\mathbb{E}[\|\cdot\|]$ is defined as the multidimensional integral over the function $\|\cdot\| : \mathbb{C}^{n,n} \mapsto \mathbb{R}$ according to its marginal distributions $\mathbb{P}_{i,j}$. The standard deviation of X with respect to the norm $\|\cdot\|$ is then defined as

$$\sigma(X) = \sqrt{\mathbb{V}[X]} = \sqrt{\mathbb{E}\left[\|X - \mathbb{E}[X]\|^2\right]}. \tag{101}$$

In what follows, we make use of the Frobenius norm, defined for a squared complex matrix $A \in \mathbb{C}^{n,n}$ as $\|A\|_F := \sqrt{\mathrm{Tr}(A^\dagger A)} = \sum_{i,j} |a_{i,j}|^2$.

Lemma 6 (Frobenius variance)**.** *Let $X, Y \in \mathbb{C}^{n,n}$ be uncorrelated random matrices with $\mathbb{E}[X] = \mathbb{E}[Y] = A$. Then the variance according to the Frobenius norm is given as*

$$\mathbb{V}_F[X] = \mathbb{E}\left[\|X - \mathbb{E}[X]\|_F^2\right], \tag{102}$$

and it holds true that

i) $\mathbb{V}_F(X + Y) = \mathbb{V}_F(X) + \mathbb{V}_F(Y)$*,*

ii) for any scalar $a \in \mathbb{R}$, $\mathbb{V}_F(aX) = a^2\mathbb{V}_F(X)$.

Proof. First, we prove that $\mathbb{V}_F[X + Y] = \mathbb{V}_F[X] + \mathbb{V}_F[Y]$ when X and Y are uncorrelated random matrices. The variance for a random matrix X with respect to the Frobenius norm is defined as

$$\mathbb{V}_F[X] = \mathbb{E}[\|X - A\|_F^2] = \mathbb{E}\left[\sum_{i,j}^{n} |x_{i,j} - a_{i,j}|^2\right], \tag{103}$$

where $A = \mathbb{E}[X]$ and $a_{i,j}$ are the elements of A. Similarly, the variance for Y is

$$\mathbb{V}_F[Y] = \mathbb{E}[\|Y - A\|_F^2] = \mathbb{E}\left[\sum_{i,j}^{n} |y_{i,j} - a_{i,j}|^2\right]. \tag{104}$$

To find the variance of $X + Y$, note that $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y] = 2A$. Therefore,

$$\begin{aligned}\mathbb{V}_F[X + Y] &= \mathbb{E}\left[\|(X + Y) - \mathbb{E}[X + Y]\|_F^2\right] \\ &= \mathbb{E}\left[\sum_{i,j}^{n} |(x_{i,j} + y_{i,j}) - 2a_{i,j}|^2\right].\end{aligned} \tag{105}$$

Expanding the squared term, we have

$$\begin{aligned}|(x_{i,j} + y_{i,j}) - 2a_{i,j}|^2 &= |\left((x_{i,j} - a_{i,j}) + (y_{i,j} - a_{i,j})\right)|^2 \\ &= |x_{i,j} - a_{i,j}|^2 + |y_{i,j} - a_{i,j}|^2 \\ &\quad + 2\,\mathrm{Re}\left((x_{i,j} - a_{i,j})\overline{(y_{i,j} - a_{i,j})}\right).\end{aligned}$$

Taking the expectation and using the fact that X and Y are uncorrelated, we get

$$\begin{aligned}\mathbb{E}\left[|(x_{i,j}-a_{i,j})+(y_{i,j}-a_{i,j})|^2\right] &= \mathbb{E}\left[|x_{i,j}-a_{i,j}|^2\right]+\mathbb{E}\left[|y_{i,j}-a_{i,j}|^2\right] \\ &\quad + 2\mathbb{E}\left[\mathrm{Re}((x_{i,j}-a_{i,j})\overline{(y_{i,j}-a_{i,j})})\right].\end{aligned} \tag{106}$$

As with X and Y, $x_{i,j}$ and $y_{i,j}$ are also uncorrelated for all $i,j=1,\dots,n$ and hence $\mathbb{E}\left[\mathrm{Re}((x_{i,j}-a_{i,j})(y_{i,j}-a_{i,j}))\right]=0$. Thus, this reduces to

$$\mathbb{E}\left[(x_{i,j}-a_{i,j})^2\right]+\mathbb{E}\left[(y_{i,j}-a_{i,j})^2\right]. \tag{107}$$

Summing over all elements (i,j), we get

$$\begin{aligned}\mathbb{V}_F[X+Y] &= \sum_{i,j}^{n}\mathbb{E}\left[(x_{i,j}-a_{i,j})^2\right]+\sum_{i,j}^{n}\mathbb{E}\left[(y_{i,j}-a_{i,j})^2\right] \\ &= \mathbb{V}_F[X]+\mathbb{V}_F[Y].\end{aligned} \tag{108}$$

Next, we prove that $\mathbb{V}_F[aX]=a^2\mathbb{V}_F[X]$ for any scalar $a\in\mathbb{R}$. The variance for aX is

$$\mathbb{V}_F[aX]=\mathbb{E}[\|aX-\mathbb{E}[aX]\|_F^2]. \tag{109}$$

Since $\mathbb{E}[aX]=a\mathbb{E}[X]=aA$, we have

$$\mathbb{V}_F[aX]=\mathbb{E}\left[\|aX-aA\|_F^2\right]. \tag{110}$$

Factoring out a from the Frobenius norm, we get

$$\|aX-aA\|_F=|a|\|X-A\|_F, \tag{111}$$

and thus

$$\|aX-aA\|_F^2=a^2\|X-A\|_F^2. \tag{112}$$

Taking the expectation, we obtain

$$\mathbb{V}_F[aX]=\mathbb{E}[a^2\|X-A\|_F^2]=a^2\mathbb{E}[\|X-A\|_F^2]=a^2\mathbb{V}_F[X]. \tag{113}$$

Therefore, we have shown that for any scalar $a\in\mathbb{R}$,

$$\mathbb{V}_F[aX]=a^2\mathbb{V}_F[X]. \tag{114}$$

□

Theorem 7 (Convergence of TJM). *Let $d\in\mathbb{N}$ be the physical dimension and $L\in\mathbb{N}$ be the number of sites in the open quantum system described by the Lindblad master equation Eq.* (1). *Furthermore, let $\boldsymbol{\rho}_N(t)=\frac{1}{N}\sum_{j=1}^{N}|\boldsymbol{\Psi_j}(t)\rangle\langle\boldsymbol{\Psi_j}(t)|$ be the approximation of the solution $\rho(t)$ of the Lindblad master equation in MPO format at time $t\in[0,T]$ for some ending time $T>0$ and $N\in\mathbb{N}$ trajectories, where $|\boldsymbol{\Psi_j}(t)\rangle$ is a trajectory sampled according to the TJM in MPS format of full bond dimension. Then, the expectation value of the approximation of the corresponding density matrix $\rho_N(t)\in\mathbb{C}^{d^L,d^L}$ is given by $\rho(t)$ and there exists a $c>0$ such that the standard deviation of $\rho_N(t)$ can be upper bounded by*

$$\sigma(\rho_N(t))\leq\frac{c}{\sqrt{N}} \tag{115}$$

for all matrix norms $\|\cdot\|$ defined on $\mathbb{C}^{d^L,d^L}$.

Proof. For a sufficiently small time step δt, it can be shown that the average of the trajectories converges to the solution $\rho(t)$ of the Lindblad equation as the number N of trajectories approaches infinity, for all t, namely

$$\lim_{N\to\infty}\frac{1}{N}\sum_{j=1}^{N}|\psi_j(t)\rangle\langle\psi_j(t)| = \lim_{N\to\infty}\rho_N(t) = \rho(t), \quad t\in[0,T]. \tag{116}$$

From Theorem 3 we know that $\rho(t) = \mathbb{E}[\rho_N(t)]$ for all N, t. Additionally, let $X(t) = X_1(t)$. We know that $\mathbb{V}_F[X(t)]$ is bounded since every realization of $X(t)$ and $\mathbb{E}[X(t)]$ are density matrices, so they have trace 1. Thus, we have that $0 \leq \|\rho_1 - \rho_2\|_F \leq 2$ for all density matrices $\rho_1, \rho_2 \in \mathcal{B}(\mathcal{H})$ regardless of the system size.

To get a realization $X(t)$ for a certain $t \in [0,T]$, we simulate a trajectory from $X(0)$ by choosing a sufficiently small discretization δt and stop the simulation at $X(t)$. Now it is easy to check that the variance of $\mathbb{V}_F[\rho_N(t)]$ decreases linearly with N. Concretely,

$$\begin{aligned}\mathbb{V}_F[\rho_N(t)] &= \mathbb{V}_F\left[\frac{1}{N}\sum_{i=1}^{N}X_i(t)\right] = \frac{1}{N^2}\mathbb{V}_F\left[\sum_{i=1}^{N}X_i(t)\right]\\ &= \frac{1}{N^2}\sum_{i=1}^{N}\mathbb{V}_F\left[X_i(t)\right] = \frac{1}{N}\mathbb{V}_F[X(t)] \qquad (117)\\ &\leq \frac{4}{N}, \qquad (118)\end{aligned}$$

where we have used that $\mathbb{V}_F[aX(t)] = a^2\mathbb{V}_F[X(t)]$, $\mathbb{V}_F[X_1(t)+X_2(t)] = \mathbb{V}_F[X_1(t)]+\mathbb{V}_F[X_2(t)]$ and the fact that the $X_i(t)$ are independent samples and identically distributed. Thus, the Frobenius norm standard deviation is upper bounded by

$$\sigma_F[\rho_N(t)] = \frac{1}{\sqrt{N}}\sigma_F[X(t)] \leq \frac{2}{\sqrt{N}}. \tag{119}$$

By the equivalence of norms on finite vector spaces, there exists $c_1, c_2 \in \mathbb{R}$ such that $c_1\|A\|_F \leq \|A\| \leq c_2\|A\|_F$ for all complex square matrices A and all matrix norms $\|\cdot\|$. Thus, the convergence rate $\mathcal{O}(1/\sqrt{N})$ also holds true in trace norm and any other relevant matrix norm and is independent of system size. The proposition follows directly. □

11 Complexity of TJM

For a TJM procedure we sample $N \in \mathbb{N}$ trajectories, each with $n = \frac{T}{\delta t} \in \mathbb{N}$ time steps, where $T \in \mathbb{R}_+$ is the terminal time and δt the time step size. Since each time step consists of the calculation of the probability over the jump operators, jump application, TDVP step, and a dissipative contraction, the total complexity can be calculated as

$$\mathcal{O}(Nn(\text{probability distribution} + \text{ jump application} + \text{TDVP} + \text{Dissipative contraction})). \tag{120}$$

For the complexity calculation, we consider χ_{max} as the maximum bond dimension of the MPS $|\boldsymbol{\Psi(t)}\rangle$, d the dimension of the local Hilbert space, D as the maximum bond dimension of the MPO representing the Hamiltonian H_0 of the closed system.

The calculation of the probability distribution can be seen as an efficient sweep across the MPS $|\boldsymbol{\Psi(t)}\rangle$, where at each site $\ell = 1, \ldots, L$ we have to contract the jump operators $L_j^{[\ell]}, j \in S(\ell)$ into the ℓ-th site tensor and calculate the inner product of the MPS which requires $\mathcal{O}(k\chi_{\text{max}}^3 d)$ operations since it has to be done for every jump operator $L_m, m = 1, \ldots, k$. The sampling of an $\epsilon \in [0, 1]$, which is uniformly distributed has complexity $\mathcal{O}(1)$. It is like sampling a jump operator according to the distribution $\Pi(t)$ [74]. Since L_m are single-site operators, the jump application is just a contraction of a matrix in $\mathbb{C}^{d\times d}$ into a site tensor of $|\boldsymbol{\Psi(t)}\rangle$, which takes $\mathcal{O}(\chi_{\text{max}}^2 d^2)$ operations.

The complexity of the 2TDVP is given by $\mathcal{O}(L(\chi_{\text{max}}^2 d^3 D^2 + \chi_{\text{max}}^3 d^2 D + \chi_{\text{max}}^3 d^3))$, whereas the single-site version scales with $\mathcal{O}(L(\chi_{\text{max}}^2 d^2 D^2 + \chi_{\text{max}}^3 dD + \chi_{\text{max}}^3 d^2))$. Since in TJM the 2TDVP is not performed with maximum bond dimension χ_{max}, the 1TDVP complexity is of primary interest. In the dissipative contraction $\mathcal{D}[\delta t]$, each of the L site tensors has to be contracted with a single site tensor $D_\ell \in \mathbb{C}^{d,d}$, which requires $\mathcal{O}(d^2\chi_{\text{max}}^2)$ operations, leading to a complexity of $\mathcal{O}(Ld^2\chi_{\text{max}}^2)$ for the dissipative contraction.

Collecting the above considerations, the total complexity of the TJM is given as

$$\begin{aligned}\mathcal{O}\bigg(& Nn\big[L(d^2D^2\chi_{\text{max}}^2 + dD\chi_{\text{max}}^3 + d^2\chi_{\text{max}}^3 \\ &+ d^2\chi_{\text{max}}^2) + kd\chi_{\text{max}}^3 + d^2\chi_{\text{max}}^2\big]\bigg).\end{aligned} \tag{121}$$

Dominant terms are the squared physical dimension d, the cubic bond dimension χ_{max} of the MPS in the TDVP sweep and the product dD of the physical dimension and the bond dimension of the MPO, such that the shorthand complexity of TJM is

$$\mathcal{O}\bigg(NnL\chi_{\text{max}}^3\big[dD + d^2\big]\bigg). \tag{122}$$

Acknowledgments

We would like to thank Phillipp Trunschke for very useful discussions. This work has been supported by the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus, and for which this work reports on results of a joint-node collaboration involving the FU Berlin and two teams at the TU Munich. It has also been funded by the the European Union's Horizon 2020 Quantum Flagship innovation program Millenion (grant agreement No. 101114305) for which this again constitutes joint node work by the FU Berlin and the TU Munich, and the Einstein Research Unit on Quantum Devices (for which this work reflects once more joint work now involving the Weierstrass Institute, the Zuse Institute and the FU Berlin). The team at Technische Universität München has received additional funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 101001318). The team at the Freie Universität Berlin has been additionally supported by the DFG (CRC 183, FOR 2724), the BMBF (MuniQC-Atoms), the European Union's Horizon 2020 Quantum Flagship innovation program PasQuans2 (grant agreement No. 101113690), Berlin Quantum, and the European Research Council (ERC DebuQC).

References

[1] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix product state representations. *Quantum Info. Comput.*, 7(5):401–430, July 2007.

[2] J. Ignacio Cirac, David Pérez-García, Norbert Schuch, and Frank Verstraete. Matrix product states and projected entangled pair states: Concepts, symmetries, theorems. *Rev. Mod. Phys.*, 93:045003, Dec 2021.

[3] I. V. Oseledets. Tensor-train decomposition. *SIAM J. Sc. Comp.*, 33(5):2295–2317, 2011.

[4] J. I. Cirac and P. Zoller. Goals and opportunities in quantum simulation. *Nature Phys.*, 8:264, 2012.

[5] S. Trotzky, Y.-A. Chen, A. Flesch, I. P. McCulloch, U. Schollwöck, J. Eisert, and I. Bloch. Probing the relaxation towards equilibrium in an isolated strongly correlated one-dimensional Bose gas. *Nature Phys.*, 8:325–330, 2012.

[6] N. Schuch, M. M. Wolf, K. G. H. Vollbrecht, and J. I. Cirac. On entropy growth and the hardness of simulating time evolution. *New J. Phys.*, 10:033032, 2008.

[7] M. Cramer, C. M. Dawson, J. Eisert, and T. J. Osborne. Exact relaxation in a class of non-equilibrium quantum lattice systems. *Phys. Rev. Lett.*, 100:030602, 2008.

[8] Román Orús. Tensor networks for complex quantum systems. *Nature Rev. Phys.*, 1(9):538–550, August 2019.

[9] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Ann. Phys.*, 326(1):96–192, January 2011.

[10] J. Eisert, M. Cramer, and M. B. Plenio. Area laws for the entanglement entropy. *Rev. Mod. Phys.*, 82:277, 2010.

[11] Guifré Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Phys. Rev. Lett.*, 93(4):040502, July 2004.

[12] F. Verstraete and J. I. Cirac. Matrix product states represent ground states faithfully. *Phys. Rev. B*, 73:094423, March 2006.

[13] Jutho Haegeman, J. Ignacio Cirac, Tobias J. Osborne, and Iztok Pižorn. Time-dependent variational principle for quantum lattices. *Phys. Rev. Lett.*, 107(7):070601, August 2011.

[14] C. M. Dawson, J. Eisert, and T. J. Osborne. Unifying variational methods for simulating quantum many-body systems. *Phys. Rev. Lett.*, 100:130501, 2008.

[15] Jutho Haegeman, Christian Lubich, Ivan Oseledets, Bart Vandereycken, and Frank Verstraete. Unifying time evolution and optimization with matrix product states. *Phys. Rev. B*, 94:165116, 2016.

[16] Sebastian Paeckel, Thomas Köhler, Andreas Swoboda, Salvatore R. Manmana, Ulrich Schollwöck, and Claudius Hubig. Time-evolution methods for matrix-product states. *Ann. Phys.*, 411:167998, December 2019.

[17] Yiqing Zhou, E. Miles Stoudenmire, and Xavier Waintal. What limits the simulation of quantum computers? *Phys. Rev. X*, 10:041038, Nov 2020.

[18] Song Cheng, Chenfeng Cao, Chao Zhang, Yongxiang Liu, Shi-Yao Hou, Pengxiang Xu, and Bei Zeng. Simulating noisy quantum circuits with matrix product density operators. *Phys. Rev. Res.*, 3:023005, Apr 2021.

[19] Zihan Cheng and Matteo Ippoliti. Efficient sampling of noisy shallow circuits via monitored unraveling. *PRX Quantum*, 4:040326, Nov 2023.

[20] 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, Nov 2004.

[21] Michael Zwolak and Guifré Vidal. Mixed-state dynamics in one-dimensional quantum lattice systems: A time-dependent superoperator renormalization algorithm. *Phys. Rev. Lett.*, 93:207205, Nov 2004.

[22] A. H. Werner, D. Jaschke, P. Silvi, M. Kliesch, T. Calarco, J. Eisert, and S. Montangero. Positive tensor network approach for simulating open quantum many-body systems. *Phys. Rev. Lett.*, 116:237201, 2016.

[23] Inés de Vega and Mari-Carmen Bañuls. Thermofield-based chain-mapping approach for open quantum systems. *Phys. Rev. A*, 92:052116, Nov 2015.

[24] Hendrik Weimer, Augustine Kshetrimayum, and Román Orús. Simulation methods for open quantum many-body systems. *Rev. Mod. Phys.*, 93(1):015008, March 2021.

[25] G. Lindblad. On the generators of quantum dynamical semigroups. *Comm. Math. Phys.*, 48:119–130, 1976.

[26] Vittorio Gorini, Andrzej Kossakowski, and E. C. G. Sudarshan. Completely positive dynamical semigroups of n-level systems. *J. Math. Phys.*, 17(5):821–825, 1976.

[27] K. Mølmer, K. Berg-Sørensen, Y. Castin, and J. Dalibard. A Monte Carlo wave function method in quantum optics. In *Optical Society of America Annual Meeting*, page MFF1. Optica Publishing Group, 1992.

[28] R. Dum, A. S. Parkins, P. Zoller, and C. W. Gardiner. Monte carlo simulation of master equations in quantum optics for vacuum, thermal, and squeezed reservoirs. *Phys. Rev. A*, 46:4382–4396, Oct 1992.

[29] Jean Dalibard, Yvan Castin, and Klaus Mølmer. Wave-function approach to dissipative processes in quantum optics. *Phys. Rev. Lett.*, 68:580–583, Feb 1992.

[30] H. J. Carmichael. *An open systems approach to quantum optics*. Springer-Verlag, Berlin Heidelberg, 1993.

[31] R. L. Hudson and K. R. Parthasarathy. Quantum Ito's formula and stochastic evolutions. *Comm. Math. Phys.*, 93(3):301–323, September 1984.

[32] Howard M. Wiseman and Gerard J. Milburn. *Quantum measurement and control*. Cambridge University Press, 2009.

[33] M. Kliesch, Gross, and J. Eisert. Matrix product operators and states: Np-hardness and undecidability. *Phys. Rev. Lett.*, 113:160503, 2014.

[34] Jian Cui, J. Ignacio Cirac, and Mari Carmen Bañuls. Variational matrix product operators for the steady state of dissipative quantum systems. *Phys. Rev. Lett.*, 114:220601, Jun 2015.

[35] Zi Cai and Thomas Barthel. Algebraic versus exponential decoherence in dissipative many-particle systems. *Phys. Rev. Lett.*, 111:150403, Oct 2013.

[36] Moritz Reh, Markus Schmitt, and Martin Gärttner. Time-dependent variational principle for open quantum systems with artificial neural networks. *Phys. Rev. Lett.*, 127:230501, Dec 2021.

[37] H. Pichler, A. J. Daley, and P. Zoller. Nonequilibrium dynamics of bosonic atoms in optical lattices: Decoherence of many-body states due to spontaneous emission. *Phys. Rev. A*, 82:063605, Dec 2010.

[38] F. W. G. Transchel, A. Milsted, and Tobias J. Osborne. A Monte Carlo time-dependent variational principle. *arXiv:1411.5546*, 2014.

[39] Marlon Brenes, Juan José Mendoza-Arenas, Archak Purkayastha, Mark T. Mitchison, Stephen R. Clark, and John Goold. Tensor-network method to simulate strongly interacting quantum thermal machines. *Phys. Rev. X*, 10:031040, Aug 2020.

[40] Archak Purkayastha, Giacomo Guarnieri, Steve Campbell, Javier Prior, and John Goold. Periodically refreshed baths to simulate open quantum many-body dynamics. *Phys. Rev. B*, 104:045417, Jul 2021.

[41] M. Moroder, M. Grundner, F. Damanet, U. Schollwöck, S. Mardazad, S. Flannigan, T. Köhler, and S. Paeckel. Stable bipolarons in open quantum systems. *Phys. Rev. B*, 107:214310, Jun 2023.

[42] Zhaoxuan Xie, Mattia Moroder, Ulrich Schollwöck, and Sebastian Paeckel. Photo-induced dynamics with continuous and discrete quantum baths. *The Journal of Chemical Physics*, 161(7), 2024.

[43] A J Daley, C Kollath, U Schollwöck, and G Vidal. Time-dependent density-matrix renormalization-group using adaptive effective hilbert spaces. *J. Stat. Mech.*, 2004(04):P04005, April 2004.

[44] G. Lindblad. On the generators of quantum dynamical semigroups. *Comm. Math. Phys.*, 48(2):119 – 130, 1976.

[45] Heinz-Peter Breuer and Francesco Petruccione. *The theory of open quantum systems*. Oxford University Press, 01 2007.

[46] Daniel Manzano. A short introduction to the Lindblad master equation. *AIP Adv.*, 10:025106, 6 2020.

[47] Emiliano Godinez-Ramirez, Richard Milbradt, and Christian B. Mendl. A Riemannian approach to the Lindbladian dynamics of a locally purified tensor network. *arXiv:2409.08127*, 2024.

[48] H. F. Trotter. On the product of semi-groups of operators. *Proc. Am. Math. Soc.*, 10(4):545–551, 1959.

[49] Masuo Suzuki. Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Comm. Math. Phys.*, 51(2):183–190, June 1976.

[50] S. Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, August 1996.

[51] Michael Müger. Notes on the theorem of Baker-Campbell-Hausdorff-Dynkin. 2019.

[52] V. Gradinaru. Strang splitting for the time-dependent Schrödinger equation on sparse grids. *SIAM J. Num. An.*, 46:103–123, 2008.

[53] Tobias Jahnke and Christian Lubich. Error bounds for exponential operator splittings. *BIT*, 40:735–744, 01 2000.

[54] Christian Lubich. *From Quantum to Classical Molecular Dynamics: Reduced Models and Numerical Analysis*. Zurich Lectures in Advanced Mathematics. EMS Press, 1 edition, September 2008.

[55] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Stand.*, 45(4):255, October 1950.

[56] Michael P. Zaletel, Roger S. K. Mong, Christoph Karrasch, Joel E. Moore, and Frank Pollmann. Time-evolving a matrix product state with long-ranged interactions. *Phys. Rev. B*, 91(16):165112, April 2015.

[57] Haggai Landa and Grégoire Misguich. Nonlocal correlations in noisy multiqubit systems simulated using matrix product operators. *SciPost Physics Core*, 6(2):037, May 2023.

[58] Jutho Haegeman, Tobias J. Osborne, and Frank Verstraete. Post-matrix product state methods: To tangent space and beyond. *Phys. Rev. B*, 88(7):075133, August 2013.

[59] Aaron Sander. Yaqs: Yet another quantum simulator. `https://github.com/cda-tum/mqt-yaqs`.

[60] R. Wille, L. Berent, T. Forster, J. Kunasaikaran, K. Mato, T. Peham, N. Quetschlich, D. Rovara, A. Sander, L. Schmid, D. Schonberger, Y. Stade, and L. Burgholzer. The MQT handbook : A summary of design automation tools and software for quantum computing. In *International Conference on Quantum Software*, pages 1–8. IEEE, 2024.

[61] J. R. Johansson, P. D. Nation, and Franco Nori. QuTiP: An open-source Python framework for the dynamics of open quantum systems. *Comp. Phys. Comm.*, 183(8):1760–1772, August 2012.

[62] J. R. Johansson, P. D. Nation, and Franco Nori. QuTiP 2: A Python framework for the dynamics of open quantum systems. *Comp. Phys. Comm.*, 184(4):1234–1240, April 2013.

[63] Jacob C. Bridgeman and Christopher T. Chubb. Hand-waving and interpretive dance: An introductory course on tensor networks. *J. Phys. A*, 50(22):223001, June 2017.

[64] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863, November 1992.

[65] C. Hubig, I. P. McCulloch, U. Schollwöck, and F. A. Wolf. Strictly single-site DMRG algorithm with subspace expansion. *Phys. Rev. B*, 91:155115, Apr 2015.

[66] B. Pirvu, V. Murg, J. I. Cirac, and F. Verstraete. Matrix product operator representations. *New J. Phys.*, 12(2):025012, February 2010.

[67] Paolo Zanardi, Christof Zalka, and Lara Faoro. Entangling power of quantum evolutions. *Phys. Rev. A*, 62(3):030301, August 2000.

[68] Cheryne Jonay, David A. Huse, and Adam Nahum. Coarse-grained dynamics of operator and state entanglement. *arXiv:1803.00089*, 2018.

[69] 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(20):207204, November 2004.

[70] Steven R. White and Adrian E. Feiguin. Real-time evolution using the density matrix renormalization group. *Phys. Rev. Lett.*, 93:076401, August 2004.

[71] Mingru Yang and Steven R. White. Time-dependent variational principle with ancillary krylov subspace. *Phys. Rev. B*, 102(9), September 2020.

[72] A. J. Dunnett. angusdunnett/mpsdynamics: version v1.0, July 2021.

[73] Angus J. Dunnett and Alex W. Chin. Dynamically evolving bond-dimensions within the one-site time-dependent-variational-principle method for matrix product states: Towards efficient simulation of non-equilibrium open quantum dynamics, 2020.

[74] Alastair J. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Soft.*, 3(3):253–256, 1977.