The Journal of Chemical Physics

Computing time-periodic steady-state currents via the time evolution of tensor network states

Cite as: J. Chem. Phys. **157**, 054104 (2022); https://doi.org/10.1063/5.0099741
Submitted: 18 May 2022 • Accepted: 27 June 2022 • Accepted Manuscript Online: 04 July 2022 • Published Online: 04 August 2022

Nils E. Strand, Hadrien Vroylandt and Todd R. Gingrich

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

Computing time-periodic steady-state currents via the time evolution of tensor network states

Cite as: J. Chem. Phys. 157, 054104 (2022); doi: 10.1063/5.0099741
Submitted: 18 May 2022 • Accepted: 27 June 2022 •
Published Online: 4 August 2022

Nils E. Strand, Hadrien Vroylandt,[a] and Todd R. Gingrich[b]

AFFILIATIONS

Department of Chemistry, Northwestern University, 2145 Sheridan Road, Evanston, Illinois 60208, USA

Note: This paper is part of the 2022 JCP Emerging Investigators Special Collection.
[a]**Now at:** ISCD, Sorbonne Université, Paris, France.
[b]**Author to whom correspondence should be addressed:** todd.gingrich@northwestern.edu

ABSTRACT

We present an approach based upon binary tree tensor network (BTTN) states for computing steady-state current statistics for a many-particle 1D ratchet subject to volume exclusion interactions. The ratcheted particles, which move on a lattice with periodic boundary conditions subject to a time-periodic drive, can be stochastically evolved in time to sample representative trajectories via a Gillespie method. In lieu of generating realizations of trajectories, a BTTN state can variationally approximate a distribution over the vast number of many-body configurations. We apply the density matrix renormalization group algorithm to initialize BTTN states, which are then propagated in time via the time-dependent variational principle (TDVP) algorithm to yield the steady-state behavior, including the effects of both typical and rare trajectories. The application of the methods to ratchet currents is highlighted, but the approach extends naturally to other interacting lattice models with time-dependent driving. Although trajectory sampling is conceptually and computationally simpler, we discuss situations for which the BTTN TDVP strategy can be beneficial.

I. INTRODUCTION

Over the past few decades, tensor networks (TNs) have emerged as one of the most powerful mathematical tools for numerically manipulating high-dimensional quantum states. The computational power is achieved because tensor networks dramatically shrink the dimensionality of quantum states and operators by decomposing intractably large vectors and matrices into a composition of smaller tensors that serve as an attractive approximation method. Furthermore, by adjusting the dimensionality of auxiliary indices that link the tensors, tensor networks enable variational calculations with controllable errors. In the limit of high-dimensional auxiliary indices, exact results are recovered, but the practical benefit is gained by reducing the dimensionality to obtain approximate results at a dramatically lower computational expense. For example, the density matrix renormalization group (DMRG) algorithm[1] is widely used to converge low-entanglement many-body quantum ground states by sweeping through a tensor network while performing computationally tractable local optimizations. Dynamics of quantum states can similarly be approximated via the time-dependent variational principle (TDVP),[2,3] which also proceeds by a sweep of local operations on tractable tensors.[4,5]

While tensor networks were initially applied to quantum systems, their use has also been extended to classical stochastic systems.[6–14] More specifically, tensor network methods have been used to compute large deviation functions, which measure the probability of dynamical fluctuations both near equilibrium and far from equilibrium. Helms *et al.* recently identified dynamical phase transitions separating jamming and flowing phases within the 1D and 2D asymmetric simple exclusion processes (ASEP). These studies used the matrix product state (MPS), a 1D chain of tensors, and the projected entangled pair state (PEPS), the 2D analog of the MPS, to probe the thermodynamic limits of the ASEP for 1D and 2D systems, respectively.[6,8] In another study by Bañuls and Garrahan, the DMRG was used to compute trajectory-space phase transitions in the activity of kinetically constrained models.[7] Still, the use of tensor networks to evaluate classical stochastic dynamics remains relatively unexplored. A major complication is that the relevant stochastic operators, unlike quantum operators, are almost always non-Hermitian, and compared to Hermitian operators,

diagonalizing and exponentiating these non-Hermitian operators is more demanding and more prone to numerical instabilities. Consequently, iterative tensor network procedures, such as the DMRG and TDVP, can present numerical complications when applied to non-symmetric operators.[15]

Here, we show that classical stochastic dynamics can nevertheless be robustly propagated by non-Hermitian operators via the TDVP using a binary tree tensor network (BTTN).[16–18] The TDVP approach offers an unexplored route to analyze the impact of many-particle interactions in noise-driven ratchets. We had previously analyzed the behavior of a single-particle 2D ratchet under time-periodic driving by discretizing space to obtain a Markovian approximation to the continuum dynamics. This discrete state Markov dynamics was amenable to spectral computations of the current via large-deviation methods.[19] Here, we leverage the TDVP with the BTTN architecture, extending this approach to compute the current generated by multiple interacting particles moving on a 1D lattice subject to a time-dependent ratcheting potential. The approach has been discussed at a high level in our recent Communication;[20] here, we focus on the technical details that allow the TDVP/BTTN methodology to compute the statistical properties of currents in time-periodic steady states[21–24] in the presence of interactions.[25] In particular, we illustrate that the calculations recapitulate Monte Carlo samples generated using the Gillespie algorithm while avoiding the sampling noise. Crucially, the TDVP/BTTN calculations agree with sampled trajectories in both the mean behavior and fluctuations. These current fluctuations are accessible at essentially no additional cost to the TN approach, whereas sampled trajectories grow increasingly more costly as the targeted fluctuations become rarer.

II. METHODS

A. 1D ratchet model

We set out to study a discretized 1D flashing ratchet with periodic boundary conditions, a tunable driving frequency, and a variable number of particles that interact through volume exclusion. Many previous investigations of single-particle 1D and 2D ratchets motivated the particular form of the ratcheting potentials.[19,26–29] The focus on interactions has some precedent. Of particular note, Kedem and Weiss simulated trajectories of many electrons interacting via a Coulomb potential in a 2D ratchet.[30] As in that work, our transported particles are subject to a spatiotemporal potential as follows:

$$U(x,t) = X(x)T(t), \tag{1}$$

where $T(t)$ and $X(x)$ are periodic in time and in space, respectively. For the flashing ratchet model, $T(t)$ is a square wave with period τ and amplitude $V_{\max}$ that toggles between *on* and *off* stages,

$$T(t) = \begin{cases} -V_{\max}, & 0 \le t < \dfrac{\tau}{2}, \\ 0, & \dfrac{\tau}{2} \le t < \tau. \end{cases} \tag{2}$$

Following the setup from Ref. 19, the spatial potential is a superposition of two sinusoids,

$$X(x) = \frac{a_1}{2}\sin\left(\frac{2\pi x}{x_{\max}}\right) + \frac{a_2}{2}\sin\left(\frac{4\pi x}{x_{\max}}\right), \tag{3}$$

where $x_{\max}$ is the length of the repeating unit and a_1 and a_2 sculpt the potential. We assign $a_1 = 1$ and $a_2 = 0.25$.

The setup is very similar to our previous single-particle work,[19] so we highlight two important distinctions. First, we are now considering a 1D ratchet with particles that can only move along the x direction; particles in our earlier work generated current along that same x direction but could additionally move along another dimension. Second, our temporal function $T(t)$ toggles between $V_{\max}$ and 0, not between $V_{\max}$ and $-V_{\max}$. For a 1D ratchet, this move from a symmetric square wave temporal drive to a flashing ratchet is needed to generate nonvanishing current.[31] Otherwise, any motion occurring within the first half of the period would be offset by motion in the opposite direction during the second half of the period. As illustrated in Fig. 1, the flashing ratchet generates current in the negative direction owing to the asymmetric sawtooth form of $U(x,t)$.

B. Time-periodic steady-state currents

By coarse graining the 1D ratchet in space, the dynamics is modeled as a nearest-neighbor Markov jump process on a periodically replicated lattice of N sites with grid spacing h. This jump process obeys the master equation as follows:

$$\frac{\partial|p\rangle}{\partial t} = \mathsf{W}(t)|p\rangle, \tag{4}$$

where $\mathsf{W}(t)$ is the time-dependent rate operator and $|p\rangle$ is the state vector consisting of the probabilities of each possible system configuration. The time-dependent $\mathsf{W}(t)$ toggles between two distinct sets of rates with a period τ,

$$\mathsf{W}(t) = \begin{cases} \mathsf{W}_1, & 0 \le t < \tau/2, \\ \mathsf{W}_2, & \tau/2 \le t < \tau. \end{cases} \tag{5}$$

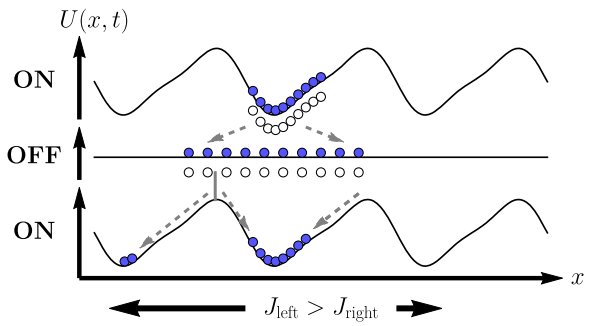

FIG. 1. Schematic of the 1D flashing ratchet. The driving protocol consists of a potential energy landscape that toggles between the *on* and *off* states with period τ. From the initial time to $\tau/2$, particles concentrate in the bottom of a well. During the subsequent half period, these particles diffuse outward on a flat landscape before again settling in the wells during the next period. The asymmetric potential generates current in the $-x$ direction, with the current magnitude depending on the frequency of flashing.

In the first half of the period, dynamics evolves on the sawtooth landscape analogous to Ref. 19, so W_1 is a rate matrix permitting nearest-neighbor hops from site i to site $i \pm 1$ with rate

$$r_{1,i\to i\pm1} = \pm\frac{V_{\max}X'(x)}{2h} + \frac{D}{h^2}, \tag{6}$$

provided that site $i \pm 1$ is vacant. The prime denotes the derivative with respect to x. In the continuum $h \to 0$ limit, the parameter D becomes the diffusion constant of the associated overdamped single-particle Langevin dynamics.[32] In the second half of the period, the potential is turned off and the evolution proceeds on a flat landscape. The rate matrix W_2 permits the same volume-excluding nearest-neighbor hops, but the rates of these hops are now $r_{2,i\to i\pm1} = D/h^2$. Consistent with Ref. 19, we set $V_{\max}$ to 0.1 and D is given the value of 12.64.

The long-time limit of Eq. (4) approaches the time-periodic steady-state vector $|\pi\rangle_t$ on the time interval $t \in [0, \tau]$. When the operators W_1 and W_2 are cast as matrices, $|\pi\rangle_0$ is simply obtained as the leading eigenvector of the full–period transition matrix $\mathsf{T} \equiv e^{\tau\mathsf{W}_2/2}e^{\tau\mathsf{W}_1/2}$. In this work, we seek period-averaged macroscopic currents around the ring, constructed in terms of the time-dependent currents from site j to site i, $j_{ij}(t)$,

$$\bar{\jmath} = \frac{1}{\tau}\int_0^\tau \mathrm{d}t \sum_{i,j} d_{ij}j_{ij}(t), \tag{7}$$

where the weights

$$d_{ij} = \begin{cases} +1, & j \text{ directly left of } i, \\ -1, & j \text{ directly right of } i, \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

pick out the oriented nearest-neighbor transitions. To characterize the mean and variance of these currents at the time-periodic steady state, we define the scaled cumulant-generating function (SCGF) $\psi(\lambda)$ as

$$\psi(\lambda) := \lim_{n\to\infty}\frac{1}{n}\ln\langle e^{\lambda n\bar{\jmath}}\rangle_n, \tag{9}$$

where n is the number of driving periods. The first and second derivatives of $\psi(\lambda)$, evaluated at $\lambda = 0$, yield the mean and variance of the current.[33] It is known that $\psi(\lambda)$ can be obtained from the largest eigenvalue of a product of matrix exponentials[33–36] as

$$\psi(\lambda) = \frac{1}{\tau}\ln\max\mathrm{eig}\left(e^{\mathsf{W}_2(\lambda)\tau/2}e^{\mathsf{W}_1(\lambda)\tau/2}\right), \tag{10}$$

with the so-called tilted rate operators $\mathsf{W}_k(\lambda)$ defined in terms of the original equation (5) rate operators as

$$[\mathsf{W}_k(\lambda)]_{ij} := [\mathsf{W}_k]_{ij}e^{\lambda d_{ij}}. \tag{11}$$

When evolving dynamics of many interacting particles, the matrix representation of the titled operator becomes untenable due to the exponential growth of the state space. Therefore, it is impractical to directly compute the product of matrix exponentials in Eq. (10). Instead, we can start with an arbitrary state vector at time zero. This initial state can be numerically propagated in time by $\mathsf{W}_1(\lambda)$ for half a period and then propagated by $\mathsf{W}_2(\lambda)$ for another half a period. This time propagation is continued until the time-periodic steady state is reached, at which point the SCGF is deduced from

$$|\pi(\lambda)\rangle_\tau = \exp(\psi(\lambda)\tau)|\pi(\lambda)\rangle_0, \tag{12}$$

with $|\pi(\lambda)\rangle_t$ being the time-periodic steady state subject to exponential bias λ. The advantage of this dynamical approach is that it can be practically implemented for many-body dynamics when the time evolution is approximated using the TDVP algorithm. This algorithm, which projects the natural dynamics onto a subspace defined by a tensor network (TN) ansatz, leverages the expression of $\mathsf{W}_k(\lambda)$ in terms of local operators acting on each lattice site. In this occupation basis, or *second quantized form*,

$$\begin{aligned}\mathsf{W}_k(\lambda) = &\sum_{i=1}^{N} r_{k,i\to i+1}(e^{\lambda}\mathbf{a}_i\mathbf{a}_{i+1}^\dagger - \mathbf{n}_i\mathbf{v}_{i+1}) \\ &+ \sum_{i=1}^{N} r_{k,i+1\to i}(e^{-\lambda}\mathbf{a}_i^\dagger\mathbf{a}_{i+1} - \mathbf{v}_i\mathbf{n}_{i+1}),\end{aligned} \tag{13}$$

where $\mathbf{a}_i$, $\mathbf{a}_i^\dagger$, $\mathbf{n}_i$, and $\mathbf{v}_i$ are annihilation, creation, particle number, and vacancy number operators at site i, respectively. Note that the periodic boundary conditions lead us to associate $N + 1 \equiv 1$. Because the second quantized operator only involves nearest-neighbor interactions, it can be expressed as a product of operator-valued matrices, one per site of the lattice. These operator-valued matrices, discussed explicitly in Appendix A and symbolically represented by gray circles in Fig. 2(a), allow Eq. (13) to be efficiently computed as the product of the operator-valued matrices, i.e., a matrix product operator (MPO).[37] If the state $|p\rangle$ is similarly decomposed into a local site representation, the action of $\mathsf{W}_k(\lambda)$ on $|p\rangle$ can be calculated even when the matrix form of $\mathsf{W}_k(\lambda)$ is too large to explicitly construct. For example, $\mathsf{W}_k(\lambda)$ would be a roughly $10^9 \times 10^9$ matrix for the 32-site lattice with 16 particles discussed in Sec. IV D.

C. Tensor networks

To leverage the MPO, we express the state vector $|p\rangle$ from Eq. (4) in terms of a product basis of local basis states $|s_i\rangle$ on site i as

$$|p\rangle = \sum_{s_1,\ldots,s_N} c_{s_1,\ldots,s_N}|s_1,\ldots,s_N\rangle. \tag{14}$$

The rank-N tensor c depends on the N physical indices $s_1,\ldots,s_N$, but c is so high dimensional that it cannot be practically computed. Rather, we introduce a tensor network (TN) ansatz in which c is generated by a network of tensors $A^{(1)}, A^{(2)}, \ldots, A^{(\chi)}$, each with modest rank. These tensors can depend on some of the physical indices $(s_1,\ldots,s_N)$ reflecting the occupation at each site and some auxiliary indices that will be summed over. We adopt the nomenclature that S_i is a set of physical indices upon which the ith tensor depends (potentially an empty set) and Q_i the set of auxiliary indices. The tensor product ansatz is a restriction that we only allow expansion coefficients of the form

$$c_{s_1,\ldots,s_N} = \sum_{Q_1,\ldots,Q_\chi} A^{(1)}_{Q_1,S_1}A^{(2)}_{Q_2,S_2},\ldots,A^{(\chi)}_{Q_\chi,S_\chi}, \tag{15}$$

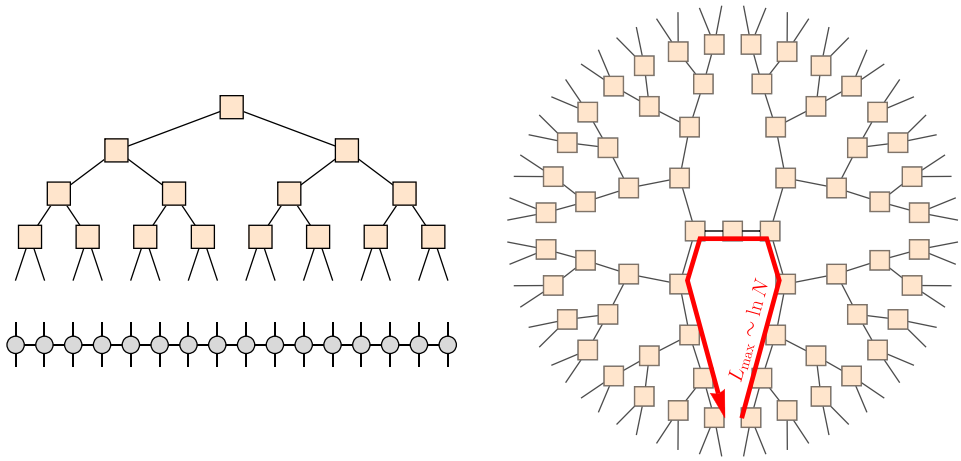

FIG. 2. Binary tree tensor network (BTTN) diagrams. (a) This BTTN corresponds to a 16-site lattice, with tensors and indices shown as squares and lines, respectively. The lines sticking out of the bottom layer of the tree represent physical indices and can contract with the corresponding physical indices in a matrix product operator (MPO), representing some rate operator W. This diagram therefore illustrates the action of W on an arbitrary state $|p\rangle$. (b) Here, the BTTN is built from a 64-site lattice and has tensors visually rearranged to emphasize the one-to-one mapping of tensors to sites along a circular lattice, as found in the 1D ratchet studied in this work and any other system subject to periodic boundary conditions. A red arrow is added to depict the largest distance L_{max} one has to travel between any two neighboring lattice sites. Contrary to a loopless MPS, here L_{max} scales logarithmically with the number of sites, thus rendering BTTN methods both accurate and tractable even with the absence of loops.

yielding a state vector that is parameterized by the set of all A's as

$$|p[A]\rangle = \sum_{\substack{s_1,\ldots,s_\chi \\ Q_1,\ldots,Q_\chi}} A^{(1)}_{Q_1,s_1} A^{(2)}_{Q_2,s_2},\ldots,A^{(\chi)}_{Q_\chi,s_\chi}|s_1,\ldots,s_N\rangle. \tag{16}$$

Note that any choice of tensors $[A]$ will yield a rank-N tensor c, but the converse may not be true. The realization of any arbitrary rank-N tensor in terms of a TN requires that the auxiliary indices linking the tensors are sufficiently high dimensional. The TN ansatz derives its computational utility by restricting this auxiliary index dimensionality, the so-called bond dimension. By finding a bond dimension that is large enough but not too large, it is often possible to make a good approximation to the exact dynamics while gaining the computational benefit of low dimensional tensors. Specifically, we cap the bond dimension at m, a tunable variational parameter, which generally controls how much the auxiliary indices can mediate correlations between nearby physical indices. A very large m value inevitably renders TN calculations intractable, whereas a very small m value generates an inflexible subspace on which variational calculations are excessively constrained. Capping the bond dimension necessarily means that one discards some information, so as we will discuss, singular value decompositions (SVD) can be strategically employed to preserve only the m most essential components of a matrix.

III. DYNAMICS OF THE TENSOR NETWORK

A. Binary tree tensor network

To actually perform a calculation, it is necessary to specialize to a particular design of how tensors are connected in a network. In other words, one must specify which indices belong to each of the sets S_i and Q_i. For 1D quantum and classical systems, the choice of network is usually a matrix product state (MPS). This MPS ansatz has proved to be convenient and robust for many applications.[37] The convenience derives from the ability to generate a canonical form or Schmidt decomposition, which allows for efficient and stable computations on an MPS.[38] Unfortunately, for systems with periodic boundary conditions, it is not possible to represent an MPS in a canonical form due to the loop in the TN structure.[37] To handle the periodic boundary conditions of the ratchet with a loopless TN that supports a canonical form, we, therefore, use a BTTN.[39,40]

The tree itself is illustrated in Fig. 2(a). Following Ref. 41, we label each tensor $A^{(i)}$ not by a single superscript (i) as in Eq. (15) but rather by the pair $[l,i]$, indicating that the tensor appears in the ith node of the lth layer of the tree. These $L \equiv \log_2 N$ layers count up from 0 at the root of the tree to $L-1$ at the base, while the sites count up from 0 to $2^l - 1$ moving from left to right across a layer. Into the base of the tree, feed N physical indices with dimension $d = 2$ corresponding to lattice sites that are either occupied or unoccupied. These tensors of the $l = L-1$ layer feed upward into parent tensors via auxiliary links. To capture all possible rank-N tensors c, the dimension of each auxiliary index must grow such that the link between layers l and $l+1$ would have dimension $M(l) = 2^{2^{L-l-1}}$. Assuming that auxiliary indices are truncated at a maximum bond dimension m, the auxiliary link between l and $l+1$ actually has dimension $\min(m, M(l))$.

The tree structure offers two principle benefits. Its loopless structure provides access to a canonical form, dramatically simplifying calculations. Furthermore, the BTTN allows correlations between pairs of lattice sites since each physical index is connected

to each other physical index by a pathway whose length grows only logarithmically with the number of lattice sites [see Fig. 2(b)]. Due to these merits, the BTTN has been applied to compute ground states via the DMRG[41] and dynamics via the TDVP.[16,17] We follow these works closely in applying the methodology to our problem.

B. Orthogonalization of the BTTN

The mapping from the $[A]$ tensors to the expansion coefficient c is many-to-one, so different combinations of values for the tensors can yield an identical state $|p\rangle$. One way this so-called *gauge freedom* can come about is by introducing a resolution of the identity, $D^{-1}D$, between tensors at neighboring nodes.[42] If one tensor is transformed by D^{-1} while its neighbor has a compensatory transformation by D, then the contraction of the tensors is unaffected though the individual tensors will change. Typically, one leverages the gauge freedom even more aggressively, transforming many tensors in a way that strategically privileges one node $[l,i]$. Observe that the tensor $\mathbf{A}^{[l,i]}$ is linked to one parental branch and two child branches. It is convenient to consider the three so-called environment tensors $|P^{[l-1,i/2]}\rangle$, $|L^{[l+1,2i]}\rangle$, and $|R^{[l+1,2i+1]}\rangle$ [see Fig. 3(c)], which capture the cumulative effect of the parent branch, left child branch, and right child branch, respectively. Note that each of these environment tensors depends on a single auxiliary index (one that feeds into $\mathbf{A}^{[l,i]}$) and all the physical indices associated with its branch of the tree. A state $|p\rangle$ is orthogonalized about $[l,i]$ (we say that $[l,i]$ is the *orthogonality center* of the tree) when it can be written as

$$\left|p^{[l,i]}[A]\right\rangle = \sum_{\alpha,\beta,\gamma} A^{[l,i]}_{\alpha\beta\gamma} \left|P^{[l-1,i/2]}_{\alpha}\right\rangle \left|L^{[l+1,2i]}_{\beta}\right\rangle \left|R^{[l+1,2i+1]}_{\gamma}\right\rangle, \tag{17}$$

with a gauge chosen such that the environment tensors satisfy the orthonormality conditions $\langle P_{\alpha'}|P_{\alpha}\rangle = \delta_{\alpha,\alpha'}$, $\langle L_{\beta'}|L_{\beta}\rangle = \delta_{\beta,\beta'}$, and $\langle R_{\gamma'}|R_{\gamma}\rangle = \delta_{\gamma,\gamma'}$. The computational benefit of this chosen gauge is clearest by computing the norm of the BTTN state,

$$\left\langle p^{[l,i]}[A]\middle|p^{[l,i]}[A]\right\rangle = \sum_{\alpha,\beta,\gamma} A^{[l,i]\dagger}_{\alpha\beta\gamma} A^{[l,i]}_{\alpha\beta\gamma}, \tag{18}$$

with † denoting Hermitian conjugation. Due to the environment tensor orthonormality, the norm only depends on the tensor at $[l,i]$.

The BTTN TDVP algorithm must advance $|p\rangle$ in time by advancing each tensor in the set $[A]$ in time, one by one. Akin to the norm calculation, the time evolution of $\mathbf{A}^{[l,i]}$ is most efficient if the BTTN has first been orthogonalized about $[l,i]$. After this propagation of $\mathbf{A}^{[l,i]}$, a new gauge transformation can reorthogonalize about a new node $[l',i']$ to allow the tensor at this node to be efficiently propagated. An explicit algorithm to carry out these BTTN orthogonalizations[17,41] is discussed in Appendix B.

C. Time evolution of tensor network states

In Sec. II B, we cast the calculation of currents in terms of a dynamic problem, requiring that we propagate a state $|p\rangle$ in time with propagators $\mathcal{W}_k(\lambda)$. If we were to represent $|p\rangle$ with the full rank-N tensor as in Eq. (14), this time evolution requires that we

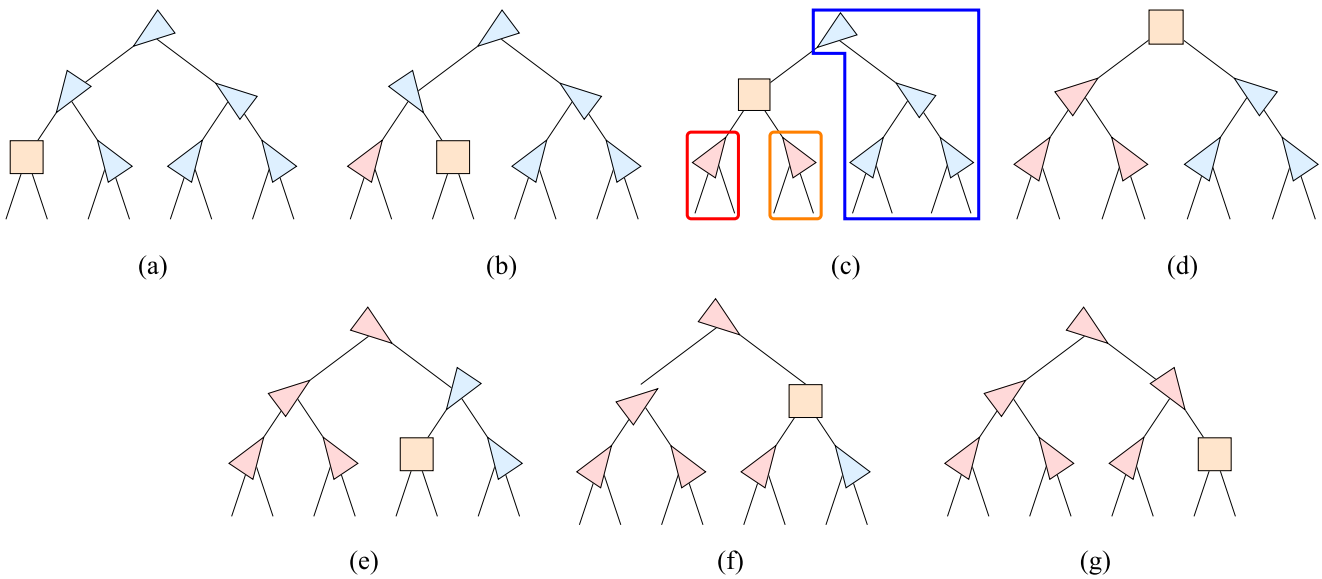

FIG. 3. TDVP traversal order [(a)–(g)] for the binary tree tensor network (BTTN) of an eight-site lattice. The starting and ending points are the left-most and right-most leaves of the tree, respectively. In each diagram, the tensor currently being updated, namely, node $[l,i]$, is colored in beige. Tensors that have already been time propagated are colored red, and the triangle shapes are used to point at the tensor serving as the orthogonalization center, $[l,i]$. Tensors that remain to be time propagated are colored in blue. Environment tensors $|P\rangle$, $|L\rangle$, and $|R\rangle$ are the composition of all tensors in the beige tensor's parent branch, left branch, and right branch, respectively. If a sweep is divided into two half-sweeps, as is often seen for matrix product state (MPS) methods, initially, steps (a)–(g) are performed, followed by a second set of updates in the reverse order.

numerically solve for the time-dependence of the expansion coefficient $\mathbf{c}$. In Sec. III A, we argued that $|p\rangle$ should instead be constructed from a set of tensors $[A]$ with a restricted bond dimension. Imagine propagating this state for time Δt with the tilted operator: $e^{\mathsf{W}_k(\lambda)\Delta t}|p[A]\rangle$. This newly evolved state generally cannot be exactly constructed in terms of the BTTN with the restricted bond dimension. Rather, the dynamics that starts with a BTTN state will have left the manifold of BTTN states and leaked into a nearby state in the space of possible rank-$N\mathbf{c}$. The earliest attempts to propagate TN states approximated the matrix exponential with a discrete time step, but these approaches, such as the time-evolving block decimation (TEBD),[43,44] could run into problems associated with the departure from the manifold of TN states.[4,45] An alternative approach seeks to propagate $|p\rangle$ with a constraint that the state remains confined on a variational manifold of allowed states. Conceptually, one can think of that constrained dynamics as consisting of the ordinary matrix exponential $e^{\mathsf{W}_k(\lambda)\Delta t}$ for infinitesimal time Δt immediately followed by a projection onto the variational states. This time-dependent variational principle (TDVP), first proposed by Dirac and Frenkel as a broad technique for variationally optimized dynamics,[2,3] was resurrected by Haegeman *et al.* when they demonstrated that the TDVP approach proved particularly effective when combined with the flexibility of a TN ansatz.[4,5]

This TN implementation of the TDVP, initially implemented for an MPS but later updated for tree tensor networks,[16,17] provides an algorithm to evolve $|p(t)\rangle$ with a discrete time step by computing an equation of motion for the tensors $[A]$ that parameterize the variational state. Furthermore, even with a discrete time step, the dynamics is constructed to rigorously remain on the TN variational manifold. The TN ansatz combines especially nicely with the TDVP approach because for a suitably orthogonalized BTTN, the algorithm implementing $[A]$'s time evolution can efficiently act on one single $\mathbf{A}^{[l,i]}$ tensor at a time. We carried out the single-center TDVP procedure reported by Bauernfeind and Aichhorn,[17] which we describe here and detail in Appendix C. To avoid truncation errors, we calculated dynamics using BTTN states with a fixed bond dimension, motivating the choice of a single-center algorithm over a two-center alternative.[16]

The algorithm starts with a set of tensors $[A]$ at time zero and carries out a step with time step Δt to yield a new set $[A']$ for this later time. Tensors in the BTTN are updated one by one according to an ordering for the tree traversal illustrated in Fig. 3. At tensor $\mathbf{A}^{[l,i]}$, one follows Appendix B to orthogonalize the BTTN about node $[l,i]$. The tensor at the orthogonalization center $[l,i]$ is then propagated forward in time for a time step of $\Delta t/2$ via a Lanczos exponentiation routine[46] by a local effective operator $\mathsf{W}_{\text{eff}}^{[l,i]}$, as described in Appendix C. This node is now said to be evolved forward by $\Delta t/2$. The tensor that had just been evolved in time is then decomposed via a singular value decomposition (SVD) into a product of orthogonal unitary operators $\mathbf{U}$ and $\mathbf{V}^\dagger$ sandwiching a diagonal matrix of singular values $\mathbf{S}$. This $\mathbf{U}$ is retained as the new time-propagated tensor at node $[l,i]$, but the product $\mathbf{SV}^\dagger$ will be contracted with the neighboring node to shift the orthogonalization center in preparation for the next node of the tree traversal sequence. Notice that, however, $\mathbf{SV}^\dagger$, which will be contracted into the neighboring node, was already advanced in time by an extra $\Delta t/2$ relative to this neighboring node. Before contracting them together, it is therefore necessary to propagate $\mathbf{SV}^\dagger$ backward in time by $\Delta t/2$. The net result is that node $[l,i]$ is advanced by $\Delta t/2$ and the orthogonalization center is shifted to the next node in sequence. One sweep of dynamics passes through the tree in the order of Fig. 3 to propagate by $\Delta t/2$ and then completes the full timestep by passing back through the tree in reverse order to propagate by an additional $\Delta t/2$.

IV. RESULTS

A. Constructing the initial BTTN state

To compute the SCGF for currents using the tilted dynamics of Eq. (12), we first must generate an initial BTTN state. This initial state should satisfy two needs. First, it should be similar to the time-periodic steady state. Due to the Perron–Frobenius theorem, an arbitrary initial state would relax into the time-periodic steady state, but the closer the initial state is, the faster the TDVP can converge. Second, the initial state must be constructed with a maximal

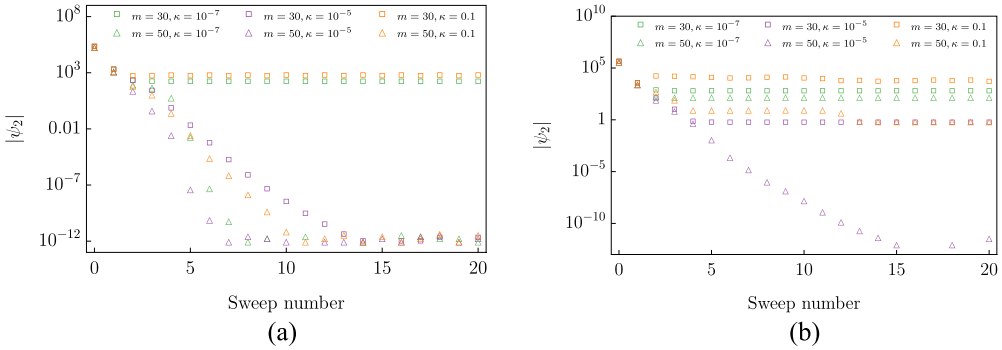

FIG. 4. Convergence to the steady state of W_2, $|\pi_2\rangle$, by the DMRG to serve as a seed in the TDVP calculations. As a rate matrix, the top eigenvalue of W_2 is zero, so convergence was assessed by monitoring how the estimate of the top eigenvalue ψ_2 approached zero for a 32-site lattice with eight (a) and 16 (b) particles. The DMRG calculations were repeated with maximal bond dimensions m of 30 (squares) and 50 (triangles) and with the subspace expansion mixing parameter values κ of 10^{-7}, 10^{-5}, and 0.1 (green, purple, and orange, respectively). For the DMRG to fully converge, m must be sufficiently large and κ must be neither too large nor too small.

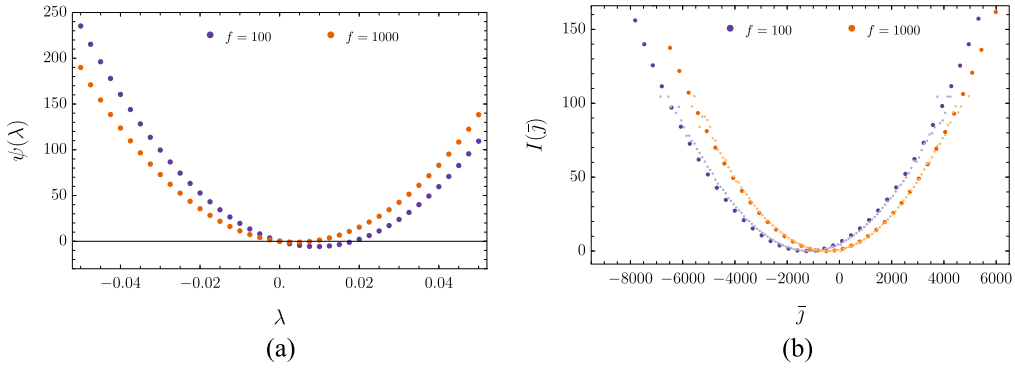

FIG. 5. Scaled cumulant-generating function (SCGF) and rate function obtained from the TDVP. (a) The SCGF $\psi(\lambda)$ is plotted as a function of the biasing parameter λ for a 32-site lattice occupied by 16 particles and two driving frequencies (100 and 1000). The slope of the SCGF at $\lambda = 0$ is observed to have a greater magnitude when $f = 100$ than when $f = 1000$, in agreement with the trends in the currents seen in Fig. 7. (b) The SCGF values were used to compute rate functions $I(\bar{\jmath})$, shown in thick dots, via a numerical Legendre transform. These rate functions were shown to be in very good agreement with Gillespie sampling (using the algorithm from Appendix E) of $10^6 t_{\text{obs}} = 0.1$ long trajectories. These Gillespie-sampled rate functions (small dots) fit a rate function from the histogram for $P(\bar{\jmath})$ using $I(\bar{\jmath}) = -\ln(P(\bar{\jmath}) - P(\langle\bar{\jmath}\rangle))/t_{\text{obs}}$.

bond dimension m that is sufficiently large that the BTTN manifold of states is a good approximation for the full state space. Because the single-center TDVP algorithm will not alter the bond dimension of this initial BTTN state, it is important that the initial state is constructed with careful control over the value of m. A version of the DMRG algorithm for BTTNs, described in Appendix D, meets both needs.

Recall that one period of the flashing ratchet first acts with W_1 for time $\tau/2$ and then with W_2 for time $\tau/2$. In the large τ, slow switching limit, the time-periodic steady state at the end of a full period will be very similar to the time-independent steady state of $e^{\mathsf{W}_2\tau/2}$, which, of course, shares eigenstates with the simpler W_2. As a seed for the TDVP, we therefore construct the top eigenstate of W_2, $|\pi_2\rangle$. Because W_2 is a rate matrix, $|\pi_2\rangle$ has an associated eigenvalue of zero and has the physical interpretation of the (equilibrium) steady state for the zero-potential "off" state of the ratchet. We build a BTTN approximation to $|\pi_2\rangle$ using a particular BTTN implementation of the DMRG, described in Appendix D, and we confirmed convergence of the algorithm by comparing the associated eigenvalue ψ_2 with zero.

For a lattice with N sites, this DMRG algorithm is seeded with any pure state [a state in which a single amplitude $c_{s_1 \ldots s_N}$ in Eq. (14) is unity and the rest are zero] with exactly N_{occ} occupied sites. The occupancy of each site specifies the physical indices of that pure state, while the auxiliary indices are initially trivial with bond dimension 1. The DMRG algorithm of Appendix D employs subspace expansion with a mixing parameter κ to allow the bond dimension to grow until it reaches the targeted value m. For lattices with 128 or fewer sites, the DMRG fully converged to the steady state of W_2 within a few dozen DMRG sweeps though the convergence generally requires a sufficiently large m and a tuned value of κ that is neither too small nor too large.[47] It is important to realize that the DMRG has more difficulty converging to the steady state as additional particles are added to the lattice. Figure 4 shows these convergence trends for a 32-site lattice with $N_{\text{occ}} = 8$ and 16.

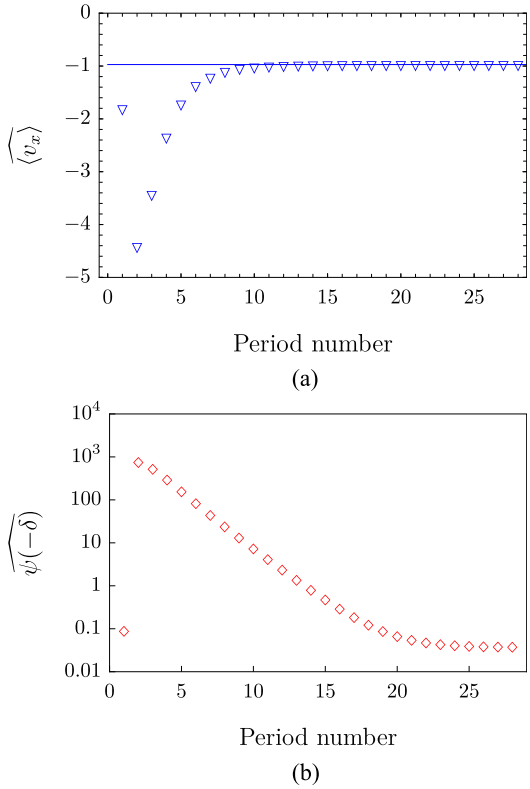

FIG. 6. Estimates (denoted with a hat) of $\langle v_x \rangle$ (a) and the SCGF $\psi(\lambda)$ (b) under weak biasing with $\lambda = -\delta \equiv -10^{-4}$ are plotted for each period of TDVP evolution with a timestep of 10^{-6}. The results are shown for a 32-site lattice occupied by 16 particles and a driving frequency of 1 MHz. The corresponding value of the average particle velocity obtained from Gillespie sampling is represented by the blue horizontal line, whose thickness is three times the standard error. The mean current extracted from TDVP agrees with the Gillespie sampling in fewer periods of driving than are required to converge the SCGF.

B. Extracting the SCGF for currents from the TDVP

The SCGF for period-averaged currents, $\psi(\lambda)$, is computed via the TDVP evolution and Eq. (12). The resulting SCGF, plotted in Fig. 5, contains information about the mean, variance, and higher cumulants of $\bar{\jmath}$. These statistical properties can be extracted from the behavior of the SCGF in the neighborhood of the origin, with the kth cumulant of $\bar{\jmath}$ computed from the kth derivative of $\psi(\lambda)$ evaluated at $\lambda = 0$. Our separate communication focuses on mean currents, in which case we needed only the slope of the SCGF at $\lambda = 0$.[20] In practice, we compute the first derivative numerically by introducing a very small biasing strength $\delta = 10^{-4}$ and approximating $\langle \bar{\jmath} \rangle = \psi'(0) \approx (\psi(\delta) - \psi(-\delta))/(2\delta)$. Starting with the $\lambda = 0$ seed $|\pi_2\rangle$, the TDVP is run with a timestep Δt for enough periods to converge the mean steady-state current. The calculation is stopped once the change in the current estimate between two adjacent periods lies within one percent of its magnitude, at which point full convergence is assumed. Figure 6 illustrates the convergence of both $\langle v_x \rangle \equiv \langle \bar{\jmath} \rangle h/N_{\mathrm{occ}}$ and $\psi(-\delta)$ over 20 periods of BTTN TDVP ratchet evolution with a half-occupied 32-site lattice. Both quantities converge in the long-time limit, but the convergence of current is noticeably faster than that of the SCGF.

The rate of convergence depends on the frequency of driving, particularly because the DMRG-generated seed $|\pi_2\rangle$ is constructed to match the low-frequency limit. For this reason, the low-frequency current can converge within one or two periods of TDVP evolution. At high frequencies, it is necessary to run tens or hundreds of periods to allow $|\pi_2\rangle$ time to evolve into the time-periodic steady state. One could converge more quickly by instead seeding with a high-frequency-limit eigenvector, the steady state of $(W_1 + W_2)/2$, but we found it sufficient (and simpler) to use the one seed for all frequencies.

The TDVP methodology extends beyond the small-λ regime, granting access also to fluctuations of $\bar{\jmath}$. These current fluctuations are characterized by a large deviation rate function $I(\bar{\jmath})$, computed as a Legendre transform of $\psi(\lambda)$.[33] In non-interacting ratchets, current statistics have previously been computed using a SCGF in models that are analytically solvable[48] or numerically tractable.[19] Our work extends these efforts to the interacting, many-particle regime. Figure 5 shows that one can compute the distribution for the current averaged over n periods of driving, $P(\bar{\jmath}) \simeq e^{-nI(\bar{\jmath})}$, by first performing the TDVP tensor network calculations of $\psi(\lambda)$ for various strengths of biasing λ. As a practical matter, these TDVP calculations are most stable if seeded by a state

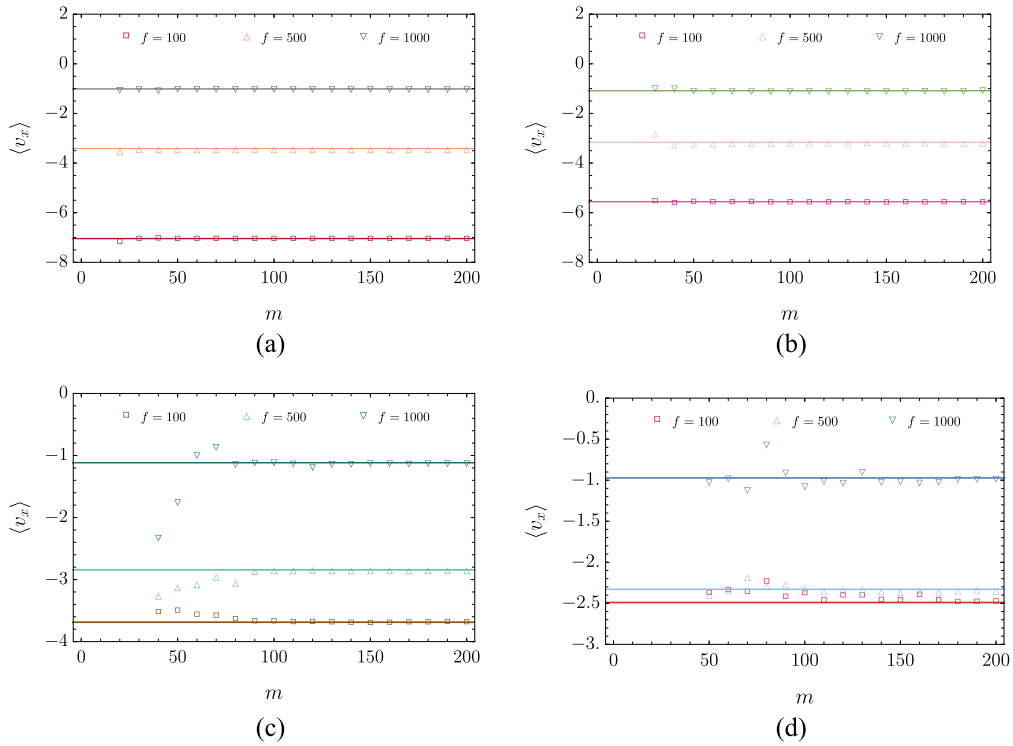

FIG. 7. Period-averaged mean particle velocity computed from the TDVP as a function of the maximal bond dimension. Average particle velocities for a 32-site lattice are plotted against m for $N_{\mathrm{occ}} = 4$ (a), 8 (b), 12 (c), and 16 (d) with driving frequencies $f = 100$, 500, and 1000. TDVP calculations with $N_{\mathrm{occ}} = 4$, 8, and 12 used $\Delta t = 10^{-6}$. The $N_{\mathrm{occ}} = 16$ calculations used $\Delta t = 10^{-7}$ to mitigate numerical instabilities that were especially prominent for $m < 150$. The DMRG fails to converge when m is very small, particularly for high occupancy. For these small m values, the TDVP calculation was not performed because it could not be seeded by $|\pi_2\rangle$. Average particle velocities obtained from Gillespie sampling are represented by horizontal lines, whose thicknesses match three standard errors. As the number of particles occupying the lattice increases, the required m increases.

that approximates the steady-state $|\pi(\lambda)\rangle_t$. We start by performing $\lambda = 0$ calculations and then increase and decrease λ in steps, seeding each calculation by a converged steady-state for a nearby value of λ.

C. Comparison with Monte Carlo sampling

As a variational method, the TDVP is not assured to work for small m. We validate that the tensor network ansatz indeed provides a good approximation by comparing with kinetic Monte Carlo sampling of the discrete-state jump process via Gillespie sampling. Due to switches between W_1 and W_2, however, waiting times for a hop no longer come from an exponential distribution and the usual Gillespie algorithm must be modified. Anderson[49] developed a rejection-based stochastic simulation algorithm (RSSA) to handle Markovian jump processes with arbitrary time dependencies. Because our 1D system relies on a square wave driving protocol, the usual Gillespie algorithm can be modified even more simply. We describe the specific algorithm in Appendix E. To estimate mean currents, 512 independent Gillespie trajectories were averaged. Each trajectory was allowed to relax to its time-periodic steady state by a burn-in lasting $t_0 = 0.01$ followed by a measurement of the current generated after an additional time $t_{\text{obs}} = 100$.

These Gillespie calculations of mean currents are simpler and less expensive than the tensor network methodology, but the TDVP approach offers some unique benefits. Figure 5 illustrates that the TDVP calculations accurately predict rare current fluctuations, even fluctuations that are more rare than can be readily observed by straightforward unbiased Gillespie sampling. Furthermore, the TDVP approach naturally generalizes to $\mathsf{W}(t)$ with an arbitrary time-dependence, whereas our Gillespie approach of Appendix E is specialized to the square-wave temporal driving. More general time dependence would require a more costly Gillespie strategy, such as RSSA.

D. TDVP with varied bond dimensions

The computational expense of the TDVP grows rapidly with the maximum bond dimension m. Consequently, to practically compute steady state properties from the TDVP, it is essential that m can be kept small while maintaining accuracy. We numerically probed the needed bond dimension by repeating the TDVP calculations on an $N = 32$ lattice with a range of m values, adjusted via the DMRG seed $|\pi_2\rangle$. We repeated these calculations for a range of driving frequencies and N_{occ} values. An optimal TDVP timestep depends on both N and N_{occ}. Too large a timestep results in numerical instability and convergence issues; too small makes a calculation unnecessarily costly. To compare the bond dimension results most simply, we used a fixed timestep of $\Delta t = 10^{-6}$, except for the case of $N_{\text{occ}} = 16$, which required $\Delta t = 10^{-7}$ to converge.

The dependencies on the maximum bond dimension are shown in Fig. 7. In all cases, the TDVP current tends to the value obtained from Gillespie simulation with a large enough m. The more this bond dimension threshold increases, the more the particles occupy the lattice, as rationalized by the vast increase in the number of states accessible by the TDVP as particles are added to the lattice. When the lattice is occupied by only four particles, a maximal bond dimension of merely 30 is sufficient for the TDVP to produce accurate ratchet currents within the driving frequency range considered, whereas the required maximal bond dimension increases dramatically (to around 180) for a half-occupied lattice ($N_{\text{occ}} = 16$).

V. DISCUSSION

We have illustrated that a BTTN with a tractable maximum bond dimension is sufficient to propagate a distribution over many-particle states evolving under a time-periodic protocol. The more conventional Gillespie approach evolves a single trajectory at a time and then averages over the trajectories. Propagating the distribution via the TDVP complements this strategy and offers several potential benefits. First, as shown in Fig. 5, the TDVP approach naturally gives information about both typical and rare events *at comparable computational expense*. While Gillespie sampling can also be biased to probe rare events, these calculations typically require significantly more computational power than the unbiased sampling of typical events. Second, the TDVP approach naturally generalizes to rate operators W with arbitrary time dependencies, a situation that can be quite challenging for Gillespie sampling. Finally, our calculations have been repeated for different system parameters, such as the frequency f. In the case of Gillespie sampling, the change in parameters demands an entirely new batch of simulated trajectories. The previous calculations do not speed up the next batch, which has to be sampled from scratch. By contrast, the TDVP calculations can leverage previous calculations to more rapidly converge steady-state dynamics with similar system parameters. We wrote about seeding our TDVP evolution from the state $|\pi_2\rangle$, but it can also be seeded from the converged state reached by a previous calculation. For example, suppose that one needs to compute $\psi(\lambda)$ for various frequencies. The converged calculation with frequency f_1 and biasing strength λ will have settled into a time-periodic state $|\pi(\lambda, f_1)\rangle_t$, which can be the initial state for the TDVP dynamics used to estimate $\psi(\lambda)$ at frequency f_2. Depending on the application, we anticipate that this ability to leverage previous calculations could warrant the extra complexity of the tensor network approach.

The present work is an initial attempt to employ tensor networks to treat time-periodic steady states in many-particle classical stochastic dynamics. Given the exceptional advances in tensor network methodologies, we anticipate future improvements to the stability and efficiency of the types of calculations we have described. Efficient new ways to compute time-evolution operators,[50] adaptive timesteps, and algorithms that adaptively construct tree tensor networks based on the structure of W[51] could all offer a path to future optimizations and improvements.

ACKNOWLEDGMENTS

We gratefully thank Schuyler Nicholson and Phillip Helms for many insightful discussions. We are also grateful to Miles Stoudenmire, Matthew Fishman, Steven White, and other developers of ITensor,[52] a library for implementing tensor network calculations, upon which this work was built. The material presented in this manuscript is based upon work supported by the National Science Foundation under Grant No. 2141385.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Nils E. Strand: Conceptualization (equal); Software (lead); Visualization (lead); Writing – original draft (equal); Writing – review & editing (equal). **Hadrien Vroylandt:** Conceptualization (equal); Software (supporting); Writing – review & editing (equal). **Todd R. Gingrich:** Conceptualization (equal); Funding acquisition (lead); Supervision (lead); Visualization (supporting); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

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

APPENDIX A: MATRIX PRODUCT OPERATOR (MPO) REPRESENTATION OF TILTED OPERATORS

Equation (13) gives a compact representation of $W_k(\lambda)$ that sums over all nearest-neighbor pairs of sites around the periodic boundary conditions. It is convenient, however, to deconstruct this sum in terms of a product of operator-valued vectors and matrices. The decomposition can be performed identically for each k. For compactness, we suppress the subscript k and write the matrix product as

$$W(\lambda) = W(1)W(2)\cdots W(N), \tag{A1}$$

where $W(1)$ is a one-by-ten row vector, $W(N)$ is a ten-by-one column vector, and the other $W(i)$ values are ten-by-ten matrices. By factorizing Eq. (13) in this manner, the tilted rate matrix is seen to be an MPO with each $W(i)$ corresponding to a shaded gray circle in Fig. 2(a). The ITensor library[52] contains an AutoMPO function that factorizes a sum like Eq. (13) into an explicit MPO. Alternatively, a finite-state machine can be employed[37] to derive the factorized local tensors for sites 1, $i = 2, \ldots, N-1$, and N, which are given by

$$W(1) = \begin{pmatrix} \mathbf{0} & r_{1\to 2}e^{\lambda}\mathbf{a} & -r_{1\to 2}\mathbf{n} & -r_{2\to 1}\mathbf{v} & r_{2\to 1}e^{-\lambda}\mathbf{a}^{\dagger} & r_{1\to N}e^{-\lambda}\mathbf{a} & -r_{1\to N}\mathbf{n} & -r_{N\to 1}\mathbf{v} & r_{N\to 1}e^{\lambda}\mathbf{a}^{\dagger} & \mathbb{I} \end{pmatrix},$$

$$W(i) = \begin{pmatrix} \mathbb{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{a}^{\dagger} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{v} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{n} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{a} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbb{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbb{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbb{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbb{I} & \mathbf{0} \\ \mathbf{0} & r_{i\to i+1}e^{\lambda}\mathbf{a} & -r_{i\to i+1}\mathbf{n} & -r_{i+1\to i}\mathbf{v} & r_{i+1\to i}e^{-\lambda}\mathbf{a}^{\dagger} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbb{I} \end{pmatrix}, \tag{A2}$$

$$W(N) = \begin{pmatrix} \mathbb{I} & \mathbf{a}^{\dagger} & \mathbf{v} & \mathbf{n} & \mathbf{a} & \mathbf{a}^{\dagger} & \mathbf{v} & \mathbf{n} & \mathbf{a} & \mathbf{0} \end{pmatrix}^{\mathsf{T}}.$$

APPENDIX B: BTTN ISOMETRIZATION PROCEDURE

As discussed in Sec. III B, an important step in DMRG, TDVP, and other BTTN methods is the isometrization or orthogonalization of the BTTN about a tensor $\mathbf{A}^{[l,i]}$. Reference 17 provides a particularly thorough discussion of the process. Here, we outline the essential steps. With the help of the gauge freedom discussed in Sec. III B, the goal is for node $[l,i]$ to become the *orthogonality center* of the tree. This procedure relies on a series of singular value decompositions (SVDs) on all tensors but $\mathbf{A}^{[l,i]}$, mapping tensor $\mathbf{A}^{[a,b]}$ to a new tensor $\mathbf{\Lambda}^{[a,b]\to[a',b']}$. The superscript denotes that the new tensor at node $[a,b]$ can now be thought of as having an orientation that points at the orthogonality center. This orientation is possible by the fact that there exists a unique path along the BTTN linking each node to $[l,i]$. Because $[a',b']$, a node neighboring $[a,b]$, is assumed to lie along the path connecting $[a,b]$ to $[l,i]$, $[a,b]$ must be orthogonalized about $[a',b']$. Isometrization of the BTTN is achieved by first performing an SVD on the furthest tensor from $[l,i]$ and repeating the process, performing the next SVD on the neighboring tensor until all tensors in the tree are oriented toward the orthogonality center (see Fig. 8). If we assume that, without loss of generality, $[a',b']$ is the right child of $[a,b]$ ($[a',b'] \equiv [a+1, 2b+1]$), then the SVD on $[a,b]$ takes the form

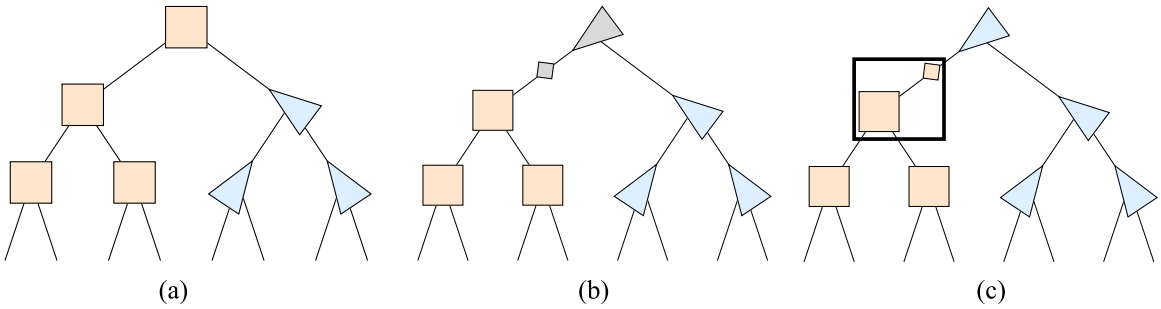

FIG. 8. Singular value decomposition (SVD) of tensor $\mathbf{A}^{[0,0]}$ as part of the procedure for isometrizing the BTTN about node $[2,0]$. In (a), any tensors 3 or more nodes away from $[l,i]$ have already been properly orthogonalized and are therefore drawn in blue. In (b), $\mathbf{A}^{[0,0]}$ undergoes an SVD, resulting in $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{V}^\dagger$, as shown in Eq. (B1). The product $\mathbf{SV}^\dagger$ is depicted as a small gray square. In (c), $\mathbf{SV}^\dagger$ is absorbed into the neighboring tensor $\mathbf{A}^{[1,0]}$ and $\mathbf{U}$ becomes the new orthogonal tensor on node $[0,0]$ ($\mathbf{\Lambda}^{[0,0]\to[1,0]}$).

$$A^{[a,b]}_{\alpha\beta\gamma} = \sum_n U_{\alpha\beta n} S_n V^\dagger_{n\gamma}. \tag{B1}$$

The orthogonal tensor $\mathbf{U}$ becomes $\mathbf{\Lambda}^{[a,b]\to[a',b']}$, while $\mathbf{SV}^\dagger$ is absorbed into the neighboring tensor on the path toward $[l,i]$.

The SVD is useful for two reasons. First, because each $\mathbf{\Lambda}$ is constructed as the orthogonal tensor of the SVD, consequently,

$$\sum_{\alpha\beta} \Lambda^{[a,b]\to[a',b']\dagger}_{\alpha\beta\gamma} \Lambda^{[a,b]\to[a',b']}_{\alpha\beta\gamma} = \mathbb{I}, \tag{B2}$$

where $\mathbb{I}$ is the identity matrix. Second, it is natural to limit the bond dimension to not exceed m by retaining only the largest m singular values when decomposing

$$A^{[a,b]}_{\alpha\beta\gamma} = \sum_{n=1}^{m} U_{\alpha\beta n} S_n V^\dagger_{n\gamma}. \tag{B3}$$

APPENDIX C: TDVP FOR THE BTTN

The core idea of the TDVP is to replace the evolution of $|p\rangle$ in Eq. (4) with the evolution of the set of tensors $[A]$ that parameterize the BTTN state $|p[A]\rangle$. While the dynamics in Eq. (4) could leave the manifold of BTTN states, $\mathcal{M}_{\mathrm{BTTN}}$, TDVP projects each infinitesimal time evolution onto $\mathcal{M}_{\mathrm{BTTN}}$. To develop such a projection for an MPS[5] and a TTN,[17] it is necessary to employ tools of differential geometry. Most significantly, infinitesimal changes to a state $|p[A]\rangle$ live in a tangent space $T_{|p[A]\rangle}$ consisting of only those changes that remain on the manifold $\mathcal{M}_{\mathrm{BTTN}}$. A key insight of Haegeman *et al.*[4,53] was that the projector onto this tangent space, $\hat{P}_{T_{|p[A]\rangle}}$, could be constructed and utilized in the time propagation of the TDVP equation of motion,

$$\frac{\partial |p[A]\rangle}{\partial t} = \hat{P}_{T_{|p[A]\rangle}} \mathcal{W}(t) |p[A]\rangle. \tag{C1}$$

We require a procedure to numerically integrate this Eq. (C1) to evolve $[A]$ and thus $|p[A]\rangle$ in time.

Before describing the integration of the equation of motion, it is necessary to elaborate upon the tangent space to ultimately construct $\hat{P}_{T_{|p[A]\rangle}}$. The tangent space at $|p[A]\rangle$ is spanned by the partial derivatives of $|p[A]\rangle$ with respect to tensors $A^{(i)}_{Q_i,S_i}$ for all $i = 1,\ldots,\chi$. An arbitrary vector in that space, $|\Theta[B]\rangle$, is parameterized by a set of tensors $[B]$,

$$\begin{aligned} |\Theta[B]\rangle &= \sum_{i=1}^{\chi} B^{(i)}_{Q_i,S_i} \frac{\partial |p[A]\rangle}{\partial A^{(i)}_{Q_i,S_i}} \\ &= \sum_{\substack{S_1,\ldots,S_\chi \\ Q_1,\ldots,Q_\chi}} \sum_{i=1}^{\chi} A^{(1)}_{Q_1,S_1},\ldots,B^{(i)}_{Q_i,S_i},\ldots,A^{(\chi)}_{Q_\chi,S_\chi} |s_1,\ldots,s_N\rangle. \end{aligned} \tag{C2}$$

Each term in the sum has the same form as Eq. (15) except $A^{(i)}_{Q_i,S_i}$ has been replaced by $B^{(i)}_{Q_i,S_i}$. In fact, for the BTTN architecture, Eq. (C2) can be compactly re-expressed in a form analogous to Eq. (17),

$$|\Theta[B]\rangle = \sum_{[l,i]} \sum_{\alpha,\beta,\gamma} B^{[l,i]}_{\alpha\beta\gamma} \left|P^{[l-1,i/2]}_{\alpha}\right\rangle \left|L^{[l+1,2i]}_{\beta}\right\rangle \left|R^{[l+1,2i+1]}_{\gamma}\right\rangle. \tag{C3}$$

The analogy with Eq. (17) goes beyond the decomposition in terms of parent, left leg, and right leg environment tensors. There is also a gauge freedom to be exploited. In this case, given a set $[B]$ that yields a tangent vector $|\Theta[B]\rangle$, one could construct the same tangent vector using $[B+\mathcal{B}]$ for any $[\mathcal{B}]$ in the kernel space, i.e., $|\Theta[\mathcal{B}]\rangle = 0$. To avoid a tangent space with an over-complete basis, it is necessary to fix the gauge, effectively choosing a single element from this kernel space.

To demonstrate the gauge fixing, we introduce some new notation. We denote by $[n,j]$ a general reference node on the graph; it is standard to choose some $[n,j]$ to be the final node along a BTTN sweeping sequence. Throughout this work, we assign $[n,j]$ to the network's bottom right leaf ($[L-1,N/2-1]$). Moreover, we have until now expressed auxiliary indices within a BTTN with α, β, and γ. As depicted in Fig. 3(c), α labeled an edge to a parent, β to a left child, and γ to a right child. The reference node $[n,j]$ is closest to the α, β, or γ index of the node at $[l,i]$. We designate this closest index $\rho_{[l,i]}$, emphasizing that it can be the α, β, or γ index depending on the location of node $[l,i]$. At each node, we denote the other two indices by q_1 and q_2. As in Appendix B, we can take the tensor at node $[l,i]$, i.e., $A^{[l,i]}_{\rho_{[l,i]},q_1,q_2}$, and orthogonalize this tensor about any of the three

neighboring nodes. We use the notation $\mathbf{\Lambda}^{[l,i]\to q}$ for the tensor that results from reorthogonalizing about the neighboring node of the q-index. We can then introduce a set of $\chi-1$ matrices $\mathbf{X}$ associated with each auxiliary index q and define the tensors $[\mathcal{B}]$ in terms of these matrices as

$$\mathcal{B}^{[l,i]}_{q_1q_2\rho_{[l,i]}} = \sum_{\rho_{[l,i]}'} \Lambda^{[l,i]\to\rho_{[l,i]}}_{q_1q_2\rho_{[l,i]}'} X_{\rho_{[l,i]}'\rho_{[l,i]}} - \sum_{q_1'} \Lambda^{[l,i]\to q_1}_{q_1'q_2\rho_{[l,i]}} X_{q_1'q_1} - \sum_{q_2'} \Lambda^{[l,i]\to q_2}_{q_1q_2'\rho_{[l,i]}} X_{q_2'q_2}. \tag{C4}$$

By construction, *for any choice of matrices* $\mathbf{X}$, $[\mathcal{B}]$ in Eq. (C4) yields $|\Theta[\mathcal{B}]\rangle = 0$.[17] Moreover, the $\mathbf{X}$ matrices uniquely specify elements of this kernel space, so fixing the gauge is equivalent to choosing a particular choice of $\mathbf{X}$. In practice, this choice is imposed implicitly by requiring that $[B]$ that parameterizes a tangent vector must satisfy $\chi-1$ constraints of the form

$$\sum_{q_1,q_2} B^{[l,i]}_{q_1q_2\rho_{[l,i]}'}{}^{\dagger} \Lambda^{[l,i]\to\rho_{[l,i]}}_{q_1q_2\rho_{[l,i]}} = \mathbf{0}. \tag{C5}$$

This set of constraints establishes a gauge that is particularly conducive to efficient computations of the inner product between two tangent vectors. A direct consequence of the gauge fixing is that this inner product involves only products over two tensors at the same node, a form reminiscent of Eq. (18), but now with a summation over all nodes $[l,i]$,

$$\langle\Theta[B']|\Theta[B]\rangle = \sum_{l,i}\sum_{\alpha,\beta,\gamma} B'^{[l,i]\dagger}_{\alpha\beta\gamma} B^{[l,i]}_{\alpha\beta\gamma}. \tag{C6}$$

Having seen how constrains of the form (C5) serve to fix a tangent vector gauge, we are finally in a position to compute the projector via a minimization problem. Given an arbitrary vector in the full Hilbert space, $|\Xi\rangle$, the projector $\hat{P}_{T_{|p[A]\rangle}}$ maps that state onto the state along the BTTN manifold. Hence, $\hat{P}_{T_{|p[A]\rangle}}|\Xi\rangle = |\Theta[B]\rangle$, where $|\Theta[B]\rangle$ solves

$$\min_B \||\Theta[B]\rangle - |\Xi\rangle\|^2. \tag{C7}$$

Due to the gauge fixing, the minimization problem of Eq. (C7) must actually be replaced by a constrained minimization problem that additionally imposes the constraints of Eq. (C5),

$$\min_B \Bigg[\sum_{[l,i]}\sum_{\alpha,\beta,\gamma} \left(B^{[l,i]\dagger}_{\alpha\beta\gamma} B^{[l,i]}_{\alpha\beta\gamma} - B^{[l,i]\dagger}_{\alpha\beta\gamma} F^{[l,i]}_{\alpha\beta\gamma} - F^{[l,i]\dagger}_{\alpha\beta\gamma} B^{[l,i]}_{\alpha\beta\gamma} \right) - \sum_{\substack{[l,i]\\ \neq[n,j]}} \sum_{\rho_{[l,i]}'} \Gamma^{[l,i]}_{\rho_{[l,i]}\rho_{[l,i]}'} \sum_{q_1,q_2} B^{[l,i]}_{q_1q_2\rho_{[l,i]}'}{}^{\dagger} \Lambda^{[l,i]\to\rho_{[l,i]}}_{q_1q_2\rho_{[l,i]}} \Bigg], \tag{C8}$$

where $F^{[l,i]}_{\alpha\beta\gamma} = \left\langle P^{[l-1,i/2]}_{\alpha}, L^{[l+1,2i]}_{\beta}, R^{[l+1,2i+1]}_{\gamma} \middle| \Xi \right\rangle$ and $\Gamma^{[l,i]}_{\rho_{[l,i]}\rho_{[l,i]}'}$ are Lagrange multipliers that impose the gauge fixing constraints. Solving Eq. (C8) leads to the following expression for the $[B]$ tensors at all nodes but the bottom right leaf ($[l,i] \neq [n,j]$),

$$B^{[l,i]}_{q_1q_2\rho_{[l,i]}} = F^{[l,i]}_{q_1q_2\rho_{[l,i]}} - \sum_{\rho_{[l,i]}'} \Lambda^{[l,i]\to\rho_{[l,i]}}_{q_1q_2\rho_{[l,i]}'} G^{[l,i]}_{\rho_{[l,i]}'\rho_{[l,i]}}, \tag{C9}$$

where

$$G^{[l,i]}_{\rho_{[l,i]}'\rho_{[l,i]}} = \sum_{q_1,q_2} \Lambda^{[l,i]\to\rho_{[l,i]}}_{q_1q_2\rho_{[l,i]}'} F^{[l,i]}_{q_1q_2\rho_{[l,i]}}. \tag{C10}$$

The first term of Eq. (C9) could be computed as the inner product between $|\Xi\rangle$ and the tensor product of the three environment tensors $\langle P|$, $\langle L|$, and $\langle R|$, each of which contracts over the parent, left child, and right child branches from $[l,i]$. The second term of Eq. (C9) has a similar simplification but in terms of two different environment tensors denoted by $|\Phi\rangle$ and $|\Psi\rangle$. In terms of these environment tensors, $G^{[l,i]}_{\rho_{[l,i]}'\rho_{[l,i]}} = \left\langle \Phi^{[l,i]}_{\rho_{[l,i]}'}, \Psi^{[l,i]}_{\rho_{[l,i]}} \middle| \Xi \right\rangle$. In the special case that $[l,i] = [n,j]$, we simply have $\mathbf{B}^{[n,j]} = \mathbf{F}^{[n,j]}$. Inserting the solution for $[B]$ of Eq. (C9) into Eq. (C3) thus gives an explicit expression for the tangent space projector,

$$\hat{P}_{T_{|p[A]\rangle}} = \sum_{[l,i]} \hat{P}^{[l-1,i/2]}_{P} \otimes \hat{P}^{[l+1,2i]}_{L} \otimes \hat{P}^{[l+1,2i+1]}_{R} - \sum_{\substack{[l,i]\\ \neq[n,j]}} \hat{P}^{[l,i]}_{\Phi} \otimes \hat{P}^{[l,i]}_{\Psi}, \tag{C11}$$

in terms of projectors for the five different environment tensors as follows:

$$\hat{P}^{[a,b]}_{P} = \sum_{q} \left|P^{[a,b]}_{q}\right\rangle\left\langle P^{[a,b]}_{q}\right|, \tag{C12}$$

$$\hat{P}^{[a,b]}_{L} = \sum_{q} \left|L^{[a,b]}_{q}\right\rangle\left\langle L^{[a,b]}_{q}\right|, \tag{C13}$$

$$\hat{P}^{[a,b]}_{R} = \sum_{q} \left|R^{[a,b]}_{q}\right\rangle\left\langle R^{[a,b]}_{q}\right|, \tag{C14}$$

$$\hat{P}^{[a,b]}_{\Phi} = \sum_{q} \left|\Phi^{[a,b]}_{q}\right\rangle\left\langle \Phi^{[a,b]}_{q}\right|, \tag{C15}$$

$$\hat{P}^{[a,b]}_{\Psi} = \sum_{q} \left|\Psi^{[a,b]}_{q}\right\rangle\left\langle \Psi^{[a,b]}_{q}\right|. \tag{C16}$$

Armed with Eq. (C11), we turn our attention back to the TDVP differential equation [Eq. (C1)] that propagates the tensors $[A]$ forward in time. The projector consists of a positive term involving $\hat{P}_P$, $\hat{P}_R$, and $\hat{P}_L$. When this positive term acts on $\mathsf{W}(t)$ in Eq. (C1), we are left with an effective rate operator, representing the action of $\mathsf{W}(t)$ in the vector space of the tensor $\mathbf{A}$. Whereas $\mathsf{W}(t)$ was a function of many physical indices, the effective rate operator at tensor $[l,i]$ acts on a small space depending on only six indices,

$$\left[\mathsf{W}^{[l,i]}_{\text{eff}}(t)\right]_{(\alpha,\beta,\gamma),(\alpha',\beta',\gamma')} = \left\langle P^{[l-1,i/2]}_{\alpha}, L^{[l+1,2i]}_{\beta}, R^{[l+1,2i+1]}_{\gamma} \middle| \mathsf{W}(t) \middle| P^{[l-1,i/2]}_{\alpha'}, L^{[l+1,2i]}_{\beta'}, R^{[l+1,2i+1]}_{\gamma'} \right\rangle. \tag{C17}$$

It is therefore possible to map the single tensor $\mathbf{A}^{[l,i]}$ forward in time with timestep $\Delta t/2$ according to

$$\frac{\partial \mathbf{A}^{[l,i]}(t)}{\partial t} = \mathsf{W}_{\text{eff}}^{[l,i]}(t)\mathbf{A}^{[l,i]}(t) \tag{C18}$$

by approximating the matrix exponential

$$\mathbf{A}^{[l,i]}(t+\Delta t/2) = \exp\left[\mathsf{W}_{\text{eff}}^{[l,i]}(t)\frac{\Delta t}{2}\right]\mathbf{A}^{[l,i]}(t) \tag{C19}$$

using a Krylov method.[46] The algorithm must loop over every node of the tree, advancing each of these tensors in time.

Additionally, the negative terms of Eq. (C11) must be allowed to act on $\mathsf{W}(t)$ in Eq. (C1). This action also yields an effective operator,

$$\left[\mathsf{K}_{\text{eff}}^{[l,i]}(t)\right]_{(\rho,\rho'),(\rho'',\rho''')} = \left\langle \Phi_{\rho}^{[l,i]}, \Psi_{\rho'}^{[l,i]} \middle| \mathsf{W}(t) \middle| \Phi_{\rho''}^{[l,i]}, \Psi_{\rho'''}^{[l,i]} \right\rangle, \tag{C20}$$

one built from decomposing the BTTN in terms of the environment operators that split at the edge associated with $[l,i]$ rather than the node at $[l,i]$. These terms have the physical interpretation of evolving the edges backward in time by $\Delta t/2$. After a tensor $\mathbf{A}^{[l,i]}$ is propagated forward by Δt, the BTTN is reorthogonalized around the next node via an SVD decomposition, generating a time-dependent bond tensor $\mathbf{C}^{[l,i]}(t+\Delta t/2) \equiv \mathbf{S}\mathbf{V}^{\dagger}$. Due to the negative term of the projector, each bond tensor must be mapped backward in time by $\Delta t/2$ in conjunction with the set [A] at the nodes being mapped forward in time by $\Delta t/2$,

$$\mathbf{C}^{[l,i]}(t) = \exp\left[-\mathsf{K}_{\text{eff}}^{[l,i]}(t)\frac{\Delta t}{2}\right]\mathbf{C}^{[l,i]}(t+\Delta t/2). \tag{C21}$$

APPENDIX D: EIGENSTATE OPTIMIZATION OF BTTN STATES

As discussed in Sec. IV A, we seed our TDVP calculations with the steady-state vector for the flat landscape, i.e., the top eigenstate $|\pi_2\rangle$ of the rate matrix W_2, satisfying

$$\mathsf{W}_2|\pi_2\rangle = \psi_2|\pi_2\rangle. \tag{D1}$$

In general, Eq. (D1) can be expressed as the variational problem

$$\langle p_2|\mathsf{W}_2|\pi_2\rangle - \psi_2\langle p_2|\pi_2\rangle = 0 \tag{D2}$$

solved by making $\psi_2 = \langle\pi_2|\mathsf{W}_2|\pi_2\rangle/\langle\pi_2|\pi_2\rangle$ stationary with respect to arbitrary perturbations from π_2 to p_2. In this work, $|\pi_2\rangle$ is practically computed via the density matrix renormalization group (DMRG) algorithm[1,37] made compatible with a binary tree tensor network (BTTN) ansatz. The DMRG iteratively performs local variational optimizations over individual tensors, sweeping through the network until ψ_2 converges. For a general BTTN representation of some state $|p\rangle$, each local optimization calculation consists of a generalized eigenvalue problem given by

$$\mathsf{W}_{\text{eff}}^{[l,i]}\mathbf{A}^{[l,i]} = \psi\mathsf{N}^{[l,i]}\mathbf{A}^{[l,i]}, \tag{D3}$$

where $\mathsf{W}_{\text{eff}}^{[l,i]}$ is the effective rate operator defined in Eq. (C17) and $\mathsf{N}^{[l,i]}$ is an effective norm obtained by contracting all tensors within the inner product $\langle p[A]|p[A]\rangle$ but $\mathbf{A}^{[l,i]}$ and its conjugate transpose. Conveniently, the loopless property of the BTTN representation renders possible the construction of a canonical form via the gauge freedom. The orthogonal basis formed from this particular gauge turns the norm $\mathsf{N}^{[l,i]}$ into the identity matrix, thus transforming Eq. (D3) into a standard eigenvalue problem, which throughout this work is, in practice, solved via Arnoldi iteration.

Within a single sweep of the DMRG, a total of $\chi = N-1$ local eigenvalue problems of the form shown in Eq. (D3) are solved, resulting in updates of each individual tensor within the binary tree tensor network (BTTN), one by one. The initial step within the single local update on tensor $\mathbf{A}^{[l,i]}$ is to isometrize the BTTN about this tensor following the algorithm detailed in Appendix B. The next step is to calculate the one-center effective operator $\mathsf{W}_{\text{eff}}^{[l,i]}$ and obtain a new tensor $\mathbf{A}^{[l,i]}$ by solving Eq. (D3). If node $[l,i]$ is the final one to be updated within the sweep, no further actions are needed for the current tensor. Otherwise, a further orthogonalization step is needed, this time toward $[l',i']$, the next chosen node in the sweep, accompanied by truncation to a maximal bond dimension m, again following the procedure described in Appendix B. An entire sweep of the DMRG involves repeating the steps outlined above once for each tensor within the BTTN, first following the traversal order defined in Fig. 3 and then following the same traversal order again but in reverse.

As mentioned in Sec. IV A, an important step of the DMRG procedure is to construct a pure state to be chosen as an initial seed. Typically, a tensor network representation of such a state has the dimensions of all auxiliary indices set to unity. To allow the DMRG routine to increase the bond dimensions from one sweep to the next, optimizations of individual tensors $\mathbf{A}^{[l,i]}$ are accompanied by an enrichment step whose role is to expand the local basis used to represent one of the bonds in $\mathbf{A}^{[l,i]}$ while still keeping the overall value of the state intact. We implemented this enrichment using the subspace expansion routine, previously been implemented for the MPS representation,[54] adapted for the BTTN. The main idea behind subspace expansion is to devise a new set of basis states for one of the auxiliary indices of tensor $A_{\alpha\beta\gamma}^{[l,i]}$. Without loss of generality, we assume that this index is γ. This basis change is accomplished by appending an expansion term $\mathbf{P}^{[l,i]}$ to the existing basis,

$$\mathbf{A}^{[l,i]} \to \tilde{\mathbf{A}}^{[l,i]} = \begin{bmatrix}\mathbf{A}^{[l,i]} & \mathbf{P}^{[l,i]}\end{bmatrix}. \tag{D4}$$

Similarly, the components of the adjacent right child tensor are expanded by adding an appropriately sized zero padding,

$$\mathbf{A}^{[l+1,2i+1]} \to \tilde{\mathbf{A}}^{[l+1,2i+1]} = \begin{bmatrix}\mathbf{A}^{[l+1,2i+1]} \\ \mathbf{0}\end{bmatrix}. \tag{D5}$$

Even though the state $|p\rangle$ remains unchanged after this basis transformation, the DMRG now has the ability to access states within a larger BTTN manifold than before. In reality, the exact expansion term $\mathbf{P}^{[l,i]}$ should be taken as the tensor $\mathbf{Z}^{[l,i]}$ of the residual

$$|Z\rangle = \mathsf{W}|p\rangle - \psi|p\rangle = \sum_{\substack{S_1,\ldots,S_\chi \\ Q_1,\ldots,Q_\chi}} Z_{Q_1}^{S_1} Z_{Q_2}^{S_2},\ldots,Z_{Q_\chi}^{S_\chi}|s_1,\ldots,s_N\rangle, \tag{D6}$$

where, as in Eq. (16), χ is the number of tensors in the BTTN and $Z^{[l,i]}_{\alpha\beta\gamma} \equiv Z_{Q_\eta}$ for some $\eta \leq \chi$. However, in practice, these tensors are difficult to compute and the subspace expansion method instead relies on the convenient perturbative expansion term,

$$\mathbf{P}^{[l,i]} = \kappa \mathbf{V}^{[l-1,i/2]} \mathbf{A}^{[l,i]} \mathbf{V}^{[l+1,2i]}, \tag{D7}$$

where $\mathbf{V}^{[l-1,i/2]}$ and $\mathbf{V}^{[l+1,2i]}$ are formed from the contraction of all tensors within the branch linking the parent and left child of node $[l,i]$, respectively, to any bottom layer nodes of the ket state $|p[A]\rangle$, the identical contraction within the bra state $\langle p[A]|$, and a contraction of tensors belonging to the matrix product operator (MPO) sandwiched between the relevant bra and ket tensors. The parameter κ represents a scalar mixing factor whose role is to tune the contributions of the perturbative terms.

Finally, it must be noted that the rate operators used for our ratchet system, defined in Eq. (13), conserve the number of particles in the lattice, and therefore, a separate steady-state distribution exists for each particle number. Given an arbitrary initial state describing a lattice occupied by N_{occ} particles, it is crucial that the DMRG produces a steady-state distribution for exactly N_{occ} particles. The classical occupation representation used in Eq. (13) happens to be isomorphic to the quantum spin-half basis, so in a sense, one can say that it conserves the Abelian S_z "quantum number" (QN). As it turns out, representing the tensors $\mathbf{A}^{[l,i]}$ from Eq. (17) in a special QN-conserving sparse block structure ensures that the total QN of the state throughout the DMRG routine remains identical to that of the input state.[42] With this representation, seeding the DMRG with an arbitrary configuration having N_{occ} of N sites occupied is guaranteed to yield the steady-state distribution for N_{occ} particles upon full convergence of the DMRG. Additionally, this QN-conserving sparse structure typically renders calculations both more efficient and less memory intensive, as many of the tensor blocks are constrained to be zero.

APPENDIX E: GILLESPIE ALGORITHM FOR SQUARE WAVE DRIVING

Because the flashing ratchet has a temporal drive with a period of τ, standard Gillespie sampling[55] must be adapted to accommodate the time-dependent propensities. As in the traditional algorithm, these propensities are used to compute a target state and a random waiting time at each step along a trajectory. Where the traditional algorithm breaks down is in the event that a drawn waiting time would span both sets of propensities. For example, if the previous hop occurred during a W_1 propagation but the next would not occur until the W_2 propagation, then the waiting time should reflect some mixture of the W_1 and W_2 rates.

Letting the time-dependent propensity be denoted by $w(t)$, the waiting time δt should solve

$$\int_{t_0}^{t_0+\delta t} w(t)dt = \ln\left(\frac{1}{s}\right), \tag{E1}$$

where $s \sim U(0,1)$ is a random number drawn uniformly from the unit interval and t_0 is the time of the most recent hop.[49] When $w(t)$ is a constant w, the integral evaluates to $w\delta t$, recovering the usual Gillespie algorithm for drawing waiting times. While it is not as simple, the integral can be similarly evaluated for the square wave driving that flips between a rate w_1 and another rate w_2.

Without loss of generality, let us assume $w(t_0) = w_1$. If $w(t)$ remains w_1 until $t = t_0 + \delta t$, that is, if $\text{Mod}(t_0, \tau) + \delta t < \tau/2$, then δt is computed as usual, namely,

$$\delta t = (1/w_1)\ln(1/s). \tag{E2}$$

If the waiting time δt would pass through the time that the rate jumps from w_1 to w_2, then Eq. (E1) instead integrates to give

$$\left(\frac{\tau}{2} - t_0\right)w_1 + \left(t_0 + \delta t - \frac{\tau}{2}\right)w_2 = \ln\left(\frac{1}{s}\right), \tag{E3}$$

which, after some algebra, yields

$$\delta t = \frac{1}{w_2}\left[\ln\left(\frac{1}{s}\right) + (w_2 - w_1)\left(\frac{\tau}{2} - t_0\right)\right]. \tag{E4}$$

Particularly, when the driving frequency is high, it is possible that a waiting time δt could pass through the time that w_1 switches to w_2 and the time that the rate switches back to w_1. In this case, the waiting time is related to the random number s as

$$\left(\frac{\tau}{2} - t_0\right)w_1 + \frac{\tau}{2}w_2 + (t_0 + \delta t - \tau)w_1 = \ln\left(\frac{1}{s}\right), \tag{E5}$$

leading to the waiting time

$$\delta t = \frac{1}{w_1}\left[\ln\left(\frac{1}{s}\right) + (w_1 - w_2)\frac{\tau}{2}\right]. \tag{E6}$$

One can continue casing out the possibilities, adding more cycles between w_1 and w_2 before the next hop occurs. For example, the next case involves waiting time

$$\delta t = \frac{1}{w_2}\left[\ln\left(\frac{1}{s}\right) + (w_2 - w_1)(\tau - t_0)\right]. \tag{E7}$$

In practice, one starts by computing the rates w_1 and w_2 associated with each possible jump from the current configuration at time t_0. Next, s is drawn and a separate waiting time δt is computed for each possible hop. $w(t_0)$ is set to w_1 if $\text{Mod}(t_0, \tau) < \tau/2$ and to w_2 otherwise. For a given hop, if Eq. (E2) yields δt consistent with the first case, meaning that δt is sufficiently short that the square wave drive will not have switched from W_1 to W_2 (or from W_2 to W_1), then this waiting time is chosen. Otherwise, we proceed to the next case, inserting s into Eq. (E4) [if $w(t_0) = w_1$]. We continue passing through the cases until the computed δt is consistent with the considered case for how many square wave flips have been experienced. Once consistency is achieved, this δt is taken to be the next waiting time for the proposed transition. Finally, the next chosen transition is the one with the smallest waiting time.

REFERENCES

[1] S. R. White, "Density matrix formulation for quantum renormalization groups," Phys. Rev. Lett. **69**, 2863–2866 (1992).
[2] P. A. M. Dirac, "Exchange phenomena in the Thomas atom," Proc. Cambridge Philos. Soc. **26**, 376–385 (1930).
[3] J. Frenkel, *Wave Mechanics: Advanced General Theory* (Claredon Press, Oxford, 1934).

[4]J. Haegeman, J. I. Cirac, T. J. Osborne, I. Pižorn, H. Verschelde, and F. Verstraete, "Time-dependent variational principle for quantum lattices," Phys. Rev. Lett. **107**, 070601 (2011).
[5]J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, and F. Verstraete, "Unifying time evolution and optimization with matrix product states," Phys. Rev. B **94**, 165116 (2016).
[6]P. Helms, U. Ray, and G. K.-L. Chan, "Dynamical phase behavior of the single- and multi-lane asymmetric simple exclusion process via matrix product states," Phys. Rev. E **100**, 022101 (2019).
[7]M. C. Bañuls and J. P. Garrahan, "Using matrix product states to study the dynamical large deviations of kinetically constrained models," Phys. Rev. Lett. **123**, 200601 (2019).
[8]P. Helms and G. K.-L. Chan, "Dynamical phase transitions in a 2D classical nonequilibrium model via 2D tensor networks," Phys. Rev. Lett. **125**, 140601 (2020).
[9]Z. Nagy, C. Appert, and L. Santen, "Relaxation times in the ASEP model using a DMRG method," J. Stat. Phys. **109**, 623–639 (2002).
[10]Y. Hieida, "Application of the density matrix renormalization group method to a non-equilibrium problem," J. Phys. Soc. Jpn. **67**, 369–372 (1998).
[11]K. Temme and F. Verstraete, "Stochastic matrix product states," Phys. Rev. Lett. **104**, 210502 (2010).
[12]M. Gorissen, J. Hooyberghs, and C. Vanderzande, "Density-matrix renormalization-group study of current and activity fluctuations near nonequilibrium phase transitions," Phys. Rev. E **79**, 020101 (2009).
[13]T. H. Johnson, S. R. Clark, and D. Jaksch, "Dynamical simulations of classical stochastic systems using matrix product states," Phys. Rev. E **82**, 036702 (2010).
[14]T. H. Johnson, T. J. Elliott, S. R. Clark, and D. Jaksch, "Capturing exponential variance using polynomial resources: Applying tensor networks to nonequilibrium stochastic processes," Phys. Rev. Lett. **114**, 090602 (2015).
[15]E. Carlon, M. Henkel, and U. Schollwöck, "Density matrix renormalization group and reaction-diffusion processes," Eur. Phys. J. B **12**, 99–114 (1999).
[16]L. Kohn, P. Silvi, M. Gerster, M. Keck, R. Fazio, G. E. Santoro, and S. Montangero, "Superfluid-to-Mott transition in a Bose-Hubbard ring: Persistent currents and defect formation," Phys. Rev. A **101**, 023617 (2020).
[17]D. Bauernfeind and M. Aichhorn, "Time dependent variational principle for tree tensor networks," SciPost Phys. **8**, 024 (2020).
[18]T. Felser, S. Notarnicola, and S. Montangero, "Efficient tensor network ansatz for high-dimensional quantum many-body problems," Phys. Rev. Lett. **126**, 170603 (2021).
[19]N. E. Strand, R.-S. Fu, and T. R. Gingrich, "Current inversion in a periodically driven two-dimensional Brownian ratchet," Phys. Rev. E **102**, 012141 (2020).
[20]N. E. Strand, H. Vroylandt, and T. R. Gingrich, "Using tensor network states for multiparticle Brownian ratchets," J. Chem. Phys. **156**, 221103 (2022).
[21]S. Rahav, J. Horowitz, and C. Jarzynski, "Directed flow in nonadiabatic stochastic pumps," Phys. Rev. Lett. **101**, 140602 (2008).
[22]O. Raz, Y. Subaşı, and C. Jarzynski, "Mimicking nonequilibrium steady states with time-periodic driving," Phys. Rev. X **6**, 021022 (2016).
[23]G. M. Rotskoff, "Mapping current fluctuations of stochastic pumps to nonequilibrium steady states," Phys. Rev. E **95**, 030101 (2017).
[24]A. C. Barato and R. Chetrite, "Current fluctuations in periodically driven systems," J. Stat. Mech. **2018**, 053207.
[25]S. Asban and S. Rahav, "No-pumping theorem for many particle stochastic pumps," Phys. Rev. Lett. **112**, 050601 (2014).
[26]R. M. da Silva, C. C. de Souza Silva, and S. Coutinho, "Reversible transport of interacting Brownian ratchets," Phys. Rev. E **78**, 061131 (2008).
[27]D. McDermott, C. J. O. Reichhardt, and C. Reichhardt, "Collective ratchet effects and reversals for active matter particles on quasi-one-dimensional asymmetric substrates," Soft Matter **12**, 8606–8615 (2016).
[28]O. Kedem, B. Lau, and E. A. Weiss, "How to drive a flashing electron ratchet to maximize current," Nano Lett. **17**, 5848–5854 (2017).
[29]O. Kedem, B. Lau, and E. A. Weiss, "Mechanisms of symmetry breaking in a multidimensional flashing particle ratchet," ACS Nano **11**, 7148–7155 (2017).
[30]O. Kedem and E. A. Weiss, "Cooperative transport in a multi-particle, multi-dimensional flashing ratchet," J. Phys. Chem. C **123**, 6913–6921 (2019).
[31]P. Reimann, "Brownian motors: Noisy transport far from equilibrium," Phys. Rep. **361**, 57–265 (2002).
[32]T. R. Gingrich, G. M. Rotskoff, and J. M. Horowitz, "Inferring dissipation from current fluctuations," J. Phys. A: Math. Theor. **50**, 184004 (2017).
[33]H. Touchette, "The large deviation approach to statistical mechanics," Phys. Rep. **478**, 1–69 (2009).
[34]J. L. Lebowitz and H. Spohn, "A Gallavotti–Cohen-type symmetry in the large deviation functional for stochastic dynamics," J. Stat. Phys. **95**, 333–365 (1999).
[35]V. Lecomte, C. Appert-Rolland, and F. van Wijland, "Thermodynamic formalism for systems with Markov dynamics," J. Stat. Phys. **127**, 51–106 (2007).
[36]L. Chabane, R. Chétrite, and G. Verley, "Periodically driven jump processes conditioned on large deviations," J. Stat. Mech. **2020**, 033208.
[37]U. Schollwöck, "The density-matrix renormalization group in the age of matrix product states," Ann. Phys. **326**, 96–192 (2011).
[38]G. Vidal, "Efficient classical simulation of slightly entangled quantum computations," Phys. Rev. Lett. **91**, 147902 (2003).
[39]V. Murg, F. Verstraete, Ö. Legeza, and R. M. Noack, "Simulating strongly correlated quantum systems with tree tensor networks," Phys. Rev. B **82**, 205105 (2010).
[40]Y.-Y. Shi, L.-M. Duan, and G. Vidal, "Classical simulation of quantum many-body systems with a tree tensor network," Phys. Rev. A **74**, 022320 (2006).
[41]M. Gerster, P. Silvi, M. Rizzi, R. Fazio, T. Calarco, and S. Montangero, "Unconstrained tree tensor network: An adaptive gauge picture for enhanced performance," Phys. Rev. B **90**, 125154 (2014).
[42]P. Silvi, F. Tschirsich, M. Gerster, J. Jünemann, D. Jaschke, M. Rizzi, and S. Montangero, "The tensor networks anthology: Simulation techniques for many-body quantum lattice systems," SciPost Phys. Lect. Notes **8**, 1 (2019).
[43]G. Vidal, "Efficient simulation of one-dimensional quantum many-body systems," Phys. Rev. Lett. **93**, 040502 (2004).
[44]F. Verstraete, J. J. García-Ripoll, and J. I. Cirac, "Matrix product density operators: Simulation of finite-temperature and dissipative systems," Phys. Rev. Lett. **93**, 207204 (2004).
[45]G. Vidal, "Classical simulation of infinite-size quantum lattice systems in one spatial dimension," Phys. Rev. Lett. **98**, 070201 (2007).
[46]M. Hochbruck and C. Lubich, "On Krylov subspace approximations to the matrix exponential operator," SIAM J. Numer. Anal. **34**, 1911–1925 (1997).
[47]M. Yang and S. R. White, "Time-dependent variational principle with ancillary Krylov subspace," Phys. Rev. B **102**, 094315 (2020).
[48]A. W. C. Lau, D. Lacoste, and K. Mallick, "Nonequilibrium fluctuations and mechanochemical couplings of a molecular motor," Phys. Rev. Lett. **99**, 158102 (2007).
[49]D. F. Anderson, "A modified next reaction method for simulating chemical systems with time dependent propensities and delays," J. Chem. Phys. **127**, 214107 (2007).
[50]B. Vanhecke, D. Devoogdt, F. Verstraete, and L. Vanderstraeten, "Simulating thermal density operators with cluster expansions and tensor networks," arXiv:2112.01507 (2021).
[51]G. Ferrari, G. Magnifico, and S. Montangero, "Adaptive-weighted tree tensor networks for disordered quantum many-body systems," Phys. Rev. B **105**, 214201 (2021).
[52]M. Fishman, S. R. White, and E. M. Stoudenmire, "The ITensor software library for tensor network calculations," arXiv:2007.14822 (2020).
[53]J. Haegeman, T. J. Osborne, and F. Verstraete, "Post-matrix product state methods: To tangent space and beyond," Phys. Rev. B **88**, 075133 (2013).
[54]C. Hubig, I. P. McCulloch, U. Schollwöck, and F. A. Wolf, "Strictly single-site DMRG algorithm with subspace expansion," Phys. Rev. B **91**, 155115 (2015).
[55]D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," J. Phys. Chem. **81**, 2340–2361 (1977).