Pairwise connected tensor network representation of path integrals

Amartya Bose [*]

Department of Chemistry, Princeton University, Princeton, New Jersey 08544, USA

It has been recently shown how the tensorial nature of real-time path integrals (PIs) involving the Feynman-Vernon influence functional can be utilized with matrix product states, taking advantage of the finite length of the bath-induced memory. Tensor networks (TNs) promise to provide a unified language to express the structure of a PI. Here, a generalized TN specifically incorporating the pairwise interaction structure of the influence functional and its invariance with respect to the average forward-backward position or the *sojourn* value in the form of the blip representation is derived and implemented. This pairwise connected TNPI (PC-TNPI) is illustrated through applications to typical spin-boson problems and explorations of the differences caused by the exact form of the spectral density. The storage and performance scalings are reported, showing the compactness of the representation and the efficiency of the contraction process. Finally, taking advantage of the compressed representation, the viability of using PC-TNPI for simulating multistate problems is demonstrated. The PC-TNPI structure can be shown to yield other TN algorithms currently in use. Consequently, it should be possible to use it as a starting point for deriving other optimized procedures.

I. INTRODUCTION

Tensor networks (TNs) are designed to be compact factorized representations of high-ranked tensors. Probably the most common use of TNs in physics is related to representations of the quantum many-body wave function which, in general, is also a high-ranked tensor. This use has been widely demonstrated in a multitude of methods such as the density matrix renormalization group (DMRG) [1,2], which uses a matrix product state (MPS) [3,4] representation, and the multiconfiguration time-dependent Hartree (MCTDH) [5] and its multilayer version (ML-MCTDH) [6–8], which use tree TNs. For multidimensional systems, an extension of MPS to multiple dimensions called projected entanglement pair states [9] is used. For systems at critical points, an MPS representation does not work because of long-range correlations necessitating the use of the so-called multiscale entanglement renormalization ansatz [10,11]. TNs, since their introduction, have proliferated in various diverse fields requiring the use of compact representations of multidimensional data like machine learning and deep neural networks.

While quantum dynamics at zero temperature can often be simulated using wave-function-based methods like time-dependent DMRG [2,12,13] or MCTDH, at finite temperatures, owing to the involvement of a manifold of vibrational and low frequency rotranslational states in the dynamics, they suffer from an exponentially growing computational requirements. Feynman's path integral (PI) provides a very convenient alternative for simulating the time-dependent reduced density matrix (RDM) for the system. The vibrational states of the solvent introduced as harmonic phonon modes under linear response [14] are integrated out, leading to the Feynman-Vernon influence functional [15]. An identical influence functional also arises in dealing with light-matter interaction through the integration of the photonic field.

The primary challenge in using influence functionals and PIs is the presence of the nonlocal history-dependent memory that leads to an exponential growth of system paths. Many recent developments have helped improve the efficiency of simulations [16–19]. Most notably, the recently developed small matrix decomposition of PIs (SMatPIs) [20–22] reduces the storage requirement to multiple matrices of the size of the RDM. However, all these methods utilize very different and deep insights into the structure of PIs. It has recently been shown that the MPS representation can also be very effectively utilized to reformulate real-time PIs involving the influence functional leveraging the finite nature of the nonlocal memory [23–26] in condensed phases and the consequently low entanglement between well-separated time points. More recently, such TN-based ideas have been extended to a two-dimensional TN that can simulate the dynamics of extended quantum systems coupled with dissipative media [27].

While the MPS structure is the simplest TN that can be used to describe the so-called path amplitude tensor, the one-dimensional topology is probably not optimal when the bath-induced memory spans a large number of time steps and suffers from growing bond dimensions. In this paper, an alternate generalized TN is introduced that directly captures the pairwise interaction structure of the Feynman-Vernon influence functional and its independence with respect to the average forward-backward or *sojourn* position by utilizing the so-called blip representation [17–19]. This pairwise connected TNPI (PC-TNPI) has an extremely compact representation that can be efficiently evaluated. We show how

---
[*]amartyab@princeton.edu; amartya.bose@gmail.com

the previous MPS representations can be thought of as special refactorings of the current PC-TNPI. TNs show great promise in being a unifying language for formulating and thinking about PI methods. The goal of this paper is to introduce a TN that can act as this unified basis for further developments and explorations in usage of TNs for simulating open quantum systems.

The construction and evaluation of the TN is discussed in Sec. II. The computational and space complexity has been rigorously derived to prove the feasibility of contraction of this TN. In Sec. III, we illustrate some typical applications of the algorithm. Because of usage of the blip representation, despite being the most naïve implementation of the idea, it performs on par with a blip-summed PI [17,18]. The implementation of this method utilized the open-source ITENSOR [28] library for tensor contractions, allowing for extremely efficient tensor contractions using highly efficient BLAS and LAPACK libraries. We end the paper in Sec. IV with some concluding remarks and outlook on future explorations.

II. METHODOLOGY

Consider a quantum system coupled to a dissipative environment described by a Caldeira-Leggett model [29–31]:

$$\hat{H} = \hat{H}_0 + \hat{H}_{\text{env}}(p, x), \tag{1}$$

$$\hat{H}_{\text{env}}(p, x) = \sum_j \frac{p_j^2}{2m_j} + \frac{1}{2}m_j\omega_j^2\left(x_j - \frac{c_j\hat{s}}{m\omega_j^2}\right)^2, \tag{2}$$

where $\hat{H}_0$ is the Hamiltonian of the D-dimensional system of interest shifted along the adiabatic path [32]. If the quantum system can be described by a two-level Hamiltonian, then $\hat{H}_0 = \epsilon\hat{\sigma}_z - \hbar\Omega\hat{\sigma}_x$, where $\hat{\sigma}_z$ and $\hat{\sigma}_x$ are the Pauli matrices. Here, $\hat{H}_{\text{env}}$ represents the Hamiltonian of the reservoir or environment modes which are coupled to some system operator $\hat{s}$. The strength of the jth oscillator is c_j. While we are using a time-independent Hamiltonian for simplicity, time dependence from an external field in the system Hamiltonian can be captured through the corresponding system propagator in a straightforward manner.

For a problem where the environment is in thermal equilibrium at an inverse temperature $\beta = \frac{1}{k_B T}$, and its final states are traced out, the interactions between the system and the environment are characterized by the spectral density [14,29]:

$$J(\omega) = \frac{\pi}{2} \sum_j \frac{c_j^2}{m\omega_j}\delta(\omega - \omega_j). \tag{3}$$

In fact, the spectral function $S(\omega)$ corresponding to the collective bath operator $X = -\sum_j c_j x_j$ is related to the spectral density as follows [33]:

$$S(\omega) = \frac{2\hbar J(\omega)}{1 - \exp(-\beta\hbar\omega)}. \tag{4}$$

For environments defined by atomic force fields or *ab initio* calculations, it is often possible to evaluate the spectral density from classical trajectory simulations.

If the initial state can be expressed as a direct product of the system's initial RDM and the bath's thermal density, then the dynamics of the system RDM is given by

$$\langle s_N^+|\rho(N\Delta t)|s_N^-\rangle = \sum_{s_0^\pm}\sum_{s_1^\pm}\ldots\sum_{s_{N-1}^\pm}\langle s_N^+|\hat{U}|s_{N-1}^+\rangle\langle s_{N-1}^+|\hat{U}|s_{N-2}^+\rangle\ldots\langle s_1^+|\hat{U}|s_0^+\rangle\langle s_0^+|\rho(0)|s_0^-\rangle\langle s_0^-|\hat{U}^\dagger|s_1^-\rangle\ldots\langle s_{N-1}^-|\hat{U}^\dagger|s_N^-\rangle F[\{s_j^\pm\}], \tag{5}$$

$$\text{where} \quad F[s_0^\pm, s_1^\pm, \ldots, s_N^\pm] = \exp\left[-\frac{1}{\hbar}\sum_{k=0}^N(s_k^+ - s_k^-)\sum_{k'=0}^k(\eta_{kk'}s_{k'}^+ - \eta_{kk'}^*s_{k'}^-)\right]. \tag{6}$$

Here, $\hat{U}$ is the short-time system propagator for Δt, and the system forward and backward paths are discretized in steps of Δt. The forward and backward states of the system at the jth point of time are denoted by s_j^+ and s_j^-. The system would be represented by D states. How D is determined is dependent on the type of the system [31]. If the system is discrete by its very nature, D is easily known ($D = 2$ for a spin system). For problems that are not intrinsically discrete, a discrete variable representation can be employed [34–36]. In such cases, the value of D would be chosen to accurately simulate the partition function at the given temperature. The dimensionality of a forward-backward state at the jth time point, denoted as $s_j^\pm$, consequently is D^2. While the forward-backward state of the system at time point j is typically denoted by $s_j^\pm$, for the sake of notational convenience, here, we use s_j as a shorthand, especially when the forward-backward state is used as an index of a tensor. The Feynman-Vernon influence functional $F[s_0^\pm, \ldots, s_N^\pm]$ [15] is dependent on the full

system path discretized as $s_0^\pm, s_1^\pm, \ldots, s_N^\pm$ and the bath response function that is discretized as the $\eta_{kk'}$ coefficients [34,35,37]. The influence functional depends upon the history of the system path, leading to the well-known non-Markovian nature of system-environment decomposed quantum dynamics. Notice that it can be factorized based on the range of interaction in the following manner:

$$F[s_0^\pm, s_1^\pm, \ldots, s_N^\pm] = \prod_{\alpha=0}^N\prod_{k=\alpha}^N I_{s_k, s_{k-\alpha}}^{(\alpha)}, \tag{7}$$

$$I_{s_{k'}, s_k}^{(\alpha)} = \exp\left[-\frac{1}{\hbar}(s_k^+ - s_k^-)(\eta_{kk'}s_{k'}^+ - \eta_{kk'}^*s_{k'}^-)\right]\delta_{k', k-\alpha}. \tag{8}$$

The influence functional creates pairwise interactions between points that are temporally separated. As it has been shown, if MPS and matrix product operators (MPO) are used to model the influence functional, the fact that these interactions can spread across long temporal spans leads to an
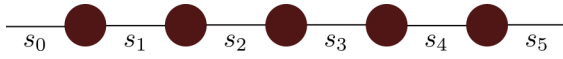
FIG. 1. Diagram for $P^{(1)}$ for a five-step propagation. Dark brown circles represent the K tensors.

FIG. 3. Diagram for $P^{(2)}$ for a five-step propagation. The darker circles which form the base represent the $\mathbb{K}$ tensors. The comparatively lighter red squares forming the second layer represent the $I^{(2)}$ tensors.

increase in the effective bond dimension. Here, the goal is to create a structure that naturally and efficiently accounts for the pairwise interactions that span long temporal separations while not being associated with any one particular representation. To motivate the TN representation, first consider the Markovian part of Eq. (5), involving just the propagators and the terms of the influence functional coupling consecutive time points. These terms can be simply rearranged as

$$P^{(1)}_{s_0,s_N} = K_{s_0,s_1} K_{s_1,s_2} \dots K_{s_{N-1},s_N}, \qquad (9)$$

$$K_{s_{j-1},s_j} = \langle s_j^+|\hat{U}|s_{j-1}^+\rangle\langle s_{j-1}^-|\hat{U}^\dagger|s_j^-\rangle I^{(1)}_{s_{j-1},s_j} I^{(0)}_{s_j,s_j}, \quad j \geq 2, \quad (10)$$

$$K_{s_0,s_1} = \langle s_1^+|\hat{U}|s_0^+\rangle\langle s_0^-|\hat{U}^\dagger|s_1^-\rangle I^{(1)}_{s_0,s_1} I^{(0)}_{s_1,s_1} I^{(0)}_{s_0,s_0}. \quad (11)$$

Here, we are implicitly summing over repeated indices that do not appear on both sides of the equation. Once again, the $\pm$ labels on the forward-backward states that act as the site indices of the tensors are omitted for convenience of notation. The superscript 1 on P is there to denote the maximum distance of interaction that we have incorporated. Equation (9) is already a TN; more specifically, it is a series of matrix multiplication, as shown in Fig. 1. Let us now bring the next-nearest neighbor interactions $I^{(2)}_{s_{j-2},s_j}$. Clearly, it is not possible to directly contract the $I^{(2)}_{s_{j-2},s_j}$ tensor to the $P^{(1)}$ tensor because the internal s_j have already been traced over. To make it possible to incorporate the $I^{(2)}$ tensors, we augment the K tensors as follows:

$$\mathbb{K}^{r_1}_{s_0,s_1} = K_{s_0,s_1}\delta_{s_0,r_1}, \qquad (12)$$

$$\mathbb{K}^{l_{N-1}}_{s_{N-1},s_N} = K_{s_{N-1},s_N}\delta_{s_N,l_{N-1}}, \qquad (13)$$

$$\mathbb{K}^{l_{j-1},r_j}_{s_{j-1},s_j} = K_{s_{j-1},s_j}\delta_{s_j,l_{j-1}}\delta_{s_{j-1},r_j} \quad \text{if } j \neq 1 \text{ and } N. \qquad (14)$$

The topology of the $\mathbb{K}$ tensors are demonstrated in Fig. 2. It is convenient to think of lower indices as the *input* indices and the upper indices as the *output* indices, though there is no other mathematical significance to the positioning of the indices. Also, in the context of the diagram, the l and r indices point to the left and right directions, respectively. With this

input-output convention in mind, it is easy to see that the internal augmented $\mathbb{K}$ tensors duplicate and flip the order of the input indices s_j. This ensures that indices that differ by two time steps are now placed adjacent in the output layer. Now the Markovian terms and the $I^{(2)}$ terms can be combined, and we get

$$P^{(2)}_{s_0,s_N} = \mathbb{K}^{r_1}_{s_0,s_1} \prod_{j=1}^{N-2} I^{(2)}_{r_j,l_j} \mathbb{K}^{l_j,r_{j+1}}_{s_j,s_{j+1}} I^{(2)}_{r_{N-1},l_{N-1}} \mathbb{K}^{l_{N-1}}_{s_{N-1},s_N}, \qquad (15)$$

which is depicted in Fig. 3. Notice that the index s_j still connects $\mathbb{K}_{s_{j-1},s_j}$ and $\mathbb{K}_{s_j,s_{j+1}}$, as in $P^{(1)}$, but now there is another connection that goes through the $I^{(2)}$ tensor in a triangular form. This feature of an $I^{(\alpha)}$ with a higher α acting as a bridge between $\mathbb{K}$ or I tensors with smaller values of α would become a recurring motif in this TN.

The pattern for inclusion of the rest of the nonlocal interactions is quite similar. Note that, in Fig. 3, if we did the same trick of duplicating and flipping the order of the inputs, in the next layer, indices that differ by three time points, like s_0 and s_3, s_1 and s_4, are going to be adjacent. The augmented influence functional tensors are denoted by $\mathbb{I}$, and their diagrams are shown in Fig. 4. Hence, this can now be multiplied by $I^{(3)}_{s_{j-3},s_j}$. Continuing like this, we can complete the network. The diagram is shown in Fig. 5 [38]; these augmented tensors are going to be written as $\mathbb{K}$, $\mathbb{I}^{(2)}$, ... The TN shown in Fig. 5, which we will conventionally denote by $P^{(\infty)}$, represents the final Green's function for the propagation of the system RDM having incorporated the nonlocal influence from the environment. Thus, $\rho(t) = P^{(\infty)}\rho(0)$.

In Fig. 5, all the indices have D^2 dimensionality corresponding to each of the possible combination of forward-backward states. However, this is not optimal. The influence functional tensors for a time difference of α can be massaged
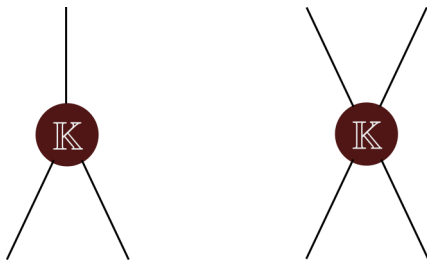

FIG. 2. Diagrams for the $\mathbb{K}$ tensors. The edge tensor is on the left and the internal tensors on the right. For the edge tensor, the output index is a left index if it is a right edge tensor [Eq. (13)] and a right index if it is a left edge tensor [Eq. (12)].
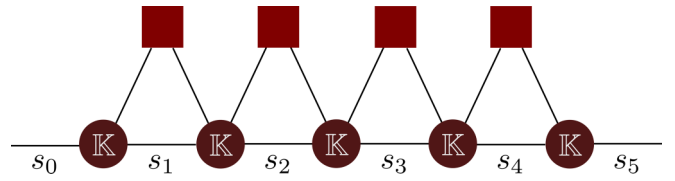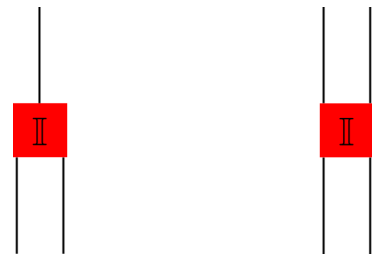

FIG. 4. Diagrams for the $\mathbb{I}$ tensors. The edge tensor is on the left and the internal tensors on the right. The exact color would vary to indicate the distance of interaction encoded by the influence functional term. The same conventions as in Fig. 2 apply.
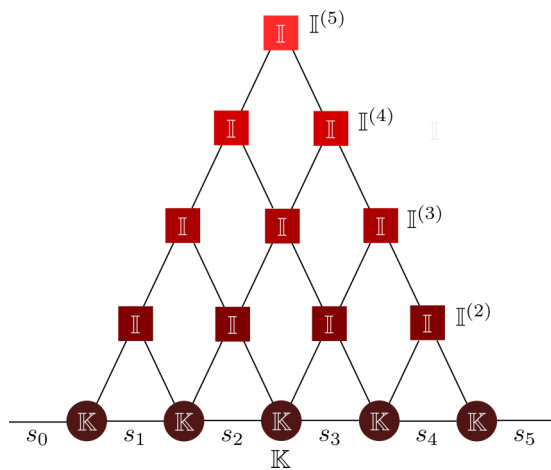
FIG. 5. Diagram for the final Green's function for a five-step propagation. Dark brown circles represent the $\mathbb{K}$ tensors. The various red squares represent the $\mathbb{I}$ tensors for different separations. The lighter shades of red show a larger separation between interacting time points. The indices that point to the top-left from the base of the triangle would be said to be a left index, and those that point to the top-right would be said to be a right index carrying on the convention set up for the $\tilde{\mathbb{K}}$ tensors.

in the following manner:

$$I^{(\alpha)}_{s_{k-\alpha},s_k} = \exp[-\Delta s_k(\eta_{k,k-\alpha}s^+_{k-\alpha} - \eta^*_{k,k-\alpha}s^-_{k-\alpha})], \quad (16)$$

where $\Delta s_k = s^+_k - s^-_k$. This expression depends only on the difference coordinates or blip values Δs_k. Thus, currently, we are carrying over more information than we need to. By transforming this coordinate $s^\pm_k$ to average or sojourn values and difference notation $\bar{s}_k = \frac{1}{2}(s^+_k + s^-_k)$ and Δs, respectively, it can be seen that there is a symmetry with respect to the average forward-backward position or the sojourn value of the latter or the kth time point. The blip-summed PI (BSPI) method [17,18] formulation achieves a complete transformation of the real-time PI algorithm in terms of these blip and average value coordinates, allowing for effective grouping, filtering, and summing of paths according to the distribution of these blips. However, with PC-TNPI, a partial transformation is possible, where only the latter time point k is transformed into its blip value, and the earlier time point still continues in the forward-backward form. In fact, significant compaction and performance gain can be achieved with minimal change in the TN discussed until now, simply by reducing the dimensionality of all the left indices that carry the excess information.

To do that, we need to start by redefining the $\mathbb{K}$ tensors. In Eqs. (12)–(14), the output indices l and r are exactly the same as the input indices except flipped in order. The l indices, consequently, carry the full information about the forward-backward state that occurs later in time, which is unnecessary. Now these tensors can be modified such that the l indices carry only the blip value or the value of Δs as follows:

$$\tilde{\mathbb{K}}^{r_1}_{s_0,s_1} = \mathbb{K}^{r_1}_{s_0,s_1}, \quad (17)$$

$$\tilde{\mathbb{K}}^{l_{N-1},s_N}_{s_{N-1},s_N} = K_{s_{N-1},s_N}\delta_{s^+_N-s^-_N,l_{N-1}}, \quad (18)$$

$$\tilde{\mathbb{K}}^{l_{j-1},r_j}_{s_{j-1},s_j} = K_{s_{j-1},s_j}\delta_{s^+_j-s^-_j,l_{j-1}}\delta_{s_{j-1},r_j} \quad \text{if } j \neq 1 \text{ and } N. \quad (19)$$
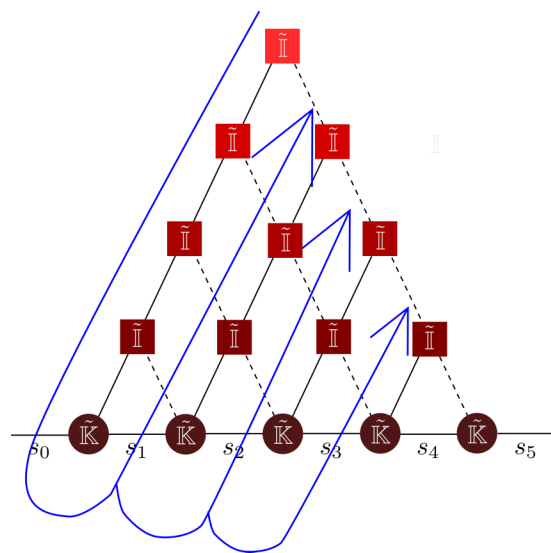

FIG. 6. Optimized diagram for the final Green's function for a five-step propagation. Dashed lines have dimension B, and solid lines carry dimension D^2. Blue arrows show the order of contraction.

Notice that the upper-right output indices in the diagrams remain exactly the same. Only the upper-left output index of $\mathbb{K}$ changes. Therefore, the tensor $\mathbb{K}^{r_1}_{s_0,s_1}$ remains unchanged. The dimensionality of the l indices is the number of unique values of the blips $\Delta s = s^+ - s^-$ that the system can have. For the most general D-level system, this value is $B = D^2 - D + 1$ instead of D^2; however, the actual symmetries present in the system might reduce this even further. Finally, the influence functional tensors must be changed to be consistent, viz. $\tilde{\mathbb{I}}^{(\alpha)}_{s_{k-\alpha},\Delta s_k} = \exp\{-\frac{1}{\hbar}\Delta s_k[\eta_{k,(k-\alpha)}s^+_{k-\alpha} - \eta^*_{k,(k-\alpha)}s^-_{k-\alpha}]\}$. Even with these changes, the basic topology of the network remains the same. The new network with the different dimensions is shown in Fig. 6. Here, we introduce a bit of terminology to facilitate the discussion of the network. The solid edges in Fig. 6 carry the full information of the forward-backward state and hence would be said to belong to the *forward-backward space*. On the other hand, the dashed lines carry only the information of Δs and hence shall be said to belong to the Δs *space* or the *blip space*.

Having discussed the TN, now let us turn to the job of contracting it. Typically, many TNs are constructed using singular value decomposition (SVD) and evaluated via the truncation of the singular values [23,26]. The PC-TNPI is constructed without resorting to any SVD calculations and consequently exact. The goal now is to find an optimal contraction scheme that preserves this exactness. The storage cost S is also evaluated at the end of every step. The canonical contraction order that we discuss below has been marked out in cyan arrows in Fig. 6. For a simulation with N time steps:

(1) Start with $\tilde{\mathbb{I}}^{(N)}_{s_0,s_N}$ and contract it with $\tilde{\mathbb{I}}^{(N-1)}_{s_0,s_{N-1}}$. Thus, $S = D^2 B^2$.

(2) Multiply by $\tilde{\mathbb{I}}^{(N-2)}_{s_0,s_{N-2}}$. Thus, $S = D^2 B^3$.

(3) Multiply all $\tilde{\mathbb{I}}^{(\alpha)}_{s_0,s_\alpha}$ followed by $\tilde{\mathbb{K}}_{s_0,s_1}\rho_{s_0}$. At this stage, the storage cost is $S = D^2 B^{N-1}$. It is interesting to note the various indices at this stage. The index corresponding to the zero time s_0 has been completely contracted away, s_1 is in the

forward-backward space, while all the other sites carry only the information about Δs.

(4) Contract the second edge sequentially, starting from $\mathbb{K}_{s_1,s_2}$. Thus, $S = (D^2)^2 B^{N-2}$.

(5) While contracting the remaining $N-3$ tensors on the second edge, the storage cost remains constant at $S = (D^2)^2 B^{N-2}$.

(6) Lastly, the topmost tensor on the second edge needs to be contracted. The storage drops to $S = D^2 B^{N-2}$. Again, after contracting the second edge, the resulting tensor does not have the s_0 and s_1 indices, s_2 is in the forward-backward space, and everything else is in the Δs space.

(7) Continuing in the same fashion, the storage requirements of contracting the internal tensors of the jth edge is $S = (D^2)^2 B^{N-j}$ when $j < N$.

(8) After contracting the final tensor on the jth edge, the storage drops to $S = D^2 B^{N-j}$.

(9) Finally, the last tensor $\mathbb{K}_{s_{N-1},s_N}$ is contracted.

In the above contraction scheme, we multiply the initial condition ρ_{s_0} and get the final RDM. While this leads to a more efficient algorithm in terms of the storage and computational cost, it is possible to reformulate the scheme in terms of the Green's function by not involving the initial condition in the contractions and evaluating $P^{(\infty)}$. An in-depth analysis of the memory and computational cost is given in Appendix. Of course, the storage requirement grows to a maximum of $(D^2)^2 B^{N-2}$ before decreasing continuously. As shown by the computational and storage cost (derived in detail in Appendix), the PC-TNPI is already quite usable, exponentially outperforming traditional iterative quasi-adiabatic propagator PI (QuAPI) [34,35] and performing on par with the iterative BSPI [17–19]. However, it does not address the problem of storage, which continues to scale exponentially. In future work, filtration schemes on top of the PC-TNPI, giving it a row-wise matrix-product representation, will be introduced that can not only deal with this problem but will also avoid the construction and storage of the full tensor. The focus of this paper is, however, on the general TN, which is quite performant even in the most naïve implementation.

A short discussion of the connection of this TN with the traditional QuAPI and the newer TN-based PI approaches would be helpful in better explaining the structure. While most methods are based on the ability to assign an amplitude to either a single path or a group of paths collectively, here, however, it is not possible in a simple manner. As the TN is evaluated, there is no point in time where all the points along the discretized system path are available simultaneously. The complete evaluation of the network finally yields the time-propagated RDM or the Green's function depending on whether the initial state is incorporated during contraction or not. More precisely, the so-called path-amplitude tensor for a PI of N steps is a tensor with indices $s_0, s_1, \ldots, s_N$ that gives the amplitude corresponding to a particular discretized system path. The contraction algorithm is built in a way that at no point is such a tensor available.

The existing TN methods are based on MPS and MPO [23,39]. One can think of them as different approaches of representing the base of the triangle in Fig. 5 or Fig. 6 in the matrix product form. It is currently not known if this is the most optimal representation possible. While in this paper
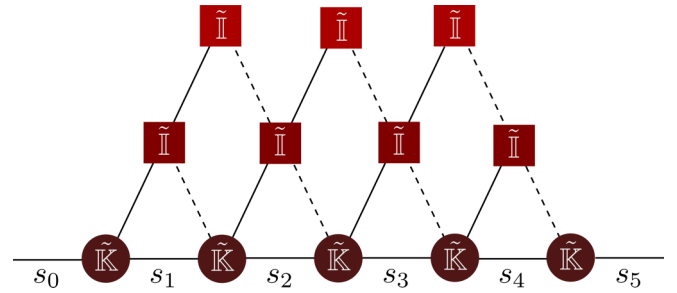

FIG. 7. Diagram for the final Green function for a five-step propagation with memory length $L = 3$.

we have outlined and analyzed the optimal brute force contraction scheme, it is by no means the only way to evaluate the network. Because this PC-TNPI can be contracted directly, it can form a basis for future investigations for different decompositions of the same. It is possible that the optimal filtering procedure built on top of the PC-TNPI would lead to a completely different matrix product representation of the problem.

It is well known that, in condensed phase dissipative environments, the nonlocal memory of the influence functional dies away with the distance between the points, allowing for a truncation of memory. This idea is commonly used both in Nakajima-Zwanzig generalized quantum master equations [40–42] and iterative QuAPI [34,35]. The length of this nonlocal memory is related to the the time scales of the bath response functions. Computationally, it is treated as a convergence parameter, which is increased until the dynamics stops changing. In the framework of the PC-TNPI, the length of the memory is equal to the depth of the resultant network. The topmost tensor encodes the interaction between the most distant points, while the bottom-most tensor captures the Markovian interactions coming through the propagator and the $I^{(1)}$ terms. At two time steps of memory, that is, $L = 2$, we basically get Fig. 3. In Fig. 7, we show the structure of the network for a five-step propagation with $L = 3$. Because s_0 does not interact with s_4 or s_5, it is not necessary to store and evaluate the full diagram at once, but it can be built iteratively. The first edge corresponding to interactions with s_0 is contracted and multiplied by the second edge using the canonical contraction scheme discussed previously. As soon as this is done, the storage of the first edge can be freed, and the third edge can be contracted. This iteration scheme turns out to be identical to the iteration scheme in any iterative PI method like the iQuAPI or the iBSPI. The first steps of the iteration algorithm is pictorially outlined in Fig. 8. First, the left-most edge is contracted as shown in Fig. 8(a). This is followed by contraction of the second edge from the left [Fig. 8(b)]. Note that in the iterative regime, both the left-most edge and the second edge from the left have the same number of tensors. Finally, the rest of the TN is contracted in an identical fashion to the contraction discussed before for the full memory simulations. This is schematically shown in Fig. 8(c).

Makri [17] has shown that it is possible to think of the memory as arising from two different causes. The influence functional F can be rewritten in terms of the real and imagi-

(a) Contract all influence functionals with s_0.

(b) Contract result with all influence functionals with s_1. Notice that the external index with s_1 from the previous step has been contracted out and now the bottom-most external index is s_2.

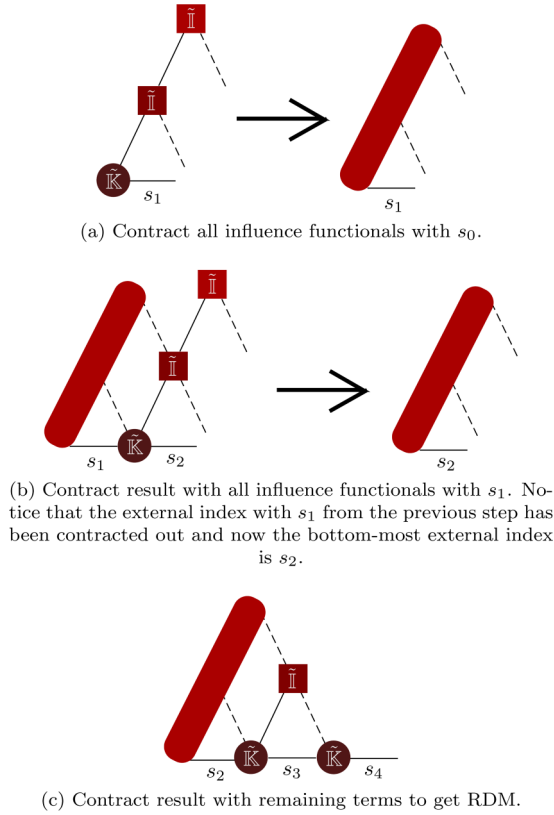(c) Contract result with remaining terms to get RDM.

FIG. 8. First steps of algorithm for iteration. The basic contractions are done in the same way as described for the full path part.

nary parts of the η coefficients as

$$F[s_0^\pm, s_1^\pm, \ldots, s_N^\pm]$$
$$= \exp\left[-\frac{1}{\hbar}\sum_k \Delta s_k \sum_{k'\leqslant k}(\Re\eta_{kk'}\Delta s_{k'} - 2i\Im\eta_{kk'}\bar{s}_{k'})\right], \quad (20)$$

where $\Delta s_k = s_k^+ - s_k^-$ and $\bar{s}_k = \frac{1}{2}(s_k^+ + s_k^-)$. The part of the influence functional that arises from $\Re\eta_{kk'}$ is called the *classical decoherence factor*. It corresponds to stimulated phonon absorption and emission [43]. This can also be obtained through classical trajectory simulations and reference propagators [44,45] in a Markovian manner. All effects of temperature are captured in the classical decoherence term. The term with the $\Im\eta_{kk'}$ is the backreaction that leads to quantum decoherence. This part of the memory is truly nonlocal and temperature independent.

The inclusion of solvent trajectory-based reference propagators is trivially possible in the PC-TNPI using the framework of harmonic backreaction [43]. However, it is interesting to note that it is possible to increase the efficiency of the approximate simulations involving only the classical decoherence even without these reference propagators by judiciously using the TN structure. This approximate dynamics would become increasingly accurate as the temperature of the simulation rises. In this, the full $\eta_{kk'}$ coefficients are used only when $k = k'$ or $k' + 1$, and otherwise, the imaginary part of $\eta_{kk'}$ is ignored. Effectively, we are modifying the $I_{s_{k'}, s_k}^{(\alpha)}$ operators to be $\exp[-1/\hbar\Re(\eta_{kk'})\Delta s_k\Delta s_{k'}]$ when $\alpha = k - k' \geqslant 2$. Just like before when the s_k lines carried unnecessary infor-

mation, now the $s_{k'}$ lines carry more information than they need to. We only need to know about $\Delta s_{k'}$. Thus, we can make the required changes to the dimensionality of the indices by putting in the corresponding projector operators in the $\mathbb{K}$ tensors, thereby reducing the cost of computation even further. The network for the classical decoherence simulations would have exactly the same structure as Fig. 5 with all edges except the base ones being B dimensional. This approximation is especially accurate at short times.

The scaling of the naïve contraction algorithm has been derived in Appendix. For a simulation of N time steps with a memory length of L, the computational complexity scales as $\mathcal{O}[(N - L)B^{L-1}]$, and the storage complexity scales as $\mathcal{O}(B^L)$, where B is the number of unique blip values corresponding to the system.

III. RESULTS

As illustrative examples, we apply the PC-TNPI to a two-level system (TLS) coupled bilinearly to a dissipative environment:

$$\hat{H}_0 = \epsilon\hat{\sigma}_z - \hbar\Omega\hat{\sigma}_x, \quad (21)$$

where $\hat{\sigma}_z$ and $\hat{\sigma}_x$ are the Pauli spin matrix. The TLS model is not just useful for describing spins but also for deep tunneling processes where the lowest doublet is enough to describe the entire process.

The dissipative environment is chosen to be defined by Ohmic model spectral densities, which are especially useful in modeling the low frequency rotranslational modes. We use the very common Ohmic form with an exponential cutoff:

$$J(\omega) = \frac{\pi}{2}\hbar\xi\omega\exp\left(-\frac{\omega}{\omega_c}\right), \quad (22)$$

where ξ is the dimensionless Kondo parameter, and ω_c is the characteristic cutoff frequency; and the Ohmic form with a Drude cutoff:

$$J(\omega) = \kappa\omega_c\frac{\omega}{\omega^2 + \omega_c^2}, \quad (23)$$

where κ is a measure of the coupling strength. Generally, these model spectral densities are often thought to be fully characterized by a reorganization energy:
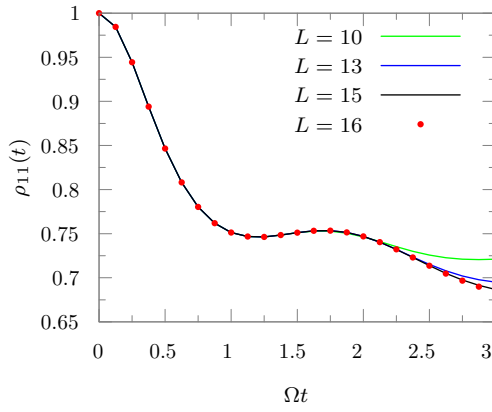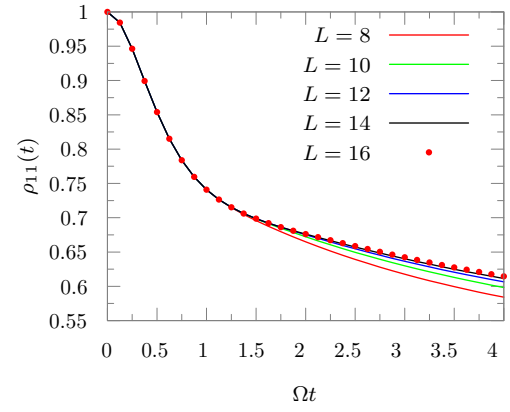
$$\lambda = \frac{2}{\pi}\int_{-\infty}^{\infty}\frac{J(\omega)}{\omega}d\omega, \quad (24)$$

and the cutoff frequency ω_c. The reorganization energies for the exponential and the drude cutoff spectral densities are as listed below:
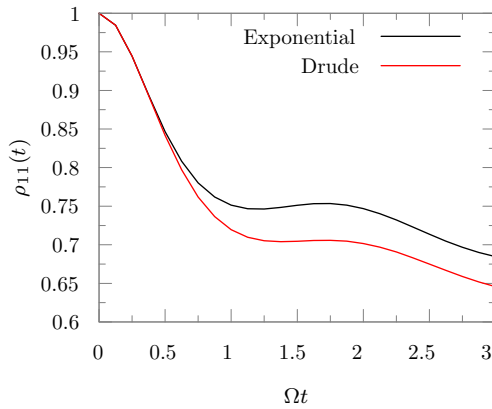
$$\lambda_{\text{Exp}} = 2\xi\omega_c, \quad (25)$$

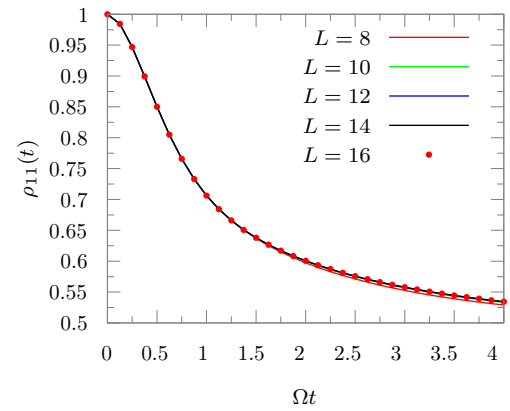$$\lambda_{\text{Drude}} = 2\kappa. \quad (26)$$

As we demonstrate through the examples, though the reorganization energy and the cutoff frequency are same, the exact dynamics of the RDM is highly dependent on the form of the decay function. Consider a symmetric TLS ($\epsilon = 0$) and $\Omega = 1$ interacting strongly ($\xi = 2$) with a sluggish bath ($\omega_c = \Omega$) initially localized on the populated system state 1. The bath has a reorganization energy of $\lambda = 4$ and is held at an inverse temperature of $\hbar\Omega\beta = 1$. The dynamics was converged at $\Delta t = 0.125$ and a memory length $L = 16$ for

(a) Convergence with respect to memory length, L.

(b) Comparison between spectral densities with exponential and Drude decay functions.

FIG. 9. Dynamics of a symmetric two-level system (TLS) interacting with a bath with $\lambda = 4$, $\omega_c = \Omega$ at an inverse temperature $\hbar\Omega\beta = 1$.

the time scales shown. The convergence is shown in Fig. 9(a) for an Ohmic bath with an exponential decay. Full quantum-classical PI simulations for this parameter have been reported for this short time interval [45]. Converged results up to equilibration requires much larger memory lengths of $L \approx 100$, which cannot be achieved without some form of filtration. These results have been reported using the SMatPI [22] and reproduced by the augmented propagator-based TNPI [26]. If the Drude form of decay is used, the dynamics changes quite significantly. The comparison between the dynamics arising from the two spectral densities is shown in Fig. 9(b).

Next, consider a case where not only is the dynamics different between the two different decay functions, but the converged memory length is different as well. The dynamics of the same TLS as above ($\epsilon = 0$, $\Omega = 1$) is now simulated in a bath with the reorganization energy $\lambda = 8$ and a characteristic cutoff frequency $\omega_c = 5$. The bath is equilibrated at an inverse temperature of $\hbar\Omega\beta = 5$. The time step is converged at $\Omega\Delta t = 0.125$. The convergence of the dynamics of the RDM on changing the memory length L is shown in Fig. 10. While the memory length for the exponential decay function spectral density is quite close to convergence at $L = 14$, for the Drude spectral function, it converges at $L = 10$.

(a) Dynamics corresponding to spectral density with exponential decay functions

(b) Dynamics corresponding to spectral density with Drude decay functions

FIG. 10. Convergence with respect to memory length for the spectral density with an exponential cutoff.

In Fig. 11, we consider a TLS coupled to a strongly coupled Ohmic bath with an exponential cutoff ($\xi = 1.2$, $\omega_c = 2.5\Omega$) equilibrated at a high temperature $\hbar\Omega\beta = 0.2$. The converged time step is $\Delta t = 0.125$. The classical memory calculations converge at a comparatively lower memory length L and agree quite well with the full simulations at short times.
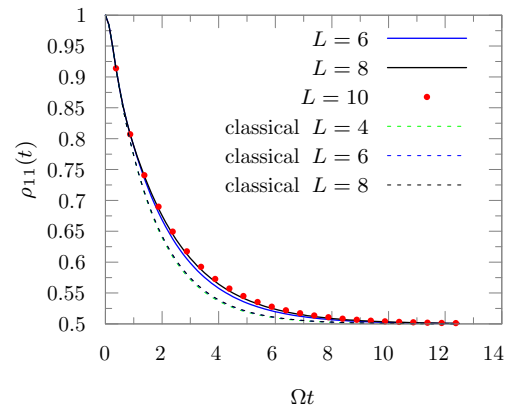

FIG. 11. Comparison between the classical and full memory calculations for a strongly coupled high-temperature bath.
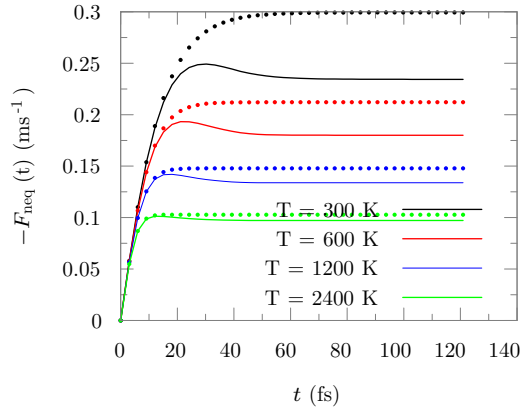
FIG. 12. Rate according to the nonequilibrium flux function. Markers: Classical memory approximate simulations. Line: Full simulations without any approximation.
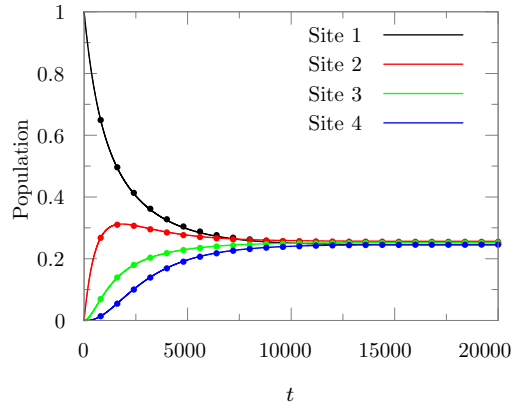

FIG. 13. Population dynamics corresponding to an initially populated first site. Lines: full simulations, markers: classical memory simulations. For full decoherence, $L = 6$; for classical decoherence, $L = 4$.

For many processes that have timescales too long to be simulated directed, rate theory provides a computatonally efficient alternative. Quantum rate theory is formulated in terms of equilibrium correlation functions involving reactive flux [46–48]. Recently, it has also been shown that it is possible to get the same information from nonequilibrium initial conditions and simulations [39,49]. Consider a model of a typical symmetric proton transfer or isomerization reaction where the tunneling splitting is significantly smaller than the vibrational frequencies of the environment. We consider the symmetric TLS ($\epsilon = 0$) studied in Refs. [49,50] with $\hbar\Omega = 0.000525 \, \mathrm{cm}^{-1}$. The strongly coupled bath is characterized by an Ohmic spectral density with an exponential cutoff at the cutoff frequency of $\omega_c = 500 \, \mathrm{cm}^{-1}$ and $\xi = 0.5$. The rate is obtained as a long-time limit of the nonequilibrium flux function corresponding to the reactant state $|1\rangle$, $\hat{F} = \frac{i}{\hbar}[\hat{H}_0, |1\rangle\langle 1|]$, whose time evolution at different temperatures ranging from $T = 300$ to $2400 \, \mathrm{K}$ is shown in Fig. 12. As can be seen, at all the temperatures, the timescales indicated by the classical-memory-only simulation, though not exactly accurate, are consistent with the full-rate-theory calculations. It is also seen that the accuracy of the classical memory simulations increase with temperature.

As a final example, consider a molecular wire described by the tight-binding Hamiltonian involving D sites:

$$\hat{H}_0 = \sum_{1 \leqslant j \leqslant D} \epsilon_j |\sigma_j\rangle\langle\sigma_j| - \hbar V \sum_{1 \leqslant j < D} (|\sigma_j\rangle\langle\sigma_{j+1}| + |\sigma_{j+1}\rangle\langle\sigma_j|). \tag{27}$$

The site energy of the jth site is ϵ_j, and the nearest neighbor couplings are V. The sites are separated by unit distance such that $|\sigma_j\rangle$ are eigenstates of the position operator $\hat{s}|\sigma_j\rangle = (j - 1)|\sigma_j\rangle$. The site energy of all but the first site is chosen to be zero $\epsilon_j = 0$ for $j \neq 1$ and $\epsilon_1 = 1$. The intersite coupling is chosen as $V = 0.025$ [51]. The computational cost grows exponentially with the number of sites. To test the efficiency of the basic contraction scheme outlined here, we use a system with $D = 4$ sites. The bath is characterized by an Ohmic spectral density with an exponential cutoff, Eq. (22) with $\omega_c = 4$ and $\xi = 0.12$ [26], equilibrated at an inverse temperature of $\beta = 0.1$. As discussed in Sec. II, the scaling

of the algorithm would go as $B < D^2$. The symmetry of the Hamiltonian enforced by the global bath in this case ensures that the number of unique values of Δs, $B = 7$ for this 4-state system, which is even less than the $D^2 - D + 1 = 13$ for a completely general Hamiltonian. The population dynamics of all the states is shown in Fig. 13. An initial state with only the first site populated was used. Because of the high temperature of the bath, the classical decoherence simulation produces practically identical dynamics but converges at a smaller memory length L.

IV. CONCLUSIONS

A generalized TN is introduced to perform PI calculations involving the Feynman-Vernon influence functional. Not only does this PC-TNPI capture the pairwise interaction structure of the influence functional, but it also benefits from the blip symmetry present in the Feynman-Vernon influence functional [17,18]. The PC-TNPI can be contracted efficiently and minimizes the storage requirements as far as possible without resorting to various path filtration algorithms. Iterative decomposition of the memory is also possible in an elegant manner. Analysis of the space and time complexity of the PC-TNPI shows that it scales like the iBSPI, clearly taking advantage of the blip symmetry.

Recently, TNs have proved to be very useful in simulating the dynamics of open quantum systems [23,26,27]. The PC-TNPI provides an alternative to the commonly used MPS representation [23,26], serving as a small step in further elucidating the deep relation between TNs and PIs. While no path filtration scheme has been developed, the PC-TNPI is already quite usable. It can easily incorporate classical trajectories through harmonic backreaction quantum-classical PIs [43–45,52,53], thereby making it possible to include anharmonic effects of the environment in an approximate manner without any additional cost. Additionally, harmonic backreaction also leads to an increase in the converged time step and a decrease in the effective memory length such that some ultrafast reactions can be simulated directly. The combined method could simulate systems with strongly coupled sluggish realistic solvents with high reorganization energy. This promises

to be a fruitful avenue of research in terms of applications to electron and proton transfer reactions.

The most important aspect of the PC-TNPI is that it gives a framework to unify and develop other TN algorithms for simulating real-time PIs. Algorithms based on MPS representations of the augmented reduced density tensor [23] or of the augmented propagator [26] can be thought of as particular optimized refactorizations of the PC-TNPI network. These methods can be thought of as different approaches of representing the base as an MPS. We have also established the viability of evaluating the PC-TNPI in a brute force manner. This suggests that using the PC-TNPI directly to generate other optimized representations might also lead to methods whose performance should significantly outstrip that of the brute force evaluation of the TN. In this context, there are optimized algorithms for doing calculations with MPSs and MPOs. It would be an interesting future research direction to figure out the optimal representation of the system using an MPS.

While ideas of path filtration were not a consideration of this paper, schemes based on the SVD can be incorporated with the PC-TNPI, leading to a method that significantly reduces the storage since the full tensor would not need to be computed and stored. This development will be discussed in a future publication, further optimizing the framework of the PC-TNPI.

APPENDIX: COST OF CONTRACTION

Consider the TN corresponding to a full path simulation spanning N time steps. To calculate the cost of contraction, the left edge of the triangular network is first considered. Consider contracting $\tilde{\mathbb{I}}^{(j)}_{s_0,s_j}$, for $j \geqslant 2$, with two D^2 indices and one B index, as schematically indicated in Fig. 14(a). The part that has already been contracted has one D^2 index and $(N - j)$ B indices. Therefore, the cost of contraction is $(D^2)^2 B^{N-j+1}$. The space requirement at this stage is $D^2 B^{N-j+1}$. To finish the contraction of the left-most edge of the triangle, we need to multiply by $\tilde{\mathbb{K}}_{s_0,s_1} \rho_{s_0}$, leading to the TN shown in Fig. 14(b). The resultant tensor does not have an index corresponding to s_0 because that has been traced over. The computational cost of this step is $C = (D^2)^2 B^{N-1}$, and the storage becomes $S = D^2 B^{N-1}$. Now the second parallel edge is to be contracted. This step, however, is started from the bottom, i.e., from $\tilde{\mathbb{K}}_{s_1,s_2}$. The first contraction, shown in Fig. 15(a), is the most costly step in the entire algorithm. The computational cost of this step is $C = (D^2)^3 B^{N-1}$, and the storage requirement increases to $S = (D^2)^2 B^{N-2}$. Continuing with the other intermediate tensors of the first parallel edge, notice that the cost of contraction remains constant at $C = (D^2)^3 B^{N-1}$, and the space required remains constant at $S = (D^2)^2 B^{N-2}$. Finally, the last, top-most tensor of this edge is to be contracted. This is illustrated in Fig. 15(b). The computational cost is $C = (D^2)^2 B^{N-1}$. The storage cost now drops to $S = D^2 B^{N-2}$.
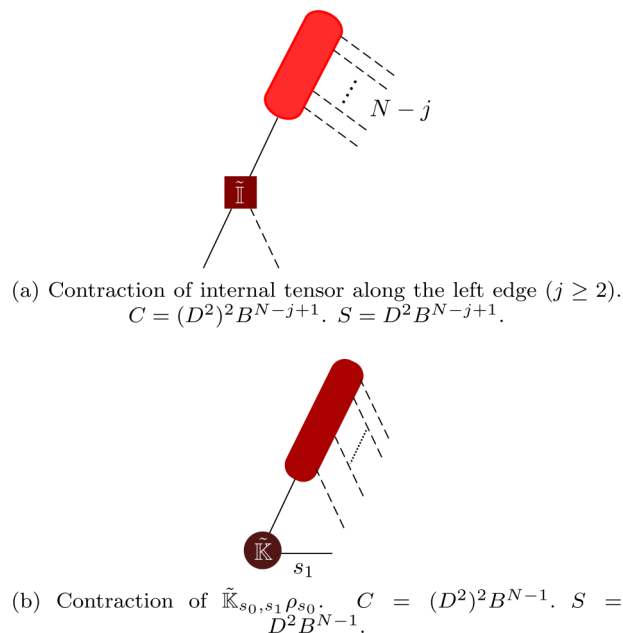
(a) Contraction of internal tensor along the left edge $(j \geq 2)$. $C = (D^2)^2 B^{N-j+1}$. $S = D^2 B^{N-j+1}$.

(b) Contraction of $\tilde{\mathbb{K}}_{s_0,s_1} \rho_{s_0}$. $C = (D^2)^2 B^{N-1}$. $S = D^2 B^{N-1}$.

FIG. 14. Contraction along the left edge of the triangle. Dashed lines show the B-dimensional indices, and solid lines show the D^2-dimensional indices.

Now consider contracting a general diagonal edge, say, the jth one. The resultant tensor from the previous contraction has one D^2 index and $(N - j + 1)$ B indices. Contracting the $\tilde{\mathbb{K}}$ tensor leads to a tensor with two D^2 indices and $(N - j)B$ indices. The cost of this contraction is $C = (D^2)^3 B^{N-j+1}$, and the storage is $S = (D^2)^2 B^{N-j}$. For all the intermediate tensors at this stage, once again, both the computational and storage costs remain the same. On contracting the last tensor of this diagonal, the storage drops to $S = D^2 B^{N-j}$.

Below, we list the total computational cost for contracting each of the parallel edges. The edge number is given

(a) Contracting the first tensor of the next parallel edge. $C = (D^2)^3 B^{N-1}$. $S = (D^2)^2 B^{N-2}$.

(b) Contracting the last tensor of the next parallel edge. $C = (D^2)^2 B^{N-1}$. $S = D^2 B^{N-2}$.
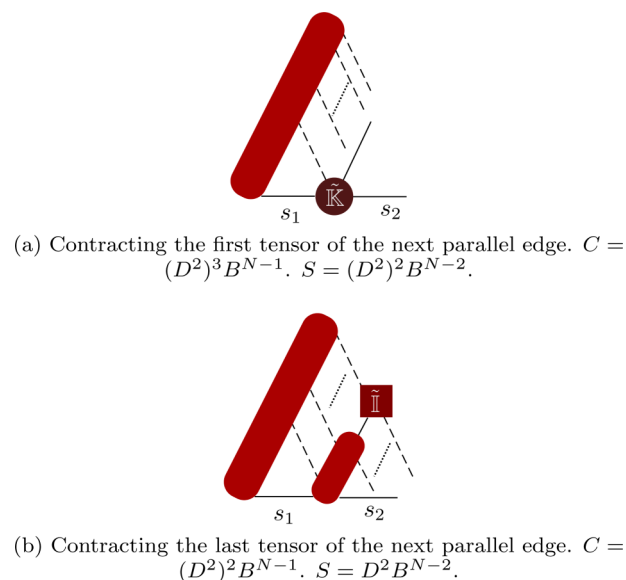
FIG. 15. Contraction along an intermediate edge, say, the one next to the left-most edge.

as the subscript:

$$C_1 = (D^2)^2 B^{N-1} + \sum_{j=2}^{N-1} (D^2)^2 B^{N-j+1}$$

$$= (D^2)^2 \left[B^{N-1} + \frac{B^2(B^{N-2} - 1)}{B - 1} \right], \quad \text{(A1)}$$

$$C_j = (D^2)^2 B^{N-j+1}[1 + (N - j)D^2], \quad 2 \leqslant j \leqslant N. \quad \text{(A2)}$$

The prefactor of the computational and storage costs is lower for classical decoherence simulations: It goes from a power of D^2 to the corresponding power of B. It is clear that the complexity of the entire contraction goes as $\mathcal{O}(B^{N-1})$, and the peak storage requirement is $\mathcal{O}(B^{N-2})$. Beyond memory, the computational cost scales linearly with the number of time steps simulated.

[1] S. R. White, Density Matrix Formulation for Quantum Renormalization Groups, Phys. Rev. Lett. **69**, 2863 (1992).

[2] U. Schollwöck, The density-matrix renormalization group, Rev. Mod. Phys. **77**, 259 (2005).

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

[4] U. Schollwöck, The density-matrix renormalization group: a short introduction, Philos. Trans. R. Soc. London A **369**, 2643 (2011).

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

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

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

[8] M. F. Shibl, J. Schulze, M. J. Al-Marri, and O. Kühn, Multilayer-MCTDH approach to the energy transfer dynamics in the LH2 antenna complex, J. Phys. B: At. Mol. Opt. Phys. **50**, 184001 (2017).

[9] R. Orús, A practical introduction to tensor networks: matrix product states and projected entangled pair states, Ann. Phys. (NY) **349**, 117 (2014).

[10] G. Vidal, Entanglement Renormalization, Phys. Rev. Lett. **99**, 220405 (2007).

[11] G. Vidal, Class of Quantum Many-Body States That Can Be Efficiently Simulated, Phys. Rev. Lett. **101**, 110501 (2008).

[12] S. R. White and A. E. Feiguin, Real-Time Evolution Using the Density Matrix Renormalization Group, Phys. Rev. Lett. **93**, 076401 (2004).

[13] H. Ma, Z. Luo, and Y. Yao, The time-dependent density matrix renormalisation group method, Mol. Phys. **116**, 854 (2018).

[14] N. Makri, The linear response approximation and its lowest order corrections: an influence functional approach, J. Phys. Chem. B **103**, 2823 (1999).

[15] R. P. Feynman and F. L. Vernon, The theory of a general quantum system interacting with a linear dissipative system, Ann. Phys. (NY) **24**, 118 (1963).

[16] N. Makri, Path integral renormalization for quantum dissipative dynamics with multiple timescales, Mol. Phys. **110**, 1001 (2012).

[17] N. Makri, Exploiting classical decoherence in dissipative quantum dynamics: memory, phonon emission, and the blip sum, Chem. Phys. Lett. **593**, 93 (2014).

[18] N. Makri, Blip decomposition of the path integral: exponential acceleration of real-time calculations on quantum dissipative systems, J. Chem. Phys. **141**, 134117 (2014).

[19] N. Makri, Iterative blip-summed path integral for quantum dynamics in strongly dissipative environments, J. Chem. Phys. **146**, 134101 (2017).

[20] N. Makri, Small matrix disentanglement of the path integral: overcoming the exponential tensor scaling with memory length, J. Chem. Phys. **152**, 041104 (2020).

[21] N. Makri, Small matrix path integral for system-bath dynamics, J. Chem. Theory Comput. **16**, 4038 (2020).

[22] N. Makri, Small matrix path integral with extended memory, J. Chem. Theory Comput. **17**, 1 (2021).

[23] A. Strathearn, P. Kirton, D. Kilda, J. Keeling, and B. W. Lovett, Efficient non-Markovian quantum dynamics using time-evolving matrix product operators, Nat. Commun. **9**, 3322 (2018).

[24] C. Guo, K. Modi, and D. Poletti, Tensor-network-based machine learning of non-Markovian quantum processes, Phys. Rev. A **102**, 062414 (2020).

[25] G. E. Fux, E. P. Butler, P. R. Eastham, B. W. Lovett, and J. Keeling, Efficient Exploration of Hamiltonian Parameter Space for Optimal Control of Non-Markovian Open Quantum Systems, Phys. Rev. Lett. **126**, 200401 (2021).

[26] A. Bose and P. L. Walters, A tensor network representation of path integrals: implementation and analysis, arXiv:2106.12523.

[27] A. Bose and P. L. Walters, A multisite decomposition of the tensor network path integrals, J. Chem. Phys. **156**, 24101 (2022).

[28] ITensor Library (version 3.0.0), ITensor Libr. (version 3.0.0) https://itensor.org.

[29] A. O. Caldeira and A. J. Leggett, Path integral approach to quantum Brownian motion, Phys. A: Stat. Mech. Appl. **121**, 587 (1983).

[30] A. O. Caldeira and A. J. Leggett, Quantum tunnelling in a dissipative system, Ann. Phys. (NY) **149**, 374 (1983).

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

[32] N. Makri, Improved Feynman propagators on a grid and nonadiabatic corrections within the path integral framework, Chem. Phys. Lett. **193**, 435 (1992).

[33] J. Leppäkangas, J. Braumüller, M. Hauck, J.-M. Reiner, I. Schwenk, S. Zanker, L. Fritz, A. V. Ustinov, M. Weides, and M. Marthaler, Quantum simulation of the spin-boson model with a microwave circuit, Phys. Rev. A **97**, 052321 (2018).

[34] N. Makri and D. E. Makarov, Tensor propagator for iterative quantum time evolution of reduced density matrices. I. Theory, J. Chem. Phys. **102**, 4600 (1995).

[35] N. Makri and D. E. Makarov, Tensor propagator for iterative quantum time evolution of reduced density matrices. II. Numerical methodology, J. Chem. Phys. **102**, 4611 (1995).

[36] D. E. Makarov and N. Makri, Path integrals for dissipative systems by tensor multiplication. Condensed phase quantum dynamics for arbitrarily long time, Chem. Phys. Lett. **221**, 482 (1994).

[37] T. C. Allen, P. L. Walters, and N. Makri, Direct computation of influence functional coefficients from numerical correlation functions, J. Chem. Theory Comput. **12**, 4169 (2016).

[38] A. Strathearn, Modelling Non-Markovian Quantum Systems Using Tensor Networks, Ph.D. thesis, University of St. Andrews (2020).

[39] A. Bose and N. Makri, Quantum-classical path integral evaluation of reaction rates with a near-equilibrium flux formulation, Int. J. Quantum Chem. **121**, qua26618 (2021).

[40] S. Nakajima, On quantum theory of transport phenomena, Prog. Theor. Phys. **21**, 659 (1958).

[41] R. Zwanzig, Ensemble method in the theory of irreversibility, J. Chem. Phys. **33**, 1338 (1960).

[42] Q. Shi and E. Geva, A new approach to calculating the memory kernel of the generalized quantum master equation for an arbitrary system-bath coupling, J. Chem. Phys. **119**, 12063 (2003).

[43] F. Wang and N. Makri, Quantum-classical path integral with a harmonic treatment of the back-reaction, J. Chem. Phys. **150**, 184102 (2019).

[44] T. Banerjee and N. Makri, Quantum-classical path integral with self-consistent solvent-driven reference propagators, J. Phys. Chem. B **117**, 13357 (2013).

[45] P. L. Walters and N. Makri, Iterative quantum-classical path integral with dynamically consistent state hopping, J. Chem. Phys. **144**, 044108 (2016).

[46] W. H. Miller, Quantum mechanical transition state theory and a new semiclassical model for reaction rate constants, J. Chem. Phys. **61**, 1823 (1974).

[47] W. H. Miller, S. D. Schwartz, and J. W. Tromp, Quantum mechanical rate constants for bimolecular reactions, J. Chem. Phys. **79**, 4889 (1983).

[48] M. Topaler and N. Makri, Quasi-adiabatic propagator path integral methods. Exact quantum rate constants for condensed phase reactions, Chem. Phys. Lett. **210**, 285 (1993).

[49] A. Bose and N. Makri, Non-equilibrium reactive flux: a unified framework for slow and fast reaction kinetics, J. Chem. Phys. **147**, 152723 (2017).

[50] M. Topaler and N. Makri, Quantum rates for a double well coupled to a dissipative bath: Accurate path integral results and comparison with approximate theories, J. Chem. Phys. **101**, 7500 (1994).

[51] R. Lambert and N. Makri, Memory propagator matrix for long-time dissipative charge transfer dynamics, Mol. Phys. **110**, 1967 (2012).

[52] R. Lambert and N. Makri, Quantum-classical path integral. I. Classical memory and weak quantum nonlocality, J. Chem. Phys. **137**, 22A552 (2012).

[53] R. Lambert and N. Makri, Quantum-classical path integral. II. Numerical methodology, J. Chem. Phys. **137**, 22A553 (2012).