Adaptive power method for estimating large deviations in Markov chains

Francesco Coghi ●*

Nordita, KTH Royal Institute of Technology and Stockholm University, Stockholm, Sweden

Hugo Touchette ●†

Department of Mathematical Sciences, Stellenbosch University, Stellenbosch, South Africa

We study the performance of a stochastic algorithm based on the power method that adaptively learns the large deviation functions characterizing the fluctuations of additive functionals of Markov processes, used in physics to model nonequilibrium systems. This algorithm was introduced in the context of risk-sensitive control of Markov chains and was recently adapted to diffusions evolving continuously in time. Here we provide an in-depth study of the convergence of this algorithm close to dynamical phase transitions, exploring the speed of convergence as a function of the learning rate and the effect of including transfer learning. We use as a test example the mean degree of a random walk on an Erdős-Rényi random graph, which shows a transition between high-degree trajectories of the random walk evolving in the bulk of the graph and low-degree trajectories evolving in dangling edges of the graph. The results show that the adaptive power method is efficient close to dynamical phase transitions, while having many advantages in terms of performance and complexity compared to other algorithms used to compute large deviation functions.

I. INTRODUCTION

The application of ideas and techniques from large deviation theory in nonequilibrium statistical physics has led to many new insights about the transport properties of steady-state nonequilibrium systems [1–3] and other interesting phenomena such as dynamical phase transitions [4–11], fluctuation symmetries [12–15], and dissipation bounds [16–20]. Within this theory, the fluctuations of observable quantities, such as particle currents and kinetic activities, are characterized by a function or potential, called the rate function, giving the rate of decay of the probability distribution of an observable in the limit where the observation time or volume of the system considered goes to infinity, similarly to the thermodynamic limit of equilibrium statistical mechanics [21].

Finding the rate function of a given observable and Markov process modeling a nonequilibrium system is difficult in general, as it involves solving a spectral problem or a related optimization problem whose dimension increases with the size of the process considered [22]. As a result, many numerical techniques have been devised over the years to compute the rate function by either solving the spectral or optimization problems using discretization or projection methods [23–27] or by simulating the underlying process in a way that goes beyond direct simulation [28–31], which is obviously inefficient for sampling large fluctuations and rare events in general. On the simulation side, techniques such as cloning [32–36], splitting [37–40], and importance sampling [41–44]

have also been developed and used with good results, although more work is needed to efficiently deal with complex systems involving many interacting particles. Recently, machine learning tools have been used to address this issue [45–48].

In this paper, we focus on a simulation method that adaptively estimates or learns the rate function using a stochastic variation of the power method for computing eigenvalues that combines techniques from importance sampling and stochastic approximations. This method or algorithm, referred to as APM for *adaptive power method*, was introduced in a series of works on stochastic control theory [49–51] and was applied in that context to estimate the likelihood of rare events arising, for example, in communication networks and portfolio optimization. Recently, APM was also applied to continuous-state and continuous-time processes, which require a function representation or projection of their state [52].

Here we investigate the performance of APM when applied to systems showing dynamical phase transitions (DPTs), i.e., phase transitions in the fluctuations, signaled by singularities in the rate function or its dual, the scaled cumulant-generating function [4–11]. The observable and process that we consider as a test case is the mean degree of a uniform random walk on an Erdős-Rényi (ER) random graphs, which appears to show a DPT separating high-degree trajectories of the random walk evolving in the bulk of the random graph and low-degree trajectories evolving in dangling edges of the graph [53–55]. For this system, we study the convergence and accuracy of APM as a function of the system size and learning rate, and also look at the effect of using transfer learning when initializing the algorithm.

Studying DPTs numerically is generally difficult because of the need of simulating large systems and the presence

_____

*francesco.coghi@su.se

†htouchette@sun.ac.za

034137-1

of slowing down effects [33–35]. Our results show that the performance of APM is not affected by DPTs, as it includes importance sampling, which guides the estimation in the relevant region of the process where the phase transition takes place. This importance sampling step can be included in other algorithms, such as cloning, but involves a more complicated feedback mechanism requiring a large number of parallel simulations [34]. By comparison, APM estimates the rate function in a direct way using only one simulated trajectory of a process that is gradually controlled towards the rare event or fluctuation of interest. This makes APM a simple and efficient algorithm for studying DPTs and long-time large deviations in general. Other advantages of APM are discussed together with some technical improvements for dealing with large systems.

II. LARGE DEVIATIONS OF MARKOV CHAINS

We review in this section the spectral method used in large deviation theory to obtain the rate function of time-integrated functions of Markov processes, which is the starting point of APM. For simplicity, we consider the case of homogenous Markov chains evolving in discrete time, which are used to model various processes, including population dynamics, queues, chemical reactions, molecular motors and other physical systems having a discrete number of states [56–58].

A. Markov chain model

We denote by $(X_\ell)_{\ell=1}^n$ the sequence of states or *trajectory* of a Markov chain evolving in a state space Γ over n time steps. We assume that Γ is discrete and finite and that the transition matrix Π, whose element $\Pi(i, j)$ gives the probability of going from the state i to j over one time step, is irreducible and aperiodic, so the Markov chain is ergodic [59]. As a result, it has a unique stationary probability distribution $p^*(i)$, $i \in \Gamma$, satisfying $p^*\Pi = p^*$ in (row) vector notation, which gives the fraction of time spent in each state in the long-time limit.

For this Markov model, we are interested to study the fluctuations of time-additive quantities, costs, or observables having the form

$$C_n = \frac{1}{n} \sum_{\ell=1}^n f(X_\ell), \qquad (1)$$

where f is some real function of the state. This random variable can represent, for example, the mean energy of a system transitioning randomly between energy levels or the fraction of time spent in some state. To be more general, one can also consider observables defined with a function $g(X_\ell, X_{\ell+1})$ instead of $f(X_\ell)$ so as to include jump contributions to C_n. This is commonly done in physics, for example, when considering particle and probability currents, as well as thermodynamic quantities such as the entropy production [60]. The results in this case follow with minor modifications of the results obtained here with $f(X_\ell)$ [61].

B. Large deviations

The theory of large deviations predicts that the probability distribution of C_n, denoted by $P_n(c)$, scales with n according

to

$$P_n(c) \sim e^{-nI(c)}, \qquad (2)$$

with subexponential corrections in n [62–64]. Thus the problem that we consider is that of estimating the decay function $I(c)$ so as to get information about the fluctuations of C_n to leading order in n. This function is called the *rate function* and is defined by the limit

$$I(c) = \lim_{n\to\infty} -\frac{1}{n} \ln P_n(c). \qquad (3)$$

We refer to den Hollander [64] for a more mathematical and rigorous definition of the rate function based on probability measures and the large deviation principle.

The rate function is positive and is equal to 0 for ergodic Markov chains only for the typical value of C_n corresponding to the ergodic expectation

$$c^* = \sum_{i\in\Gamma} p_i^* f(i). \qquad (4)$$

Indeed, by the ergodic theorem, C_n converges in probability to c^* as $n \to \infty$, so that $P_n(c)$ concentrates around c^* in that limit. The scaling (2) extends this result by showing that $P_n(c)$ concentrates *exponentially* around c^*, meaning that fluctuations away from the typical value c^* are exponentially unlikely with n.

The rate function can be obtained using different methods from large deviation theory [63]. The most common proceeds by considering the scaled cumulant generating function (SCGF), defined by the limit

$$\Psi(s) = \lim_{n\to\infty} \frac{1}{n} \ln E[e^{snC_n}], \qquad (5)$$

where $s \in \mathbb{R}$ and $E[\cdot]$ denotes the expectation. For ergodic Markov chains, the SCGF is known [63] to be given by the logarithm of the dominant eigenvalue of a transformation of the transition matrix Π, referred to as the *tilted matrix*, defined by

$$\tilde{\Pi}_s(i, j) = \Pi(i, j)e^{sf(i)}. \qquad (6)$$

As a result, we have

$$\Psi(s) = \ln \zeta_s, \qquad (7)$$

where ζ_s denotes the unique (Perron-Frobenius) dominant eigenvalue of $\tilde{\Pi}_s$, which is a positive matrix. From this point, one then uses the Gärtner-Ellis theorem [63] to obtain $I(c)$ as the Legendre transform of $\Psi(s)$, given by

$$I(c) = s_c c - \Psi(s_c), \qquad (8)$$

where s_c the unique solution of

$$c = \Psi'(s). \qquad (9)$$

This has an obvious similarity with the Legendre transform of thermodynamics connecting the entropy and free energy. For this reason it is common to interpret the parameter s of the SCGF as a Legendre parameter or "temperature" conjugated or dual to C_n [21].

C. Effective Markov chain

The result above shows that the problem of finding the rate function reduces to solving a spectral problem, namely,

$$\tilde{\Pi}_s r_s = \zeta_s r_s, \qquad (10)$$

for the dominant eigenvalue ζ_s and its corresponding eigenvector r_s with components $r_s(i)$, $i \in \Gamma$. The knowledge of this eigenvector is also important in large deviation theory, as it provides a way to understand how a given fluctuation $C_n = c$ away from c^* is created by a *modified Markov chain* with transition matrix

$$\Pi_s(i, j) = \frac{\tilde{\Pi}_s(i, j) r_s(j)}{\zeta_s r_s(i)} = \frac{e^{sf(i)} \Pi(i, j) r_s(j)}{\zeta_s r_s(i)}. \qquad (11)$$

Since $r_s > 0$ and $\zeta_s > 0$, Π_s is a well-defined nonnegative matrix, which is also stochastic and ergodic [65]. As a result, it defines another ergodic Markov chain on Γ, which is known to be equivalent to the original Markov chain conditioned on the event $\{C_n = c\}$ in the long-time limit where $n \to \infty$, provided that s is chosen according to the duality relation (9) [61,66,67]. Hence, the modified Markov chain can be interpreted as an effective Markov model that describes the subset of paths of the original Markov chain that lead to the fluctuation $C_n = c$.

From a simulation point of view, the Markov chain described by Π_s can also be seen an exponential tilting of the original Markov chain, which can be used to efficiently sample the event $\{C_n = c\}$ using importance sampling [29–31]. This follows because the typical value of C_n in the modified Markov chain is $c_s = \Psi'(s)$, so choosing s according to (9) gives us $c_s = c$ [67]. In other words, what is a rare event for the original Markov chain is transformed into a typical event for the modified Markov chain.

This can be applied to estimate the SCGF. The exponential expectation $E[e^{snC_n}]$ entering in the definition of the SCGF is dominated by rare fluctuations of C_n centered around c_s [21]. Sampling those fluctuations with the original Markov chain is inefficient when $s \neq 0$, whereas sampling them with the modified process is efficient [67]. This is important for understanding APM.

III. ADAPTIVE POWER METHOD

The dominant eigenvalue ζ_s and its corresponding eigenvector r_s can be obtained numerically by diagonalizing $\tilde{\Pi}_s$, but this method is not efficient for large systems, as it generally requires $O(|\Gamma|^3)$ steps for a system of size $|\Gamma|$. Since we need only the dominant eigenvalue, we can use instead the power method, which proceeds by choosing an initial (positive) vector $r^{(1)}$ and by calculating successive approximations of r_s in a recursive way using the (column) matrix product

$$r^{(m+1)} = \tilde{\Pi}_s r^{(m)}. \qquad (12)$$

From this update, the dominant eigenvalue is then approximated as

$$\zeta^{(m+1)} = \frac{r^{(m+1)}(i)}{r^{(m)}(i)}, \qquad (13)$$

where i is any component in Γ.

Provided that the norm of $r^{(m)}$ is kept constant at every iteration by including a normalization step, as explained in

detail below, we have $r^{(m)} \to r_s$ up to an arbitrary multiplicative constant and $\zeta^{(m)} \to \zeta_s$ in the limit where $m \to \infty$. In particular, we can divide $r^{(m)}$ by its maximum component, located say at i_0, before the next matrix product to obtain

$$\zeta^{(m+1)} = r^{(m+1)}(i_0). \qquad (14)$$

This corresponds to choosing the infinity norm for the normalization.

The complexity of the power method is $O(|\Gamma|^2)$, since it is based on repeated matrix products, which is better than diagonalization but still inefficient for large systems. The idea of APM is to apply the power method in a statistical way by observing that the action of $\tilde{\Pi}_s$ is a conditional expectation over one step of the Markov chain:

$$(\tilde{\Pi}_s r)(i) = \sum_{j \in \Gamma} e^{sf(i)} \Pi(i, j) r(j)$$
$$= e^{sf(i)} E[r(X_{\ell+1}) | X_\ell = i]. \qquad (15)$$

This suggests estimating the expectation using Monte Carlo simulations by drawing samples of the Markov chain. In this way, we avoid calculating the matrix product over all the states, focusing instead on those that are actually visited.

This approach for calculating the SCGF was proposed by the group of Borkar [49–51], who suggested two different simulation schemes for estimating the expectation (15): a concurrent or *synchronous* scheme in which many copies of the Markov chain are simulated starting from the state i, and an *asynchronous* scheme in which the expectation is estimated as an ergodic average using a single trajectory of the Markov chain. The latter scheme appears to us simpler and is therefore the one that we consider here. It is summarized in Algorithm 1 and involves three steps explained in the next subsections.

A. Importance sampling

The estimation of the expectation in (15) using samples of the Markov chain $(X_\ell)_{\ell=1}^n$ becomes inefficient as more and more steps of the power method are taken, since this method tries to estimate the exponential expectation in the definition of the SCGF [52], which cannot be estimated efficiently, as mentioned, from trajectories of the original Markov chain. Instead, we must generate samples from the modified Markov chain, so as to write the same expectation with importance sampling as

$$(\tilde{\Pi}_s r)(i) = \sum_{j \in \Gamma} e^{sf(i)} \Pi_s(i, j) \frac{\Pi(i, j)}{\Pi_s(i, j)} r(j)$$
$$= e^{sf(i)} E[r(\hat{X}_{\ell+1}) R_s(\hat{X}_\ell, \hat{X}_{\ell+1}) | \hat{X}_\ell = i], \quad (16)$$

where $\hat{X}_\ell$ denotes the modified Markov chain with transition matrix Π_s given by (11) and

$$R_s(i, j) = \frac{\Pi(i, j)}{\Pi_s(i, j)} \qquad (17)$$

is the likelihood factor that corrects for the fact that the expectation involves $\hat{X}_\ell$ rather than X_ℓ.

In the algorithm, states of the modified Markov chain are generated from the current estimates $\zeta^{(\ell)}$ and $r^{(\ell)}$ of the dominant spectral elements at time ℓ [68], resulting in the following

estimate of the modified transition matrix:

$$\Pi_s^{(\ell)}(i, j) = \frac{e^{sf(i)}\Pi(i, j)r^{(\ell)}(j)}{\zeta^{(\ell)}r^{(\ell)}(i)\eta^{(\ell)}(i)}, \tag{18}$$

which includes an additional normalization factor $\eta^{(\ell)}(i)$, compared to (11), because $r^{(\ell)}$ is not exactly the dominant eigenvector of $\tilde{\Pi}_s$. As a result, we must introduce this factor so that

$$\sum_{j \in \Gamma} \Pi_s^{(\ell)}(i, j) = 1 \tag{19}$$

for all $i \in \Gamma$. It is easy to check that imposing this normalization is equivalent to rewriting $\Pi_s^{(\ell)}$ as

$$\Pi_s^{(\ell)}(i, j) = \frac{\Pi(i, j)r^{(\ell)}(j)}{\gamma^{(\ell)}(i)}, \tag{20}$$

where

$$\gamma^{(\ell)}(i) = \sum_{j \in \Gamma} \Pi(i, j)r^{(\ell)}(j) \tag{21}$$

is a different normalization factor.

B. Stochastic approximation

The expectation in (16) is not calculated as a matrix product, as mentioned before, but evaluated pointwise on realizations of the modified Markov chain. Considering, the power method update in (12) with the result (20) means that the eigenvector is updated stochastically as

$$r^{(\ell+1)}(i) = \frac{e^{sf(i)}\gamma^{(\ell)}(i)}{\zeta^{(\ell)}} \tag{22}$$

at the realization i of X_ℓ at time ℓ.

C. Annealing

Applying (22) recursively on random samples of the modified Markov chain will not result in a convergent estimation of r_s or ζ_s, since $\gamma^{(\ell)}$ always changes randomly as new states are visited. To filter this "noise," we can use an annealing scheme whereby the update of r_s is progressively reinforced according to

$$r^{(\ell+1)}(i) = (1 - a_\ell)r^{(\ell)}(i) + a_\ell \frac{e^{sf(i)}\gamma^{(\ell)}(i)}{\zeta^{(\ell)}} \tag{23}$$

using a sequence $(a_\ell)_{\ell \geqslant 1}$, called the *learning sequence*, that vanishes as $\ell \to \infty$ in such a way that

$$\sum_{\ell=1}^{\infty} a_\ell = \infty, \qquad \sum_{\ell=1}^{\infty} a_\ell^2 < \infty. \tag{24}$$

These two conditions are known to be sufficient for the stochastic approximation to converge [69], and are meant intuitively to balance the need between exploration and exploitation. In practice, other decreasing sequences can be used that satisfy

$$a_\ell = \ell^{-\alpha} \tag{25}$$

with $\alpha > 0$. We call the exponent α the *learning rate*.

In the end, the steps combine to work as detailed in Algorithm 1: From the state x_ℓ reached at the ℓth iteration, we

---

Algorithm 1 Adaptive power method (APM)

Data: Transition matrix Π, initial state x_1, Lagrange parameter s, number of time steps n, learning sequence a_ℓ.
Initialization: $r^{(1)} = 1$, $\zeta^{(1)} = 1$, $\Pi_s^{(1)} = \Pi$, $\gamma^{(1)} = 1$
for $\ell = 1, \dots, n-1$ **do**
  Draw $x_{\ell+1}$ from x_ℓ with probability $\Pi_s^{(\ell)}(x_\ell, \cdot)$
  Update $r^{(\ell)}$ to $r^{(\ell+1)}$ at $i = x_\ell$ using (23)
  Locate max of eigenvector: $i_0 \leftarrow \arg\max_k r^{(\ell+1)}(k)$
  Update eigenvalue: $\zeta^{(\ell+1)} \leftarrow r^{(\ell+1)}(i_0)$
  Compute normalizing factor $\gamma^{(\ell+1)}$ at $i = x_\ell$ using (21)
  Update $\Pi_s^{(\ell+1)}$ at $i = x_\ell$ and $j = x_{\ell+1}$ using (20)
return: $\hat{\Psi}_n(s) = \ln \zeta^{(n)}$.

---

generate a new random state $x_{\ell+1}$ with probability $\Pi_s^{(\ell)}(x_\ell, \cdot)$, where $\Pi_s^{(\ell)}$ is the current estimate of the modified transition matrix given by (18) or (20). From these two states, we then update the eigenvector at the location $i = x_\ell$ according to (23) and the modified Markov matrix at the locations $i = x_\ell$ and $j = x_{\ell+1}$ according to (20), having computed the normalizing factor in (21). This can be done in $O(|\Gamma|)$ steps, since only one entry of $\Pi_s^{(\ell)}$ is modified, so only one of its rows needs to be normalized. The complexity of one step of the APM is thus reduced compared to the power method from $O(|\Gamma|^2)$ to $O(|\Gamma|)$.

The result of the algorithm is an eigenvalue estimate $\hat{\Psi}_n(s)$ of the SCGF $\Psi(s)$ for one value of s, which can be repeated so as to obtain an interpolation of this function over a range of values. Alternatively, the SCGF can be estimated using the time-additive estimator

$$\bar{\Psi}_n(s) = sC_{n,s} - K_{n,s}, \tag{26}$$

where $C_{n,s}$ is the observable value obtained by running the algorithm for the parameter s, which converges in probability to $c_s = \Psi'(s)$ as $n \to \infty$, and

$$K_{n,s} = -\frac{1}{n}\sum_{\ell=1}^{n-1} \ln R_s(x_\ell, x_{\ell+1}) \tag{27}$$

is an additive cost associated with the likelihood ratio, which is also estimated using the trajectory of the modified Markov chain generated by the algorithm. The form of $\bar{\Psi}_n(s)$ follows from a control interpretation of large deviation functions for Markov processes [67] and is advantageous for computing error bars.

To obtain the rate function, we can in principle take the Legendre transform of either estimator of the SCGF. However, although $\hat{\Psi}(s)$ is convex by definition [21], neither $\hat{\Psi}_n(s)$ nor $\bar{\Psi}_n(s)$ is found in simulations to be convex because of statistical errors (see Fig. 4), so taking their Legendre transform would yield spurious results. To avoid this problem, which is not specific to our model [52], it is best to estimate $I(a)$ using the additive cost $K_{n,s}$, which converges in probability to $I(c_s)$ as $n \to \infty$ [67]. By computing the pair $(C_{n,s}, K_{n,s})$ for varying s values, we then obtain an approximation of the rate function in parametric form as $(c_s, I(c_s))$ [52]. An advantage of this method is that, as for the SCGF, error bars can be computed directly.

IV. APPLICATION

We investigate in this section the performance of the APM algorithm for a system showing a DPT, specifically, a simple random walk evolving on an ER random graph, for which the observable is taken to be the mean of the degrees of the nodes visited by the random walk [53–55]. We define this model next and review recent results about its large deviations and the existence of a first-order DPT in the thermodynamic limit. We then reproduce these results using APM, testing different annealing schedule and the effect of transfer learning.

A. Model

We consider a random walk evolving on an undirected graph G with N vertices and M edges or links, determined by the adjacency matrix

$$A_{ij} = \begin{cases} 1 & i, j \text{ are connected} \\ 0 & \text{otherwise.} \end{cases} \tag{28}$$

The graph is chosen randomly from the ER random ensemble of graphs in which edges are chosen in a binomial way over the $N(N-1)/2$ possible edges with probability $p = \kappa/N$, where $\kappa > 1$ [70]. The random walk itself is determined by the transition probabilities

$$\Pi(i, j) = \frac{A_{ij}}{k_i}, \tag{29}$$

where k_i is the degree of node i. This defines an unbiased random walk (URW) over the graph, assigning equal probability for transitions to happen from one node i to its neighbours $j \in \partial i$, which is known to be ergodic.

Many dynamical observables can defined for the URW. The one that we consider is the mean degree

$$C_n = \frac{1}{n} \sum_{\ell=1}^{n} k_{X_\ell}, \tag{30}$$

visited by a trajectory of the URW over a time n. The large deviations of this observable have been studied recently [53–55] and are interesting in that they show very different mechanisms creating large fluctuations of C_n above the expected value $c^* = \kappa + 1$ and small fluctuations of C_n below c^*. Large degree fluctuations, on the one hand, come from rare trajectories of the URW that localize around a hub of highly connected nodes [see Fig. 1(a)], resulting in an effective Markov chain for $s > 0$ that concentrates on this hub as $s \to \infty$. On the other hand, small degree fluctuations come from rare trajectories that localize on low-degree nodes and, for the smallest degree fluctuations, on "dangling chains" or "hairs" consisting of two nodes of degrees 2 and 1, respectively, resulting in $c_{\min} = 1.5$ (see Fig. 1 and Fig. 2 in [55]).

These two fluctuations regimes are very different because large ER graphs typically have a single hub but many dangling chains. Moreover, dangling chains are not connected to each other, but are connected instead via high-degree nodes around the hub, so the small degree fluctuation regime, related to $s < 0$, must come with a symmetry-breaking mechanism, akin to a phase transition, whereby the URW localizes on a particular dangling chain (out of many equivalent chains),
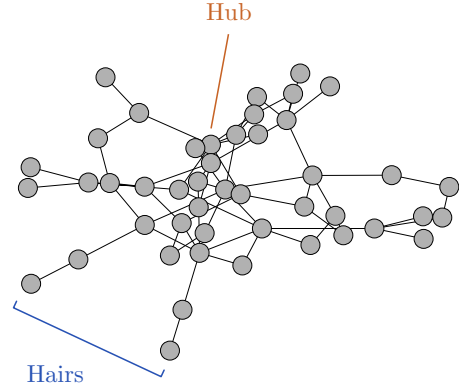

FIG. 1. ER graph of size $N = 46$ obtained with $\kappa = 3$ showing a hub node and two dangling chains or hairs.

which depends on the initial value of the random walk and the graph itself [55].

This transition is especially visible when the connectivity parameter κ is small and translates mathematically into a sharp corner of $\Psi(s)$ near $s = 0$ related to a near-linear branch of $I(c)$ left of the expected value c^*, corresponding to the zero of $I(c)$ (see Fig. 4 and Fig. 1 in [55]). These features of $\Psi(s)$ and $I(c)$ get more pronounced as the graph size N increases, suggesting the existence of a DPT in the fluctuations of C_n, which should be first-order given that $\Psi'(s)$ develops a jump singularity. Whether this DPT is a "true" phase transition in the thermodynamic limit where $N \to \infty$ is still unresolved. The numerical evidence published so far suggest that it is, but there is no rigorous proof yet that $\Psi(s)$ has a singularity in the thermodynamic limit [71].

The exact nature of the transition is not crucial for us; what is interesting is that a sharp transition or crossover is visible already for $N = 50$, so one does not need to consider extremely large system sizes to study its effect on numerical computations of large deviation functions. For this reason, the URW on ER graphs is a good benchmark for testing the efficiency of numerical methods near DPTs and for comparing such methods. It is also a simple benchmark, as it does not involve many interacting particles. For an application of the APM to larger graphs, showing the efficiency of the algorithm on larger systems, we refer to the recent study of Di Bona *et al.* [72].

B. Convergence of APM

We show in Fig. 2 the evolution of the SCGF returned by APM over time (i.e., iterations) for two values of s, namely, $s = -1$ and $s = 1$, and different graph sizes. As can be seen, the convergence of $\hat{\Psi}_n(s)$ towards the exact value $\Psi(s)$, calculated by exact diagonalization [73], is slow because the APM needs to learn by single-step exploration where the corresponding rare event takes place in the graph, leading to a long "exploration" phase starting from the initial position x_1, which is chosen to be a random node on the graph. The case $s = 1$, for example, corresponds to a larger-than-average degree fluctuation, associated with paths of the random walk concentrating on the high-degree nodes, which need to be found and explored sufficiently in time before the eigenvalue
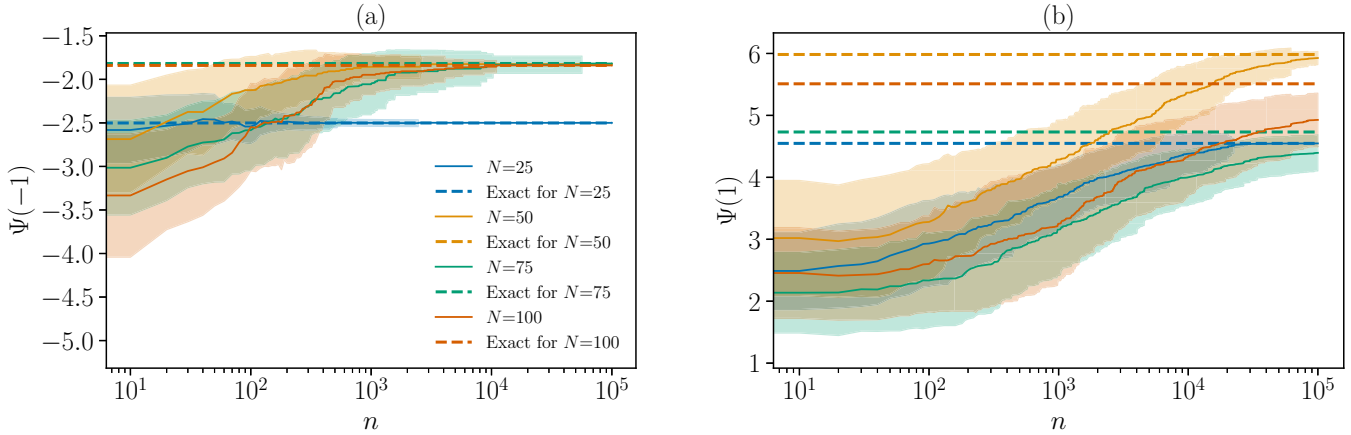
FIG. 2. Convergence of APM starting from random initial states for (a) $s = -1$ and (b) $s = 1$. The colored lines show the average of the eigenvalue estimate $\hat{\Psi}_n(s)$ calculated over 100 simulations for different graph sizes (see legend) while the colored regions show the standard deviation. Learning rate: $\alpha = 0.1$. The graph used is the one shown in Fig. 1.

can converge. This takes longer the bigger the graph is, as the random walk needs to reach the hub of the graph starting from random nodes, which are further from the hub, on average, the bigger the graph is. For $s = -1$, on the other hand, the random walk needs to reach hairs of the graph, a faster process, as shown in Fig. 2(a), as the number of hairs grows with the system size.

In practice, we should take advantage of the exploration phase, since it visits different regions of the graphs, associated with different degree fluctuations and, therefore, different s values. Thus, instead of trying to reach the correct eigenvalue for a given s, we should start the algorithm at a small value of s close to $s = 0$, let it run for this value, which is only a small perturbation of the result $\zeta_0 = 1$, and repeat the process by incrementing or "quenching" s in small steps of Δs [48,52], reinitializing the annealing sequence at each step. The result with $\Delta s = 0.25$ is shown in Fig. 3 again for $s = -1$ and $s = 1$, two graph sizes, and different number of iterations for each quench level. As can be seen, the APM now converges quickly for each value of s considered, leading to a much smaller

error region (shaded region) compared to the results of Fig. 2, especially for $s = 1$. Moreover, the convergence now does not depend significantly on the system's size, especially again for $s = 1$, which shows quenched plateaux being reached more or less at the same times for the two sizes considered ($N = 50$ and 100). This arises because the simulation for a given s "locally" informs the next simulation for $s + \Delta s$ by transferring the state of the Markov chain and the computed eigenvector from one simulation to another.

We call this transfer of information between different runs of the algorithm "transfer learning" in analogy with a similar method used in machine learning for transferring parameters from one learned model to another. An obvious advantage of this method is that we are able to efficiently obtain the SCGF for a range of s values by using a small step Δs, as shown in Figs. 4(a)–4(c), which was obtained for the graph shown in Fig. 1 using $\Delta s = 0.02$. In this figure, we also show the effect of using different iterations for each run, steps or "quenches" of s, as well as different learning rates α. The results show overall that convergence with small error bars
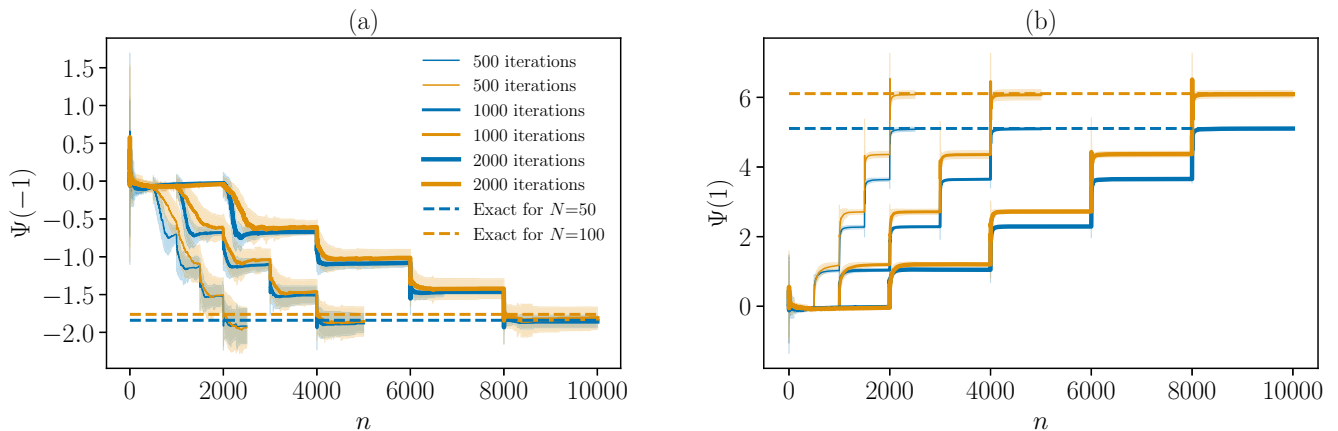

FIG. 3. Convergence of APM with transfer learning for (a) $s = -1$ and (b) $s = 1$. The colored lines show the average of $\hat{\Psi}_n(s)$ calculated over 100 simulations for two different graph sizes (see legend) while the colored regions show the standard deviation. Four steps were used to reach the final SCGF by changing s in steps of $\Delta s = 0.25$. Learning rate: $\alpha = 0.1$.
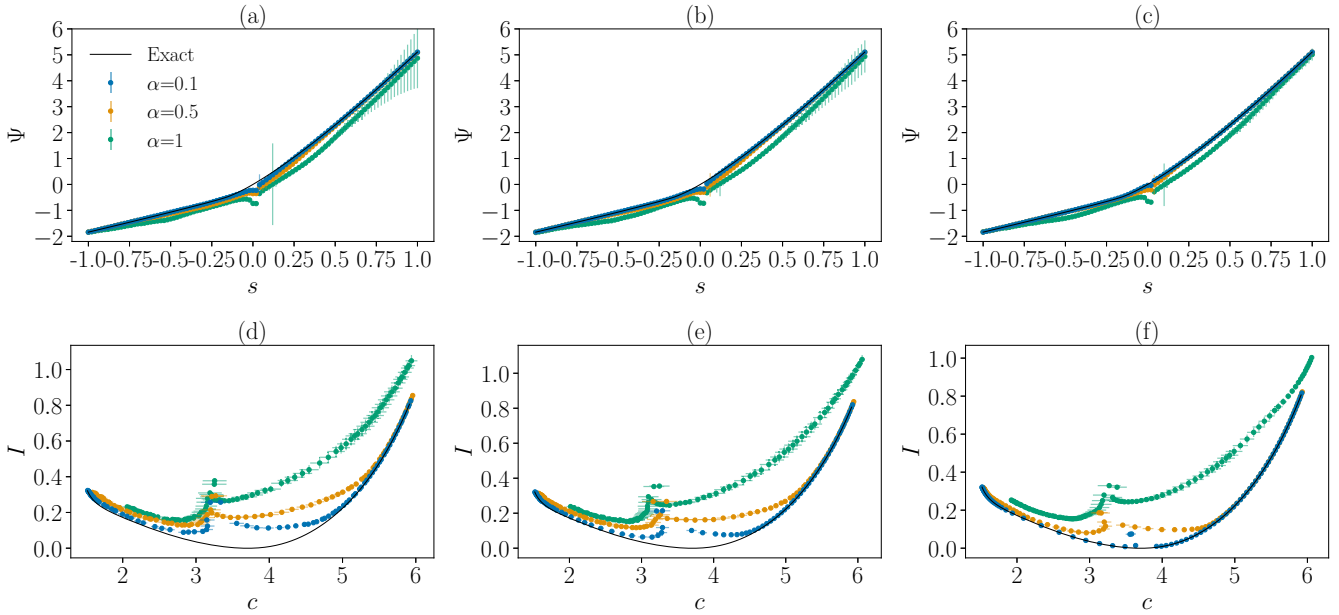
FIG. 4. Estimation of the SCGF and rate function with transfer learning for the graph shown in Fig. 1. Top row: Estimated SCGF from the dominant eigenvalue for (a) 500, (b) 1000, and (c) 10 000 iterations per s value, i.e., per transfer learning step. Botton row: Estimated rate function for (d) 500, (e) 1000, and (f) 10 000 steps per s value.

is achieved with 1000 iterations, and lower when considering the tails of the SCGF. What is more important for convergence is the learning rate, which needs to be low enough so that the learning is annealed in a slow way, favoring exploration. By trying different values, we have found that $\alpha = 0.1$ gives good results for the graph sizes considered. Above this value, the APM is annealed too quickly, leading to a negative bias in the SGCF, especially near the DPT around $s = 0$, as shown in Fig. 4.

This bias is seen more clearly at the level of the rate function, shown in the lower plots of Fig. 4. There we see that the APM gets trapped if annealed too quickly in different phases of the DPT, corresponding to different regions of the graphs that either have a low or a high connectivity, resulting in a positive bias in the estimated rate function.

The bias is reminiscent of hysteresis effects seen in Monte Carlo simulations of equilibrium systems having first-order phase transitions and can be alleviated by performing longer simulations and, in our case, by using a low learning rate. Convergence around the DPT can also be improved by using a dynamical variant of the replica exchange method [74], whereby two simulations with different s values are randomly exchanged, as proposed and tested recently [48]. We have not used this method for the APM, but we think that it might be useful when dealing with very large systems showing DPTs.

It is important to note again that $I(c)$ is estimated from the additive cost $K_{n,s}$ shown in (27) and not from the Legendre transform of $\hat{\Psi}_n(s)$, since the latter is not necessarily convex, as seen from Fig. 4. The SCGF itself can also be estimated as a time average, as noted before, using the estimator $\bar{\Psi}_n(s)$ defined in (26). This method in fact converges faster and is more stable, as shown in Fig. 5, and is therefore our preferred method for computing the SCGF. Using the dominant eigenvalue gives good results, but the additive

estimator gives better results for the same number of iterations and learning rate simply because the latter is a self-averaging quantity.

To obtain an idea of the error associated with the additive estimator, and therefore of APM in general, we have computed $\bar{\Psi}_n(s)$ for $s = -1$ using four levels or quenches separated by $\Delta s = 0.25$ and have computed the relative error ϵ between the final value of the estimator and the exact SCGF obtained by direct diagonalization. The results, also computed for $s = 1$, are shown in Fig. 6 as a function of the number n of iterations per quench and for different system sizes. As expected, the relative error decreases with n in a way
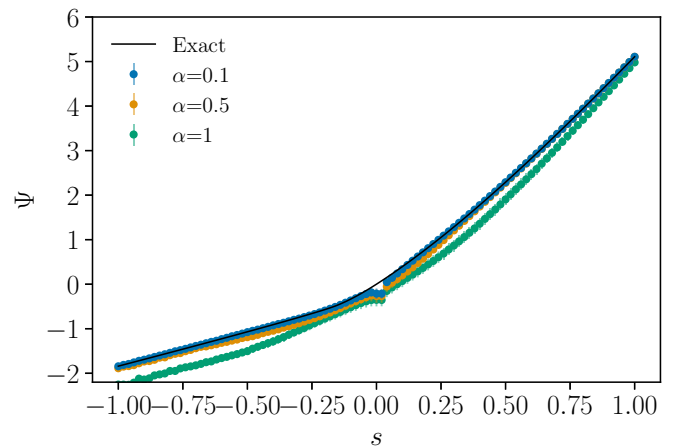

FIG. 5. Estimated SCGF with transfer learning using the additive estimator $\bar{\Psi}_n(s)$, defined in Eq. (26), instead of the eigenvalue estimator $\hat{\Psi}_n(s)$. Parameters: 1000 iterations per transfer plateau spaced with $\Delta s = 0.02$.
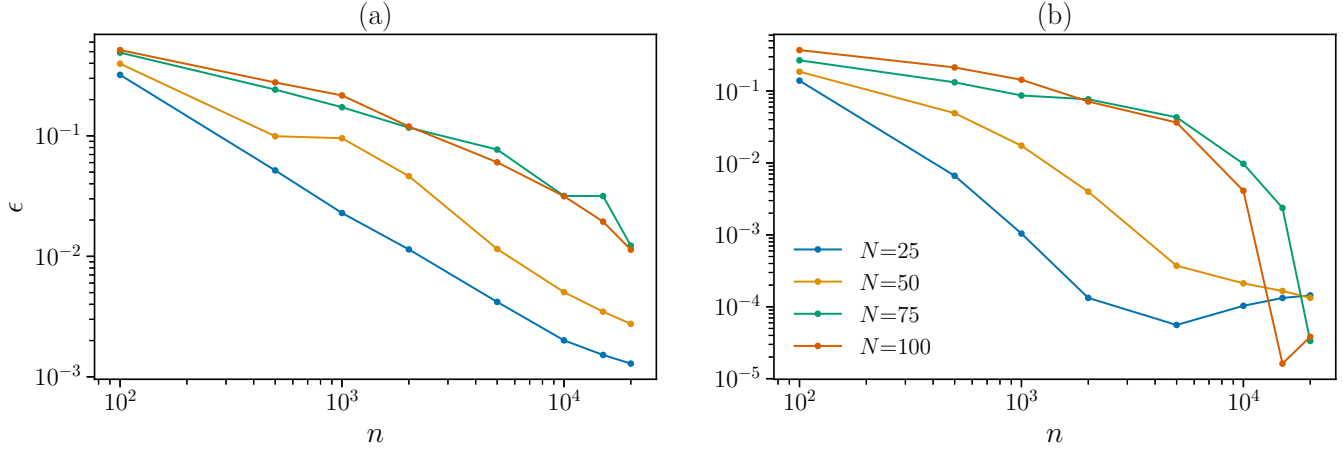
FIG. 6. Relative error of $\bar{\Psi}_n(s)$ for (a) $s = -1$ and (b) $s = 1$ as a function of the number of iterations per step of the transfer learning. Parameters: $\alpha = 0.1$ and $\kappa = 3$.

that appears roughly to be polynomial with n for all system sizes, demonstrating the efficiency of APM coming from the importance sampling step, which has the effect of biasing the random walk in the relevant fluctuation region regardless of system size. From the plot, we can also see that the error is smaller for $s = 1$, confirming the smaller error bars shown in Fig. 3 for this value of s.

An error analysis can also be performed, in principle, for the eigenvalue estimator $\hat{\Psi}_n(s)$ of the SCGF, although in this case one must decide whether to compare the final value estimated or average over an entire simulation. We do not show results for this estimator, as they are less stable compared with the additive estimator $\bar{\Psi}_n(s)$.

The error bars shown in all the plots discussed so far were obtained by repeating runs of the APM algorithm, but it is important to note that error bars could have been obtained for the additive estimators from single runs using batch mean methods [29]. This, in fact, is a further advantage of using additive estimators for the SCGF and the rate function. Defining error bars for the eigenvalue estimator based on single runs is more difficult, since this estimator is not a time average, but an annealed version of the eigenvalue returned by APM. As far as we know, no theoretical work has been done on deriving a central limit theorem for the eigenvalue estimator, which would serve as the basis for defining error bars for this estimator. Our advice, based on the results that we have obtained, is to use additive estimators whenever possible. They converge quicker and have better variance properties compared to the eigenvalue estimator.

C. Effective Markov chain

The APM yields not only the SCGF and rate function, but also the effective process associated with the set of trajectories creating a given fluctuation $C_n = c$ in the long-time limit. This process was discussed extensively in an earlier study [55]. In Fig. 7 we show the stationary distribution p_s of this process obtained by running the APM for some time after it has found the SCGF. The results are plotted, together with the dominating eigenvector r_s, for four s values located around the DPT and compared with the theoretical results obtained again by direct diagonalization. Note that, for convenience, we

plot r_s and p_s as a function of the node degree k rather than as a function of the node state, although both are of course functions of $i \in \Gamma$ [75]. Moreover, we normalized r_s in such a way that $\max_k r_s(k) = 1$.

The plots show that APM is able to recover r_s and p_s with good accuracy. We recall that r_s is used in the importance sampling step to bias the random walk in the region where a given degree fluctuation is likely to take place, while p_s shows where that region is. In the case $s = -1$, for example, which corresponds to small degree fluctuations, the plot of p_s shows that the effective random walk concentrates on small degree nodes (*viz.*, on hairs), whereas for $s = 1$ the random walk concentrates on higher degree nodes located around the hub of the graph. We refer again to [55] for a detailed discussion of this concentration and its relation to the DPT.

What is important to note in relation to the APM is that, for the purpose of biasing the process in the importance sampling step, one does not need to estimate the dominant eigenvector $r_s(i)$ with high precision for all states i of the state space Γ, but only for those states that are most often visited, as determined by p_s. This arises because the additive estimators of the SCGF and the rate function are in fact estimators of expectations with respect to p_s [67]. As a result, we do not need in practice to represent r_s over the whole of Γ, but can truncate that representation to states that have been visited a certain number of times, in analogy with real-space renormalization methods [76]. This can be done dynamically within simulations to save memory (as done, e.g., in reinforcement learning algorithms [77]), opening the possibility of applying APM to very large systems.

V. CONCLUDING REMARKS

We have tested in this paper the APM algorithm on a simple random walk to show the efficacy of this algorithm for computing large deviation functions characterizing the fluctuations of time-additive functions of Markov processes. Compared to previous studies on this algorithm [49–52], we have studied the effect of the learning rate on the convergence of the algorithm and demonstrated that the algorithm is not slowed down around DPTs when a relatively small
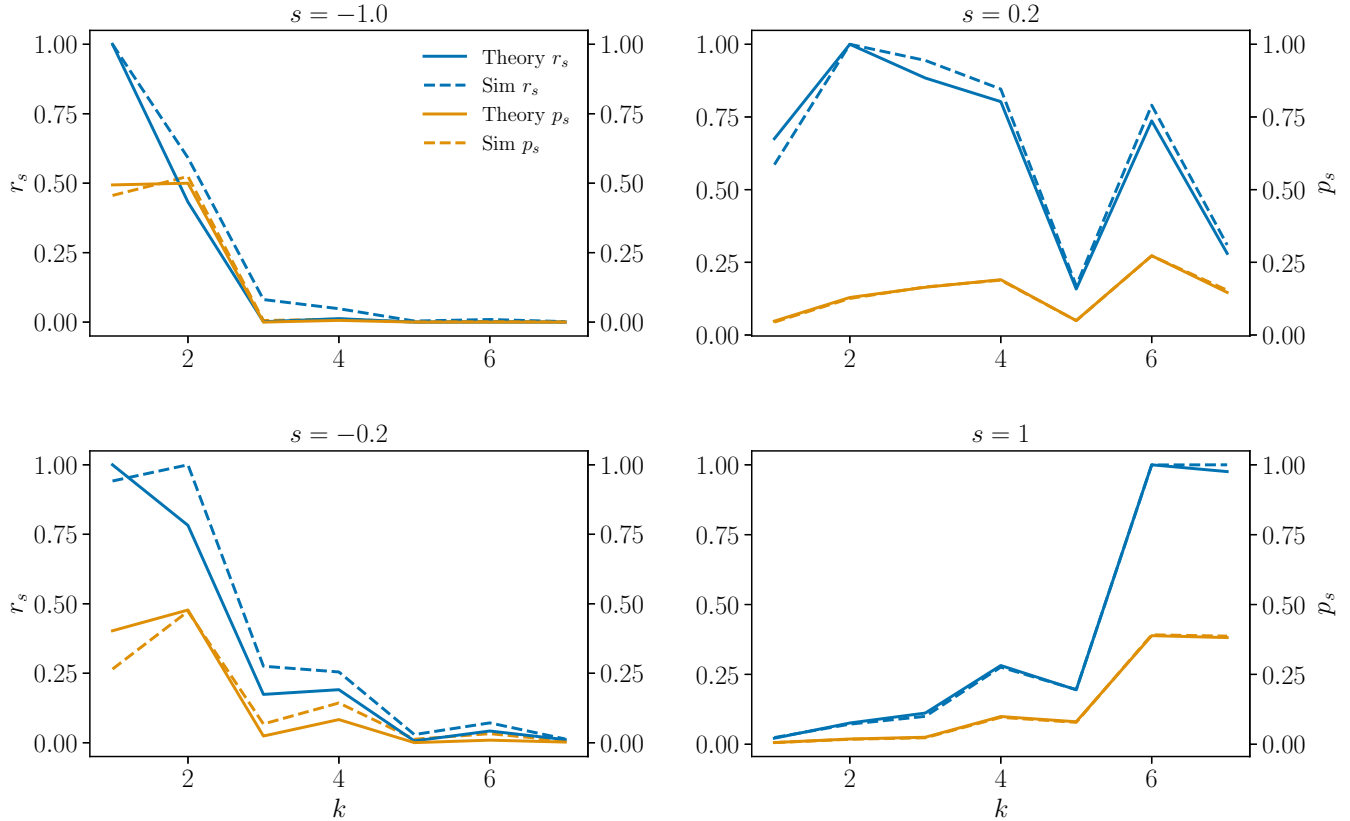
FIG. 7. Dominant eigenvector r_s and stationary distribution p_s of the effective Markov chain plotted as a function of node degree. The numerical result, averaged over 100 simulations, are compared for different s with theoretical results obtained by direct diagonalization. Learning rate: $\alpha = 0.1$. The graph used is the one shown in Fig. 1.

learning rate is used. In practice, different learning rates (starting from large to small), initial conditions, and simulation times should be used to test whether the APM converges in a robust way. One advantage of estimating the rate function using the cost estimator is that the value obtained is an upper bound on the true rate function (within statistical errors), so reducing that cost with different parameters necessarily leads to an improvement in the estimated function, as illustrated in Fig. 4.

Following [52], we have also shown that the SCGF and rate function are most efficiently computed by performing long simulations in which the large deviation parameter s is increased in small steps, using the knowledge of the computed eigenvalue and eigenvector at each step to guide with importance sampling the random walk in the fluctuation region of interest. The use of importance sampling is critical for the algorithm to work—indeed, for any large deviation simulation method to work—and leads the APM to gradually learn the effective process, which is known to be efficient for sampling large deviations and rare events in general [29–31].

The effective process can be estimated or learned in other ways, for instance, using splitting or cloning algorithms [34], stochastic optimization [48], or reinforcement learning [45–47]. Compared to these approaches, APM has some advantages:

(1) The algorithm is guaranteed to converge to the correct SCGF [49], and so has no bias, essentially because the

feedback mechanism that updates the importance sampling is based on the power method. Other methods based on variational representations of the SCGF (see, e.g., [48]) are generally found to converge, but it is often difficult to guarantee that the solution they return is the optimal solution (global minimum) corresponding to the SCGF.

(2) The annealing of the stochastic approximation of the eigenvector has the effect of stabilizing the adaptive importance sampling, as it gradually reinforces previous updates and filters out the randomness or noise inherent in the new generated states. Importance sampling can also be included and adapted in cloning but the feedback mechanism that has been proposed so far [34] is not annealed and has not been proved to converge.

(3) Since the estimators of the SCGF and rate function are time-additive quantities, just like the observable itself, their statistical errors can be computed directly in single runs of the algorithm using standard batch mean methods [29]. The same applies to stochastic optimization and reinforcement learning approaches, but not to cloning, which is not based on additive estimators.

More work is needed to compare these methods and the many more that have been proposed recently for computing large deviation functions and, moreover, to develop good benchmarks for comparing different methods. In a sense, all methods should have the same complexity, if implemented efficiently, since they are aimed at solving the

same problem—that of finding a dominant eigenvalue and its eigenvector [67]. However, some methods might be easier to implement, depending on the process and observable considered. Moreover, some methods might be advantageous in terms of stability, for proving convergence, or for computing errors, as discussed here.

[1] B. Derrida, Non-equilibrium steady states: Fluctuations and large deviations of the density and of the current, J. Stat. Mech. (2007) P07023.

[2] L. Bertini, A. D. Sole, D. Gabrielli, G. Jona-Lasinio, and C. Landim, Stochastic interacting particle systems out of equilibrium, J. Stat. Mech. (2007) P07014.

[3] L. Bertini, A. De Sole, D. Gabrielli, G. Jona-Lasinio, and C. Landim, Macroscopic fluctuation theory, Rev. Mod. Phys. **87**, 593 (2015).

[4] J. P. Garrahan, R. L. Jack, V. Lecomte, E. Pitard, K. van Duijvendijk, and F. van Wijland, Dynamical First-Order Phase Transition in Kinetically Constrained Models of Glasses, Phys. Rev. Lett. **98**, 195702 (2007).

[5] L. O. Hedges, R. L. Jack, J. P. Garrahan, and D. Chandler, Dynamic order-disorder in atomistic models of structural glass formers, Science **323**, 1309 (2009).

[6] J. P. Garrahan, R. L. Jack, V. Lecomte, E. Pitard, K. van Duijvendijk, and F. van Wijland, First-order dynamical phase transition in models of glasses: An approach based on ensembles of histories, J. Phys. A: Math. Theor. **42**, 075007 (2009).

[7] J. P. Garrahan and I. Lesanovsky, Thermodynamics of Quantum Jump Trajectories, Phys. Rev. Lett. **104**, 160601 (2010).

[8] C. P. Espigares, P. L. Garrido, and P. I. Hurtado, Dynamical phase transition for current statistics in a simple driven diffusive system, Phys. Rev. E **87**, 032115 (2013).

[9] G. Bunin, Y. Kafri, and D. Podolsky, Cusp singularities in boundary-driven diffusive systems, J. Stat. Phys. **152**, 112 (2013).

[10] P. Tsobgni Nyawo and H. Touchette, A minimal model of dynamical phase transition, Europhys. Lett. **116**, 50009 (2016).

[11] A. Lazarescu, Generic dynamical phase transition in one-dimensional bulk-driven lattice gases with exclusion, J. Phys. A: Math. Theor. **50**, 254004 (2017).

[12] G. Gallavotti and E. G. D. Cohen, Dynamical Ensembles in Nonequilibrium Statistical Mechanics, Phys. Rev. Lett. **74**, 2694 (1995).

[13] J. Kurchan, Fluctuation theorem for stochastic dynamics, J. Phys. A: Math. Gen. **31**, 3719 (1998).

[14] J. L. Lebowitz and H. Spohn, A Gallavotti-Cohen-type symmetry in the large deviation functional for stochastic dynamics, J. Stat. Phys. **95**, 333 (1999).

[15] R. J. Harris and G. M. Schütz, Fluctuation theorems for stochastic dynamics, J. Stat. Mech. (2007) P07020.

[16] A. C. Barato and U. Seifert, Thermodynamic Uncertainty Relation for Biomolecular Processes, Phys. Rev. Lett. **114**, 158101 (2015).

[17] P. Pietzonka, A. C. Barato, and U. Seifert, Universal bounds on current fluctuations, Phys. Rev. E **93**, 052145 (2016).

[18] T. R. Gingrich, J. M. Horowitz, N. Perunov, and J. L. England, Dissipation Bounds All Steady-State Current Fluctuations, Phys. Rev. Lett. **116**, 120601 (2016).

[19] T. R. Gingrich, G. M. Rotskoff, and J. M. Horowitz, Inferring dissipation from current fluctuations, J. Phys. A: Math. Theor. **50**, 184004 (2017).

[20] J. Li, J. M. Horowitz, T. R. Gingrich, and N. Fakhri, Quantifying dissipation using fluctuating currents, Nat. Commun. **10**, 1666 (2019).

[21] H. Touchette, The large deviation approach to statistical mechanics, Phys. Rep. **478**, 1 (2009).

[22] H. Touchette, Introduction to dynamical large deviations of Markov processes, Physica A **504**, 5 (2018).

[23] U. Ray, G. K.-L. Chan, and D. T. Limmer, Exact Fluctuations of Nonequilibrium Steady States from Approximate Auxiliary Dynamics, Phys. Rev. Lett. **120**, 210602 (2018).

[24] 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).

[25] C. Casert, T. Vieijra, S. Whitelam, and I. Tamblyn, Dynamical Large Deviations of Two-Dimensional Kinetically Constrained Models Using a Neural-Network State Ansatz, Phys. Rev. Lett. **127**, 120602 (2021).

[26] L. Causer, M. C. Bañuls, and J. P. Garrahan, Optimal sampling of dynamical large deviations via matrix product states, Phys. Rev. E **103**, 062144 (2021).

[27] A. Das and D. T. Limmer, Variational design principles for nonequilibrium colloidal assembly, J. Chem. Phys. **154**, 014107 (2021).

[28] S. Juneja and P. Shahabuddin, Rare-event simulation techniques: An introduction and recent advances, in *Handbooks in Operations Research and Management Science*, edited by S. G. Henderson and B. L. Nelson (Springer, Amsterdam, 2006), Vol. 13, pp. 291–350.

[29] S. Asmussen and P. W. Glynn, *Stochastic Simulation: Algorithms and Analysis*, Stochastic Modelling and Applied Probability (Springer, New York, 2007).

[30] J. A. Bucklew, *Introduction to Rare Event Simulation* (Springer, New York, 2004).

[31] A. Guyader and H. Touchette, Efficient large deviation estimation based on importance sampling, J. Stat. Phys. **181**, 551 (2020).

[32] C. Giardina, J. Kurchan, and L. Peliti, Direct Evaluation of Large-Deviation Functions, Phys. Rev. Lett. **96**, 120603 (2006).

[33] V. Lecomte and J. Tailleur, A numerical approach to large deviations in continuous time, J. Stat. Mech. (2007) P03004.

[34] T. Nemoto, F. Bouchet, R. L. Jack, and V. Lecomte, Population-dynamics method with a multicanonical feedback control, Phys. Rev. E **93**, 062123 (2016).

[35] C. Perez-Espigares and P. I. Hurtado, Sampling rare events across dynamical phase transitions, Chaos **29**, 083106 (2019).

[36] L. Angeli, S. Grosskinsky, A. M. Johansen, and A. Pizzoferrato, Rare event simulation for stochastic dynamics in continuous time, J. Stat. Phys. **176**, 1185 (2019).

[37] F. Cérou and A. Guyader, Adaptive multilevel splitting for rare event analysis, Stoch. Anal. Appl. **25**, 417 (2007).

[38] T. Dean and P. Dupuis, Splitting for rare event simulation: A large deviation approach to design and analysis, Stoch. Proc. Appl. **119**, 562 (2009).

[39] F. Cérou, A. Guyader, and M. Rousset, Adaptive multilevel splitting: Historical perspective and recent results, Chaos **29**, 043108 (2019).

[40] C.-E. Bréhier and T. Lelièvre, On a new class of score functions to estimate tail probabilities of some stochastic processes with adaptive multilevel splitting, Chaos **29**, 033126 (2019).

[41] A. Kundu, S. Sabhapandit, and A. Dhar, Application of importance sampling to the computation of large deviations in nonequilibrium processes, Phys. Rev. E **83**, 031119 (2011).

[42] U. Ray, G. K.-L. Chan, and D. T. Limmer, Importance sampling large deviations in nonequilibrium steady states. I, J. Chem. Phys. **148**, 124120 (2018).

[43] K. Klymko, P. L. Geissler, J. P. Garrahan, and S. Whitelam, Rare behavior of growth processes via umbrella sampling of trajectories, Phys. Rev. E **97**, 032123 (2018).

[44] U. Ray and G. K.-L. Chan, Constructing auxiliary dynamics for nonequilibrium stationary states by variance minimization, J. Chem. Phys. **152**, 104107 (2020).

[45] T. H. E. Oakes, A. Moss, and J. P. Garrahan, A deep learning functional estimator of optimal dynamics for sampling large deviations, Mach. Learn. Sci. Tech. **1**, 035004 (2020).

[46] D. C. Rose, J. F. Mair, and J. P. Garrahan, A reinforcement learning approach to rare trajectory sampling, New J. Phys. **23**, 013013 (2021).

[47] A. Das, D. C. Rose, J. P. Garrahan, and D. T. Limmer, Reinforcement learning of rare diffusive dynamics, J. Chem. Phys. **155**, 134105 (2021).

[48] J. Yan, H. Touchette, and G. M. Rotskoff, Learning nonequilibrium control forces to characterize dynamical phase transitions, Phys. Rev. E **105**, 024115 (2022).

[49] V. S. Borkar, S. Juneja, and A. A. Kherani, Performance analysis conditioned on rare events: An adaptive simulation scheme, Commun. Info. Syst. **3**, 259 (2003).

[50] T. P. I. Ahamed, V. S. Borkar, and S. Juneja, Adaptive importance sampling technique for Markov chains using stochastic approximation, Oper. Res. **54**, 489 (2006).

[51] A. Basu, T. Bhattacharyya, and V. S. Borkar, A learning algorithm for risk-sensitive cost, Math. Oper. Res. **33**, 880 (2008).

[52] G. Ferré and H. Touchette, Adaptive sampling of large deviations, J. Stat. Phys. **172**, 1525 (2018).

[53] C. D. Bacco, Decentralized network control, optimization and random walks on networks, Ph.D. thesis, Université Paris Sud (2015).

[54] C. D. Bacco, A. Guggiola, R. Kühn, and P. Paga, Rare events statistics of random walks on networks: Localization and other dynamical phase transitions, J. Phys. A: Math. Theor. **49**, 184003 (2016).

[55] F. Coghi, J. Morand, and H. Touchette, Large deviations of random walks on random graphs, Phys. Rev. E **99**, 022137 (2019).

[56] C. W. Gardiner, *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, 2nd ed., Springer Series in Synergetics, Vol. 13 (Springer, New York, 1985).

[57] G. Grimmett and D. Stirzaker, *Probability and Random Processes* (Oxford University Press, Oxford, 2001).

[58] K. Jacobs, *Stochastic Processes for Physicists: Understanding Noisy Systems* (Cambridge University Press, Cambridge, 2010).

[59] We use the notation $\Pi(i, j)$ instead of the more usual Π_{ij} to denote the (i, j) entry of Π because other subscripts will be used later.

[60] U. Seifert, Stochastic thermodynamics: From principles to the cost of precision, Physica A **504**, 176 (2018).

[61] R. Chetrite and H. Touchette, Nonequilibrium Markov processes conditioned on large deviations, Ann. Henri Poincaré **16**, 2005 (2015).

[62] A. Shwartz and A. Weiss, *Large Deviations for Performance Analysis*, Stochastic Modeling Series (Chapman and Hall, London, 1995).

[63] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*, 2nd ed. (Springer, New York, 1998).

[64] F. den Hollander, *Large Deviations*, Fields Institute Monograph (AMS, Providence, RI, 2000).

[65] H. D. Miller, A convexity property in the theory of random variables defined on a finite Markov chain, Ann. Math. Stat. **32**, 1260 (1961).

[66] R. Chetrite and H. Touchette, Nonequilibrium Microcanonical and Canonical Ensembles and Their Equivalence, Phys. Rev. Lett. **111**, 120601 (2013).

[67] R. Chetrite and H. Touchette, Variational and optimal control representations of conditioned and driven processes, J. Stat. Mech. (2015) P12001.

[68] Spectral elements are updated at the same time as states of the modified Markov chain are generated, so that $m = \ell$.

[69] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint* (Cambridge University Press, Cambridge, 2008).

[70] This scaling of the link probability p defines the sparse regime of the ER ensemble, in which most graphs have a connected component of size proportional to N and have a degree distribution that becomes Poissonian with expectation κ in the "thermodynamic limit" $N \to \infty$.

[71] For a proof of the existence of DPTs in other types of graphs, see the recent work of Carugno, Vivo, and Coghi [78].

[72] G. Di Bona, L. Di Gaetano, V. Latora, and F. Coghi, Maximal dispersion of adaptive random walks, Phys. Rev. Res. **4**, L042051 (2022).

[73] The large deviation functions have not been obtained analytically for this system [55], so we rely on exact diagonalization results for comparison.

[74] R. H. Swendsen and J.-S. Wang, Replica Monte Carlo Simulation of Spin-Glasses, Phys. Rev. Lett. **57**, 2607 (1986).

[75] The mapping from component to degree is done simply by adding the components of the eigenvector with the same degree.

[76] 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(R) (2009).

[77] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. (MIT Press, Cambridge, MA, 2018).

[78] G. Carugno, P. Vivo, and F. Coghi, Delocalization-localization dynamical phase transition of random walks on graphs, Phys. Rev. E **107**, 024126 (2023).