The Journal of Chemical Physics

Reinforcement learning of rare diffusive dynamics

Cite as: J. Chem. Phys. **155**, 134105 (2021); https://doi.org/10.1063/5.0057323
Submitted: 19 May 2021 • Accepted: 12 September 2021 • Published Online: 01 October 2021

Avishek Das, Dominic C. Rose, Juan P. Garrahan, et al.

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

Reinforcement learning of rare diffusive dynamics

Cite as: J. Chem. Phys. 155, 134105 (2021); doi: 10.1063/5.0057323
Submitted: 19 May 2021 • Accepted: 12 September 2021 •
Published Online: 1 October 2021

Avishek Das,[1,a] Dominic C. Rose,[2,3,b] Juan P. Garrahan,[2,3,c] and David T. Limmer[1,4,5,6,d]

AFFILIATIONS

[1] Department of Chemistry, University of California, Berkeley, California 94609, USA
[2] School of Physics and Astronomy, University of Nottingham, Nottingham NG7 2RD, United Kingdom
[3] Centre for the Mathematics and Theoretical Physics of Quantum Non-Equilibrium Systems, University of Nottingham, Nottingham NG7 2RD, United Kingdom
[4] Kavli Energy NanoScience Institute, Berkeley, California 94609, USA
[5] Materials Science Division, Lawrence Berkeley National Laboratory, Berkeley, California 94609, USA
[6] Chemical Science Division, Lawrence Berkeley National Laboratory, Berkeley, California 94609, USA

[a] **Electronic mail:** avishek_das@berkeley.edu
[b] **Electronic mail:** dom.rose@ucl.ac.uk
[c] **Electronic mail:** juan.garrahan@nottingham.ac.uk
[d] **Author to whom correspondence should be addressed:** dlimmer@berkeley.edu

ABSTRACT

We present a method to probe rare molecular dynamics trajectories directly using reinforcement learning. We consider trajectories that are conditioned to transition between regions of configuration space in finite time, such as those relevant in the study of reactive events, and trajectories exhibiting rare fluctuations of time-integrated quantities in the long time limit, such as those relevant in the calculation of large deviation functions. In both cases, reinforcement learning techniques are used to optimize an added force that minimizes the Kullback–Leibler divergence between the conditioned trajectory ensemble and a driven one. Under the optimized added force, the system evolves the rare fluctuation as a typical one, affording a variational estimate of its likelihood in the original trajectory ensemble. Low variance gradients employing value functions are proposed to increase the convergence of the optimal force. The method we develop employing these gradients leads to efficient and accurate estimates of both the optimal force and the likelihood of the rare event for a variety of model systems.

I. INTRODUCTION

Rare but important events play a significant role in phenomena occurring throughout the sciences, ranging from physics[1] and chemistry[2] to climate science[3] and economics.[4] As a consequence, methods developed to study rare events can transcend disciplines. In molecular systems, rare events determine the rates by which chemical reactions occur and phases interconvert,[5] and they also encode the response of systems driven to flow or unfold.[6–10] Strategies that afford a means of studying rare dynamical events in statistically unbiased ways are particularly desired in order to deduce the intrinsic pathways by which they occur and to evaluate their likelihoods. Borrowing notions from reinforcement learning,[11] we have developed a method to generate rare dynamical trajectories directly through the optimization of an auxiliary dynamics that generates an ensemble of trajectories with the correct relative statistical weights. Within this ensemble of trajectories, a variational estimate of the likelihood of the rare event is obtainable from a simple expectation value.

Much research has been devoted to the enhanced sampling of molecular dynamics simulations, yet there remain active areas of open research. Methods for sampling dynamical fluctuations, especially those away from equilibrium, are considerably less developed than their equilibrium and configurational counterparts.[12,13] Recent work has sought to construct methods for finding an effective auxiliary dynamics,[14–20] with the goal of sampling rare dynamical fluctuations with the corresponding correct statistical weights directly, by evolving simulations with additional parameterized forces. Such methods are often designed to approximate the so-called Doob transform,[21–24] which is the unique force that evolves a trajectory conditioned on a rare event.

A general approach to the optimization of a sampling dynamics based on a variational principle for the Doob transform for diffusive processes has recently been developed.[25] Within this context

of diffusive processes, optimal forces have been used to elucidate mechanisms and rates of nonlinear response,[26,27] to encode dynamical phase diagrams,[28–30] and to deduce inverse design principles.[31,32] In this work, we aim to extend a reinforcement learning[11] based approach to the optimization of a sampling dynamics to diffusive systems, building on the work of Refs. 25 and 33 and past literature on reinforcement learning for continuous time processes.[34–41]

The techniques of reinforcement learning aim at learning the best decisions to make in each state in order to achieve some goal. Algorithms developed in this context have led to many significant advancements in recent years across tasks requiring an intelligent agent to interact with an environment, such as in gameplay[42–44] and robotics,[45–47] with a variety of recent applications in physics.[48–55] However, many of these situations are framed as discrete time problems, with relatively little work done in stochastic continuous time control.[34,35] For diffusive processes and importance sampling molecular dynamics, we formulate a reinforcement learning procedure to learn the correct force to influence the probability of choosing each next state. From this perspective, we take a policy gradient based approach,[35,45,46,56,57] learning a generative model for the evolution of the state. The optimized force found is such that rare events are made typical while staying close to the original force, providing a dynamics that can aid in efficiently sampling the targeted trajectory ensemble.

A key advantage of the reinforcement learning techniques we develop is the use of an additional learning process for a function that guides the optimization of the dynamics, a so-called value function,[58] which describes how relevant each state is to the rare events of interest. This value function substantially reduces the variance in estimates of the gradient of the parameters specifying a force, allowing for the use of less data in each optimization step and subsequently more complex approximations to the auxiliary dynamics. We show how this approach can be successfully applied to both finite time problems in which the dynamics is constrained to guarantee the occurrence of some rare transition like a barrier crossing and to time-homogeneous problems where we are interested in the statistics of time-integrated observables in the long time limit as characterized by its large deviation function.

II. TRAJECTORY ENSEMBLE FORMALISM

We consider systems evolving with a diffusive dynamics over time t of a configuration $\mathbf{x}$. These configurations evolve according to a force vector $\mathbf{F}(\mathbf{x},t)$ and noise vector of equal dimension $\mathbf{W}$ with associated constant noise matrix $\mathbb{G}$ invertible within the stochastically evolving subspace, represented by the following stochastic differential equation (SDE):

$$d\mathbf{x} = \mathbf{F}(\mathbf{x},t)dt + \mathbb{G}\cdot d\mathbf{W}, \tag{1}$$

where the noise $\mathbf{W}$ follows a Wiener process, with increments $d\mathbf{W}$ drawn from a Gaussian with zero mean and dt variance. Throughout we will work in dimensionless variables that imply unit energy scales and mobilities. The requirement of $\mathbb{G}$ being invertible within the stochastic subspace may, in principle, be relaxed; however, in that case, there may be multiple noise vectors corresponding to the same change of state, making the evaluation of transition probabilities necessary for our optimization approach difficult. We will follow the Ito convention for ease of notation and implementation with standard numerical integrators. Throughout, we do not assume in Eq. (1) that the force is gradient or that the noise obeys a detailed balance, and thus, our approach is generally applicable to equilibrium and nonequilibrium dynamics.

We aim to probe rare fluctuations in trajectory observables. Here, we consider trajectories, $\mathbf{X}_{0,T}$, defined as the sequence of configurations over an observation time T, although generalizations of fluctuating observation times are possible.[59] Generally, we will consider observables that are functions of time-integrated variables over the trajectory,

$$O[\mathbf{X}_{0,T}] = \int_0^T dt\, A[\mathbf{x}_t,t] + \mathbf{B}[\mathbf{x}_t,t]\cdot\dot{\mathbf{x}}(t), \tag{2}$$

where the first term is a state dependent observable, while the second term depends on a stochastic increment, with both $A[\mathbf{x}_t,t]$ and $\mathbf{B}[\mathbf{x}_t,t]$ being state dependent. However, we will also consider cases in which $A[\mathbf{x}_t,t]$ is a function of a single time in order to impose end point conditioning. Expectations of functions of such observables are defined through path integrals of the form

$$\langle f(O[\mathbf{X}_{t,t'}])\rangle_p = \int D\mathbf{X}_{t,t'}d\mathbf{x}_t\, P[\mathbf{X}_{t,t'}]f(O[\mathbf{X}_{t,t'}]), \tag{3}$$

where $P[\mathbf{X}_{t,t'}]$ is the total probability of a trajectory decomposable into $P[\mathbf{X}_{t,t'}] = p[\mathbf{X}_{t,t'}|\mathbf{x}_t]\rho(\mathbf{x}_t)$, where $p[\mathbf{X}_{t,t'}|\mathbf{x}_t]$ is the transition probability conditioned on starting in configuration $\mathbf{x}_t$ with initial probability $\rho(\mathbf{x}_t)$.

Probabilities for trajectories between times t and t' starting at $\mathbf{x}_t$ are defined by

$$p[\mathbf{X}_{t,t'}|\mathbf{x}_t] \propto \exp\left\{-\frac{1}{2}\int_t^{t'} dt''\left|\mathbb{G}^{-1}\cdot(\dot{\mathbf{x}}-\mathbf{F})\right|^2\right\} \tag{4}$$

where we suppressed the arguments of $\mathbf{x}_t$ and $\mathbf{F}[\mathbf{x}_t,t]$ for shorthand. This is the standard Onsager–Machlop form for the diffusive dynamics considered here.[60] The measure over paths between times t and t' starting from position $\mathbf{x}_t$ is defined such that

$$\int D\mathbf{X}_{t,t'}p[\mathbf{X}_{t,t'}|\mathbf{x}_t] = 1, \tag{5}$$

where the transition probability is normalized when integrated over all trajectories. These path probabilities satisfy

$$p[\mathbf{X}_{t,t''}|\mathbf{x}_t] = p[\mathbf{X}_{t',t''}|\mathbf{x}_{t'}]p[\mathbf{X}_{t,t'}|\mathbf{x}_t] \tag{6}$$

and

$$D\mathbf{X}_{t,t''} = D\mathbf{X}_{t',t''}D\mathbf{X}_{t,t'} \tag{7}$$

due to the Markovian noise in Eq. (1).

Trajectories sampled with $P[\mathbf{X}_{0,T}]$ will be dominated by the most typical values of $O[\mathbf{X}_{0,T}]$. We will encode the rare trajectories with atypical values of $O[\mathbf{X}_{0,T}]$ by reweighting the original trajectory ensemble defined by Eq. (4), multiplying each trajectory by an observable dependent factor. Such reweightings occur naturally in statistical studies of rare events and are isomorphic to extended ensemble approaches in equilibrium configurational problems. The

ensemble of events we are interested in is constructed by weighting the probability of trajectories in the original dynamics by an exponentially positive number,

$$P_s[\mathbf{X}_{0,T}] = e^{-sO[\mathbf{X}_{0,T}]-\lambda(s,T)}P[\mathbf{X}_{0,T}], \tag{8}$$

where $P_s[\mathbf{X}_{0,T}]$ is denoted as a tilted path ensemble, biased by a statistical field s in such a way to promote rare fluctuations in $O[\mathbf{X}_{0,T}]$. The quantity $\lambda(s,T)$ normalizes the tilted distribution, and it is identifiable as a cumulant generating function (CGF),

$$\lambda(s,T) = \ln Z(s,T) = \ln\left\langle e^{-sO[\mathbf{X}_{0,T}]}\right\rangle_p, \tag{9}$$

and it is equal to the logarithm of the tilted path partition function $Z(s,T)$. The reweighted path ensemble generally defines a new transition probability $p_s[\mathbf{X}_{t,t'}|\mathbf{x}_t]$ and initial condition. The evaluation of $\lambda(s,T)$ is a common objective in studies of diffusive systems as it describes the statistics of $O[\mathbf{X}_{0,T}]$. Contributions to $\lambda(s,T)$ or $P_s[\mathbf{X}_{0,T}]$ are dominated by trajectories with large or small values of $O[\mathbf{X}_{0,T}]$ depending on the sign of s. The exponential bias, $\exp(-sO[\mathbf{X}_{0,T}])$, can also be constructed to function as a filter based on fulfilling specific criteria. In such cases, $P_s[\mathbf{X}_{0,T}]$ is identified as the probability that a trajectory fulfills a specific conditioning and its ensemble fulfills a corresponding conditioned path ensemble. Common examples are Brownian bridges,[61–63] where trajectories are conditioned to end at $\mathbf{x}_T = \mathbf{x}'$, in which $O[\mathbf{X}_{0,T}]$ is 1 if $\mathbf{x}_T = \mathbf{x}'$ and is 0 otherwise, and s is taken sufficiently negative that only those trajectories for which the constraint is satisfied have significant weight.

III. GRADIENT OPTIMIZATION FOR FINITE TIME CONSTRAINED DYNAMICS

Our aim is to find a dynamics that generates trajectories with probability as close to the reweighted trajectories ensemble as possible. For the diffusive dynamics considered here, this is exactly achievable, in principle, through the so-called generalized Doob transformation.[21,22,64–67] The generalized Doob transformation defines a modified dynamics with an added drift force that is generally time-dependent but with an identical noise as in the original SDE. However, constructing this transformation is often not possible, in practice, as it requires diagonalizing a modified Fokker–Planck operator, which in interacting systems is exponentially complex.[24] Here, we aim to parameterize a drift force with tunable parameters θ to approximate the generalized Doob transform. With the modified force, $\mathbf{F}_\theta(\mathbf{x},t)$, we have a modified SDE

$$d\mathbf{x} = \mathbf{F}_\theta(\mathbf{x},t)dt + \mathbb{G}\,d\mathbf{W}, \tag{10}$$

with corresponding trajectory probabilities

$$p_\theta[\mathbf{X}_{t,t'}|\mathbf{x}_t] \propto \exp\left\{-\frac{1}{2}\int_t^{t'} dt''\left|\mathbb{G}^{-1}\cdot(\dot{\mathbf{x}}-\mathbf{F}_\theta)\right|^2\right\}, \tag{11}$$

which still satisfy the Markovian properties of the original dynamics and the same normalization constant. See Ref. 33 for a discussion of problems in which the optimal dynamics is required to be non-Markovian in the context of discrete time Markov processes.

We seek to learn a set of parameters θ to minimize the Kullback–Leibler (KL) divergence between the modified dynamics and the reweighted trajectory ensemble defined by Eq. (8). The KL divergence is defined as

$$D_{\mathrm{KL}}(p_\theta|p_s) = \left\langle \ln\left(\frac{p_\theta[\mathbf{X}_{0,T}|\mathbf{x}_0]\rho(\mathbf{x}_0)}{p_s[\mathbf{X}_{0,T}|\mathbf{x}_0]\rho(\mathbf{x}_0)}\right)\right\rangle_{p_\theta}, \tag{12}$$

where the expectation is taken with respect to the parameterized dynamics. This quantity is a measure of the similarity between the modified and reweighted trajectory ensembles. Achieving a zero value when p_θ is given by the generalized Doob transform, the KL divergence has a unique minimum when this Doob transformed dynamics is contained within the space of parameterized dynamics, providing a variational estimate of the CGF. We note that this definition of the KL divergence differs from much of the literature, considering optimization of a parameterized diffusive dynamics,[17,68–70] where the parameterized dynamics p_θ and target dynamics p_s appear in an opposite way. In principle, the initial distribution should also be parameterized as it will be modified by the reweighting; however, depending on the space of distributions chosen, these can be hard to sample. We drop this modification for simplicity.

A. Low variance gradient estimation

In order to optimize the force, $\mathbf{F}_\theta$, we follow techniques introduced in the reinforcement learning literature.[11,45,71–74] Substituting the parameterized and reweighted trajectory probabilities into the KL divergence, we may rewrite it as an average over a parameter dependent time-integrated observable

$$D_{\mathrm{KL}}(p_\theta|p_s) = -\langle R[\mathbf{X}_{0,T}]\rangle_{p_\theta} + \lambda(s,T), \tag{13}$$

where in the language of reinforcement learning, we define a return, $R[\mathbf{X}_{0,T}]$, as

$$R[\mathbf{X}_{0,T}] = -sO[\mathbf{X}_{0,T}] - \ln\left(\frac{p_\theta[\mathbf{X}_{0,T}|\mathbf{x}_0]}{p[\mathbf{X}_{0,T}|\mathbf{x}_0]}\right), \tag{14}$$

with the negative of the average of the second term measuring the KL divergence, $D_{\mathrm{KL}}(p_\theta|p)$, between the parameterized dynamics and the original dynamics. This return is analogous to a regularized form of reinforcement learning[72,74] similar to that considered in maximum-entropy reinforcement learning.[45,46,73] When evaluated at the generalized Doob transform, the KL divergence vanishes and the return evaluates to the CGF. Away from the Doob transform, the positivity of the KL divergence results in the return, variationally bounding the CGF from below.[23]

We aim to minimize the KL divergence through stochastic gradient descent in the parameter space. For this, we need the gradient of $D_{\mathrm{KL}}(p_\theta|p_s)$ with respect to θ,

$$\nabla_\theta D_{\mathrm{KL}}(p_\theta|p_s) = -\langle R[\mathbf{X}_{0,T}]\nabla_\theta \ln p_\theta[\mathbf{X}_{0,T}|\mathbf{x}_0]\rangle_{p_\theta}, \tag{15}$$

where we note that

$$\langle \nabla_\theta R[\mathbf{X}_{0,T}]\rangle_{p_\theta} = 0 \tag{16}$$

due to conservation of probability.[33] The factor multiplying the return is commonly referred to as the Malliavin weight in the stochastic analysis literature[75] and corresponds to a particular case of the eligibility traces found in reinforcement learning,[11,58,76–78] which we denote as $y_\theta(T) = \nabla_\theta \ln p_\theta[\mathbf{X}_{0,T}|\mathbf{x}_0]$. It can be rewritten by substituting the path probability,

$$y_\theta(t'') - y_\theta(t') = \int_{t'}^{t''} dt\, \dot{y}_\theta(t), \tag{17}$$

where

$$\dot{y}_\theta(t) = \left[\mathbb{G}^{-1}\cdot(\dot{\mathbf{x}}(t) - \mathbf{F}_\theta(t))\right]\cdot\left[\mathbb{G}^{-1}\cdot\nabla_\theta\mathbf{F}_\theta(t)\right] \tag{18}$$

is the integrand of the Malliavin weight.

Were we to stop at Eq. (15), we would proceed to optimize a generative model (the diffusive dynamics with our parameterized force) of the trajectories using a score-function based approach, similar to standard unsupervised learning. However, following the methods of reinforcement learning, we can use a combination of the Markovianity of the generative model and other variance reduction techniques to produce a gradient estimator, which is much more efficient to estimate. To begin with, we can simplify Eq. (15) by noting that due to Markovianity, the Malliavin weight only correlates with the return in the future, and we can rewrite the gradient as

$$\begin{aligned}\nabla_\theta D_{\mathrm{KL}}(p_\theta|p_s) &= -\left\langle \int_0^T dt R[\mathbf{X}_{t^-,T}]\dot{y}_\theta(t)\right\rangle_{p_\theta}\\ &= \chi_{\mathrm{MCR}}(\theta,T),\end{aligned} \tag{19}$$

where we used t^- as a shorthand for $t-\epsilon$ for some small positive ϵ. We refer to the optimization of the modified dynamics using this formulation of the gradient as χ_{MCR}, as it is analogous to the Monte Carlo Returns (MCR) or the REINFORCE[79,80] policy gradient algorithm in reinforcement learning. In the long observation time limit, employing this gradient in stochastic optimization reduces to previous variational Monte Carlo procedures.[25]

This estimator of the gradient is non-optimal for two reasons. First, it requires evaluation of a two-time correlation function. In the steady state, stationarity can be invoked to eliminate one of those integrals; however, under finite time conditioning, this simplification is not possible. Second, it has a high variance and requires significant averaging to converge accurate gradients. This is because both the Malliavin weight and the return undergo a random walk with linearly increasing variance.[75] Building on the analogies with the reinforcement learning formalism, we define a value function as a path average of the return,

$$V(\mathbf{x},t) = \langle R[\mathbf{X}_{t,T}]\rangle_{p_\theta,\mathbf{x}}, \tag{20}$$

conditioned on starting at the position and time, $\mathbf{x}_t = \mathbf{x}$. Introduced into the gradients of $D_{\mathrm{KL}}(p_\theta|p_s)$ in distinct ways, the value functions can be used to tame both problems of the naive MCR gradient estimate.

First, we introduce a value function as a baseline that only depends on the state at the time t in order to reduce the variance of the gradient. We note that $\dot{y}_\theta(t)$ is linear in the noise and thus averages to zero when multiplied by a function of the state at or before t. Defining a temporal difference error

$$\delta[\mathbf{X}_{t^-,T},t] = R[\mathbf{X}_{t^-,T}] - V(\mathbf{x}_t,t), \tag{21}$$

we write the dynamical gradient as

$$\begin{aligned}\nabla_\theta D_{\mathrm{KL}}(p_\theta|p_s) &= -\left\langle \int_0^T dt\delta[\mathbf{X}_{t^-,T},t]\dot{y}_\theta(t)\right\rangle_{p_\theta}\\ &= \chi_{\mathrm{MCVB}}(\theta,T)\end{aligned} \tag{22}$$

where we have formally subtracted zero. We refer to this gradient estimator as χ_{MCVB} for Monte Carlo Value Baseline (MCVB).[11] The subtraction of the state point dependent value function reduces the variance of the gradient by accounting for the mean uncorrelated part of each return between t^- and T with $\dot{y}_\theta(t)$, focusing on how this return differs from the average behavior encoded by the value function.

Second, we introduce a value function that encodes an estimate of the return in the future in order to further reduce the variance and also the complications associated with estimating the two-time correlation function. We can replace part of the return by a value function that is conditioned at some τ such that $t^- < \tau < T$,

$$\langle R[\mathbf{X}_{t^-,T}]\dot{y}_\theta(t)\rangle = \langle V(\mathbf{x}_{t+\tau},t+\tau)\dot{y}_\theta(t)\rangle + \langle R[\mathbf{X}_{t^-,t+\tau}]\dot{y}_\theta(t)\rangle, \tag{23}$$

where we set the value function to zero for $V(\mathbf{x},t)$ with $t > T$. Combining this value function form of the kernel of the gradient with the value baseline, we define another temporal difference error

$$\delta'[\mathbf{X}_{t^-,t+\tau},t] = V(\mathbf{x}_{t+\tau},t+\tau) + R[\mathbf{X}_{t^-,t+\tau}] - V(\mathbf{x}_t,t), \tag{24}$$

and we arrive at a distinct formulation of the gradient

$$\begin{aligned}\nabla_\theta D_{\mathrm{KL}}(p_\theta|p_s) &= -\left\langle \int_0^T dt\,\delta'[\mathbf{X}_{t^-,t+\tau},t]\dot{y}_\theta(t)\right\rangle_{p_\theta}\\ &= \chi_{\mathrm{AC}}(\theta,T),\end{aligned} \tag{25}$$

which we denote as $\chi_{\mathrm{AC}}(\theta,T)$ for the actor–critic (AC) gradient estimator, for the analogous algorithm in reinforcement learning.[11,45] Here, the value function is seen as criticizing the transitions generated by the dynamics, i.e., the actor. Variance reduction in gradient estimates is, therefore, achieved by replacing potentially noisy return samples with the average behavior expected in the future of the $\mathbf{x}_{t+\tau}$ state. In Sec. IV, we will compare the accuracy and statistical efficiency of these three gradient estimators: MCR, MCVB, and AC. Before that, we discuss how the value functions are simultaneously parameterized and learnt alongside the modified force.

B. Parameterizing value functions

While the gradient expressions are exact and the use of value functions is expected to facilitate their convergence, using them requires the knowledge of the exact value function for the modified dynamics, a formidable task in complex problems. In order to make their use tractable, we optimize a representation of the value function in addition to the modified force. Specifically, we introduce a parameterization of the value function denoted as V_ψ. To optimize this approximation, we note that the value functions satisfy a self-consistency equation called the Bellman equation,[81]

$$V(\mathbf{x},t) = \langle V(\mathbf{x}_{t+\tau}, t+\tau) + R[\mathbf{X}_{t,t+\tau}]\rangle_{p_\theta,\mathbf{x}}, \tag{26}$$

which has a unique solution for a given dynamics and return [as defined by the tilting observable and the dynamics via Eq. (14)]. We aim to minimize the error in this equation, thus optimizing our parameterized value toward this unique solution. Our approach is to minimize the squared difference between the two sides of Eq. (26) with the true value function replaced by the parameterized value function and apply gradient descent to it. Such an approach is the subject of gradient temporal difference methods[82–84] but produces a gradient estimate, which is difficult to evaluate, containing products of expectations, which require independent samples. A part of the resultant gradient is, however, simpler to compute. We derive it by substituting only the right-hand side of Eq. (26) with our parameterized value function to provide a fixed target for the left and defining a corresponding error function based on the squared difference. To construct a loss, we integrate these errors along each trajectory and average them over the trajectory ensemble. This results in a loss function $L(\psi,\psi_i)$, which we take as a function of two weights, ψ and ψ_i,

$$L(\psi,\psi_i) = \frac{1}{2}\left\langle \int_0^T dt \Big\{ \langle V_{\psi_i}(\mathbf{x}_{t+\tau}, t+\tau) + R[\mathbf{X}_{t,t+\tau}]\rangle_{p_\theta,\mathbf{x}} - V_\psi(\mathbf{x}_t,t)\Big\}^2 \right\rangle_{p_\theta}, \tag{27}$$

where the weight ψ_i is the weights after update i, used to provide the fixed target estimate toward which we want to move the functional of ψ. The derivative is then taken with respect to ψ before setting $\psi = \psi_i$ to find the gradient of this loss for the current parameters. Such an approach is referred to as the semi-gradient in the reinforcement learning literature,[11] used to achieve the majority of the state-of-the-art reinforcement learning results, and proves to be stable, provided that the data used to estimate the gradient are sampled using a dynamics, which is close to p_θ as we intend to do. As mentioned above, alternative methods that additionally consider the variation of the target with ψ can be found in the reinforcement learning literature, allowing for the use of data sampled from an alternative dynamics, utilized via importance sampling.[82–84]

Writing an approximate temporal difference for the value function parameterization within MCVB,

$$\delta_\psi[\mathbf{X}_{t^-,T}, t] = R[\mathbf{X}_{t^-,T}] - V_\psi(\mathbf{x}_t, t), \tag{28}$$

or for AC,

$$\delta'_\psi[\mathbf{X}_{t^-,t+\tau}, t] = V_\psi(\mathbf{x}_{t+\tau}, t+\tau) + R[\mathbf{X}_{t^-,t+\tau}] - V_\psi(\mathbf{x}_t, t), \tag{29}$$

we have gradients of the form

$$\nabla_\psi L(\psi,\psi_i)\big|_{\psi=\psi_i} = -\left\langle \int_0^T dt\ \delta_{\psi_i}[\mathbf{X}_{t^-,T}, t]\, \nabla_\psi V_\psi(\mathbf{x}_t,t)\big|_{\psi=\psi_i} \right\rangle_{p_\theta} \tag{30}$$

for the loss function from the value function parameterization, where for the AC algorithm, δ_{ψ_i} is replaced with δ'_{ψ_i}. Given this value function approximation, we can approximate the gradient of the KL divergence by replacing the exact temporal difference with these approximate temporal differences. We then use the same trajectories to estimate the force and value function gradients and simultaneously learn both. For the MCVB algorithm, an approximate value function does not bias the gradients as the future return that correlates with the Malliavin weight stays intact and the expectation of the Malliavin weight is identically 0. However, for the AC algorithm, an approximate value function can introduce a bias into gradients as it replaces the average of the future return, which it may not accurately represent.

ALGORITHM 1. Gradient optimization using finite time trajectories.

inputs dynamical approximation $F_\theta(\mathbf{x},t)$, value approximation $V_\psi(\mathbf{x},t)$
parameters learning rates α^θ, α^ψ; total optimization steps I; trajectory length T consisting of J time steps of duration Δt each; number of trajectories N
initialize choose initial weights θ and ψ, define iteration variables i and j, force and value function gradients δ_P, δ_V, temporal difference δ (can be $R[\mathbf{X}_{t^-,T}]$ or $\delta_\psi[\mathbf{X}_{t^-,T}, t]$ or $\delta'_\psi[\mathbf{X}_{t^-,t+\tau}, t]$ for MCR/MCVB/AC)
$i \leftarrow 0$
repeat
  Using chosen method to generate trajectories $\mathbf{X}_{0,T}$ with configurations, times and temporal differences denoted by $\mathbf{x}_j$, t_j and δ_j, respectively.
  $j \leftarrow 0$
  $\delta_P \leftarrow 0$
  $\delta_V \leftarrow 0$
  repeat
    $\delta_P \leftarrow \delta_P + \delta_j \dot{y}_\theta(t_j)\Delta t$
    $\delta_V \leftarrow \delta_V + \delta_j \nabla_\psi V_\psi(\mathbf{x}_j, t_j)\Delta t$
    $j \leftarrow j + 1$
  until $j = J$
  average δ_P, δ_V over N trajectories to get $\bar{\delta}_P$, $\bar{\delta}_V$
  $\theta \leftarrow \theta + \alpha^\theta \bar{\delta}_P$
  $\psi \leftarrow \psi + \alpha^\psi \bar{\delta}_V$
  $i \leftarrow i + 1$
until $i = I$

Employing gradients with or without value functions, we can construct a stochastic descent algorithm to optimize the modified forces, which can be used to estimate the likelihoods of rare events and the trajectories by which they emerge. The algorithms require the evaluation of the forces, value function, their parametric gradients, and noises over the course of simulating trajectories. Ensembles of trajectories can then be used to construct an empirical estimate of the gradient via computing the Malliavin weights, returns, and the temporal difference. These empirical estimates then iterate the two weights with respective learning rates α^θ and α^ψ for the force and value function, respectively. The resultant algorithm is outlined in the pseudocode in Algorithm 1. Detailed versions of the individual algorithms with computationally efficient on-the-fly implementations for simulating trajectories with discrete time steps are presented in Appendix A.

IV. RARE FLUCTUATIONS IN FINITE TIME

We have used the algorithms discussed above to examine rare fluctuations of trajectories of fixed duration, starting from a fixed

point in configuration space. The specific observable we have investigated is an indicator function for reaching a desired region, Γ, in configuration space, $O[\mathbf{X}_{0,T}] = h_\Gamma[\mathbf{x}_T]$, where

$$h_\Gamma[\mathbf{x}_T] = \begin{cases} 1, & \mathbf{x}_T \in \Gamma, \\ 0, & \text{otherwise}, \end{cases}$$

at the final time T. Rare trajectories reaching a target basin in configuration space are often of interest as transition paths for reactive events, and significant development has been undertaken to efficiently generate them.[85–89] Computing optimal drift forces for generating these rare trajectories enables the study of reactive dynamics in a direct manner. We expect these algorithms to find use in the study of diffusive dynamics where Monte Carlo approaches have difficulty in sampling.[90–93] Furthermore, as the modified force is used with the original noise from the SDE, we have access to the full reactive trajectory ensemble, allowing for the interrogation of the statistics of the reactive events in a way that other direct path methods, such as nudged elastic band and zero temperature string methods, do not as they represent only the dominant path.[94–97] As a consequence, we expect that our method will find use when there is a large path space entropy.

The CGF for an indicator variable is given by

$$\lambda(s, T) = \ln \left\langle e^{-s h_\Gamma[\mathbf{x}_T]} \right\rangle_p \tag{31}$$

as an average in the original reference dynamics. From Eq. (13), the KL divergence being non-negative implies that the average return is bounded above by the value of the CGF $\lambda(s, T)$. The bound can be saturated only by the unique optimal drift force. We compare the value of the optimized return to the numerically exact estimates of the CGF, given as

$$\lambda(s, T) = \ln\left\{1 + (e^{-s} - 1)\int_\Gamma d\mathbf{x}\, \rho(\mathbf{x}, T)\right\}, \tag{32}$$

where the definition of the indicator function and the final time distribution $\rho(\mathbf{x}, T)$ evolved from a specific initial condition has been used. This form demonstrates that the statistics of a single time indicator observable is described solely by its mean,

$$\langle h_\Gamma \rangle_p = \int_\Gamma d\mathbf{x}\, \rho(\mathbf{x}, T). \tag{33}$$

For a rare fluctuation such that $\langle h_\Gamma \rangle_p < 0.5$, this form indicates that there are two distinct regimes in the biased ensemble with $s < 0$. For a small magnitude of the bias, the indicator function stays close to the unbiased value. Below a critical value of $s^* = -\ln[\langle h_\Gamma \rangle_p / (1 - \langle h_\Gamma \rangle_p)]$, the indicator crosses over to being close to 1. For all our calculations, we choose a fixed value of s estimated to be smaller than the threshold. With this value of s, we compute the right-hand side of Eq. (32) using an eigen-expansion of the propagator of the Fokker–Planck equation of the original dynamics and compare with the value of the average return from the gradient descent algorithms having the same value of s. The details of this calculation and comparison to an approximate Kramers escape rate are in Appendix C.

A. Softened Brownian bridges

The first example we consider is a softened version of the so-called Brownian bridge,[61,98] in which a one-dimensional Brownian motion starting from the origin is biased to end near a particular point. The reference dynamics is simply given by free diffusion,

$$dx = \sqrt{2} dW, \tag{34}$$

where comparing to Eq. (1), we have $G = \sqrt{2}$. We consider the target well, $\Gamma(x)$, to be defined as $\{1 - \epsilon \le x \le 1 + \epsilon\}$ with $\epsilon = 0.1$. The dynamics is simulated with a discrete time step of 0.001. We use a tilting parameter $s = -100$ to bias the original ensemble toward higher occurrence of the rare event.

We optimize a force and value function parameterized by linear combinations of Gaussian distributions with fixed variance and mean. Given a set of means $\{(x_m, t_m)\}_{m=0}^M$ and variances $\{\sigma_m\}_{m=0}^M$, the force and value function of a position x at time t are given by the coefficients $\{\theta_m\}_{m=0}^M$ and $\{\psi_m\}_{m=0}^M$ as

$$\begin{aligned} F_\theta(x, t) &= F(x) + \sum_{m=0}^M \theta_m e^{-\frac{(x-x_m)^2 + (t-t_m)^2}{2\sigma_m}}, \\ V_\psi(x, t) &= \sum_{m=0}^M \psi_m e^{-\frac{(x-x_m)^2 + (t-t_m)^2}{2\sigma_m}}, \end{aligned} \tag{35}$$

where the basis sets are initially a grid of 31 × 21 Gaussians in the x–t space. The Gaussians in time are spaced uniformly between $t \in [0, T)$, with standard deviations equal to half the grid spacing. A third of the Gaussians in space is placed between $x \in [-4, -0.5]$, a third in $x \in (-0.5, 1.5)$, and a third in $x \in [1.5, 5]$. Each of these three families of Gaussians has standard deviations half of the corresponding grid spacings. We initialize all $\theta_m = \psi_m = 0$.

We consider the performance of the three algorithms differing in the gradient used to optimize them. These include an algorithm that uses no value function (MCR), one that uses a value baseline (MCVB), and one that uses a value function for future returns with $\tau = 0.1$ (AC). We evaluate the efficiency of the algorithms by comparing learning curves, convergence with respect to the basis, and properties of the learned dynamics, shown in Fig. 1. All figures comparing different algorithms use the same noise history and the same amount of statistics such that the differences are solely ascribed to the learned dynamics. The MCR algorithm uses a learning rate of $\alpha^\theta = 0.4$. The MCVB algorithm uses learning rates of $\alpha^\theta = 0.4$ and $\alpha^\psi = 50$, and the AC algorithm uses learning rates of $\alpha^\theta = 1$ and $\alpha^\psi = 0.05$.

In Figs. 1(a)–1(c), we show learning curves for the total return, the average of the indicator observable, and the KL divergence, generated with 12 trajectories at each optimization step for each of the three algorithms. We have compared the results obtained with this finite basis to the numerically exact value of the optimal return and the corresponding observable average and KL divergence, obtained from Eq. (32) where for free diffusion, the distribution is known. We find that while all three algorithms quickly achieve a dynamics, which mostly fulfills the indicator function conditioning, the MCR algorithm struggles to optimize the KL divergence cost, while the MCVB and AC algorithms achieve converged values efficiently. As expected, each algorithm provides a variational estimate to the CGF with the MCVB and AC outperforming MCR. Trajectories with

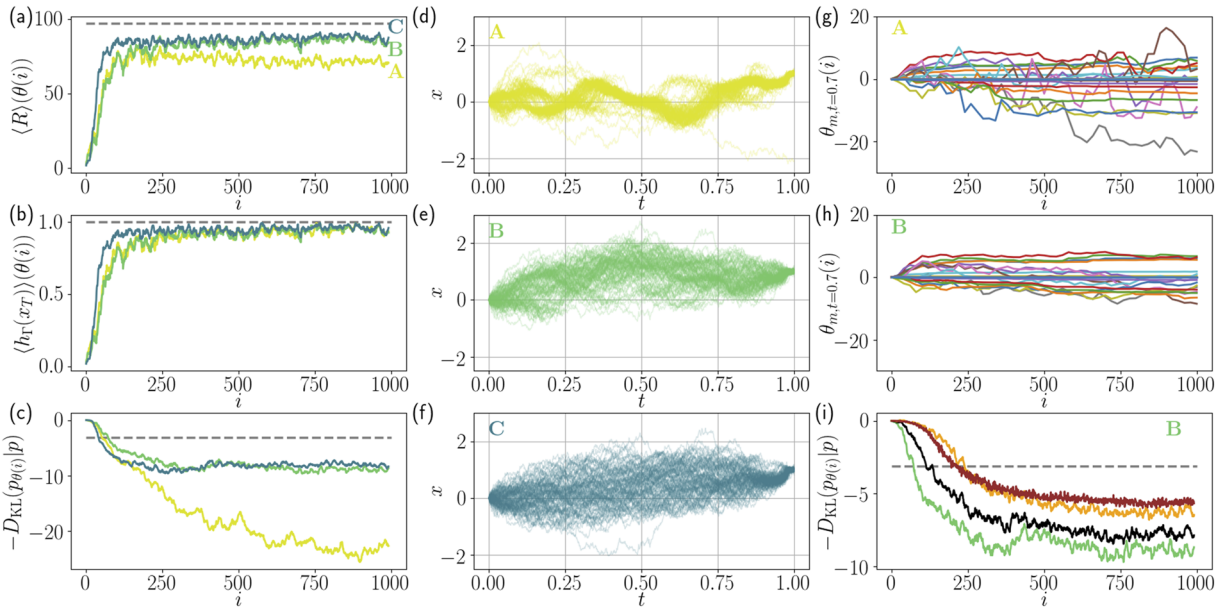

FIG. 1. Softened Brownian bridges: (left column) smoothened learning curves showing running estimates of the CGF (a), the average value of the indicator observable with the optimized dynamics (b), and the average cost function (c), as functions of optimization steps i, with the MCR ("A", yellow), MCVB ("B", green), and AC ("C", blue) algorithms. The horizontal gray dashed lines denote the numerically exact values. Middle column: 100 trajectories obtained with the final converged dynamics from the three different algorithms but with the same noise history. Right column: [(g) and (h)] the smoothened convergence of a time slice of the force parameters, as a function of optimization steps i, in the absence (MCR) and presence (MCVB) of a value function. (i) The convergence of the KL divergence cost with finer basis sets optimized with the MCVB algorithm. The green ($31x \times 21t$), black ($31x \times 41t$), orange ($31x \times 81t$), and brown ($41x \times 201t$) curves show that in the increasing basis limit, the cost-function estimate approaches the value expected from the numerically exact CGF.

the final learned dynamics for the three algorithms are plotted in Figs. 1(d)–1(f). The MCR algorithm finds forces that constrain the bridge trajectories too excessively, which results in the suboptimal estimate of the KL divergence. The AC trajectories are closest to the optimal bridge trajectories,[61] while the MCVB trajectories lie in between. The main reason for the difference in performance in the three algorithms is the resultant suppression in the statistical errors in the gradient estimate. This is illustrated in Figs. 1(g) and 1(h) where the convergence of the gradients of the 31 Gaussian coefficients at a time slice of $t = 0.7$ is shown for both MCR and MCVB. Since the α^θ learning rate is the same in both algorithms, the large suppression of fluctuations in the MCVB learning curves results from a more statistically converged gradient estimate using a value function. This suppression of gradient errors at limited statistics in the MCVB and AC algorithms is directly illustrated in Appendix B.

We have studied the convergence of the KL divergence estimate toward the optimal value extracted from the numerically exact CGF using the MCVB algorithm with an increasing position and time basis. We increased the number of time Gaussians, from 21 to 41 to 81, to observe the KL divergence cost shrinking as the finer-grained force can better support the singular indicator function condition at the end of the trajectory. We also ran the optimization with a much bigger basis of $41x \times 201t$ Gaussians and used 248 trajectories at every optimization step and learning rates of $\alpha^\theta = 5$ and $\alpha^\psi = 1000$. The Gaussians in x have standard deviations equal to half the grid spacing, while the Gaussians in t have standard deviations equal to a third of the grid spacing. While the estimate increased, in this particular problem, obtaining the numerically exact KL divergence would require the use of still finer-grained Gaussians in space and time in order to represent the singularities of the edges of the target region and of the last time step.

B. Barrier crossing with multiple reaction pathways

We now investigate the ability of the three algorithms to find the optimal dynamics in two-dimensional barrier-crossing problems, the first involving a potential allowing for multiple reaction pathways. The two-dimensional potential $U(\mathbf{x})$ we consider[99] has two minima and two degenerate reaction pathways involving the upper and lower halves of the $\mathbf{x} = (x, y)$ plane, as illustrated in Fig. 2. Barrier crossing from one well to another is a rare event occurring with one randomly chosen pathway.[100] Without prior knowledge of the possibility of multiple reaction paths, path sampling algorithms typically need special techniques to discover them.[101] We use our reinforcement learning algorithms to compute an optimal force $F_\theta(\mathbf{x}, t)$ that reproduces unbiased and uncorrelated reaction paths.

The reference equation of motion we consider is

$$d\mathbf{x} = -\nabla U(\mathbf{x}) + \sqrt{2}d\boldsymbol{W}, \tag{36}$$

where the matrix $\mathbb{G}$ is proportional to the identity. We use a discretization time step of 0.001. The trajectories start from the minimum of the left well, at $(x, y) = (-1.11, 0)$, and are allowed to run for a duration of $T = 1.5$ and checked for reaching the right target well defined as $x > 0$, $U(x, y) < 0$. This small region centered around

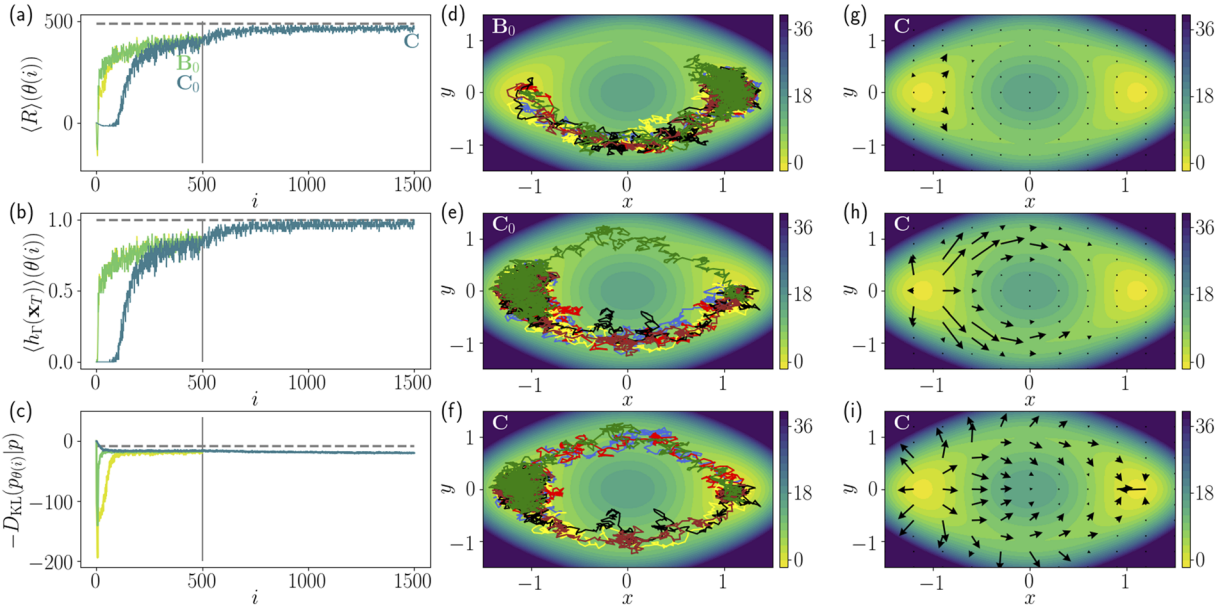

FIG. 2. Multiple reaction pathways: (left column) smoothened learning curves showing running estimates of the CGF (a), the average value of the indicator observable with the optimized dynamics (b), and the average cost function (c), as functions of optimization steps i, with the MCR (yellow), MCVB (green), and AC (blue) algorithms. The vertical gray lines denote the end of initialization and beginning of optimization run. The horizontal gray dashed lines denote the numerically exact values. The parameter values from the end of the initialization with MCVB and AC have been called B_0 and C_0, respectively. The forces at the end of optimization with AC are called C. Middle column: six representative trajectories obtained with forces B_0 (d), C_0 (e), and C (f). Right column: two-dimensional vectorial representation of the spatially dependent forces as a function of time, at $t = 1$ (g), $t = 1.3$ (h), and $t = 1.5$ (i), obtained from the converged parameters at C.

(1.11, 0) is used as Γ for defining the indicator function observable. The value of T has been chosen to be slightly greater than the typical transition path timescale such that the optimized force should reproduce trajectories that follow the natural steady state fluctuations of the system. As long as the choice of T is arbitrarily larger than the typical transition path timescale, the optimally generated trajectories will represent unbiased reactive transitions, with additional times being spent in the initial or final metastable states.[102] In the absence of an approximate transition path time estimate, the optimization can be performed over a range of T increasing by orders of magnitude until one enters the regime where side–side correlation functions for the dynamics of barrier crossing behave linearly.[100] We use a value of $s = -500$ to obtain the CGF. The force and the value function are approximated again as a grid of Gaussians with optimizable coefficients, a simple generalization of the one-dimensional Brownian bridge.

The duration of the trajectories we consider, T, is much smaller than the typical first passage time for the rare fluctuation we are interested in studying. As such, a general complication arises in initializing our algorithms in that in the absence of a modified force, few trajectories satisfy the indicator function condition. Consequently, the gradients for updating the modified forces are generally very small and noisy. In order to initialize our learning process, we start with a softened version of the indicator function of the form

$$\tilde{h}[\mathbf{x}_T] = -[(x_T - x_f)^2 + (y_T - y_f)^2], \tag{37}$$

which is quadratic and non-vanishing across the full domain. After optimizing the return with this observable, we obtain a force that can surpass the barrier, and the optimization with the sharp indicator function observable can begin. This technique of breaking down the optimization of the return into two segments prioritizing each of the two terms of the return is analogous to curriculum learning in reinforcement learning.[103] In many-body systems, the quadratic metric can be defined only in the space of the order parameter that distinguishes the initial and product states. For our multichannel problem, we initialize learning with $(x_f, y_f) = (1.11, 0)$ in the softened indicator, which is the minimum of the target well. Our approach consists of comparing the performance of the three algorithms MCR, MCVB, and AC in the initialization with the quadratic observable and then using the AC algorithm to optimize the return with the indicator function observable.

Figures 2(a)–2(c) demonstrate the learning curves for the full return, the average of the indicator function, and the KL divergence cost. Each of the three initializations uses 60 trajectories at every optimization step. The basis functions for the force and value function used are a grid of $21 \times 21 \times 41$ Gaussians in the $\mathbf{x} - t$ space for each component independently. The Gaussians are placed uniformly on the time axis $t \in [0, T)$, while the position Gaussians are distributed uniformly between $x \in [-1.5, 1.5]$ and $y \in [-1.5, 1.5]$. The learning rates used in the initialization are $\alpha^\theta = 1$ for MCR; $\alpha^\theta = 1$ and $\alpha^\Psi = 0.5$ for MCVB; and $\alpha^\theta = 1, \alpha^\Psi = 0.5$, and $\tau = 0.001$ for AC, and the learning rate for the final optimization is $\alpha^\theta = 0.2$,

$\alpha^{\psi} = 0.08$, and $\tau = 0.1$ in the AC algorithm. In the learning curves, we compare the convergence of the return with the numerically exact values obtained by computing the RHS of Eq. (32) with a spectral expansion using a discrete variable representation basis.[104] We see that all three algorithms quickly find forces that satisfy the conditioning, but the KL divergence cost is optimized best by the AC algorithm. While each affords a similar variational estimate after the initial optimization, we find qualitative differences in the family of barrier-crossing trajectories obtained from the MCR/MCVB and from the AC algorithm.

Typical trajectories obtained with forces from the end of initialization with MCVB and AC, and at the end of optimization with AC, are shown in Figs. 2(d)–2(f). The force obtained from MCVB spontaneously breaks the symmetry in the potential and chooses one reaction path out of the two. This force solution is a local optimum in the MCR and MCVB algorithms, and it does not naturally relax to a symmetric force that would be representative of the degeneracy of the reaction paths. Trajectories from the AC algorithm spend significant amount of time exploring the initial well such that the discovered forces recognize the presence of multiple pathways approximately. These forces are further refined during the second optimization such that the reactive trajectories obtained at the end are restored to be almost fully symmetric like the natural barrier-crossing fluctuations of the system are expected to be. These symmetric two-dimensional forces obtained at the end of the AC optimization are plotted at three slices of time in Figs. 2(g)–2(i). The forces grow in magnitude as a function of time and generally follow the contours of the underlying potential, and toward the end, they gather support in unlikely parts of the potential. The ability of the AC algorithm to discover time-dependent forces that lead to exploration of multiple reaction pathways can prove valuable in uncovering reactive trajectories in systems where such degeneracies are not known *a priori*.

C. Barrier crossing with a long-lived intermediate

Another difficult problem in the generation of transition paths and reactive trajectories typically comes from the presence of long-lived intermediates. In order to study the usefulness of our learning algorithms in this context, we consider as an example the dynamics on the so-called Müller–Brown potential.[105] This two-dimensional potential surface has been used extensively as a testing case for methods relying on the instantonic approximation for barrier-crossing trajectories.[102,106] The potential is a sum of four Gaussians,[107] where three local minima are separated by two barriers, as illustrated in Fig. 3. We employed our algorithms to find forces that generate uncorrelated trajectories that cross both barriers, starting from a local minimum and ending in the global minimum, that are positioned on the either side of the third metastable minimum.

The system evolves with diffusive Langevin dynamics of the same form as Eq. (36) using a time step of 0.000 1. We are interested in trajectories starting from $\mathbf{x} = (0.63, 0.03)$ in the rightmost local minimum and ending near the global minimum, centered around $\mathbf{x} = (-0.5, 1.5)$, with the indicator function region Γ being defined by $U(\mathbf{x}) < 145$). The trajectories are chosen to be of a fixed duration of $T = 0.15$, which is on the order of the expected total transition path timescale from Kramers theory added to the expected relaxation time in the intermediate well.[102,108] For initializing the forces, we use a softened quadratic modification of the indicator, in Eq. (37), with $s = -10\,000$, while we use a bias value of $s = -2000$ with the indicator observable to compute the CGF. To represent the x and y components independent of the time-dependent optimal force and to represent the value function, we use a basis of Gaussians with optimizable coefficients placed on a $21 \times 21 \times 21$ grid in $\mathbf{x} - t$. The time Gaussians are placed uniformly between $t \in [0, T)$, while the space Gaussians are placed uniformly between $x \in [-1.5, 1.5]$ and $y \in [-0.5, 2]$.

In Figs. 3(a)–3(c), we have compared the learning curves with MCR, MCVB, and AC algorithms during initialization with the smooth indicator function in Eq. (37) and the AC algorithm for the final optimization of the full return with the sharp indicator function. Each algorithm uses 60 trajectories at every optimization step to estimate the gradient. The learning rates for the initialization are $\alpha^{\theta} = 1$ for MCR; $\alpha^{\theta} = 1$ and $\alpha^{\psi} = 1$ for MCVB; and $\alpha^{\theta} = 0.5$, $\alpha^{\psi} = 0.2$, and $\tau = 0.0001$ for AC, and the learning rates for the final optimization are $\alpha^{\theta} = 0.1$, $\alpha^{\psi} = 0.01$, and $\tau = 0.01$ for AC. The learning curves have been compared with the approximately calculated values of the CGF and the KL div obtained with a Kramers escape rate estimate along the minimum energy path.[94]

We find that all the three algorithms optimize the quadratic observable relatively quickly, but the AC algorithm performs the best at optimizing the KL divergence cost. In Figs. 3(d)–3(h), we illustrate a few uncorrelated trajectories generated with the modified forces at various stages of the initialization and optimization with the AC method and the end of the initialization with the MCVB method. We find that the forces with the AC algorithm are such that the trajectories discover and cross the two barriers and the metastable well between them one after another. At the end of the AC initialization, the trajectories have discovered the metastable well and have crossed both barriers to end in the target well. The AC algorithm by this stage of optimization has also moved the major part of the short trajectory from staying in the initial well to the metastable well. This feature is constant throughout the AC optimization, with only minor changes in the force being carried out inside the target end well. The force from the MCVB initialization, on the other hand, only generates trajectories that connect the initial and target well without relaxing significantly in the metastable well. This would be contrary to the instantonic relaxation mechanism in the system, as the stochastic action is minimized by the local relaxation in the metastable well. In Fig. 3(i), we have plotted the potential energy as a function of time for 100 uncorrelated barrier-crossing trajectories, which are driven by the final force from the AC algorithm. The trajectories cross the two barriers at roughly fixed times and spend majority of the time in the metastable well.

The comparison of the three algorithms illustrates the significant improvement of convergence performance of the MCVB and AC algorithm over the naive MCR approach afforded by value functions. For rare reactive events, we have found that the AC algorithm is best suited to find trajectories that explore configuration space the most in search for the easier barriers to cross and thus is closest in resembling the natural fluctuations of the system. The errors in the converged values of the CGF depend on the truncation of the force basis and statistical uncertainties. The MCVB and AC algorithms preserve the computational scaling of the MCR with the trajectory duration and only change the prefactors of the scaling by a small

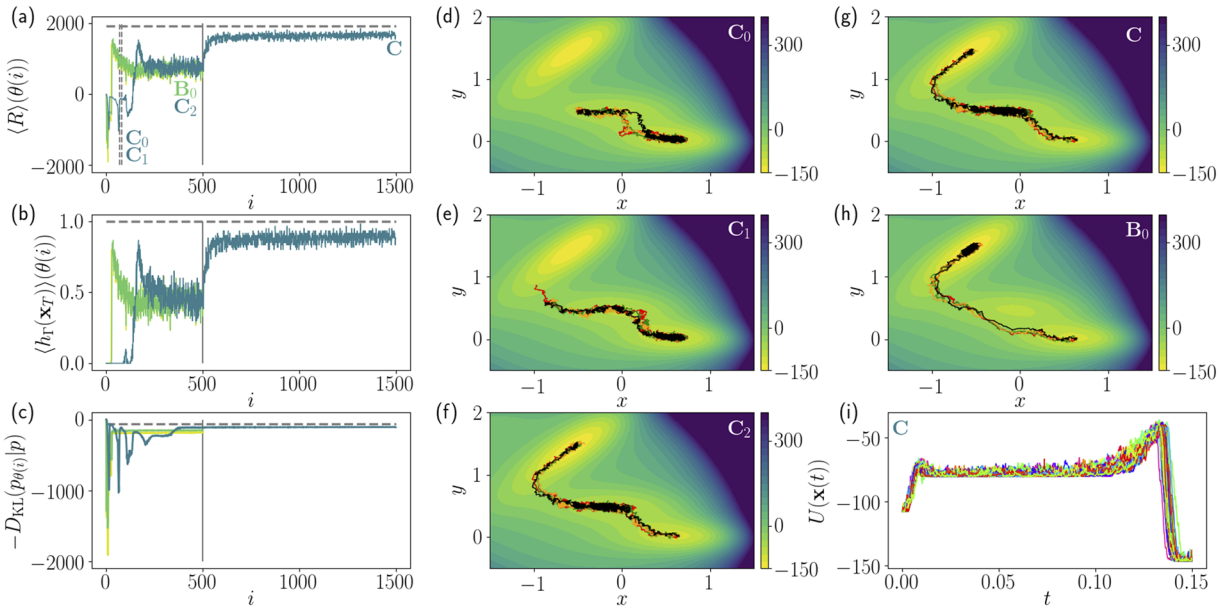

FIG. 3. Müller–Brown potential: (left column) smoothened learning curves showing running estimates of the CGF (a), the average value of the indicator observable with the optimized dynamics (b), and the average cost function (c), as functions of optimization steps i, with the MCR (yellow), MCVB (green), and AC (blue) algorithms. The vertical gray lines denote the end of initialization and beginning of optimization run. The horizontal gray dashed lines denote the approximate values from a Kramers escape rate approximation. On the AC learning curve in (a), the parameter values at $i = 70$ and $i = 80$ (the vertical dashed lines) have been called C_0 and C_1, respectively. The values at the end of initialization with MCVB and AC are called B_0 and C_2, and those at the end of AC optimization are called C. Middle column: four representative trajectories obtained with forces C_0 (d), C_1 (e), and C_2 (f). Right column: four representative trajectories obtained with forces C (g) and B_0 (h). (i) Potential energy as a function of time for 100 representative trajectories driven with the force parameters C.

fraction, making them viable methods for applications to complex systems. The AC algorithm with a small τ will incur a systematic error in the gradients if the value approximation is not accurate, which goes away at an intermediate τ but at the expense of a larger memory cost that may slow down the algorithm without any change in the scaling. Nevertheless, it is possible to use these algorithms with useful combinations of hyperparameters to achieve efficient convergence with a small amount of averaging. The value functions obtained during the optimizations serve as dynamical equivalents of the committor function in that they encode the expected value of the probability to reach the target well and the associated KL divergence cost while starting from any point in configuration space at any point in time. Understanding these connections to reaction coordinate design is likely a fruitful future direction of research.

V. GRADIENT OPTIMIZATION FOR INFINITE TIME DYNAMICS

We now generalize the approach of Sec. IV to focus on the statistics of time-integrated quantities in the long time limit. While for finite time, the generalized Doob transform is time-dependent, under mild assumptions in the long time limit, the optimal dynamics is time-homogeneous.[21] As a consequence, the parameterization of the modified force and value function is simplified and only explicitly dependent on the instantaneous configuration of the system. The generalization of the algorithms to this case consists of two main changes. First, we employ online learning since there is no end to each trajectory. Second, a modified definition of return and value is required to avoid divergences in the infinite time limit.

We formulate the infinite time problem by adapting an approach in reinforcement learning based on time-averaged returns.[57,109–111] Specifically, we consider the long time average of the KL divergence of the trajectory ensemble. Under assumptions of time-independence and ergodicity,

$$\begin{aligned} d_{\mathrm{KL}}(p_\theta|p_s) &= \lim_{T\to\infty}\frac{1}{T}D_{\mathrm{KL}}(p_\theta|p_s) \\ &= -\langle r(\mathbf{x},\dot{\mathbf{x}})\rangle_{p_\theta} + \lambda(s), \end{aligned} \tag{38}$$

the time average KL divergence reduces to an average over the steady state distribution of the instantaneous change in the return $r(\mathbf{x},\dot{\mathbf{x}})$. Above, we have defined a scaled CGF (SCGF),

$$\lambda(s) = \lim_{T\to\infty}\frac{1}{T}\ln Z(s,T), \tag{39}$$

which is finite as long as the cumulants of the time-integrated observable are time extensive. The reward, $r(\mathbf{x},\dot{\mathbf{x}})$, is defined as

$$\begin{aligned} r(\dot{\mathbf{x}},\mathbf{x}) &= -sA[\mathbf{x}] - s\mathbf{B}[\mathbf{x}]\cdot\dot{\mathbf{x}} \\ &\quad + \frac{1}{2}\left\{|\mathbb{G}^{-1}\cdot(\dot{\mathbf{x}}-\mathbf{F}_\theta)|^2 - |\mathbb{G}^{-1}\cdot(\dot{\mathbf{x}}-\mathbf{F})|^2\right\} \end{aligned} \tag{40}$$

and is time-independent and evaluable within the steady state. A gradient expression analogous to MCR can be derived straightforwardly.[25]

The previous definition of the value will diverge in the infinite time limit. A simple modification to address this issue is to remove the average reward scaled by the length of the trajectory segment, defining a differential return

$$\Delta R[\mathbf{X}_{t,t'}] = R[\mathbf{X}_{t,t'}] - (t'-t)\langle r(\dot{\mathbf{x}},\mathbf{x})\rangle_{p_\theta} \tag{41}$$

and corresponding differential value function

$$V(\mathbf{x}) = \lim_{T\to\infty}\langle \Delta R[\mathbf{X}_{0,T}]\rangle_{p_\theta,\mathbf{x}}, \tag{42}$$

which satisfies a modified Bellman equation

$$V(\mathbf{x}) = \langle V(\mathbf{x}_\tau) + \Delta R[\mathbf{X}_{0,\tau}]\rangle_{p_\theta,\mathbf{x}}, \tag{43}$$

containing the differential return between states, rather than the standard return, and relating the value of states separated by a period of time τ.

This modified Bellman equation can be simply rearranged to give an alternative equation for our time-averaged KL divergence

$$d_{\mathrm{KL}}(p_\theta|p_s) = -\frac{1}{\tau}\langle V(\mathbf{x}_\tau) + R[\mathbf{X}_{0,\tau}] - V(\mathbf{x})\rangle_{p_\theta,\mathbf{x}} + \lambda(s), \tag{44}$$

which we note holds for all $\mathbf{x}$. Differentiating the right-hand side of this equation with respect to θ does not involve the gradient of the stationary state. Therefore, taking the derivative and then averaging over the stationary state under F_θ,[112] we can write an estimate of the dynamical gradient as

$$\nabla_\theta d_{\mathrm{KL}}(p_\theta|p_s) = -\frac{1}{\tau}\langle \delta[\mathbf{X}_{0,\tau'}] y_\theta(\tau)\rangle_{p_\theta}, \tag{45}$$

where we have defined the differential temporal difference error

$$\delta[\mathbf{X}_{0,\tau'}] = V(\mathbf{x}_{\tau'}) + \Delta R[\mathbf{X}_{0,\tau'}] - V(\mathbf{x}_0), \tag{46}$$

reached after introducing an additional baseline in the form of $\tau'\langle r(\dot{\mathbf{x}},\mathbf{x})\rangle_{p_\theta}$. In this equation, we have arrived at a gradient estimate, which only depends on the gradient of the transition probabilities, contained in the Malliavin weights $y_\theta(\tau)$, and not on the gradient of the stationary state itself. This can thus be easily calculated during a simulation using the parameterized dynamics.

Note the period of time τ' over which the temporal difference is calculated is independent of the period of time τ over which the Malliavin weight is calculated, provided that the former is longer. The specific algorithm we consider involves taking the time τ small enough so that the Malliavin weight can be approximated by $\tau\dot{y}_\theta[\mathbf{x}_0]$, which is possible due to the time-homogeneous steady state we average within. We thus calculate the estimate as

$$\begin{aligned}\nabla_\theta d_{\mathrm{KL}}(p_\theta|p_s) &= -\langle \delta[\mathbf{X}_{0,\tau'}]\dot{y}_\theta(0)\rangle_{p_\theta} \\ &= \chi_{\mathrm{AC}}(\theta),\end{aligned} \tag{47}$$

which we denote as the actor–critic gradient in the long time limit. In practice, we will take $\tau' = \Delta t$, a single time step in a numerical simulation. A long time limit generalization of the MCVB gradient could be constructed similarly, but this is not considered here.

ALGORITHM 2. KL regularized differential actor–critic.

inputs force approximation $\mathbf{F}_\theta(\mathbf{x})$, value approximation $V_\psi(\mathbf{x})$
parameters learning rates α_i^θ, α_i^ψ, α_i^R; total updates N
initialize choose initial weights θ and ψ, initial average $\bar{r}$, define iteration variable i, individual error δ
$i \leftarrow 0$
repeat
  Generate a transition from $\mathbf{x}$ to $\mathbf{x}'$ according to the dynamics given by $\mathbf{F}_\theta(\mathbf{x})$ and noise vector $\mathbf{w} \sim \mathcal{N}(0,1)$
  $\dot{y}_\theta = \frac{\mathbf{w}^T\mathbb{G}^{-1}\nabla_\theta\mathbf{F}_\theta}{\sqrt{\Delta t}}$
  $\delta \leftarrow V_\psi(\mathbf{x}') + r(\mathbf{x},\mathbf{x}') - \bar{r} - V_\psi(\mathbf{x})$
  $\theta \leftarrow \theta + \alpha_i^\theta\delta\dot{y}_\theta$
  $\psi \leftarrow \psi + \alpha_i^\psi\delta\nabla_\psi V_\psi(\mathbf{x})$
  $\bar{r} \leftarrow \bar{r} + \alpha_i^R\delta$
  $i \leftarrow i+1$
until $i = N$

As in the finite time case, to construct this estimate, we also need an approximation to the value function, $V_\psi(\mathbf{x})$. Following a similar construction for the loss function as before, averaging the error over the stationary state, we estimate the gradient by which to update the value function parameters as

$$\nabla_\psi L(\psi) = -\langle \delta_\psi[\mathbf{X}_{0,\tau'}]\nabla_\psi V_\psi(\mathbf{x}_0)\rangle_{p_\theta}, \tag{48}$$

with the approximate temporal difference

$$\delta_\psi[\mathbf{X}_{0,\tau'}] = V_\psi(\mathbf{x}_{\tau'}) + \Delta R[\mathbf{X}_{0,\tau'}] - V_\psi(\mathbf{x}_0), \tag{49}$$

which also replaces the exact temporal difference in gradient estimates for the dynamics. Finally, we also have flexibility with our estimate of the scaled CGF. This can be done using a running average of the reward,

$$\langle r\rangle_{p_{\theta_i}} = \langle r\rangle_{p_{\theta_{i-1}}} + \alpha_r(\langle r\rangle_{p_{\theta_i}} - \langle r\rangle_{p_{\theta_{i-1}}}), \tag{50}$$

where α_r is the learning rate and the subscript p_{θ_i} denotes the parameters from the ith iteration. Alternatively, a lower variance, higher bias estimate may be constructed by noting that we can rearrange Eq. (43) to find

$$\langle r\rangle_{p_{\theta_i}} = \langle r\rangle_{p_{\theta_{i-1}}} + \alpha_r\langle \delta_\psi[\mathbf{X}_{0,\tau'}]\rangle_{p_{\theta_i}}, \tag{51}$$

an alternative equation for the average. After discretization, an algorithm based on utilizing single-transition estimates of these gradients is outlined in the pseudocode in Algorithm 2.

VI. RARE FLUCTUATIONS IN THE LONG TIME LIMIT

Here, we apply our approach to study the statistics of time-integrated currents in the long time limit. Persistent currents are the hallmark of a nonequilibrium system, and their fluctuations have been studied intensively.[26,113–115] Foundational results have been derived that constrain the symmetries of current fluctuations and relate their cumulants. For example, the fluctuation theorems dictate that the CGF satisfies a reflection symmetry about the driving force for the current due to the microscopic reversibility of the underlying

stochastic dynamics.[116,117] A number of numerical approaches have been developed to evaluate the scaled cumulant generating function, an example of a large deviation function.[1,85,118–122] These functions provide information of the long time behavior of stochastic systems and encode response relationships and stability. Within this context, our approach is similar to other controlled dynamics[14–19,25,123] based means of evaluating large deviation functions in the continuum and can be used directly as we show below or in concert with Monte Carlo algorithms.

To study the accuracy and efficiency of the algorithm, we consider statistics of the velocity of a particle on a ring of length $L = 2\pi$ with position x moving in a periodic potential. The periodic potential has the form $U(x) = U_0\cos(x)$ with magnitude U_0 and is driven by a constant force f such that

$$F(x) = -\frac{dU(x)}{dx} + f \tag{52}$$

is the total force for the particle on the ring. The observable we consider is the integrated current, $O[\mathbf{X}_{0,T}] = J[\mathbf{X}_{0,T}]$, given by

$$J[\mathbf{X}_{0,T}] = \int_0^T dt\, \dot{x}(t). \tag{53}$$

This observable has a different interpretation depending on whether the dynamics are under- or overdamped, both of which we consider below. In the underdamped case, the current is simply a function of the state with $A(\mathbf{x}) = v$ and $B = 0$, while in the overdamped case, it depends on the stochastic increment, $A(\mathbf{x}) = 0$, $B(\mathbf{x}) = 1$.

The corresponding scaled CGF we aim to compute is

$$\lambda(s) = \lim_{T\to\infty} \frac{1}{T} \ln \left\langle e^{-sJ[\mathbf{X}_{0,T}]} \right\rangle_p. \tag{54}$$

The first derivative of $\lambda(s)$,

$$v(s) = -\frac{d\lambda(s)}{ds}, \tag{55}$$

reports on the average velocity in the tilted ensemble and is a useful indicator of the tails of the reference distribution. The scaled CGF exhibits a Lebowitz–Spohn symmetry[116] such that

$$\lambda(s) = \lambda(-f - s), \tag{56}$$

where f is the affinity for the current. The scaled CGF can be computed by the numerical solution of a generalized eigenvalue problem,[25] which we use for this low dimensional system to compare the accuracy of our results.

Despite its simplicity, this system has been shown to present non-trivial non-equilibrium phenomena due to the competition between ballistic motion and diffusive motion.[122,124,125] Here, the overdamped regime acts as a simple benchmark, which can be easily solved by diagonalizating a projection of the Fokker–Planck equation.[125] The underdamped regime is a much more difficult problem to solve due to a higher dimensional state space and long relaxation time. Indeed, despite the access to the SCGF via diagonalization,[125] accurate results for the force in the underdamped case have been elusive. However, the actor–critic approach can solve this problem easily.

A. Current fluctuations of an overdamped particle

In the overdamped case, the evolution equation for the particle on a ring is given by

$$dx = F(x)dt + \sqrt{2}dW, \tag{57}$$

which is a dimensionless one-dimensional SDE. We integrate this equation with a time step of 0.001. Since the position is periodic, an ideal representation of both the force and value function is given by a Fourier series

$$F_\theta(x) = F(x) + a^\theta + \sum_{i=1}^{M} b_i^\theta \sin(ix) + c_i^\theta \cos(ix) \tag{58}$$

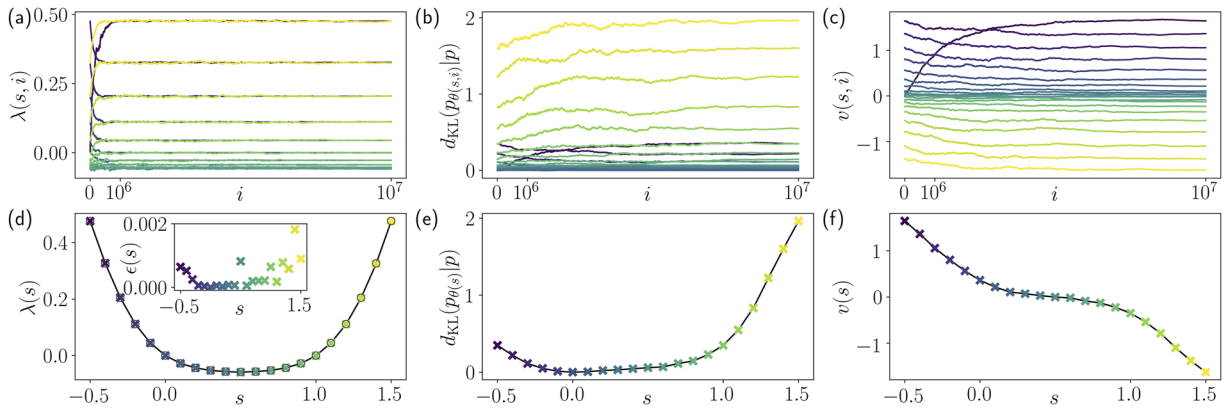

FIG. 4. Overdamped current fluctuations: (a) learning curves showing running estimates of the SCGF, (b) time-averaged KL divergence to the original dynamics $d_{\mathrm{KL}}(p_{\theta(s,i)}|p)$ during training for bias s at step i, and (c) the time-averaged velocity. The color of each curve indicates the value of the bias s, corresponding to the colors of the data points in the lower plots. Estimates of the (d) SCGF, (e) time-averaged KL divergence with the original dynamics, and (f) time-averaged velocity for the final dynamics found at each value of the bias s indicated on the x axis. The inset of (d) shows the absolute error with numerical diagonalization results, represented by the gray circles in (d).

and

$$V_\psi(x) = a^\psi + \sum_{i=1}^{M} b_i^\psi \sin(ix) + c_i^\psi \cos(ix), \tag{59}$$

with coefficients a^θ, a^ψ, $\{b_i^\theta, c_i^\theta\}_{i=1}^M$, and $\{b_i^\psi, c_i^\psi\}_{i=1}^M$ truncated to dimension M.

The results of the differential AC algorithm are shown in Fig. 4. We have truncated the basis with $M = 5$ and used learning rates of $\alpha_\theta = 0.1$ and $\alpha_\psi = 0.01$. We annealed across the range s considered, first learning the dynamics at $s = -0.5$, before sweeping across to $s = 1.5$ in steps of $\Delta s = 0.1$. The reward learning rate began at $\alpha_R = 10^{-5}$ and decreased linearly to $\alpha_R = 10^{-6}$ throughout training at each value of s to enable rapid convergence to an accurate result.

We detail estimates of three quantities calculated during the learning process. In Fig. 4(a), we show the estimate of $\lambda(s)$, the quantity the algorithm is attempting to maximize. In Fig. 4(b), we show an estimate of the time-averaged KL divergence. In Fig. 4(c), we show an estimate of the time-averaged velocity. These estimates are running averages calculated using the samples taken from each transition, with learning rates of $0.1\alpha_R$. Learning curves are plotted for training at each individual bias s during the annealing process. For small changes of s, we see that convergence to an accurate estimate of the scale CGF is achieved in $\sim 10^6$ training steps, each utilizing data from a single transition. This results in a speed of up to two orders of magnitude over the MCR algorithm.[25]

In Figs. 4(d)–4(f), we plot the end points of each of these learning curves for the three observables plotted in Figs. 4(a)–4(c). In Fig. 4(d), we see the expected Lebowitz–Spohn symmetry with reflection about $s = 1/2$ for the scaled CGF. The inset of Fig. 4 shows the absolute error compared to the diagonalization of the Fokker–Planck equation, $\epsilon(s)$, which illustrates quantitative accuracy across the s values considered. The maximal error is on the order of 1%. Likewise, we see the expected anti-symmetry in the time-averaged KL divergence and velocity in Figs. 4(e) and 4(f). Both of these are also quantitatively accurate. This antisymmetry implies that the optimal force differs from the reference force more for $s > 1$ than $s < 0$. This demonstrates that the regular production of trajectories with significant negative time-integrated velocities requires a substantial change in the systems dynamics, in contrast to those with a significant positive velocity. Nevertheless, the learning algorithm employed here is capable of parameterizing the modified force sufficiently well to work across these regimes.

B. Current fluctuations of an underdamped particle

In the underdamped case, the position and velocity evolve according to two coupled SDEs, given by

$$\begin{aligned} dx &= v\,dt, \\ dv &= F(x)dt - v\,dt + \sqrt{2}\,dW, \end{aligned} \tag{60}$$

where the noise acts only on the velocity, v, and the friction, inverse temperature, and mass are taken as unity. As mentioned before, we discretize our equations with a time step of 0.001. For the underdamped case, the modified force and value function depend on both the position and velocity of the particle. The approximation needs to only provide a single output for a force applied to the velocity, as the optimal dynamics cannot change the evolution of the position since the position is not directly influenced by noise. To do accomplish this, a simple approach we have taken is to discretize the force and value function approximation along the velocity dimension. More precisely, we can adapt the Fourier series from the overdamped case,

$$F_\theta(x, v) = a^\theta(v) + \sum_{i=1}^{M_1} b_i^\theta(v) \sin(ix) + c_i^\theta(v) \cos(ix), \tag{61}$$

with velocity dependent coefficients given by

$$a^\theta(v) = a_0 I_0(v) + a_{M_2+1} I_{M_2+1}(v) + \sum_{j=1}^{M_2} a_j I_{j,j+1}(v), \tag{62}$$

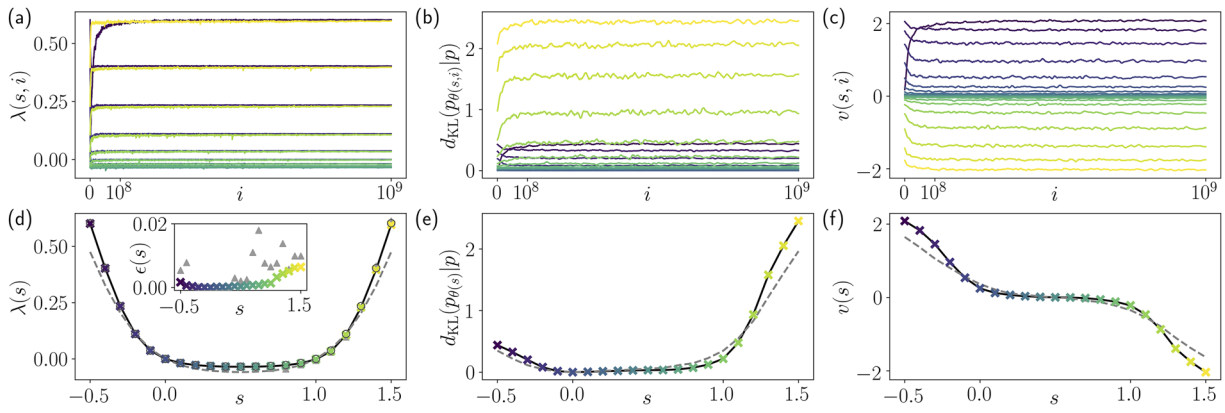

FIG. 5. Underdamped current fluctuations: (a) learning curves showing running estimates of the SCGF, (b) time-averaged KL divergence to the original dynamics $d_{\mathrm{KL}}(p_{\theta(s,i)}|p)$ during training for bias s at step i, and (c) the time-averaged velocity, calculated as the dynamics is trained. The color of each curve indicates the value of the bias s, corresponding to the colors of the data points in the lower plots. Estimates of (d) the SCGF, (e) time-averaged KL divergence with the original dynamics, and (f) time-averaged velocity of the final dynamics for each value of the bias s indicated on the x axis. The inset of (d) shows the absolute error with numerical diagonalization results, represented by the gray circles in (d). The results with estimated corrections using the algorithm in Ref. 25 are shown as triangles in (d) and its inset. The dashed curves in (d)–(f) show the results for the overdamped case for comparison.

where

$$I_{j,j+1}(v) = \begin{cases} 1, & v_0 + j\Delta v < v < v_0 + (j+1)\Delta v, \\ 0, & \text{else}, \end{cases} \tag{63}$$

and the boundary cases $I_0(v)$ and $I_{M_2+1}(v)$ return 1 for v less than v_0 or greater than $v_0 + (M_2 + 1)\Delta v$, respectively. We employ analogous equations for $b_i^\theta(v)$ and $c_i^\theta(v)$. To achieve accurate results, we find a spacing of $\Delta v = 0.02$ that is sufficient, with $v_0 = -8$ and $M_2 = 700$, providing a broad enough range to encompass all relevant velocities at the biases considered. We use a Fourier basis with $M_1 = 5$. As mentioned before, we use the same functional for the value function as for this modified force.

Figure 5 shows estimates of same three quantities as the overdamped case throughout the same annealed learning process. Here, we increased the value learning rate to $\alpha_\psi = 0.1$, retain a dynamics learning rate of $\alpha_\theta = 0.1$, and keep the scaled CGF learning rate fixed to $\alpha_R = 10^{-6}$ throughout training. The curves in Figs. 5(b) and 5(c) are produced from data calculated using the same learning rate as the scaled CGF before using a windowed average over 100 steps to smooth the curve. We generally see fast convergence to an accurate result in $\sim 10^8$ transitions worth of updates. The large learning time compared to the overdamped results reflects the significantly finer basis employed for the underdamped model.

The ends of these curves are plotted below in Figs. 5(d)–5(f). In the inset of Fig. 5(d), we see that we find accurate results compared to the numerically exact answers across the range of s considered. We see analogous results to the overdamped case, reproduced by the dashed lines in Figs. 5(e) and 5(f); the underdamped system obeys the expected Lebowitz–Spohn symmetry. Compared to the overdamped system, the features of the KL divergence and average velocity in underdamped system are sharper.

There are three distinct behaviors for the system as a function of s. For large negative s, the velocity increases significantly. For very large positive s, the velocity decreases analogously. For small and intermediate positive s, there is a broad plateau where the velocity is close to zero. These distinct regions are clearly demonstrated in

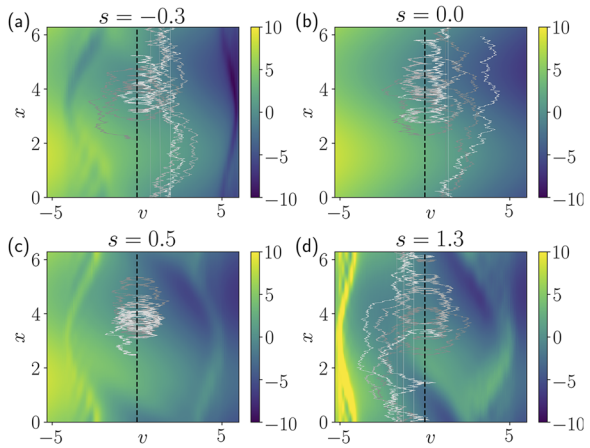

FIG. 6. Modified forces and their dynamics: the final forces learnt during the optimization process for biases $s = -0.3$ (a), 0.0(b), 0.5 (c), and 1.3 (d) with three sample trajectories of length $T = 10$ for each force.

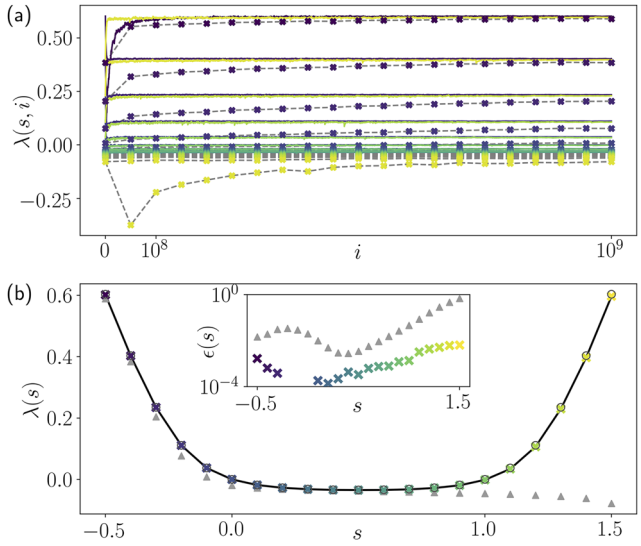

FIG. 7. Comparison between AC and MCR algorithms: (a) learning curves plotted vs the amount of data used during training for the AC algorithm (solid, colored lines) and the MCR algorithm (colored crosses and gray dashed lines). The curves and crosses are color coded by the value of the bias s being trained for. (b) Final results for the AC algorithm (colored crosses) and the MCR algorithm (gray triangles), with absolute errors to the value from numerical diagonalization shown in the inset.

Fig. 6 where we plot the final optimized forces for a set of s, along with sample trajectories generated by these forces. We see different behaviors for biases of $s < 0$, $0 < s < 1$, and $1 < s$. For $s < 0$, the trajectories regularly loop round the ring in the positive direction. For $0 < s < 1$, the trajectories generally do not transition round the ring and instead remain in a small region of space. For $s > 1$, the trajectories loop around the ring in the negative direction.

For comparison, we have optimized the same functional form using the MCR algorithm, as analogous to Ref. 25. The AC algorithm provides more accurate results than MCR, when optimized using the same amount of statistics.[25] The MCR results are produced by annealing across from $s = 1.5$ down to $s = -0.5$ in steps of 0.1. Training for each value of s involves 20 updates constructed using 50 trajectories with 10^6 time steps each for a total of 10^9 transitions worth of data. After optimizing the hyperparameters, we see in Fig. 7 that the convergence in the MCR algorithm is still much slower than the AC algorithm. As a consequence, the best results we can achieve using the same amount of transitions fail to converge to the correct values of the scaled CGF for biases close to $s \gtrsim 1$. This demonstrates one key advantage of utilizing value functions. Due to the reduction in variance of gradient estimates using a small amount of data, we can perform many more updates using the same amount of transitions, improving convergence.

VII. CONCLUSIONS

In this paper, we have demonstrated how regularized reinforcement learning algorithms can be used to optimize a diffusive dynamics to effectively sample rare trajectories. A key ingredient of our approach is a value function that estimates how relevant each state is to the rare dynamics, a function learnt while simultaneously

guiding optimization of the dynamics, allowing for reduced data generation and more detailed function approximations. Across a range of systems and observables, we found that the lower variance estimate of the gradient employing value functions enabled accurate and efficient characterization of rare dynamical fluctuations. In finite time problems, the AC algorithm, in particular, was able to solve particularly challenging problems associated with multiple reactive channels and long-lived intermediates. In the long time limit, the AC algorithm reproduces exact results for the cumulant generating function by directly optimizing to an accurate representation of the Doob dynamics, removing the need to calculate additional corrections or do additional importance sampling.

While we have focused here on the simulation of rare event dynamics and the direct evaluation of their likelihoods, the methods of finding optimized forces developed here can be straightforwardly combined with trajectory importance sampling methods, such as transition path sampling[85] or cloning,[120] to correct for inaccuracies associated with an incomplete basis. Indeed, previous work has demonstrated that auxiliary dynamics can significantly improve the statistical efficiency of trajectory sampling methods.[25,126–128] Furthermore, Monte Carlo approaches can be used to generate data to train the optimal dynamics in a feedback routine as previously demonstrated.[14,15] This could emphasize the parts of the state space relevant to the rare events earlier than by simply generating data with the current dynamics, thus speeding up optimization. Application to more complex models, such as many-body systems, will be an important development of this line of research. Accurate approximation of the force in many-body problems may require the use of more sophisticated function approximations, such as neural networks; however, a difficult balance will need to be struck between the representative power of the approximation and the computational cost to calculate it. More powerful function approximations will also necessitate the use of more sophisticated algorithms as training such approximations can become unstable when using correlated data, as we do here.

ACKNOWLEDGMENTS

A.D. and D.T.L. were supported by the NSF (Grant No. CHE1954580). D.C.R. and J.P.G. were supported by the University of Nottingham under Grant No. FiF1/3 and EPSRC Grant No. EP/R04421X/1. J.P.G. acknowledges All Souls College, Oxford, for support through a Visiting Fellowship during part of this work. D.C.R. is grateful for access to the University of Nottingham Augusta HPC service.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

A.D. and D.C.R. contributed equally to this work.

DATA AVAILABILITY

The data that support the findings of this study are openly available in Zenodo at https://doi.org/10.5281/zenodo.5513614.[129]

APPENDIX A: DISCRETE TIME STEP IMPLEMENTATIONS OF FINITE TIME ALGORITHMS

We now describe how the time-continuous equations of the reinforcement learning algorithm are efficiently implemented in simulations with a fixed discrete timestep Δt, though variable timesteps may be easily used. We use an Euler propagator to integrate the SDE in Eq. (10) as

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \Delta t \mathbf{F}_\theta(\mathbf{x}_t, t) + \mathbb{G}\Delta\mathbf{W}_t, \tag{A1}$$

where $\Delta\mathbf{W}$ is a Gaussian random variable with mean 0 and variance Δt. The trajectory probability from Eq. (11) is now given by products of stepwise probabilities

$$p_\theta[\mathbf{X}_{t,t+\Delta t}|\mathbf{x}_t] = \frac{\exp\left\{-\frac{1}{2\Delta t}\left|\mathbb{G}^{-1}(\mathbf{x}_{t+\Delta t} - \mathbf{x}_t - \Delta t\mathbf{F}_\theta(\mathbf{x}_t, t))\right|^2\right\}}{2\pi\Delta t\ \det(\mathbb{G})}. \tag{A2}$$

Next, we discretize the gradient of the logarithm of trajectory probabilities using the Ito convention. We propagate the Malliavin weights from Eq. (18) as

$$y_\theta(t + \Delta t) = y_\theta(t) + \left[\mathbb{G}^{-1}(\mathbf{x}_{t+\Delta t} - \mathbf{x}_t - \Delta t\mathbf{F}_\theta(\mathbf{x}_t, t))\right]\left[\mathbb{G}^{-1}\nabla_\theta\mathbf{F}_\theta(t)\right]. \tag{A3}$$

We also write the full return (14) through a sum of stepwise rewards as

$$R[\mathbf{x}_{t^-,t+\tau}] = \sum_{j;j\Delta t<\tau} r(\mathbf{x}_{j+1}, \mathbf{x}_j, t + j\Delta t), \tag{A4}$$

where the time step index j starts from -1 in this sum, with the notation t^- accounting for the time step before the current one, and the subscript j refers to the time $t + j\Delta t$. The reward at each step is defined as

$$\begin{aligned} r(\mathbf{x}_{j+1}, \mathbf{x}_j, t + j\Delta t) = &- s(A_j\Delta t + \mathbf{B}_j \cdot (\mathbf{x}_{j+1} - \mathbf{x}_j) + A(\mathbf{x}_{j+1})\delta_{jn}) \\ &+ \frac{\left[\mathbb{G}^{-1}(\mathbf{x}_{j+1} - \mathbf{x}_j - \Delta t\mathbf{F}_\theta(\mathbf{x}_j, t_j))\right]^2}{2} \\ &- \frac{\left[\mathbb{G}^{-1}(\mathbf{x}_{j+1} - \mathbf{x}_j - \Delta t\mathbf{F}(\mathbf{x}_j, t_j))\right]^2}{2}, \end{aligned} \tag{A5}$$

using the definition of the observable from Eq. (2) and accounting for an additional singular reward at the end of the trajectory after the last time step n. Here, the first three terms come from the observable and the last two terms represent the KL divergence between the original and optimized dynamics.

Now, we combine the rewards, Malliavin weights, and value functions in multiple ways to produce the gradients in the different algorithms. The pseudocodes of efficient implementations of these are presented below.

1. Monte Carlo returns

The gradient in the Monte Carlo returns algorithm can be rewritten from Eq. (19) as

$$\begin{aligned}\chi_{\mathrm{MCR}}(\theta,T) &= -\left\langle \int_0^T dt\; R[\mathbf{X}_{t^-,T}]\dot{y}_\theta(t)\right\rangle_{p_\theta}\\ &= -\left\langle \int_0^T dt\; \dot{y}_\theta(t)\int_{t^-}^T dt'\; \dot{R}(t')\right\rangle_{p_\theta}\\ &= -\left\langle \int_0^T dt\; \dot{R}(t)\int_0^{t^+} dt'\; \dot{y}_\theta(t')\right\rangle_{p_\theta}\\ &= -\left\langle \int_0^T dt\; \dot{R}(t) y_\theta(t^+)\right\rangle_{p_\theta}, \end{aligned} \tag{A6}$$

where the return has been written as a time integral of its differential changes and t^+ is shorthand for $t+\epsilon$ for some small positive ϵ. This has converted the double time integral into a single time integral, which is then evaluated on-the-fly while propagating the trajectory. An implementation of this algorithm with a fixed time step Δt is described in the pseudocode in Algorithm 3.

2. Monte Carlo returns with a value baseline

We use a similar technique to rewrite the double time integral for the gradient in the Monte Carlo value baseline algorithm, Eq. (22), using a single time integral as

$$\begin{aligned}&\chi_{\mathrm{MCVB}}(\theta,T)\\ &\quad= -\left\langle \int_0^T dt\; \{R[\mathbf{X}_{t^-,T}] - V_\psi(\mathbf{x}_t,t)\}\dot{y}_\theta(t)\right\rangle_{p_\theta,\psi=\psi_i}\\ &\quad= -\left\langle \int_0^T dt\; \left\{\dot{R}(t)y_\theta(t^+) - V_\psi(\mathbf{x}_t,t)\dot{y}_\theta(t)\right\}\right\rangle_{p_\theta,\psi=\psi_i}. \end{aligned} \tag{A7}$$

ALGORITHM 3. Finite time MCR.

inputs dynamical approximation $\mathbf{F}_\theta(\mathbf{x},t)$
parameters learning rate α^θ; total optimization steps I; trajectory length Tconsisting of J time steps of duration Δt each; number of trajectories N
initialize choose initial weights θ, define iteration variables i and j, force gradient δ_P, stepwise rewards r representing the increments in return
$i \leftarrow 0$
repeat
  Using chosen method to generate trajectories $\mathbf{X}_{0,T}$ with configurations, times, noises, Malliavin weights and rewards denoted by $\mathbf{x}_j, t_j, \Delta\mathbf{W}_j, y_\theta(t_j)$ and $r(\mathbf{x}_{j+1},\mathbf{x}_j,t_j) = r_j$, respectively.
  $j \leftarrow 0$
  $\delta_P \leftarrow 0$
  $y_\theta(t_0) \leftarrow 0$
  repeat
    $y_\theta(t_{j+1}) \leftarrow y_\theta(t_j) + \Delta\mathbf{W}_j \cdot [\mathbb{G}^{-1}\nabla_\theta \mathbf{F}_\theta(\mathbf{x}_j,t_j)]$
    $\delta_P \leftarrow \delta_P + r_j y_\theta(t_{j+1})$
    $j \leftarrow j+1$
  until $j = J$
  Average δ_P over N trajectories to get $\overline{\delta}_P$
  $\theta \leftarrow \theta + \alpha^\theta \overline{\delta}_P$
  $i \leftarrow i+1$
until $i = I$

We rewrite the gradient of the value error in Eq. (30) similarly as

$$\begin{aligned}\nabla_\psi L(\psi,\psi_i)|_{\psi=\psi_i} &= -\left\langle \int_0^T dt\; \left\{\dot{R}(t)\left(\int_0^{t^+} dt'\nabla_\psi V_\psi(t')\right)\right.\right.\\ &\qquad \left.\left. - V_\psi(t)\nabla_\psi V_\psi(t)\right\}\right\rangle_{p_\theta,\psi=\psi_i}\\ &= -\left\langle \int_0^T dt\; \left\{\dot{R}(t) z_\psi(t^+) - V_\psi(t)\dot{z}_\psi(t)\right\}\right\rangle_{p_\theta,\psi=\psi_i}, \end{aligned} \tag{A8}$$

where the arguments of the value function $V_\psi(\mathbf{x}_t,t)$ have been suppressed as $V_\psi(t)$ and the integral of the gradient of the value function up to and including current time has been denoted as $z_\psi(t^+)$. We explicitly set $V(\mathbf{x}_t,t)$ to 0 for any $t \geq T$, i.e., after the last time step, in these expressions. The single time integral is then evaluated on-the-fly as the trajectory is propagated. If the force and the value function approximations use the same set of basis functions as we do with a fixed grid of Gaussians, the MCVB algorithm incurs no additional computational cost over the MCR algorithm. An

ALGORITHM 4. Finite time MCVB.

inputs dynamical approximation $\mathbf{F}_\theta(\mathbf{x},t)$, value approximation $V_\psi(\mathbf{x},t)$
parameters learning rates α^θ, α^ψ; total optimization steps I; trajectory length T consisting of J time steps of duration Δt each; number of trajectories N
initialize choose initial weights θ and ψ, define iteration variables i and j, force and value function gradients δ_P, δ_V, stepwise rewards r representing the increments in return
$i \leftarrow 0$
repeat
  Using chosen method to generate trajectories $\mathbf{X}_{0,T}$ with configurations, times, noises, Malliavin weights, integral of value function gradients, and rewards denoted by $\mathbf{x}_j, t_j, \Delta\mathbf{W}_j, y_\theta(t_j)$, $z_\psi(t_j)$ and $r(\mathbf{x}_{j+1},\mathbf{x}_j,t_j) = r_j$, respectively.
  $j \leftarrow 0$
  $\delta_P \leftarrow 0$
  $\delta_V \leftarrow 0$
  $y_\theta(t_0) \leftarrow 0$
  $z_\psi(t_0 \leftarrow 0)$
  repeat
    $\dot{y}_\theta(t_j) \leftarrow \Delta\mathbf{W}_j \cdot [\mathbb{G}^{-1}\nabla_\theta \mathbf{F}_\theta(\mathbf{x}_j,t_j)]/\Delta t$
    $y_\theta(t_{j+1}) \leftarrow y_\theta(t_j) + \Delta t \dot{y}_\theta(t_j)$
    $\dot{z}_\psi(t_j) \leftarrow \nabla_\psi V_\psi(\mathbf{x}_j,t_j)$
    $z_\psi(t_{j+1}) \leftarrow z_\psi(t_j) + \Delta t \dot{z}_\psi(t_j)$
    $\delta_P \leftarrow \delta_P + r_j y_\theta(t_{j+1}) - V_\psi(\mathbf{x}_j,t_j)\dot{y}_\theta(t_j))$
    $\delta_V \leftarrow \delta_V + r_j z_\psi(t_{j+1}) - V_\psi(\mathbf{x}_j,t_j)\dot{z}_\psi(t_j)$
    $j \leftarrow j+1$
  until $j = J$
  Average δ_P, δ_V over N trajectories to get $\overline{\delta}_P$, $\overline{\delta}_V$
  $\theta \leftarrow \theta + \alpha^\theta \overline{\delta}_P$
  $\psi \leftarrow \psi + \alpha^\psi \overline{\delta}_V$
  $i \leftarrow i+1$
until $i = I$

implementation of this algorithm with a fixed time step Δt is described in the pseudocode in Algorithm 4.

3. Actor–critic

We rewrite the gradient in the actor–critic algorithm from Eq. (25) using a shift in time origin as

$$\chi_{\mathrm{AC}}(\theta,T) = -\left\langle \int_0^T dt\ \delta'[\mathbf{X}_{t^-,t+\tau},t]\dot{y}_\theta(t)\right\rangle_{p_\theta,\psi=\psi_i} = -\left\langle \int_\tau^{T+\tau} dt\ \delta'[\mathbf{X}_{t^--\tau,t},t-\tau]\dot{y}_\theta(t-\tau)\right\rangle_{p_\theta,\psi=\psi_i}, \quad \text{(A9)}$$

where the change in the return and the value function for $t \geq T$ is explicitly set to 0. We similarly write the gradient of the value error from Eq. (30) as

ALGORITHM 5. Finite time AC.

inputs dynamical approximation $\mathbf{F}_\theta(\mathbf{x},t)$, value approximation $V_\psi(\mathbf{x},t)$
parameters learning rates α^θ, α^ψ; total optimization steps I; trajectory length T consisting of J time steps of duration Δt each; temporal delay $M = \tau/\Delta t$; number of trajectories N
initialize choose initial weights θ and ψ, define iteration variables i and j, force and value function gradients δ_P, δ_V, stepwise rewards r representing the increments in return
$i \leftarrow 0$
repeat
  Using chosen method to generate trajectories $\mathbf{X}_{0,T}$ with configurations, times, noises, changes in Malliavin weights, value function gradients, temporal difference, rewards and cumulative rewards denoted by $\mathbf{x}_j, t_j, \Delta\mathbf{W}_j, \Delta y_\theta(t_j), \dot{z}_\psi(t_j), \delta'_j$, $r(\mathbf{x}_{j+1},\mathbf{x}_j,t_j) = r_j$ and $R\left[\mathbf{X}_{t_j-\tau,t_j}\right] = R_{j-M,j}$, respectively, and $r_j = V(\mathbf{x},t_j) = 0$ whenever $j < 0$ or $j \geq J$
  $j \leftarrow 0$
  $\delta_P \leftarrow 0$
  $\delta_V \leftarrow 0$
  $R_{-M,0} \leftarrow 0$
  repeat
    $R_{j-M,j} \leftarrow R_{j-M-1,j-1} + r_j - r_{j-M}$
    if $j < J$ **then**
      $\Delta y_\theta(t_j) \leftarrow \Delta\mathbf{W}_j \cdot [\mathbb{G}^{-1}\nabla_\theta \mathbf{F}_\theta(\mathbf{x}_j,t_j)]$
      $\dot{z}_\psi(t_j) \leftarrow \nabla_\psi V_\psi(\mathbf{x}_j,t_j)$
    end if
    if $j \geq M$ **then**
      $\delta'_j \leftarrow V(\mathbf{x}_j,t_j) + R_{j-M,j} - V(\mathbf{x}_{j-M},t_{j-M})$
      $\delta_P \leftarrow \delta_P + \delta'_j \Delta y_\theta(t_{j-M})$
      $\delta_V \leftarrow \delta_V + \delta'_j \dot{z}_\psi(t_{j-M})$
    end if
    $j \leftarrow j+1$
  until $j = J+M$
  Average δ_P, δ_V over N trajectories to get $\overline{\delta}_P, \overline{\delta}_V$
  $\theta \leftarrow \theta + \alpha^\theta \overline{\delta}_P$
  $\psi \leftarrow \psi + \alpha^\psi \overline{\delta}_V$
  $i \leftarrow i+1$
until $i = I$

$$\nabla_\psi L(\psi,\psi_i)\Big|_{\psi=\psi_i} = -\left\langle \int_\tau^{T+\tau} dt\ \delta'[\mathbf{X}_{t^--\tau,t},t-\tau] \times \nabla_\psi V_\psi(\mathbf{x}_{t-\tau},t-\tau)\right\rangle_{p_\theta,\psi=\psi_i}. \quad \text{(A10)}$$

These integrals are then evaluated on-the-fly along with trajectory propagation. Since the gradients involve correlations of the differential return r with the differential Malliavin weight $\dot{y}_\theta$ and the value function gradient $\dot{z}_\psi = \nabla_\psi V_\psi$ from τ time in the past, this makes it necessary to store and use this history, along with the reward and the value function, for the past $\tau/\Delta t$ time steps. Aside from this additional memory requirement, given a delay time τ, which is much smaller than the trajectory duration, the actor–critic algorithm has similar computational cost comparable to the MCR and MCVB algorithms. This implementation of the algorithm is described in the pseudocode in Algorithm 5.

APPENDIX B: COMPARING ERRORS IN GRADIENT ESTIMATES

In Fig. 8, we have directly compared the three algorithms for their ability to reduce the variance of the gradient estimates during optimization in the softened Brownian bridge problem. We have chosen the force and value function coefficients θ and ψ from the $i = 100$ step of the MCVB optimization run in Fig. 1(b) in the Brownian bridge problem. This value function is thus not exact for the corresponding force but is representative of typical inaccuracies encountered during learning. Keeping these coefficients fixed, we have estimated the gradients of the KL divergence using the three algorithms, while varying the number of uncorrelated trajectories N_w over which the estimates are averaged. Plotted in Fig. 8 are the total variance in the gradient estimate summed over all components, $\sum_m \mathrm{Var}[\nabla_{\theta_m} D_{\mathrm{KL}}(p_\theta|p_s)]$, from the different algorithms. The variances are computed from fluctuations over ten uncorrelated sets of N_w trajectories. The dependence on N_w in log–log scale corresponds to a linear trend with a slope of −1 as expected from the variance of sample means of uncorrelated samples. We find that the use of the MCVB and AC algorithms greatly reduces the variance compared to the MCR approach, equivalent to a 5 to 100 times

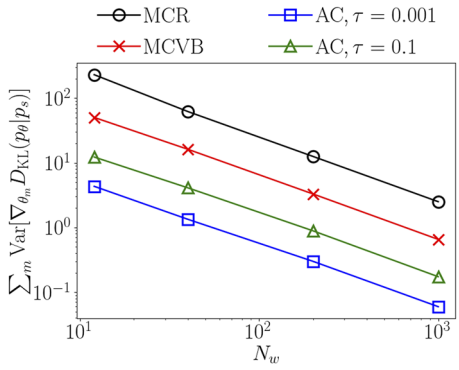

FIG. 8. Statistical convergence of gradient estimates: total variance of the gradient summed over all components using MCR (black), MCVB (red), AC with $\tau = 0.001$ (blue), and AC with $\tau = 0.1$ (green) as a function of the number of uncorrelated trajectories N_w for averaging.

increase in the amount of input trajectory data. We find that the smallest variance corresponds to the AC algorithm with the smallest possible τ, set to the time step of 0.001. However, this choice incurs a systematic error in the expectation of the gradient due to the inaccuracy in the value function, while neither MCVB nor AC with a large τ is susceptible to it. This is manifested in the scaled L^1 norm of the error in the expected gradient from the algorithms. The expectation is calculated over 10^5 trajectories, and the error in MCR is zero by definition. The L^1 norms of the errors, divided by that of the true gradient, are 0.22, 7.49, and 1.16 from MCVB, AC ($\tau = 0.001$), and AC ($\tau = 0.1$), respectively. This shows that the systematic error incurred by AC at small τ can be reduced by having a larger τ while still having significantly less variance than MCVB and MCR. The crossover between the systematic and statistical errors in the AC algorithm depending on τ is also the reason starting the optimization with a small τ and later annealing with a large τ is an efficient strategy, given that the memory requirement scales linearly with τ. We note that the systematic error is formally zero by definition in the expectation of the MCVB gradient estimate as well: the small non-zero value stems from a finite number of samples being used to estimate the expectation.

APPENDIX C: ALTERNATIVE CGF ESTIMATES

1. Numerically exact CGF

We have compared the CGF from the reinforcement learning algorithms in Sec. IV B with numerically exact values obtained from explicitly calculating $\langle h_\Gamma \rangle_p$ in Eq. (33) by solving the corresponding Fokker–Planck operator. The Fokker–Planck operator for the original dynamics in Eq. (36) is given by

$$L = -\nabla.\mathbf{F}(\mathbf{x}) + \nabla^2, \tag{C1}$$

where $\mathbf{F}(\mathbf{x}) = -\nabla U(x)$ is the underlying conservative force.

We want to use this operator in order to find the probability $\langle h_\Gamma \rangle_p$ as

$$\langle h_\Gamma \rangle_p = \int_\Gamma d\mathbf{x}\, \rho(\mathbf{x}, T) = \int_\Gamma d\mathbf{x}\, e^{LT}\delta(\mathbf{x} - \mathbf{x}_0). \tag{C2}$$

We exponentiate the operator in its spectral eigenbasis. Since the forces in the original dynamics are conservative, diagonalizing L becomes easier through a similarity transform into a Hermitian operator $\mathcal{L}$,[61,130]

$$\begin{aligned}\mathcal{L} &= e^{U(\mathbf{x})/2} L e^{-U(\mathbf{x})/2} \\ &= \nabla^2 - \frac{1}{4}(\nabla U(\mathbf{x}))^2 + \frac{1}{2}\nabla^2 U(\mathbf{x}).\end{aligned} \tag{C3}$$

We diagonalize $\mathcal{L}$ to obtain eigenvalues $-\lambda_n$ and eigenfunctions $\phi_n(\mathbf{x})$,

$$\mathcal{L}\phi_n(\mathbf{x}) = -\lambda_n \phi_n(\mathbf{x}). \tag{C4}$$

Since $\mathcal{L}$ is Hermitian, the eigenfunctions $\{\phi_n(\mathbf{x})\}$ are mutually orthonormal and can be used to introduce a resolution of identity

$$\delta(\mathbf{x} - \mathbf{x}_0) = \sum_n \phi_n(\mathbf{x}_0)\phi_n(\mathbf{x}). \tag{C5}$$

The original operator L related by the similarity transform has eigenvalues $-\lambda_n$ and eigenfunctions $e^{-U(\mathbf{x})/2}\phi_n(\mathbf{x})$. This spectral expansion of L can be used to estimate the probability $\langle h_\Gamma \rangle_p$ as

$$\begin{aligned}\langle h_\Gamma \rangle_p &= \int_\Gamma d\mathbf{x}\, e^{LT}\delta(\mathbf{x} - \mathbf{x}_0) \\ &= e^{U(\mathbf{x}_0)/2}\sum_n e^{-\lambda_n T}\int_\Gamma d\mathbf{x}\, e^{-U(\mathbf{x})/2}\phi_n(\mathbf{x}).\end{aligned} \tag{C6}$$

The final time T that we use in our barrier-crossing simulations is chosen such that $\tau_{\mathrm{rlx}} < T < \tau_{\mathrm{rxn}}$, where τ_{rlx} and τ_{rxn} are, respectively, the timescale of relaxation in the starting or the ending well and the timescale of the barrier-crossing reaction, which is expected to be the slowest dynamical mode in the system. Hence, when the set $\{\lambda_n\}$ is ordered, the factor $e^{-\lambda_n T}$ should be negligible for all but the few smallest values of n. The sum over n in Eq. (C6) is thus expected to converge within a few terms.

We diagonalize the operator $\mathcal{L}$ using a discrete variable representation basis constructed from Hermite polynomials[104] in two dimensions, $\chi_{M,N}(\alpha x, \alpha y)$, where $\alpha = 5$ is a scaling factor. We obtain identically converged estimates of $\langle h_\Gamma \rangle_p$ with basis sizes ranging from 50×50 to 100×100 using ten terms in the spectral expansion. The CGF value is then calculated using $\langle h_\Gamma \rangle_p$ in Eqs. (32) and (33).

2. CGF from Kramers escape rate

In one dimension, corresponding to a dynamics of

$$dq = -U'(q) + \sqrt{2}dW, \tag{C7}$$

an approximate expression for the barrier-crossing probability in time T is given by the Kramers escape rate in the overdamped limit[131] as

$$\langle h_\Gamma \rangle_p \approx \frac{T}{2\pi}(U''(q_A)|U''(q^\dagger)|)^{1/2} e^{-(U(q^\dagger) - U(q_A))}, \tag{C8}$$

where q is the reaction coordinate and q_A and $q^\dagger$ are the locations of the initial well and the barrier, respectively.

In the case of the Müller–Brown potential, we assume the ideal reaction coordinate to be along the minimum energy path obtained using a nudged elastic band method.[94–96] With the potential energy $U(q)$ computed along this path q, we use quadratic fits around the initial well (q_A) and around the largest barrier ($q^\dagger$) to find the double-derivative terms. Finally, we use this approximate value of $\langle h_\Gamma \rangle_p$ in Eqs. (32) and (33) to obtain the CGF.

REFERENCES

[1] H. Touchette, "The large deviation approach to statistical mechanics," Phys. Rep. **478**(1), 1–69 (2009).
[2] D. Chandler, "Barrier crossings: Classical theory of rare but important events," Classical Quantum Dyn. Condens. Phase Simul. **523**, 3–23 (1998).
[3] R. J. Webber, D. A. Plotkin, M. E. O'Neill, D. S. Abbot, and J. Weare, "Practical rare event sampling for extreme mesoscale weather," Chaos: Interdiscip. J. Nonlinear Sci. **29**(5), 053109 (2019).
[4] H. E. Stanley, X. Gabaix, P. Gopikrishnan, and V. Plerou, "Economic fluctuations and statistical physics: Quantifying extremely rare and less rare events in finance," Physica A **382**(1), 286–301 (2007).
[5] B. Peters, *Reaction Rate Theory and Rare Events* (Elsevier, 2017).
[6] C. Y. Gao and D. T. Limmer, "Transport coefficients from large deviation functions," Entropy **19**(11), 571 (2017).

[7]C. Y. Gao and D. T. Limmer, "Nonlinear transport coefficients from large deviation functions," J. Chem. Phys. **151**(1), 014101 (2019).
[8]D. T. Limmer, C. Y. Gao, and A. R. Poggioli, "A large deviation theory perspective on nanoscale transport phenomena," Eur. Phys. J. B **94**, 145 (2021).
[9]B. Kuznets-Speck and D. T. Limmer, "Dissipation bounds the amplification of transition rates far from equilibrium," Proc. Natl. Acad. Sci. U. S. A. **118**(8) (2021).
[10]F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl, "Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations," Proc. Natl. Acad. Sci. U. S. A. **106**(45), 19011–19016 (2009).
[11]R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. (MIT Press, 2018).
[12]D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Elsevier, 2001), Vol. 1.
[13]J. Zhang, Y. I. Yang, and F. Noé, "Targeted adversarial learning optimized sampling," J. Phys. Chem. Lett. **10**(19), 5791–5797 (2019).
[14]T. Nemoto, F. Bouchet, R. L. Jack, and V. Lecomte, "Population-dynamics method with a multicanonical feedback control," Phys. Rev. E **93**(6), 062123 (2016).
[15]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. Technol. **1**(3), 035004 (2020).
[16]S. Whitelam, D. Jacobson, and I. Tamblyn, "Evolutionary reinforcement learning of dynamical large deviations," J. Chem. Phys. **153**(4), 044113 (2020).
[17]H. J. Kappen and H. C. Ruiz, "Adaptive importance sampling for control and inference," J. Stat. Phys. **162**(5), 1244–1266 (2016).
[18]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).
[19]G. Ferré and H. Touchette, "Adaptive sampling of large deviations," J. Stat. Phys. **172**(6), 1525–1544 (2018).
[20]J. Zhang, Y.-K. Lei, Z. Zhang, X. Han, M. Li, L. Yang, Y. I. Yang, and Y. Q. Gao, "Deep reinforcement learning of transition states," Phys. Chem. Chem. Phys. **23**(11), 6888–6895 (2021).
[21]R. Chetrite and H. Touchette, "Nonequilibrium Markov processes conditioned on large deviations," Ann. Henri Poincaré **16**(9), 2005–2057 (2014).
[22]R. L. Jack and P. Sollich, "Effective interactions and large deviations in stochastic processes," Eur. Phys. J. Spec. Top. **224**(12), 2351–2367 (2015).
[23]R. Chetrite and H. Touchette, "Variational and optimal control representations of conditioned and driven processes," J. Stat. Mech. **2015**(12), P12001.
[24]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).
[25]A. Das and D. T. Limmer, "Variational control forces for enhanced sampling of nonequilibrium molecular dynamics simulations," J. Chem. Phys. **151**(24), 244123 (2019).
[26]T. GrandPre and D. T. Limmer, "Current fluctuations of interacting active Brownian particles," Phys. Rev. E **98**(6), 060601 (2018).
[27]L. Tociu, É. Fodor, T. Nemoto, and S. Vaikuntanathan, "How dissipation constrains fluctuations in nonequilibrium liquids: Diffusion, structure, and biased interactions," Phys. Rev. X **9**(4), 041026 (2019).
[28]T. GrandPre, K. Klymko, K. K. Mandadapu, and D. T. Limmer, "Entropy production fluctuations encode collective behavior in active matter," Phys. Rev. E **103**, 012613 (2021).
[29]T. Nemoto, E. Fodor, M. E. Cates, R. L. Jack, and J. Tailleur, "Optimizing active work: Dynamical phase transitions, collective motion, and jamming," Phys. Rev. E **99**, 022605 (2019).
[30]Y.-E. Keta, É. Fodor, F. van Wijland, M. E. Cates, and R. L. Jack, "Collective motion in large deviations of active particles," Phys. Rev. E **103**, 022603 (2021).
[31]A. Das and D. T. Limmer, "Variational design principles for nonequilibrium colloidal assembly," J. Chem. Phys. **154**(1), 014107 (2021).
[32]W. D. Piñeros and T. Tlusty, "Inverse design of nonequilibrium steady states: A large-deviation approach," Phys. Rev. E **103**(2), 022101 (2021).
[33]D. C. Rose, J. F. Mair, and J. P. Garrahan, "A reinforcement learning approach to rare trajectory sampling," New J. Phys. **23**(1), 013013 (2021).
[34]R. Munos and P. Bourgine, "Reinforcement learning for continuous stochastic control problems," in *Advances in Neural Information Processing Systems*, edited by M. Jordan, M. Kearns, and S. Solla (MIT Press, 1998), Vol. 10.
[35]R. Munos, "Policy gradient in continuous time," J. Mach. Learn. Res. **7**, 771–791 (2005).
[36]K. Doya, "Reinforcement learning in continuous time and space," Neural Comput. **12**(1), 219–245 (2000).
[37]S. J. Bradtke and M. O. Duff, "Reinforcement learning methods for continuous-time Markov decision problems" *Advances in Neural Information Processing Systems* **7**(7), 393 (1995).
[38]K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," Automatica **46**, 878–888 (2010).
[39]N. Frémaux, H. Sprekeler, and W. Gerstner, "Reinforcement learning using a continuous time actor–critic framework with spiking neurons," PLOS Comput. Biol. **9**, e1003024 (2013).
[40]R. W. Beard, G. N. Saridis, and J. T. Wen, "Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation," Automatica **33**(12), 2159–2177 (1997).
[41]M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," Automatica **41**(5), 779–791 (2005).
[42]V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," Nature **518**, 529–533 (2015).
[43]O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," Nature **575**, 350 (2019).
[44]D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," Science **362**(6419), 1140–1144 (2018).
[45]T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor–critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning* (PMLR, 2018), Vol. 80, pp. 1861–1870.
[46]T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor–critic algorithms and applications," arXiv:1812.05905 (2018).
[47]OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving Rubik's cube with a robot hand," arXiv:1910.07113 (2019).
[48]M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, "Reinforcement learning in different phases of quantum control," Phys. Rev. X **8**, 031086 (2018).
[49]M. Bukov, "Reinforcement learning for autonomous preparation of Floquet-engineered states: Inverting the quantum Kapitza oscillator," Phys. Rev. B **98**(22), 224305 (2018).
[50]J. Yao, M. Bukov, and L. Lin, "Policy gradient based quantum approximate optimization algorithm," in *Proceedings of The First Mathematical and Scientific Machine Learning Conference* (PMLR, 2020), Vol. 107, pp. 605–634.
[51]T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, "Reinforcement learning with neural networks for quantum feedback," Phys. Rev. X **8**, 031084 (2018).
[52]F. Chen, J.-J. Chen, L.-N. Wu, Y.-C. Liu, and L. You, "Extreme spin squeezing from deep reinforcement learning," Phys. Rev. A **100**, 041801(R) (2019).

[53]M. Dalgaard, F. Motzoi, J. J. Sørensen, and J. Sherson, "Global optimization of quantum dynamics with alphazero deep exploration," npj Quantum Inf. **6**, 1–9 (2020).
[54]A. Barr, W. Gispen, and A. Lamacraft, "Quantum ground states from reinforcement learning," in *Proceedings of The First Mathematical and Scientific Machine Learning Conference, volume 107 of Proceedings of Machine Learning Research (PMLR)*, edited by J. Lu and R. Ward (Princeton University, Princeton, NJ, 2020), pp. 635–653.
[55]W. Gispen and A. Lamacraft, "Ground states of quantum many body lattice models via reinforcement learning" arXiv:2012.07063 (2020).
[56]R. S. Sutton, D. Mcallester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in Neural Information Processing Systems* (MIT Press, 2000), pp. 1057–1063.
[57]P. Marbach and J. N. Tsitsiklis, "Approximate gradient methods in policy-space optimization of Markov reward processes," Discrete Event Dyn. Syst. **13**(1), 111–148 (2003).
[58]R. S. Sutton, "Learning to predict by the methods of temporal differences," Mach. Learn. **3**(1), 9–44 (1988).
[59]A. A. Budini, R. M. Turner, and J. P. Garrahan, "Fluctuating observation time ensembles in the thermodynamics of trajectories," J. Stat. Mech. Theory Exp. **2014**(3), P03012.
[60]T. Taniguchi and E. G. D. Cohen, "Onsager-Machlup theory for nonequilibrium steady states and fluctuation theorems," J. Stat. Phys. **126**(1), 1–41 (2007).
[61]S. N. Majumdar and H. Orland, "Effective Langevin equations for constrained stochastic processes," J. Stat. Mech. Theory Exp. **2015**(6), P06039 (2015).
[62]J. Grela, S. N. Majumdar, and G. Schehr, "Non-intersecting Brownian bridges in the flat-to-flat geometry," J. Stat. Phys. **183**, 49 (2021).
[63]B. De Bruyne, S. N. Majumdar, and G. Schehr, "Generating discrete-time constrained random walks and Lévy flights," Phys. Rev. E **104**, 024117 (2021).
[64]V. S. Borkar, S. Juneja, and A. A. Kherani, "Peformance analysis conditioned on rare events: An adaptive simulation scheme," Commun. Inf. Syst. **3**(4), 259–278 (2003).
[65]V. Popkov, G. M. Schütz, and D. Simon, "ASEP on a ring conditioned on enhanced flux," J. Stat. Mech. Theory Exp. **2010**(10), P10007.
[66]R. L. Jack and P. Sollich, "Large deviations and ensembles of trajectories in stochastic models," Prog. Theor. Phys. Suppl. **184**, 304–317 (2010).
[67]F. Carollo, J. P. Garrahan, I. Lesanovsky, and C. Pérez-Espigares, "Making rare events typical in Markovian open quantum systems," Phys. Rev. A **98**, 010103 (2018).
[68]H. J. Kappen, V. Gómez, and M. Opper, "Optimal control as a graphical model inference problem," Mach. Learn. **87**(2), 159–182 (2012).
[69]V. Y. Chernyak, M. Chertkov, J. Bierkens, and H. J. Kappen, "Stochastic optimal control as non-equilibrium statistical mechanics: Calculus of variations over density and current," J. Phys. A **47**(2), 022001 (2013).
[70]S. Thijssen and H. J. Kappen, "Path integral control and state-dependent feedback," Phys. Rev. E **91**, 032104 (2015).
[71]E. Todorov, "Efficient computation of optimal actions," Proc. Natl. Acad. Sci. **106**(28), 11478–11483 (2009).
[72]G. Neu, A. Jonsson, and V. Gómez, "A unified view of entropy-regularized Markov decision processes," arXiv:1705.07798 (2017).
[73]S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," arXiv:1705.07798 (2018).
[74]M. Geist, B. Scherrer, and O. Pietquin, "A theory of regularized Markov decision processes," in ICML, 2019.
[75]P. Warren and R. Allen, "Malliavin weight sampling: A practical guide," Entropy **16**(1), 221–232 (2013).
[76]D. Precup, R. S. Sutton, and S. P. Singh, "Eligibility traces for off-policy policy valuation," in ICML '00 (Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2000), pp. 759–766.
[77]T. Degris, M. White, and R. S. Sutton, "Off-policy actor–critic," in ICML, 2012.
[78]C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. thesis (Cambridge University, 1989).
[79]R. J. Williams, "Reinforcement-learning connectionist systems," Technical report No. NU-CCS-87-3, Northeastern University, 1987.
[80]R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," Mach. Learn. **8**(3), 229–256 (1992).
[81]L. Baird and A. W. Moore, "Gradient descent for general reinforcement learning," in *Advances in Neural Information Processing Systemx* (MIT Press, 1999), pp. 968–974.
[82]R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function pproximation," in *ICML'09* (ACM, New York, NY, 2009), pp. 993–1000.
[83]H. R. Maei, C. Szepesvári, S. Bhatnagar, D. Precup, D. Silver, and R. S. Sutton, "Convergent temporal-difference learning with arbitrary smooth function approximation," in *NIPS'09* (Curran Associates, Inc., 2009), pp. 1204–1212.
[84]H. R. Maei, "Gradient temporal-difference learning algorithms," Ph.D. thesis, University of Alberta, 2011.
[85]P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, "Transition path sampling: Throwing ropes over rough mountain passes, in the dark," Annu. Rev. Phys. Chem. **53**, 291–318 (2002).
[86]M. Invernizzi, P. M. Piaggi, and M. Parrinello, "Unified approach to enhanced sampling," Phys. Rev. X **10**(4), 041034 (2020).
[87]Y. Khoo, J. Lu, and L. Ying, "Solving for high-dimensional committor functions using artificial neural networks," Res. Math. Sci. **6**(1), 1–13 (2019).
[88]Q. Li, B. Lin, and W. Ren, "Computing committor functions for the study of rare events using deep learning," J. Chem. Phys. **151**(5), 054112 (2019).
[89]G. M. Rotskoff, A. R. Mitchell, and E. Vanden-Eijnden, "Active importance sampling for variational objectives dominated by rare events: Consequences for optimization and generalization," arXiv:2008.06334 (2020).
[90]T. R. Gingrich and P. L. Geissler, "Preserving correlations between trajectories for efficient path sampling," J. Chem. Phys. **142**(23), 234104 (2015).
[91]M. Grünwald, C. Dellago, and P. L. Geissler, "Precision shooting: Sampling long transition pathways," J. Chem. Phys. **129**(19), 194101 (2008).
[92]N. Guttenberg, A. R. Dinner, and J. Weare, "Steered transition path sampling," J. Chem. Phys. **136**(23), 234103 (2012).
[93]G. Stoltz, "Path sampling with stochastic dynamics: Some new algorithms," J. Comput. Phys. **225**(1), 491–508 (2007).
[94]G. Henkelman, B. P. Uberuaga, and H. Jónsson, "A climbing image nudged elastic band method for finding saddle points and minimum energy paths," J. Chem. Phys. **113**(22), 9901–9904 (2000).
[95]G. Henkelman and H. Jónsson, "Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points," J. Chem. Phys. **113**(22), 9978–9985 (2000).
[96]G. Henkelman, "Methods for calculating rates of transitions with application to catalysis and crystal growth," Ph.D. thesis, 2001.
[97]E. Weinan, W. Ren, and E. Vanden-Eijnden, "String method for the study of rare events," Phys. Rev. B **66**(5), 052301 (2002).
[98]D. Revuz and M. Yor, *Continuous Martingales and Brownian Motion* (Springer Science & Business Media, 2013), Vol. 293.
[99]The potential we use is $U(x,y) = 4/3[4(1-x^2-y^2)^2 + 2(x^2-2)^2 + ((x+y)^2-1)^2 - ((x-y)^2-1)^2 - 2]$.
[100]C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, "Transition path sampling and the calculation of rate constants," J. Chem. Phys. **108**(5), 1964–1977 (1998).
[101]H. Fujisaki, M. Shiga, and A. Kidera, "Onsager–Machlup action-based path sampling and its combination with replica exchange for diffusive and multiple pathways," J. Chem. Phys. **132**(13), 134101 (2010).
[102]M. Delarue, P. Koehl, and H. Orland, "*Ab initio* sampling of transition paths by conditioned Langevin dynamics," J. Chem. Phys. **147**(15), 152703 (2017).
[103]Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning* (Association for Computing Machinery, New York, 2009), pp. 41–48.
[104]V. Szalay, "Discrete variable representations of differential operators," J. Chem. Phys. **99**(3), 1978–1984 (1993).
[105]K. Müller and L. D. Brown, "Location of saddle points and minimum energy paths by a constrained simplex optimization procedure," Theor. Chim. Acta **53**(1), 75–93 (1979).

[106]S. Bonfanti and W. Kob, "Methods to locate saddle points in complex landscapes," J. Chem. Phys. **147**(20), 204104 (2017).
[107]The potential takes the form $U(x,y) = \sum_i A_i \exp[a_i(x-\bar{x}_i)^2 + b_i(x-\bar{x}_i)(y-\bar{y}_i) + c_i(y-\bar{y}_i)^2]$, where $A = (-200, -100, -170, 15)$, $a = (-1, -1, -6.5, 0.7)$, $b = (0, 0, 11, 0.6)$, $c = (-10, -10, -6.5, 0.7)$, $\bar{x} = (1, 0, 0.5, -1)$, and $\bar{y} = (0, 0.5, 1.5, 1)$.
[108]M. Laleman, E. Carlon, and H. Orland, "Transition path time distributions," J. Chem. Phys. **147**(21), 214103 (2017).
[109]A. Schwartz, "A reinforcement learning method for maximizing undiscounted rewards," in ICML, 1993.
[110]D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic Programming* (Springer, 1996).
[111]J. N. Tsitsiklis and B. Van Roy, "Average cost temporal-difference learning," Automatica **35**(11), 1799–1808 (1999).
[112]Taking this derivative results in gradients of the value function at x and x_T with respect to θ, however, these cancel out when averaging over the stationary state.
[113]B. Derrida, "Non-equilibrium steady states: Fluctuations and large deviations of the density and of the current," J. Stat. Mech. Theory Exp. **2007**(07), P07023.
[114]P. Pietzonka, A. C. Barato, and U. Seifert, "Universal bounds on current fluctuations," Phys. Rev. E **93**(5), 052145 (2016).
[115]T. Bodineau and B. Derrida, "Current fluctuations in nonequilibrium diffusive systems: An additivity principle," Phys. Rev. Lett. **92**(18), 180601 (2004).
[116]J. L. Lebowitz and H. Spohn, "A Gallavotti–Cohen-type symmetry in the large deviation functional for stochastic dynamics," J. Stat. Phys. **95**(1), 333–365 (1999).
[117]G. E. Crooks, "Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences," Phys. Rev. E **60**(3), 2721 (1999).
[118]F. Cérou and A. Guyader, "Adaptive multilevel splitting for rare event analysis," Stoch. Anal. Appl. **25**(2), 417–443 (2007).
[119]V. Lecomte and J. Tailleur, "A numerical approach to large deviations in continuous time," J. Stat. Mech. **2007**(03), P03004.
[120]C. Giardinà, J. Kurchan, and L. Peliti, "Direct evaluation of large-deviation functions," Phys. Rev. Lett. **96**(12), 120603 (2006).
[121]T. Lestang, F. Ragone, C.-E. Bréhier, C. Herbert, and F. Bouchet, "Computing return times or return periods with rare event algorithms," J. Stat. Mech. Theory Exp. **2018**(4), 043213.
[122]P. Tsobgni Nyawo and H. Touchette, "Large deviations of the current for driven periodic diffusions," Phys. Rev. E **94**, 032101 (2016).
[123]J. Dolezal and R. L. Jack, "Large deviations and optimal control forces for hard particles in one dimension," J. Stat. Mech. Theory Exp. **2019**(12), 123208.
[124]X.-g. Ma, Y. Su, P.-Y. Lai, and P. Tong, "Colloidal dynamics over a tilted periodic potential: Forward and reverse transition probabilities and entropy production in a nonequilibrium steady state," Phys. Rev. E **96**, 012601 (2017).
[125]L. P. Fischer, P. Pietzonka, and U. Seifert, "Large deviation function for a driven underdamped particle in a periodic potential," Phys. Rev. E **97**, 022143 (2018).
[126]U. Ray, G. K.-L. Chan, and D. T. Limmer, "Importance sampling large deviations in nonequilibrium steady states. I," J. Chem. Phys. **148**(12), 124120 (2018).
[127]T. Nemoto, R. L. Jack, and V. Lecomte, "Finite-size scaling of a first-order dynamical phase transition: Adaptive population dynamics and an effective model," Phys. Rev. Lett. **118**(11), 115702 (2017).
[128]G. Bartolucci, S. Orioli, and P. Faccioli, "Transition path theory from biased simulations," J. Chem. Phys. **149**(7), 072336 (2018).
[129]D. C. Rose, A. Das, D. T. Limmer, and J. P. Garrahan (2021). "Reinforcement learning of rare diffusive dynamics," Zenodo, V. 1.0. https://doi.org/10.5281/zenodo.5513614.
[130]H. Risken, "Fokker–Planck equation," in *The Fokker–Planck Equation* (Springer, 1996), pp. 63–95.
[131]R. Zwanzig, *Nonequilibrium Statistical Mechanics* (Oxford University Press, 2001).