Learning nonequilibrium control forces to characterize dynamical phase transitions

Jiawei Yan (闫嘉伟),[1] Hugo Touchette,[2] and Grant M. Rotskoff[1,*]
[1]*Department of Chemistry, Stanford University, Stanford, California 94305, USA*
[2]*Department of Mathematical Sciences, Stellenbosch University, Stellenbosch 7600, South Africa*

(Received 12 July 2021; revised 18 November 2021; accepted 24 January 2022; published 9 February 2022)

Sampling the collective, dynamical fluctuations that lead to nonequilibrium pattern formation requires probing rare regions of trajectory space. Recent approaches to this problem, based on importance sampling, cloning, and spectral approximations, have yielded significant insight into nonequilibrium systems but tend to scale poorly with the size of the system, especially near dynamical phase transitions. Here we propose a machine learning algorithm that samples rare trajectories and estimates the associated large deviation functions using a many-body control force by leveraging the flexible function representation provided by deep neural networks, importance sampling in trajectory space, and stochastic optimal control theory. We show that this approach scales to hundreds of interacting particles and remains robust at dynamical phase transitions.

DOI: 10.1103/PhysRevE.105.024115

I. INTRODUCTION

Large deviation techniques have been used recently to gain physical insight into the steady state and fluctuations of a diverse set of systems driven away from equilibrium, including diffusive and colloidal systems [1–3], glassy dynamics [4–7], interacting particle systems driven by external reservoirs [8–10], and active matter [11–14]. Fluctuations of dynamical quantities, such as currents and kinetic activities, provide information about complex pattern formation and phase behavior that can emerge in these systems when detailed balance is broken. The study of nonequilibrium fluctuations has also led to the discovery of fundamental results, such as the fluctuation relation [15–17], which encodes symmetries in the distribution of the entropy production, and, more recently, the thermodynamic uncertainty relation [18–20], which connects current fluctuations to dissipation.

The likelihood of fluctuations is described in large deviation theory by functions playing the role of nonequilibrium potentials, which are notoriously difficult to compute for complex and high-dimensional systems. While analytical treatment is possible for some systems [21–23], we must generally estimate these functions numerically. Many algorithms have been proposed for this purpose, based either on spectral methods or on sampling rare trajectories, using a combination of importance sampling [24–27], cloning [28–31], and reinforcement learning [32–35]. Good results are reported with most methods, although it remains challenging to obtain good convergence in systems with many degrees of freedom, especially when probing fluctuations near phase transitions [25].

In this paper, we present an algorithm that combines control theory, importance sampling, and, crucially, the robust and flexible function representations offered by neural networks to calculate large deviation functions. The algorithm uses recent developments in machine learning approaches to PDEs [36,37] and estimates large deviation functions by adaptively constructing a many-body control force that drives a nonequilibrium system of interest in an optimal way towards a given dynamical fluctuation. Unlike other methods that construct a control force, our approach is based on a direct stochastic optimization of a cost functional for trajectories, in which gradients are computed through the dynamics or via an adjoint stochastic dynamics, which is robust over long trajectories [38].

We illustrate our algorithm with two stochastic models: a simple diffusion showing a dynamical phase transition in the low-noise limit and a model of active Brownian particles driven by pair interactions and an alignment force. The results for both show that our approach is robust near dynamical phase transitions and efficiently scales to large systems of interacting particles, which are difficult to treat with spectral methods or cloning algorithms. For the active Brownian particle model, we are able for instance to estimate large deviation functions for systems of up to 200 particles, which is unreachable for cloning without substantial computational power. Our algorithm requires fewer parallel replicas than cloning algorithms, uses much less memory by relying on single trajectories, and converges faster, as we demonstrate with the simple diffusion model.

II. MODEL AND LARGE DEVIATIONS

We consider systems described by a stochastic differential equation (SDE) having the general form

$$d\boldsymbol{X}_t = b(\boldsymbol{X}_t)dt + \sigma d\boldsymbol{W}_t, \tag{1}$$

where $\boldsymbol{X}_t \in \mathbb{R}^d$ is the state of the system, $b : \mathbb{R}^d \to \mathbb{R}^d$ is the drift function, and $\boldsymbol{W}_t$ is a Wiener process acting as a noise source (often represented as a Gaussian random variable

*To whom correspondence should be addressed: rotskoff@stanford.edu

δ-correlated in time), which is multiplied by the noise matrix σ. This model captures the diffusive dynamics of many physical systems, as the drift or "force" $b(\boldsymbol{x})$ can include the gradient of a many-body potential energy describing the interactions among a large number of particles, in addition to nonconservative and hence nonequilibrium external forces. We assume that the drift and the noise source are such that $\boldsymbol{X}_t$ is ergodic, so that it has a unique probability stationary density, reached from any initial distribution in the long-time limit. For simplicity, we also assume that σ is independent of $\boldsymbol{x}$ and that the corresponding diffusion tensor $D = \sigma\sigma^T$ is invertible.

Given the dynamics for $\boldsymbol{X}_t$, we are interested in finding the distribution of time-integrated or "dynamical" observables having the form

$$A_T = \frac{1}{T}\int_0^T f(\boldsymbol{X}_t)dt + \frac{1}{T}\int_0^T g(\boldsymbol{X}_t)\circ d\boldsymbol{X}_t, \tag{2}$$

which represent many physical quantities of interest, depending on the choice for the functions $f : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}^d$. These include, for example, residence times, the entropy production, and other worklike quantities arising in stochastic thermodynamics [39,40]. While the exact probability density $\rho_T(a)$ of A_T cannot be obtained exactly, in general, it is known to scale for large observation times T as

$$\rho_T(a) \asymp e^{-TI(a)}, \tag{3}$$

where the symbol $\asymp$ denotes asymptotic equality up to logarithmic corrections. This result defines the large deviation approximation of $\rho_T(a)$, characterised by the rate function $I : \mathbb{R} \to \mathbb{R}$ [41]. Calculating or estimating this function has become a central problem in statistical physics, as it not only determines the likelihood of fluctuations of A_T around its typical value, but also provides information about the phase behavior and symmetries of nonequilibrium systems [42–44].

In most cases, the rate function is obtained not directly from the density of A_T, but from the Legendre transform of the scaled cumulant generating function (SCGF) of A_T, defined as

$$\psi(\lambda) = \lim_{T\to\infty}\frac{1}{T}\log \mathbb{E}_X e^{\lambda T A_T}, \tag{4}$$

where $\mathbb{E}_X$ denotes an expectation over (1) and $\lambda \in \mathbb{R}$ is a parameter conjugate to A_T. For Markov processes, the SCGF is the dominant eigenvalue of a linear operator, corresponding in the case of diffusions to a modification of the Fokker–Planck generator [45]. Hence, the computation of the SCGF and, in turn, the rate function, reduces to a spectral problem, which can be solved if the system's size or dimension is not too large. Alternatively, one can attempt to sample trajectories using path space Monte Carlo to estimate the expectation in the SCGF; however, this approach is not efficient, in general, since it involves exponentially rare events that do not occur spontaneously on timescales accessible to simulations.

To address these limitations, many strategies have been proposed recently, based on various numerical methods, including the power method [27], diffusion Monte Carlo [26,28–31], recurrent neural network [46], and reinforcement learning algorithms [33–35]. The method that we propose is based on importance sampling and proceeds by changing the process $\boldsymbol{X}_t$ to a new process $\boldsymbol{X}_t^u$ governed by the SDE

$$d\boldsymbol{X}_t^u = u_t\big(\boldsymbol{X}_t^u\big)dt + \sigma d\boldsymbol{W}_t, \tag{5}$$

in which the drift $b(\boldsymbol{x})$ is replaced by the control drift $u_t(\boldsymbol{x})$, to rewrite the expectation of the SCGF in terms of this new process as

$$\psi(\lambda) = \lim_{T\to\infty}\frac{1}{T}\log \mathbb{E}_{X^u}\left(e^{\lambda T A_T}\frac{d\mathbb{P}[\boldsymbol{X}^u]}{d\mathbb{P}_u[\boldsymbol{X}^u]}\right). \tag{6}$$

The idea with this change of process is to bias the estimation of the expectation towards trajectories that most contribute to the expectation—hence, the phrase "importance sampling"—thereby reducing the variance of the simulated estimator. These trajectories are rare with respect to $\boldsymbol{X}_t$; the goal is to make them typical with respect to the new process $\boldsymbol{X}_t^u$. The ratio $d\mathbb{P}[\boldsymbol{X}^u]/d\mathbb{P}_u[\boldsymbol{X}^u]$ is called the Radon–Nikodym derivative and is there to correct for the fact that the expectation is computed not from the original path probability (or path ensemble) $\mathbb{P}[\boldsymbol{X}^u]$, as in Eq. (4), but from a biased path probability $\mathbb{P}_u[\boldsymbol{X}^u]$ related to $\boldsymbol{X}_t^u$. This ratio can be computed explicitly along a given path using the Girsanov theorem [47].

The optimal change of process or optimal control process that achieves the smallest variance in importance sampling is known [48]. Its drift maximizes the cost

$$\mathcal{L}[\boldsymbol{X}^u, u] = \lambda T A_T - \frac{1}{2}\int_0^T \big(u_s - b\big)D^{-1}(u_s - b)\big(\boldsymbol{X}_s^u\big)ds, \tag{7}$$

which we derive in Appendix A. Moreover, it is known that, in the limit $T \to \infty$, the maximizing control drift is time-independent and that the maximum of the Lagrangian is the SCGF [48], so that

$$\psi(\lambda) = \lim_{T\to\infty}\frac{1}{T}\sup_u \mathbb{E}_{X^u}\mathcal{L}[\boldsymbol{X}^u, u]. \tag{8}$$

This variational representation of the SCGF has a clear interpretation: the first term in the Lagrangian (7) enforces the target rare event (constraint) $A_T = a$ with a Lagrange multiplier λ, while the second term is the Girsanov weight related to the change of drift that measures the extent to which the controlled process deviates from the original process [49]. From this point of view, the optimal control process is interpreted as the process closest to the original process, as measured by the Girsanov weight, that achieves $A_T = a$ as a typical rather than a rare event. In a more physical way, we can also interpret the optimal drift $u^*(\boldsymbol{x})$ of that process as an effective drift that "creates" the fluctuation $A_T = a$ [48,50,51]. This provide as physical mechanism explaining how fluctuations are created in time, which is useful for studying dynamical phase transitions.

III. ALGORITHM

The variational representation of the SCGF shown in Eq. (8) has a form that is standard in control theory and, as such, is amenable to Ritz-type methods that optimize a parametric representation $u(\boldsymbol{x}, \lambda; \boldsymbol{\theta})$ with respect to some set of variational parameters $\boldsymbol{\theta}$. Directly carrying out this optimization is nontrivial, as it requires representing a potentially complex, many-body force, motivating several sophisticated

Algorithm 1. Concurrent Training.

Data: Lagrangian $\mathcal{L}[\boldsymbol{X}^u, \delta u(\boldsymbol{X}^u_t, \lambda; \boldsymbol{\theta}), \lambda]$, initial $\boldsymbol{\theta}$, $k_{\max} \in \mathbb{N}$ total iterations, $T \in \mathbb{R}$ the duration of sampled trajectory, $N(m) \in \mathbb{N}$ the batch size. $\{\lambda_1, \lambda_2, \cdots, \lambda_M\} \subset \mathbb{R}$, $\alpha > 0$ the learning rate.
$k = 0$
while $k < k_{\max}$ **do**
  for $m = 1, \ldots, M$ **do**
    for $n = 1, \ldots, N(m)$ **do**
      Sample $\boldsymbol{X}^{u(\lambda_m)}_{[0,T],n}$ according to $d\boldsymbol{X}^u_t = [b(\boldsymbol{X}^u_t) + \delta u(\boldsymbol{X}^u_t, \lambda_m; \boldsymbol{\theta})]dt + \sigma d\boldsymbol{W}_t$ with initial condition $\boldsymbol{X}^{u(\lambda_m)}_{0,n}$;
  Compute $\mathcal{L}^{(m,n)}_{\boldsymbol{\theta}} = \lambda_n \int_0^T f(\boldsymbol{X}^{u(\lambda_m)}_{t,n})dt + g(\boldsymbol{X}^{u(\lambda_m)}_{t,n}) \circ d\boldsymbol{X}^{u(\lambda_m)}_{t,m}$ $-\frac{1}{2}\int_0^T \delta u(\boldsymbol{X}^{u(\lambda_m)}_{t,n}, \lambda_m; \boldsymbol{\theta}) D^{-1} \delta u(\boldsymbol{X}^{u(\lambda_m)}_{t,n}, \lambda_m; \boldsymbol{\theta})dt \; \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})$ $= \frac{1}{M}\sum_{m=1}^{M} \frac{1}{N(m)} \sum_{n=1}^{N(m)} \nabla_{\boldsymbol{\theta}} \mathcal{L}^{(m,n)}_{\boldsymbol{\theta}}$
  Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$
  Update the initial condition $\boldsymbol{X}^{u(\lambda_m)}_{0,n} \leftarrow \boldsymbol{X}^{u(\lambda_m)}_{T,n}$
  procedure (OPTIONAL) REPLICA EXCHANGE:
    Select two random integers n_1, n_2 such that $n_i \leqslant N(m_i)$;
    Compute the Radon-Nikodym derivative: $M_T = \frac{d\mathbb{P}[\boldsymbol{X}^{u(\lambda_{m_1})}_{[0,T],n_1}]}{d\mathbb{P}[\boldsymbol{X}^{u(\lambda_{m_2})}_{[0,T],n_2}]} = \frac{\exp\{-\frac{1}{4D}\int_0^T |\dot{\boldsymbol{X}}^{u(\lambda_{m_1})}_{t,n_1} - u(\boldsymbol{X}^{u(\lambda_{m_1})}_{t,n_1}, \lambda_{m_1})|^2 dt\}}{\exp\{-\frac{1}{4D}\int_0^T |\dot{\boldsymbol{X}}^{u(\lambda_{m_2})}_{t,n_2} - u(\boldsymbol{X}^{u(\lambda_{m_2})}_{t,n_2}, \lambda_{m_2})|^2 dt\}}$
    $u \sim \text{Uniform}(0, 1)$
    if $u < \min[1, M_T]$
      exchange $\boldsymbol{X}^{u(\lambda_{m_1})}_{0,n_1}$ and $\boldsymbol{X}^{u(\lambda_{m_2})}_{0,n_2}$.
return: $\boldsymbol{\theta}$.

strategies that rely on intricate basis functions, Malliavin weight sampling, and reinforcement learning [33–35,52].

Our contribution is to solve this high-dimensional control problem using gradient-based optimization and deep neural networks, which are well-suited to this task [53–58] due to their robust function approximation properties, even in high-dimensional settings. The pseudocode of our optimization algorithm is presented in Algorithm 1 and a Python source code is available online [59]. There are four important components to our algorithm:

a. Neural network representation of the drift. Following recent works on the deep Ritz method [36], we represent the change in control drift

$$\delta u(\boldsymbol{x}) = u(\boldsymbol{x}) - b(\boldsymbol{x}), \tag{9}$$

using a neural network that contains multiple layers L_i, where each layer consists of two linear transformation, two nonlinear activation functions and a residual connection:

$$L_i(\boldsymbol{X}) = \phi[W_{i,2} \cdot \phi(W_{i,1}\boldsymbol{X} + b_{i,1}) + b_{i,2}] + \boldsymbol{X}, \tag{10}$$

where $W_{i,j} \in \mathbb{R}^{h\times h}$ and $b_{i,j} \in \mathbb{R}^h$ are parameters for the ith layer, h is the dimension of the hidden layers, and ϕ is the activation function. The residual connection expressing each layer as $L_i(\boldsymbol{X}) = f(\boldsymbol{X}) + \boldsymbol{X}$ helps with stability and avoiding the vanishing gradient problem.

Since our approach requires simulating trajectories from Eq. (5), an unbounded activation such as ReLU may lead to divergence of the sampled trajectories. To avoid this problem, we use $\tanh(\cdot)$ as the activation function throughout this paper though other nonlinearities may also be suitable. The full network can then be expressed as

$$z_{\boldsymbol{\theta}}(\boldsymbol{X}) = L_n \otimes \cdots \otimes L_1(\boldsymbol{X}). \tag{11}$$

The input $\boldsymbol{X} \in \mathbb{R}^d$ for the first layer is padded by a zero vector when $d < h$. Finally, the ansatz $\delta u(\boldsymbol{X}_t, \lambda; \boldsymbol{\theta}) \in \mathbb{R}^d$ is expressed as a linear transform of $z_{\boldsymbol{\theta}}(\boldsymbol{X})$.

b. Loss estimation and gradient. The loss function is estimated, for a given change of drift $\delta u(\cdot, \lambda; \boldsymbol{\theta})$, with a collection or "batch" of N trajectories generated in parallel using direct Langevin dynamics. The variance and convergence of the resulting estimator are discussed in Appendix B, which shows that short time trajectories suffice when the batch size is large.

From the estimated loss, we proceed to compute the loss gradient to update the parameters $\boldsymbol{\theta}$ by differentiating through the solution of the SDE (5) using recent developments in machine learning [38,60]. Over short times, we use direct backpropagation of the dynamics through a Stratonovich time discretization of the SDE to compute $\nabla_{\boldsymbol{\theta}}\mathcal{L}$. The computational graph that contains all the gradient information consumes significant memory resources in this case, so over longer timescales, we calculate $\nabla_{\boldsymbol{\theta}}\mathcal{L}$ by solving instead an adjoint SDE, detailed in Appendix C. This method is stable and only requires that we keep the noise history and solve the SDE backward in time.

c. Estimation of the SCGF and rate function. The repeated gradient minimization of the loss yields, after enough gradient steps, a single estimated point $\psi(\lambda)$. To obtain the rate function, the SCGF must be estimated by training the neural network for multiple values of λ either simultaneously or sequentially. In the first case, which we term concurrent training, the loss function at each training step is evaluated as the mean of the loss function with each λ_m from a set $\{\lambda_1, \lambda_2, \cdots, \lambda_M\}$. We find that the expressiveness of the neural networks we use allows a single force function $u(\cdot, \lambda)$ to capture the control forces for a wide range of λ, even when there are multiple dynamical phases. For high-dimensional systems, where the batch size is limited, one may alternatively start with a given λ, e.g., 0, and sequentially increase or decrease λ. This sequential training approach, which is similar to transfer learning [61], shows fast convergence.

d. Replica exchange. Near dynamical phase transitions, which lead to rapid changes of the optimal control forces as a function of λ, we have found that it is useful to share information from distinct values of λ by employing a path space variant of the replica exchange method [62], in which two trajectories corresponding to different λ are swapped according to a Metropolis-Hastings algorithm that uses the loss function in place of an energy. This increases the likelihood of sampling trajectories in different phases, leading to a more accurate estimate of the SCGF.

IV. APPLICATIONS

We test our algorithm in this section on two models, which have been studied before in the context of large deviations and which illustrate two different challenges faced by large deviation numerical methods, namely, critical slowing-down effects related to dynamical phase transitions, and the representation of the control force for high-dimensional systems, in particular, many-body systems.

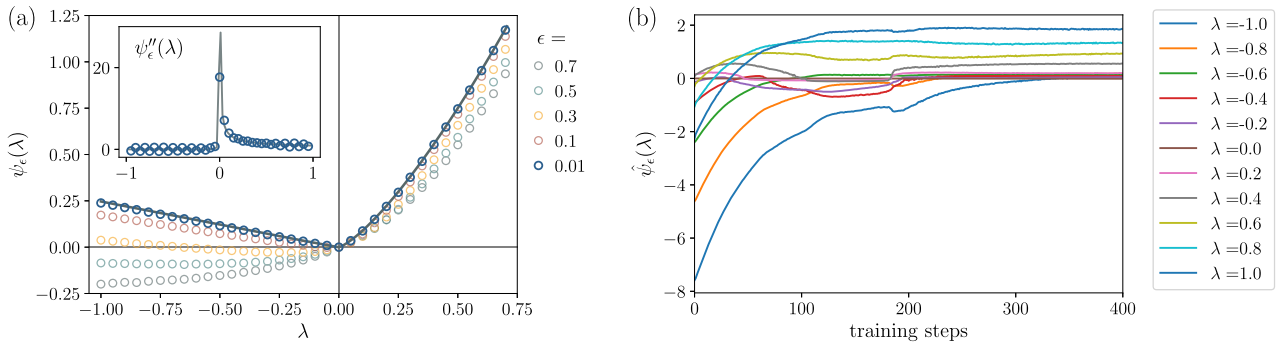

FIG. 1. Results of the diffusion in the quartic potential. (a) SCGF for the observable (13) for decreasing temperatures ϵ. The solid line represents the exact solution (14) in the zero-noise limit. The inserted figure shows the second derivative of the SCGF for $\epsilon = 0.01$, confirming a second order dynamical phase transition. In this example the hidden layer dimension and number of layers of the neural network are 50 and 2, respectively. A smaller hidden layer dimension such as 10 is able to generate results with similar accuracy but requires longer time for training. We first select 11 λ values uniformly from -1 to 1, where each λ contains 20 replica. At each training step, a total number of 220 trajectories with $T = 5$ are generated with the Euler-Maruyama method ($dt = 10^{-3}$). The neural network is updated through standard back propagation where the gradient is computed by the adaptive gradient algorithm method (AdaGrad) with a learning rate 5×10^{-3}. The resulting estimation of the SCGF is then refined by changing λ and simulating the resulting driven process. (b) Illustration of the concurrent training with $\epsilon = 0.01$. Each line corresponds to the evolution of the cost function with a specific λ.

A. Simple diffusion

For the first test, we consider a 1D diffusion in a quartic potential,

$$dX_t = -X_t^3 dt + \sqrt{2\epsilon} dW_t, \tag{12}$$

and we focus on the observable,

$$A_T = \frac{1}{T}\int_0^T X_t(X_t + 1)dt. \tag{13}$$

For this model, the SCGF scaled by the strength ϵ of the noise is known to display a second-order dynamical phase transition in the small-noise limit, meaning that the derivative of $\psi_\epsilon(\lambda) = \epsilon\psi(\lambda)$ is not differentiable at $\lambda = 0$ when considering the additional limit $\epsilon \to 0$. This can be checked from the exact result

$$\psi_0(\lambda) = \lim_{\epsilon\to 0} \psi_\epsilon(\lambda) = \max_q\{\lambda(q^2 + q) - q^6/4\}. \tag{14}$$

Resolving this phase transition using cloning algorithms is challenging, due to a critical slowing down of the dynamics, which can be alleviated to some degree by incorporating adaptive feedback methods [25].

The low-noise limit is not a bottleneck for our algorithm. Using short trajectories ($T = 5$), we concurrently trained a single neural network with a set of values for λ in the range $[-1, 1]$. The results, plotted in Fig. 1(a), agree exceptionally well for $\epsilon = 0.01$ with the exact result obtained in the low-noise limit. For most values of λ, we find in fact that the normalized mean squared error between our estimate of the SCGF and the exact result is about 0.2%. This can be reduced by training the network for a single λ rather than concurrently for many λ values. Replica exchange is not crucial here and does not noticeably improve the accuracy. The numbers of steps required to reach $\psi(\lambda)$ is shown in Fig. 1(b) to vary little for different ϵ—typically in the range of 400 to 600 steps. The rapid convergence that we observe away from the dynamical phase transition may be due to the fact that we employ overparameterized neural networks, which do not suffer from overfitting and converge to global minimizers when the loss function can be repeatedly sampled, a setting known as online learning [54,63,64].

To compare our algorithm with the cloning algorithm with feedback, we have applied the latter to the same model. In brief, the cloning method [29] evaluates the SCGF by simulating a batch of trajectories (clones) and by duplicating or eliminating trajectories according to weights computed from the trajectory ensemble. Generally, this population dynamics method requires a exponentially large number of replicas of the system as the desired event becomes rarer (or equivalently, as the magnitude of the noise decreases). To overcome this issue, [25] proposed a feedback approach, in which a controlling potential function is adaptively constructed to modify the original dynamics.

The convergence of this cloning algorithm with feedback with that of our algorithm for various batch or clone sizes and values of λ are compared in Fig. 2 in terms of computational time on a single machine measured in minutes. The results clearly show that our algorithm is significantly faster and more stable than the cloning algorithm, especially when the biasing parameter λ is far from 0. The difference in performance is partly due to the fact that the feedback in the cloning algorithm relies on estimating two probability distributions, which is limited by the relaxation time of the original system. Moreover, as λ deviates from 0, more iterations for updating the control potential are required to realize the rare events. In our algorithm, no distributions are estimated: the control drift is obtained directly by the taking the gradient of the estimated cost over a number of batches, which, contrary to the cloning algorithm, need not be stored in memory. Moreover, the results plotted in Fig. 2 show again that the number of steps needed to converge to the optimal drift does not vary much with λ.

There is another significant difference in that the control potential in the cloning algorithm is represented by a linear combination of a set of basis functions such as polynomials, so it requires *a priori* knowledge to choose the adequate basis

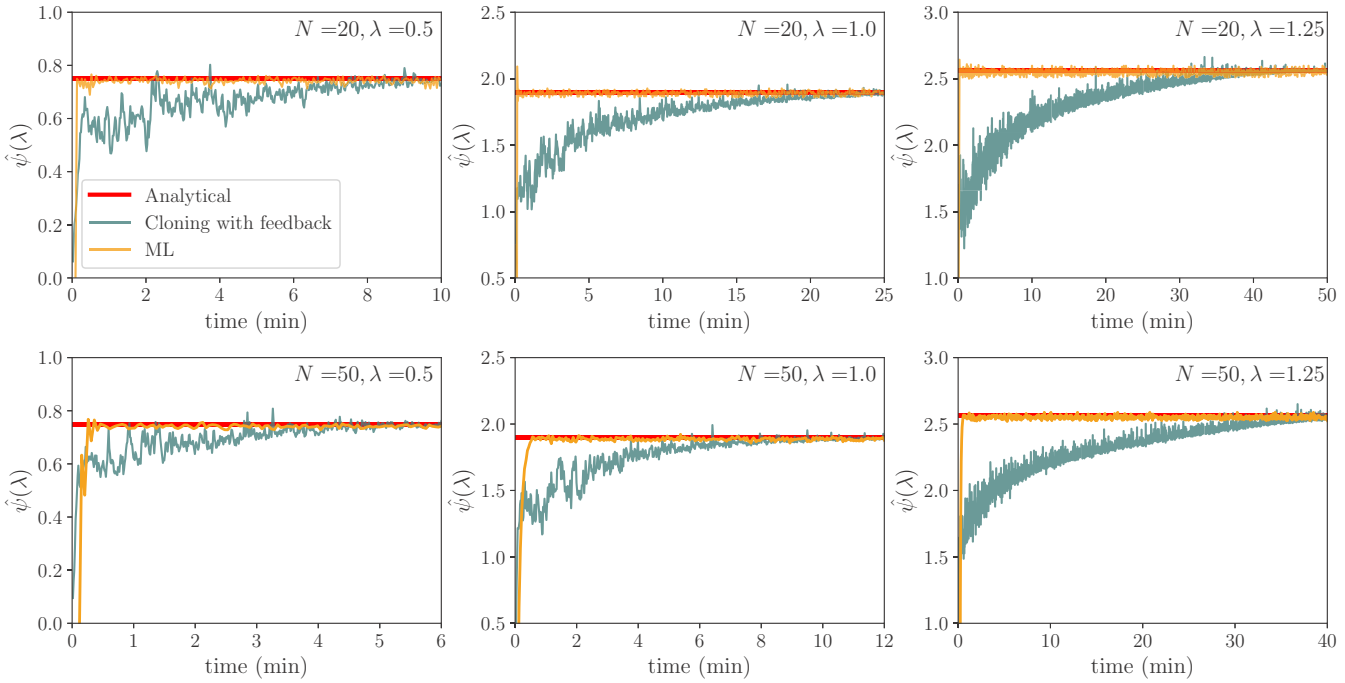

FIG. 2. Comparison of our machine learning approach with the cloning method with feedback. We computed the SCGF for the 1D diffusion system described as in (12) with $\epsilon = 0.01$. N is the batch size (for the ML approach) or the number of clones (for the cloning method). For the cloning method, we set the trajectory length to be $T = 0.3$ ($dt = 10^{-3}$), and updated the control potential every 75 steps so the time interval between updating is 22.5. All results were computed in the same personal computer, using CPU only.

functions, often in a case by case manner. By comparison, the machine learning approach that we propose is agnostic, meaning that it can be applied to a broad class of problems without any modifications of the algorithm or specific knowledge of the underlying structure of the problem [65]. Yet another advantage is that it has an inherently parallel structure and can evaluate the SCGF for multiple λ simultaneously. Cloning does not benefit from this parallel structure as the SCGF must be evaluated with a potential specific to each value of λ.

B. Active Brownian particles

Theoretical [66–68] and numerical [69] characterizations of active matter provide a compelling model for nonequilibrium phenomena. Minimal models, such as actively driven Brownian particles (ABPs) with purely repulsive interaction potentials, exhibit a rich spectrum of collective fluctuations and nonequilibrium phase separation emerging from the impact of persistent, directional motion on the local diffusivity of the constituent particles. The precise connection between energy dissipation and pattern formation in these nonequilibrium transitions remains a topic of intense research [70–72]. For example, the correlation between the structure formation in ABPs and fluctuations in entropy production was recently described by GrandPre *et al.* [73]. Probing the connection between rare dynamical behavior and collective fluctuations, however, is extremely challenging because the onset of clustering in ABPs requires large system sizes and high densities that can be accessed by cloning type algorithms only with a large number of replicas.

To test our algorithm, we consider the ABP model in which the motion of the ith particle is governed by the following equations:

$$d\boldsymbol{X}_t^{(i)} = \left[-\mu \frac{\partial U(\boldsymbol{X}_t)}{\partial \boldsymbol{x}^{(i)}} + v\boldsymbol{b}_t^{(i)} \right] dt + \sqrt{2D} d\boldsymbol{W}_t^{(i)},$$
$$\boldsymbol{b}^{(i)} = \left[\cos\phi_t^{(i)}, \sin\phi_t^{(i)} \right]^\top, \; d\phi_t^{(i)} = \sqrt{6D} dW_t^{\phi^{(i)}}. \quad (15)$$

The potential $U(\boldsymbol{X}_t)$ defining the conservative interparticle force is taken here to be a purely repulsive Weeks-Chandler-Andersen (WCA) pair potential that depends on the relative distance l_{ij} according to

$$U(l_{ij}) = \begin{cases} 4\epsilon[(\sigma/l_{ij})^{12} - (\sigma/l_{ij})^6] + \epsilon, & l_{ij} \leqslant 2^{1/6}\sigma, \\ 0, & l_{ij} > 2^{1/6}\sigma, \end{cases} \quad (16)$$

The nonconservative self-propulsion term $v\boldsymbol{b}^{(i)}$ represents the dissipative "active" force in which $\boldsymbol{b}_t^{(i)}$ are unit vectors that rotate diffusively and v is the magnitude of the active force. Finally, $\boldsymbol{W}_t^{(i)}$ and $W_t^{\phi^{(i)}}$ are independent standard Wiener processes representing noise sources for the state and angle. The simulations are performed with periodic boundary condition, and the relative distance matrix l_{ij} is adjusted by the minimum image convention. The unit of length is also normalized by σ and we set $\epsilon = 1$.

The phase separation properties of this model have been studied extensively [66]. When the Péclet number and the density of particles are high enough, the system exhibits a motility induced phase transition in which a macroscopic aggregate of particles forms. This transition has a natural dynamical correlate with the average entropy production

$$s = \frac{1}{NT} \sum_{i=1}^{N} \int_0^T v\boldsymbol{b}_t^{(i)} D^{-1} \circ d\boldsymbol{X}_t^{(i)}. \quad (17)$$

When the system enters the phase separated state, much of the directional motion also ceases, leading to a drop in the average entropy production compared to an unclustered trajectory. Indeed, several studies have pointed to entropy production being a natural observable for studying motility-induced phase separation [73] and nonequilibrium pattern formation in liquids [71,72], though a control-based approach has not been pursued for these systems to date.

Using our algorithm, we computed the many-body control forces associated with fluctuations of the entropy production for various particle numbers ($N = 40, 80, 200$). The results, shown in Fig. 3, were computed through the sequential training, since concurrent training requires a large total batch size which is computationally costly for high-dimensional systems. For this system, it is crucial that we do not include the direction of the active particles ϕ_t in the state, otherwise the entropy production rate can trivially be reduced by learning control forces antiparallel to the direction of the active force; this choice has a physical justification, namely, the directors are in equilibrium and are not reversed under time-reversal.

The simulations converge over relatively long times when first driving the system into the clustering phase; however, once we obtain a control force, they converge fast when sequentially altering λ. For $N = 40, 80$, we also noticed that replica exchange is required to obtain a convex SCGF, whereas for $N = 200$, replica exchange is not necessary. The replica exchange is implemented by concurrently training with multiple λ values and swapping trajectories between them. We set $\lambda_0 = -0.05$, with batch size 75 and 75, respectively. Then at each step all the 75 trajectories are attempted to be exchanged, as explained in Algorithm 1. In Fig. 4 we plot the results with and without replica exchange, respectively, in the $N = 40$ case. The results indicate that replica exchange is essential for obtaining a convex SCGF. We note that local convergence can be monitored due to the existence of a variational principle, but it is not possible to ensure that the convergence is global when the target rate function is not known; the same is true for cloning or other machine learning algorithms.

Going back to our results in Fig. 3 (see also the supplementary movie), we can see that particles start to aggregate when the biasing field λ is sufficiently negative. For all system sizes, the entropy production rate changes dramatically as a function of λ around a value coinciding with the onset of clustering. This sharp transition signifies a first-order dynamical phase transition in the entropy production rate, shown in Fig. 3(b), related to a singularity in the rate function at the transition point. Examining the learned controls provides further insight into the microscopic origins of the transition. As shown in the inset of Fig. 3(b), the learned control forces lead to net forces on the particles that favor the aggregated state.

The nonequilibrium fluctuations of active systems have been studied in a variety of contexts, using unbiased sampling [74], cloning [73], and reinforcement learning [33]. Our approach considerably simplifies the computation compared to reinforcement learning because we do not need to learn an expected value function. Moreover, unlike cloning, our approach scales to high-dimensional systems without incurring significant additional computational cost; training for various

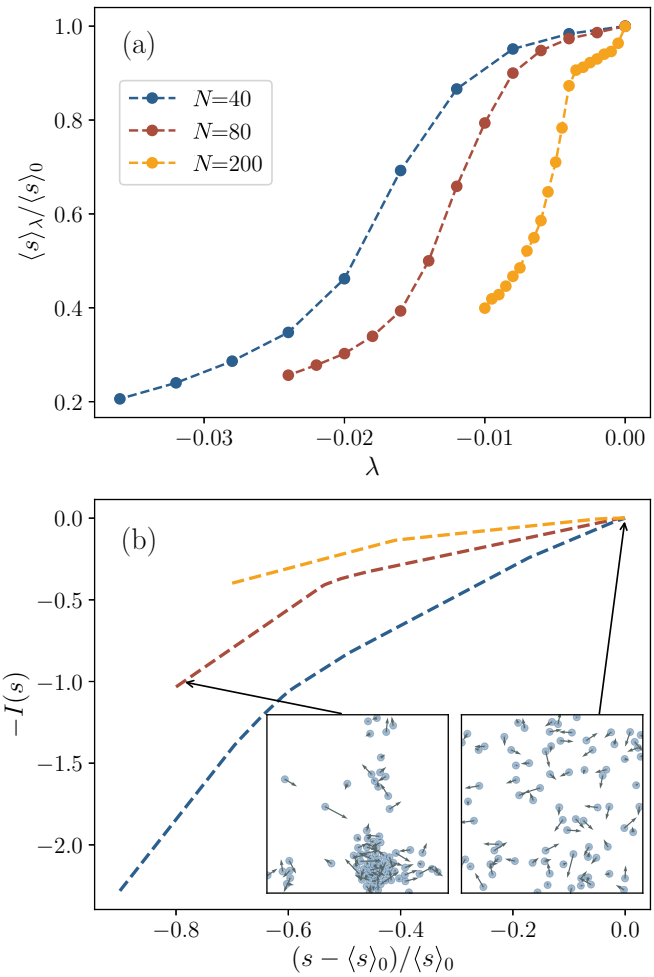

FIG. 3. Small entropy production indicates particle clustering. (a) The average entropy production at given λ for different system sizes: $N = 40$ (blue lines, left), 80 (red lines, middle), and 200 (yellow lines, right). (b) The corresponding rate functions ($N = 40$ bottom, $N = 80$ middle, $N = 200$ top). The inserted figures show snap shots of typical behaviors in the high entropy production phase ($\lambda = 0$) and low entropy production phase ($\lambda = -0.05$), respectively. The arrow represents the direction of motion. In this example the hidden layer dimension and number of layers of the neural network are 1000 and 6 respectively. The batch sizes for results of 40 and 80 particles are 75, and 20 for the 200 particle case. $T = 0.1$ and $dt = 10^{-4}$. The density of particles throughout all three cases is $\rho = N/L^2 = 0.1$ where L is the length of the simulation box. To avoid boundary effect, the input of the neural network is not the absolute position of particles but its relative position to a particular one, i.e., $u(\{\boldsymbol{X}^{(i)} - \boldsymbol{X}^{(0)}\})$ instead of $u(\{\boldsymbol{X}^i\})$. This step is essential otherwise the learned control force would force particles to the boundary.

λ is easily parallelizable and the integration of the trajectories can be carried out on heterogeneous hardware. Indeed, the cloning algorithm would require prohibitive computational resources compared with our algorithm.

V. DISCUSSION

The results presented in this paper demonstrate the efficacy of a machine learning algorithm that adaptively learns optimal

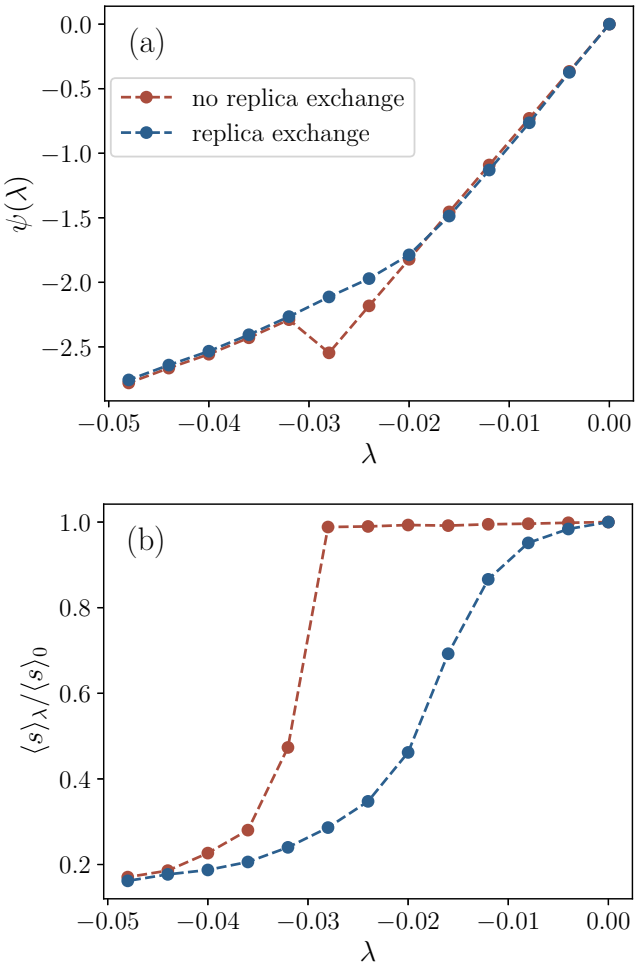

FIG. 4. The estimate of the SCGF and corresponding changes of average entropy production for $N = 40$ with/without replica exchange. The blue line (no kink) corresponds to the results with replica exchange and the red line shows the results without.

control forces to directly estimate large deviation functions for systems extremely challenging for conventional methods. The algorithm that we have proposed relies on direct stochastic optimization based on a small number of trajectories, which themselves may not need to have a long duration—a fact that requires further investigation. Importantly, the Lagrangian that we optimize is explicit and exact in the long time limit, requiring no additional approximation or optimization, as only the control function is learned. We have shown that the approach is robust both near dynamical phase transitions and in the limit of small noise. Like many methods based on machine learning, the method we propose shows favorable performance in high-dimensional systems and still identifies many-body control forces that realize the rare fluctuations defining dynamical phase transitions.

The examples we have explored here are continuous-time stochastic differential equations with a constant diffusion term (and hence additive noise), but it is straightforward to adapt our algorithm to other types of systems, including those with multiplicative noise, or with discrete, but innumerable state spaces such as unbounded Markov jump processes where directly evaluating the principal eigenvalue is not possible. Our approach could also be extended to finite-time large deviations, though we anticipate that this would require longer trajectories and therefore the adjoint state method would likely be mandatory. Learning control forces that drive the system locally, and hence can be transferred to systems of increasing size and complexity is among the most attractive possibilities for future investigation. For interacting particle systems, if the form of the input and the architecture of the neural network are carefully designed, it may be possible to obtain the optimal control force for systems with thousands of particles by training on smaller, more computationally tractable systems.

ACKNOWLEDGMENTS

G.M.R. acknowledges support from the Terman Faculty Fellowship. The authors also thank Eric Vanden-Eijnden and Suri Vaikuntanathan for thoughtful comments on this manuscript.

APPENDIX A: DERIVATION OF THE COST FUNCTIONAL

Using importance sampling, we write the expression for the SCGF as an expectation over a "tilted" or biased path measure,

$$\psi(\lambda) = \lim_{T\to\infty} \frac{1}{T} \log \int e^{\lambda T A_T[\boldsymbol{X}^u]} \frac{d\mathbb{P}[\boldsymbol{X}^u]}{d\mathbb{P}_u[\boldsymbol{X}^u]} d\mathbb{P}_u[\boldsymbol{X}^u]. \quad \text{(A1)}$$

This expectation must be estimated for each λ of interest by collecting trajectories from the controlled process (6). The relative statistical weight of the unperturbed path measure $\mathbb{P}$ to the path measure of the controlled process $\mathbb{P}_u$ is defined through the Radon–Nikodym derivative. In our case, using the parametrization $u(\boldsymbol{x}, \lambda) = b(\boldsymbol{x}) + \delta u(\boldsymbol{x}, \lambda)$, this derivative can be written explicitly using the Girsanov theorem [47] as

$$\frac{d\mathbb{P}[\boldsymbol{X}^u]}{d\mathbb{P}_u[\boldsymbol{X}^u]} \equiv M_T = \exp\left[-\int_0^T \sigma^{-1} \delta u(\boldsymbol{X}^u) dW_t - \frac{1}{2}\int_0^T \delta u\big(\boldsymbol{X}_t^u\big) D^{-1} \delta u\big(\boldsymbol{X}_t^u\big) dt \right], \quad \text{(A2)}$$

where we use the notation M_T to emphasize the fact that M_T is a martingale. The first integral in the exponential can be neglected when the deterministic contribution is finite and we are left with an expression for Eq. (A1):

$$\psi(\lambda) = \lim_{T\to\infty} \frac{1}{T} \log \mathbb{E}_{\boldsymbol{X}^u} \exp\left[\lambda T A_T[\boldsymbol{X}^u] - \frac{1}{2}\int_0^T \delta u\big(\boldsymbol{X}_t^u\big) D^{-1} \delta u\big(\boldsymbol{X}_t^u\big) dt \right]. \quad \text{(A3)}$$

The term inside the exponential is evidently time-extensive and, in the limit $T \to \infty$, the integral will be dominated by a saddle point, following the Laplace approximation. As a result, we obtain

$$\psi(\lambda) = \lim_{T\to\infty} \frac{1}{T} \sup_{\delta u} \mathbb{E}_{\boldsymbol{X}^u} \{ \lambda T A_T[\boldsymbol{X}^u] - \frac{1}{2}\int_0^T \delta u\big(\boldsymbol{X}_t^u\big) D^{-1} \delta u\big(\boldsymbol{X}_t^u\big) dt \}. \quad \text{(A4)}$$

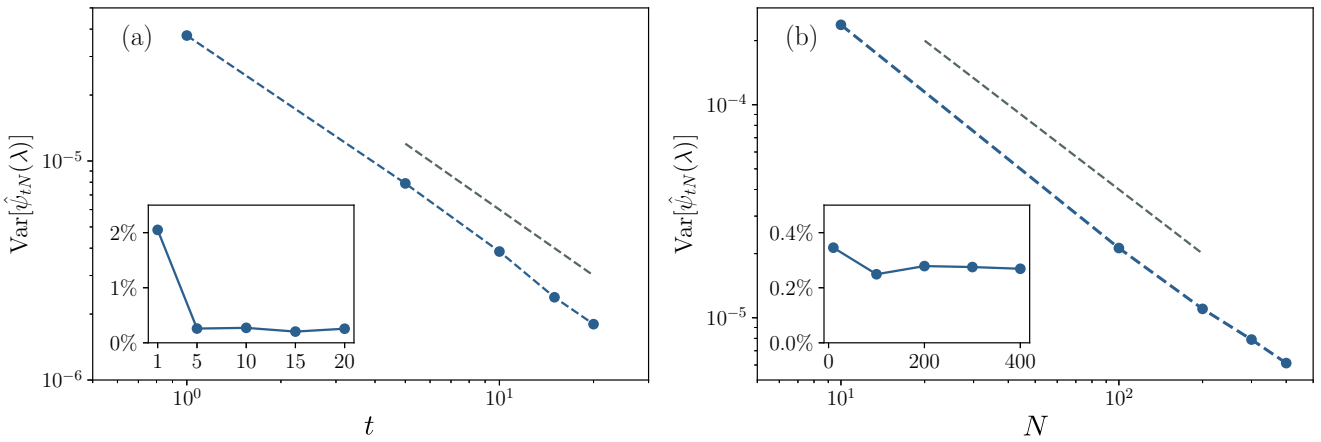

FIG. 5. Variance of the estimator. We illustrate the scaling property of the variance of our estimator (B1) using the small noise example. By fixing $\lambda = 1$ and $\epsilon = 0.01$, the neural network is trained with (a) a fixed batch size $N = 300$ with various trajectory length t from 1 to 20, or (b) a fixed $t = 5$ and various batch size from 10 to 400. The neural networks in all cases are trained for more than 400 steps, and the variance is estimated by collecting the data from the last 100 steps. The insert figures show the relative absolute error $|\hat{\psi}(\lambda) - \psi(\lambda)|/\psi(\lambda)$. The gray dashed line represents a -1 slope.

Hence, the argument of the supremum becomes a natural variational objective for δu, which we denote

$$\mathcal{L}[\boldsymbol{X}^u, u] = \lambda \int_0^T f(\boldsymbol{X}_t^u)dt + g(\boldsymbol{X}_t^u) \circ d\boldsymbol{X}_t^u - \frac{1}{2}\int_0^T \delta u D^{-1} \delta u(\boldsymbol{X}_t^u)dt. \tag{A5}$$

APPENDIX B: COST ESTIMATOR

We compute the cost functional numerically by simulating N independent trajectories $\boldsymbol{X}_{t,i}^u$ of the controlled process, referred to as replica, over a finite time window or horizon $[0, t]$ by using the estimator

$$\hat{\psi}_{Nt}(\lambda) = \frac{1}{Nt}\sum_{i=1}^N \hat{\psi}_{t,i}(\lambda), \tag{B1}$$

where

$$\hat{\psi}_{t,i}(\lambda) = \frac{1}{t}\left\{\lambda \int_0^t f(\boldsymbol{X}_{s,i}^u)ds + g(\boldsymbol{X}_{s,i}^u) \circ d\boldsymbol{X}_{s,i}^u - \frac{1}{2}\int_0^t \delta u(\boldsymbol{X}_{s,i}^u) D^{-1} \delta u(\boldsymbol{X}_{s,i}^u)ds\right\} \tag{B2}$$

is the estimator of the cost functional for one replica. By the ergodic theorem and the law of large numbers, $\hat{\psi}_{Nt}(\lambda)$ converges to the SCGF $\psi(\lambda)$ in the double limit $t \to \infty$ and $N \to \infty$, provided that u is the optimal control drift u^*. Note however that, since u^* is time-independent, we can obtain the long-time limit of the optimal cost by considering a finite-time estimator provided that we take the limit $N \to \infty$, so that there is only one limit to consider.

This point is illustrated for the 1D diffusion in the low-noise limit in Fig. 5, which shows the mean square error (MSE) of the loss estimator or, equivalently, its variance since it is unbiased:

$$\mathrm{MSE} = \mathbb{E}[\hat{\psi}_{Nt}(\lambda) - \psi(\lambda)]^2 = \mathrm{Var}[\hat{\psi}_{Nt}(\lambda)]. \tag{B3}$$

Since the N replica are independent, the variance of $\hat{\psi}_{Nt,i}(\lambda)$ must scale with N^{-1} due to the central limit theorem, yielding $\mathrm{MSE} = \mathrm{Var}[\hat{\psi}_{t,i}(\lambda)]/N$. In general, $\hat{\psi}_{t,i}(\lambda)$ itself is a time-extensive variable that satisfies a large deviation principle, so its variance $\mathrm{Var}[\hat{\psi}_{t,i}(\lambda)]$ scales with t^{-1}. Therefore, overall, the MSE of the loss estimator decreases as $(tN)^{-1}$. Hence, a large-batch and short-time estimator is equivalent to a small-batch and long-time estimator. In Fig. 5, we confirm this scaling property by plotting the estimator of $\psi_\epsilon(\lambda)$ for the simple diffusion model as a function of integration time or batch size.

APPENDIX C: ADJOINT STATE METHOD

The adjoint state method for Stratonovich SDEs differs only marginally from the classical adjoint method for ODEs, though we note that the method can be extended to multiplicative noise [38]. These methods require forward and backward integration of the differential equation and, in the stochastic case, one must solve the SDE backward in time with the same Wiener process $\boldsymbol{W}_t$ used in the forward direction, meaning that the noise history must be stored. We explain the method for the ODE case and refer to Ref. [38] for further details.

Consider the ODE

$$\frac{dx}{dt} = u(x, t, \theta), \quad x(0) = x_0, \tag{C1}$$

and some objective function $\mathcal{L}[x(T)]$, which we would like to minimize with respect to θ. We note that $\mathcal{L}$ depends on θ through the dynamics because

$$x(T) = x_0 + \int_0^T u(x, t, \theta)dt. \tag{C2}$$

The dependence of $\mathcal{L}$ on θ can be computed using classical sensitivity analysis techniques. Assuming that we can easily evaluate the cost functional at the final integration time T, we need to compute

$$\frac{\partial \mathcal{L}[x(T)]}{\partial \theta} = \frac{\partial \mathcal{L}[x(T)]}{\partial x(T)}\frac{\partial x(T)}{\partial \theta}, \tag{C3}$$

where $x(t)$ is constrained to follow the dynamics (C1). Using the method of Lagrange multipliers, we can turn this into an unconstrained optimization where the time-dependent multiplier $\mathfrak{A}(t)$ is chosen to impose the constraint $\dot{x} = u$. That is, the cost functional becomes

$$\tilde{\mathcal{L}}[x(T)] = \mathcal{L}[x(T)] - \int_0^T \mathfrak{A}(t)[\dot{x} - u(x,\theta,t)]dt, \quad \text{(C4)}$$

so that

$$\begin{aligned}\frac{\partial\tilde{\mathcal{L}}[x(T)]}{\partial\theta} &= \frac{\partial\mathcal{L}[x(T)]}{\partial x(T)}\frac{\partial x(T)}{\partial\theta} - \mathfrak{A}(T)\frac{\partial x(T)}{\partial\theta} \\ &\quad - \frac{\partial}{\partial\theta}\int_0^T \dot{\mathfrak{A}}(t)[x(t) - u(x,\theta,t)]dt \\ &= \frac{\partial\mathcal{L}[x(T)]}{\partial x(T)}\frac{\partial x(T)}{\partial\theta} - \mathfrak{A}(T)\frac{\partial x(T)}{\partial\theta} \\ &\quad + \int_0^T \dot{\mathfrak{A}}(t)\frac{\partial x(t)}{\partial\theta} + \mathfrak{A}(t)\frac{\partial u(x,\theta,t)}{\partial x(t)}\frac{\partial x(t)}{\partial\theta} \\ &\quad + \mathfrak{A}(t)\frac{\partial u(x,\theta,t)}{\partial\theta}dt. \quad \text{(C5)}\end{aligned}$$

From this result, we then choose $\mathfrak{A}$ so that

$$\dot{\mathfrak{A}}(t) = -\mathfrak{A}(t)\frac{\partial u(x,\theta,t)}{\partial x(t)}, \quad \mathfrak{A}(T) = \frac{\partial\mathcal{L}[x(T)]}{\partial x(T)}, \quad \text{(C6)}$$

to write the gradient as

$$\frac{\partial\mathcal{L}[x(T)]}{\partial\theta} = -\int_T^0 \mathfrak{A}(t)\frac{\partial u(x,\theta,t)}{\partial\theta}dt, \quad \text{(C7)}$$

which is solved backward in time because we know the final condition for the adjoint $\mathfrak{A}(T)$.

The stochastic variant of this algorithm is operationally similar to the procedure outlined above and is particularly straightforward for Stratonovich SDEs (the convention we use in numerical experiments with currentlike observables) [38].

[1] R. van Zon, S. Ciliberto, and E. G. D. Cohen, Phys. Rev. Lett. **92**, 130601 (2004).
[2] S. Ciliberto, S. Joubaud, and A. Petrosyan, J. Stat. Mech. (2010) P12003.
[3] S. Ciliberto, Phys. Rev. X **7**, 021051 (2017).
[4] M. Merolle, J. P. Garrahan, and D. Chandler, Proc. Natl. Acad. Sci. USA **102**, 10837 (2005).
[5] J. P. Garrahan, R. L. Jack, V. Lecomte, E. Pitard, K. van Duijvendijk, and F. van Wijland, Phys. Rev. Lett. **98**, 195702 (2007).
[6] L. O. Hedges, R. L. Jack, J. P. Garrahan, and D. Chandler, Science **323**, 1309 (2009).
[7] D. Chandler and J. P. Garrahan, Annu. Rev. Phys. Chem. **61**, 191 (2010).
[8] B. Derrida, J. Stat. Mech. (2007) P07023.
[9] L. Bertini, A. D. Sole, D. Gabrielli, G. Jona-Lasinio, and C. Landim, J. Stat. Mech. (2007) P07014.
[10] L. Bertini, A. De Sole, D. Gabrielli, G. Jona-Lasinio, and C. Landim, Rev. Mod. Phys. **87**, 593 (2015).
[11] F. Cagnetta, F. Corberi, G. Gonnella, and A. Suma, Phys. Rev. Lett. **119**, 158002 (2017).
[12] T. GrandPre and D. T. Limmer, Phys. Rev. E **98**, 060601(R) (2018).
[13] S. Whitelam, K. Klymko, and D. Mandal, J. Chem. Phys. **148**, 154902 (2018).
[14] Y.-E. Keta, E. Fodor, F. van Wijland, M. E. Cates, and R. L. Jack, Phys. Rev. E **103**, 022603 (2021).
[15] G. E. Crooks, Phys. Rev. E **60**, 2721 (1999).
[16] J. L. Lebowitz and H. Spohn, J. Stat. Phys. **95**, 333 (1999).
[17] R. J. Harris and G. M. Schütz, J. Stat. Mech. (2007) P07020.
[18] P. Pietzonka, A. C. Barato, and U. Seifert, Phys. Rev. E **93**, 052145 (2016).
[19] A. C. Barato and U. Seifert, Phys. Rev. E **92**, 032127 (2015).
[20] T. R. Gingrich, J. M. Horowitz, N. Perunov, and J. L. England, Phys. Rev. Lett. **116**, 120601 (2016).
[21] B. Derrida and J. L. Lebowitz, Phys. Rev. Lett. **80**, 209 (1998).
[22] A. Lazarescu, Exact large deviations of the current in the asymmetric simple exclusion process with open boundaries, Ph.D. thesis, Institut de Physique Théorique, CEA-Saclay, 2015.
[23] T. Bodineau and B. Derrida, J. Stat. Phys. **123**, 277 (2006).
[24] J. A. Bucklew, *Introduction to Rare Event Simulation* (Springer, New York, 2004).
[25] T. Nemoto, F. Bouchet, R. L. Jack, and V. Lecomte, Phys. Rev. E **93**, 062123 (2016).
[26] U. Ray, G. K.-L. Chan, and D. T. Limmer, Phys. Rev. Lett. **120**, 210602 (2018).
[27] G. Ferré and H. Touchette, J. Stat. Phys. **172**, 1525 (2018).
[28] P. Grassberger, Comput. Phys. Commun. **147**, 64 (2002).
[29] C. Giardinà, J. Kurchan, and L. Peliti, Phys. Rev. Lett. **96**, 120603 (2006).
[30] V. Lecomte and J. Tailleur, J. Stat. Mech. (2007) P03004.
[31] U. Ray and G. K.-L. Chan, J. Chem. Phys. **152**, 104107 (2020).
[32] S. Whitelam, D. Jacobson, and I. Tamblyn, J. Chem. Phys. **153**, 044113 (2020).
[33] A. Das and D. T. Limmer, J. Chem. Phys. **151**, 244123 (2019).
[34] D. C. Rose, J. F. Mair, and J. P. Garrahan, New J. Phys. **23**, 013013 (2021).
[35] A. Das, D. C. Rose, J. P. Garrahan, and D. T. Limmer, J. Chem. Phys. **155**, 134105 (2021).
[36] W. E and B. Yu, Commun. Math. Stat. **6**, 1 (2018).
[37] G. M. Rotskoff, A. R. Mitchell, and E. Vanden-Eijnden, in *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference* (Lausanne, Switzerland, 2021).
[38] X. Li, T.-K. L. Wong, R. T. Q. Chen, and D. Duvenaud, in *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research (PMLR, Palermo, Sicily, Italy, 2020), Vol. 108, pp. 3870–3882.
[39] H. Touchette, Physica A **504**, 5 (2018).
[40] U. Seifert, Rep. Prog. Phys. **75**, 126001 (2012).
[41] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*, 2nd ed. (Springer, New York, 1998).

[42] R. V. Kohn, M. G. Reznikoff, and E. Vanden-Eijnden, J. Nonlin. Sci. **15**, 223 (2005).
[43] T. Speck, A. Engel, and U. Seifert, J. Stat. Mech. (2012) P12001.
[44] É. Fodor, R. L. Jack, and M. E. Cates, Annu. Rev. Condens. Matter Phys. **13**, 1 (2022).
[45] H. Touchette, Phys. Rep. **478**, 1 (2009).
[46] C. Casert, T. Vieijra, S. Whitelam, and I. Tamblyn, Phys. Rev. Lett. **127**, 120602 (2021).
[47] B. Oksendal, *Stochastic Differential Equations: An Introduction with Applications*, Universitext (Springer, Berlin, 1992).
[48] R. Chetrite and H. Touchette, J. Stat. Mech. (2015) P12001.
[49] Alternatively, the optimal control drift can be obtained by contracting rate function characterizing the joint fluctuations of the empirical density and empirical current, commonly known as the "level-2.5" large deviation function.
[50] R. Chetrite and H. Touchette, Ann. Henri Poincaré **16**, 2005 (2015).
[51] R. L. Jack and P. Sollich, Eur. Phys. J.: Spec. Top. **224**, 2351 (2015).
[52] T. H. E. Oakes, A. Moss, and J. P. Garrahan, Mach. Learn. Sci. Tech. **1**, 035004 (2020).
[53] G. Rotskoff and E. Vanden-Eijnden, in *Advances in Neural Information Processing Systems 31*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Red Hook, NY, 2018), pp. 7146–7155.
[54] L. Chizat and F. Bach, in *Advances in Neural Information Processing Systems 31*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Red Hook, NY, 2018), pp. 3036–3046.
[55] S. Mei, A. Montanari, and P.-M. Nguyen, Proc. Natl. Acad. Sci. USA **115**, E7665 (2018).
[56] J. Sirignano and K. Spiliopoulos, SIAM J. Appl. Math. **80**, 725 (2020).
[57] A. R. Barron, IEEE Trans. Inf. Theory **39**, 930 (1993).
[58] G. Cybenko, Math. Control Signal Systems **2**, 303 (1989).
[59] The Python source code is available online at github.com/quark-strange/machine_learning_LDP.
[60] B. Tzen and M. Raginsky, arXiv:1905.09883 (2019).
[61] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, arXiv:2010.08895 (2021).
[62] D. Frenkel and B. Smit, in *Understanding Molecular Simulation*, edited by D. Frenkel and B. Smit (Academic Press, San Diego, CA, 2002).
[63] G. M. Rotskoff and E. Vanden-Eijnden, arXiv:1805.00915 (2018).
[64] M. Belkin, D. Hsu, S. Ma, and S. Mandal, Proc. Natl. Acad. Sci. USA **116**, 15849 (2019).
[65] While it is not the case for this simple example, it may be possible for the cloning method to outperform the machine learning approach if the basis is expertly tailored to the problem at hand.
[66] M. E. Cates and J. Tailleur, Annu. Rev. Condens. Matter Phys. **6**, 219 (2015).
[67] T. Speck, J. Bialké, A. M. Menzel, and H. Löwen, Phys. Rev. Lett. **112**, 218304 (2014).
[68] P. Pietzonka, K. Kleinbeck, and U. Seifert, New J. Phys. **18**, 052001 (2016).
[69] G. S. Redner, M. F. Hagan, and A. Baskaran, Phys. Rev. Lett. **110**, 055701 (2013).
[70] M. Nguyen and S. Vaikuntanathan, Proc. Natl. Acad. Sci. USA **113**, 14231 (2016).
[71] L. Tociu, E. Fodor, T. Nemoto, and S. Vaikuntanathan, Phys. Rev. X **9**, 041026 (2019).
[72] É. Fodor, T. Nemoto, and S. Vaikuntanathan, New J. Phys. **22**, 013052 (2020).
[73] T. GrandPre, K. Klymko, K. K. Mandadapu, and D. T. Limmer, Phys. Rev. E **103**, 012613 (2021).
[74] L. Caprini, A. Puglisi, and A. Sarracino, Symmetry **13**, 81 (2021).