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, CA 94305, USA*
[2] *Department of Mathematical Sciences, Stellenbosch University, Stellenbosch 7600, South Africa*

(Dated: July 8, 2021)

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.

Techniques from large deviation theory have provided physical insight into both 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 that are notoriously difficult to compute for complex and high-dimensional systems. While analytical treatment is possible in some systems [21–23], we must generally make numerical estimates of these functions. 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 algorithms [32–34]. With most methods, it remains challenging however to obtain good convergence in systems with many degrees of freedom, especially when probing fluctuations near phase transitions [35].

Here we present an approach that combines importance sampling, path-space Monte Carlo methods, and, crucially, the robust and flexible function representations offered by neural networks to calculate large deviation functions. The approach that we describe combines control theory with recent developments using machine learning to solve high-dimensional variational PDEs [36, 37] to adaptively construct 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 in which gradients are computed through the dynamics or via an adjoint stochastic dynamics, which is robust even over long trajectories [38]. Results obtained for two models, including a model of active Brownian particles, show that our approach i) efficiently scales to large, interacting particle systems which are difficult to treat with spectral methods or cloning algorithms, ii) is robust near dynamical phase transitions, and iii) does not slow down in the low-noise limit.

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, which is multiplied by the noise matrix σ. This model captures the diffusive dynamics of many physical systems; b could be comprised of the gradient of a many-body potential energy describing the interactions among a large number of particles in addition to non-conservative and hence nonequilibrium external forces. For simplicity, we assume that σ is independent of $\boldsymbol{x}$ and that the corresponding diffusion tensor $D = \sigma\sigma^T$ is invertible. Moreover, we assume that $\boldsymbol{X}_t$ is ergodic, which means that it has a unique probability stationary density, reached from any initial distribution in the long-time limit.

While there is no canonical form for the stationary density when the drift or "force" $b(\boldsymbol{x})$ is non-conservative, the probability density of a large class of time-extensive observables A_T computed along trajectories is known to satisfy a large deviation form and can thus be characterized in a general way by a so-called rate function, which can be seen as a nonequilibrium analog of the entropy function. These "dynamical" observables typically considered for diffusive systems have the form

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

and can represent many different physical quantities depending on the choice of the function $f : \mathbb{R}^d \to \mathbb{R}$, connected with "density-like" observables, and $g : \mathbb{R}^d \to \mathbb{R}^d$, connected with "current-like" observables. In this setting, the probability density $\rho_T(a)$ that A_T realizes some fixed value $a \in \mathbb{R}$ 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 and $I : \mathbb{R} \to \mathbb{R}$ is the rate function [39]. In most cases, this 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

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

where $\mathbb{E}_{\boldsymbol{X}}$ denotes an expectation over (1) and λ is a real parameter conjugate to A_T. The SCGF itself can, in numerically tractable cases, be computed using spectral methods, as it corresponds to the dominant eigenvalue of a linear operator, called the tilted generator [40].

Computing rate functions has become a central problem in statistical physics as they provide a lens into the phase behavior and symmetries of nonequilibrium systems [41, 42]. However, as is the case with the equilibrium entropy and the free energy, computing rate functions is a difficult task, especially when dealing with complex and high-dimensional systems, as it relies on sampling exponentially rare events or, in the case of the SCGF, on solving a high-dimensional, non-Hermitian spectral problem. Many strategies have been deployed recently to address these problems, including ones based the power method [27], diffusion Monte Carlo [26, 28–31], and reinforcement learning algorithms [32–34].

Any estimate of large deviations necessitates computing the probability of extremely rare events, which do not occur spontaneously on timescales accessible to simulation and hence require importance sampling. With an appropriate change of measure, the SCGF (4) can be estimated by instead evaluating the tilted expectation

$$\psi(\lambda) = \lim_{T\to\infty} \frac{1}{T} \log \mathbb{E}_{\boldsymbol{X}^u} \left(e^{\lambda T A_T} \frac{d\mathbb{P}[\boldsymbol{X}^u]}{d\mathbb{P}_u[\boldsymbol{X}^u]} \right) \tag{5}$$

where $\boldsymbol{X}^u$ denotes a process controlled by a drift $u \neq b$. The Radon-Nikodym derivative measures the relative path weight in the unperturbed ensemble with the biased ensemble—this quantity can be computed explicitly along a given path using the Girsanov theorem. While this "tilting" of the path leads to an exact rewriting of (4), the change of measure can reduce the variance of the expectation and potentially aid convergence of an estimator.

This change of measure can be implemented as a modified dynamics that satisfies the SDE

$$d\boldsymbol{X}^u_t = u_t(\boldsymbol{X}^u_t)dt + \sigma d\boldsymbol{W}_t, \tag{6}$$

in which the drift $b(x)$ is replaced by the control drift $u_t(x)$. This drift is chosen in such a way as to maximize the cost function or Lagrangian

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

which we derive in Appendix A. This explicit objective function offers a route to direct optimization of the control forces without reinforcement learning. In the limit $T \to \infty$, it can be shown [43] that the maximizing control drift is time-independent and that the maximum is the SCGF, so that

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

The expression (8) is a variational objective and, as such, is amenable to Ritz-type methods that optimize a parametric representation $u(\boldsymbol{x}, \lambda; \theta)$ with respect to some set of variational parameters θ. The solution u^* of the variational problem of maximizing (7) over control drifts can be interpreted, as shown in Appendix A, as the optimal change of process in importance sampling that yields the SCGF: the first integral in the Lagrangian (7) enforces the constraint $A_T = a$—realizing the target rare event—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 unperturbed process [44]. Directly carrying out this optimization is nontrivial, as it requires representing a potentially complex, many-body force, which has motivated several sophisticated strategies that rely on intricate basis functions, Malliavin weight sampling, and reinforcement learning [32–34, 45].

Here, we solve the optimization problem directly using simulated trajectories of the controlled process by representing the control drift with deep neural networks, which are well-suited to this task [46–51] due to their robust function approximation properties, even in high-dimensional settings. To compute the necessary gradients, we differentiate through the solution of the SDE (6) using recent developments in the machine learning literature [38, 52]. Over short times, we use direct backpropagation of the dynamics through a Stratonovich time-discretization of the SDE to compute $\nabla_\theta \mathcal{L}$. The computational graph that contains all the gradient information consumes significant memory resources in this case, so over longer time scales, we calculate $\nabla_\theta \mathcal{L}$ by solving instead an adjoint SDE, detailed in Appendix B 1. This method is stable and only requires that we keep the noise history and solve the SDE backward in time. Solving the adjoint SDE adds computational cost, but is not prohibitive even for large systems.

We discuss the details of our optimization algorithm and the exact representations of the networks that we use, which are inspired by pioneering work on the deep Ritz method [36], in Appendix B. The estimator of the cost function that we use, which involves a collection or "batch" of N trajectories $\{\boldsymbol{X}^u_{[0,t],i}\}_{i=1}^N$, is presented in Appendix C. There we discuss the variance of the esti-

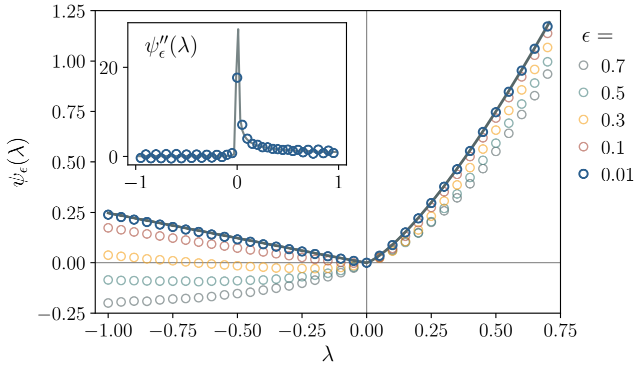

Figure 1. The SCGF for Eq. (9) for decreasing temperatures ϵ. The solid line represents the analytical exact solutions at zero-noise limit: $\psi_{\epsilon\to 0}(\lambda) = \max_q\{\lambda(q^2+q) - q^6/4\}$, and inserted figure shows the second derivative of the SCGF of $\epsilon = 0.01$, confirming a second order dynamical phase transition.

mator and show in numerical examples that short time trajectories suffice when the batch size is large.

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_n from a set $\{\lambda_1, \lambda_2, \cdots, \lambda_N\}$. We find that the expressiveness of the neural networks we use allows a single force function $u(\cdot, \lambda)$ to capture the control forces, 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 [53], shows fast convergence.

In practice, we have found that sequential training is better than concurrent training when dealing with dynamical phase transitions, which lead to rapid changes of the optimal control forces as a function of λ. In this case, we can increase the likelihood of sampling trajectories in different phases, and therefore increase the accuracy of the estimated SCGF, by employing a path space variant of the replica exchange method [54], in which two trajectories corresponding to different λ are swapped according to a Metropolis-Hastings using the action functional in place of an energy (see Appendix B).

To test our method, we first consider a 1D diffusion in a quartic potential

$$dX_t = -X_t^3 dt + \sqrt{2\epsilon} dW_t, \tag{9}$$

and focus on the observable

$$A_T = \frac{1}{T}\int_0^T X_t(X_t+1)dt. \tag{10}$$

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 $\epsilon \to 0$ [35]. 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 [35].

The low-noise limit is not a bottleneck in our approach. Using short-time trajectories, we concurrently trained a single neural network with a set of λ in the range of $[-1, 1]$—the numerical results for various ϵ are plotted in Fig. 1 (see Appendix B for numerical details). For this system, direct backpropagation and the adjoint state approach give indistinguishable results. Replica exchange is not required to obtain good agreement with the exact result even near the phase transition, this may be due to the fact that the transition occurs at $\lambda = 0$, where no control force is needed. Our numerical results at $\epsilon = 0.01$ agree exceptionally well with the exact result in the zero-noise limit. In addition, the exact result at $\epsilon \to 0$ enables us to analyze the estimated error. We found the normalized mean squared error of our estimation (averaged over the 40 points except $\lambda = 0$ in Fig. 1) is about 0.2%. This error can be further reduced by training the network at a single λ. Rapid convergence away from dynamical phase transitions may be due to the fact that we employ overparameterized neural networks which do not suffer from overfitting and converge to global minimizers in settings where the loss function can be repeatedly sampled, a setting known as online learning [47, 55, 56]. The results here demonstrate the efficacy of our algorithm for systems with small noise, we next turn to study the high dimensional interacting particle system.

Theoretical [57–59] and numerical [60] characterizations of active matter provide a compelling model for nonequilibrium phenomena. Minimal models, for example, actively driven Brownian particles with purely repulsive WCA interaction potentials (ABPs) exhibit a rich spectrum of collective fluctuations, leading to nonequilibrium phase separation. This motility induced phase separation emerges 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 [61–63]; for example, the correlation between the structure formation in ABPs and fluctuations in entropy production was recently described by GrandPre *et al.* [64]. 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 and hence necessitates an exponentially large number of replicas for cloning type algorithms.

We examined our approach in the context of ABPs, a model in which the motion of the ith particle is governed

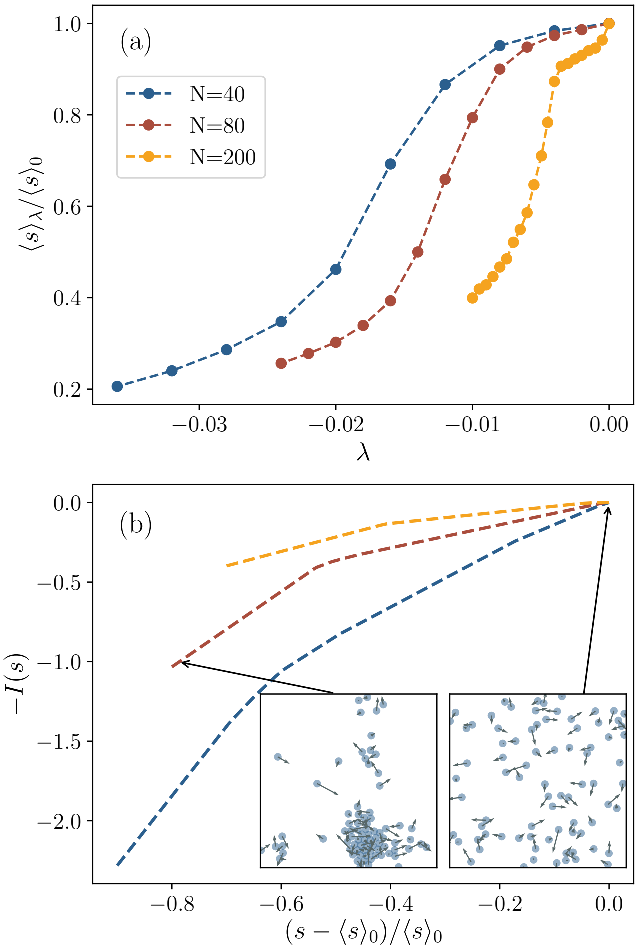

Figure 2. Small entropy production indicates particle clustering. (a): The average entropy production at given λ for different system sizes: $N = 40$ (blue lines), 80 (red lines), and 200 (yellow lines). (b): The corresponding rate functions. The inserted figures show snap shots of typical behaviors in the high entropy production phase ($\lambda = 0$) and low entropy production phase ($\lambda = -0.05$), respectively. The arrow represents the direction of motion.

by the following equation,

$$\begin{aligned} d\boldsymbol{X}_t^{(i)} &= [-\mu \frac{\partial U(\boldsymbol{X}_t)}{\partial \boldsymbol{x}^{(i)}} + v\boldsymbol{b}_t^{(i)}]dt + \sqrt{2D_t} d\boldsymbol{W}_t^{(i)}, \\ \boldsymbol{b}^{(i)} &= [\cos\phi_t^{(i)}, \sin\phi_t^{(i)}]^\top, \quad d\phi_t^{(i)} = \sqrt{6D_t} d\boldsymbol{W}_t^{\phi^{(i)}}. \end{aligned} \tag{11}$$

The potential $U(\boldsymbol{X}_t)$, a purely repulsive WCA pair potential that depends on the positions of all particles, constitutes the conservative interparticle force. The nonconservative self-propulsion term $v\boldsymbol{b}^{(i)}$ represents the dissipative "active" force. In the ABP model, $\boldsymbol{b}_t^{(i)}$ are unit vectors which rotate diffusively and v is the magnitude of the active force. Here, $\boldsymbol{W}_t^{(i)}$ and $\boldsymbol{W}_t^{\phi^{(i)}}$ are independent standard Wiener processes. The phenomenology of motility induced phase separation has been studied extensively (cf. [57]); at a high level, when the Péclet number and the density of particles are high enough, the system will exhibit 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

$$s = \frac{1}{NT} \sum_{i=1}^{N} \int_0^T v\boldsymbol{b}_t^{(i)} D_t^{-1} \circ d\boldsymbol{X}_t^{(i)}. \tag{12}$$

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 as a natural dynamical observable for motility induced phase separation [64] and nonequilibrium pattern formation in liquids [62, 63], though a control-based approach has not been pursued on these systems to date. We computed many-body control forces for this system using Alg. 1, for a variety of system sizes (N=40,80,200). For this system, it is crucial that we do not include the director of the active particles ϕ_t in the state, otherwise the entropy production rate can trivially be reduced by learning control forces anti-parallel 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.

When the biasing field λ is sufficiently negative, particles start to aggregate as shown in Fig. 2 and the supplementary movie. For all system sizes, the entropy production rate changes dramatically as a function of λ, which coincides with the onset of clustering. This sharp transition signifies a dynamical phase transition in the entropy production rate, as shown in Fig. 2(b), where we observe a singularity in the rate function at the transition point. The numerical results emphasize that the method captures the dynamical phase transition in the entropy production, but examining the learned controls provides further insight into the microscopic origins of the transition. As shown in the inset of Fig. 2 (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 [32, 64, 65], using unbiased sampling, cloning, and reinforcement learning. The approach we take considerably simplifies the computation compared to reinforcement learning because we do not need to learn an expected value function. Moreover, unlike cloning, the algorithm considered here scales to high-dimensional systems without incurring significant additional computational cost; training for various λ is easily parallelizable and the integration of the trajectories can be carried out on heterogeneous hardware.

Taken together, the results here demonstrate the efficacy of a machine learning algorithm that adaptively learns optimal control forces to directly estimate large deviation functions for systems extremely challenging for conventional methods. The algorithm 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—only the control function is learned. We show the approach is robust both near the 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 explore 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. This approach could 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.

[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 (2010).

[3] S. Ciliberto, Phys. Rev. X **7**, 021051 (2017).

[4] M. Merolle, J. P. Garrahan, and D. Chandler, Proc. Nat. 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, Ann. Rev. Chem. Phys. **61**, 191 (2010).

[8] B. Derrida, J. Stat. Mech. **2007**, P07023 (2007).

[9] L. Bertini, A. D. Sole, D. Gabrielli, G. Jona-Lasinio, and C. Landim, J. Stat. Mech. **2007**, P07014 (2007).

[10] L. Bertini, A. D. 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 (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 (2007).

[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*, PhD 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 (2017).

[27] G. Ferré and H. Touchette, J. Stat. Phys. **172**, 1525 (2018).

[28] P. Grassberger, Comp. Phys. Comm. **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 (2007).

[31] U. Ray and G. K.-L. Chan, J. Chem. Phys. **152**, 104107 (2020).

[32] A. Das and D. T. Limmer, J. Chem. Phys. **151**, 244123 (2019).

[33] D. C. Rose, J. F. Mair, and J. P. Garrahan, New J. Phys. **23**, 013013 (2021).

[34] A. Das, D. C. Rose, J. P. Garrahan, and D. T. Limmer, arXiv:2105.04321 [cond-mat, physics:physics] (2021), arXiv:2105.04321 [cond-mat, physics:physics].

[35] T. Nemoto, F. Bouchet, R. L. Jack, and V. Lecomte, Phys. Rev. E **93**, 062123 (2016).

[36] W. E and B. Yu, Commun. Math. Stat. **6**, 1 (2018).

[37] G. M. Rotskoff, A. R. Mitchell, and E. Vanden-Eijnden, arXiv:2008.06334 [cond-mat, physics:physics, stat] (2021), arXiv:2008.06334 [cond-mat, physics:physics, stat].

[38] X. Li, T.-K. L. Wong, R. T. Q. Chen, and D. Duvenaud, in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, Vol. 108 (PMLR, 2020) pp. 3870–3882.

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

[40] H. Touchette, Phys. Rep. **478**, 1 (2009).

[41] R. V. Kohn, M. G. Reznikoff, and E. Vanden-Eijnden, J

Nonlinear Sci **15**, 223 (2005).
[42] T. Speck, A. Engel, and U. Seifert, J. Stat. Mech. Theor. Exp. **2012**, P12001 (2012).
[43] R. Chetrite and H. Touchette, J. Stat. Mech. **2015**, P12001 (2015).
[44] 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.
[45] T. H. E. Oakes, A. Moss, and J. P. Garrahan, Machine Learning: Sci. & Tech. **1**, 035004 (2020).
[46] 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, Inc., 2018) pp. 7146–7155.
[47] 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, Inc., 2018) pp. 3036–3046.
[48] S. Mei, A. Montanari, and P.-M. Nguyen, Proc Natl Acad Sci USA **115**, E7665 (2018).
[49] J. Sirignano and K. Spiliopoulos, arXiv (2018), arXiv:1805.01053v1.
[50] A. R. Barron, IEEE Transactions on Information Theory **39**, 930 (1993).
[51] G. Cybenko, Math. Control Signal Systems **2**, 303 (1989).
[52] B. Tzen and M. Raginsky, arXiv:1905.09883 [cs, stat] (2019), arXiv:1905.09883 [cs, stat].
[53] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, arXiv:2010.08895 [cs, math] (2021), arXiv:2010.08895 [cs, math].
[54] D. Frenkel and B. Smit, *Understanding Molecular Simulation*, edited by D. Frenkel and B. Smit (Academic Press, San Diego, 2002).
[55] G. M. Rotskoff and E. Vanden-Eijnden, arXiv:1805.00915 [cond-mat, stat] (2018), arXiv:1805.00915 [cond-mat, stat].
[56] M. Belkin, D. Hsu, S. Ma, and S. Mandal, Proc Natl Acad Sci USA **116**, 15849 (2019).
[57] M. E. Cates and J. Tailleur, Annu Rev Condens Matter Phys **6**, 219 (2015).
[58] J. Bialké, A. M. Menzel, H. Löwen, and T. Speck, Phys. Rev. Lett. **112**, 218304 (2014).
[59] P. Pietzonka, K. Kleinbeck, and U. Seifert, New J. Phys. (2016).
[60] M. F. Hagan, A. Baskaran, and G. S. Redner, Phys. Rev. Lett. **110**, 055701 (2013).
[61] M. Nguyen and S. Vaikuntanathan, PNAS **113**, 14231 (2016).
[62] É. Fodor, T. Nemoto, S. Vaikuntanathan, and L. Tociu, Phys. Rev. X **9**, 041026 (2019).
[63] É. Fodor, T. Nemoto, and S. Vaikuntanathan, New J. Phys. **22**, 013052 (2020).
[64] T. GrandPre, K. Klymko, K. K. Mandadapu, and D. T. Limmer, Physical Review E **103**, 012613 (2021).
[65] L. Caprini, A. Puglisi, and A. Sarracino, Symmetry **13**, 81 (2021).
[66] B. Oksendal, *Stochastic Differential Equations: An Introduction with Applications*, Universitext (Springer, Berlin, 1992).
[67] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, arXiv:1701.06547 [cs] (2017), arXiv:1701.06547 [cs].

Appendix A: Derivation of the cost functional

In analogy to importance sampling, we can write the expression for the scaled cumulant generating function as an expectation over a "tilted" or biased path measure,

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

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 Girsanov theorem [66]. In our case, using the parameterization that $u(\boldsymbol{x},\lambda) = b(\boldsymbol{x}) + \delta u(\boldsymbol{x},\lambda)$, the Radon-Nikodym derivative can be written explicitly

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

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 (A1)

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

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

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

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

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

Appendix B: Algorithm and Computational Details

The goal of our algorithm, detailed in Alg. 1, is to learn the optimal (time-independent) control drift u^* of the modified process

$$d\boldsymbol{X}^u_t = b(\boldsymbol{X}^u_t)dt + \delta u(\boldsymbol{X}^u_t, \lambda; \theta)dt + \sigma d\boldsymbol{W}_t \tag{B1}$$

expressed here in terms of the drift perturbation $\delta u(\boldsymbol{x}, \lambda; \theta)$ involving a set of parameters θ and the Lagrange parameter λ entering in the Lagrangian. Depending on the system considered, the gradient of the Lagrangian can be evaluated, as mentioned in the main text, either by using back-propagation when the integration time is short or by using adjoint state methods when longer integration times might be necessary or desirable to save memory resources. The latter method is explained in the next section.

The crux of our method is to encode $\delta u(\cdot, \lambda; \theta)$ using a neural network similar to the architecture used in the deep Ritz method [36]. The architecture contains multiple layers L_i, where each layer consists of two linear transformation, two nonlinear activation functions and a residual connection:

$$L_i(\boldsymbol{X}) = \phi[W_{i,2} \cdot \phi(W_{i,1}\boldsymbol{X} + b_{i,1}) + b_{i,2}] + \boldsymbol{X} \tag{B2}$$

where $W_{i,j} \in \mathbb{R}^{h\times h}$ and $b_{i,j} \in \mathbb{R}^h$ are parameters for the i-th layer, h is the dimension of the hidden layers, and ϕ is the activation function. The residual connection helps with stability and helps avoid the vanishing gradient problem. Since our approach requires simulating trajectories from Eq. (B1), 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

$$z_\theta(\boldsymbol{X}) = L_n \otimes \cdots \otimes L_1(\boldsymbol{X}). \tag{B3}$$

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

All computations presented here were performed using Python, Pytorch, and the torchsde [67] package. Our source code and associated data are available at github.com/quark-strange/machine_learning_LDP.

Algorithm1 Concurrent Training

```
**Data:** Lagrangian $\mathcal{L}[\boldsymbol{X}^u, \delta u(\boldsymbol{X}_t^u, \lambda; \boldsymbol{\theta}), \lambda]$, initial $\boldsymbol{\theta}$, $k_{\max} \in \mathbb{N}$ total duration, $T \in \mathbb{R}$ the duration of sampled trajectory, $M(n) \in \mathbb{N}$ the batch size. $\{\lambda_1, \lambda_2, \cdots, \lambda_N\} \subset \mathbb{R}$, $\alpha > 0$ the learning rate.
$k = 0$
**while** $k < k_{\max}$ **do**
    **for** $n = 1, \ldots, N$ **do**
        **for** $m = 1, \ldots, M(n)$ **do**
            Sample $\boldsymbol{X}^{u(\lambda_n)}_{[0,T],m}$ according to $d\boldsymbol{X}_t^u = [b(\boldsymbol{X}_t^u) + \delta u(\boldsymbol{X}_t^u, \lambda_n; \theta)]dt + \sigma d\boldsymbol{W}_t$ with initial condition $\boldsymbol{X}^{u(\lambda_n)}_{0,m}$;
    Compute
        $$\mathcal{L}_{\boldsymbol{\theta}}^{(n)} = \frac{1}{M(n)} \sum_{m=1}^{M(n)} \lambda_n \int_0^T f(\boldsymbol{X}_{t,m}^{u(\lambda_n)})dt + g(\boldsymbol{X}_{t,m}^{u(\lambda_n)}) \circ d\boldsymbol{X}_{t,m}^{u(\lambda_n)} - \frac{1}{2}\int_0^T \delta u(\boldsymbol{X}_{t,m}^{u(\lambda_n)}, \lambda_n; \boldsymbol{\theta}) D^{-1} \delta u(\boldsymbol{X}_{t,m}^{u(\lambda_n)}, \lambda_n; \boldsymbol{\theta}) dt$$
        $$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\theta) = \frac{1}{N} \sum_{n=1}^{N} \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\theta}}^{(n)}$$
    Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$
    Update the initial condition $\boldsymbol{X}^{u(\lambda_n)}_{0,m} \leftarrow \boldsymbol{X}^{u(\lambda_n)}_{T,m}$
    **procedure** (Optional) Replica exchange:
        Select two random $n_1 \neq n_2$ and $m_i \leqslant M(n_i)$;
        Compute the Radon-Nikodym derivative:
            $$M_T = \frac{d\mathbb{P}[\boldsymbol{X}^{u(\lambda_{n_1})}_{[0,T],m_1}]}{d\mathbb{P}[\boldsymbol{X}^{u(\lambda_{n_2})}_{[0,T],m_2}]} = \frac{\exp\left\{-\frac{1}{4D}\int_0^T |\dot{\boldsymbol{X}}^{u(\lambda_{n_1})}_{t,m_1} - u(\boldsymbol{X}^{u(\lambda_{n_1})}_{t,m_1}, \lambda_{n_1})|^2 dt\right\}}{\exp\left\{-\frac{1}{4D}\int_0^T |\dot{\boldsymbol{X}}^{u(\lambda_{n_2})}_{t,m_2} - u(\boldsymbol{X}^{u(\lambda_{n_2})}_{t,m_2}, \lambda_{n_2})|^2 dt\right\}} \tag{B4}$$
        $u \sim \mathrm{Uniform}(0, 1)$
        **if** $u < \min[1, M_T]$ **then**
            exchange $\boldsymbol{X}^{u(\lambda_{n_1})}_{0,m_1}$ and $\boldsymbol{X}^{u(\lambda_{n_2})}_{0,m_2}$.
**return:** $\boldsymbol{\theta}$.
```

1. Adjoint State Methods

The adjoint state method for Stratonovich SDEs (note that the choice of Ito or Stratonovich convention is immaterial in our examples because we consider only SDEs with additive noise) 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/backward integration of the differential equation and, in the stochastic case, one must solve the SDE backward in time with the same Weiner 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 an ODE

$$\frac{dx}{dt} = u(x, t, \theta); \qquad x(0) = x_0 \tag{B5}$$

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

$$x(T) = x_0 + \int_0^T u(x, t, \theta) dt. \tag{B6}$$

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

$$\frac{\partial \mathcal{L}(x(T))}{\partial \theta} = \frac{\partial \mathcal{L}(x(T))}{\partial x(T)} \frac{\partial x(T)}{\partial \theta} \tag{B7}$$

where $x(t)$ is constrained to follow the dynamics (B5). 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

$$\tilde{\mathcal{L}}(x(T)) = \mathcal{L}(x(T)) - \int_0^T \mathfrak{A}(t)(\dot{x} - u(x, \theta, t)) dt \tag{B8}$$

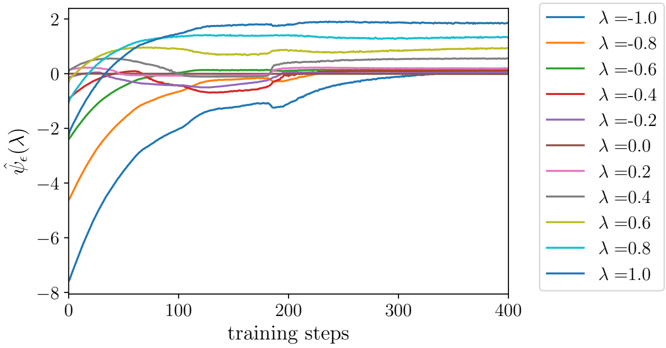

Figure 3. Illustration of the concurrent training for the small noise example with $\epsilon = 0.01$. Each line corresponds to the evolution of the cost function with a specific λ.

so that

$$\begin{aligned}\frac{\partial \tilde{\mathcal{L}}(x(T))}{\partial\theta} &= \frac{\partial \mathcal{L}(x(T))}{\partial x(T)}\frac{\partial x(T)}{\partial\theta} - \mathfrak{A}(T)\frac{\partial x(T)}{\partial\theta} - \frac{\partial}{\partial\theta}\int_0^T \dot{\mathfrak{A}}(t)(x(t)-u(x,\theta,t))dt \\ &= \frac{\partial \mathcal{L}(x(T))}{\partial x(T)}\frac{\partial x(T)}{\partial\theta} - \mathfrak{A}(T)\frac{\partial x(T)}{\partial\theta} + \int_0^T \dot{\mathfrak{A}}(t)\frac{\partial x(t)}{\partial\theta} + \mathfrak{A}(t)\frac{\partial u(x,\theta,t)}{\partial x(t)}\frac{\partial x(t)}{\partial\theta} + \mathfrak{A}(t)\frac{\partial u(x,\theta,t)}{\partial\theta}dt\end{aligned} \tag{B9}$$

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

$$\dot{\mathfrak{A}}(t) = -\mathfrak{A}(t)\frac{\partial u(x,\theta,t)}{\partial x(t)}; \qquad \mathfrak{A}(T) = \frac{\partial \mathcal{L}(x(T))}{\partial x(T)} \tag{B10}$$

in order to write the gradient as

$$\frac{\partial \mathcal{L}(x(T))}{\partial\theta} = -\int_T^0 \mathfrak{A}(t)\frac{\partial u(x,\theta,t)}{\partial\theta}dt \tag{B11}$$

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 current-like observables) [38].

2. Computational details for the small noise example

The SCGF for the small noise example in the Main Text is computed through Alg. 1. 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 uniformly from -1 to 1, where each λ contains 20 replica. At each training step, a total number of 220 trajectories with $T = 10$ are generated by 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 SCGF is then refined by changing λ and simulate the resulting driven process. Fig. 3 shows the convergence of $\hat{\psi}_\epsilon(\lambda)$ for $\epsilon = 0.01$. The numbers of steps required vary little for different ϵ, which are typically in the range of 400 to 600. The replica exchange is not crucial here and does not noticeably improve the accuracy in this case.

3. Computational details for the active Brownian particle example

For the active Brownian particle, its motion is determined by Eq. (11) where U is the WCA potential which is a function of the relative distance l_{ij} of all particles:

$$U(l_{ij}) = \begin{cases} 4\epsilon\left[(\sigma/l_{ij})^{12} - (\sigma/l_{ij})^6\right] + \epsilon, & l_{ij} \leqslant 2^{1/6}\sigma \\ 0, & l_{ij} > 2^{1/6}\sigma, \end{cases} \tag{B12}$$

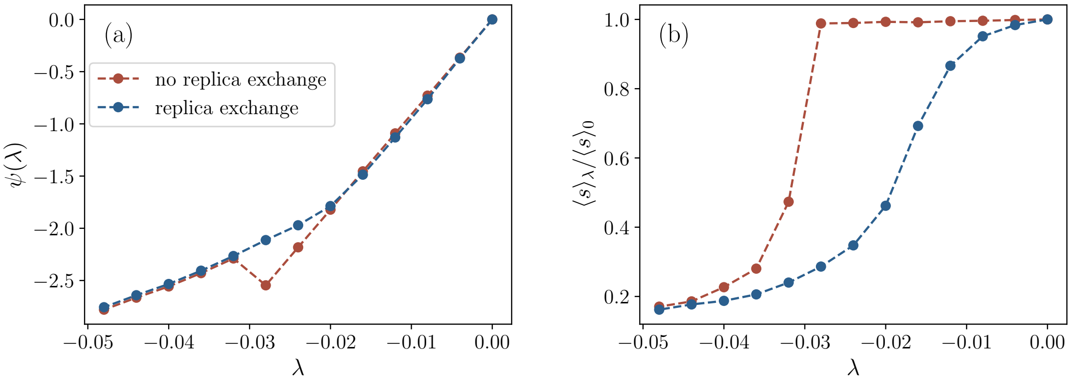

Figure 4. (a) The estimate of the SCGF for $N = 40$ with/without replica exchange. The blue line corresponds to the results with replica exchange and the red line shows the results without. (b) The corresponding changes of average entropy production as decreasing λ.

In the simulation, the unit of length is normalized by σ and we set $\epsilon = 1$. The simulations are performed with periodic boundary condition, and the relative distance matrix l_{ij} is adjusted by the minimum image convention. 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.

The active Brownian particle results are computed through the sequential training since concurrent training requires a large total batch size which is computationally costly for high dimensional systems. We notice that it requires much longer time for convergence when first driving the system into the clustering phase, but once we obtain a control force, it converges fast when sequentially altering λ. Here 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. We used Adadelta as the optimizer.

For $N = 40, 80$, replica exchange is required to obtain a convex SCGF (which must be the case by definition). For $N = 200$, replica exchange is not necessary. The replica exchange is implemented by concurrently training with λ_i and $\lambda_0 = -0.05$, with batch size 75 and 75, respectively. Then at each step all the 75 trajectories are attempted to be exchanged, as stated in Alg. 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.

Appendix C: Cost estimator

Because we are interested in the limit $T \to \infty$, the control force u is necessarily time-independent and hence it must be the case that for any $t < \infty$, we have a short time estimator that converges to the long time limit

$$\frac{1}{Nt}\sum_{i=1}^{N}\sup_{\delta u}\left\{\lambda\int_0^t f(\boldsymbol{X}^u_{s,i})ds + g(\boldsymbol{X}^u_{s,i})\circ d\boldsymbol{X}^u_{s,i} - \frac{1}{2}\int_0^t \delta u(\boldsymbol{X}^u_{s,i})D^{-1}\delta u(\boldsymbol{X}^u_{s,i})ds\right\} \overset{N\to\infty}{\longrightarrow} \lim_{T\to\infty}\frac{1}{T}\sup_{\delta u}\mathcal{L}[\boldsymbol{X}^u_{[0,T]},u], \tag{C1}$$

where $\boldsymbol{X}^u_{0,i}$ are sampled from the steady state of the controlled process (6). We compute the cost functional numerically by simulating N independent trajectories $\boldsymbol{X}^u_{t,i}$, referred to as replicas, over a finite time-window or horizon $[0, t]$ by using the estimator

$$\hat{\psi}_{Nt}(\lambda) = \frac{1}{Nt}\sum_{i=1}^{N}\hat{\psi}_{t,i}(\lambda), \tag{C2}$$

where

$$\hat{\psi}_{t,i}(\lambda) = \frac{1}{t}\left\{\lambda\int_0^t f(\boldsymbol{X}^u_{s,i})ds + g(\boldsymbol{X}^u_{s,i})\circ d\boldsymbol{X}^u_{s,i} - \frac{1}{2}\int_0^t \delta u(\boldsymbol{X}^u_{s,i})D^{-1}\delta u(\boldsymbol{X}^u_{s,i})ds\right\}. \tag{C3}$$

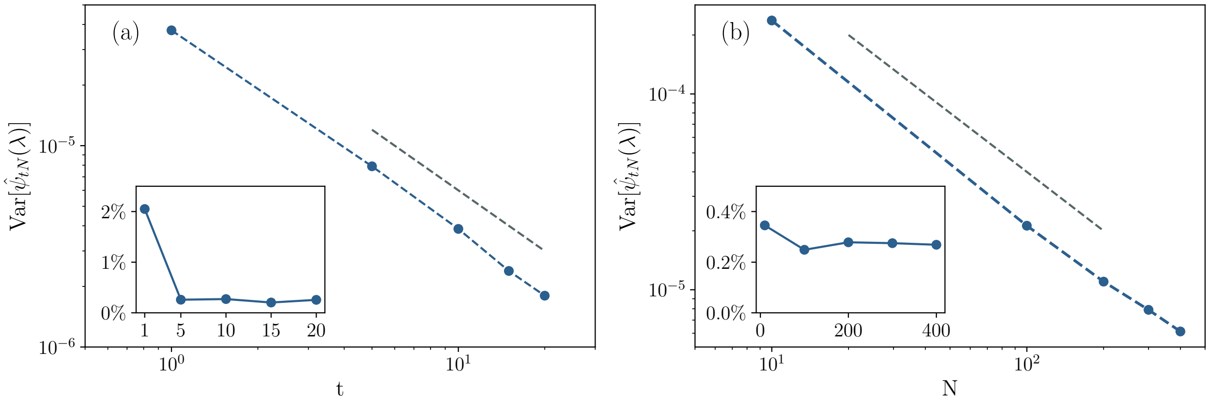

Figure 5. Variance of the estimator. We illustrate the scaling property of the variance of our estimator Eq. (C2) using the small noise example. By fixing $\lambda = 1$ and $\epsilon = 0.01$, the neural network is train with (a) a fixed batch size $N = 300$ with various trajectory length t from 1 to 20, or (b) a fixed $t = 5$ and various batch size from 10 to 400. The neural network 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 grey dashed line represents a -1 slope.

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^*.

The mean squared error (MSE) of the estimator is $\mathbb{E}[\hat{\psi}_{Nt}(\lambda) - \psi(\lambda)]^2 = \mathrm{Var}[\hat{\psi}_{Nt}(\lambda)]$ since it is unbiased. Moreover, 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 our estimator decreases with the scaling $(tN)^{-1}$. In other words, a large-batch and short-time estimator is equivalent to a small-batch and long-time estimator. In Fig. 5 we plot this scaling property of $\psi_\epsilon(\lambda)$ in the small noise example ($\epsilon = 0.01$ and fixed $\lambda = 1$), which confirm that a short time estimator converges to the long time limit.