Evolutionary reinforcement learning of dynamical large deviations

Stephen Whitelam, Daniel Jacobson, and Isaac Tamblyn

Molecular Foundry, Lawrence Berkeley National Laboratory,
1 Cyclotron Road, Berkeley, CA 94720, USA
Division of Chemistry and Chemical Engineering,
California Institute of Technology,
Pasadena, California 91125, USA
University of Ottawa & National Research Council of Canada,
Ottawa, ON, Canada

We show how to calculate dynamical large deviations using evolutionary reinforcement learning. An agent, a stochastic model, propagates a continuous-time Monte Carlo trajectory, and receives a reward conditioned upon the values of certain path-extensive quantities. Evolution produces progressively fitter agents, allowing the calculation of a piece of a large-deviation rate function for a particular model and path-extensive quantity. For models with small state spaces the evolutionary process acts directly on rates, and for models with large state spaces the process acts on the weights of a neural network that parameterizes the model’s rates. The present approach shows how path-extensive physics problems can be considered within a framework widely used in machine learning.

**Introduction**— Machine learning has begun to provide the physics community with methods that complement the traditional ones of physical insight and manipulation of equations. Many-parameter ansätze, sometimes encoded in the form of neural networks, can learn connections between physical properties (such as the positions of atoms and a system’s internal energy) and detect phase transitions, without drawing upon an underlying physical model. Reinforcement learning (RL) is a branch of machine learning concerned with learning a set of actions, through interactions with an environment, so as to maximize a numerical reward. It has a close connection to ideas of stochastic control enacted by variational methods. Many-parameter ansätze, sometimes encoded in the form of neural networks, can learn connections between physical properties (such as the positions of atoms and a system’s internal energy) and detect phase transitions, without drawing upon an underlying physical model. Reinforcement learning (RL) is a branch of machine learning concerned with learning a set of actions, through interactions with an environment, so as to maximize a numerical reward. It has a close connection to ideas of stochastic control enacted by variational methods.

Reinforcement learning (RL) is a branch of machine learning concerned with learning a set of actions, through interactions with an environment, so as to maximize a numerical reward. It has a close connection to ideas of stochastic control enacted by variational methods. Many-parameter ansätze, sometimes encoded in the form of neural networks, can learn connections between physical properties (such as the positions of atoms and a system’s internal energy) and detect phase transitions, without drawing upon an underlying physical model. Reinforcement learning (RL) is a branch of machine learning concerned with learning a set of actions, through interactions with an environment, so as to maximize a numerical reward. It has a close connection to ideas of stochastic control enacted by variational methods.

**Large deviations by change of dynamics**— To set the large-deviations problem in a form amenable to reinforcement learning, consider a continuous-time Monte Carlo dynamics on a set of discrete states, with \( W_{xy} \) the rate for passing between states \( x \) and \( y \), and \( R_x = \sum_{y \neq x} W_{xy} \) the escape rate from \( x \). This dynamics generates a trajectory \( \omega = x_0 \rightarrow x_1 \rightarrow \cdots \rightarrow x_N(\omega) \) consisting of \( N(\omega) \) jumps \( x_n \rightarrow x_{n+1} \) and associated jump times \( \Delta t_n \). In the language of reinforcement learning, \( W_{xy} \) is a policy that stochastically selects a new state and a jump time given a current state. Stochastic trajectories can be characterized by path-extensive observables \( A = aT \), with

\[
a = T^{-1} \sum_{n=0}^{N-1} A_{x_n x_{n+1}}.
\]

Here \( \alpha_{xy} \) is the change of the observable upon moving between \( x \) and \( y \). This type of observable describes many physically important quantities, including work, entropy production, and non-decreasing counting observables. The large-deviation form quantifies the likelihood of observing atypical values of \( a \). Finite-time fluctuations \( a \neq a_0 \) occur with a probability controlled by the distribution \( \rho_T(A) \), taken over all trajectories of length \( T \), which for large \( T \) often adopts the large-deviation form

\[
\rho_T(A) \approx e^{-T J(a)}.
\]

**Large deviations by change of dynamics**— To set the large-deviations problem in a form amenable to reinforcement learning, consider a continuous-time Monte Carlo dynamics on a set of discrete states, with \( W_{xy} \) the rate for passing between states \( x \) and \( y \), and \( R_x = \sum_{y \neq x} W_{xy} \) the escape rate from \( x \). This dynamics generates a trajectory \( \omega = x_0 \rightarrow x_1 \rightarrow \cdots \rightarrow x_N(\omega) \) consisting of \( N(\omega) \) jumps \( x_n \rightarrow x_{n+1} \) and associated jump times \( \Delta t_n \). In the language of reinforcement learning, \( W_{xy} \) is a policy that stochastically selects a new state and a jump time given a current state. Stochastic trajectories can be characterized by path-extensive observables \( A = aT \), with

\[
a = T^{-1} \sum_{n=0}^{N-1} A_{x_n x_{n+1}}.
\]

Here \( \alpha_{xy} \) is the change of the observable upon moving between \( x \) and \( y \). This type of observable describes many physically important quantities, including work, entropy production, and non-decreasing counting observables. The large-deviation form quantifies the likelihood of observing atypical values of \( a \). Finite-time fluctuations \( a \neq a_0 \) occur with a probability controlled by the distribution \( \rho_T(A) \), taken over all trajectories of length \( T \), which for large \( T \) often adopts the large-deviation form

\[
\rho_T(A) \approx e^{-T J(a)}.
\]
reference-model trajectory be \( \tilde{a}_0 \). Then an upper bound on \( J(a) \) at \( a = \tilde{a}_0 \) is given by the value of

\[
J_0 = -T^{-1} \sum_{n=0}^{N-1} q_{x_n x_{n+1}}
\]  

(3)

for a long reference-model trajectory, where

\[
q_{x_n x_{n+1}} = \ln \frac{W_{x_n x_{n+1}}}{W_{x_n x_{n+1}}} - \Delta t_n (R_{x_n} - \tilde{R}_{x_n}).
\]  

(4)

Here \( \Delta t_n = -\ln \eta/\tilde{R}_{x_n} \) is the jump time of the reference model, and \( \eta \) is a random number uniformly distributed on \((0,1] \). Eq. (3) follows from straightforward algebra (see Section S1). It can be motivated by noting that the probability of a jump \( x \rightarrow y \) in time \( \Delta t \) occurs in the reference model with probability \( W_{xy} e^{-\tilde{R}_t \Delta t} \), and in the original model with probability \( W_{xy} e^{-R_t \Delta t} \). Eq. (4) is the sum over a trajectory of the log-ratio of such terms.

Our aim is to use evolutionary reinforcement learning to find a reference model (a new policy) \( W_{xy} \) that produces particular values of (1), and, given this, minimizes (3). Formulated in this way this is an extreme example of reinforcement learning in which there is no instantaneous reward, only an overall reward (or return) associated with the entire trajectory [15]. Given that we possess a constraint on \( a \) and work in continuous time, this problem also falls outside the Markov decision process framework [61, 62].

Large deviations via evolutionary reinforcement learning—As proof of principle we consider the example of entropy production in the 4-state model [63] of Ref. [64]. The model’s rates do not satisfy detailed balance, and so it produces nonzero entropy on average. The dynamical observable \( a \) is (1) with \( \alpha_{xy} = \ln(p_{xy}/p_{yx}) \), where \( p_{xy} = W_{xy}/R_x \). In Fig. 1a we depict the model (the middle picture), with states \( x \) numbered clockwise from 1 in the top left. Red and blue links denote connections \( x \rightarrow y \) with negative and positive entropy production, respectively, and the thickness of the links is proportional to the rate associated with the connection. The model’s state space is small enough that the master operator can be solved by diagonalization [49], yielding the exact rate function \( J(a) \), shown as a black dashed line in Fig. 1b).

We perform evolutionary reinforcement learning by mutating the rates \( W_{xy} \) of a set of reference models until desired values of (1) and (3) are achieved (see Section S2). Some of the models produced in this way are shown in Fig. 1a), and the associated rate-function values are shown in panel (b). All points \( \langle \tilde{a}_0, J_0 \rangle \) derived from the reference models lie on the exact rate function of the original model (if we did not possess the exact answer we could verify, by measuring the fluctuations of the reference models, that this is the case [55]), indicating that each reference model’s typical dynamics is equivalent to the conditioned rare dynamics of the original model. Some of the reference models so obtained are shown in panel (a) and in the right-bottom inset of panel (b). The gray dashed line is the universal current rate-function bound [63, 65]. This bound measures, for given \( a \), the probability with which typical trajectories of a reference model whose rates are uniformly rescaled versions of the original model (right-top inset) would have been generated by the original model. Such trajectories are much less likely than are trajectories of the models identified by evolution.

A neural-network ansatz for models with large state spaces—Thus evolutionary reinforcement learning using 12 trainable parameters (the 12 rates of the reference model) has permitted accurate computation of probabilities exponentially small in the trajectory length \( T \). However, direct application of rate-based evolution is impractical for models with a large number of rates, necessitating a more efficient representation of those rates [15, 16]. One way to do this is to encode the rates of the reference model in the form of a neural network. As an illustration of this procedure we consider the one-dimensional

![FIG. 1](image_url)
Fredrickson-Andersen (FA) model with periodic boundary conditions [67]. This is a lattice model with dynamical rules that give rise to slow relaxation and complex space-time behavior [68]. On each site \( i \) of a lattice of length \( L \) lives a spin \( S_i \), which can be up (+1) or down (−1). Up spins (resp. down spins) flip down (resp. up) with rate \( 1 - c \) (resp. \( c \)) if at least one of their neighboring spins is up; if not, then they cannot flip. We take the dynamical observable \( \alpha \) to be the number of configuration changes per unit time, \( \alpha_{xy} = 1 \), often called activity [11, 69]. To determine the large-deviation rate function \( J(a) \) for activity we chose a reference-model parameterization

\[
\tilde{W}_{xy} = W_{xy}e^{w_0}e^{f_y - f_x}.
\]  

(5)

Here \( w_0 \) is a parameter that effectively speeds up or slows down the clock [55], and \( f_x \) is the value in state \( x \) of the neural network sketched in Fig. 2. This network is inspired by the convolutional neural networks used to recognize images [70, 71], and consists of a set of spin “filters” that scan the lattice for specified spin patterns. Here we consider filters called \( K^\alpha \), each having \( L \) hidden nodes; the output of a node is 1 if the \( K \) consecutive spins to which it is attached are all in state \( \alpha \), and is zero otherwise (i.e. the activation function is a step function). The network has one hidden layer. The weights connecting the input layer (the lattice) to the hidden layer are unity, and the weights connecting the hidden layer to the output node are denoted \( w_{K^\alpha} \); these are the trainable parameters of the network. All weights within a filter have the same value, a constraint suggested by the translational invariance of the model. The output of the network is

\[
f_x = w_1 g_1(S_x) + \sum_{k=2}^{K} \sum_{a=\pm 1} w_k^a g_k^a(S_x),
\]  

(6)

where \( S_x \) is the configuration of the lattice in state \( x \), and \( g_k^a(\cdot) \) returns the number of active hidden nodes in the filter \( K^a \) [see Fig. 2(a)]. The reference model contains \( 2K \) trainable parameters: \( w_0, w_1 \) (only one type of 1-spin filter is necessary), and \( w_2^\pm, w_3^\pm, \ldots, w_K^\pm; K = L \) when all filter types are used.

The form of \( J(\cdot) \) is similar to the multi-parameter auxiliary potential of Ref. [18], used to improve the convergence of the cloning method [14] in order to calculate the large-deviation function of the FA model. The present approach is different, however, in that the calculation is done using direct simulations of a reference model whose parameters are determined by an evolutionary process (rather than using rare-event algorithms such as cloning or transition-path sampling [22]), and results in the calculation of \( J(a) \) directly (rather than its Legendre transform, which in general contains less information [40]).

To test the method we considered the FA model with \( c = 0.3 \) and \( L = 15 \), the latter value being small enough that the exact \( J(a) \) can be determined by diagonalization of the model’s rate matrix; that function is shown as a black dashed line in Fig. 3(a). We next introduce the reference model [5], and do evolutionary reinforcement learning on the weights of the network (see Section S3). In Fig. 3(a) we show results using spin filters up to order \( K = 7 \). Increasing \( K \) from 0 improves the quality of the bound until, for \( K \gtrsim 4 \), the bound becomes numerically close to the exact answer (Fig. S2). The neural network contains many fewer parameters than the model has rates (unlike in many deep-learning studies), and so we do not necessarily expect the bound to be exact. As long as it is tight enough, the exact answer can be calculated by computing a correction term [55]. Here, though, the correction term is very small, indicating that the typical dynamics of this set of reference models is statistically similar to the conditioned rare behavior of the original model. Comparison of these results with the exact result, and with the \((c, \lambda)\)-bound from Ref. [55] (Fig. S2), indicates that rare trajectories of the FA model with parameter \( c \) resemble the typical trajectories of versions of the FA model with different values of the parameter \( c \), but with slightly different tendencies to display spin domains of different lengths. These tendencies are quantified by the weights of the neural network, some of which are shown in Fig. 3(b). In panel (c) we show space-time plots of the trajectories of 5 reference models.
In previous work we showed how to calculate dynamical large deviations can be done using direct simulation guided by evolutionary reinforcement learning using a multi-parameter ansatz or "VARDnet" removes the need for such insight. Evolved models allow the accurate calculation of probabilities exponentially small in the length of the path, far beyond the reach of simulations of the original model. Our approach does not rely on the formal results of large-deviation theory, making it complementary to the growing body of methods based on such results [16, 18, 19, 40, 41, 43, 46].

Our results also suggest ways of addressing other physical problems. For one, the neural networks shown in Fig. 2 and Fig. S9(a) treat a lattice model much as a convolutional neural network treats images, and it is straightforward to imagine extending this approach to other models and higher spatial dimensions, taking the pixels of the configuration as information on which the evolutionary process can act. More generally, many physical problems involve time- or path-extensive quantities – for instance, the yield of molecular self-assembly depends on the history of the interactions of molecules – and we have shown that it is possible to generate dynamical trajectories conditioned upon rare instances of path-extensive quantities without physical insight or access to formal results.

Acknowledgments—We thank Hugo Touchette for comments. This work was performed as part of a user project at the Molecular Foundry, Lawrence Berkeley National Laboratory, supported by the Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. D.J. acknowledges support from the Department of Energy Computational Science Graduate Fellowship. I.T. performed work at the National Research Council of Canada under the auspices of the AI4D Program.

\[ \frac{\partial}{\partial t} \langle \mathcal{O} \rangle = \frac{\partial}{\partial a_0} \left[ \sum_{\text{config}} \mathcal{O} \mathcal{P}(\mathcal{P} \mid a_0) \right] \]

\[ \mathcal{P}(\mathcal{P} \mid a_0) = \left( \prod_{i=1}^{L} \mathcal{P}(\mathcal{P}_i \mid a_0) \right) \]

\[ \mathcal{P}(\mathcal{P}_i \mid a_0) = \frac{\exp(\mathcal{H}(\mathcal{P}_i))}{\sum_{\mathcal{P}_i} \exp(\mathcal{H}(\mathcal{P}_i))} \]

\[ \mathcal{H}(\mathcal{P}_i) = -a_0 \mathcal{P}_i \]

\[ \frac{\partial}{\partial t} \langle \mathcal{O} \rangle = \frac{\partial}{\partial a_0} \left[ \sum_{\text{config}} \mathcal{O} \mathcal{P}(\mathcal{P} \mid a_0) \right] \]

\[ \mathcal{P}(\mathcal{P} \mid a_0) = \left( \prod_{i=1}^{L} \mathcal{P}(\mathcal{P}_i \mid a_0) \right) \]

\[ \mathcal{P}(\mathcal{P}_i \mid a_0) = \frac{\exp(\mathcal{H}(\mathcal{P}_i))}{\sum_{\mathcal{P}_i} \exp(\mathcal{H}(\mathcal{P}_i))} \]

\[ \mathcal{H}(\mathcal{P}_i) = -a_0 \mathcal{P}_i \]
S1. LARGE DEVIATIONS BY CHANGE OF MODEL

For completeness we present the derivation of Eq. (3) of the main text, which follows straightforwardly from the definition of the probability distribution. The derivation follows Ref. [53] with minor notational changes. For more on the ideas of dynamic importance sampling see e.g. Refs. [10, 83, 53] and Ref. [10] (esp. Section 5).

Consider a continuous-time dynamics on a set of discrete states, defined by the master equation [73]

$$\partial_t p_x(t) = \sum_{y \neq x} W_{xy} p_y(t) - R_x p_x(t). \tag{S1}$$

Here $p_x(t)$ is the probability that the system is in (micro)state $x$ at time $t$, $W_{xy}$ is the rate for passing from state $x$ to state $y$, and $R_x = \sum_{y \neq x} W_{xy}$ is the escape rate from $x$. A standard way of simulating (S1) is as follows [53]: from state $x$, choose a new state $y$ with probability

$$p_{xy} = \frac{W_{xy}}{R_x}, \tag{S2}$$

and a time increment $\Delta t$ from the distribution

$$p_x(\Delta t) = R_x e^{-R_x \Delta t}. \tag{S3}$$

The dynamics defined by (S2) and (S3) generates a trajectory $\omega = x_0 \to x_1 \to \cdots \to x_{N(\omega)}$ consisting of $N(\omega)$ jumps $x_n \to x_{n+1}$ and associated jump times $\Delta t_n$. Associated with an ensemble of trajectories of length $T$ is the probability distribution

$$\rho_T(A) = \sum_{\omega} p(\omega) \delta(T) \delta(A) \tag{S4}$$

of a time-extensive dynamical observable

$$A(\omega) = \sum_{n=0}^{N(\omega)-1} \alpha_{x_n x_{n+1}}. \tag{S5}$$

In these expressions $\delta(X) \equiv \delta(X(\omega) - X)$ specifies a constraint on the trajectory, $\alpha_{xy}$ is the change of $A$ upon moving from $x$ to $y$, and $A(\omega)$ is the sum of these quantities over a single trajectory $\omega$. We define $a(\omega) \equiv A(\omega)/T(\omega)$ as the time-intensive version of $A$. $T(\omega)$ is the elapsed time of trajectory $\omega$, and $p(\omega)$ is the probability of a trajectory $\omega$, proportional to a product of factors (S2) and (S3) for all jumps of the trajectory.

Fluctuations of $a$ are quantified by $\rho_T(A)$, which for large $T$ often adopts a large-deviation form [40, 42].

$$\rho_T(A) \approx e^{-T \cdot J(a)}. \tag{S6}$$

Direct evaluation of (S4) using the dynamics (S2) and (S3) leads to good sampling of $J(a)$ near the typical value $a_0$, where $J(a_0) = 0$, and poor sampling elsewhere. To overcome this problem we can introduce a reference dynamics

$$\tilde{p}_{xy} = \frac{\tilde{W}_{xy}}{\tilde{R}_x}, \tag{S7}$$

and

$$\tilde{p}_x(\Delta t) = \tilde{R}_x e^{-\tilde{R}_x \Delta t}, \tag{S8}$$

in which $\tilde{W}_{xy}$ is a modified version of the rate of the original model, and $\tilde{R}_x \equiv \sum_y \tilde{W}_{xy}$. Let $\tilde{p}(\omega)$ be the trajectory weight of the reference dynamics, proportional to a product of factors (S7) and (S8) for all jumps of the trajectory. We write

$$\langle \cdot \rangle \equiv \sum_{\omega} p(\omega) \langle \cdot \rangle \delta(T) \delta(aT) \tag{S9}$$

and

$$\langle \cdot \rangle_{\text{ref}} = \sum_{\omega} \tilde{p}(\omega) \langle \cdot \rangle \delta(T) \delta(aT) \tag{S10}$$

for the ensemble averages over trajectories (having length $T$ and observable $A = aT$) of the original and reference models, respectively. We can then write (S4) as

$$\rho_T(A) = (1)^a \equiv (\epsilon^{T q(\omega)})^a_{\text{ref}} = (\epsilon^{T q(\omega)})^a_{\text{ref}} (\epsilon^{T \delta q(\omega)})^a_{\text{ref}}. \tag{S11}$$

$$\rho_T(A) = (1)^a \equiv (\epsilon^{T q(\omega)})^a_{\text{ref}} = (\epsilon^{T q(\omega)})^a_{\text{ref}} (\epsilon^{T \delta q(\omega)})^a_{\text{ref}}. \tag{S12}$$
Here $e^{Tq(\omega)} = p(\omega)/\tilde{p}(\omega)$ is the reweighting factor (also known as the likelihood ratio or Radon-Nikodym derivative [16, 54]). We have

$$q(\omega) = T^{-1} \ln \frac{p(\omega)}{\tilde{p}(\omega)} = -T^{-1} \sum_{n=0}^{N-1} q_{x,n,x_{n+1}},$$

where

$$q_{x,n,x_{n+1}} = \ln \frac{W_{x_n,x_{n+1}}}{W_{x_n,x_{n-1}}}. \Delta t_n (R_{x_n} - \tilde{R}_{x_n}).$$

Here $\Delta t_n = -\ln \eta/\tilde{R}_{x_n}$ is the jump time of the reference model ($\eta$ is a random number uniformly distributed on $(0, 1)$). In (S12) the quantity $\delta q(\omega) = q(\omega) - \langle q(\omega) \rangle_{\text{ref}}$. Taking logarithms of (S12) and the large-T limit gives us

$$J(\tilde{a}_0) = J_0(\tilde{a}_0) + J_1(\tilde{a}_0),$$

where

$$J_0(\tilde{a}_0) = -\langle q(\omega) \rangle_{\tilde{a}_0 \text{ref}}$$

and

$$J_1(\tilde{a}_0) = -\frac{1}{T} \ln \langle e^{T\delta q(\omega)} \rangle_{\tilde{a}_0 \text{ref}}.$$}

In these expressions $\tilde{a}_0$ is the typical value of $\alpha$ for the reference model. The term (S16) is by Jensen’s inequality an upper bound on the piece of the rate function $J(\alpha)$ at the point $\alpha = \tilde{a}_0$, i.e.,

$$J(\tilde{a}_0) \leq J_0(\tilde{a}_0).$$

The bound can be determined by computing the values of (S16) and (S17) for a suitably long reference-model trajectory. The correction term (S17) can be determined by sampling values of $q$ for the reference model, provided that the typical dynamics of the reference model is close enough to the desired conditioned dynamics of the original model [55].

If the reference model’s typical dynamics is similar to the conditioned rare dynamics of the original model (something we generally do not know in advance), then the bound $J_0(\tilde{a}_0)$ will be tight, and if it is tight enough the exact value of $J(\tilde{a}_0)$ can be calculated by sampling the (slightly) atypical behavior of the reference model [56].

The optimal reference model, called the driven or auxiliary process [16, 40, 45, 54], is one for which the bound is exact, meaning that its typical behavior is equivalent to the conditioned rare behavior of the original model. In the main text we show that evolutionary reinforcement learning can generate optimal or near-optimal reference models, for which the correction term is negligible.

**S2. 4-STATE MODEL EVOLUTIONARY PROCEDURE**

The evolutionary procedure used to make Fig. 1 of the main text is as follows. We start by running a trajectory of the reference model (of $N = 10^4$ events) and recording the typical value of the observable and bound, the long-time limits of (11) and (3), respectively. Initially the reference model is the original model, $\tilde{W}_{xy} = W_{xy}$, and so $a = \tilde{a}_0$ and $J_0 = 0$. To perform an evolutionary step we create a mutant model whose rates are

$$\tilde{W}_{xy} = e^{(\eta_{xy}-1/2)}W_{xy}.$$}

Here $\epsilon$ is an evolutionary rate and $\eta_{xy}$ is a uniformly distributed random number on $(0, 1)$. With this new set of rates we run a new trajectory and compute the new values of $a$ and $J_0$, called $\tilde{a}$ and $\tilde{J}_0$, respectively. If our selection criteria are fulfilled (see below) then we accept the mutation, and set $\tilde{W}_{xy} = W_{xy}$, $a = \tilde{a}$, and $J_0 = \tilde{J}_0$ (the mutant model becomes the new reference model); if not, we retain the current reference model.

We imposed two types of selection criteria. For the first, called $a$-evolution, we accepted the mutation if $\tilde{a}$ is closer than $a$ to a specified target value $a^\star$, i.e., if

$$|\tilde{a} - a^\star| < |a - a^\star|.$$}

For the second, called $J$-evolution, we accept the mutation if $\tilde{J}_0$ is smaller than $J_0$ and if $\tilde{a}$ lies within a tolerance $\delta = 0.1$ of a specified pinning value $a^\dagger$, i.e., if

$$\tilde{J}_0 < J_0 \quad \text{and} \quad |\tilde{a} - a^\dagger| < \delta.$$}

The process of $a$-evolution leads to reference models able to generate values of $a$ far from $\tilde{a}_0$, while $J$-evolution leads to reference models that generate values of $a$ in a manner as close as possible to the original model.

![FIG. S1. Similar to Fig. 1 of the main text, using an evolutionary scheme in which we alternate 5 steps of $a$-evolution with $N_{\text{ev}}$ steps of $J$-evolution, for different target values of the observable $a$. The results reported in Fig. 1 were produced from the set $N_{\text{ev}} = 50$ using an additional $10^5$ steps of $J$-evolution once the desired value of $a$ was achieved.](image)
propose a set of weights was accepted if

$$|\hat{a} - a^*| < |a - a^*| \quad \text{and} \quad \hat{a} < a + \epsilon. \quad (S23)$$

We then did $J$-evolution using a tolerance of $\delta = 0.02$ [see Eq. (S21)], for $3 \times 10^4$ proposed trajectories, with higher-order spin filters applied. We ran 50 simulations, each with a different target value of $a$.

In Fig. S2 we reproduce some of the results shown in Fig. S3(a) of the main text, together with results obtained for different values of $K$.

We also tested the method on the 100-site FA model of Ref. [17] (for $c = 0.1$), whose state space is large enough that state-of-the-art methods are needed to compute its large-deviation rate function. Initial tests done using evolutionary learning on the network shown in Fig. 2 (with $K = 5$) produced a bound that was inexact, but close enough to the exact answer to suggest that modifications of that scheme can converge to the correct answer. We therefore replaced the network shown in Fig. 2 of the main text with the one shown in Fig. S3(a). Each hidden node in the network couples to $K$ input nodes (lattice sites), and takes one of $2^K$ values. This value is determined by the state of the $K$ spins (in microstate $x$) to which it is connected, via binary arithmetic:

$$h_x(i) = \sum_{m=0}^{K} 2^m \frac{1 + S_{i+m}^x} {2} \quad (S24)$$

(here we assume periodic boundaries). The output of the network in microstate $x$ is then

$$f_x = \sum_{i=1}^{L} w_{h_x(i)}, \quad (S25)$$

where the $2^K$ weights $w_{h_x(i)}$ are, along with $w_0$, the trainable parameters of the model. This filter is capable of learning which features (spin patterns) are most significant; by contrast, the filters shown in Fig. 2 look only for homogenous blocks of spins. The reference-model ansatz is again (5), but now with (S25) replacing (6).

We ran 40 evolutionary simulations with target values of $a$ between 0.1 and 30 (the typical value of the original model is approximately 3.5). We turned on the neural network from the start, and used trajectories of $N = 2 \times 10^5$ events. We used $a$-evolution (using Eq. (S20)) to generate the desired values of $a$, and then did $J$-evolution for $\approx 3 \times 10^4$ proposed trajectories. Results are shown in Fig. S3(b): for $K \geq 4$, the bound is numerically close to the exact answer. We show space-time plots of reference model dynamics in panel (c).

FIG. S2. (a) Similar to Fig. 3(a) of the main text, showing results for neural-network spin filters up to order $K = 2, 4, 7$. For $K \geq 4$ the bound is numerically close to the exact answer. Also shown is the CMP universal activity bound [66] (gray), which results from the typical dynamics of a reference model whose rates are uniform multiples of those of the original model, and the $(\lambda, c)$ bound of Ref. [55] (blue). The latter is essentially equivalent to the case $K = 1$.

To produce Fig. 1 we alternated 5 steps of $a$-evolution, using an evolutionary rate of $\epsilon = 0.1$, with 50 steps of $J$-evolution, using an evolutionary rate of $\epsilon = 0.05$. During $J$-evolution we chose the pinning value $a^+$ to be the last value of $a$ produced by the preceding phase of $a$-evolution. Upon reaching a specified value $a^*$ we carried out an additional $10^5$ steps of $J$ evolution (with $a^+ = a^*$), again using $\epsilon = 0.05$. We carried out 100 independent simulations, each with a different target value $a^*$.

Fig. 1 shows results obtained in this way; Fig. S1 shows results obtained using fewer steps of $J$-evolution.

S3. FA MODEL EVOLUTIONARY PROCEDURE

Each proposed evolutionary move consisted of a shift of each weight $w \in \{w_0, w_1, w_R\}$ of the reference model [5] by independent Gaussian-distributed random numbers of zero mean and variance $\sigma^2 = 10^{-4}$:

$$w \rightarrow w + N(0, \sigma^2). \quad (S22)$$

We ran trajectories for $N = 10^5$ events, and recorded the values of $\{\hat{a}\}$ and $\{\hat{a}\}$ after each proposed trajectory. We did $a$-evolution on the parameters $w_0$ and $w_1$ until a specified value $a^*$ was reached. This procedure was as described for the 4-state model, with the additional restriction that the new bound must be not more than a value $\epsilon = 0.2$ larger than the current bound. That is, the
S4. THE CMP UNIVERSAL ACTIVITY BOUND

The Conway–Maxwell–Poisson (CMP) formula

\[ J[\text{CMP}] = \frac{k_0}{a_0} \left( a \ln \frac{a}{a_0} + a_0 - a \right), \tag{S26} \]

gives a bound on the large-deviation rate function \( J(a) \) for any non-decreasing counting observable \( a; \) here \( a_0 \) is the typical value of the observable, and \( k_0 \) is the typical dynamical activity \( k \) (the total number of configuration changes per unit time). Eq. \( \text{(S26)} \) was derived in Ref. \([66]\) from Level 2.5 of large deviations \([74, 75]\), and we have used this form in Fig. \(3\) of the main text, Fig. \(S2\) and Fig. \(S3\) (for the case \( a = k \)).

We note that the CMP formula can also be straightforwardly derived from the bound \([3]\). Let \( a_0 \) and \( k_0 \) be the typical activities produced by an original model \( W_{xy} \). Then a reference model \( \tilde{W}_{xy} = \gamma W_{xy} \), whose rates are uniformly rescaled versions of those of the original model, will produce typical activities \( \gamma a_0 \) and \( \gamma k_0 \) (a uniform rescaling of rates does not affect the choice of new state, i.e. \( \tilde{W}_{xy}/\tilde{R}_x = W_{xy}/R_x \), and so the reference model will visit the same set of states as the original model, just faster or slower).

In \([3]\) we can replace the fluctuating jump time \( \tilde{\Delta}t_n \) with its mean \( 1/\tilde{R}_{x_n} \), valid in the long-time, steady-state limit, and replace \( R_{x_n} \) with \( \gamma R_{x_n} \), giving

\[ J_0 = T^{-1} \sum_{n=0}^{N-1} \left( \ln \gamma + \frac{1 - \gamma}{\gamma} \right) \]

\[ = k \left( \ln \frac{a}{a_0} + \frac{a_0 - a}{a} \right) \]

\[ = \frac{k}{a} \left( a \ln a - a_0 - a \right) \]

\[ = \frac{k_0}{a_0} \left( a \ln \frac{a}{a_0} + a_0 - a \right), \tag{S27} \]

which is Eq. \( \text{(S26)} \). In deriving \( \text{(S27)} \) we have written \( k = N/T \) for the total number of configuration changes per unit time, and have used the relations \( k = \gamma k_0 \) and \( a = \gamma a_0 \).