# A deep learning functional estimator of optimal dynamics for sampling large deviations

View the article online for updates and enhancements.

MACHINE
LEARNING
Science and Technology

**PAPER**

# A deep learning functional estimator of optimal dynamics for sampling large deviations

Tom H E Oakes[1,2] ⓘ, Adam Moss[1] and Juan P Garrahan[1,2]

[1] School of Physics and Astronomy, University of Nottingham, Nottingham NG7 2RD, United Kingdom
[2] Centre for the Mathematics and Theoretical Physics of Quantum Non-Equilibrium Systems, University of Nottingham, Nottingham NG7 2RD, United Kingdom

**E-mail:** tom.oakes@nottingham.ac.uk

## Abstract

In stochastic systems, numerically sampling the relevant trajectories for the estimation of the large deviation statistics of time-extensive observables requires overcoming their exponential (in space and time) scarcity. The optimal way to access these rare events is by means of an auxiliary dynamics obtained from the original one through the so-called 'generalised Doob transformation'. While this optimal dynamics is guaranteed to exist its use is often impractical, as to define it requires the often impossible task of diagonalising a (tilted) dynamical generator. While approximate schemes have been devised to overcome this issue they are difficult to automate as they tend to require knowledge of the systems under study. Here we address this problem from the perspective of deep learning. We devise an iterative semi-supervised learning scheme which converges to the optimal or Doob dynamics with the clear advantage of requiring no prior knowledge of the system. We test our method in a paradigmatic statistical mechanics model with non-trivial dynamical fluctuations, the fully packed classical dimer model on the square lattice, showing that it compares favourably with more traditional approaches. We discuss broader implications of our results for the study of rare dynamical trajectories.

## 1. Introduction

In this paper we consider the problem of efficiently sampling rare trajectories of stochastic systems. A natural approach to studying stochastic dynamics is by means of trajectory ensemble methods, e.g. [1–18] (for reviews see [19–24]). For the statics, the ensemble method of equilibrium statistical mechanics [25] deals with sets of configurations or microstates and their probabilities of occurring under equilibrium conditions and connects their statistical properties to macroscopically observable phenomena. The straightforward generalisation to dynamics [19–24] is to replace microstates by stochastic trajectories, defining the ensemble in terms of all possible trajectories of the dynamics and the probabilities of them occurring under stochastic evolution. In the dynamical setting, order parameters (meaning macroscopic quantities that classify physical behaviour) correspond to observables of the whole trajectories, while the large size limit involves also the long time limit. Just like in the static case, in such 'thermodynamic' regime large deviation (LD) principles may apply [19–24], so that the statistics of observables become encoded in LD functions that play the role of free-energies for the dynamics (see below for definitions).

The usual setting is of a system with Markovian dynamics defined via its evolution operator (such as the exponential of a Markov generator for continuous-time dynamics) and one or more trajectory observables that one wishes to study [19–24]. As usual, physical intuition about the problem at hand will suggest that these are appropriate dynamical order parameters to describe the phenomenon of interest. For example they could correspond to macroscopic (particle or energy) currents for driven problems such as exclusion processes, or dynamical activities for systems with slow dynamics such as glasses. The aim is to calculate the probability distributions of these dynamical quantities. These distributions are 'physical' in the sense that they are in principle observable with access to enough statistics of the dynamics.

In practice, accessing the LD statistics of dynamical observables is challenging in general [26–38]. The extensive nature of the observables makes them exponential (in system size and time) concentrated around their typical or average value. That is, to observe trajectories other than typical is exponentially hard if sampled with the dynamics that generates them. Alternatively, the LD statistics is encoded in the spectrum of a deformation - or tiltings (see below for definitions) - of the evolution operator, but diagonalising such tilted operators is not possible except when the state space is of modest size.

Many techniques have been developed to overcome these difficulties. Most of these approaches attempt to sample rare long-time trajectories efficiently by improving their exponential scarcity within the original dynamics. Two popular methods are based either on population dynamics, such as cloning or splitting [26–28, 30], or on importance sampling in trajectory space, such as as transition path sampling (TPS) [1, 7]. While these tame some of the exponential cost, further sampling improvements are needed for computational tractability in many systems of interest. Recently various schemes based on umbrella sampling in trajectory space have been shown to be useful [32–34], together with related adaptive schemes based on optimal control [31, 35, 39] (more on both of these below). An alternative approach is via variational approximations, for example directly at the level of LD rate functions [38], or by using tensor networks to find leading eigenvectors of tilted generators [29, 36, 37].

While efficient for many problems of interest, the methods above often require some information on the problem being studied. Improving cloning or TPS with trajectory umbrella sampling implies the choice of an alternative sampling dynamics. While an optimal one is known to exist (via the so-called generalised Doob transformation [9, 16, 40], see below), to find it requires solving the problem in the first place. Adaptive methods [31, 35] attempt to find an approximation to this optimal dynamics, but for systems of interest a suitable parameterisation needs to be selected and this is also problem dependent. Alternatives such as those based on variational tensor networks [29, 36, 37], which in principle are generic, are in practice often restricted to one dimensional problems with short range interactions.

Here we address the problem of finding in an automatic manner - i.e. without the usual physical input that goes into the parameterisation - an efficient auxiliary dynamics, close to the optimal Doob dynamics, for sampling rare trajectories in the LD regime. We address this problem from the perspective of deep learning [41, 42], since this appears to be an unsupervised learning (USL) task of unknown features extractable from fluctuations of the original system dynamics. Many sampling techniques, like Monte Carlo (MC), rely on a form of encoding the configuration into either a unique descriptor or some macroscopic value. The task of converting a configuration (or image) into a single value which encodes the likelihood of that configuration from a given distribution is clearly a deep learning problem [43–45]. What we know from studies like [43] is that Neural Networks (NN) can encode more of these physical features than merely a label of which distribution the sample was taken from. Therefore given the right architecture and learning task a NN can be used to condense a configuration into a multitude of different parameters. In the past few years, there have been approaches that take advantage of these NN's to speed up equlibration via sampling methods such as MC [46–49]. This type of approach to MC modification is perfectly framed for finding the auxiliary dynamics needed for the LD regime, using only a configuration as input and an appropriate learning process.

We focus for concreteness on a paradigmatic system with rich dynamics, namely, the fully packed classical dimer model (CDM) on the square lattice [50, 51], which is known to display a range of interesting dynamical LD behaviour [52]. We construct an iterative USL scheme which in progressively approximates with increased precision the optimal Doob dynamics for sampling. The clear advantage of our approach is that it does not require any *a priori* knowledge of the system. For the CDM we show that our method compares favourably with more standard approaches.

The paper is organised as follows. In section 2 we review the main concepts about trajectory ensembles and dynamical large deviations. In section 3 we describe the classical dimer model which we use as a testbed for our ideas and describe the issues relating to sampling rare trajectories. Sections 4 and 5 present our main results. In section 4 we consider the problem of approximating an optimal dynamics for LD sampling via neural networks and supervised learning. It established the main principles and the appropriate architecture for the NNs. Subsequently, in section 5 we generalise the methods via semi-supervised learning to make them scalable and thus applicable to systems for which exact diagonalisation is not possible. Finally in section 6 we present our conclusions and outlook.

## 2. Trajectory ensembles and dynamical large deviations

### 2.1. Continuous-time Markov dynamics and trajectories

For concreteness, we will focus on systems with stochastic Markovian dynamics in continuous time. The issues we discuss, however, are easily generalisable to discrete time dynamics. As a general setting we consider

a system with configurations $\{x_1, \cdots, x_{\mathcal{N}}\}$ where $\mathcal{N}$ indicates the size of configuration space (for example $\mathcal{N} = 2^N$ for a system of $N$ Ising spins).

The master equation for the evolution of the probability over configurations can be written in general as

$$\partial_t |P_t\rangle = \mathbb{W} |P_t\rangle, \tag{1}$$

where $\{|x\rangle\}$ is an orthonormal configuration basis, $|P_t\rangle$ is the probability vector,

$$|P_t\rangle = \sum_{n=1}^{\mathcal{N}} P_t(x) |x_n\rangle, \tag{2}$$

and $P_t(x)$ the probability of configuration $x$ at time $t$. Throughout the paper we use the standard physics notation of 'bra-kets' for vectors. Specifically, the ket $|P_t\rangle$ represents a column vector

$$|P_t\rangle = \begin{pmatrix} P_t(x_1) \\ P_t(x_2) \\ \vdots \\ P_t(x_{\mathcal{N}}) \end{pmatrix} \tag{3}$$

obtained from equation (2) given that ket $|x_n\rangle$ represents the column basis vector

$$|x_n\rangle = \begin{pmatrix} \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix}, \tag{4}$$

where the only non-zero entry is at position $n$. Conversely the bra $\langle x_n|$ represents the row basis vector

$$\langle x_n| = \begin{pmatrix} \cdots & 0 & 1 & 0 & \cdots \end{pmatrix}, \tag{5}$$

where again the only non-zero entry is at position $n$.

The Markov generator of the dynamics (sometimes also called master operator) is given by

$$\mathbb{W} = \sum_{x,x' \neq x} w_{x \to x'} |x'\rangle \langle x| - \sum_x R_x |x\rangle \langle x|, \tag{6}$$

The positive terms are off-diagonal and encode the possible transitions $x \to x'$ and their rates $w_{x \to x'}$. The negative terms are diagonal, with $R_x$ the escape rate from configuration $x$, $R_x = \sum_{x' \neq x} w_{x \to x'}$. The form (6) guarantees probability conservation: the largest eigenvalue of $\mathbb{W}$ is zero and its left eigenvector is the *uniform* (or 'flat') state:

$$\langle -| \mathbb{W} = 0, \qquad \qquad \langle -| = \sum_x \langle x|. \tag{7}$$

In many cases, the dynamics generated by (6) also has a stationary state,

$$\mathbb{W} |ss\rangle = 0, \qquad \qquad |ss\rangle = \sum_x P_{ss}(x) |x\rangle, \tag{8}$$

where $P_{ss}(x)$ denotes the (time independent) stationary state probability over configurations. The generator (6) defines a continuous time Markov chain. Samples from such dynamics are stochastic trajectories, corresponding to a sequence of configurations and jumps between them at random times,

$$\omega_t = (x_0 \to x_{t_1} \to \ldots \to x_{t_K}), \tag{9}$$

where $t_i$ ($i = 1, \ldots, K$) indicate the times at which jumps between configurations occur, with time ordering $0 < t_1 < \cdots < t_K < t$ and we use the symbol $\omega_t$ to label a trajectory of total time extension $t$. Between jumps the configuration does not change, meaning that from the time of the last jump, $t_K$ and the final time $t$, the configuration in (9) remains as $x_{t_K}$. The set of all possible trajectories $\omega_t$ generated by dynamics for, say, fixed total time extent $t$, together with their probability of occurring, $\pi(\omega_t)$, defines the (original or untilted, see below) ensemble of trajectories.

## 2.2. Dynamical large deviations

We are interested in the statistics of trajectory observables, i.e. functions of the whole trajectory which are extensive both in time and size of the system. One of the simplest is the *dynamical activity* [3, 4, 8, 53] defined as the number of configuration changes in a trajectory, $K(\omega_t)$. The distribution of these dynamical observables is obtained by contraction from that of the trajectories

$$p_t(K) = \sum_{\omega_t} \pi(\omega_t) \boldsymbol{\nabla} ta[K(\omega_t) - K] . \tag{10}$$

Given that the trajectories are independent and identically distributed samples from the ensemble of trajectories we can say for long times $p_t(K)$ obeys a large deviation (LD) principle

$$p_t(K) \sim e^{t\varphi(K/t)} , \tag{11}$$

where the ~ sign is used to indicate that the left hand side approximates the right hand side up to factors which are exponentially subleading in time. Specifically

$$\varphi(k) = \lim_{t \to \infty} t^{-1} \ln p_t(kt) . \tag{12}$$

The (extensive in space) function $\varphi(k)$ is often called the rate function and plays the role of an entropy density for trajectories. One can alternatively consider the statistics of $K$ in terms of its moment generating function (MGF) [19–24],

$$Z_t(s) = \sum_K P_t(K) e^{-sK} = \sum_{\omega_t} \pi(\omega_t) e^{-sK(\omega_t)}, \tag{13}$$

which also has a LD form at long times

$$Z_t(s) \sim e^{t\theta(s)}. \tag{14}$$

The scaled cumulant generating function(SCGF) $\theta(s)$ is a free-energy (per unit time, but extensive in system size) for trajectories, while $Z_t(s)$ is the corresponding trajectory partition sum. As in static thermodynamics, the rate function and SCGF are connected via a Legendre transform [19–24],

$$\theta(s) = -\min_k [\varphi(k) + sk] . \tag{15}$$

## 2.3. Exponential tilting and biased ensembles

The trajectory partition function (13) encodes an exponential tilting of the probabilities of trajectories. That is, it is the normalisation of a reweighing of the probability of a trajectory through the value of the observable $K$,

$$\pi_s(\omega_t) = \frac{\pi(\omega_t) e^{-sK(\omega_t)}}{Z_t(s)}. \tag{16}$$

These tilted probabilities correspond to a biased trajectory ensemble, where averages of arbitrary trajectory observables are given by

$$\langle A \rangle_s = \frac{\langle A e^{-sK} \rangle}{Z_t(s)} = \frac{\sum_\omega A(\omega) \pi(\omega) e^{-sK(\omega)}}{Z_t(s)}$$
$$= \sum_\omega A(\omega) \pi_s(\omega). \tag{17}$$

At $s = 0$ these correspond to the averages under the process dynamics. For $s \neq 0$, they do not, but by tuning $s$ the exponential tilting allows to (formally) explore the properties of atypical trajectories.

The dynamical partition sum (13) can be written in terms of a 'transfer matrix',

$$Z_t(s) = \langle -|e^{t\mathbb{W}_s}|\text{i} \rangle , $$

where the probability vector $|\text{i}\rangle$ represents the distribution from which the initial state is drawn. The *tilted generator* $\mathbb{W}_s$ is a deformation of the original dynamical generator [19–24]. For the specific case of tilting against the activity (generalisations are straightforward) we have

$$\mathbb{W}_s = e^{-s} \sum_{x,x' \neq x} w_{x \to x'} |x'\rangle \langle x| - \sum_x R_x |x\rangle \langle x| . \tag{19}$$

The SCGF is the largest eigenvalue of $\mathbb{W}_s$,

$$\mathbb{W}_s |r_s\rangle = \theta(s) |r_s\rangle, \quad \langle \ell_s | \mathbb{W}_s = \theta(s) \langle \ell_s |, \tag{20}$$

where $|r_s\rangle$ and $\langle \ell_s |$ are the corresponding right and left eigenvectors.

### 2.4. Basic considerations on sampling exponentially tilted ensembles

The spectral structure (20) provides an important simplification. It converts the problem of estimating the LD statistics into one of minimising an operator (or finding the minimum eigenvalue) - something that is routine in physics for example in the context of finding ground states of quantum Hamiltonians. However, diagonalising $\mathbb{W}_s$ is often impractical in many-body systems once size exceeds modest values. In most cases of interest it is therefore necessary to resort to numerical approximation schemes.

Consider the problem of calculating averages such as (17). Doing so using the original dynamics, (6), is extremely impractical, as the exponential factor in the first line of (17) makes convergence exponentially (in space and time) hard. Ideally one would like to sample $\pi_s(\omega)$ directly, but there is no easy way to generate trajectories compatible with (16) starting from the known dynamics (6).

The first step in addressing the sampling problem is to account for the exponential factor in the first line of (17) in a systematic manner, by means of importance sampling in trajectories. One such scheme is transition path sampling (TPS), see discussion below. It amounts to doing a biased random walk in trajectory space guaranteed to converge asymptotically to the biased trajectory ensemble (16). TPS will form the basis of the numerical scheme we develop below.

## 3. Trajectory sampling and classical dimer model

### 3.1. Transition path sampling, trajectory umbrella sampling and optimal reference dynamics

As discussed in section 2.4, for many-body system - including the example of the classical dimer model (CDM) that we discuss in detail below - accessing the LD statistics of a relevant dynamical order parameter such as the activity for large systems sizes and long times is non-trivial. Sampling of rare trajectories requires a strategy that overcomes the fact that they are exponentially suppressed both in space and in time. This is the central consideration of this paper.

The first step is to choose a basic scheme to do importance sampling of trajectories, that is, to improve beyond the computational trap of brute force sampling of exponentially rare trajectories with the original dynamics. For this purpose we employ transition path sampling (TPS) [1]. We consider specifically TPS as formulated for the study of LDs in time-translation invariant dynamics, see e.g. [7, 52] (rather than as originally devised to access rare trajectories in transition pathways where the rare events are conditioned to start in one region of phase space and end in another). Alternative numerical methods include 'cloning' (or 'splitting') [30], variational approximations [38], or tensor network approaches [36, 37].

TPS is Monte Carlo (MC) sampling of trajectories. Just like MC for configurations, TPS is a form of 'importance sampling', progressively guiding the trajectories it samples towards the relevant region of trajectory space according to the desired target measure, in our case the tilted distribution equation (16). Like in standard MC there are two relevant basic steps in the TPS algorithm: (i) proposal of trial moves and (ii) acceptance or rejection. For (i) new trajectories are proposed typically via 'shooting' and/or 'shifting' moves [1], where sections of the current trajectory are regenerated to create the proposed new trajectory. For (ii), a Metropolis rule if often employed, whereby the probability of acceptance is given by $\min\left(1, e^{-s\Delta K}\right)$, where $\Delta K$ is the change in the observable (e.g. the activity) between the current and proposed trajectories. This approach is guaranteed to converge to trajectories sampled from (16).

While TPS (or alternative methods such as cloning) improve the efficiency of sampling with respect to brute force, for many-body systems they may still suffer from slow convergence, especially near LD transitions. In the case of TPS this is related to the proposal of trial trajectories and their low acceptance, especially for long times in systems with large state spaces. To improve convergence one can use umbrella sampling in trajectory space [32–34, 52]. If we consider an exponentially tilted average such as the numerator of (17) we can write

$$\langle A e^{-sK} \rangle = \sum_\omega \pi_{\text{ref}}(\omega) A(\omega) e^{-sK(\omega)} \frac{\pi(\omega)}{\pi_{\text{ref}}(\omega)}$$

$$= \langle A e^{-sK+G} \rangle_{\text{ref}}, \tag{21}$$

where

$$G(\omega) = \log \frac{\pi(\omega)}{\pi_{\text{ref}}(\omega)}. \tag{22}$$

This means that we can estimate (17) using a different dynamics, the 'reference', if we exponentially weight according also to the trajectory observable $G$ to account for the change in measure [32–34, 52]. A clever choice of reference may reduce the exponential sampling error. In fact, the optimal choice for the reference dynamics is given via a generalised Doob transformation [9, 16], which removes the exponential cost of the sampling, basically by cancelling the $-sK$ in the exponent of (21) with the reweighing term $G$.

The alternative dynamics has generator

$$\mathbb{W}_{\text{ref}} = \sum_{x,x' \neq x} w^{\text{ref}}_{x \to x'} |x'\rangle \langle x| - \sum_x R^{\text{ref}}_x |x\rangle \langle x| \,, \tag{23}$$

with the following restriction: $w^{\text{ref}}_{x \to x'} \neq 0$ if $w_{x \to x'} \neq 0$, that is, the reference dynamics connects (with in general different rates) the same configurations as the original dynamics; and, of course, to maintain stochasticity, the escape rates are given by $R^{\text{ref}}_x = \sum_{x'} w^{\text{ref}}_{x \to x'}$. The reweighing factor for a trajectory $\omega_t$ is then

$$e^{G(\omega_t)} = \frac{\pi(\omega_t)}{\pi_{ref}(\omega_t)}$$

$$= \prod_{x,y \neq x} \left( \frac{w_{x \to x'}}{w^{\text{ref}}_{x \to x'}} \right)^{K_{x \to x'}(\omega_t)} e^{\int_0^t dt' \Delta R_{x_{t'}}} \,, \tag{24}$$

where $K_{x \to x'}(\omega)$ indicates the total number of jumps between configurations $x$ and $x'$ in trajectory $\omega$ and $\Delta R_{x_{t'}} = R^{\text{ref}}_{x_{t'}} - R_{x_{t'}}$ with $x_{t'}$ the configuration in the trajectory at time $t'$. Trajectory averages with the reference dynamics, (21), can be computed with TPS by adjusting the acceptance criterion. If the old trajectory was $\omega$ and the proposed one $\omega'$ the probability of acceptance $\Gamma_{\text{acc}}(\omega \to \omega')$ reads

$$\Gamma_{\text{acc}}(\omega \to \omega') = \min\left( 1, \frac{e^{-sK(\omega') + G(\omega')}}{e^{-sK(\omega) + G(\omega)}} \right) \,. \tag{25}$$

In the limit of long times the optimal reference dynamics is given by the long-time Doob transformation. This optimal dynamics has generator $\tilde{\mathbb{W}}$ with transitions rates given by

$$\tilde{w}_{x \to x'} = \frac{\ell_{x'}}{\ell_x} e^{-s} w_{x \to x'} \,, \tag{26}$$

where the $\ell_x$ are the elements of the leading left eigenvector of $\mathbb{W}_s$, cf (20), and escape rates give by the original ones shifted by the SCGF

$$\tilde{R}_x = R_x - \theta(s) \,. \tag{27}$$

Note that this is true only for jump observable that are symmetric in transitioning from $x$ to $x'$ and back. If the reference dynamics is the one generated by $\tilde{\mathbb{W}}$ the reweighing factor then reads,

$$e^G = e^{t\theta(s)} e^{sK} \frac{\ell_{x_0}}{\ell_{x_t}} \,. \tag{28}$$

Note that with such reweighing the exponential of $K$ in (21) is removed, while the exponential of the SCGF cancels the normalisation in (17). In this way the exponential suppression disappears when using the optimal dynamics as the reference to calculate averages such as (17). Note however that the factor $\ell_{x_0}/\ell_{x_t}$ which only depends on the time-boundaries of the trajectory remains. For $s \neq 0$ where in general $\theta(s)$ is non-vanishing this will set an upper limit to the efficiency of the sampling with regards to TPS acceptance.

### 3.2. Arriving to the optimal dynamics iteratively through feedback

The key quantity in defining the optimal dynamics (26) is the left eigenvector $\langle \ell_s \|$, cf (20). In practice the problem is that to obtain (26) one needs to diagonalise $W_s$ first, something which is often not possible to do in systems of interest. The aim of what follows will be to find a way to estimate the components $\ell_x$ in an efficient manner.

We consider dynamics which obeys detailed balance which offers a useful simplification (this is the case of the CDM we study below). For systems with such dynamics we have in general

$$P_{\text{eq}}(x) w_{x \to x'} = P_{\text{eq}}(x') w_{x' \to x} \,, \tag{29}$$

where $P_{eq}$ is the equilibrium stationary distribution. In operator form this relation can be written as

$$\mathbb{W}^{\dagger} = \mathbb{P}_{eq} \mathbb{W} \mathbb{P}_{eq}^{-1}, \tag{30}$$

where $\mathbb{P}_{eq}$ is a diagonal matrix with entries $P_{eq}(x)$. For the case of an observable like the activity, the tilted operator also obeys a similar relation

$$\mathbb{W}_s^{\dagger} = \mathbb{P}_{eq} \mathbb{W}_s \mathbb{P}_{eq}^{-1}, \tag{31}$$

with the same matrix $\mathbb{P}_{eq}$.

The optimal dynamics (26) is obtained via a generalised (and long time) Doob transform [9, 16],

$$\tilde{\mathbb{W}} = \mathbb{L} \left[ \mathbb{W}_s - \theta(s) \right] \mathbb{L}^{-1}, \tag{32}$$

where the components $\ell_x$ are used to construct the diagonal matrix $\mathbb{L}$, such that $\langle - | \mathbb{L} = \langle \ell_x |$. The operator $\tilde{\mathbb{W}}$ obeys $\langle - | \tilde{\mathbb{W}} = 0$, as it generates the proper stochastic dynamics (26). The Doob dynamics also obeys detailed balance,

$$\tilde{\mathbb{W}}^{\dagger} = \tilde{\mathbb{P}} \tilde{\mathbb{W}} \tilde{\mathbb{P}}^{-1}, \tag{33}$$

where $\tilde{\mathbb{P}}$ is the diagonal matrix with entries $\tilde{P}(x)$ given by the equilibrium distribution of dynamics (26). Combining equations (31)–(33) we get

$$\mathbb{L}^2 = \mathbb{P}_{eq}^{-1} \tilde{\mathbb{P}}, \tag{34}$$

which in components gives the relation between the elements of the left eigenvector and the stationary probabilities of both the original dynamics and the optimal one,

$$\ell_x = \sqrt{\frac{\tilde{P}(x)}{P_{eq}(x)}}. \tag{35}$$

Equation (35) relates the objects we need to obtain to define the optimal dynamics, $\ell_x$, to two equilibrium distributions [16]. The first one is that of the original dynamics, which we either known exactly - as in the case of the CDM - or we can sample from direct simulation. The second one is the equilibrium probability of the Doob dynamics. This one we do not know, but we can *estimate* from TPS simulations. This forms the basis of an iterative feedback approach for obtaining the optimal dynamics similar to those used in references [31, 35, 39]:

- We consider a reference dynamics with a structure similar to (26)

$$w_{x \to x'}^{ref} = \frac{\ell_{x'}^{ref}}{\ell_x^{ref}} e^{-s} w_{x \to x'}. \tag{36}$$

We could begin for example with $\ell_x^{ref} = 1$ for all $x$ as a naive initial guess.
- We run TPS under this reference and estimate $\tilde{P}$.
- We use (35) to obtain a new set of $\ell_x^{ref}$, we insert them in (36) to define a new reference dynamics and repeat the sampling.

Via this procedure the reference dynamics is guaranteed to eventually converge to the optimal Doob dynamics. The practical limitations relate to the size of state space and the ability to estimate the necessary equilibrium densities, which we address in the following sections via machine learning techniques.

### 3.3. Model

For concreteness we will consider the dynamics of a paradigmatic statistical mechanics model, that of fully packed dimers on a square lattice - with periodic boundary conditions - or classical dimer model (CDM) [50–52]. Figure 1 sketches the model. It comprises of dimers which occupy two adjacent sites on a square lattice (with periodic boundary conditions) under the conditions that all sites are covered and dimers do not overlap. For dynamics with local changes to the configuration $x$ which 'flip' a pair of neighbouring parallel dimers (or *plaquette*), between horizontal to vertical and visa versa, without having any direct effect of the rest of the configuration, see figure 1.
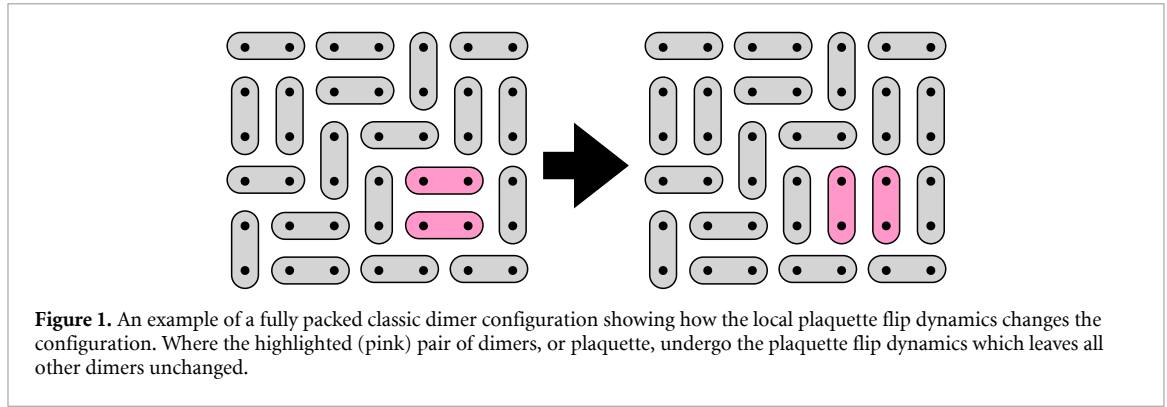
**Figure 1.** An example of a fully packed classic dimer configuration showing how the local plaquette flip dynamics changes the configuration. Where the highlighted (pink) pair of dimers, or plaquette, undergo the plaquette flip dynamics which leaves all other dimers unchanged.

The stochastic generator, cf (6), for the CDM can be written schematically

$$\mathbb{W} = \sum \left( |\text{II}\rangle\langle =| + | =\rangle|\text{II}\rangle \right) - \sum \left( |\text{II}\rangle\langle\text{II}| + | =\rangle\langle =| \right), \tag{37}$$

where the first sum contains all the non-diagonal elements that give rise to the plaquette transitions, while the second sum is diagonal and just counts them. Both the right eigenvector with eigenvalue zero (i.e. the equilibrium states) and the corresponding left eigenvector, equations (7)-(8), are given up to normalisation by the flat state over all allowed dimer configurations. As such, the dynamics generated by (37) is an 'infinite temperature' dynamics, i.e. one where the stationary state is one of maximum entropy.

The CDM is known [52] to have a LD first order transition when tilted by the dynamical activity, in this case given by the number of plaquette flips in a trajectory. This transition occurs at s = 0 in the large size limit. The corresponding tilted generator, cf (19), reads

$$\mathbb{W}_s = e^{-s} \sum \left( |\text{II}\rangle\langle =| + | =\rangle\langle\text{II}| \right) - \sum \left( |\text{II}\rangle\langle\text{II}| + | =\rangle\langle =| \right), \tag{38}$$

and the transition manifests in a first-order singularity in the SCGF $\theta(s)$ at $s = 0$ in the large size limit.

## 4. Optimal dynamics via neural networks: supervised learning and network architecture

The are several hurdles to overcome in order to implement the iterative feedback procedure of section 3.2. They all stem from the fact that configuration space grows exponentially with the system size, which in the case of the CDM is given by the number of sites in the lattice. For many-body systems like the CDM the aim is to study the problem for a range of relevant system sizes which immediately leads to a computational bottleneck that is difficult to solve.

Specifically, for the scheme of section 3.2 we face two related problems. First, in order to define a dynamics (36) one has to be able to specify $\ell_x^{\text{ref}}$ for all configurations $x$ of the system. As soon as the system exceeds modest sizes there are too many states to tabulate. Secondly, to use (35) to get $\ell_x^{\text{ref}}$ (that is, an approximation to the true $\ell_x$) not only one needs to have an estimate of $\tilde{P}(x)$ for all $x$ - the same problem as for $\ell_x^{\text{ref}}$ - but furthermore, this estimate has to be constructed from the states sampled in TPS trajectories, which are often much fewer than the total number of states (as is common in sampling).

These two problems are amenable to solution via standard machine learning methods [41, 42]: the first one we can address via *function approximation* whereby we represent the functions $\ell_x^{\text{ref}}$ and $\tilde{P}(x)$ by means of a neural network (NN) with an overall number of parameters that is much smaller than the dimension of configuration space; the second problem is one of *density estimation*, whereby from limited sampled data we approximate the distribution for all possible data. Of course, one can apply functional approximation and density estimation without using NNs [41, 42], through a choice of approximate description that depends on a small number of parameters. The problem is that if the approximation chosen does not contain the necessary physical information - for example when studying very different dynamical regimes - then it might have little effect on the sampling process. Using a NN as the approximator mitigates this problem by not hard-constraining physical assumptions.

### 4.1. Network structure
In terms of using a NN, learning how to take a configuration $x$ and map it to a single output $\ell_x^{\text{ref}}$ can be viewed as a regression problem. If a NN can be trained to provide a value of $\ell_x^{\text{ref}}$ for each $x$, then the NN can be incorporated in a continuous-time MC algorithm to provide values for the rates of potential moves.
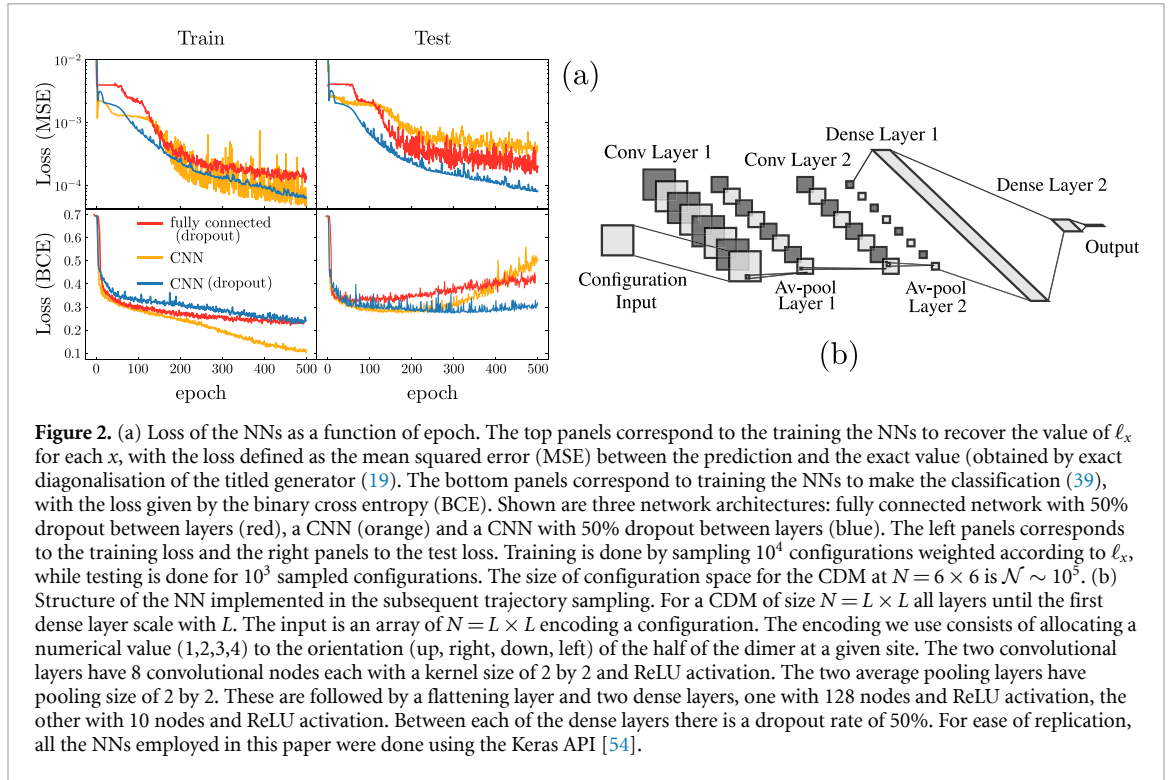
**Figure 2.** (a) Loss of the NNs as a function of epoch. The top panels correspond to the training the NNs to recover the value of $\ell_x$ for each $x$, with the loss defined as the mean squared error (MSE) between the prediction and the exact value (obtained by exact diagonalisation of the titled generator (19)). The bottom panels correspond to training the NNs to make the classification (39), with the loss given by the binary cross entropy (BCE). Shown are three network architectures: fully connected network with 50% dropout between layers (red), a CNN (orange) and a CNN with 50% dropout between layers (blue). The left panels corresponds to the training loss and the right panels to the test loss. Training is done by sampling $10^4$ configurations weighted according to $\ell_x$, while testing is done for $10^3$ sampled configurations. The size of configuration space for the CDM at $N = 6 \times 6$ is $\mathcal{N} \sim 10^5$. (b) Structure of the NN implemented in the subsequent trajectory sampling. For a CDM of size $N = L \times L$ all layers until the first dense layer scale with $L$. The input is an array of $N = L \times L$ encoding a configuration. The encoding we use consists of allocating a numerical value (1,2,3,4) to the orientation (up, right, down, left) of the half of the dimer at a given site. The two convolutional layers have 8 convolutional nodes each with a kernel size of 2 by 2 and ReLU activation. The two average pooling layers have pooling size of 2 by 2. These are followed by a flattening layer and two dense layers, one with 128 nodes and ReLU activation, the other with 10 nodes and ReLU activation. Between each of the dense layers there is a dropout rate of 50%. For ease of replication, all the NNs employed in this paper were done using the Keras API [54].

The optimal reference is the Doob dynamics (26). The first test for our NN approximator is therefore to learn the true form of the Doob transformation vector $\langle \ell_s |$. For CDM this means we are limited to a maximum lattice of size $N = 6 \times 6$ for which we are still able to obtain $\langle \ell_s |$ by exact diagonalisation. The supervised learning problem of training a NN to accurately reproduce the true components $\ell_x$ of $\langle \ell_s |$ given an input configuration $x$ for the exactly solvable $L = 6$ case will provide information about the correct network architecture for large scales.
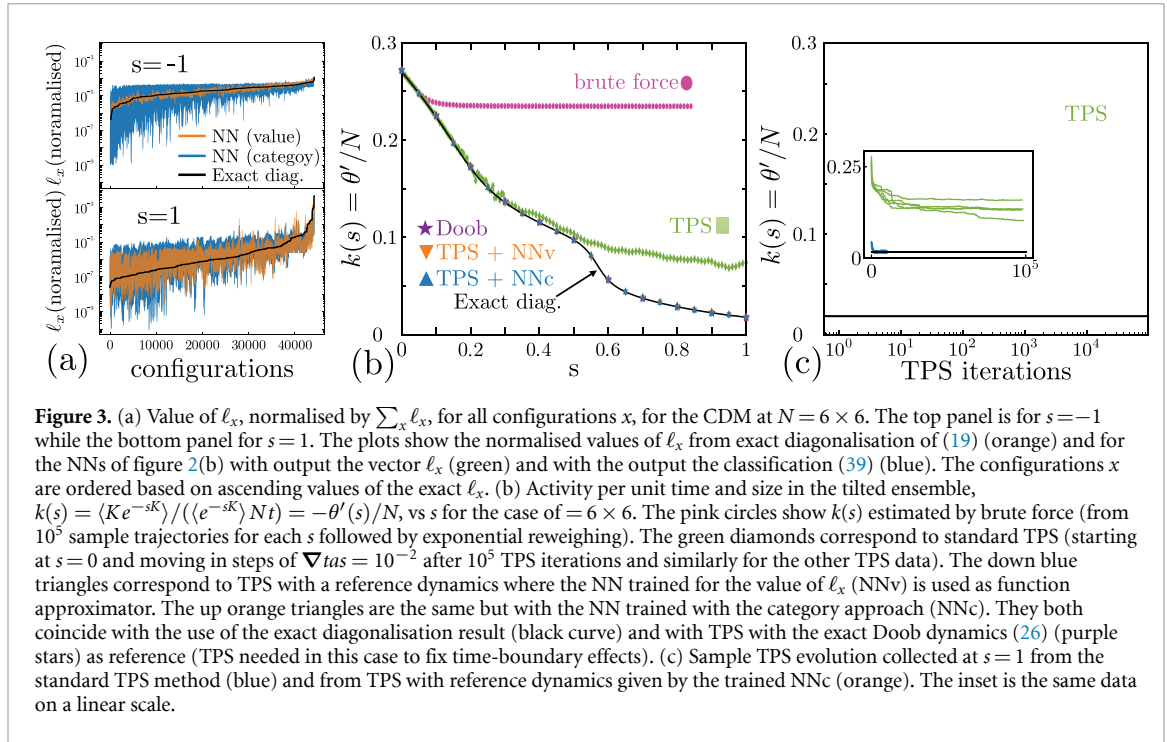
There are many hyper-parameters associated with finding a suitable network structure. Several possible schemes are compared in figure 2(a). One such is a NN with dense layers and regularisation such as dropout [41, 42]. However, if we consider configurations as highly correlated images then it makes sense the use of convolutional layers [41, 42]. We can see in figure 2(a) the effect of such layers, in combination with dropout regularisation, has on the training and generalisation of the model. Through this analysis we conclude that the convolutional neural network (CNN) [43–45] shown in figure 2(b) is an appropriate architecture for a functional approximator of $\ell_x$.

An important step is an adaptation of the loss function which will become very useful when we scale the problem up in the next section. The key expression for what follows is (35). What this equation shows is that in order to obtain the vector $\ell_x$ that defines the reference dynamics we are not interested in the individual probability distributions, $P_{\mathrm{eq}}$ and $\tilde{P}$, but on the ratio between them for each configuration $x$. This fact can be exploited using a method of learning that rephrases the issue of density estimation as a semi-supervised learning problem [55].

Specifically, consider a situation where we are generating sample configurations $x$ from two distributions, the equilibrium one $P_{\mathrm{eq}}$ corresponding to the original dynamics (6) and a second stationary distribution $\mathbb{H}atP$, for example an estimate of the true Doob equilibrium $\tilde{P}$. Consider the problem of training a NN to discriminate between configurations generated from $P_{\mathrm{eq}}$ or $\mathbb{H}atP$. That is, the output of a NN such as that of figure 2(b) could be a function $\mu(x) \in [0, 1]$ trained to reproduce

$$\mu(x) = \frac{\mathbb{H}atP(x)}{\mathbb{H}atP(x) + P_{\mathrm{eq}}(x)} = \frac{\frac{\mathbb{H}atP(x)}{P_{\mathrm{eq}}(x)}}{1 + \frac{\mathbb{H}atP(x)}{P_{\mathrm{eq}}(x)}} . \tag{39}$$

The above expression corresponds to the average of a binary label $Y$ which is equal to $Y = 1$ if $x$ is generated exclusively by $\mathbb{H}atP$ and $Y = 0$ if $x$ is generated exclusively by $P_{\mathrm{ref}}$. This is a standard classification problem. We can define our NN figure 2(b) with a sigmoid activation function before the output, corresponding to $\mu(x)$ and train it with data obtained in an i.i.d. from both $\mathbb{H}atP$ and $P_{\mathrm{eq}}$ and labelled

**Figure 3.** (a) Value of $\ell_x$, normalised by $\sum_x \ell_x$, for all configurations $x$, for the CDM at $N = 6 \times 6$. The top panel is for $s = -1$ while the bottom panel for $s = 1$. The plots show the normalised values of $\ell_x$ from exact diagonalisation of (19) (orange) and for the NNs of figure 2(b) with output the vector $\ell_x$ (green) and with the output the classification (39) (blue). The configurations $x$ are ordered based on ascending values of the exact $\ell_x$. (b) Activity per unit time and size in the tilted ensemble, $k(s) = \langle Ke^{-sK} \rangle / (\langle e^{-sK} \rangle Nt) = -\theta'(s)/N$, vs $s$ for the case of $= 6 \times 6$. The pink circles show $k(s)$ estimated by brute force (from $10^5$ sample trajectories for each $s$ followed by exponential reweighing). The green diamonds correspond to standard TPS (starting at $s = 0$ and moving in steps of $\nabla tas = 10^{-2}$ after $10^5$ TPS iterations and similarly for the other TPS data). The down blue triangles correspond to TPS with a reference dynamics where the NN trained for the value of $\ell_x$ (NNv) is used as function approximator. The up orange triangles are the same but with the NN trained with the category approach (NNc). They both coincide with the use of the exact diagonalisation result (black curve) and with TPS with the exact Doob dynamics (26) (purple stars) as reference (TPS needed in this case to fix time-boundary effects). (c) Sample TPS evolution collected at $s = 1$ from the standard TPS method (blue) and from TPS with reference dynamics given by the trained NNc (orange). The inset is the same data on a linear scale.

accordingly. Furthermore, from the output $\mu(x)$, we can obtain an estimate of the left eigenvector $\ell_x$ from the output of the NN by substituting (35) it into (39) and rearranging,

$$\ell_x = \sqrt{\frac{\mu(x)}{1 - \mu(x)}}. \tag{40}$$

Since this relationship between $\ell_x$ and $\mu$ arises due to (35) we can also see how it depends on detailed balance being present in the system of interest. In our case the CDM obeys this property.

### 4.2. Proof of principle: LD sampling for $N = 6 \times 6$

As a proof of principle on the use of a NN for the function approximator of the reference dynamics we consider the CDM for size $N = 6 \times 6$ and periodic boundaries, which is solvable by exact diagonalisation. In this case we can calculate exactly the vector $\ell_x$ for all configurations $x$. Figure 3(a) shows the ability of a NN like that of figure 2(b) to learn the exact $\ell_x$. We show for comparison the NN trained to reproduce the classification (39) (note that for the CDM the equilibrium probability $P_{\rm eq}$ is uniform over all configurations), with a similar network trained to reproduce the actual value of $\ell_x$. While the classification/category NN (NNc) gives larger fluctuations than the value NN (NNv), as we will see below, it provides a method that can be applied to system sizes too large for the value method.

The aim is to use this network within a TPS simulation as an aid to define the reference dynamics. Figure 3(b) compares the results from various sampling schemes with the exact diagonalisation of the tilted generator (19) for the $N = 6 \times 6$. We plot the average activity per unit time

$$k(s) = -\theta'(s) \tag{41}$$

against $s$, cf [52]. The sampling data is shown for the same number of generated trajectories ($10^5$) for each $s$, in order to compare their efficiency. The pink circles show brute force sampling via post-selection: as expected, the exponential cost of sampling away from $s = 0$ makes the estimation of the true $k(s)$ very poor. The green diamonds are results obtained via standard TPS, where trajectories are generated using the original dynamics: while this is an improvement over brute force sampling, for $s$ large enough the simulation fails to reproduce the exact values (black curve). The down blue triangles and up orange triangles correspond to TPS with a reference dynamics provides by NNs trained with the value (NNv) and category (NNc) of $\ell_x$. They coincide with TPS using the exact Doob dynamics (26) as reference (purple stars) and with the exact diagonalisation result. Figure 3(b) demonstrates that TPS supplemented by appropriately trained NNs [with the architecture of figure 2(b)] can efficiently recover the dynamical LD statistics of the CMD, at least for $N = 6 \times 6$. The next section addresses how to scale up this approach.
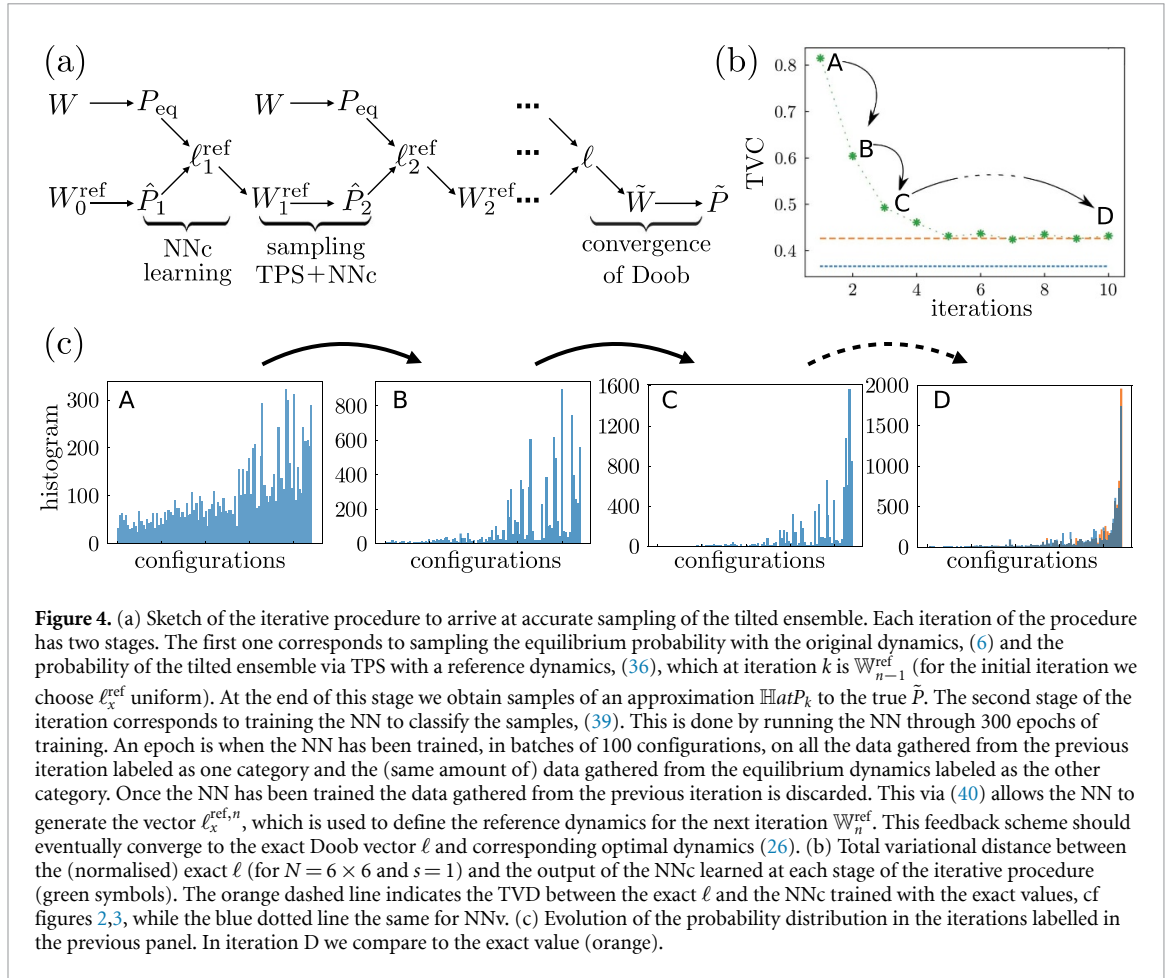
**Figure 4.** (a) Sketch of the iterative procedure to arrive at accurate sampling of the tilted ensemble. Each iteration of the procedure has two stages. The first one corresponds to sampling the equilibrium probability with the original dynamics, (6) and the probability of the tilted ensemble via TPS with a reference dynamics, (36), which at iteration $k$ is $\mathbb{W}^{\text{ref}}_{n-1}$ (for the initial iteration we choose $\ell^{\text{ref}}_x$ uniform). At the end of this stage we obtain samples of an approximation $\mathbb{H}atP_k$ to the true $\tilde{P}$. The second stage of the iteration corresponds to training the NN to classify the samples, (39). This is done by running the NN through 300 epochs of training. An epoch is when the NN has been trained, in batches of 100 configurations, on all the data gathered from the previous iteration labeled as one category and the (same amount of) data gathered from the equilibrium dynamics labeled as the other category. Once the NN has been trained the data gathered from the previous iteration is discarded. This via (40) allows the NN to generate the vector $\ell^{\text{ref},n}_x$, which is used to define the reference dynamics for the next iteration $\mathbb{W}^{\text{ref}}_n$. This feedback scheme should eventually converge to the exact Doob vector $\ell$ and corresponding optimal dynamics (26). (b) Total variational distance between the (normalised) exact $\ell$ (for $N = 6 \times 6$ and $s = 1$) and the output of the NNc learned at each stage of the iterative procedure (green symbols). The orange dashed line indicates the TVD between the exact $\ell$ and the NNc trained with the exact values, cf figures 2,3, while the blue dotted line the same for NNv. (c) Evolution of the probability distribution in the iterations labelled in the previous panel. In iteration D we compare to the exact value (orange).

As an alternative measure of efficiency we can compare the number of TPS iterations required to equilibrate (in the TPS sense) to the correct value of $k(s)$. Figure 3 shows such comparison between standard TPS and TPS supplemented by the NNc for $s = 1$. We see from the figure that for these conditions the NNc enhanced TPS converges to the exact value orders of magnitude faster than standard TPS.

## 5. Optimal dynamics via Neural networks: Semi-supervised learning and scalability

### 5.1. Iterative feedback procedure
For system sizes beyond $N = 6 \times 6$ it is not possible to numerically diagonalise the tilted generator of the CMD. In order to scale up the procedure of the last section we can adapt the optimal control procedure introduced in [31] (see also [35, 39]) to our setting, as discussed above in section 3.2. Figure 4(a) sketches our feedback scheme. The procedure consists of two stages per iteration. The first one samples both the equilibrium configurations from $P_{\text{eq}}$ with the original dynamics (6) (something which is redundant in the CMD as the equilibrium distribution is uniform) and configurations in the ensemble tilted by $s$, using TPS and a reference dynamics. This latter step is imperfect and only produces samples according to an approximation $\mathbb{H}atP$ to the true target distribution $\tilde{P}$. The second stage of each iteration is to train the NN to classify between configuration generated by $P_{\text{eq}}$ and $\tilde{P}$. The NN we use has the structure of that of figure 2(b). The trained network is then used to generate the vector $\ell^{\text{ref}}$ that determines the reference dynamics, cf (36), for the next iteration of the procedure. For enough iterations we eventually converge to the optimal dynamics (26).

We can test the effectiveness of this iterative procedure by looking at the probability distribution produced by each step in the iteration. Figure 4(b) shows the distance between the left eigenvector $\ell^{\text{ref},n}_x$ at each iteration $n$ the feedback procedure and the exact one $\ell_x$. If we normalise these vectors we obtain probability distributions over the configurations and we can compare them in terms of total variational distance (TVD)

$$\text{TVD}(\ell^{\text{ref}}, \ell) = \sum_x \left| \frac{\ell^{\text{ref}}_x}{\sum_y \ell^{\text{ref}}_y} - \frac{\ell_x}{\sum_y \ell_y} \right|. \tag{42}$$
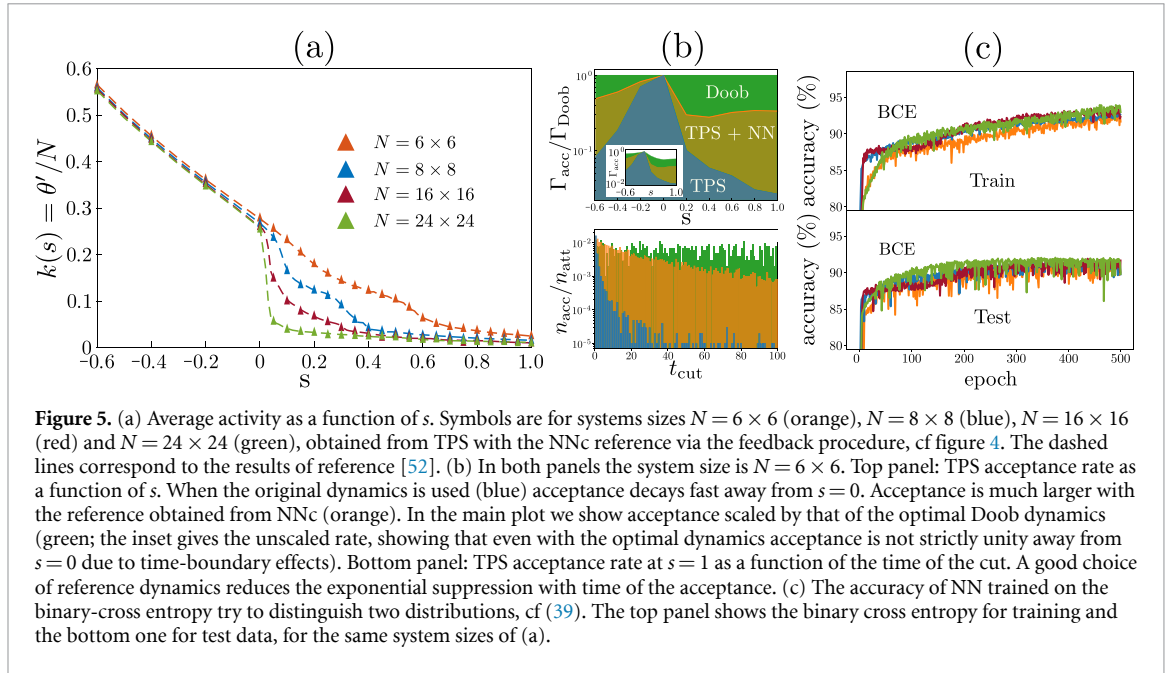
**Figure 5.** (a) Average activity as a function of *s*. Symbols are for systems sizes $N = 6 \times 6$ (orange), $N = 8 \times 8$ (blue), $N = 16 \times 16$ (red) and $N = 24 \times 24$ (green), obtained from TPS with the NNc reference via the feedback procedure, cf figure 4. The dashed lines correspond to the results of reference [52]. (b) In both panels the system size is $N = 6 \times 6$. Top panel: TPS acceptance rate as a function of *s*. When the original dynamics is used (blue) acceptance decays fast away from $s = 0$. Acceptance is much larger with the reference obtained from NNc (orange). In the main plot we show acceptance scaled by that of the optimal Doob dynamics (green; the inset gives the unscaled rate, showing that even with the optimal dynamics acceptance is not strictly unity away from $s = 0$ due to time-boundary effects). Bottom panel: TPS acceptance rate at $s = 1$ as a function of the time of the cut. A good choice of reference dynamics reduces the exponential suppression with time of the acceptance. (c) The accuracy of NN trained on the binary-cross entropy try to distinguish two distributions, cf (39). The top panel shows the binary cross entropy for training and the bottom one for test data, for the same system sizes of (a).

We see from figures 4(b,c) for the solvable $N = 6 \times 6$ case that over a small number of iterations the NNs learn the values as well as they would have done so using the exact diagonalisation data as input.

### 5.2. Sampling results

We are now in a position to use the semi-supervised learning method to sample systems that are not accessible to exact diagonalisation. Figure 5(a) shows the average activity per unit time and size, $k(s)/N$, as a function of *s* for different system sizes in the CMD. We see from the figure that the results coincide with those of reference [52], currently the state of the art for this model. The key observation of [52] was the existence of a transition at the LD level at $s = 0$ in the limit of large system size, seen as the change in $k(s)$ that is tending to a discontinuity. Accurate sampling of trajectories responsible for the transition is the main computational difficulty in this system. The results of figure 5(a) show that our NN assisted feedback procedure is capable of sampling such rare events efficiently.

One caveat of this process, arose from the choice of initial dynamics for the largest system size we explored, $N = 24 \times 24$. In the smaller system sizes we simply set $\ell_x^{\text{ref}}$ to be uniform as an initial dynamics. However for $N = 24 \times 24$, the distribution produced by setting $\ell_x^{\text{ref}}$ to be uniform was not sufficiently distinct from the equilibrium distribution for the iterative procedure to converge in a reasonable time. To solve this problem we explored the use of transfer learning by Training a NN on a tiled version of a smaller system size, $N = 6 \times 6$; such that the input to the NN is sufficiently sized for $N = 24 \times 24$ configurations. The initial dynamics can be implemented using the trained NN instead of the uniform $\ell_x^{\text{ref}}$. The initial distribution produced from the trained NN was then capable of converging in a reasonable time.

The effectiveness of the iterative NN method can be seen in the ability to find a reference dynamics under which to run TPS where the acceptance of proposed trajectory moves is larger than in standard TPS. In figure 5(b) (top panel) we show the TPS acceptance as a function of *s*. For TPS where trajectories are generated with the original dynamics, this acceptance decreases fast with *s*. Furthermore, as shown in figure 5(b) (bottom panel) for the $s = 1$ case, acceptance is exponentially suppressed with the time point at which the current trajectory is cut to generate the proposed one. In contrast, TPS with the NN reference dynamics has a much larger overall acceptance probability away from $s = 0$ and the exponential reduction with cut time is much suppressed. In this sense it is getting closer to the optimal Doob dynamics, for which acceptance probability is in principle independent of cut time [but not quite equal to unity away from $s = 0$ due to the need to correct the temporal boundaries in trajectories, see discussion following equation (28)].

A final check on how well the semi-supervised NN method is performing, instead of looking at the loss function of the NN under training/testing, we can consider the accuracy as shown in figure 5(c). This is defined as the percentage of correctly categorised configurations and is a metric that can be applied to any system size. We use the exact data from $N = 6 \times 6$ as a benchmark.

## 6. Conclusions

We have addressed the problem of numerically sampling rare dynamical trajectories for the estimation of the large deviation statistics of dynamical observables. We have considered this problem specifically for the case of the classical dimer model as a characteristic many-body system where the sampling complexity arises from its correlated dynamics and associated trajectory phase transition. We have shown that it is possible to implement a scheme where the optimal dynamics with which to sample exponentially tilted trajectory ensembles is obtained from an iterative feedback mechanism with minimal input regarding the nature of the problem. Specifically, by using a neural network as a functional approximator for the eigenvector over the many-body configurations that defines the reference dynamics, we are able to implement our scheme in a manner that is scalable with system size. For the CDM we showed that this procedure efficiently recovers the LD phase transition of the model. Since the aim of the paper was to present a Deep Learning approach to LD trajectory sampling we chose to compare to the CDM to validate our method. In future work we hope to apply it to yet unsolved problems, in the context of dimer models and interesting one would be to go beyond two dimensions. We expect the key property of our systematic approach to be beneficial especially in problems where a physically inspired approximation to the eigenvectors is not clear (which is the most widespread situation in many-body problems). Our approach should be directly applicable to the study of similar types of time extensive observables, like the activity, on other many-body systems of interest that also obey detailed balance. An interesting extension would be to systems without detailed balance and/or non-symmetric observables such as currents. More generally, our results here are yet another example of the broad potential of deep learning methods for study of non-equilibrium systems more generally.

## Acknowledgments

## Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## ORCID iD

Tom H E Oakes ⓘ https://orcid.org/0000-0002-3319-4433

## References

[1] Bolhuis P G, Chandler D, Dellago C and Geissler P L 2002 *Annu. Rev. Phys. Chem.* **53** 291
[2] Merolle M, Garrahan J P and Chandler D 2005 *Proc. Natl. Acad. Sci. USA* **102** 10837
[3] Lecomte V, Appert-Rolland C and van Wijland F 2007 *J. Stat. Phys* **127** 51–106
[4] Garrahan J P, Jack R L, Lecomte V, Pitard E, van Duijvendijk K and van Wijland F 2007 *Phys. Rev. Lett.* **98** 195702
[5] Maes C and Netocny K 2008 *Europhys. Lett* **82** 30003
[6] Garrahan J P, Jack R L, Lecomte V, Pitard E, van Duijvendijk K and van Wijland F 2009 *J. Phys. A: Math.General* **42** 075007
[7] Hedges L O, Jack R L, Garrahan J P and Chandler D 2009 *Science* **323** 1309
[8] Baiesi M, Maes C and Wynants B 2009 *Phys. Rev. Lett.* **103** 010602
[9] Jack R L and Sollich P 2010 *Prog. Theor. Phys. Supp.* **184** 304
[10] Pitard E, Lecomte V and van Wijland F 2011 *Europhys. Lett* **96** 56002
[11] Bodineau T and Toninelli C 2012 *Commun. Math. Phys.* **311** 357
[12] Speck T, Malins A and Royall C P 2012 *Phys. Rev. Lett.* **109** 195703
[13] Chetrite R and Touchette H 2013 *Phys. Rev. Lett.* **111** 120601
[14] Espigares C P, Garrido P L and Hurtado P I 2013 *Phys. Rev.* E **87** 032115
[15] Budini A A, Turner R M and Garrahan J P 2014 *J. Stat. Mech.* **2014** P03012
[16] Chetrite R and Touchette H 2015 *Ann. Henri Poincaré* **16** 2005
[17] Jack R L, Thompson I R and Sollich P 2015 *Phys. Rev. Lett.* **114** 060601
[18] Bertini L, Faggionato A and Gabrielli D 2015 *Stochastic Process. Appl* **125** 2786
[19] Eckmann J P and Ruelle D 1985 *Rev. Mod. Phys.* **57** 617
[20] Ruelle D 2004 *Thermodynamic Formalism* (Cambridge: Cambridge University Press)
[21] Gaspard P 2005 *Chaos, Scattering and Statistical Mechanics* (Cambridge: Cambridge University Press)
[22] Touchette H 2009 *Phys. Rep.* **478** 1
[23] Garrahan J P 2018 *Physica* **504** 130

[24] Jack R L 2019 *Euro. Phys. J.* B **93** 1–22
[25] Chandler D 1987 *Introduction to Modern Statistical Mechanics* (Oxford: Oxford University Press)
[26] Giardina C, Kurchan J and Peliti L 2006 *Phys. Rev. Lett.* **96** 120603
[27] Cérou F and Guyader A 2007 *Stoch. Anal. Appl.* **25** 417
[28] Lecomte V and Tailleur J 2007 *J. Stat. Mech.* **2007** P03004
[29] Gorissen M, Hooyberghs J and Vanderzande C 2009 *Phys. Rev.* E **79** 020101
[30] Giardina C, Kurchan J, Lecomte V and Tailleur J 2011 *J. Stat. Phys* **145** 787
[31] Nemoto T, Bouchet F, Jack R L and Lecomte V 2016 *Phys. Rev.* E **93** 062123
[32] Ray U, Chan G K-L and Limmer D T 2018a *Phys. Rev. Lett.* **120** 210602
[33] Ray U, Chan G K-L and Limmer D T 2018b *J. Chem. Phys* **148** 124120
[34] Klymko K, Geissler P L, Garrahan J P and Whitelam S 2018 *Phys. Rev.* E **97** 032123
[35] Ferré G and Touchette H 2018 *J. Stat. Phys* **172** 1525
[36] Bañuls M C and Garrahan J P 2019 *Phys. Rev. Lett.* **123** 200601
[37] Helms P, Ray U and Chan G K-L 2019 *Phys. Rev.* E **100** 022101
[38] Jacobson D and Whitelam S 2019 *Phys. Rev.* E **100** 052139
[39] Das A and Limmer D T 2019 *J. Chem. Phys.* **151** 244123
[40] Carollo F, Garrahan J P, Lesanovsky I and Pérez-Espigares C 2018 *Phys. Rev.* A **98** 010103
[41] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)
[42] Mehta P, Bukov M, Wang C-H, Day A G, Richardson C, Fisher C K and Schwab D J 2019 *Phys. Rep.* **810** 1–124
[43] Kashiwa K, Kikuchi Y and Tomiya A 2019 *Progress Theor. Exp. Phys.* **2019** 083A04
[44] Tanaka A and Tomiya A 2017 *J. Phys. Soc. Japan* **86** 063001
[45] Efthymiou S, Beach M J and Melko R G 2019 *Phys. Rev.* B **99** 075113
[46] Huang L and Wang L 2017 *Phys. Rev.* B **95** 035105
[47] Shen H, Liu J and Fu L 2018 *Phys. Rev.* B **97** 205140
[48] Nagai Y, Shen H, Qi Y, Liu J and Fu L 2017 *Phys. Rev.* B **96** 161102
[49] Huang L, Yang Y-f and Wang L 2017 *Phys. Rev.* E **95** 031301
[50] Fisher M E 1961 *Phys. Rev.* **124** 1664
[51] Henley C L 2010 *Annu. Rev. Condens. Matter Phys.* **1** 179
[52] Oakes T, Powell S, Castelnovo C, Lamacraft A and Garrahan J P 2018 *Phys. Rev.* B **98** 064302
[53] Maes C 2020 *Phys. Rep.* **850** 1–33
[54] Chollet F *et al* 2015 Keras https://keras.io
[55] Friedman J, Hastie T and Tibshirani R 2001 *The Elements of Statistical Learning* (New York: Springer series in statistics) vol 1