Title
Probing reaction channels via reinforcement learning

Journal
Machine Learning: Science and Technology, 4(4)

ISSN
2632-2153

Authors
Liang, Senwei
Singh, Aditya N
Zhu, Yuanran
et al.

Publication Date
2023-12-01

DOI
10.1088/2632-2153/acfc33

Probing reaction channels via reinforcement learning

IOP Publishing Mach. Learn.: Sci. Technol. 4 (2023) 045003 https://doi.org/10.1088/2632-2153/acfc33

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
31 May 2023

REVISED
5 September 2023

ACCEPTED FOR PUBLICATION
21 September 2023

PUBLISHED
6 October 2023

PAPER

Probing reaction channels via reinforcement learning

Senwei Liang[1], Aditya N Singh[2,3], Yuanran Zhu[1], David T Limmer[2,3,4,5] and Chao Yang[1,*]

[1] Applied Mathematics and Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States of America
[2] Department of Chemistry, University of California Berkeley, Berkeley, CA 94720, United States of America
[3] Chemical Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States of America
[4] Materials Science Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States of America
[5] Kavli Energy Nanoscience Institute at Berkeley, Berkeley, CA 94720, United States of America
* Author to whom any correspondence should be addressed.

E-mail: cyang@lbl.gov

Keywords: chemical reaction, transition path, transition state, reactive channel, committor function, neural network, reinforcement learning

Abstract

Chemical reactions are dynamical processes involving the correlated reorganization of atomic configurations, driving the conversion of an initial reactant into a result product. By virtue of the metastability of both the reactants and products, chemical reactions are rare events, proceeding fleetingly. Reaction pathways can be modelled probabilistically by using the notion of reactive density in the phase space of the molecular system. Such density is related to a function known as the committor function, which describes the likelihood of a configuration evolving to one of the nearby metastable regions. In theory, the committor function can be obtained by solving the backward Kolmogorov equation (BKE), which is a partial differential equation (PDE) defined in the full dimensional phase space. However, using traditional methods to solve this problem is not practical for high dimensional systems. In this work, we propose a reinforcement learning based method to identify important configurations that connect reactant and product states along chemical reaction paths. By shooting multiple trajectories from these configurations, we can generate an ensemble of states that concentrate on the transition path ensemble. This configuration ensemble can be effectively employed in a neural network-based PDE solver to obtain an approximation solution of a restricted BKE, even when the dimension of the problem is very high. The resulting solution provides an approximation for the committor function that encodes mechanistic information for the reaction, paving a new way for understanding of complex chemical reactions and evaluation of reaction rates.

1. Introduction

The study of rare reactive events is a fundamental topic within the field of chemical physics [1]. These reactions are dynamical processes that can be characterized by the transition of a molecular system from one collection of meta-stable atomic configurations, i.e. a reactant, to another, i.e. a product. This metastability arises from the features of the potential energy surface, where metastable states are configurations near the local minimizer separated by high energy saddle points. When the saddle points lie along the typical transition paths between reactants and products, they are referred to as transition states. The direct numerical studies through molecular dynamics simulations are typically prohibitive computationally because these reactions are rare relative to the timescales of the typical thermal fluctuations of the system [2].

A primary quantity of interest that has been employed to infer a mechanistic understanding of reactive events is the committor, a function that maps the phase space of the system to the probability of reacting [3, 4]. This function encodes the ideal reaction coordinate, an abstract one dimension coordinate that can be thought of as a nonlinear function of the full state of the system along which chemical reaction can be well characterized, and the computation of this function can be framed as a multidimensional optimization

problem [5–7]. The high-dimensionality of this problem poses difficulties in its inference, however a multitude of methods have been developed over the past decade that have made the computation of this function tractable. Some notable examples include the string method [5, 6], diffusion maps (DMs) [8–11] and neural networks (NNs) [12–21].

This optimization of the committor can be framed as two separate but intertwined problems—(1) finding configurations with high reactive densities and (2) fitting a nonlinear ansatz to solve the optimization problem on those configurations to compute the committor. To solve the first problem, one needs a measure of the reactive density that depends on the committor itself, and to solve the second problem, one needs to access those configurations. One way to solve the first problem is to obtain an ensemble of reactive trajectories via Transition Path Sampling [3, 22–24], a Monte-Carlo based method that enables generation of new reactive trajectories from old ones. While this method provides access to the ideal set of configurations to compute the committor on, it often suffers from low acceptance rates resulting in long decorrelation times between subsequent trajectories, and further refinements based on approximations of the committor are required to enable efficient sampling. Beyond transition path sampling, a wide range of methods have been developed in the past two decades to access rare but important configurations that encode information of reactive events. Such examples include but are not limited to metadynamics [25, 26], weighted ensemble method [27, 28] and variational enhanced sampling [29]. These methods rely on applying a carefully chosen bias potential to the original Hamiltonian along some low-rank ansatz of the reaction coordinate, often referred to as an order parameter to generate more atomic configurations near the transition state. While these methods can obtain accurate estimates of thermodynamic and kinetic observables, they strongly hinge on the overlap between the order parameter and the reaction coordinate. As such, the ensembles of configurations are often limited by the choice of the order parameter.

In this paper, we use a reinforcement learning (RL) based method to obtain an ensemble of configurations with high reactive densities. This is done through a two-step method in which we first find saddle points along the reactive probability density, which we refer to as *connective configurations*. Connective configurations are a subset of the transition state ensemble defined as the collection of configurations where the probability of reacting is one half. Connective configurations additionally are the maxima of reactive probability density, they are the most likely set of transition states to be encountered during a reactive event. While transition states can be thought of as a surface in the full configurational phase space from which a configuration has equal likelihood to react or not, connective configurations are a set of points along this surface where the reaction is most likely to cross the iso-committor surface. The optimization to find these saddle points proceeds through an RL algorithm where an agent moves from one state to another by taking an action in order to achieve a certain goal. Each action is associated with a reward, and which action to take depends on a policy function designed to guide the agent toward the goal. In the context of searching for connective configurations, a state is an atomic configuration, an action simply moves the agent from one configuration to another according to a policy that is updated (or learned) over time. The optimal policy, which is obtained after performing several RL episodes with each episode consisting of a sequence of actions, would allow us (the agent) to move from any arbitrary configuration toward a connective configuration. In our RL algorithm, the reward function, which we will describe in detail in section 3, is chosen as a proxy to the true objective function to be maximized.

While RL based methods have been previously used to identify transition states [30] and perform transition path sampling [31–33] or cloning [34, 35], the proposed method has a different objective and yields different quantities of interest. Once we have obtained these configurations, we perform shooting operations from these points to obtain a set of configurations along a relatively small subspace in the configuration space, which we refer to as *reaction channels*. These channels can be understood as a set of configurations distributed according to the reactive probability density. Obtaining these configurations solves the first part of our problem without utilizing importance sampling methods [22, 36]. We also note that the notion of reaction channels is similar to the concept of transition tubes introduced in within the Transition Path Theory framework [7]. The distinction lies in the fact that transition tubes are assumed to be localized within a narrow domain of the system. This is not a requirement in our case as the reactive channels contain configurations from multiple reactive tubes when degenerate reactive pathways exist.

Once we obtain configurations within each reaction channels, we train a feed-forward neural network (FNN) to solve the backward-Kolmogorov equation (BKE). While a range of methods have been proposed to approximate a committor using an FNN, the method demonstrated in this work is unique in that it solves the exact BKE rather than the variational form [13–16] or the Feynman–Kac form [17, 18]. This form of optimization is more accurate, however similar to other methods, it is strongly sensitive to the configurations that it is trained on. We find that training on samples obtained from RL based method proposed in this work outperforms optimization on samples generated from a grid-based or high-temperature based sampling method by at least one order of magnitude. Beyond the accuracy of the committor, the fidelity of this method

is also demonstrated through accurate estimates of reactive rates, that are often non-trivial to converge. Thus, we are able to solve the problem of obtaining samples with high reactive densities using RL (figure 1(a)), and the problem of computing the committor by parameterizing an FNN as the solution to the exact BKE (figure 1(b)).

The rest of the paper is organized as follows. In section 2, we introduce terminologies and concepts that are relevant to the approach we take to study chemical reactions. We outline the basic scheme we use with some detail in section 3. Section 3.1 is devoted to the details of the RL algorithm we use to identify connective configurations. In particular, we discuss how to define the reward function and design an effective policy. We discuss how to use NN to obtain the values of the committor function at selected configurations generated within reaction channels in section 3.2. We demonstrate the effectiveness of our approach using a few examples in section 4. We conclude the paper with some additional perspectives in section 5. Some of the computational details and discussions are provided in the appendix.

2. Preliminaries

In this section, we introduce terminologies and concepts needed to present our algorithms in subsequent sections. Specifically, we define a reactive trajectory associated with an overdamped Langevin dynamics, the committor function associated with transition paths, and describe how the reaction rate constant can be calculated from the committor function in section 2.1. The formalism of transition path theory discussed in section 2.1 that we will use in this paper is well reviewed in the previous literature [7, 37]. We show how the committor function can be computed by using a FNN in section 2.2.

2.1. Transition path theory

We denote $\Omega \subset \mathbb{R}^d$ as the configuration space of a system, which represents the coordinates of the system. For example, if a molecule of interest has L atoms, then the dimension of the configuration space is $d = 3L$. This is because each atom has 3 spatial coordinates. Let $\boldsymbol{x}(t) = (x_1(t), x_2(t), \ldots, x_d(t)) \in \Omega$ be the evolution of configuration that satisfy an overdamped Langevin dynamics defined by

$$\gamma_i \dot{x}_i(t) = -\frac{\partial V(\boldsymbol{x}(t))}{\partial x_i} + \xi_i(t), \quad i = 1, \ldots, d, \tag{1}$$

where $V : \Omega \to \mathbb{R}$ is a potential energy function, γ_i is the friction coefficient for x_i, $\xi_i(t)$ is white noise with mean $\langle \xi_i(t) \rangle = 0$ and variance $\langle \xi_i(t)\xi_j(t') \rangle = 2\beta^{-1}\gamma_i \delta(t - t')\delta_{ij}$ and β is the inverse of the product of temperature and Boltzmann's constant.

Let A and B be two disjoint metastable regions of interest in the configuration space $\Omega \subset \mathbb{R}^d$. They correspond to regions surrounding two distinct local minima of the potential energy $V(\boldsymbol{x})$. We are interested in trajectories that start in A and terminate in B. Along such a trajectory, $\boldsymbol{x}(t)$ must escape out of one metastable region and cross over a transition region $\Omega \setminus (A \cup B)$ before reaching another metastable region. Such a trajectory is often referred to as a *transition path*. For chemical systems, a transition region is associated with a chemical reaction. A transition path is also referred to as a *reactive trajectory*.

The probability density of $\boldsymbol{x}$ follows the Boltzmann–Gibbs distribution $p(\boldsymbol{x}) = \exp(-\beta V(\boldsymbol{x}))/Z$, where $Z = \int_\Omega \exp(-\beta V(\boldsymbol{x}))\mathrm{d}\boldsymbol{x}$ is a normalization constant or the partition function. The probability of observing $\boldsymbol{x}$ in a transition region relative to the probability of observing $\boldsymbol{x}$ in a metastable region is very low. As a result, a transition path or a reactive trajectory is a rare event and generally requires a very long period of simulation time to observe.

Committor function. Reactive trajectories are not unique. In fact, a chemical reaction is often characterized by an ensemble of reactive trajectories. Along each reactive trajectory, we are particularly interested in configurations that are equally likely to evolve (or commit) to either one of the metastable regions A and B. The probability that a configuration will initially transition into one metastable region rather than the other can be characterized by what is known as a *committor function*. To give a precise definition of a committor function $q(\boldsymbol{x})$, let $\tau_D(\boldsymbol{x})$ be the first hitting time of region D when the dynamics is initiated from $\boldsymbol{x}$, i.e. $\tau_D(\boldsymbol{x})$ represents the time it takes for the chemical system to first enter region D when the dynamics starts from $\boldsymbol{x}$. A committor function, denoted by $q(\boldsymbol{x})$, is defined as the probability that a trajectory $\bar{\boldsymbol{x}}(t)$, starting from a point $\bar{\boldsymbol{x}}(0) = \boldsymbol{x}$ within a given set Ω, reaches B before it reaches A, namely, $q(\boldsymbol{x}) = \mathrm{Prob}\,(\tau_B < \tau_A \mid \bar{\boldsymbol{x}}(0) = \boldsymbol{x})$. It is well known that $q(\boldsymbol{x})$ satisfies the BKE

$$\begin{aligned} \mathcal{L}(q) := \sum_{i=1}^{d} \left(-\gamma_i^{-1} \frac{\partial V(\boldsymbol{x})}{\partial x_i} \frac{\partial q(\boldsymbol{x})}{\partial x_i} + \gamma_i^{-1} \beta^{-1} \frac{\partial^2 q(\boldsymbol{x})}{\partial x_i^2} \right) = 0, \\ \boldsymbol{x} \in \Omega \setminus (A \cup B), \quad q(\boldsymbol{x}) = 0,\ \boldsymbol{x} \in A, \quad q(\boldsymbol{x}) = 1,\ \boldsymbol{x} \in B. \end{aligned} \tag{2}$$

When dimension of Ω is low (2 or 3), this partial differential equation (PDE) can be solved numerically by standard methods such as the finite difference or finite element method. However, when the dimension of Ω is high, it is not practical to use these methods to solve for $q(\boldsymbol{x})$. We will discuss an alternative approach that uses a FNN to solve for $q(\boldsymbol{x})$ in section 2.2.

Probability of being reactive. We use $\rho(\boldsymbol{x})$ to denote the probability of observing a reactive trajectory crossing $\boldsymbol{x}$. Observing a reactive trajectory crossing $\boldsymbol{x}$ involves two events. First, there's the event of observing $\boldsymbol{x}$, which has a probability $p(\boldsymbol{x})$ following the Boltzmann–Gibbs distribution. Second, the trajectory crossing $\boldsymbol{x}$ is reactive. This reactive trajectory can be split into two sub-paths: one starts at $\boldsymbol{x}$ and first reaches point B rather than A, with probability $q(\boldsymbol{x})$; the other starts within A rather than B and reaches $\boldsymbol{x}$, which has probability $(1-q(\boldsymbol{x}))$ under the assumption of time reversibility at thermal equilibrium [37]. Therefore, $\rho(\boldsymbol{x})$ can be expressed by

$$\rho(\boldsymbol{x}) = (1-q(\boldsymbol{x}))\, q(\boldsymbol{x})\, p(\boldsymbol{x}). \tag{3}$$

A configuration $\boldsymbol{x}$ that has a high reactive density $\rho(\boldsymbol{x})$ is of particular interest because it marks a transition region along a reactive trajectory associated with an overdamped Langevin dynamics. The function $\rho(\mathbf{x}) = (1-q(\mathbf{x}))q(\mathbf{x})p(\mathbf{x})$ has two components: $(1-q(\mathbf{x}))q(\mathbf{x})$ and $p(\mathbf{x})$. The first component reaches its maximum when $\{\boldsymbol{x} : q(\mathbf{x}) = 0.5\}$, representing configurations on the half-isocommittor surface $\{\mathbf{x} : q(\mathbf{x}) = 0.5\}$. It decreases as the configuration approaches either metastable region. On the other hand, $p(\mathbf{x})$ takes on larger values as $\mathbf{x}$ gets closer to the metastable states. Therefore, the maximum of $\rho(\mathbf{x})$ arises from a delicate balance between these two competing factors. This maximum point may reside on the configurations of the lowest energy along the half-isocommittor surface. To evaluate $\rho(\boldsymbol{x})$, the definition of $q(\boldsymbol{x})$ given in (3) requires the committor function $q(\boldsymbol{x})$ to be known in advance. Because $q(\boldsymbol{x})$ is generally difficult to calculate for an arbitrary $\boldsymbol{x}$, it is not easy to calculate $\rho(\boldsymbol{x})$ in practice.

Reaction rate. The committor function is used in [7] to introduce the notion of the *current* or *flux* associated with a reactive trajectory. At a configuration $\boldsymbol{x}$ along a reactive trajectory, the flux $\boldsymbol{J}(\boldsymbol{x})$ across $\boldsymbol{x}$ is defined as

$$J_i(\boldsymbol{x}) = Z^{-1} e^{-\beta V(\boldsymbol{x})} \gamma_i^{-1} \beta^{-1} \frac{\partial q(\boldsymbol{x})}{\partial x_i}, \tag{4}$$

where $\boldsymbol{J}(\boldsymbol{x}) = (J_1(\boldsymbol{x}), \ldots, J_d(\boldsymbol{x}))$ and Z is again the partition function.

If $\boldsymbol{J}(\boldsymbol{x})$ is known for all $\boldsymbol{x}$ along a dividing surface S, a surface that separates A and B, we can use it to evaluate the reaction rate constant κ via

$$\kappa = \int_S n_S(\boldsymbol{x})\, \boldsymbol{J}(\boldsymbol{x})\, \mathrm{d}\sigma_S(\boldsymbol{x}), \tag{5}$$

where $n_S(\boldsymbol{x})$ is normal vector of S.

Note that the main contribution to the integral (5) comes from configurations $\boldsymbol{x}$ along the dividing surface that has a relatively large magnitude of the flux $\boldsymbol{J}(\boldsymbol{x})$. Because these configurations typically occupy a small area S' on S for rare events [7, 37], we can focus on this area and approximate (5) by

$$\kappa \approx \int_{S'} n_{S'}(\boldsymbol{x})\, \boldsymbol{J}(\boldsymbol{x})\, \mathrm{d}\sigma_S(\boldsymbol{x}). \tag{6}$$

The area S' can be defined by the intersection of S and the so-called reaction channel to be defined in section 3.

We should note that as long as S' can be easily identified and $\boldsymbol{J}(\boldsymbol{x})$ can be efficiently evaluated for all $\boldsymbol{x} \in S'$, the formula (6) provides a more practical way to compute the rate constant compared to a brute force approach in which the rate constant is computed according to the alternative definition

$$\kappa = \lim_{T\to\infty} \frac{N_T}{T}, \tag{7}$$

where N_T is the number of trajectory segments that leave A and enter B within the time interval $[0, T]$ [7]. In the latter approach, κ can be approximated by a *direct simulation*, i.e. we can generate a sufficiently long overdamped Langevin trajectory from a random starting point and count N_T. However, when the system contains a high barrier, it becomes extremely rare to observe a reactive trajectory even with a long time simulation.

If the dividing surface is not accessible, an alternative method is to calculate the integral of the flux on the transition region, which can be done as follows:

$$\kappa = \int_{\Omega\setminus(A\cup B)} Z^{-1} e^{-\beta V(\boldsymbol{x})} \beta^{-1} \sum_{i=1}^{d} \gamma_i^{-1} \left(\frac{\partial q(\boldsymbol{x})}{\partial x_i}\right)^2 \mathrm{d}\boldsymbol{x}, \tag{8}$$

for the systems considered here.

2.2. Solving committor function via deep learning

In recent year, deep learning has demonstrated remarkable progress in various domains of scientific exploration [38–40], thanks to the exceptional approximation and generalization capabilities of NNs [41]. In particular, deep learning has emerged as a powerful tool for solving a wide range of PDEs, including those formulated in high-dimensional spaces where conventional solvers like finite difference and finite element methods suffer from the curse of dimensionality [42]. Furthermore, NN-based solvers can be easily adapted to solve PDEs defined on an irregular domain. These two key attributes of an NN-based PDE solver make it highly advantageous for the specific problem we aim to address in this study. As we will show in section 4, an NN-based PDE solver enables us to solve a 66-dimensional BKE associated with a 22-atom molecule within a reaction channel that consists of an ensemble of configurations not uniformly distributed or organized in a regular domain in the configuration space. The NN-based PDE solver overcomes these complexities and enables us to effectively handle this challenging scenario.

To apply an NN-based PDE solver to BKE (2) on Ω, the approximation to the solution (i.e. the committor function) is represented as an NN $q(\boldsymbol{x};\boldsymbol{\theta})$, where a vector $\boldsymbol{\theta}$ denotes a set of weights and biases. The network takes $\boldsymbol{x}$ as the input and generates $q(\boldsymbol{x};\boldsymbol{\theta})$ as the output. The NN parameters are determined in an iterative training procedure that minimizes a loss function with respect to $\boldsymbol{\theta}$ for different choices of the input $\boldsymbol{x}$. To learn $\boldsymbol{\theta}$, we define the loss function as

$$\|\mathcal{L}(q(\boldsymbol{x};\boldsymbol{\theta}))\|^2_{L^2(\Omega)} + \ell\|q(\boldsymbol{x};\boldsymbol{\theta})\|^2_{L^2(A)} + \ell\|q(\boldsymbol{x};\boldsymbol{\theta}) - 1\|^2_{L^2(B)}, \tag{9}$$

where ℓ is a penalty coefficient used to impose the boundary constraints. In practice, the L^2-norm in (9) is evaluated by summing the loss on the data points sampled randomly and uniformly in Ω and $\mathcal{L}$ is computed by auto-differentiation using advanced deep learning frameworks (e.g. Pytorch [43]). The NN-based optimization (9) can be conducted by stochastic gradient descent (SGD), such as Adam [44], a variant of SGD based on momentum. Similarly, when solving BKE (2) on K sub-domains $\Omega_1,\ldots,\Omega_K \subset \Omega$, the PDE becomes $\mathcal{L}(q) = 0$ for $\boldsymbol{x} \in \cup_{s=1}^{K}\Omega_s$ along with the boundary conditions, which is referred to as *restricted BKE*. We can define the NN-optimization problem by

$$\min_{\boldsymbol{\theta}} \sum_{s=1}^{K} \|\mathcal{L}(q(\boldsymbol{x};\boldsymbol{\theta}))\|^2_{L^2(\Omega_s)} + \ell\|q(\boldsymbol{x};\boldsymbol{\theta})\|^2_{L^2(A)} + \ell\|q(\boldsymbol{x};\boldsymbol{\theta}) - 1\|^2_{L^2(B)}. \tag{10}$$

When we have data points $\{\boldsymbol{x}^{i,s}\}_{i=1}^{N_s} \subset \Omega_s$, $s = 1,\ldots,K$, $\{\hat{\boldsymbol{x}}^i\}_{i=1}^{N_\alpha} \subset A$ and $\{\tilde{\boldsymbol{x}}^i\}_{i=1}^{N_\beta} \subset B$, the loss function in (10) can be evaluated as

$$\frac{1}{\sum_{s=1}^{K} N_s} \sum_{s=1}^{K}\sum_{i=1}^{N_s} \left(\mathcal{L}q\left(\boldsymbol{x}^{i,s};\boldsymbol{\theta}\right)\right)^2 + \frac{\ell}{N_\alpha}\sum_{i=1}^{N_\alpha} q\left(\hat{\boldsymbol{x}}^i;\boldsymbol{\theta}\right)^2 + \frac{\ell}{N_\beta}\sum_{i=1}^{N_\beta}\left(q\left(\tilde{\boldsymbol{x}}^i;\boldsymbol{\theta}\right) - 1\right)^2. \tag{11}$$

Here, N_s stands for the number of configurations within the sth subdomain Ω_s for $s = 1,\ldots,K$, while N_α and N_β represent the number of configurations within a small neighborhood of the metastable states A and B respectively.

3. Methodology

As we indicated in section 2.1, configurations $\boldsymbol{x}$ with high reactive density $\rho(\boldsymbol{x})$ are of interest because they mark a transition region in which reactive trajectories are more likely to be observed. Intuitively, if we shoot a trajectory from a configuration $\boldsymbol{x}$ with a high reactive density $\rho(\boldsymbol{x})$ and initiate it with a random momentum, it is likely that the trajectory will stay within a region where reactive trajectories pass through. Even though such a trajectory may not be part of a reactive trajectory, the configurations along such a trajectory occupy a small subspace that is likely to contain several reactive trajectories. We will refer to the subspace formed by these configurations as a *reaction channel*. Because configurations within such a subspace are likely to have high reactive flux $\boldsymbol{J}(\boldsymbol{x})$, we will focus on these configurations, and solve a restricted BKE within a reaction

channel using the NN technique discussed in section 2.2 to obtain an approximate committor function and its gradient at configurations within the channel. With these, one can approximately calculate the reaction rate constant by evaluating (6).

We should note that the concept of reaction channel introduced here is similar in spirit to the notion of *transition tube* introduced in [7]. A transition tube is defined to be an ensemble of regions on non-intersection dividing surfaces between two metastable regions A and B that have localized flux. Because a transition tube is characterized by configurations with relatively high reactive flux which depends on the unknown committor function, it is not easy to identify directly. Although the central curve within the transition tube can be approximated by the minimum energy path which can be computed by the string method [5], defining the region of the tube is still not trivial. The transition tube and reaction channel share conceptual similarities, as they both aim to characterize the average behavior of reactive trajectories between metastable states A and B. However, there are some key differences: The transition tube is a concept within Transition Path Theory that requires the assumption that the reactive pathway is dominated by a single localized tube (a chain of configurations that connect the two metastable states) that spans a narrow subset of the complete state space of the system. However, this assumption can break down when there are degenerate pathways that the system evolves through to go from one metastable well to the other. As an example, in section 4.1, the potential energy surface contains two pathways or tubes through which the reaction can occur through. On the other hand, we use the concept of reaction channels to define samples of configurations that are distributed according to the reactive probability density. Hence, the two concepts should be the same for reactions where there is a single dominant reactive pathway.

Because a reaction channel is generated by shooting trajectories from a single configuration, it is relatively easy to produce as long as we can select a proper configuration to shoot from. Ideally, that configuration should be the one that has a high reactive density $\rho(\boldsymbol{x})$. However, because $\rho(\boldsymbol{x})$ is defined in terms of the committor function, it is not easy to identify configurations with high $\rho(\boldsymbol{x})$ directly because that would require solving the original BKE (2). In the following, we will present a RL-based technique to identify configurations that are likely to have a high $\rho(\boldsymbol{x})$, and we will refer to these configurations as *connective configurations*.

To create reaction channels, we start by performing a shooting procedure from connective configurations. Within each reaction channel, we can determine the committor function on each configuration by solving a restricted BKE using a NN. Using the NN solution and its gradient, we can then calculate the reactive flux for every configuration within the reaction channels. As we will see in the next section, the reaction channel generated by shooting trajectories from connective configurations is likely to contain configurations with relatively high reactive flux. This is sufficient to provide a good estimation of statistics, such as rate, even if not all configurations within the channel have high reactive flux.

3.1. Seeking connective configurations via RL

In this section, we show how to use a RL method to identify connective configurations. Our basic strategy is to treat each configuration as a state $\boldsymbol{x}^t$ and train an agent to take a sequence of actions $\{\boldsymbol{a}^0, \boldsymbol{a}^1, \ldots, \boldsymbol{a}^n\}$ to move from an arbitrary state $\boldsymbol{x}^0$ to $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots$ successively through the operation $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t + \boldsymbol{a}^t$, for $t = 0, 1, \ldots, n-1$, until it ultimately reaches a desired state $\boldsymbol{x}^n$ which corresponds to a connective configuration. The sequence of state-action pairs $\{(\boldsymbol{x}^t, \boldsymbol{a}^t)\}$, with $t = 0, 1, 2, .., n-1$ is an instance of a policy π the agent follows, which is initially not optimal. However, over a multi-episode learning process, the policy is gradually improved based on the feedback the agent receives from the environment, which consists of configurations not being visited, through a policy gradient. An optimal policy allows the agent to move from an arbitrary state to the desired state efficiently.

In an RL algorithm, an action that an agent takes at a particular state $\boldsymbol{x}$ is associated with a reward $r(\boldsymbol{x}, \boldsymbol{a})$ that measures the effectiveness of that state action pair. The policy an agent follows at a particular state $\boldsymbol{x}$ is often designed to maximize not just the reward $r(\boldsymbol{x}, \boldsymbol{a})$, but also the expectation of a sequence of discounted future rewards, i.e.

$$\mathbb{E}_{\tau\sim\pi}\left[R(\tau)\,|\boldsymbol{x}^0 = \boldsymbol{x}, \boldsymbol{a}^0 = \boldsymbol{a}\right], \tag{12}$$

where τ is an instance of a policy π that is specified by a sequence of state action pairs $\tau := \{(\boldsymbol{x}^0, \boldsymbol{a}^0), \ldots, (\boldsymbol{x}^t, \boldsymbol{a}^t), \ldots\}$, and

$$R(\tau) := \sum_{t=1}^{\infty} \eta^t r(\boldsymbol{x}^t, \boldsymbol{a}^t),$$

for some discount factor $0 < \eta \leqslant 1$. Such expectation of discounted future rewards is often referred to as a Q-value function or Q-function in short and denoted by $Q^\pi(\boldsymbol{x}, \boldsymbol{a})$.

We use $A(\boldsymbol{x})$ to denote the action that maximizes $Q^{\pi}(\boldsymbol{x},\boldsymbol{a})$ for a given policy π, i.e.

$$A(\boldsymbol{x}) = \arg\max_{\boldsymbol{a}} Q^{\pi}(\boldsymbol{x},\boldsymbol{a}). \tag{13}$$

It is well known that the optimal Q-function $Q^{*}(\boldsymbol{x},\boldsymbol{a}) := \max_{\pi} Q^{\pi}(\boldsymbol{x},\boldsymbol{a})$ satisfies the Bellman equation [45]

$$Q^{*}(\boldsymbol{x},\boldsymbol{a}) = \mathbb{E}_{\boldsymbol{x}'}\left[r(\boldsymbol{x},\boldsymbol{a}) + \eta \max_{\boldsymbol{a}'} Q^{*}(\boldsymbol{x}',\boldsymbol{a}')\right], \tag{14}$$

where the expectation is taken with respect to the conditional probability of the new state $\boldsymbol{x}'$ given $\boldsymbol{x}$ and $\boldsymbol{a}$. Associated with this optimal Q-function is the optimal action

$$A^{*}(\boldsymbol{x}) = \arg\max_{\boldsymbol{a}} Q^{*}(\boldsymbol{x},\boldsymbol{a}). \tag{15}$$

Because the state space that contains $\boldsymbol{x}$ and the action space that contains $\boldsymbol{a}$ in our problem are not discrete and because the analytical form of the optimal Q-function is generally unknown, it is difficult to solve the Bellman equation (14) or the optimization problem (15) directly. Finding the optimal Q-function is often an intractable problem, and approximate solutions are commonly used instead. Several methods such as the deep deterministic policy gradient [46] (DDPG) method, twin delayed DDPG [47] (TD3) method have been developed to solve the problem approximately. In these methods, $Q(\boldsymbol{x},\boldsymbol{a})$ and $A(\boldsymbol{x})$ are represented by NNs with parameter sets Ψ and Φ respectively. Here, Ψ and Φ are used to represent trainable parameters including weights and biases within the respective NNs. These NNs are trained by a set of data $\mathbb{P}$ that contains a collection of $(\boldsymbol{x},\boldsymbol{a},r(\boldsymbol{x},\boldsymbol{a}),\boldsymbol{x}')$, where $\boldsymbol{x}'$ denotes a new state reached by the agent after taking the action $\boldsymbol{a}$. Roughly speaking, the parameters Ψ and Φ are optimized in an alternate fashion by maximizing the objectives derived from (15) and the Bellman equation. We have included a diagram in appendix A to provide a visual illustration of the Q-learning training process. To be specific, the training process is used to solve the following optimization problems alternately

$$\max_{\Phi} \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{a},r,\boldsymbol{x}')\sim\mathbb{P}} Q(\boldsymbol{x},A(\boldsymbol{x};\Phi);\Psi) \tag{16}$$

and

$$\min_{\Psi} \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{a},r,\boldsymbol{x}')\sim\mathbb{P}} \left[Q(\boldsymbol{x},\boldsymbol{a};\Psi) - (r(\boldsymbol{x},\boldsymbol{a}) + \eta Q(\boldsymbol{x}',A(\boldsymbol{x}';\Phi);\Psi))\right]^2. \tag{17}$$

Reward. Note that the solution of the second problem (17) depends on how the reward function $r(\boldsymbol{x},\boldsymbol{a})$ is defined. Because our ultimate goal is to identify connective configurations that are expected to have a high reactive density $\rho(\boldsymbol{x})$, ideally, we would like to use $\rho(\boldsymbol{x}')$, where $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{a}$, as the reward function. The difficulty is that, as we indicated earlier, $\rho(\boldsymbol{x})$ is defined in terms of the committor function $q(\boldsymbol{x})$ which is unknown in general. Therefore, setting $r(\boldsymbol{x},\boldsymbol{a})$ to the exact $\rho(\boldsymbol{x}')$ is not practical.

However, as $q(\boldsymbol{x})$ is defined as the probability of a trajectory starting from a point $\boldsymbol{x}$ and reaching B before reaching A, we can estimate this probability numerically by shooting trajectories from $\boldsymbol{x}$, and performing statistical analysis of these trajectories. To be specific, we propose to use a shooting procedure to estimate $q(\boldsymbol{x})$ by counting the number of trajectories originating from $\boldsymbol{x}$ with a random momentum and terminating in one of the metastable regions A or B within a fixed number of time steps.

We shoot N trajectories from $\boldsymbol{x}$ with a random momentum or force. Let N_A be the number of trajectories reaching A first rather than B within a fixed number of time T, and N_B be the number of trajectories reaching B first rather than A. Typically, T is far smaller than the time scale required to observe a reactive trajectory. Clearly, $N_A + N_B \leqslant N$ since some trajectories may hover around the starting configuration for a long time and never reach either A or B within T time interval. We can view $N_A/(N_A+N_B)$ as an approximation to $q(\boldsymbol{x})$ and $N_B/(N_A+N_B)$ as an approximation to $1-q(\boldsymbol{x})$. Consequently, we can use

$$\hat{\rho}(\boldsymbol{x}) = \frac{N_A N_B}{(N_A+N_B)^2} p(\boldsymbol{x}) \tag{18}$$

as a proxy for $\rho(\boldsymbol{x})$. As a result, the reward function associated with the state action pair $(\boldsymbol{x},\boldsymbol{a})$ can be defined as

$$r(\boldsymbol{x},\boldsymbol{a}) := \frac{N_A N_B}{(N_A+N_B)^2} \cdot \frac{\exp(-\beta V(\boldsymbol{x}'))}{Z}, \tag{19}$$

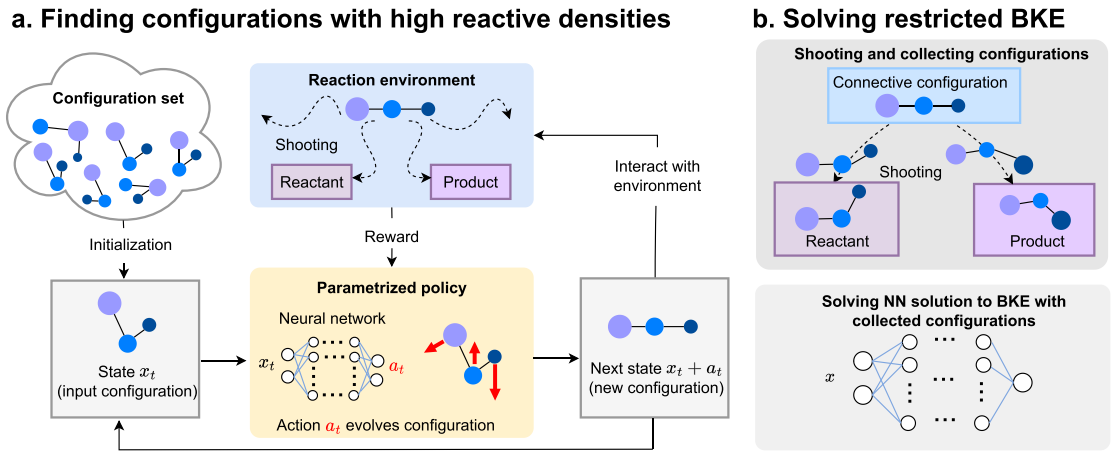


Figure 1. The proposed workflow. (a) A schematic description of the proposed RL action-reward feedback loop. In each episode, a sequence of states (configurations) $\{\boldsymbol{x}^0, \ldots, \boldsymbol{x}^t, \ldots\}$ is generated iteratively. Each iteration consists of the following stages. ① Initialization: a configuration $\boldsymbol{x}^0$ is randomly drawn from the configuration set; ② Evolution: the parameterized policy yields the action $\boldsymbol{a}^t$ after observing the state $\boldsymbol{x}^t$. After taking the action, the agent moves to a new configuration $\boldsymbol{x}^{t+1}=\boldsymbol{x}^t+\boldsymbol{a}^t$; ③ Interaction: we shoot multiple trajectories from $\boldsymbol{x}^{t+1}$, and evaluate its reward $r(\boldsymbol{x}^t, \boldsymbol{a}^t)$. The quadruples $(\boldsymbol{x}^t, \boldsymbol{a}^t, r(\boldsymbol{x}^t, \boldsymbol{a}^t), \boldsymbol{x}^{t+1})$ is used in Q-learning to update the policy. (b) We shoot multiple trajectories starting from the identified connective configurations. We collect configurations along these trajectories that characterize the reaction channels. We then use these configurations to solve the NN approximation to BKE.

where $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{a}$ is a new state. In practice, we can ignore the constant Z in (19).

Once the NNs are properly trained, the optimal action to be taken at each $\boldsymbol{x}$ satisfies

$$Q^*(\boldsymbol{x}, A(\boldsymbol{x})) = \max_{\boldsymbol{a}} Q^*(\boldsymbol{x}, \boldsymbol{a}).$$

RL is a multi-episode iterative learning process. In each episode, a random configuration $\boldsymbol{x}^0$ is chosen to start the learning process. A NN that represents the action function $A(\boldsymbol{x})$ takes the configuration as the input and generates an action $\boldsymbol{a}^0$ as the output. Taking such an action yields a new configuration $\boldsymbol{x}^1$ for which a reward r can be obtained by shooting several trajectories from $\boldsymbol{x}^1$ and evaluating (19). This process can be repeated several times until we generate a sequence of state action pairs $\{(\boldsymbol{x}^t, \boldsymbol{a}^t)\}$. Figure 1 gives a schematic illustration of the process of generating a sequence of state action pairs within a single episode.

The generated sequence of state action pairs, together with the rewards evaluated for states form the training data set $\mathbb{P}$ that we use to optimize the NN representations of the Q-function $Q(\boldsymbol{x}, \boldsymbol{a})$ and action $A(\boldsymbol{x})$ by solving the minimization problems (16) and (17) in an alternate fashion. This data set $\mathbb{P}$ is continuously updated as the agent interacts with the environment and adapts its policy over time. The action network and Q network are trained by resampling from the $\mathbb{P}$ collection. The reward function $r(\boldsymbol{x}, \boldsymbol{a})$ guides the update of the action network to generate valid actions. In addition, In order to discourage the agent from exploring regions with extremely low Boltzmann–Gibbs distribution probabilities, we interrupt an episode when a configuration ends up in an area of very low probability. As a result, the action strategy directs the state away from these low-probability regions, thus strengthening the generation of valid actions. The RL code incorporates a user-defined parameter to regulate the range of the action strategy. This prevents the action strategy from producing excessively large values, as any actions that surpass this predefined range are clipped.

We use the TD3 method to perform the optimization of $A(\boldsymbol{x})$ and $Q(\boldsymbol{x}, \boldsymbol{a})$. This variant of the DDPG method uses a variety of techniques to improve the stability of the training process and mitigate the risk of potentially over-estimating the Q-value function. In particular, TD3 uses the moving average of parameters associated with multiple NNs to solve (16) and (17). Furthermore, TD3 uses a 'delayed' policy update, where the policy network (which is used to determine the optimal action) is only updated after a certain number of Q-network updates. This delayed updating scheme helps to stabilize the training process and reduces the likelihood of the policy network training being stuck in an undesirable local minimum.

The main steps of a multi-episode RL algorithm for seeking the optimal policy that allows us to quickly identify connection configurations from an arbitrary starting configuration is shown in algorithm 1.

Algorithm 1. An RL algorithm for seeking connective configurations.

Input: Random parameter initialization Ψ_1 and Ψ_2 of critic networks and Φ of actor network.
Output: Action network $A(\cdot;\Phi)$.

Initialize target networks $\Psi_1' \leftarrow \Psi_1$, $\Psi_2' \leftarrow \Psi_2$ and $\Phi' \leftarrow \Phi$
Replay buffer $\mathbb{P} \leftarrow \emptyset$
for episode from 1 to the maximal number of episodes **do**
  $\boldsymbol{x}^0 \sim \text{Uniform}(\Omega)$ ▷ Start a new episode
  for t from 0 to $L-1$ **do**
    Obtain a new action $\boldsymbol{a}^t = A(\boldsymbol{x}^t;\Phi)$ and a new state $\boldsymbol{x}^{t+1} = \boldsymbol{x}^t + \boldsymbol{a}^t$ ▷ New state update rule
    if $p(\boldsymbol{x}^{t+1})$ is smaller than a threshold **then** Break
    end if
    Compute the reward $r(\boldsymbol{x}^t, \boldsymbol{a}^t)$ using (19) ▷ Compute the reward by shooting
    Append $(\boldsymbol{x}^t, \boldsymbol{a}^t, r(\boldsymbol{x}^t, \boldsymbol{a}^t), \boldsymbol{x}^{t+1})$ to $\mathbb{P}$
  end for
  for tt from 0 to t **do**
    Sample a mini-batch $(\boldsymbol{x}, \boldsymbol{a}, r, \boldsymbol{x}')$ of size B from $\mathbb{P}$
    $y \leftarrow r + \eta \min\{Q(\boldsymbol{x}', A(\boldsymbol{x}';\Phi');\cdot)) : \Psi_1', \Psi_2'\}$
    Update Ψ_i with the loss $\frac{1}{B}\sum (Q(\boldsymbol{x}, \boldsymbol{a}; \Psi_i) - y)^2$ for $i = 1, 2$
    if tt mod $policy\_delay = 0$ **then**
      Update Φ with the loss $\frac{1}{B}\sum -Q(s, A(s;\Phi);\Psi_1)$
      Update target networks: $\Phi' \leftarrow \tau\Phi + (1-\tau)\Phi'$ and $\Psi_i' \leftarrow \tau\Psi_i + (1-\tau)\Psi_i'$ for $i = 1, 2$
    end if
  end for
end for

3.2. Generating reaction channels and computing reaction rate constant

After performing several episodes of RL using algorithm 1, we obtain an optimal action function $A(\cdot;\Phi^*)$ parameterized by Φ^*. Such a function yields the policy we follow at each configuration $\boldsymbol{x}$ to quickly move towards a connective configuration. Figure 2(a) shows how such a policy (represented as the vector field) looks like for a simple potential energy surface. The value of $A(\boldsymbol{x})$, which is a vector with two components, is plotted as an arrow for each $\boldsymbol{x}$ uniformly sampled in $[-2.0, 2.0] \times [-1.5, 2.0]$. We see the arrows point to two configurations marked by crosses. These correspond to two connective configurations.

Once we identify a connective configuration, we perform additional shooting operations from that configuration to generate multiple trajectories originating from the connective configuration. The configurations generated along these trajectories are considered as samples within a reaction channel between A and B. If multiple connective configurations are identified, each one of them can be used to generate a distinct reaction channel. Specifically, we shoot N trajectories from each of the identified connective configurations using a time step of Δt up to a total time of T.

The configurations generated within reaction channels can be used to compute the committor function $q(\boldsymbol{x})$ by solving a restricted BKE on these configurations using a deep NN as we described in section 2.2. The utilization of NN-based PDE solver is often associated with an implicit bias towards fitting smooth functions that exhibit fast decay in the frequency domain [48]. Consequently, This bias can make it challenging for NN models to capture drastic changes in the committor function. However, by choosing an appropriate training dataset, one can mitigate this bias [49]. To generate an appropriate dataset, one can adjust how to perform shooting from the identified connective configurations so that enough configurations cover the area of interest. Both the hyperparameters T and N can be tuned, however the minimum value of these parameters to guarantee accurate estimates is understood [24]. The choice of T, the trajectory length depends on the relaxation timescale of the system and can be multiple order of magnitudes smaller than the first passage time of the reaction. The dependence of the estimate on N, the number of trajectories can be computed by noting that the outcome of the individual trials corresponds to a Bernoulli distribution. Hence, getting accurate estimate of the committor along the isocommittor surface requires the most number of samples. However, even for those points, one can get estimates within value of 0.1 with $N = 20$. In appendix D.2, we demonstrate the advantage of using configurations within reaction channels to train the NN designed to solve the restricted BKE.

The NN not only returns $q(\boldsymbol{x})$ for each $\boldsymbol{x}$ within all reaction channels, but also its gradient $\nabla q(\boldsymbol{x})$. This will allow us to compute the reactive flux at each $\boldsymbol{x}$ within the reaction channel. As a result, by using (6), we can calculate the rate constant.

4. Numerical results

In this section, we present several numerical experiments that demonstrate the effectiveness of the RL method introduced in section 3 for identifying connective configurations and using them to generate configurations that characterize reactive channels. Our experiments were performed on model potentials (triple-well and rugged Muller potentials) as well as the Alanine dipeptide (ADP) molecule in vacuum. The triple-well potential contains several reaction pathways, and we utilize this potential to evaluate the effectiveness of the proposed RL method in identifying different connective configurations within these channels. The choice of the Rugged Muller-Brown potential is motivated by its rough energy landscape, including a number of local minima between the states of interest. This selection is used to assess the ability of the proposed RL method in handling complex systems. We examined two numerical models at distinct temperatures: a lower temperature and a relatively higher one. As temperature decreases, observing transitional paths and reaction channels becomes increasingly challenging. This choice of two temperatures aims to verify the capacity of the proposed RL method to identify reaction channel across a range of temperature conditions, particularly low temperatures. While we demonstrate the RL results with initializations uniformly sampled from Ω in this section, we have the flexibility to relax this constraint by considering initializations from metastable states (see appendix D.1).

4.1. Potential with multiple reaction pathways

We consider the triple-well potential defined by

$$V(x_1,x_2) = 3e^{-x_1^2-\left(x_2-\frac{1}{3}\right)^2} - 3e^{-x_1^2-\left(x_2-\frac{5}{3}\right)^2} - 5e^{-(x-1)^2-x_2^2} - 5e^{-(x_1+1)^2-x_2^2} + 0.2x_1^4 + 0.2\left(x_2-\frac{1}{3}\right)^4. \quad (20)$$

We focus on the domain $\Omega = [-2,2]\times[-1.2,2]$. Figure 2(a) shows this potential as a color-mapped image. The two meta-stable regions A and B are defined by

$$A = \{\boldsymbol{x} : V(x_1,x_2) < -2 \text{ and } x_1 \leqslant -0.1\}, \quad B = \{\boldsymbol{x} : V(x_1,x_2) < -2 \text{ and } x_1 \geqslant 0.1\}.$$

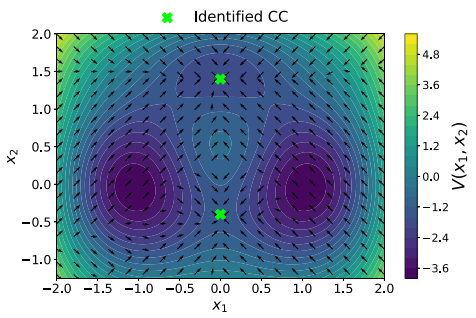
We can see from figure 2(a) that there are two transition paths from A to B. The top transition path goes through the third well in the top part of the image and two transition states between the third well and A (B). The bottom transition path goes directly from A to B and crosses the transition state. Because the potential function is symmetric with respect to $x_1 = 0$, all configurations along $\{\boldsymbol{x} : x_1 = 0\}$ have an equal probability of reaching either A or B. As a result, $\{\boldsymbol{x} : x_1 = 0\}$ represents the half-isocommittor surface. Here, the half-isocommittor surface is a set of configurations whose committor value is 0.5, i.e. $\{\boldsymbol{x} : q(\boldsymbol{x}) = 0.5\}$. We experimented with both a relatively low-temperature regime $\beta = 6.67$ and a relatively high-temperature regime $\beta = 1.67$. We discuss the results for $\beta = 6.67$ here, which is more challenging for studying rare events. We report a similar observation for the $\beta = 1.67$ case in appendix B.1. By default, the friction coefficient is set as 1.

Identifying reaction channels. In the presented experiments, the initial configuration for each RL episode is randomly sampled from a uniform distribution of configurations in Ω. The reward $r(\boldsymbol{x},\boldsymbol{a})$ (19) is obtained by shooting $N = 50$ trajectories from $\boldsymbol{x}$. The maximal number of evolution steps is set to $L = 15$. The maximum number of episodes used in RL is set to 1000. While we initially set a large number of episodes for the RL training process, it is worth noting that the convergence of RL does not require an extensive number of episodes. Additional discussion can be found in appendix B.3.

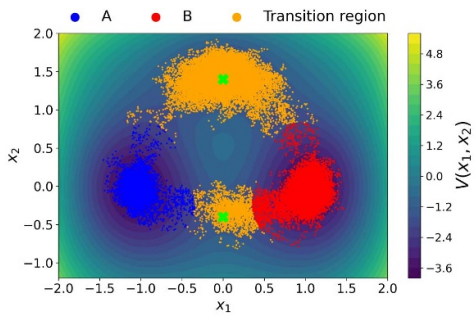
Figure 2(a) shows the learned action $A(\boldsymbol{x};\Phi^*)$ as a vector field that represents an optimal policy. Two attractors can be seen from this policy field. They correspond to two connective configurations located in two different transition paths.

From each of the identified connective configuration, we shoot 50 trajectories by simulating the overdamped Langevin dynamics (1) using the Euler–Maruyama scheme. We choose a uniform time step size $\Delta t = 5\times10^{-3}$, and propagate the solution from $T = 0.0$ to $T = 2.0$. The total number of configurations generated along these trajectories is 40 000. These configurations lie either in the metastable region A or B or two transition regions between A and B as shown in figure 2(b). These two transition regions correspond to the two reactive channels associated with this potential energy surface.

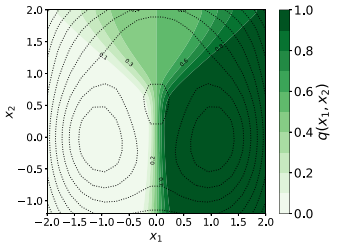
Solving BKE. We then solve the restricted BKE in the identified reaction channels by using the NN-based solver discussed in section 2.2. The loss function (11) is optimized by the Adam optimizer [44]. The hyperparameters used in the NN are listed in appendix C. Figure 2(d) shows the contour plots of the NN solution (colored contour lines) and the potential (dotted contour lines). The 0.5-level set of NN solution marked in the figure almost coincides with the true half-isocommittor $\{\boldsymbol{x} : x_1 = 0\}$. As a reference, we also used the finite difference method (FDM) to solve the BKE on the entire domain Ω. The absolute difference



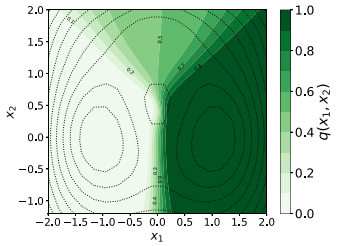
(a) Learned action function (vector field) and the connective configurations it reveals (crosses).



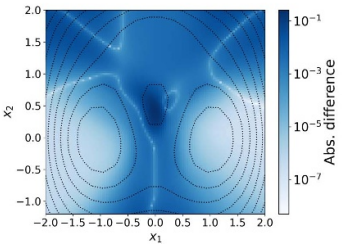
(b) Sampled configurations generated by shooting from two identified connective configurations.



(c) The FDM solution of (2) on Ω.



(d) The NN solution of (2).



(e) The difference between (c) and (d).

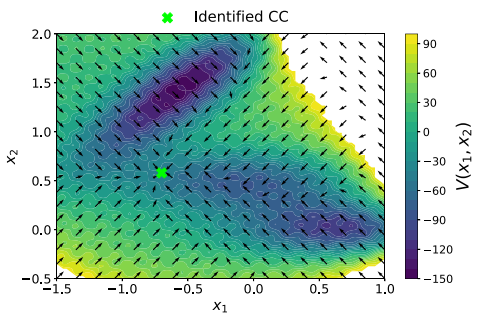
Figure 2. The results obtained from running the RL algorithm for Triple-well potential at inverse temperature $\beta = 6.67$. (a) The action function (vector field) learned by the proposed RL. The action function reveals two connective configurations (crosses). (b) The configurations generated by shooting trajectories initiated at the identified connective configurations. (c)–(e) The FDM (c) and NN (d) solutions of the BKE (2) and their difference (e).

Table 1. Reaction rate of the triple-well potential under various temperatures. 'N/A' means that no reactive trajectory is observed in the long trajectory of time $T = 2 \times 10^7$.

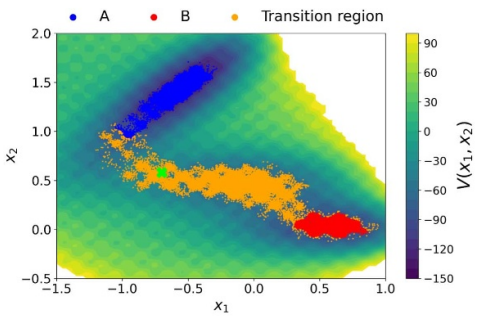
	β	
Methods	1.67	6.67
Direct simulation	2.18×10^{-2}	N/A
Finite difference (equation (6))	2.16×10^{-2}	7.43×10^{-8}
NN (equation (6))	2.00×10^{-2}	7.11×10^{-8}
NN (equation (8))	1.99×10^{-2}	7.32×10^{-8}

between the NN and FDM solutions on a 100×100 uniform mesh is shown in figure 2(e). We observe small errors in the NN solution in the regions that contain a large number of sampled configurations (such as the top transition path) and relatively large errors in regions where configurations are sparsely sampled (such as the region near $(0.0, 0.5)$).

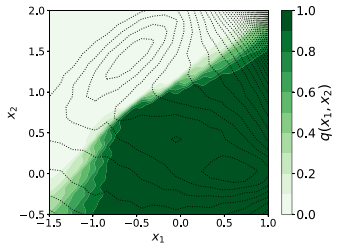
Rate estimation. The NN approximation to the committor function and its gradient on configurations with two reactive channels are used to calculate the reaction rate by integrating (5) over the dividing sub-surfaces identified with each reaction channel based on the data. We compare the computed rate with the one obtained from a direct dynamics simulation (see section 2.1) and that computed from the committor function obtained from the finite difference solution of the KBE on the entire domain Ω in table 1. We list the computed rates for both $\beta = 1.67$ and $\beta = 6.67$. In the direct dynamics simulation, we generate a long trajectory by time evolving the solution to the overdampled Langevin equation to $T = 2 \times 10^7$. In the FDM calculation, the rate is computed with numerical integration of (5) on the entire line segment $\{\boldsymbol{x} : x_1 = 0, -1.2 \leqslant x_2 \leqslant 2\}$. From these numerical experiments, we find that the generated configurations mainly cross the line segment $\{\boldsymbol{x} : x_1 = 0, -1.0 \leqslant x_2 \leqslant 2.2\}$ for $\beta = 1.67$ and line segments $\{\boldsymbol{x} : x_1 = 0, -0.8 \leqslant x_2 \leqslant 0.0, 0.8 \leqslant x_2 \leqslant 1.8\}$ for $\beta = 6.67$. We then compute the rates using the NN solutions on these segments. As we can see, the NN solution on reaction channels gives comparable rates as the one obtained by the FDM and a direct simulation. Finally, we validate the rate calculation with the proposed NN solution using the formula (8).



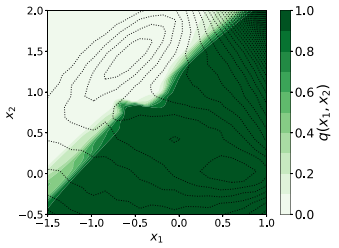
(a) Learned action function (vector field) and the connective configurations it reveals (crosses).



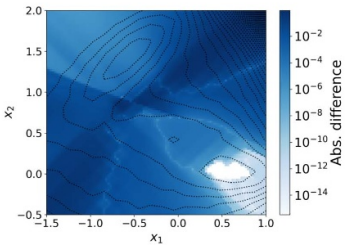
(b) Sampled configurations generated by shooting trajectories from the identified connective configuration.



(c) The FDM solution of (2) on Ω.



(d) The NN solution of (2).



(e) The difference between (c) and (d).

Figure 3. The results obtained from running the RL algorithm for the Rugged Muller potential at the inverse temperature $\beta = 0.25$. (a) The action function (vector field) learned by the RL algorithm. It reveals one connective configuration (cross). (b) The configurations generated by shooting trajectories from the identified connective configurations. (c)–(e) The FDM (c) and NN (d) solutions of the BKE (2) and their difference (e).

4.2. Potential with rough landscape

In the second example, we consider the rugged Muller potential on the domain $\Omega = [-1.5, 1] \times [-0.5, 2]$. The potential function is defined by

$$V(x_1, x_2) = \sum_{i=1}^{4} D_i \exp\left[a_i (x_1 - X_i)^2 + b_i (x_1 - X_i)(x_2 - Y_i) + c_i (x_2 - Y_i)^2\right] + \gamma \sin(2k\pi x_1) \sin(2k\pi x_2). \tag{21}$$

Here the parameters γ and k control the roughness of the landscape, which are set to 9 and 5, respectively. Other model parameters $(a_i, b_i, c_i, X_i, Y_i, D_i, i = 1, \ldots, 4)$ are exactly the same as the ones used in [14, 15]. By default, the friction coefficient is set as 1.

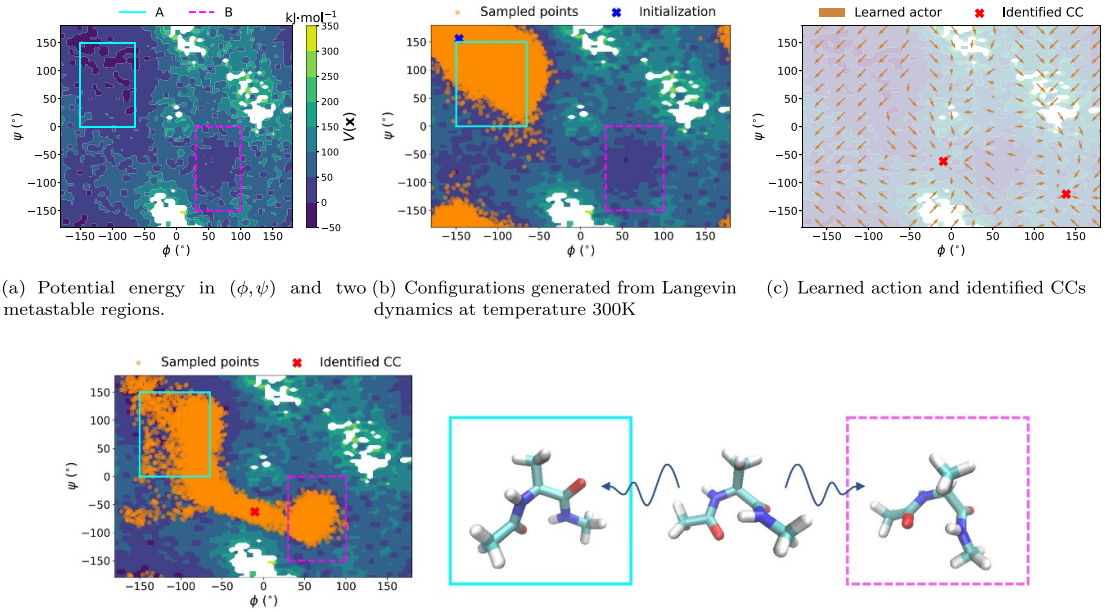
Identifying reaction channels. We discuss the numerical results in the low temperature regime $\beta = 0.25$ (figure 3) here and refer readers to appendix B.2 for results obtained for $\beta = 0.1$. In this example, the reward in the RL algorithm is calculated by shooting $N = 20$ trajectories up to $T = 0.25$. The step size used in each trajectory is set to 5×10^{-5}. Each RL episode consists of $L = 20$ steps (actions). We ran the RL algorithm for 1000 episodes. Figure 3(a) shows that the learned policy points to a single connective configuration. From that configuration, we shoot 50 trajectories using the Euler–Maruyama scheme. These trajectories contain 100 000 configurations that lie in the metastable regions A and B as well as the transition region in between (figure 3(b)). The latter is viewed as the reactive channel for this particular potential energy surface. The hyperparameter setting for the NNs used in the RL algorithm is listed in appendix C.

Solving BKE. We use the configurations contained in the reactive channel to solve the BKE via a NN. Figures 3(c) and (d) show the contour plots of the solution to the BKE obtained from both the FDM and the NN, respectively. The difference between the two is also shown in figure 3(e). From these plots, we observe that the NN solution agrees well with the FDM solution. In particular, the NN solution captures drastical changes near the point $(-0.8, 0.6)$.

Rate estimation. The computed committor functions obtained by the FDM and the NN approach are used to compute the reaction rate under different β values. The computed rates are compared with those obtained from direct numerical simulations of the corresponding overdamped Langevin dynamics in table 2. In direction numerical simulations, we ran a long trajectory until $T = 1.5 \times 10^4$ with a time step size of 10^{-5}. When $\beta = 0.25$, no reactive trajectory is observed. When the committor function is obtained from the FDM, the rate is computed from the numerical integration of (5) on the dividing surface $\{x_1 = 0.0,$

Table 2. Reaction rate of the rugged Muller potential under various temperatures. 'N/A' means that no reactive trajectory is observed in the long trajectory of time $T = 1.5 \times 10^4$.

	β	
Methods	0.1	0.25
Direct simulation	4.75×10^{-3}	N/A
Finite difference (equation (6))	4.31×10^{-3}	1.78×10^{-10}
NN (equation (6))	4.36×10^{-3}	6.27×10^{-10}



(a) Potential energy in (ϕ, ψ) and two metastable regions.

(b) Configurations generated from Langevin dynamics at temperature 300K

(c) Learned action and identified CCs

(d) Reaction channel generated from CC_1

(e) Change of configuration from from CC_1 to metastable states.

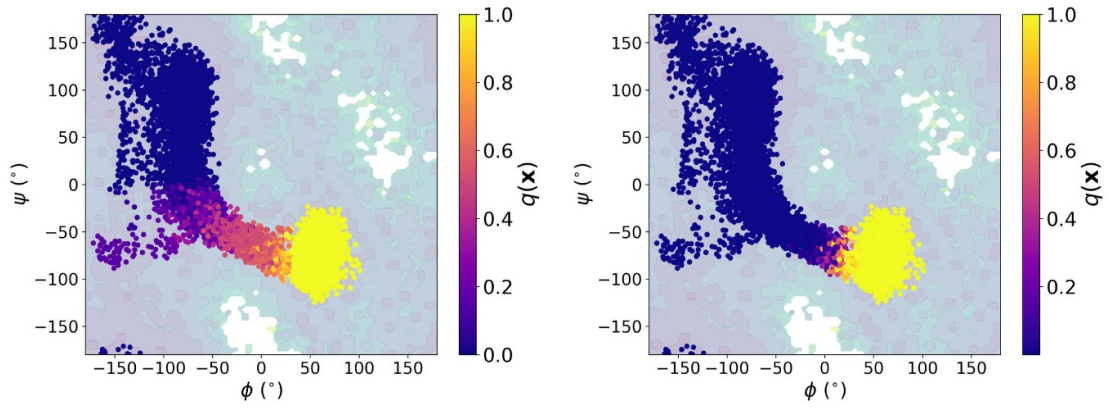
Figure 4. Alanine dipeptide in vacuum under temperature 300 K. (a) The potential energy landscape visualized in (ϕ, ψ)-space and two metastable states (solid and dotted rectangle). (b) configurations generated from overdamped Langevin dynamics at temperature 300 K. (c) The action function learned by the proposed RL algorithm. The action function reveals two connective configurations (CC's). (d) Configurations generated by shooting from the identified connective configurations. (e) The change of molecule configuration from the identified connective configuration to metasable states.

$-0.5 \leqslant x_2 \leqslant 2.0\}$. When using an NN to compute the commitor function within the reactive channel, the reaction rate is computed from the numerical integration of (5) along the line segments $\{x_1 = -0.5, 0.0 \leqslant x_2 \leqslant 0.8\}$ for $\beta = 0.1$. When $\beta = 0.25$, we calculate the rate by integrating (5) along the line segment $\{x_1 = -0.5, 0.3 \leqslant x_2 \leqslant 0.65\}$. We observe that the rates computed by all three methods are comparable when $\beta = 0.1$. When $\beta = 0.25$, the rates obtained from the NN and FDM approximation of the committor function have the same magnitude.

4.3. ADP in vacuum

In this example, we show how the RL algorithm introduced above can be used to identify a reaction channel of an ADP molecule in vacuum that corresponds to its isomerization process. In the following numerical experiment, we set the temperature to 300 K. The ADP molecule contains 22 atoms (see figure 4(e)). Therefore, the dimension of the configuration space is 66. The isomerization process is principally described by two the dihedral angles $(\phi, \psi) \in [-180°, 180°]^2$ of a subset of atoms (indexed by $4, 6, 8, 14$ and $6, 8, 14, 16$, respectively.) With a slight abuse of notation, we use $\phi(\boldsymbol{x})$ and $\psi(\boldsymbol{x})$ to donate the mapping from a configuration $\boldsymbol{x}$ to the two specific torsion angles. Figure 4(a) shows the potential energy landscape in the (ϕ, ψ)-space. The plot is constructed as follows. We first generate a long trajectory at a relatively high temperature (1200 K). We denote the set of configurations along this trajectory by $\mathbb{S}$. The potential energy for each configuration in $\mathbb{S}$ is stored. Next, we discretize (ϕ, ψ) in $(-180°, 180°] \times (-180°, 180°]$ by generating a 100×100 uniform grid. For each (ϕ_i, ψ_j) pair on the grid, we define a neighborhood

$$C(\phi_i, \psi_j) = \{\boldsymbol{x} \in \mathbb{S} : |\phi(\boldsymbol{x}) - \phi_i| \leqslant 5°, |\psi(\boldsymbol{x}) - \psi_j| \leqslant 5°\}.$$



(a) Diffusion map. (b) NN.

Figure 5. A comparison of solutions of the BKE (2) obtained from a diffusion map and NN respectively at temperature 300 K.

If $C(\phi_i, \psi_j) \neq \emptyset$, we set the potential energy associated with (ϕ_i, ψ_j) to $V_{\min}$, where

$$V_{\min} = \min_{\boldsymbol{x} \in C(\phi_i, \psi_j)} V(\boldsymbol{x}).$$

Otherwise, the potential energy of (ϕ_i, ψ_j) is undefined and colored by white in figure 4. The two metastable regions A (solid box in figure 4(a)) and B (dotted box) are defined by $A = \{\boldsymbol{x} : -150^\circ \leqslant \phi(\boldsymbol{x}) \leqslant -65^\circ, 0^\circ \leqslant \psi(\boldsymbol{x}) \leqslant 150^\circ\}, B = \{\boldsymbol{x} : 30^\circ \leqslant \phi(\boldsymbol{x}) \leqslant 100^\circ, -150^\circ \leqslant \psi(\boldsymbol{x}) \leqslant 0^\circ\}$. Figure 4(b) shows the snapshots of one trajectory initiated at a configuration near A of $T = 1 \times 10^8$ fs when the temperature is 300 K and we can see that the no reactive trajectory is observed.

Identifying reaction channels. The proposed RL algorithm aims to find a connective configuration in the (ϕ, ψ)-space. Subsequently, we use this configuration to generate additional configurations that bridge two metastable states. We should note that, for the ADP system, the actions taken in the RL algorithm are defined in a low-dimensional space specified by (ϕ, ψ) whereas the shooting procedure used to evaluate the reward takes place in the 66-dimensional configuration space. To address this disparity in dimensionality, it is necessary to establish a one-to-one mapping between each (ϕ, ψ) pair and a configuration in the phase space. To this end, we first construct a configuration set $\mathbb{P}$ by generating trajectories at high temperature 1200 K. We then map a given torsion coordinate in (ϕ, ψ) back to the configuration space by choosing a configuration from $\mathbb{P}$ with lowest potential energy. We simulate the Langevin dynamics with a step size of 2 fs and friction 10ps^{-1} using the package Openmm Python API [50]. The reward for each action is computed by shooting 10 trajectories of $T = 2 \times 10^3$ fs with kinetic initialization randomly sampled from the Boltzmann–Gibbs distribution. Each RL episode consists of $L = 10$ steps (actions). Figure 4(c) shows the learned action function reveals 2 different connective configurations, i.e. $(-10^\circ, -62^\circ)$ and $(139^\circ, -120^\circ)$. However, our primary interest is in $(-10^\circ, -62^\circ)$ configuration, as it has been extensively studied in the existing literature [14, 20, 51]. We shoot 100 trajectories of $T = 2 \times 10^3$ fs from this connective configuration to generate the additional configurations that bridge two metastable states as shown in figure 4(d). We also visualize the change of molecular structures from the connective configuration to two metastable states in figure 4(e).

Solving BKE. Our next objective is to solve the BKE (2) in a 66-dimensional space. The obtained numerical solution is used to generate the plot of the approximate committor function in (ϕ, ψ). Such an approximation is then compared with the approximation obtained from the DM [8, 52] method. Note that it is not possible to use traditional PDE solvers, such as finite difference and finite element to solve (2) because they suffer from the curse of dimensionality, i.e. their computational cost increase exponentially with respect to dimension of the problem [42]. The DM method is another sample-based method that allows for the solution of the BKE to be approximated on an arbitrary set of configurations $\{\boldsymbol{x}^i\}$. However, it is important to note that DM may not always produce an accurate approximation of the derivatives at configurations near the boundary [53, 54]. This can result in less accurate DM solutions to BKE near the boundaries of a fixed domain.

We use the dataset $\{\boldsymbol{x}^i\}$ identified by the proposed RL method as shown in figure 4(d) and apply DM and NN method to get the solutions of the BKE. The solutions are presented in figure 5. We observe that the half-isocommittor region of the solution, i.e. the set of configurations on which the committor function value is close to 0.5, obtained from the DM method occupies a relatively large area in the (ϕ, ψ) plane,

whereas the half-isocommittor region defined by the NN solution is confined to a small area defined by $[-25°, 25°] \times [-80°, 25°]$. Our solution is found to be more consistent with the results presented in figure 1 panel 2 of [51].

Rate estimation. To estimate the reaction rate, we use formula (8) and approximate the evaluation of the integral using the generated configurations $\{\boldsymbol{x}^i\}$ obtained through the shooting method. This approach yields the approximation formula:

$$\kappa = \frac{\int_\Omega e^{-\beta V(\boldsymbol{x})} \beta^{-1} \|\nabla q(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x}}{\int_\Omega e^{-\beta V(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}} \approx \frac{\sum_{i=1}^N e^{-\beta V(\boldsymbol{x}_i)} \beta^{-1} \|\nabla q(\boldsymbol{x}_i)\|^2}{\sum_{i=1}^N e^{-\beta V(\boldsymbol{x}_i)}}. \tag{22}$$

Finally, evaluating (22) using the approximated solution of KBE, we obtain a rate of $1.58 \times 10^{-5}\,\mathrm{ps}^{-1}$. This calculated estimate is comparable to the approximated value $4.54 \times 10^{-5}\,\mathrm{ps}^{-1}$ (reported in figure 5 of [20]).

5. Conclusion and discussion

We presented a novel RL-based approach for identifying and characterizing an ensemble of configurations where reactive trajectories are likely to be found. The optimized action function returned from the RL algorithm reveals connective configurations that have high reactive probabilities. One of the key elements of the RL algorithm is the proper construction of a reward function that serves as a surrogate for measuring the reactive probability of a configuration, which is normally defined in terms of the value of the committor function. Because the exact committor function is generally unknown in advance, we employ a randomized shooting procedure to estimate its value at an arbitrary configuration. Using the identified connective configurations, we generate trajectories directed towards metastable regions. The configurations along these trajectories are utilized to define reactive channels on which a restricted BKE is solved by a NN-based PDE solver. The solution yields an approximate committor function evaluated within these channels. This committor function can then be used to estimate reaction rates. Our numerical results showcase the capability of our RL approach in identifying reaction channels across multiple model problems of different sizes. Furthermore, we attain a accurate approximation of the committor function on the reaction channels using a NN-based PDE solver.

While our RL method effectively identifies the reaction channels and the NN-based PDE solver allows us to approximate the committor function on the reaction channels, there are still several aspects that merit further exploration.

To achieve a more accurate estimation of the committor function, which is utilized in the reward function (19), it might be necessary to conduct a greater number of shooting operations, along with longer shooting durations. However, an increase in the values of N and T will inevitably escalate the associated costs. It becomes imperative to explore a reward function that maintains computational efficiency.

In our approach, we train a NN specifically designed to solve the restricted BKE within reaction channels. While the committor values outside the reaction channel might not be of primary interest, it is important to note that the approximated solution may be less accurate outside the reaction channel where limited data is available.

As shown in figure 4(c), our RL method reveals two connective configurations. In particular, it identifies one close to $[140°, -120°]$ that is rarely discussed in the literature. We can potentially gain new chemical insights from these identified reaction channels that have received little attention. Furthermore, we could explore applications of our method on more complex systems, such as ADP in solvent.

Data availability statement

No new data were created or analyzed in this study. The codes for implementation of the proposed RL method are available at the online data warehouse: https://github.com/LeungSamWai/ReinforcementLearning4ReactionChannel. The source codes are released under MIT license.

Acknowledgment

This material is based upon work supported by the U S Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of Basic Energy Science, Scientific Discovery through Advanced Computing (SciDAC) program under Contract No. DE-AC02-05CH11231. This work used the computational resources of the National Energy Research Scientific Computing (NERSC) center under NERSC Award ASCR-ERCAP m1027 for 2023, which is supported by the Office of Science of the U S Department of Energy under Contract No. DE-AC02-05CH11231.

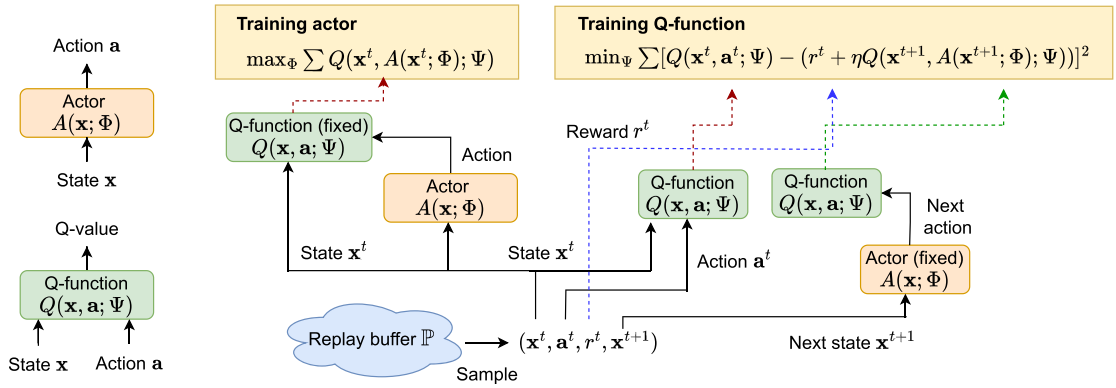


Figure 6. A schematic illustration of how the Q-learning is performed for a continuous action.

Appendix A. Q-learning algorithm

Figure 6 gives a schematic description of how the action function and the Q-function are optimized in the RL algorithm used to identify connective configurations.

Appendix B. Numerical results for high temperature regime

B.1. Triple-well potential

This section presents the results obtained from running the RL algorithm to identify connective configurations for a triple-well potential with an inverse temperature of $\beta = 1.67$. The reward function defined in equation (19) was evaluated by shooting 50 trajectories of that evolve up to time $T = 0.75$. The maximum number of time steps taken in each trajectory is set to $L = 20$. The RL procedure was run for a total of 1000 episodes.

Figure 7(a) shows the action function $A(x;\Phi)$ produced at the end of the RL run. We mark two distinct connectivity configurations identified by crosses. They are nearly identical to the connective configurations we found when the trajectories were generated using $\beta = 6.67$. In figure 7(b), we plot configurations generated by shooting trajectories from two connective configurations. We observe that the configurations generated from running trajectories using $\beta = 1.67$ cover a wider region of the configuration space that includes the local maximum near $(0.0, 0.5)$, as well as some configurations outside the pre-defined domain $\Omega = [-2, 2] \times [-1.2, 2]$. Figures 7(c)–(e) show that the NN solution to BKE is comparable to solution obtained from the FDM.

B.2. Rugged Muller potential

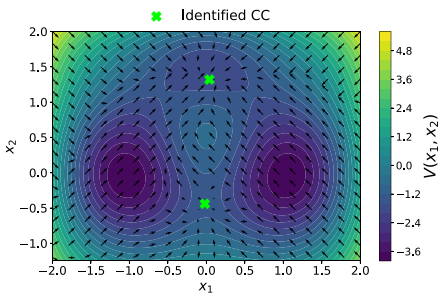
Figure 8 shows the results obtained from applying the previously presented RL algorithm to the Rugged Muller potential at an inverse temperature of $\beta = 0.1$. The reward function defined in equation (19) was computed based on shooting $N = 10$ trajectories that evolve up to time $T = 0.05$. The maximum number of time steps taken in each trajectory was set to $L = 20$. The RL algorithm was run for a total of 1000 episodes. We can observe that the configurations generated by shooting trajectories from the identified connective configurations with $\beta = 0.1$ (see figure 8(b)) cover a wider area compared to that obtained from performing RL and generating trajectories at $\beta = 0.25$. The committor function appears to be more stable near the region around $(-0.8, 0.6)$. The error is relatively small in the region covered by the sampled configurations.

B.3. Monitoring RL progress

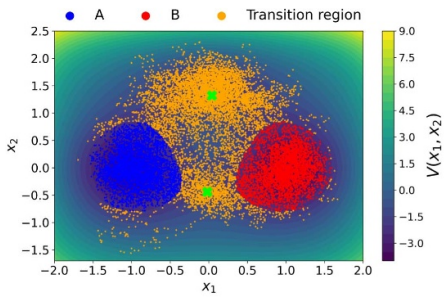
Figure 9 shows the reward as a function of the episode in the RL algorithm. A maximum of 1000 episodes are allowed. However, the largest rewards are achieved around episode 500. This suggests that the RL process is efficient and reaches a high level of performance before reaching the limit on the allowed number of episodes.

Appendix C. Implementation details of NN solutions of the BKE

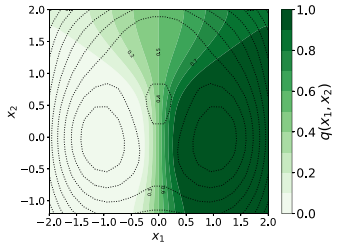
In this paper, we approximate the solution of the BKE by a fully connected neural network (FNN), which can be viewed as a composition of L simple nonlinear functions, i.e. $\phi(\mathbf{x};\boldsymbol{\theta}) := \sigma_2 \circ \mathbf{a}^\top \mathbf{h}_L \circ \mathbf{h}_{L-1} \circ \ldots \circ \mathbf{h}_1(\mathbf{x})$. Here, $\mathbf{h}_\ell(\mathbf{x}) = \sigma(\mathbf{W}_\ell \mathbf{x} + \mathbf{b}_\ell)$ with $\mathbf{W}_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$, $\mathbf{b}_\ell \in \mathbb{R}^{N_\ell}$ for $\ell = 1, \ldots, L$, $\mathbf{a} \in \mathbb{R}^{N_L}$, σ is the tanh function,



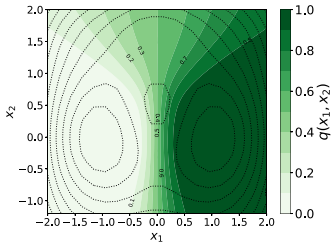
(a) Learned action function and the connective configurations it reveals.



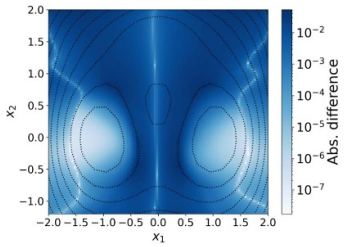
(b) Sampled configurations generated by shooting trajectories from the two identified connective configurations.



(c) The FDM solution of (2) on Ω

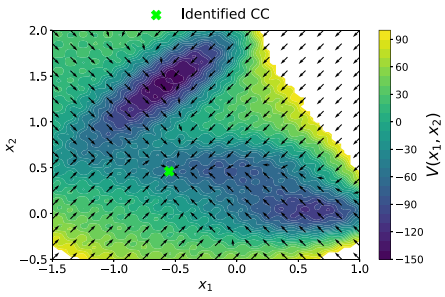


(d) The NN solution of (2).

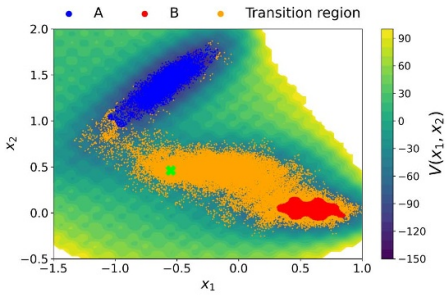


(e) The difference between (c) and (d).

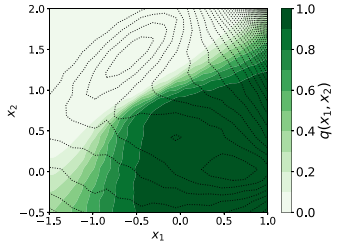
Figure 7. The results obtained from running the RL algorithm for Triple-well potential at inverse temperature $\beta = 1.67$. (a) The action function learned by our proposed RL. The action function reveals two connective configurations (crosses). (b) The configurations generated by shooting trajectories initiated at the identified connective configurations. (c)–(e) The FDM (c) and NN (d) solutions of the BKE (2) and their difference (e).



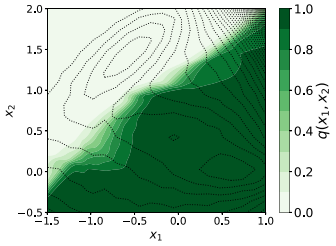
(a) Learned action function and the connective configurations it reveals.



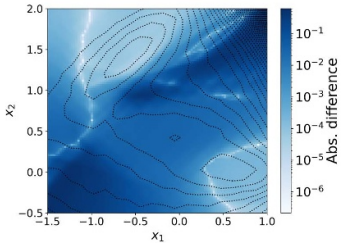
(b) Sampled configurations generated by shooting from identified connective configurations.



(c) The FDM solution of (2).



(d) The NN solution of (2).



(e) The difference between (d) and (d).

Figure 8. The results obtained from running the RL algorithm for the Rugged Muller potential at the inverse temperature $\beta = 0.1$. (a) The action function learned by the RL algorithm. It reveals two connective configurations (crosses). (b) The configurations generated by shooting trajectories from the identified connective configurations. (c)–(e) The FDM (c) and NN (d) solutions of the BKE (2) and their difference (e).

and σ_2 is a sigmoid function such that the range of output is $[0, 1]$. We use an FNN with $L = 2$ and a uniform width m, i.e. $N_\ell = m$ for all $\ell \neq 0$.

We split the collected configurations into 90% for training and 10% for validation. We tuned the hyperparameters, such as the width of the FNN, the boundary penalty coefficient, and the number of training iterations, by monitoring the equation error on the validation set. We optimized the hyperparameters to

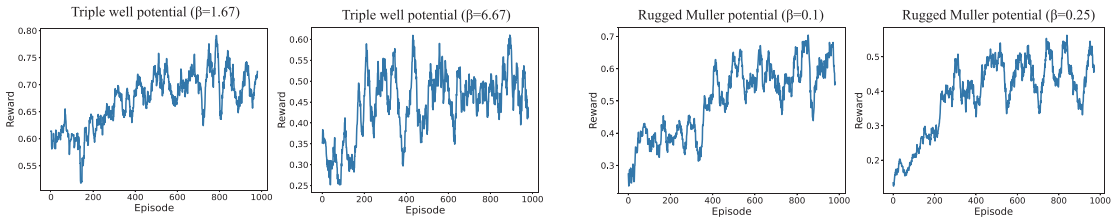


Figure 9. The moving average of the reward as a function of the episode number during the RL procedure. Here the value of the reward is linearly scaled to the interval $[0, 1]$ by min-max normalization.

Table 3. Hyperparameter setting for solving BKE with NN. Note that we use two different boundary coefficients for boundary conditions of A and B in the Alanine dipeptide case.

Example	Triple-well		Rugged Muller		Alanine dipeptide
β	1.67	6.67	0.1	0.25	1/2.5 kJ^{-1}mol
Width m	50	50	100	100	200
Boundary coefficient ℓ	10	100	10 000	100 000	1 million & 2 million
Iteration	30 000	30 000	100 000	30 000	100 000

obtain the low equation loss on the validation set. The FNN is optimized by Adam [44]. In Adam, we use an initial learning rate of 0.001 for T iterations. The learning rate is then adjusted in each iteration by a factor of $0.5(\cos(\frac{\pi t}{T}) + 1)$, where t is the current iteration number. We set the batch size to be the total number of training points. The hyperparameter setting for each numerical example is listed in table 3.

Appendix D. Additional notes on the RL method

D.1. RL initialization from stable states

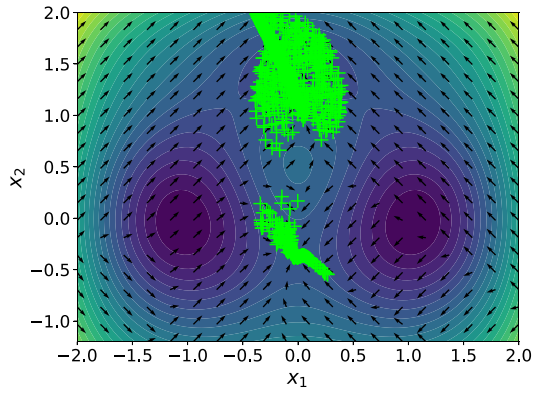
In the numerical results presented in section B, the initialization of the RL algorithm is performed by randomly sampling from Ω uniformly. In a revised approach, we revise the initialization scheme by sampling the initial configurations from the meta-stable regions. To show the impact of this change, we retrain the RL model and plot the final configurations obtained from RL procedure using 1000 different initializations sampled from the meta-stable region in figure 10.

With this initialization scheme, the RL method still successfully identifies two regions with a high reactive probability for the triple-well potential. Similarly, in the case of the Rugged Muller potential, the RL algorithm converges closely to the configurations depicted in figures 3(a) and 8(a). Furthermore, in the ADP case, the RL approach reveals reactive regions similar to those shown in figure 4(c). These findings demonstrate the robustness of the RL model with respect to different initialization strategies.

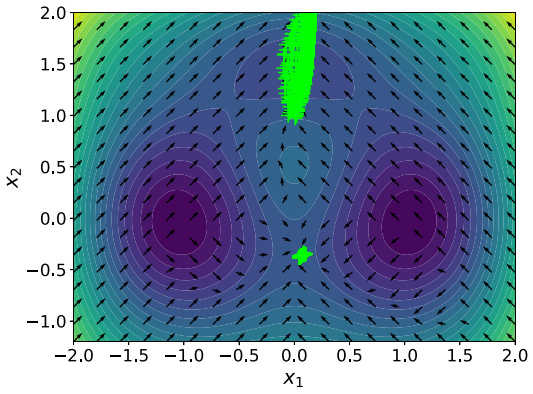
D.2. Solving the BKE using NN with different training datasets

The utilization of NN-based optimization methods is often associated with an implicit bias toward fitting smooth functions that exhibit fast decay in the frequency domain [48]. Consequently, training NN models that can be used to approximate the committor function can be challenging when we attempt to capture drastic changes in the committor function. Such implicit bias can be mitigated by using an appropriate training dataset [49]. By carefully selecting the training dataset to provide the necessary samples that encompass the desired variations in the committor function, we can overcome the limitations posed by the implicit bias of NN-based optimization. Here we compare NN solutions of the BKE trained on various datasets. These datasets consist of configurations sampled uniformly on Ω, configurations obtained from overdamped Langevin dynamics ran at a higher temperature, and configurations generated by shooting from CC's, as depicted in figures 11 and 12 respectively. We evaluate the performance of the NN on two examples (a triple-well potential and a Rugged Muller potential) at a low temperature. Figures 13 and 14 show the mean square error (MSE) of the training loss and the NN solutions obtained from different training datasets.

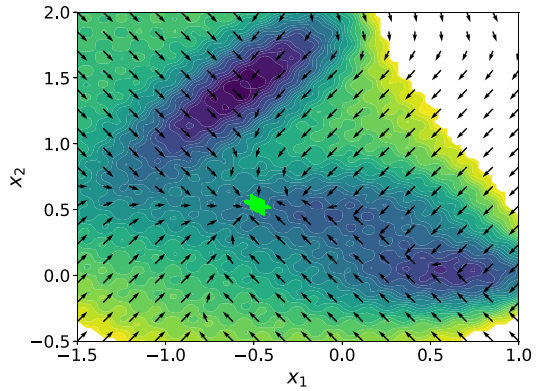
From these results, it is evident that the NN solution trained on the dataset generated by shooting trajectories from CC's achieves a lower MSE and yields a more accurate approximation.



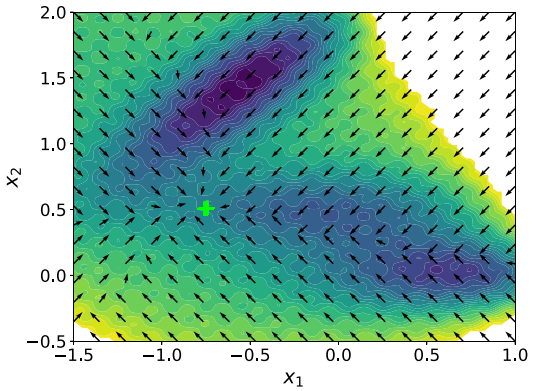
(a) Triple-well $\beta = 1.67$



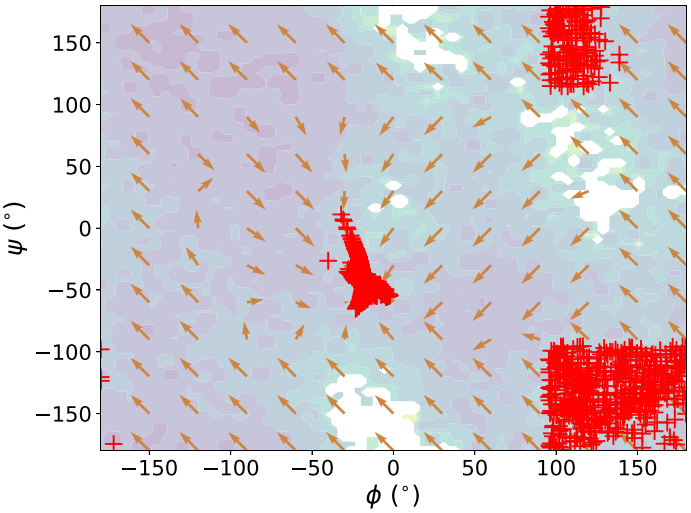
(b) Triple-well $\beta = 6.67$



(c) Rugged Muller $\beta = 0.1$

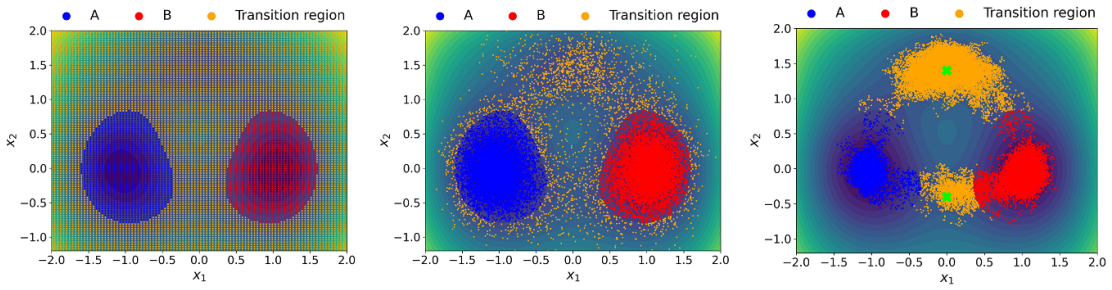


(d) Rugged Muller $\beta = 0.25$



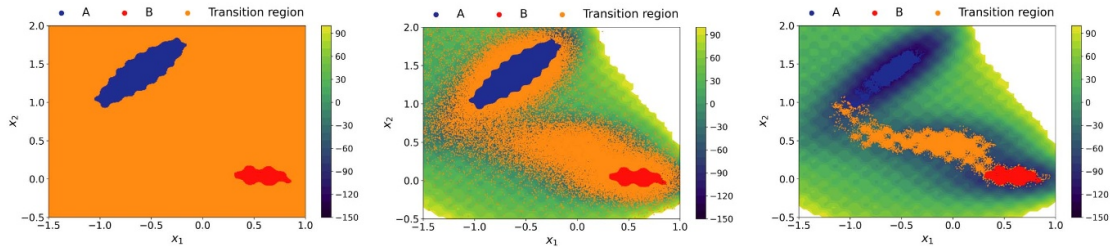
(e) Alanine dipeptide

Figure 10. The learned action produced by the RL algorithm where the initial configurations are randomly sampled from the meta-stable region uniformly. Each '+' represents the configuration generated in the final configuration of RL episode.



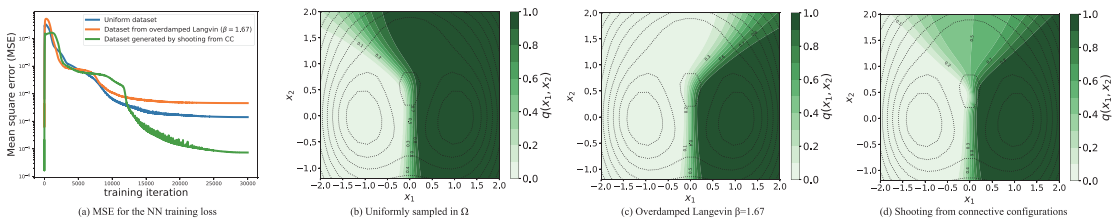
(a) Uniformly sampled in Ω (b) Overdampled Langevin β = 1.67 (c) Shooting from connective configurations

Figure 11. Different training datasets used for NN solutions of the BKE for the triple well potential with $\beta = 6.67$. (a) Uniform samples from Ω. (b) Samples along a overdamped Langevin dynamics trajectory with $\beta = 1.67$. (c) Configurations generated by shooting trajectories from the identified connective configurations.



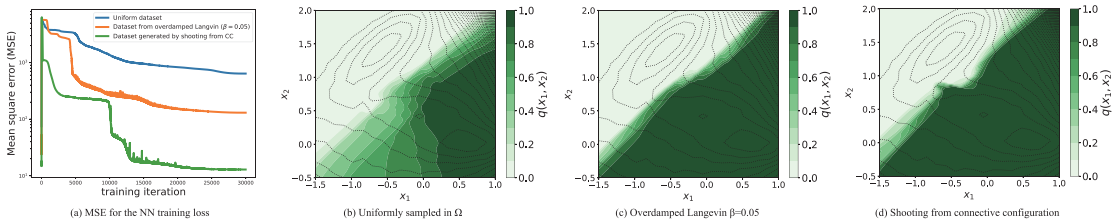
(a) Uniformly sampled in Ω (b) Overdamped Langevin (β=0.05) (c) Shooting from the connective configuration

Figure 12. Different training datasets used for NN solutions of the BKE for the Rugged Muller potential with $\beta = 0.25$. (a) Uniformly sampled in Ω. (b) Sample along a overdamped Langevin dynamics trajectory with $\beta = 0.05$. (c) Configurations generated by shooting from the identified connective configurations.



(a) MSE for the NN training loss (b) Uniformly sampled in Ω (c) Overdamped Langevin β=1.67 (d) Shooting from connective configurations

Figure 13. (a) Mean square error of the training loss at configurations sampled from the transition region of the triple well potential. (b)–(d) NN solutions obtained from different training datasets.



(a) MSE for the NN training loss (b) Uniformly sampled in Ω (c) Overdamped Langevin β=0.05 (d) Shooting from connective configuration

Figure 14. (a) Mean square error of the training loss at configurations sampled from the transition region of the rugged Muller-Brown potential. (b)–(d) NN solutions obtained from different training datasets.

ORCID iDs

Senwei Liang https://orcid.org/0000-0002-3558-6828
Aditya N Singh https://orcid.org/0000-0002-8019-2967
Yuanran Zhu https://orcid.org/0000-0001-6851-4161
David T Limmer https://orcid.org/0000-0002-2766-0688
Chao Yang https://orcid.org/0000-0001-7172-7539

References

[1] Peters B 2017 *Reaction Rate Theory and Rare Events* (Elsevier)
[2] Chandler D 1978 Statistical mechanics of isomerization dynamics in liquids and the transition state approximation *J. Chem. Phys.* **68** 2959–70
[3] Bolhuis P G, Chandler D, Dellago C and Geissler P L 2002 Transition path sampling: throwing ropes over rough mountain passes, in the dark *Annu. Rev. Phys. Chem.* **53** 291–318
[4] Geissler P L, Dellago C and Chandler D 1999 Kinetic pathways of ion pair dissociation in water *J. Phys. Chem.* B **103** 3706–10
[5] Weinan E, Ren W and Vanden-Eijnden E 2002 String method for the study of rare events *Phys. Rev.* B **66** 052301
[6] Vanden-Eijnden E, Vanden-Eijnden E 2006 Towards a theory of transition paths *J. Stat. Phys.* **123** 503–23
[7] Vanden-Eijnden E , Vanden-Eijnden E 2010 Transition-path theory and path-finding algorithms for the study of rare events *Annu. Rev. Phys. Chem.* **61** 391–420
[8] Coifman R R, Kevrekidis I G, Lafon S, Maggioni M and Nadler B 2008 Diffusion maps, reduction coordinates and low dimensional representation of stochastic systems *Multiscale Model. Simul.* **7** 842–64
[9] Thiede E H, Giannakis D, Dinner A R and Weare J 2019 Galerkin approximation of dynamical quantities using trajectory data *J. Chem. Phys.* **150** 244111
[10] Evans L, Cameron M K and Tiwary P 2022 Computing committors via mahalanobis diffusion maps with enhanced sampling data *J. Chem. Phys.* **157** 214107
[11] Trstanova Z, Leimkuhler B and Lelièvre T 2020 Local and global perspectives on diffusion maps in the analysis of molecular systems *Proc. R. Soc.* A **476** 20190036
[12] Ma A and Dinner A R 2005 Automatic method for identifying reaction coordinates in complex systems *J. Phys. Chem.* B **109** 6769–79
[13] Rotskoff G M, Mitchell A R and Vanden-Eijnden E 2022 Active importance sampling for variational objectives dominated by rare events: consequences for optimization and generalization *Mathematical and Scientific Machine Learning* (PMLR) pp 757–80
[14] Li Q, Lin B and Ren W 2019 Computing committor functions for the study of rare events using deep learning *J. Chem. Phys.* **151** 054112
[15] Khoo Y, Lu J and Ying L 2019 Solving for high-dimensional committor functions using artificial neural networks *Res. Math. Sci.* **6** 1–13
[16] Hasyim M R, Batton C H and Mandadapu K K 2022 Supervised learning and the finite-temperature string method for computing committor functions and reaction rates *J. Chem. Phys.* **157** 184111
[17] Strahan J, Finkel J, Dinner A R and Weare J 2023 Predicting rare events using neural networks and short-trajectory data *J. Comput. Phys.* **488** 112152
[18] Strahan J, Guo S C, Lorpaiboon C, Dinner A R and Weare J 2023 Inexact iterative numerical linear algebra for neural network-based spectral estimation and rare-event prediction *J. Chem. Phys.* **159** 014110
[19] Jung H *et al* 2023 Machine-guided path sampling to discover mechanisms of molecular self-organization *Nat. Comput. Sci.* **3** 334–45
[20] Singh A N and Limmer D T 2023 Variational deep learning of equilibrium transition path ensembles *J. Chem. Phys.* **159** 024124
[21] Zhu Y, Tang Y-H and Kim C 2023 Learning stochastic dynamics with statistics-informed neural network *J. Comput. Phys.* **474** 111819
[22] Dellago C, Bolhuis P G and Geissler P L 2002 Transition path sampling *Adv. Chem. Phys.* **123** 1–78
[23] Bolhuis P G and Swenson D W 2021 Transition path sampling as Markov chain Monte Carlo of trajectories: recent algorithms, software, applications and future outlook *Adv. Theory Simul.* **4** 2000237
[24] Dellago C, Bolhuis P G and Geissler P L 2006 Transition path sampling methods *Computer Simulations in Condensed Matter Systems: from Materials to Chemical Biology Volume 1* (Springer) pp 349–91
[25] Barducci A, Bussi G and Parrinello M 2008 Well-tempered metadynamics: a smoothly converging and tunable free-energy method *Phys. Rev. Lett.* **100** 020603
[26] Barducci A, Bonomi M and Parrinello M 2011 Metadynamics *Wiley Interdiscip. Rev.-Comput. Mol. Sci.* **1** 826–43
[27] Huber G A and Kim S 1996 Weighted-ensemble Brownian dynamics simulations for protein association reactions *Biophys. J.* **70** 97–110
[28] Zwier M C *et al* 2015 Westpa: an interoperable, highly scalable software package for weighted ensemble simulation and analysis *J. Chem. Theory Comput.* **11** 800–9
[29] Valsson O and Parrinello M 2014 Variational approach to enhanced sampling and free energy calculations *Phys. Rev. Lett.* **113** 090601
[30] Zhang J, Lei Y-K, Zhang Z, Han X, Li M, Yang L, Yang Y I and Gao Y Q 2021 Deep reinforcement learning of transition states *Phys. Chem. Chem. Phys.* **23** 6888–95
[31] Das A, Rose D C, Garrahan J P and Limmer D T 2021 Reinforcement learning of rare diffusive dynamics *J. Chem. Phys.* **155** 134105
[32] Das A, Kuznets-Speck B and Limmer D T 2022 Direct evaluation of rare events in active matter from variational path sampling *Phys. Rev. Lett.* **128** 028005
[33] Lelièvre T, Robin G, Sekkat I, Stoltz G and Cardoso G V 2022 Generative methods for sampling transition paths in molecular dynamics (arXiv:2205.02818)
[34] Das A and Limmer D T 2019 Variational control forces for enhanced sampling of nonequilibrium molecular dynamics simulations *J. Chem. Phys.* **151** 244123
[35] Rose D C, Mair J F and Garrahan J P 2021 A reinforcement learning approach to rare trajectory sampling *New J. Phys.* **23** 013013
[36] Allen R J, Valeriani C and Ten Wolde P R 2009 Forward flux sampling for rare event simulations *J. Phys.: Condens. Matter* **21** 463102
[37] Metzner P, Schütte C and Vanden-Eijnden E 2006 Illustration of transition path theory on a collection of simple examples *J. Chem. Phys.* **125** 084110
[38] Zhang X, Landsness E C, Culver J P and Anastasio M A 2022 Identifying functional brain networks from spatial-temporal wide-field calcium imaging data via a recurrent autoencoder *Neural Imaging and Sensing 2022* (SPIE) p C1194612
[39] Nan Y and Ji H 2020 Deep learning for handling kernel/model uncertainty in image deconvolution *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 2388–97
[40] Lin B, Li Q and Ren W 2022 A data driven method for computing quasipotentials *Mathematical and Scientific Machine Learning* (PMLR) pp 652–70

[41] Shen Z, Yang H and Zhang S 2021 Deep network with approximation error being reciprocal of width to power of square root of depth *Neural Comput.* **33** 1005–36
[42] Weinan E, Han J and Jentzen A 2021 Algorithms for solving high dimensional PDEs: from nonlinear Monte Carlo to machine learning *Nonlinearity* **35** 278
[43] Paszke A *et al* 2019 Pytorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems* vol 32
[44] Kingma D P and Ba J 2015 Adam: a method for stochastic optimization *3rd Int. Conf. on Learning Representations ICLR 2015* (*Conf. Track Proc.*) (*San Diego, CA, USA, 7–9 May 2015*) ed Y Bengio and Y LeCun
[45] Sutton R S and Barto A G 2018 *Reinforcement Learning: An Introduction* (MIT press)
[46] Silver D *et al* 2014 Deterministic policy gradient algorithms *Int. Conf. on Machine Learning* (PMLR) pp 387–95
[47] Fujimoto S, Hoof H and Meger D 2018 Addressing function approximation error in actor-critic methods *Int. Conf. on Machine Learning* (PMLR) pp 1587–96
[48] Xu Z-Q J, Zhang Y, Luo T, Xiao Y and Ma Z 2020 Frequency principle: Fourier analysis sheds light on deep neural networks *Commun. Comput. Phys.* **28** 1746–67
[49] Liang S, Huang Z and Zhang H 2022 Stiffness-aware neural network for learning Hamiltonian systems *Int. Conf. on Learning Representations*
[50] Eastman P *et al* 2017 Openmm 7: rapid development of high performance algorithms for molecular dynamics *PLoS Comput. Biol.* **13** e1005659
[51] Kikutsuji T, Mori Y, Okazaki K-I, Mori T, Kim K and Matubayasi N 2022 Explaining reaction coordinates of alanine dipeptide isomerization obtained from deep neural networks using explainable artificial intelligence (xai) *J. Chem. Phys.* **156** 154108
[52] Ko T *et al* 2023 Using diffusion maps to analyze reaction dynamics for a hydrogen combustion benchmark dataset *J. Chem. Theory Comput.* **19** 5872–85
[53] Liang S, Jiang S W, Harlim J and Yang H 2021 Solving PDEs on unknown manifolds with machine learning (arXiv:2106.06682)
[54] Jiang S W and Harlim J 2023 Ghost point diffusion maps for solving elliptic PDEs on manifolds with classical boundary conditions *Commun. Pure Appl. Math.* **76** 337–405