Proceedings of Machine Learning Research vol 145:652–670, 2021 2nd Annual Conference on Mathematical and Scientific Machine Learning

A Data Driven Method for Computing Quasipotentials

Bo Lin E0046836@U.NUS.EDU
Department of Mathematics, National University of Singapore, Singapore 119076

Qianxiao Li QIANXIAO@NUS.EDU.SG
Department of Mathematics, National University of Singapore, Singapore 119076

Weiqing Ren MATRW@NUS.EDU.SG
Department of Mathematics, National University of Singapore, Singapore 119076

Editors: Joan Bruna, Jan S Hesthaven, Lenka Zdeborova

Abstract

The quasipotential is a natural generalization of the concept of energy functions to non-equilibrium systems. In the analysis of rare events in stochastic dynamics, it plays a central role in characterizing the statistics of transition events and the likely transition paths. However, computing the quasipotential is challenging, especially in high dimensional dynamical systems where a global landscape is sought. Traditional methods based on the dynamic programming principle or path space minimization tend to suffer from the curse of dimensionality. In this paper, we propose a simple and efficient machine learning method to resolve this problem. The key idea is to learn an orthogonal decomposition of the vector field that drives the dynamics, from which one can identify the quasipotential. We demonstrate on various example systems that our method can effectively compute quasipotential landscapes without requiring spatial discretization or solving path-space optimization problems. Moreover, the method is purely data driven in the sense that only observed trajectories of the dynamics are required for the computation of the quasipotential. These properties make it a promising method to enable the general application of quasipotential analysis to dynamical systems away from equilibrium.

Keywords: Non-equilibrium Systems, Quasipotential, Machine Learning, Rare Events, Hamilton-Jacobi Equations

1. Introduction

Dynamical systems under the influence of random perturbations are widely used in scientific modelling, including nucleation events during phase transitions, chemical reactions and biological networks. For these systems, understanding the mechanism and statistics of transitions between stable states is of great interest, especially when the noise has very small amplitude. According to large deviation theory (1), the transition dynamics become predictable in the small noise limit, and is completely characterized by the *quasipotential*. The latter generalizes the notion of equilibrium potential to non-equilibrium systems. Consequently, the quasipotential landscape gives an intuitive description of the essential dynamical features of complex systems that are out of equilibrium (9; 10; 14; 15).

However, computing quasipotentials is a challenging problem, especially when the system is high dimensional, or when a global landscape is sought. To date, there are two classes of methods for computing the quasipotential. The first type relies on the variational formulation of the quasipotential based on the Freidlin-Wentzell action functional (2; 4; 5). Here, the value of the quasipotential with respect to two chosen points is computed based on the solution of a path-space minimization

problem. These methods have the advantage that they can handle high-dimensional systems, and moreover, a most likely transition path is identified together with the computation. However, the key disadvantage is the behavior of the quasipotential away from the chosen points (and a minimum action path connecting them) remains unknown. In particular, computing a quasipotential landscape is prohibitively expensive using such methods. The second class of methods is developed to compute the quasipotential on 2D or 3D meshes. These methods are based on the dynamic programming principle. At each step, the estimated quasipotential values at selected spatial points are updated by solving the associated Hamilton-Jacobi equation (6) or directly solving the action minimization problem locally (7; 8; 16). Contrasting with previous variational approaches in path space, these methods compute an entire quasipotential landscape and do not require *a priori* information to select special points of interest. However, due to the requirement of a discretization mesh, they are limited to low dimensional system, as the computational complexity and cost grow exponentially over dimensions.

In practical applications, it is often the case that we need to analyze transition events or asymptotic occupational probabilities in high dimensional spaces, e.g. applications in biological networks (10; 11; 12). For such computations, the quasipotential is a very useful object. Thus, it is of importance to develop a method that can effectively address the previously mentioned limitations. In this paper, we introduce a machine learning based method for computing quasipotential landscapes. The method is not only scalable to high dimensions, but also yields the entire quasipotential landscape. Moreover, it has the advantage that no explicit dynamical models are required, and the quasipotential can be constructed directly from sampled trajectory data. In fact, this method simultaneously learns the force field of the dynamical system from the trajectories. This makes the computation of quasipotential landscapes, thus the analysis of rare events, for practical applications a much more tractable task.

The paper is organized as follows. We first introduce some theoretical background in Section 2 and then propose the machine learning based method in Section 3, including parameterization of the orthogonal decomposition of the vector field and the loss function. In Section 4, we illustrate the proposed method on several numerical examples. Finally we draw the conclusions in Section 5.

2. Background

We consider a dynamical system driven by small white noise. Its evolution is described by the stochastic differential equation

$$d\mathbf{x} = \mathbf{f}(\mathbf{x})dt + \sqrt{\epsilon}d\mathbf{W}, \quad \mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d, \tag{1}$$

where $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$ is a continuously differentiable vector field, $\mathbf{W}$ is the standard Brownian motion and ϵ is a small parameter, typically identified as a scaled temperature. For a given continuous path $\boldsymbol{\varphi}(t) \in \mathbb{R}^d$ on the time interval $t \in [0, T]$, the Freidlin-Wentzell action functional of the path associated with the system is defined as

$$\mathcal{A}[\boldsymbol{\varphi}(\cdot); T] = \int_0^T \frac{1}{2} \left| \dot{\boldsymbol{\varphi}} - \mathbf{f}(\boldsymbol{\varphi}) \right|^2 dt. \tag{2}$$

Denote by $\mathbf{x}^\epsilon(t)$ the trajectory of the system (1) starting from $\boldsymbol{\varphi}(0)$. The Freidlin-Wentzell theory tells that for sufficiently small ϵ, δ, the probability that $\mathbf{x}^\epsilon(t)$ stays in the neighborhood of the path

$\boldsymbol{\varphi}(t)$ on the time interval $[0, T]$ can be estimated by

$$\mathbb{P}[\sup_{0\le t\le T} |\mathbf{x}^\epsilon(t) - \boldsymbol{\varphi}(t)| < \delta] \approx \exp(-\frac{1}{\epsilon}\mathcal{A}[\boldsymbol{\varphi}(\cdot); T]). \tag{3}$$

We assume that the deterministic dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ exhibits a finite number of stable equilibria or limit cycles, such that almost every trajectory of the system is asymptotically convergent to those isolated attractors. Let A be one of the attractors. The quasipotential at the state $\mathbf{x}$ with respect to the attractor A is defined as

$$U_A(\mathbf{x}) = \inf_{T>0} \inf_{\boldsymbol{\varphi}} \mathcal{A}[\boldsymbol{\varphi}(\cdot); T], \tag{4}$$

where the infimum of the action functional is taken over all time horizon $T > 0$ and all absolutely continuous paths $\boldsymbol{\varphi}$ connecting the attractor A and the state $\mathbf{x}$, *i.e.* $\boldsymbol{\varphi}(0) \in A$ and $\boldsymbol{\varphi}(T) = \mathbf{x}$. The quasipotential with respect to the attractor A describes the difficulty of exiting the basin of A for the system (1) when the strength of noise ϵ is small. According to the large deviation theory (1), the statistics of the escaping event from the attractor A can be estimated using the quasipotential. For instance, the maximum likelihood path from A to another attractor is characterized by the quasipotential - the tangent of the path is parallel to $\mathbf{f} + \nabla U_A$ along the path. Also, the expected exit time τ from the attractor A is determined by the minimum of the quasipotential on the boundary of the basin of A: $\lim_{\epsilon\to 0} \epsilon \log \mathbb{E}[\tau] = \min_{\mathbf{x}\in\partial\mathcal{B}(A)} U_A(\mathbf{x})$, where $\mathcal{B}(A)$ is the basin of the attractor A.

The central idea of our approach relies on an alternative characterization of the quasipotential through an orthogonal decomposition of the vector field. Suppose $\mathbf{f}$ can be decomposed as

$$\mathbf{f}(\mathbf{x}) = -\nabla V(\mathbf{x}) + \mathbf{g}(\mathbf{x}), \quad \text{with } \nabla V(\mathbf{x})^T \mathbf{g}(\mathbf{x}) = 0, \tag{5}$$

where the term $-\nabla V(\mathbf{x})$ is referred to as the potential component of $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ as the rotational component. It is proved in the following theorem that under certain conditions, $2V$ coincides with the quasipotential of system (1) up to an additive constant.

Under mild conditions, such an orthogonality decomposition exists. For instance, it is sufficient if the quasipotential defined in (4) is continuously differentiable (6).

Theorem 1 *Suppose the vector field* $\mathbf{f}$ *in the system* (1) *has the orthogonal decomposition* (5) *and* V *attains its strict local minimum at a point or limit cycle, denoted by* A*. If there is a bounded domain* $\mathcal{D}$ *containing* A *such that*

- V *is continuously differentiable in* $\mathcal{D} \cup \partial\mathcal{D}$*;*
- $V(\mathbf{x}) > V(A)$ *and* $\nabla V(\mathbf{x}) \neq 0$ *for all* $\mathbf{x} \in \mathcal{D} \cup \partial\mathcal{D}$ *and* $\mathbf{x} \notin A$*,*

then the quasipotential of the system (1) *with respect to the attractor* A *in the set* $\{\mathbf{x} \in \mathcal{D} \cup \partial\mathcal{D} : V(\mathbf{x}) \le \min_{\mathbf{y}\in\partial\mathcal{D}} V(\mathbf{y})\}$ *coincides with* $2V(\mathbf{x})$ *up to an additive constant.*

Proof See Ref. (1). ■

The quasipotential is related to the equilibrium distribution of the dynamical system perturbed by small noise. For the system with multiple attractors, each attractor corresponds to a local quasipotential. These local quasipotential can be used to construct the global quasipotential (1; 9; 13). The global quasipotential is related to the invariant measure of the dynamical system when the noise is small: $\lim_{\epsilon\to 0} \epsilon \log p_\infty(\mathbf{x}) = -U(\mathbf{x})$, where $p_\infty(\mathbf{x})$ is the invariant probability distribution of the system.

3. Methods

We construct the quasipotential based on the orthogonal decomposition (5), where the potential and rotational components are parameterized by neural networks. For systems with multiple attractors, we use a single neural network for the potential component and a single neural network for the rotational component across the whole domain of interest. The local quasipotential with respect to each attractor can be obtained by confining the parameterized function to the corresponding basin of attraction.

Once we have a suitable parameterization of V and $\mathbf{g}$, they can then be trained by minimizing a loss function over the trajectory data from the deterministic system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = -\nabla V(\mathbf{x}) + \mathbf{g}(\mathbf{x}). \tag{6}$$

The loss function is designed to reconstruct the dynamics of the original system (6) and to impose the orthogonality condition between the potential and rotational components.

3.1. Parameterization of the Orthogonal Decomposition

The function V is parameterized by the sum of a neural network and a quadratic function,

$$V_\theta(\mathbf{x}) = \hat{V}_\theta(\mathbf{x}) + |\mathbf{x}|^2, \tag{7}$$

where the activation function of the network $\hat{V}_\theta$ is taken as the hyperbolic tangent function. The quadratic term is to make V_θ and $|\nabla V_\theta|$ radially unbounded. The rotational component $\mathbf{g}$ is parameterized by a neural network $\mathbf{g}_\theta$ with continuously differentiable activation (e.g. $\tanh(z)$ or $\mathrm{ReLU}^2(z)$ (26)). Therefore, the parameterized vector field is

$$\mathbf{f}_\theta(\mathbf{x}) = -\nabla V_\theta(\mathbf{x}) + \mathbf{g}_\theta(\mathbf{x}). \tag{8}$$

The neural networks $V_\theta(\mathbf{x})$ and $\mathbf{g}_\theta(\mathbf{x})$ are constructed to obey the following properties:

(i) V_θ is real analytic;

(ii) Both V_θ and $|\nabla V_\theta|$ are radially unbounded, i.e. $V_\theta(\mathbf{x}) \to \infty$ and $|\nabla V_\theta(\mathbf{x})| \to \infty$, as $|\mathbf{x}| \to \infty$;

(iii) $\mathbf{g}_\theta$ is continuously differentiable.

The following theorem shows that under the above three conditions, the set $\{\mathbf{x} \in \mathbb{R}^d : \nabla V_\theta(\mathbf{x}) = 0\}$ is bounded and has Lebesgue measure zero in $\mathbb{R}^d$, and the learned dynamics $\dot{\mathbf{x}} = \mathbf{f}_\theta(\mathbf{x})$ is stable with respect to this set. Hence, any dynamics parameterized as such enjoys good stability properties, and are suitable candidates to model physical systems.

Theorem 2 *Let $\mathbf{f}(\mathbf{x}) = -\nabla V(\mathbf{x}) + \mathbf{g}(\mathbf{x})$ where $\nabla V(\mathbf{x})^T \mathbf{g}(\mathbf{x}) = 0$ and V, $\mathbf{g}$ satisfy the conditions (i),(ii),(iii). Then any trajectory $\{\mathbf{x}(t)\}_{t\geq 0}$ of the system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ approaches the bounded measure-zero set $\mathcal{C} := \{\mathbf{x} \in \mathbb{R}^d : \nabla V(\mathbf{x}) = 0\}$ as $t \to \infty$, i.e.*

$$\lim_{t\to\infty} \inf_{\mathbf{y}\in\mathcal{C}} |\mathbf{x}(t) - \mathbf{y}| = 0. \tag{9}$$

Proof As V is real analytic, all partial derivatives of V are also real analytic. Since V is radially unbounded, the zero sets of these partial derivatives are all measure-zero in $\mathbb{R}^d$. Thus, the set $\mathcal{C}$ has measure of zero in $\mathbb{R}^d$. Furthermore, $|\nabla V(\mathbf{x})| \to \infty$, as $|\mathbf{x}| \to \infty$, which implies that $\mathcal{C}$ is also bounded.

For any trajectory $\{\mathbf{x}(t)\}_{t\geq 0}$ of the system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, we have

$$\frac{dV(\mathbf{x}(t))}{dt} = \nabla V(\mathbf{x}(t)) \cdot \mathbf{f}(\mathbf{x}(t)) = -|\nabla V(\mathbf{x}(t))|^2 \leq 0. \tag{10}$$

Therefore V is the Lyapunov function of this system and we have $V(\mathbf{x}(t)) \leq V(\mathbf{x}(0))$, for all t. Furthermore, since V is radially unbounded, the sub-level set

$$S_0 = \{\mathbf{x} \in \mathbb{R}^d : V(\mathbf{x}) \leq V(x(0))\} \tag{11}$$

is bounded. The trajectory $\{\mathbf{x}(t)\}_{t\geq 0}$ is contained in the bounded set S_0. By Lasalle's theorem (17), the trajectory $\{\mathbf{x}(t)\}_{t\geq 0}$ approaches the set $\mathcal{C}$ as $t \to \infty$. ■

Remark. *Incidentally, the data-driven nature of our method also gives a way to learn stable and interpretable dynamical systems from trajectory data, as shown in Theorem 2. Up to this paper, a large amount of efforts have been devoted to the various data-driven methods for system identification in two main directions. One is to learn closed form equations with some prior knowledge on the underlying mechanism. Related methods include Kronecker product representations (18), sparse identification of nonlinear dynamics (19), Gaussian processes (20) and PDE-net (21). The other direction employs black box methods to learn a model with better accuracy in the prediction. These methods exploit the expressive power of deep neural networks (22; 23), which could potentially learn more complicated models of the nonlinear dynamical systems. However, stability and interpretability is not generally ensured. The attempts to balance expressive power and physical relevance is investigated in (24; 25). The current method falls into this category, in that stability is ensured by construction, and subsequent flexibility is introduced via neural network approximation.*

3.2. Loss Function

Once we have parameterized V_θ and $\mathbf{g}_\theta$, it remains to define a suitable loss function over the data in order to train them to ensure reconstruction ($\mathbf{f} \approx -\nabla V_\theta + \mathbf{g}_\theta$) and orthogonality ($\nabla V_\theta^T \mathbf{g}_\theta \approx 0$).

The observation dataset $X = \{X_i(t_j), X_i(t_j + \Delta t) : \ i = 1, \ldots, N, \ j = 0, \ldots, M\}$ contains N trajectories of the deterministic system (6) where $X_i(t)$ denotes the i^{th} trajectory. Along each trajectory, $2M + 2$ data points are sampled at the times

$$t_0, t_0 + \Delta t, t_1, t_1 + \Delta t, ..., t_M, t_M + \Delta t, \tag{12}$$

where $t_0 < t_1 < ... < t_M$ and Δt is a small time step. The loss function consists of two parts

$$L = L^{dyn} + \lambda L^{orth}, \tag{13}$$

where L^{dyn} is to reconstruct the dynamics in (6), L^{orth} is to impose the orthogonality condition $\nabla V_\theta(\mathbf{x})^T \mathbf{g}_\theta(\mathbf{x}) = 0$, and λ is a parameter.

The term L^{dyn} depends on the difference between the learned dynamics and the observed trajectories,

$$\begin{aligned} L^{dyn} &= \frac{1}{N(M+1)} \sum_{i=1}^{N} \sum_{j=0}^{M} \bar{h}\left(\mathbf{e}_{ij}; \delta_1\right), \\ \mathbf{e}_{ij} &= \frac{1}{\Delta t}\left(\mathcal{I}_{\Delta t}[\mathbf{f}_\theta; X_i(t_j)] - X_i(t_j + \Delta t)\right), \end{aligned} \tag{14}$$

where $\mathcal{I}_{\Delta t}[\mathbf{f}_\theta; X_i(t_j)]$ is the state obtained by performing the numerical integration of the learned dynamics $\dot{\mathbf{x}} = \mathbf{f}_\theta(\mathbf{x})$ by one time step Δt from the state $X_i(t_j)$, and $\bar{h}(\mathbf{e}; \delta_1)$ denotes the mean Huber loss of the vector $\mathbf{e} = (e_1, \ldots, e_d)$ with threshold δ_1,

$$\begin{aligned} \bar{h}(\mathbf{e}; \delta_1) &= \frac{1}{d} \sum_{i=1}^{d} h(e_i; \delta_1), \\ h(e_i; \delta_1) &= \begin{cases} \frac{1}{2} e_i^2, & |e_i| < \delta_1, \\ \delta_1 |e_i| - \frac{1}{2}\delta_1^2, & \text{otherwise.} \end{cases} \end{aligned} \tag{15}$$

The Huber loss reduces the dominating effect of large components in the vector $\mathbf{e}$.

The orthogonality between ∇V_θ and $\mathbf{g}_\theta$ is imposed by the penalty term λL^{orth} with

$$L^{orth} = \frac{1}{S} \sum_{i=1}^{S} w\left(\frac{\nabla V_\theta(\tilde{X}_i)^T \mathbf{g}_\theta(\tilde{X}_i)}{|\nabla V_\theta(\tilde{X}_i)| \cdot |\mathbf{g}_\theta(\tilde{X}_i)|}; \delta_2\right), \tag{16}$$

where $w(y; \delta_2) = y^2 I_{y>0} + \delta_2 y^2 I_{y<0}$, δ_2 is a parameter and $\tilde{X}_1, \ldots, \tilde{X}_S$ are representative data points sampled from X by using Algorithm 1. The representative data points are chosen such that each of them covers a ball of radius r and no other representative data points lie inside this ball. Because the data points in the sampled trajectories are clumped together near the attractors and not uniformly distributed in the region of interest, the loss term L^{orth} is dominated by the data points near the attractors and the orthogonality condition is only effectively imposed near the attractors if all data points are used. In contrast, the representative data points sampled using Algorithm 1 are roughly uniformly distributed in the region where the sampled trajectories visit; as a result, the orthogonality condition can be effectively imposed in this region.

Algorithm 1 Sampling the representative dataset

```
function GETXHAT(X, r)
    Initialize the sets Y = X and X̃ = ∅
    while Y ≠ ∅ do
        Randomly select x ∈ Y and append x to the set X̃
        Delete all the points belonging to the ball B_r(x) from Y
    end while
    Return X̃
end function
```

4. Numerical Examples

We now illustrate using various numerical examples that the proposed method can efficiently compute the quasipotential and at the same time learn stable dynamics. Section 4.1 contains two ODE systems: one with two stable equilibrium points and the other with a limit cycle. The quasipotentials are known in these two examples, and we use these exact solutions to benchmark the numerical method. Section 4.2 is a biological system which models the reproduction process of a budding yeast cell cycle. Section 4.3 contains two high-dimensional systems which are obtained from the discretization of partial differential equations (PDEs).

In the examples, we generate trajectories by simulating the deterministic dynamics in Eq. (6) using the forth-order Runge-Kutta method with the time step Δt on the time interval $[0, T]$. The initial states are randomly sampled from certain distributions which will be specified in the examples. From these trajectories, we obtain the dataset X by collecting the data points at the times $t_j = jm\Delta t$ and $t_j + \Delta t$ where $j = 0, 1, \ldots, M$ and m is some positive integer. The set of trajectories is split into three parts: 70% (training), 20% (validation) and 10% (test). The representative datasets are sampled from these three datasets respectively using Algorithm 1 with various choices of the parameter r. The parameters Δt, T, m, r and the number of trajectories N are given in Table 1.

Table 1: Parameters in the numerical examples.

Example	N	Δt	T	m	r	δ_1	λ	# nodes in each hidden layer
1	2×10^3	10^{-2}	5	10	0.1	1	1	50
2	2×10^3	10^{-2}	5	10	0.05	1	0.02	50
3	1×10^4	10^{-2}	50	100	0.1	1	0.005	100
4	1×10^4	10^{-3}	2	20	0.2	1	1	100
5	5×10^4	10^{-4}	2	200	0.2	1	0.1	200

The networks $\hat{V}_\theta$, $\mathbf{g}_\theta$ for the potential and rotational components in the parameterized vector field (8) are both taken as fully connected neural networks of 2 hidden layers with the same number of nodes in each hidden layer. The nonlinear activation function in $\hat{V}_\theta$ is $\tanh$ in all the examples, and the activation function in $\mathbf{g}_\theta$ is $\tanh$ in Examples 1-3 and ReLU^2 in Examples 4-5. The input to the parameterized vector field $\mathbf{f}_\theta$ is centered so that the centered data points have mean-zero.

In the loss function, we use the second-order Runge-Kutta method as the numerical integrator $\mathcal{I}$ and set $\delta_2 = \frac{1}{10}$. The two parameters δ_1, λ are chosen so that the orthogonality error L^{orth} and the error of the predicted long-term dynamics over the test dataset are both small. To quantify the accuracy of the predicted long-term dynamics, we solve the learned dynamics

$$\dot{\mathbf{x}}_\theta = -\nabla V_\theta(\mathbf{x}_\theta) + \mathbf{g}_\theta(\mathbf{x}_\theta) \tag{17}$$

using the second-order Runge-Kutta method on the time interval $[0, T]$, and compare the solution with the original dynamics:

$$\epsilon = \frac{\sqrt{\sum_{j=1}^{M} |\mathbf{x}_\theta(t_j) - \mathbf{x}(t_j)|^2}}{\sqrt{\sum_{j=1}^{M} |\mathbf{x}(t_j)|^2}}, \tag{18}$$

where $\mathbf{x}(t)$ is the trajectory from the test dataset with the same initial sate as $\mathbf{x}_\theta(t)$.

The networks are trained with Adam optimizer (27) using mini-batches of size 5000, while the learning rate exponentially decays over the training steps.

4.1. ODE systems with known quasipotentials

First, we consider two low-dimensional systems: one with two stable equilibrium points and the other with a limit cycle. The quasipotentials are known in these two examples, and we use these exact quasipotentials to benchmark the proposed method.

Example 1. We consider the following system in three-dimensional space (16),

$$\begin{aligned} \frac{dx}{dt} &= -2(x^3 - x) - (y + z), \\ \frac{dy}{dt} &= -y + 2(x^3 - x), \\ \frac{dz}{dt} &= -z + 2(x^3 - x), \end{aligned} \tag{19}$$

where the state of the system is $\mathbf{x} = (x, y, z)^T$. This system has two stable equilibrium points, one at $\mathbf{x}_a = (-1, 0, 0)$ and the other at $\mathbf{x}_b = (1, 0, 0)$ and one unstable equilibrium point at $\mathbf{x}_c = (0, 0, 0)$. In the basins of the two stable equilibrium points, the quasipotential is known and given by

$$U(x, y, z) = (1 - x^2)^2 + y^2 + z^2. \tag{20}$$

We generate 2000 trajectories by solving the equations in (19) starting from initial states sampled from the uniform distribution on the domain $\mathcal{D} = [-2, 2] \times [-1.5, 1.5]^2$. Along each trajectory, we collect 100 data points. In total, X contains 2×10^5 data points. Out of these data points, 8571 representative data points are used to impose the orthogonality condition.

The test dataset contains 200 trajectories. To quantify the accuracy of the predicted long-term dynamics, we solve the learned dynamics starting from the initial states of these trajectories and compute the error in (18) for each trajectory. Fig. 1 (lower panel) shows the comparison of three trajectories of the learned dynamics with those of the original dynamics in the test dataset. These errors have the mean 5.069×10^{-4} and the standard deviation 1.565×10^{-3}.

The learned quasipotential is given by $U_\theta(\mathbf{x}) = 2V_\theta(\mathbf{x}) - C$, where the constant C is such that the minimum of $U_\theta(\mathbf{x})$ on the domain $\mathcal{D}$ equals zero. Fig. 1 (upper panel) shows the comparison of $U_\theta(\mathbf{x})$ with the exact quasipotential in (20). To quantify the accuracy of the learned quasipotential, we compute the relative root mean square error (rRMSE) and the relative mean absolute error (rMAE),

$$\text{rRMSE} = \frac{\sqrt{\sum_{i=1}^{L} (U(\mathbf{x}_i) - U_\theta(\mathbf{x}_i))^2}}{\sqrt{\sum_{i=1}^{L} U^2(\mathbf{x}_i)}}, \quad \text{rMAE} = \frac{\sum_{i=1}^{L} |U(\mathbf{x}_i) - U_\theta(\mathbf{x}_i)|}{\sum_{i=1}^{L} |U(\mathbf{x}_i)|}, \tag{21}$$

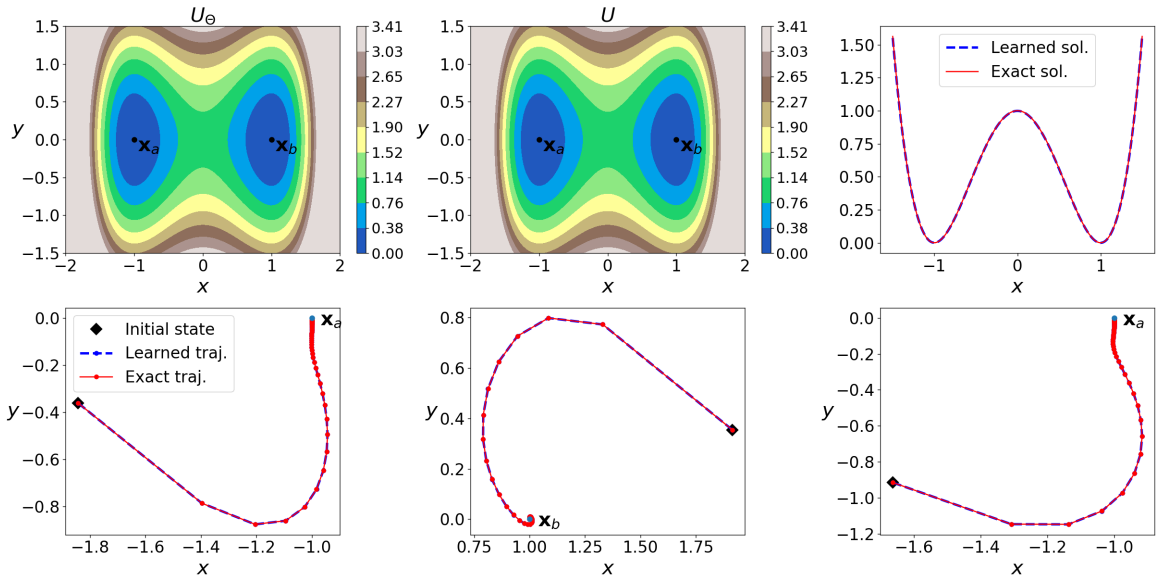

Figure 1 (Example 1): *Upper Panel*: Contour plots of the learned quasipotential U_θ (left) and exact quasipotential U (middle) projected onto the xy plane with $z = 0$, and plot of the learned quasipotential along the line $y = z = 0$ (right). *Lower Panel*: Comparison of trajectories of the learned dynamics and the original dynamics (19) from different initial states.

where $\{\mathbf{x}_i\}_{i=1}^L$ are the grid points of the uniform mesh on $\mathcal{D}$. The rRMSE and rMAE for the learned quasipotential are 0.0037 and 0.0017, respectively. The errors are computed with $L = 10^6$.

The numerical result is insensitive to the choice of r as long as r is not too small or too large. If r is too small, the orthogonality condition is only effectively imposed in small regions near the attractors because most of the data points are concentrated in those regions; on the other hand, if r is too large, then only a very small number of data points are used and the orthogonality condition is not effectively imposed either. In this example, we implemented the numerical method with r ranging from 0.1 to 0.5 and obtained similar results. The results are provided in the Appendix.

Example 2. We consider the system with the quasipotential

$$U(x,y) = \left((x-a)^2 + (x-a)(y-b) + (y-b)^2 - \frac{1}{2}\right)^2, \quad (x,y) \in R^2, \tag{22}$$

where a, b are two parameters. The function U attains its local maximum at the point (a,b) and attains its minimum on the ellipse

$$\left\{(x,y) \in R^2 : (x-a)^2 + (x-a)(y-b) + (y-b)^2 = \frac{1}{2}\right\}. \tag{23}$$

The dynamics for the system is governed by

$$\begin{aligned} \frac{dx}{dt} &= -\frac{1}{2}\frac{\partial U}{\partial x}(x,y) - 2\,(x + 2y - a - 2b)\,, \\ \frac{dy}{dt} &= -\frac{1}{2}\frac{\partial U}{\partial y}(x,y) + 2\,(2x + y - 2a - b)\,, \end{aligned} \tag{24}$$

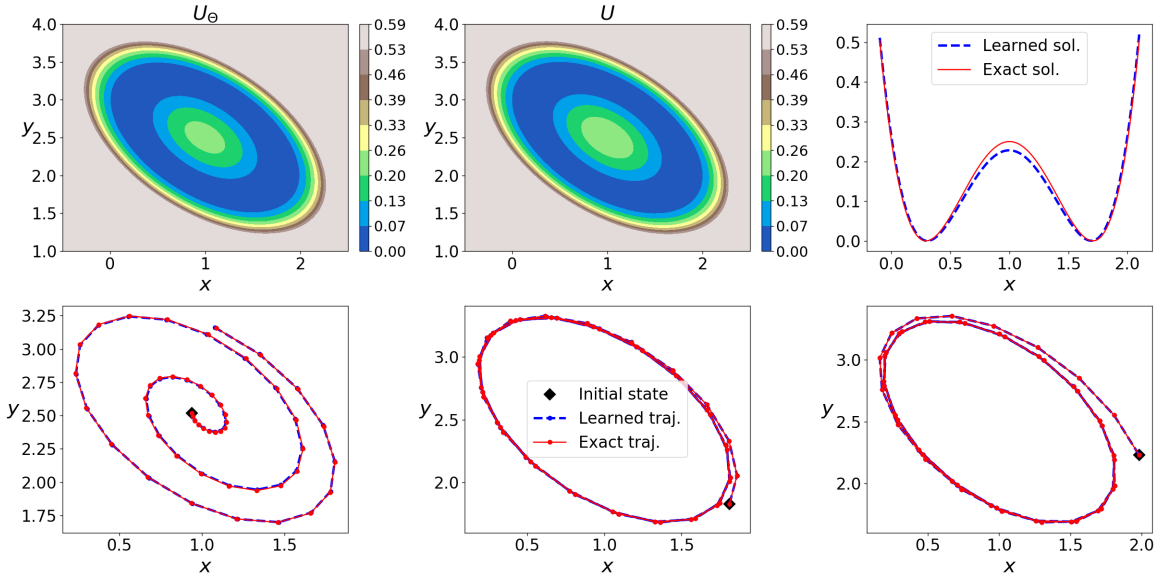

Figure 2 (Example 2): *Upper Panel*: Contour plots of the learned quasipotential U_θ (left) and exact quasipotential U (middle) and plot of the learned quasipotential along the line $y = b$ (right). *Lower Panel*: Comparison of trajectories of the learned dynamics and the original dynamics (24) from different initial states.

where the state of the system is $\mathbf{x} = (x, y)^T$. This dynamical system has a stable limit cycle on the ellipse in (23) and an unstable equilibrium point at (a, b) inside the limit cycle.

We take $a = 1$, $b = 2.5$ and generate 2000 trajectories by solving the equations in (24) starting from initial states sampled from the uniform distribution on the domain $\mathcal{D} = [-0.5, 2.5] \times [1, 4]$. Along each trajectory, we collect 100 data points. In total, X contains 2×10^5 data points. Out of these data points, 3712 representative data points are used to impose the orthogonality condition.

Fig. 2 (lower panel) shows a comparison of three trajectories of the learned dynamics with that of the original dynamics in the test dataset. The statistics (mean $\pm$ deviation) of the errors of 200 trajectories is $4.797 \times 10^{-4} \pm 2.923 \times 10^{-4}$. A comparison of the learned quasipotential with the exact quasipotential in (22) is shown in Fig. 2 (upper panel). The rRMSE and rMAE for the learned quasipotential on the domain $\mathcal{D}$ are 0.0141 and 0.0090, respectively. The errors are computed using Eqs. (21) with $L = 10^4$.

4.2. Biological system: budding yeast cell cycle

The previous two examples are toy problems where the exact quasipotential is known. Now, we test our method on a more challenging problem where computing quasipotential landscapes using traditional methods may be very expensive.

Example 3. We study the robustness of the reproduction process of a budding yeast cell cycle by constructing the quasipotential (10). The simplified network of yeast cell is composed of three modules: the $G1/S$ module, the early M module and the late M module. Based on the feedback of each module and the interactions between different modules, the following dynamics has been

proposed for the cell cycle

$$\begin{aligned}\frac{dx}{dt} &= \frac{x^2}{j_1^2+x^2} - k_1 x - xy + a_0,\\ \frac{dy}{dt} &= \frac{y^2}{j_2^2+y^2} - k_2 y - yz + k_{a1} x,\\ \frac{dz}{dt} &= \frac{k_s z^2}{j_3^2+z^2} - k_3 z - k_i zx + k_{a2} y,\end{aligned} \tag{25}$$

where x, y, z represent the concentration of certain key regulators in the $G1/S$, early M and late $M/G1$ phase, respectively. The values for the parameters j_1,j_2,j_3,k_1,k_2,k_3,k_i,k_s,k_{a1},k_{a2},a_0 are taken from Ref. (10). The dynamics has a stable equilibrium state $G1$ approximately at $(0, 0, z_{max})$ where $z_{max} = 4.342$. The yeast cell cycle is termed a *robust process* in (10), in the sense that most transition paths stay close to a particular pathway due to the dynamical landscape. This pathway starts from the excited $G1$ state and ends at the stable $G1$ state by going through the S phase approximately at $(x_{max}, 0, 0)$ where $x_{max} = 4.335$ and the early M state approximately at $(0, y_{max}, 0)$ where $y_{max} = 4.353$.

We generate 10^4 trajectories by solving the equations in (25) starting from initial states sampled from the uniform distribution on the set

$$\{\mathbf{x} = (x, y, z) \in [0, 5]^3 : \|\mathbf{f}(\mathbf{x})\|_\infty < 5\}, \tag{26}$$

where the notation $\|\mathbf{y}\|_\infty$ denotes the maximum of absolute values of the components in the vector $\mathbf{y}$. The last condition excludes states far away from the regions of interest corresponding to the transition events. Along each trajectory, we collect 100 data points. In total, X contains 10^6 data points. Out of these data points, 8384 representative data points are used to impose the orthogonality condition.

Fig. 3 (lower panel) shows a comparison of one trajectory of the learned dynamics and that of the original dynamics in the test dataset. The statistics (mean $\pm$ deviation) of the errors of the 1000 trajectories is 0.161 ± 0.226. The cross-sections of the learned quasipotential at $z = 0$ and $x = 0$ are shown in Fig. 3 (upper panel). The quasipotential characterizes the robust process of the cell cycle, which agrees well with the result in Ref. (10) using the geometric minimum action method. Moreover, notice that the quasipotential we compute can be evaluated at arbitrary points in space (in the regions explored by the sampled data) and is not limited by any meshes, or choice of beginning and end points for path-based methods.

4.3. High-dimensional systems: discretized PDEs

We next apply the proposed method to two high-dimensional systems which are obtained from the discretization of PDEs. After discretization, the first system is a gradient system in the 50-dimensional space with known quasipotential, and the second one is a non-gradient system in the 40-dimensional space.

Example 4. We consider the Ginzburg-Landau equation

$$u_t = \delta u_{xx} - \delta^{-1} V'(u), \quad x \in [0, 1], \tag{27}$$

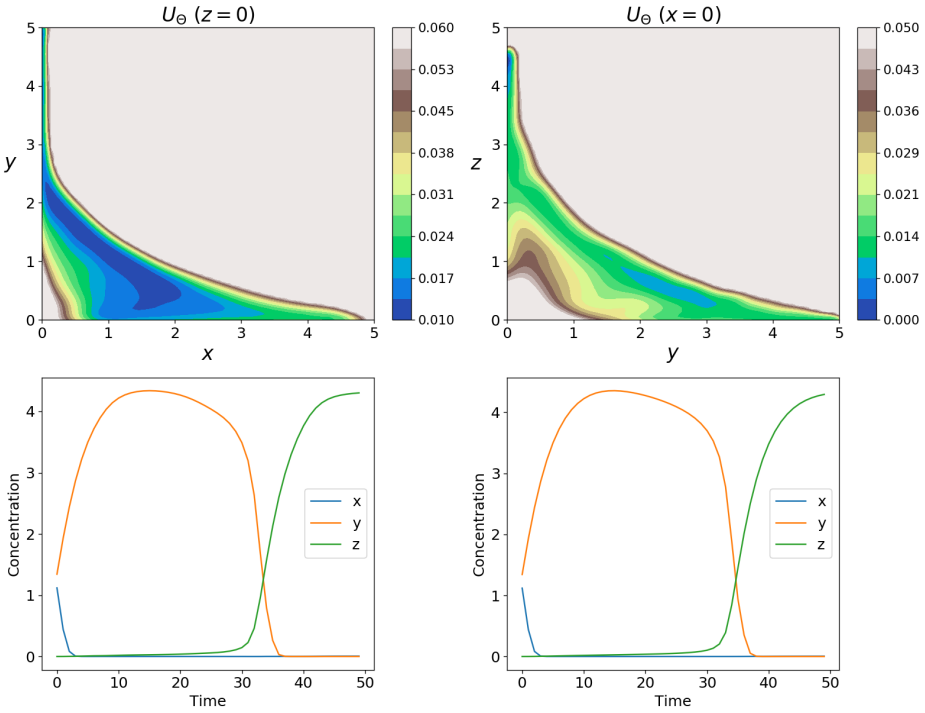

Figure 3 (Example 3): *Upper Panel*: Contour plots of the quasipotential projected onto the xy-plane with $z = 0$ (left) and the yz-plane with $x = 0$ (right). *Lower Panel*: Comparison of one trajectory of the learned dynamics (left) and the original dynamics in (25) (right).

with the boundary conditions $u(0,t) = u(1,t) = 0$ and the initial condition $u(x,0) = u^0(x)$, where $V(u) = \frac{1}{4}(1-u^2)^2$ is the double-well potential and δ is a small parameter. The equation is a gradient flow associated with the energy

$$E[u] = \int_0^1 \left(\frac{1}{2}\delta u_x^2 + \delta^{-1} V(u) \right) dx. \tag{28}$$

We partition the interval $[0,1]$ using $I+1$ grid points x_0,...,x_I, where $x_i = ih$ and $h = 1/I$. Then we approximate the spatial derivatives in Eq. (27) using the central finite difference and obtain the following system of ODEs

$$\frac{du_i}{dt} = \delta \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} - \delta^{-1} V'(u_i), \quad 1 \le i \le I-1, \tag{29}$$

with $u_0 = u_I = 0$ and the initial condition $u_i(0) = u^0(x_i)$ for $1 \le i \le I-1$, where u_i denotes the approximate solution at the grid point x_i. The state of the system is denoted by $\mathbf{u} = (u_1, \ldots, u_{I-1})$. The ODE system is a gradient flow associated with the energy

$$E_h[\mathbf{u}] = \sum_{i=1}^{I} \frac{1}{2}\delta \left(\frac{u_i - u_{i-1}}{h} \right)^2 + \delta^{-1} V(u_i), \tag{30}$$

which is a discretization of the energy (28), up to the factor h. The dynamics (29) has two stable states at the two local minima $\mathbf{u}_\pm$ of the energy (30), which are shown in Fig. 4 (last column) for

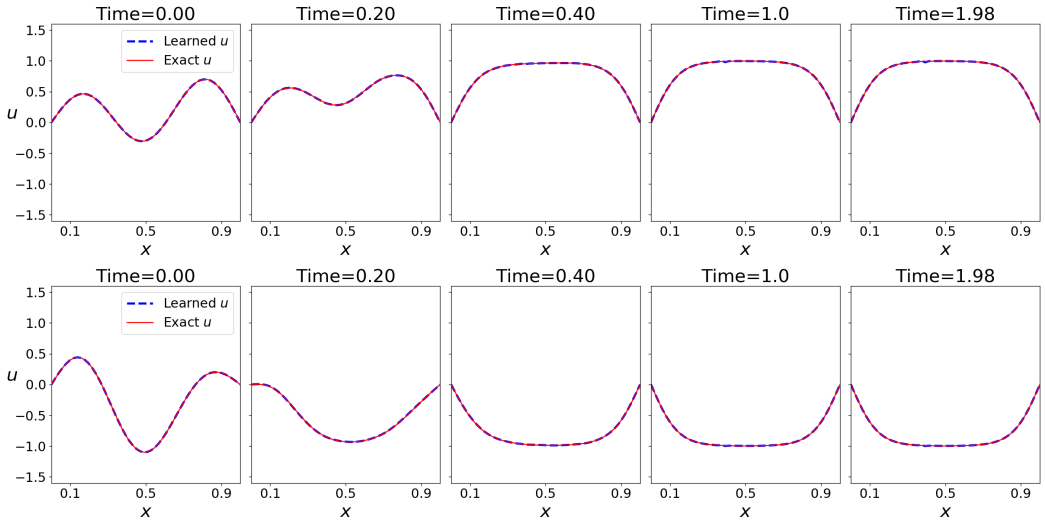

Figure 4 (Example 4): Comparison of trajectories of the learned dynamics and the original dynamics (29) from different initial states.

$\delta = 0.1$. The quasipotential with respect to the two stable states is

$$U(\mathbf{u}) = 2E_h[\mathbf{u}] + C \tag{31}$$

in the basins of attraction, where C is constant.

The number of discretization points is taken as $I = 51$. We generate 10^4 trajectories by solving the dynamics in (29) starting from the initial states:

$$u^0(x) = \frac{a \cdot \tilde{u}(x)}{\max_y |\tilde{u}(y)|}, \tag{32}$$

where $\tilde{u}(x) = \sum_{k=1}^{4} \hat{u}_k \sin(k\pi x)$ and $\{\hat{u}_k\}_{k=1}^{4}$, a are drawn from the uniform distributions: $u_k \sim \mathcal{U}(-1, 1)$, $a \sim \mathcal{U}\left(0, \frac{3}{2}\right)$. Along each trajectory, we collect 200 data points. In total, X contains 2×10^6 data points. Out of these data points, 76044 representative data points are used to impose the orthogonality condition.

Fig. 4 shows a comparison of two trajectories of the learned dynamics and those of the original dynamics in the test dataset. The statistics (mean $\pm$ deviation) of the errors of the 1000 trajectories is $1.220 \times 10^{-2} \pm 7.734 \times 10^{-2}$. To assess the accuracy of the learned quasipotential, we compare U_θ and U in (31) along the minimum energy path (MEP) from $\mathbf{u}_-$ to $\mathbf{u}_+$. The MEP is computed using the string method (3). The comparison is shown in Fig. 5, from which a good agreement can be observed. In particular, the learned quasipotential accurately captures the energy barrier between the two stable states. Also, the magnitudes of the learned potential and rotational components are quantified. The rotational component $\mathbf{g}_\theta$ is small and its l_2-norm is on the order of 10^{-2} on the set of sampled representative data points; in contrast, the mean of the l_2-norm $\|\nabla V_\theta\|_2$ is on the order of 10 on this set. This demonstrates that the proposed method is capable of identifying the gradient nature of the dynamics.

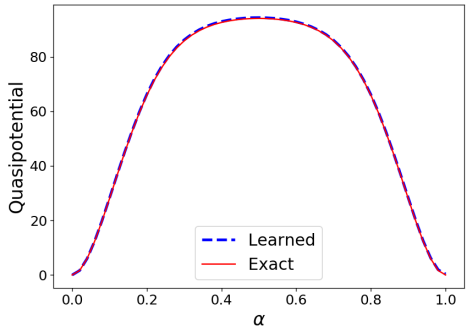

Figure 5 (Example 4): Comparison of the learned and exact quasipotentials for the discretized Ginzburg-Landau equation along the MEP, where α is the normalized arc-length parameter along the MEP.

Example 5. We consider the dynamics of the Brusselator on the spatial interval $[0, 1]$,

$$\begin{aligned} u_t &= \frac{1}{\alpha}\left(u_{xx} + 1 + u^2 v - (1+A)u\right), \\ v_t &= \beta v_{xx} + Au - u^2 v, \end{aligned} \tag{33}$$

with the Neumann boundary conditions $u_x(0,t) = u_x(1,t) = 0$, $v_x(0,t) = v_x(1,t) = 0$, and the initial condition $u(x,0) = u^0(x)$, $v(x,0) = v^0(x)$, where α, β and A are parameters. We discretize the interval $[0,1]$ with grid points $x_0, \ldots, x_I$, where $x_i = ih$ and $h = 1/I$. Then we approximate the spatial derivatives in Eq. (33) using the central finite difference and obtain the following system of ODEs

$$\begin{aligned} \frac{du_i}{dt} &= \frac{1}{\alpha}\left(\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} + 1 + u_i^2 v_i - (1+A)u_i\right), \\ \frac{dv_i}{dt} &= \beta\frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} + Au_i - u_i^2 v_i, \end{aligned} \tag{34}$$

for $0 \le i \le I$, with the Neumann boundary conditions imposed by $u_{-1} = u_1$, $u_{I+1} = u_{I-1}$, $v_{-1} = v_1$, $v_{I+1} = v_{I-1}$, and the initial condition $u_i(0) = u^0(x_i)$, $v_i(0) = v^0(x_i)$ for $0 \le i \le I$, where (u_i, v_i) denotes the solution of Eq. (33) at x_i. The state of the system is denoted by $\mathbf{x} = (u_0, ..., u_I, v_0, ..., v_I)$. The dynamics has a stable state: $u_i = 1$, $v_i = A$ for $0 \le i \le I$.

We consider the system with $\beta = 1$ and $\beta = 0.01$, respectively, and take $\alpha = 0.1$ and $A = 0.5$. The number of discretization points is taken as $I = 19$, so the discretized system is in the 40-dimensional space. We generate 5×10^4 trajectories by solving the dynamics in (34) starting from the initial states:

$$u^0(x) = a_1 + \frac{a_2 \cdot \tilde{u}(x)}{\max_y |\tilde{u}(y)|}, \quad v^0(x) = a_3 + \frac{a_4 \cdot \tilde{v}(x)}{\max_y |\tilde{v}(y)|}, \tag{35}$$

where $\tilde{u}(x) = \sum_{k=1}^4 \hat{u}_k \cos(k\pi x)$, $\tilde{v}(x) = \sum_{k=1}^4 \hat{v}_k \cos(k\pi x)$ and $\{\hat{u}_k\}_{k=1}^4$, $\{\hat{v}_k\}_{k=1}^4$, a_1, a_2, a_3, a_4 are drawn from the uniform distributions

$$\begin{aligned} &\hat{u}_k \sim \mathcal{U}(-1,1),\ \hat{v}_k \sim \mathcal{U}(-1,1),\ k = 1, \ldots, 4, \\ &a_1 \sim \mathcal{U}(0,2),\ a_2 \sim \mathcal{U}(0, 1 - |a_1 - 1|),\ a_3 \sim \mathcal{U}(0,1),\ a_4 \sim \mathcal{U}\left(0, \frac{1}{2} - \left|a_3 - \frac{1}{2}\right|\right). \end{aligned} \tag{36}$$

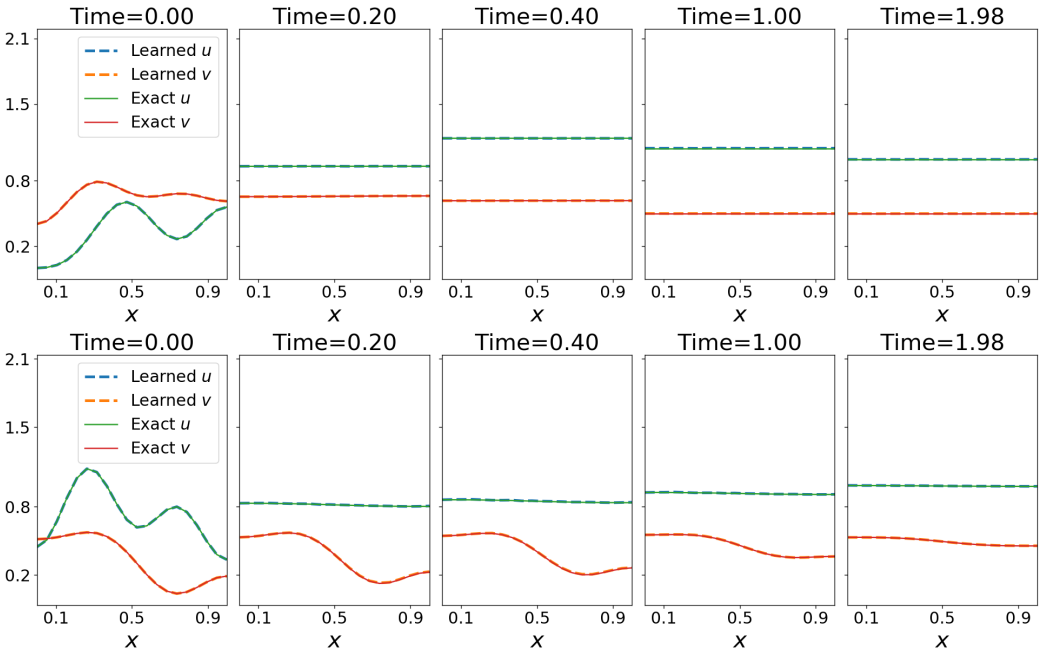

Figure 6 (Example 5): Comparison of one trajectory of the learned dynamics and the original dynamics in (34). The parameter β is $\beta = 1$ (*Upper Panel*) and $\beta = 0.01$ (*Lower Panel*).

Along each trajectory, we collect 200 data points. In total, X contains 10^7 data points. Out of these data points, 57239 and 140354 representative data points are used to impose the orthogonality condition for the system with $\beta = 1$ and $\beta = 0.01$, respectively.

Fig. 6 shows a comparison of one trajectory of the learned dynamics and that of the original dynamics in the test dataset. The statistics (mean $\pm$ deviation) of the errors of 5000 trajectories is $1.036 \times 10^{-3} \pm 4.139 \times 10^{-4}$ (for $\beta = 1$) and $1.944 \times 10^{-3} \pm 7.053 \times 10^{-4}$ (for $\beta = 0.01$). The quasipotential is shown in Fig. 7 as a function of $(\hat{u}_0, \hat{v}_0)$, where $(\hat{u}_0, \hat{v}_0)$ corresponds to the states $u(x) \equiv \hat{u}_0$, $v(x) \equiv \hat{v}_0$ (left), and as a function of $(\hat{u}_1, \hat{v}_1)$, where $(\hat{u}_1, \hat{v}_1)$ corresponds to the states $u(x) = 1 + \hat{u}_1 \cos(\pi x)$, $v(x) = 0.5 + \hat{v}_1 \cos(\pi x)$ (right). The numerical results for the system with $\beta = 1$ agree well with those computed using the minimum action method (2).

5. Conclusion

In this paper, we proposed a method for computing the quasipotential for dynamical systems and at the same time learning the dynamics from the trajectory data. This method is based on learning an orthogonal decomposition of the force field into potential and rotational components, each parameterized by a neural network. The neural networks are trained by minimizing a loss function composed of two parts: one is to reconstruct the dynamics and the other one is to impose the orthogonality condition between the potential and rotational components. The quasipotential associated with each attractor of the dynamical system can be obtained by confining the potential component to the corresponding basin of attraction. We successfully applied the method to various examples including systems with stable equilibrium points, limit cycles and systems in high dimensions. The method is purely data driven in the sense that no explicit form of the dynamical system is required;

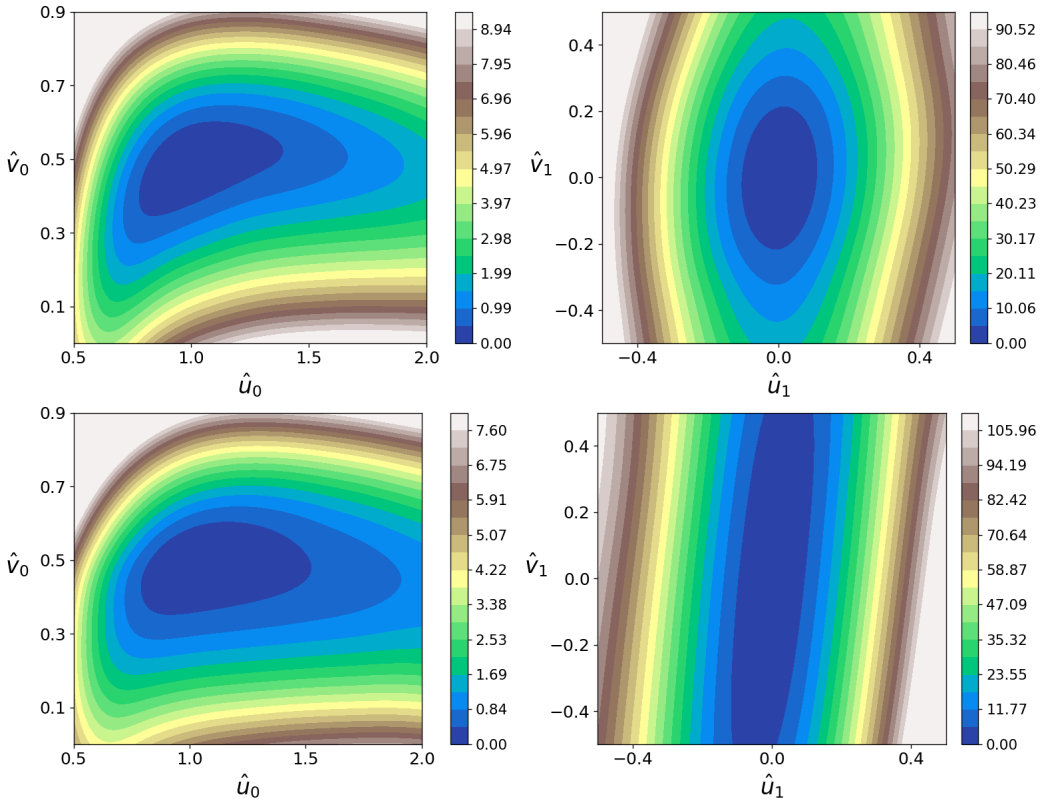

Figure 7 (Example 5): Contour plots of the quasipotential as a function of $(\hat{u}_0, \hat{v}_0)$ corresponding to the states $u(x) \equiv \hat{u}_0$, $v(x) \equiv \hat{v}_0$ (left), and as a function of $(\hat{u}_1, \hat{v}_1)$ corresponding to the states $u(x) = 1 + \hat{u}_1 \cos(\pi x)$, $v(x) = 0.5 + \hat{v}_1 \cos(\pi x)$ (right). The parameter β is $\beta = 1$ (*Upper Panel*) and $\beta = 0.01$ (*Lower Panel*).

in fact, an explicit model for the dynamics is learned from the observed trajectories in this method. To the best of our knowledge, this is the first efficient and accurate method that can be used to map the landscape of the quasipotential in high dimensions.

After we obtain the quasipotential, we can compute other interesting objects associated with the dynamical system perturbed by small noise. For example, we can identify the minimum action path between the attractor A and another state. Using the fact that the tangent of the path is parallel to $\mathbf{f} + \nabla U_A$ along the minimum action path, the path can be computed using the string method. The expected exit time from the basin of attraction can also be estimated using the minimum value of the quasipotential on the boundary of the basin of attraction.

In the current work, we demonstrated the effectiveness of the proposed method using examples with different features. In the future, we plan to apply the method to problems of practical interest such as dynamical systems in fluid mechanics and biological systems. Furthermore, as the effective dimension for many practical high-dimensional systems is low, machine learning tools for dimension reduction such as autoencoder can be incorporated into the parameterization of the force field to discover the effective variables of the dynamics. This will be left to our future work.

Acknowledgments

The work of Ren was supported in part by Singapore MOE AcRF grant R-146-000-267-114, and the NSFC grant (No. 11871365). QL is supported by the National Research Foundation, Singapore, under the NRF fellowship (NRF-NRFF13-2021-0005).

References

[1] M. I. Freidlin and A. D. Wentzell, "Random Perturbations of Dynamical Systems," 3rd Ed, Springer Press (2012).

[2] W. E, W. Ren, and E. Vanden-Eijnden, "Minimum action method for the study of rare events," Commun. Pure Appl. Math. **57**, 637–656 (2004).

[3] W. E, W. Ren, and E. Vanden-Eijnden, "Simplified and improved string method for computing the minimum energy paths in barrier-crossing events," J. Chem. Phys. **126**, 164103 (2007).

[4] X. Zhou, W. Ren, and W. E, "Adaptive minimum action method for the study of rare events," J. Chem. Phys. **128**, 104111 (2008).

[5] M. Heymann and E. Vanden-Eijnden, "The geometric minimum action method: A least action principle on the space of curves," Commun. Pure Appl. Math. **61**, 1052-1117 (2008).

[6] M. K, Cameron, "Finding the quasipotential for nongradient SDEs," Physica D **241**, 1532-1550 (2012).

[7] D. Dahiya and M. Cameron, "Ordered line integral methods for computing the quasi-potential," J. Scientific Computing **75**, 1351-1384 (2018).

[8] D. Dahiya and M. Cameron, "An ordered line integral method for computing the quasi-potential in the case of variable anisotropic diffusion," Physica D **382**, 33-45 (2018).

[9] P. Zhou and T. Li, "Construction of the landscape for multi-stable systems: Potential landscape, quasipotential, A-type integral and beyond," J. Chem. Phys. **144**, 094109 (2016).

[10] C. Lv, X. Li, F. Li, and T. Li, "Energy landscape reveals that the budding yeast cell cycle is a robust and adaptive multi-stage process," PLoS Comput. Biol. **11**, e1004156 (2015).

[11] C. Li and G. Balazsi, "A landscape view on the interplay between EMT and cancer metastasis," NPJ Syst. Biol. Appl. **4**, 1-9 (2018).

[12] J. Wang, C. Li, and E. Wang, "Potential and flux landscapes quantify the stability and robustness of budding yeast cell cycle network," Proc. Natl. Acad. Sci. **107**, 8195-8200 (2010).

[13] F. Bouchet, K. Gawedzki, and C. Nardini, "Perturbative calculation of quasi-potential in non-equilibrium diffusions: a mean-field example," J. Stat. Phys. **163**, 1157-1210 (2016).

[14] B. C. Nolting and K. C. Abbott, "Balls, cups, and quasi-potentials: quantifying stability in stochastic systems," Ecology **97**, 850-864 (2016).

[15] C. Lv, X. Li, F. Li, and T. Li, "Constructing the energy landscape for genetic switching system driven by intrinsic noise," PLoS One **9**, e88167 (2014).

[16] S. Yang, F. P. Samuel, and K. C. Maria, "Computing the quasipotential for nongradient SDEs in 3D," J. Comput. Phys. **379**, 325-350 (2019).

[17] J. LaSalle, "Some extensions of Liapunov's second method," IRE Trans. Circuit Theory **7**, 520-527 (1960).

[18] C. Yao and E. M. Bollt, "Modeling and nonlinear parameter estimation with Kronecker product representation for coupled oscillators and spatiotemporal systems," Physica D **227**, 78-99 (2007).

[19] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," Proc. Natl. Acad. Sci. **113**, 3932-3937 (2016).

[20] M. Raissi and G. E. Karniadakis, "Hidden physics models: Machine learning of nonlinear partial differential equations," J. Comput. Phys. **357**, 125-141 (2018).

[21] Z. Long, Y. Lu, and B. Dong, "PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network," J. Comput. Phys. **399**, 108925 (2019).

[22] M. Raissi, and P. Perdikaris, and G. E. Karniadakis, "Multistep neural networks for data-driven discovery of nonlinear dynamical systems," arXiv preprint arXiv:1801.01236 (2018).

[23] M. Raissi, "Deep hidden physics models: Deep learning of nonlinear partial differential equations," J. Mach. Learn. Res. **19**, 932-955 (2018).

[24] H. Yu, X. Tian, and Q. Li, "OnsagerNet: Learning Stable and Interpretable Dynamics using a Generalized Onsager Principle," arXiv preprint arXiv:2009.02327 (2020).

[25] G. Manek and J. Z. Kolter, "Learning stable deep dynamics models," arXiv preprint arXiv:2001.06116 (2020).

[26] B. Li, S. Tang, and H. Yu, "Better approximations of high dimensional smooth functions by deep neural networks with rectified power units," arXiv preprint arXiv:1903.05858 (2019).

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proceedings of International Conference on Learning Representations (ICLR), San Diego (2015).

Appendix: Sensitivity of numerical results to the choice of r in Algorithm 1

Only a set of representative data points are used to impose the orthogonality condition in the numerical method. These data points are sampled using non-overlapping balls with radius r. To investigate the effect of the choice of r on the numerical results, we carried out simulations with different r in the range of $[0, 0.5]$ in *Example 1*. The scatter plots of sampled representative data points projected onto the xy-plane for different values of r are shown in Fig. 8. All parameters and the neural network architecture, except λ (the coefficient of the orthogonality term L^{orth} in the loss function), are the same as those specified in *Example 1*. As mentioned earlier, the parameter λ is tuned so that the orthogonality error L^{orth} and the error of the predicted long-term dynamics over the test dataset are both small. The number of sampled representative data points and relative errors of the learned quasipotential with different choice of r are given in Table 2.

From the table, we see that very small r can lead to poor performance of the trained model. This is because when r is very small, a majority of the sampled representative data points lie in small regions near the attractors, and as a result, the loss term L^{orth} is dominated by those data points and the orthogonality condition is not well-imposed in regions away from the attractors. When r becomes larger, the sampled representative data points are nearly uniformly distributed. In this example, it is observed from the table that the quasipotential is well learned and the choice of r in the range $[0.1, 0.5]$ has little effect on the numerical results. Our experience in the other examples also confirmed this observation. So in practice, it is not difficult to choose an appropriate value for the parameter r.

Table 2: Numerical results for the errors with different choice of r.

r	# representative data points	rRMSE	rMAE
0	200000	0.3186	0.2089
0.01	45307	0.0491	0.0288
0.05	15803	0.0093	0.0049
0.1	8375	0.0029	0.0015
0.2	2901	0.0047	0.0026
0.3	1348	0.0029	0.0017
0.4	741	0.0031	0.0015
0.5	454	0.0049	0.0036

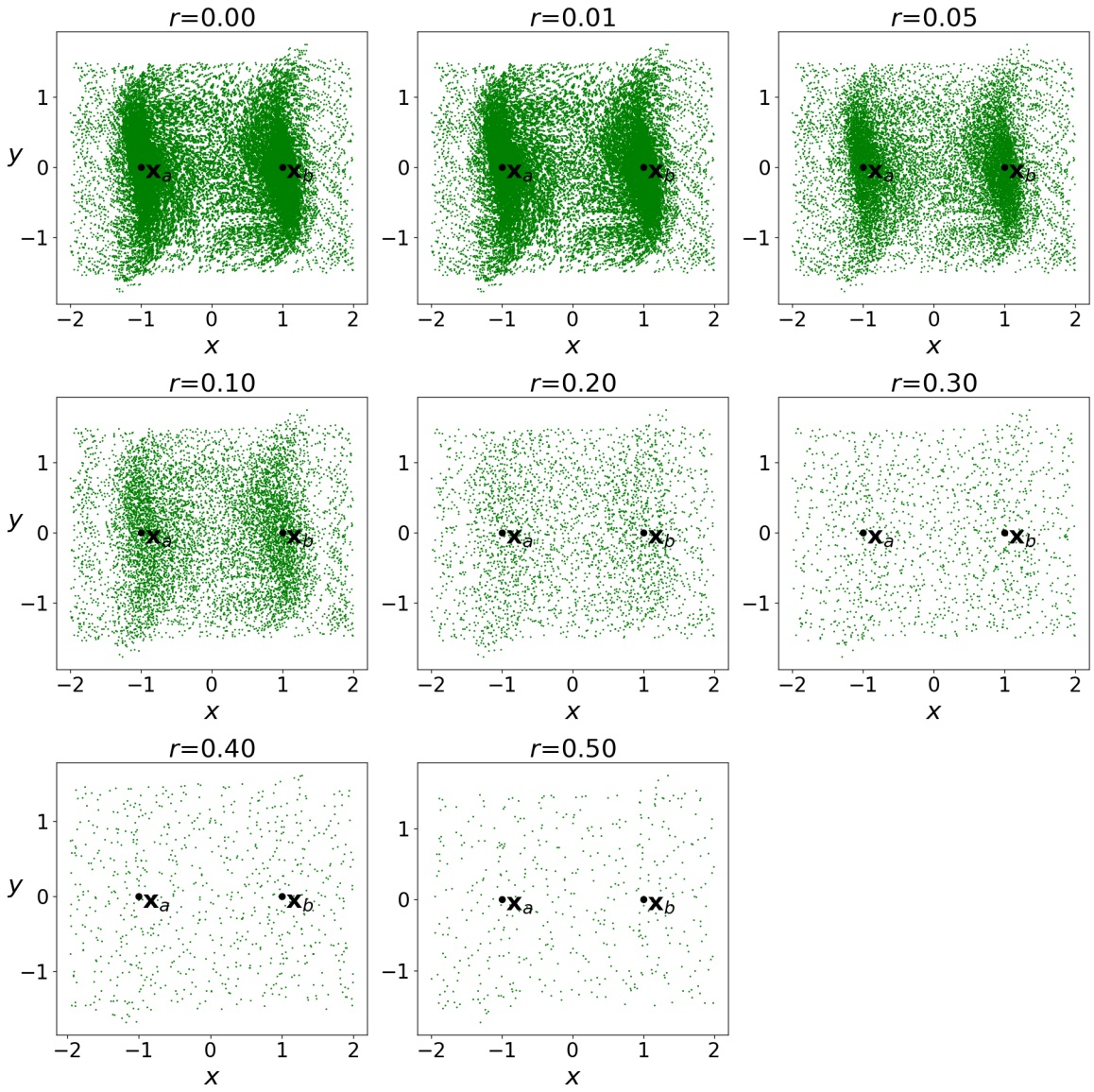

Figure 8 (Example 1): Plots of the sampled representative data points projected onto the xy-plane for different values of r.