

Computing large deviation prefactors of stochastic dynamical systems based on machine learning

Yang Li(李扬), Shenglan Yuan(袁胜兰), Linghongzhi Lu(陆凌宏志), and Xianbin Liu(刘先斌)

Citation: Chin. Phys. B, 2024, 33 (**4**): 040501. DOI: 10.1088/1674-1056/ad12a8
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Detecting physical laws from data of stochastic dynamical systems perturbed by non-Gaussian α-stable Lévy noise

Linghongzhi Lu(陆凌弘志), Yang Li(李扬), and Xianbin Liu(刘先斌)
Chin. Phys. B, 2023, 32 (**5**): 050501. DOI: 10.1088/1674-1056/aca7ee

Data-driven modeling of a four-dimensional stochastic projectile system

Yong Huang(黄勇) and Yang Li(李扬)
Chin. Phys. B, 2022, 31 (**7**): 070501. DOI: 10.1088/1674-1056/ac4e0d

Stability analysis of hydro-turbine governing system based on machine learning

Yuansheng Chen(陈元盛) and Fei Tong(仝飞)
Chin. Phys. B, 2021, 30 (**12**): 120509. DOI: 10.1088/1674-1056/ac04a9

Complex network perspective on modelling chaotic systems via machine learning

Tong-Feng Weng(翁同峰), Xin-Xin Cao(曹欣欣), and Hui-Jie Yang(杨会杰)
Chin. Phys. B, 2021, 30 (**6**): 060506. DOI: 10.1088/1674-1056/abd9b3

Nonlinear dynamics of a classical rotating pendulum system with multiple excitations

Ning Han(韩宁) and Pei-Pei Lu(鲁佩佩)
Chin. Phys. B, 2020, 29 (**11**): 110502. DOI: 10.1088/1674-1056/ab9df2

Computing large deviation prefactors of stochastic dynamical systems based on machine learning

Yang Li(李扬)[1], Shenglan Yuan(袁胜兰)[2,3,†], Linghongzhi Lu(陆凌宏志)[4], and Xianbin Liu(刘先斌)[4]

[1] *School of Automation, Nanjing University of Science and Technology, Nanjing* 210094, *China*
[2] *Department of Mathematics, School of Sciences, Great Bay University, Dongguan* 523000, *China*
[3] *Center for Mathematical Sciences, Huazhong University of Science and Technology, Wuhan* 430074, *China*
[4] *State Key Laboratory of Mechanics and Control for Aerospace Structures, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing* 210016, *China*

(Received 19 June 2023; revised manuscript received 28 November 2023; accepted manuscript online 6 December 2023)

We present a large deviation theory that characterizes the exponential estimate for rare events in stochastic dynamical systems in the limit of weak noise. We aim to consider a next-to-leading-order approximation for more accurate calculation of the mean exit time by computing large deviation prefactors with the aid of machine learning. More specifically, we design a neural network framework to compute quasipotential, most probable paths and prefactors based on the orthogonal decomposition of a vector field. We corroborate the higher effectiveness and accuracy of our algorithm with two toy models. Numerical experiments demonstrate its powerful functionality in exploring the internal mechanism of rare events triggered by weak random fluctuations.

Keywords: machine learning, large deviation prefactors, stochastic dynamical systems, rare events

PACS: 05.10.–a, 05.10.Gg, 05.40.–a, 02.50.–r **DOI:** 10.1088/1674-1056/ad12a8

1. Introduction

The phenomena of rare events exiting from the domain of attraction of a stable state induced by noise have been attracting increasing attention in recent years, ranging from physics,[1–3] chemistry[4] and biology[5,6] to engineering.[7,8] Even for weak noise, rare events will almost certainly occur if the observations are performed on a sufficiently long time scale. The expected time required for observing the occurrence of such phenomena, i.e., the mean exit time, typically grows exponentially as the intensity of the random perturbations tends to zero.

Freidlin and Wentzell established large deviation theory to understand and analyze such dynamics in the limit of weak noise.[9] They proposed the vital concept of an action functional to estimate the probability of the stochastic trajectory passing through the neighborhood of a given curve. The global minimum of the action functional is called quasipotential and exponentially dominates the magnitude of the stationary probability distribution and mean exit time. However, the exponential estimate of large deviation theory is too rough since it smears all the possible polynomial coefficients prior to the exponent. Thus a reasonable estimation or an asymptotic approximation of the large deviation prefactor is required in order to gain a more accurate mean exit time.

Analytically, the prefactor function can be approximately derived by path integral formulation and the perturbation approximation method.[10–13] Numerically, the shooting method is also a feasible technique for computing the prefactor of mean exit time. Its idea is to derive a group of ordinary differential equations via the Wentzel–Kramers–Brillouin (WKB) approximation and method of characteristics.[12,14–19] Then the WKB prefactor and further the prefactor of mean exit time can be computed by integrating this group of equations. However, this method requires searching for the most probable path connecting the fixed point to the point with minimal quasipotential on the boundary, which is a challenging task, especially for high-dimensional systems.

We have witnessed in recent years the rapid development of data science and computer technology.[20] In view of the powerful nonlinear representational ability of machine learning, many researchers have applied it to the investigation of stochastic dynamics. For example, machine learning methods have been used to discover stochastic dynamical systems from sample path data via non-local Kramers–Moyal formulas,[21,22] physics-informed neural networks[23,24] and variational inference.[25] They have greatly improved the efficiency and accuracy of solving physical quantities of stochastic dynamical systems by computing the most probable path,[26,27] quasipotential[28–30] and probability density.[31] These experimental results have brought about potential applications of a combination of machine learning and stochastic dynamics, including forward and inverse problems.

In this research, our goal is to develop a machine learning method to compute the large deviation prefactors in highly complex nonlinear non-deterministic systems. We generate a specific algorithm for training multilayer artificial networks

†Corresponding author. E-mail: shenglanyuan@hust.edu.cn

© 2024 Chinese Physical Society and IOP Publishing Ltd

and thus perform complex computations through a learning process. The information-processing architecture is composed of neurons and synaptic weights to express complex functions. A specific loss function is constructed to approximate the quasipotential, most probable path and mean exit time.

The structure of this article is as follows. In Section 2 we simply introduce the large deviation theory and the quasipotential concept. In Section 3 we describe the results for the prefactors of mean exit time for the cases of non-characteristic and characteristic boundaries. In Section 4 we design the machine learning method for computing the large deviation prefactors. In Section 5 numerical experiments are performed to verify the effectiveness and accuracy of the algorithm. Section 6 presents the conclusion and innovations.

2. Large deviation theory

We consider here an n-dimensional stochastic differential equation (SDE) modeled by

$$\mathrm{d}x = b(x)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}B_t, \quad x \in \mathbb{R}^n, \tag{1}$$

where the vector field $b(x) = (b^1(x), b^2(x), \ldots, b^n(x))$ is continuously differentiable, $B_t = (B_t^1, B_t^2, \ldots, B_t^n)$ denotes Brownian motion and ε represents a small noise intensity. We assume that $\bar{x}$ is the stable fixed point of the corresponding deterministic system $\dot{x} = b(x)$ and the region D is its basin of attraction. Under weak random perturbations, the stochastic process will wander around the fixed point. If the observation is performed over a sufficiently long time scale, rare events, such as exit phenomena from the domain D induced by noise, will occur with probability 1, even though noise is weak. This is referred to as the exit problem in stochastic dynamics. Freidlin and Wentzell proposed large deviation theory to analyze and compute these phenomena theoretically.[9]

For the solution $x(t)$, $t \in [0,T]$ of system (1), Freidlin and Wentzell defined the action functional concept, which is in the form of

$$\mathscr{S}[x] = \int_0^T \mathscr{L}(x,\dot{x})\,\mathrm{d}t$$

with the corresponding Lagrangian $\mathscr{L}(x,\dot{x}) = (1/2)|\dot{x} - b(x)|^2$. For sufficiently small ε and δ, the probability of the stochastic trajectory passing through the neighborhood of a reference curve $\varphi(t)$ is estimated as

$$\mathbb{P}\left\{\sup_{t\in[0,T]} |x(t) - \varphi(t)| < \delta\right\} \sim \exp\left\{-\varepsilon^{-1}\mathscr{S}[\varphi]\right\}. \tag{2}$$

Thus the exit problem is transformed into a constrained minimization problem of an action functional. Another important notion in large deviation theory is the quasipotential, defined as

$$V(x) := \inf_T \inf_{\varphi\in C[0,T]} \left\{\mathscr{S}[\varphi] : \varphi(0) = \bar{x}, \varphi(T) = x\right\}. \tag{3}$$

Note that the quasipotential of the fixed point equals zero, i.e., $V(\bar{x}) = 0$. The quasipotential $V(x)$ actually illustrates the possibility of the system appearing around x.

The associated Fokker–Planck equation of SDE (1) is formulated as

$$\partial_t p(x,t) = -\sum_{i=1}^n \partial_i(b^i(x)p(x,t)) + \frac{1}{2}\varepsilon\sum_{i=1}^n \partial_i^2 p(x,t),$$

where $p(x,t)$ is the probability density. The stationary probability density $p_{\mathrm{s}}(x)$ satisfies

$$\begin{aligned} 0 &= -\sum_{i=1}^n \partial_i(b^i(x)p_{\mathrm{s}}(x)) + \frac{1}{2}\varepsilon\sum_{i=1}^n \partial_i^2 p_{\mathrm{s}}(x) \\ &= -\langle b(x), \nabla p_{\mathrm{s}}(x)\rangle + \frac{1}{2}\varepsilon\Delta p_{\mathrm{s}}(x). \end{aligned} \tag{4}$$

Assume that the stationary distribution is approached by the WKB approximation

$$p_{\mathrm{s}}(x) \sim C(x)\exp\{-\varepsilon^{-1}V(x)\}, \tag{5}$$

where $C(x)$ stands for the WKB prefactor. Substituting the WKB approximation (5) into the stationary Fokker–Planck equation (4) yields the Hamilton–Jacobi equation

$$H(x,\nabla V(x)) = \langle b(x), \nabla V(x)\rangle + \frac{1}{2}\langle \nabla V(x), \nabla V(x)\rangle = 0. \tag{6}$$

Introducing the canonical momentum

$$p := \frac{\partial \mathscr{L}(x,\dot{x})}{\partial \dot{x}} = \dot{x} - b(x),$$

the Hamiltonian $H(x,p) = \langle b(x), p\rangle + (1/2)\langle p, p\rangle$ is actually the Legendre transformation of the Lagrangian.

According to Eq. (6), it follows that

$$\langle b(x) + \frac{1}{2}\nabla V(x), \nabla V(x)\rangle = 0,$$

which infers the orthogonality relation

$$b(x) + \frac{1}{2}\nabla V(x) \perp \nabla V(x). \tag{7}$$

We define $l(x) := b(x) + \frac{1}{2}\nabla V(x)$. It is immediately obvious that $b(x) = -(1/2)\nabla V(x) + l(x)$, which is concerned with an orthogonal decomposition of the vector field $b(x)$ as seen in Fig. 1. This orthogonal decomposition is unique because the quasipotential is unique. The vector field $G(x) := (1/2)\nabla V(x) + l(x)$ dictates the direction of the most probable path.

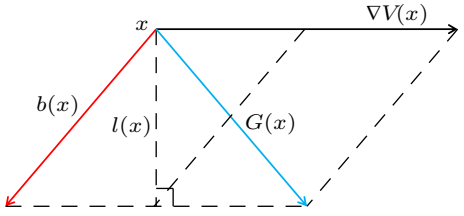


Fig. 1. The orthogonal decomposition of the vector field $b(x) = -(1/2)\nabla V(x) + l(x)$. The vector $G(x) = (1/2)\nabla V(x) + l(x)$ dominates the direction of the most probable path.

It should be pointed out that the WKB prefactor is determined by

$$C(x) \sim \sqrt{\frac{\det \nabla^2 V(\bar{x})}{(2\pi\varepsilon)^n}} \exp\Big\{-\int_{-\infty}^{0} \mathrm{div}\, l(\varphi_t^x)\,\mathrm{d}t\Big\},$$

where φ^x represents the most probable exit path ending at x, div denotes the divergence and $\bar{x}$ is the point at which V attains its global minimum, i.e., the stable fixed point.[32]

3. Mean exit time

Given an open subset domain $D \subset \mathbb{R}^n$, the first exit time is expressed by the stopping time

$$\tau_D^\varepsilon := \inf\{t \geq 0 : x(t) \notin D, x(0) = \bar{x}\}, \quad \text{for } \bar{x} \in D.$$

Estimating the mean first exit time $\mathbb{E}\tau_D^\varepsilon$ starting from the stable fixed point $\bar{x}$ requires minimizing the action functional $\mathscr{S}[x]$ over all exit paths and times in terms of the Freidlin–Wentzell large deviation theory. Performing this minimization over paths from the point $\bar{x}$ to the basin boundary ∂D yields the Arrhenius law

$$\lim_{\varepsilon\to 0} \varepsilon \ln \mathbb{E}\tau_D^\varepsilon = \inf_{x\in\partial D} V(x).$$

Therefore,

$$\mathbb{E}\tau_D^\varepsilon = L_D^\varepsilon \exp\{\varepsilon^{-1} \inf_{x\in\partial D} V(x)\}. \tag{8}$$

We compute the above prefactor L_D^ε in two different cases.

Case A Non-characteristic boundary.[12]

We assume that the domain D is an open, smooth and connected subset of $\mathbb{R}^n$ satisfying the following conditions, where $n(y)$ indicates the exterior normal vector at $y \in \partial D$.

(A1) The deterministic system $\dot{x} = b(x)$ possesses a unique fixed point $\bar{x}$ in D, which attracts all the trajectories that start from D, and $\langle b(y), n(y)\rangle < 0$ for all $y \in \partial D$.

(A2) The function V is C^1-continuous in D; for any $x \in \bar{D}$, the most probable path φ_t^x goes to $\bar{x}$ as $t \to -\infty$; and $\langle \nabla V(y), n(y)\rangle > 0$ for all $y \in \partial D$.

(A3) The minimum of V over ∂D is reached at a single point x^*, at which

$$\mu^* = \langle \frac{1}{2}\nabla V(x^*) + l(x^*), n(x^*)\rangle > 0,$$

and the quadratic form $h^* : \xi \mapsto \langle \xi, \nabla^2 V(x^*)\xi\rangle$ has positive eigenvalues on the hyperplane $n(x^*)^\perp = \{\xi \in \mathbb{R}^n : \langle \xi, n(x^*)\rangle = 0\}$.

If $\langle b(y), n(y)\rangle < 0$ for all $y \in \partial D$, the boundary is said to be non-characteristic. This condition ensures that the dynamical trajectories, which start from the closure $\bar{D}$, will remain within D, and that the vector field is transverse to the boundary. By Assumption (A1), we derive the integral formula[12,32]

$$\lambda_D^\varepsilon = \int_{x\in\partial D} \langle \frac{1}{2}\nabla V(x) + l(x), n(x)\rangle C(x) \exp\{-\varepsilon^{-1} V(x)\}\,\mathrm{d}x$$

for the exit rate $\lambda_D^\varepsilon = [\mathbb{E}\tau_D^\varepsilon]^{-1}$. Using the second-order expansion of V in the neighborhood of x^* in this formula, we obtain the equivalent relation of the prefactor

$$L_D^\varepsilon \sim \frac{1}{C(x^*)\mu^*}\sqrt{\frac{\det h^*}{(2\pi\varepsilon)^{n-1}}} \tag{9}$$

$$\sim \frac{1}{\mu^*}\sqrt{\frac{2\pi\varepsilon \det h^*}{\det \nabla^2 V(\bar{x})}} \exp\Big\{\int_{-\infty}^{0} \mathrm{div} l(\varphi_t^{x^*})\,\mathrm{d}t\Big\}, \tag{10}$$

where we use

$$C(x^*) \sim \sqrt{\frac{\det \nabla^2 V(\bar{x})}{(2\pi\varepsilon)^n}} \exp\Big\{-\int_{-\infty}^{0} \mathrm{div}\, l(\varphi_t^{x^*})\,\mathrm{d}t\Big\}.$$

Case B Characteristic boundary.[12]

The basin domain D is characteristic in the sense that $\langle b(y), n(y)\rangle = 0$ for all $y \in D$. We consider the metastable case when the deterministic system $\dot{x} = b(x)$ possesses two stable fixed points $\bar{x}_1$ and $\bar{x}_2$, whose basins of attraction are separated by a smooth hypersurface S. We consider the exit events from the basin of attraction D for $\bar{x}_1$ and formulate the following set of assumptions:

(B1) All the trajectories of the deterministic system $\dot{x} = b(x)$ that started on S remain in S and converge to a single fixed point $x^* \in S$; in addition, the Jacobi matrix $\nabla b(x^*)$ possesses $n-1$ eigenvalues with a negative real part and a single positive eigenvalue λ^*.

(B2) Indicating the quasipotential with respect to $\bar{x}_1$ as V, there exists a unique (up to time shift) trajectory $\rho = (\rho_t)_{t\in\mathbb{R}} \subset D$ such that

$$\lim_{t\to-\infty} \rho_t = \bar{x}_1, \quad \lim_{t\to+\infty} \rho_t = x^*, \quad \text{and} \quad V(x^*) = \mathscr{S}_{-\infty,+\infty}[\rho].$$

(B3) The quasipotential V is smooth in the neighborhood of $\rho = (\rho_t)_{t\in\mathbb{R}}$, and the vector field l defined by $l(x) = b(x) + (1/2)\nabla V(x)$ satisfies the orthogonality relation.

In this context, the quasipotential V achieves its minimum on S at the point x^*. Moreover, the path ρ is called the most probable exit path and it satisfies

$$\dot{\rho}_t = \frac{1}{2}\nabla V(\rho_t) + l(\rho_t), \quad \forall t \in \mathbb{R}.$$

For any $t \in \mathbb{R}$, it coincides with the path $(\varphi_s^x)_{s\leq 0}$ connecting $\bar{x}$ with $x = \rho_t$, in the sense that

$$\varphi_s^x = \rho_{s+t}, \quad \forall s \leq 0.$$

In order to describe the prefactor L_D^ε in this case, we formulate the following supplementary assumption:

(B4) The matrix $H^* = \lim_{t\to+\infty} \nabla^2 V(\rho_t)$ exists and has $n-1$ positive eigenvalues and one negative eigenvalue.

Based on these four assumptions, an asymptotic formula for estimating the expected time taken by the process to exit D was obtained in Ref. [32], i.e.,

$$L_D^\varepsilon \sim \frac{\pi}{\lambda^*}\sqrt{\frac{|\det H^*|}{\det \nabla^2 V(\bar{x})}} \exp\Big\{\int_{-\infty}^{\infty} \mathrm{div}\, l(\varphi_t)\,\mathrm{d}t\Big\}. \tag{11}$$

4. Method

To compute the large deviation prefactors defined in Section 3, we developed a machine learning method inspired by Ref. [29]. The critical idea is to design the graphic of the feed-forward multilayer neural network shown in Fig. 2 to realize the orthogonal decomposition of the vector field. The neural network is made up of three basic components: an input vector $a^{(0)} = x \in \mathbb{R}^n$, a set of hidden layers with some neurons connected by linear functions and nonlinear activation functions, and an output layer $a^{(L)} = (\widehat{V}_\theta, l_\theta) \in \mathbb{R}^{n+1}$. For $l = 1, \ldots, L-1$, there are n_l neurons in the l-th hidden layer. The subscript θ denotes the training parameters in the neural network.

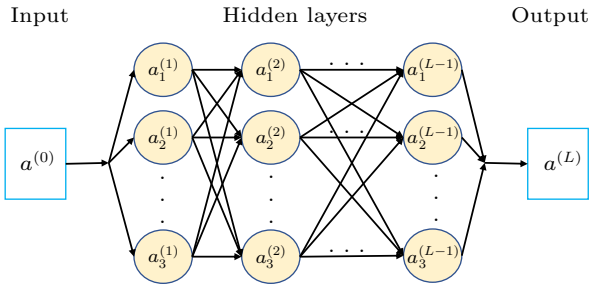


Fig. 2. Architecture of the neural network with $L-1$ hidden layers. The input is $a^{(0)}$, the output is $a^{(L)}$, and $a_j^{(l)}$ denotes the value of the j-th neuron in the l-th hidden layer for $j = 1, \ldots, n_l$, $l = 1, \ldots, L-1$.

Note that the quasipotential is similar to a quadratic function around the stable fixed point $\bar{x}$. In order to guarantee that the quasipotential V_θ and its gradient $|V_\theta|$ are all unbounded, we define the quasipotential $V_\theta = \widehat{V}_\theta + |x - \bar{x}|^2$. To train the neural network, we construct the loss function $L := L_{\text{dyn}} + \gamma_1 L_{\text{orth}} + \gamma_2 L_0$, where γ_1 and γ_2 are weighted parameters harmonizing the proportion of several parts of the loss function.

We emphasize that the SDE (1) or the vector field is already known, and we need to implement it in the loss function numerically. First, we should find the stable fixed point $\bar{x}$ and choose a domain of interest D surrounding the fixed point which is a subset of its basin of attraction. Then we randomly select points $X = \{x_i, i = 1, 2, \ldots, N\}$ uniformly distributed in this domain. Since the vector field can be decomposed as $b(x) = -(1/2)\nabla V(x) + l(x)$, the first part of the loss function can be set as

$$L_{\text{dyn}} = \frac{1}{N}\sum_{i=1}^{N}[b(x_i) + \frac{1}{2}\nabla V_\theta(x_i) - l_\theta(x_i)]^2,$$

where the gradient of the quasipotential in the loss function is realized by the automatic differentiation technique. Due to the orthogonal condition $\nabla V(x) \perp l(x)$, we specify the second part of the loss function as

$$L_{\text{orth}} = \frac{1}{N}\sum_{i=1}^{N}\frac{[\nabla V_\theta(x_i)\cdot l_\theta(x_i)]^2}{|\nabla V_\theta(x_i)|^2 \cdot |l_\theta(x_i)|^2 + \delta},$$

where we add a sufficiently small parameter $\delta > 0$, avoiding the divergence of L_{orth} in the case of $l_\theta(x) = 0$. The quasipotential is zero at the fixed point, so we fix $L_0 = V_\theta(\bar{x})^2$. After choosing suitable hyperparameters such as neural network layers, optimizer, learning rate η, activation function and weighted parameters γ_1 and γ_2, the neural network can be trained by gradient descent method; we then obtain the approximate quasipotential $V_\theta(x)$ and rotational component $l_\theta(x)$.

To examine the accuracy of the algorithm, we define the approximation error functions for the quasipotential and the rotational component of vector field by

$$e_V = \frac{\max_x |V_\theta(x) - V(x)|^2}{\max_x |V(x)|^2}, \quad e_l = \frac{\max_x |l_\theta(x) - l(x)|^2}{\max_x |l(x)|^2},$$

where V_θ and l_θ are the predicted results of the neural network, while V and l are the true results.

According to the expression of the prefactor L_D^ε of the mean first exit time in Section 3, the computation of the prefactor essentially depends on the results of the most probable path φ_t and the rotational component $l(x)$. After training the neural network, the most probable path satisfies

$$\dot{\varphi}_t = b(\varphi_t) + \nabla V_\theta(\varphi_t).$$

For numerical convenience, we transform the time parameter of this equation into the arc length σ of the most probable path. Then

$$\frac{\mathrm{d}\sigma}{\mathrm{d}t} = |\dot{\varphi}| = |b(\varphi) + \nabla V_\theta(\varphi)| = |b(\varphi)|,$$

where the last equality comes from the relation $|G(x)| = |b(x)|$ as in Fig. 1. Therefore, the most probable path satisfies

$$\frac{\mathrm{d}\varphi}{\mathrm{d}\sigma} = \frac{b(\varphi) + \nabla V_\theta(\varphi)}{|b(\varphi)|}. \tag{12}$$

We can obtain the most probable path from inverse time integration of the above equation starting from the endpoint x^* until reaching the neighborhood of the fixed point.

With regard to Case A, we choose the point x^* at which the quasipotential reaches its minimum value on the boundary ∂D. Note that x^* can be determined via the Lagrange multiplier method. This can be seen from the fact that it is an optimization problem of $V_\theta(x)$ with the constraint $g(x) = 0$, which is assumed to be the formulation of the boundary. In addition, we calculate $\operatorname{div} l_\theta(x)$ on the basis of the neural network results of $l_\theta(x)$. Then the integral in L_D^ε can be transformed into

$$\int_{-\infty}^{0} \operatorname{div} l_\theta(\varphi_t^{x^*})\,\mathrm{d}t = \int_0^L \frac{\operatorname{div} l_\theta(\varphi_\sigma^{x^*})}{|b(\varphi_\sigma^{x^*})|}\,\mathrm{d}\sigma.$$

The calculation of the WKB prefactor $C(x^*)$ is already included in the calculation of L_D^ε for Case A. It will no longer be considered separately.

In Case B, we take x^* as the saddle point $x_{\rm sad}$. For the sake of efficient numerical implementation, our actual operation is to take $x^* = x_{\rm sad} + \delta_1 e_u$ with the help of a small parameter $\delta_1 > 0$, where e_u is the unstable direction of $x_{\rm sad}$. For a small parameter $\delta_2 > 0$, the inverse time integration terminates when it enters into the δ_2 neighborhood of the fixed point $\bar{x}$. For Case B, we rewrite the integral in L_D^ε as

$$\int_{-\infty}^{\infty} \mathrm{div}\, l_\theta(\varphi_t)\,\mathrm{d}t = \int_0^L \frac{\mathrm{div}\, l_\theta(\varphi_\sigma)}{|b(\varphi_\sigma)|}\,\mathrm{d}\sigma.$$

The framework of the algorithm is outlined in Table 1.

Table 1. Algorithm for computation of prefactor.

Algorithm 1: Computing large deviation prefactor.

Input: Stochastic differential equation (1).
Output: Quasipotential $V_\theta(x)$, most probable path $\varphi_t^{x^*}$, prefactor L_D^ε, mean exit time $\mathbb{E}\tau_D^\varepsilon$.
Data: Compute stable fixed point $\bar{x}$, randomly sample $X = \{x_i : i = 1,2,\ldots,N\}$ in a subset of the basin of attraction of $\bar{x}$.
Initialization: Hyperparameters: neural network layers, optimizer, learning rate η, activation function, weighted parameters γ_1 and γ_2.
Loss function: $L = L_{\rm dyn} + \gamma_1 L_{\rm orth} + \gamma_2 L_0$.
Training:
for epoch=1:iteration
  calculate loss function and update neural network parameters
end
Error: Compute approximation errors e_V and e_l
Result: Determine the exit point x^* (saddle for characteristic boundary or location of minimal quasipotential for non-characteristic boundary). Integrate most probable path φ^{x^*} by Eq. (12). Calculate divergence of rotational component $\mathrm{div}\, l_\theta(\varphi^{x^*})$. Compute prefactor L_D^ε via Eq. (9) or Eq. (11) and mean exit time $\mathbb{E}\tau_D^\varepsilon = L_D^\varepsilon \exp\{\varepsilon^{-1} V_\theta(x^*)\}$.

5. Numerical experiments

To verify the effectiveness and accuracy of the algorithm, we use two toy models with two and five dimensions to test our approach. For the convenience of comparison, the two chosen examples both have explicit quasipotential functions as the ground truth. For both numerical experiments, the selection of superparameters in the algorithm consists of six hidden layers with 20 neurons per layer. We choose the Adam optimizer with the learning rate 0.002. The nonlinear activation function for hidden layer is the hyperbolic tangent function tanh. The activation function for output is the identity function. The number of training epochs is 100000. We fix the hyperparameters $\gamma_1 = 1$, $\gamma_2 = 0.1$, $\delta = 0.001$, $\delta_1 = 0.05$ and $\delta_2 = 0.01$.

5.1. A two-dimensional system

The first example we investigate is a two-dimensional system

$$\mathrm{d}x = b(x)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}B_t, \quad x \in \mathbb{R}^2.$$

Here the vector field $b(x)$ has an explicit orthogonal decomposition $b(x) = -(1/2)\nabla V(x) + l(x)$ with the following quasipotential:

$$V(x_1,x_2) = v_1(x_1) + v_2(x_2),$$

where $v_1(x_1) = (1/2)x_1^4 - x_1^2 + 1/2$ and $v_2(x_2) = \alpha x_2^2$. The rotational component $l(x)$ has the following form:

$$l(x_1,x_2) = c(x_1,x_2)\begin{pmatrix} -v_2'(x_2) \\ v_1'(x_1) \end{pmatrix},$$

where $c(x_1,x_2) = \beta x_1$. It follows from

$$\langle \nabla V(x), l(x)\rangle = c(x_1,x_2)\begin{pmatrix} v_1'(x_1) & v_2'(x_2) \end{pmatrix}\begin{pmatrix} -v_2'(x_2) \\ v_1'(x_1) \end{pmatrix} = 0$$

that $\nabla V(x) \perp l(x)$.

Considering the case $\alpha = 0.5$ and $\beta = 3$, we have

$$b(x) = \begin{pmatrix} x_1 - x_1^3 - 3x_1x_2 \\ 6x_1^4 - 6x_1^2 - (1/2)x_2 \end{pmatrix}. \tag{13}$$

If $b(x) = 0$, then we know that there are three fixed points $\mathrm{SN1} = (-1,0)^{\rm T}$, $\mathrm{SN2} = (1,0)^{\rm T}$ and $\mathrm{US} = (0,0)^{\rm T}$. To classify those fixed points, we take into account the Jacobian matrix

$$J = \begin{pmatrix} 1 - 3x_1^2 - 3x_2 & -3x_1 \\ 24x_1^3 - 12x_1 & -1/2 \end{pmatrix}.$$

The Jacobian matrix at $(\pm 1,0)^{\rm T}$ is given by

$$J_{(\pm 1,0)^{\rm T}} = \begin{pmatrix} -2 & \mp 3 \\ \pm 12 & -1/2 \end{pmatrix}.$$

The characteristic equation is $\lambda^2 + (5/2)\lambda + 37 = 0$, which has complex roots

$$\lambda_\pm = \frac{-5 \pm 9\sqrt{7}\mathrm{i}}{4}.$$

Thus the two fixed points $(\pm 1,0)^{\rm T}$ are asymptotically stable. Using the Jacobian matrix

$$J_{(0,0)^{\rm T}} = \begin{pmatrix} 1 & 0 \\ 0 & -1/2 \end{pmatrix},$$

the eigenvalues for $(0,0)^{\rm T}$ are $\lambda_1 = 1$ and $\lambda_2 = -1/2$. Since one eigenvalue is positive and the other is negative, the origin is an unstable saddle point. We sketch a phase portrait for the deterministic system $\dot{x} = b(x)$ in Fig. 3. The eigenvector for $\lambda_2 = -1/2$ is $(0,1)^{\rm T}$. The stable manifold is the x_2-axis, which forms the boundary of the left and right basins of attraction. Thanks to the symmetry of the image, we only consider the exit problem starting from the left side.

Now we apply the proposed method to this model to compute the prefactor. We randomly and uniformly select $N = 100, 1000, 10000, 100000$ points on the region $[-1.5, 0] \times$

$[-0.8, 0.8]$. After completing the training process, we calculate the approximation errors e_V and e_l with the results described in Table 2. In the beginning, the errors e_V and e_l gradually decrease as the amount of data increases from $N = 100$ to $N = 10000$. However, it can be observed that the change is not significant for $N = 100000$. At this point, the accuracy of the algorithm is limited by other parameters.

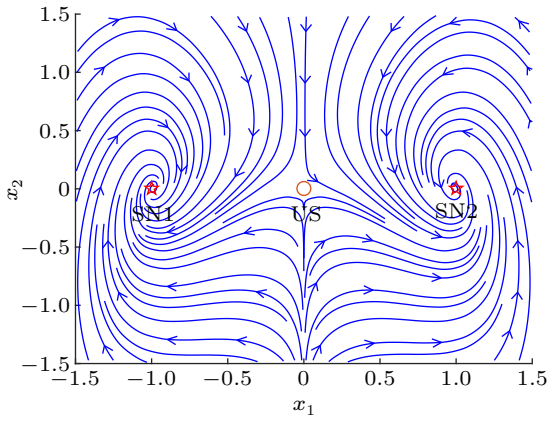


Fig. 3. The flow of the vector field $b(x)$ as in Eq. (13). SN1 $= (-1,0)^{\mathrm{T}}$ and SN2 $= (1,0)^{\mathrm{T}}$ denote two stable fixed points, and US $= (0,0)^{\mathrm{T}}$ indicates a saddle.

Table 2. Approximation errors.

N	100	1000	10000	100000
e_V	0.3122%	0.1392%	0.1267%	0.0681%
e_l	1.3692%	0.2127%	0.0738%	0.0965%

Comparing the numerical results of learned and true quasipotential and rotational components displayed in Fig. 4 for $N = 100000$, it can be seen that they are consistent with each other.

When $N = 100000$, we compute the prefactors for Case A and Case B.

Case A Non-characteristic boundary.

As depicted in Fig. 4, we choose the dashed black line $x_1 = -0.5$ as the non-characteristic boundary for the basin of attraction of SN1. The pink star $x^* = (-0.4990, -0.0048)^{\mathrm{T}}$ denotes the exit point, which represents the minimum of the quasipotential on the boundary. By utilizing the results of the neural network and the true system starting from x^*, we simulate the most probable paths marked as red and dashed green curves, respectively. As indicated in Fig. 5, the two curves match very well.

We define $\bar{H} := \nabla^2 V(\bar{x})$, which satisfies the algebraic Riccati equation

$$\bar{H}^2 = \bar{Q}^{\mathrm{T}}\bar{H} + \bar{H}\bar{Q},$$

where the matrix

$$\bar{Q} = -\nabla b(\bar{x}) = \begin{pmatrix} 2 & -\alpha\beta \\ 2\beta & \alpha \end{pmatrix}.$$

Then we have

$$\bar{H} = \begin{pmatrix} 4 & 0 \\ 0 & 2\alpha \end{pmatrix}.$$

Owing to the calculation results of $l_\theta(x)$, we can get $L_D^\varepsilon \sim 2.1525\sqrt{\varepsilon}$. The true values of $l(x)$ can lead to $L_D^\varepsilon \sim 2.2164\sqrt{\varepsilon}$. The approximate values of L_D^ε employing two different methods are very close.

Substituting the prefactor $L_D^\varepsilon \sim 2.1525\sqrt{\varepsilon}$ and the quasipotential $V_\theta(x^*) = 0.2835$ into the expression for mean exit time $\mathbb{E}\tau_D^\varepsilon = L_D^\varepsilon \exp\{\varepsilon^{-1} V_\theta(x^*)\}$, we plot a comparison graph of mean exit times via machine learning and Monte Carlo simulation in Fig. 6. The red asterisk denotes the results of Monte Carlo simulation, which is completely consistent with the learning results and has very high accuracy.

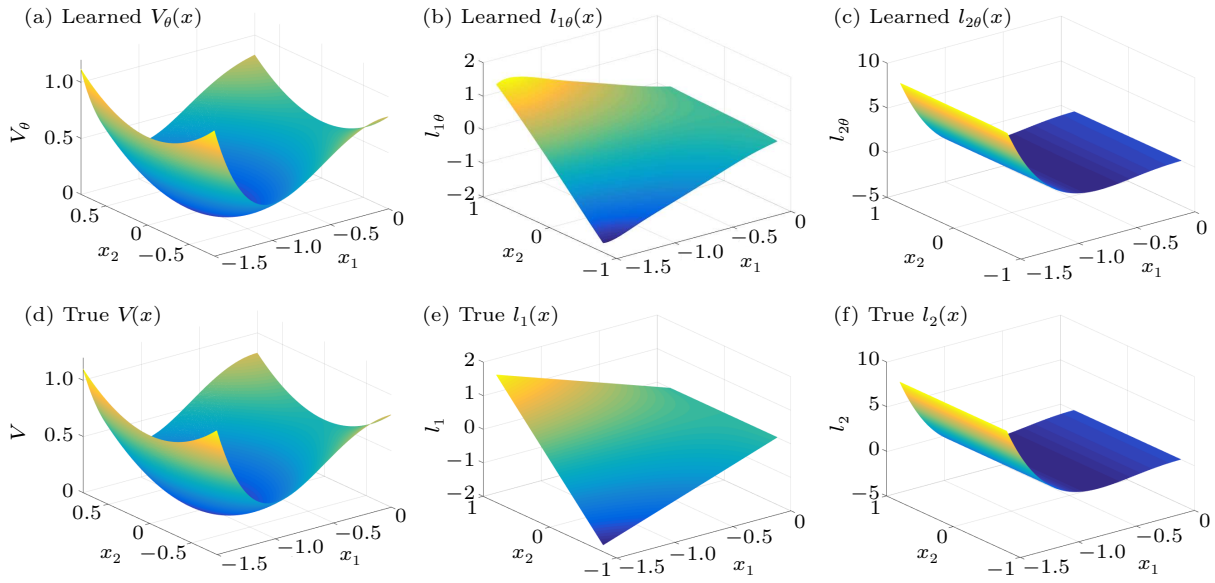


Fig. 4. Comparison between learned and true results of quasipotential $V(x)$ and rotational component $l(x)$ for the basin of the left fixed point SN1 $= (-1,0)^{\mathrm{T}}$ of the two-dimensional system with $N = 100000$ points selected.

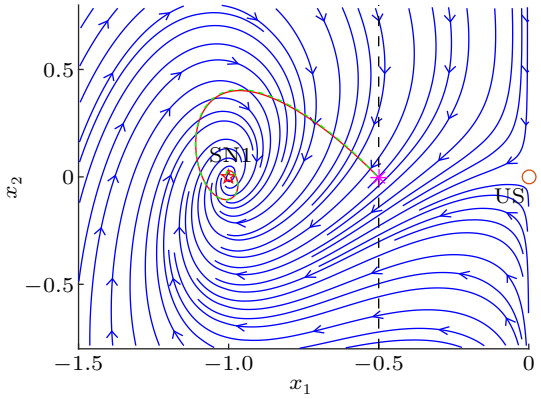


Fig. 5. Comparison between the most probable exit paths for Case A of the two-dimensional system computed by machine learning and the true system, denoted by red and dashed green curves, respectively. The blue curves represent the flow of the vector field and the dashed black line indicates the boundary $x_1 = -0.5$. The pink star $x^* = (-0.4990, -0.0048)^{\mathrm{T}}$ is the exit point.

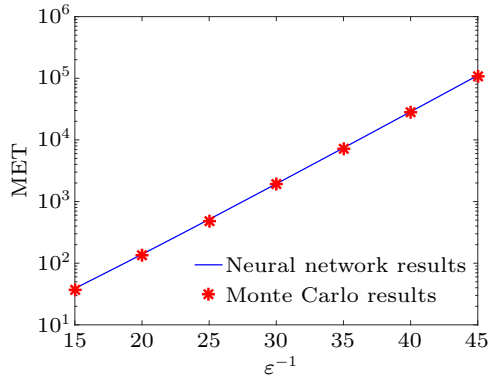


Fig. 6. Comparison between mean exit times versus reciprocal of noise intensity for Case A of the two-dimensional system via machine learning and Monte Carlo simulation, denoted by a blue curve and red star, respectively.

Case B Characteristic boundary.

We treat the x_2-axis, i.e., the stable manifold of US, as the boundary of the left and right basins of attraction. On the basis of the neural network, we acquire the most probable path by the dashed green curve that coincides with the red curve of the true system, as depicted in Fig. 7.

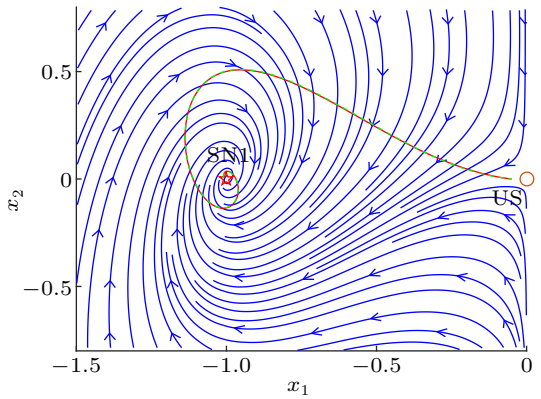


Fig. 7. Comparison between the most probable exit paths for Case B of the two-dimensional system computed by machine learning and the true system, denoted by red and dashed green curves, respectively.

We define $H^* := \nabla^2 V(\mathrm{US})$, which satisfies the algebraic Riccati equation

$$H^{*2} = Q^{*\mathrm{T}} H^* + H^* Q^*,$$

where the matrix

$$Q^* = -\nabla b(\mathrm{US}) = \begin{pmatrix} -1 & 0 \\ 0 & \alpha \end{pmatrix}.$$

Then we have

$$H^* = \begin{pmatrix} -2 & 0 \\ 0 & 2\alpha \end{pmatrix}.$$

We evaluate $L_D^\varepsilon \sim 1.0121$ using the calculated result for $l_\theta(x)$, and obtain $L_D^\varepsilon \sim 1.0958$ on the basis of the calculated result for $l(x)$. Putting $V_\theta(\mathrm{US}) = 0.5081$ and $L_D^\varepsilon \sim 1.0121$ into the mean exit time $\mathbb{E}\tau_D^\varepsilon = L_D^\varepsilon \exp\{\varepsilon^{-1} V(x^*)\}$, a comparison between the neural network and Monte Carlo results is presented in Fig. 8. It is worth noticing that the training result of the neural network is highly consistent with the experimental Monte Carlo results.

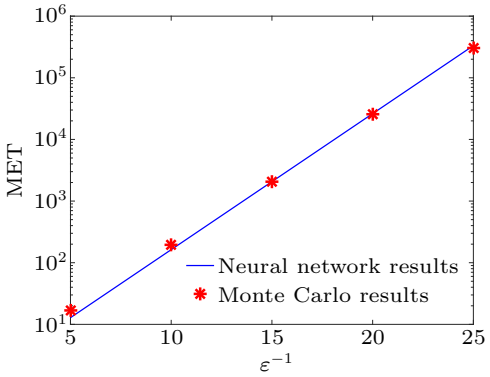


Fig. 8. Comparison between mean exit time versus reciprocal of noise intensity for Case B of the two-dimensional system computed by machine learning and Monte Carlo simulation, denoted by a blue curve and red star, respectively.

5.2. A five-dimensional system

To examine the effectiveness of our algorithm with higher-dimensional systems for verification purposes, we choose a five-dimensional system with the following vector field:

$$b(x) = \begin{pmatrix} -2(x_1^3 - x_1) - (x_2 + x_3 + x_4 + x_5) \\ -x_2 + 2(x_1^3 - x_1) \\ -x_3 + 2(x_1^3 - x_1) \\ -x_4 + 2(x_1^3 - x_1) \\ -x_5 + 2(x_1^3 - x_1) \end{pmatrix}. \tag{14}$$

This system also has an explicit orthogonal decomposition $b(x) = -(1/2)\nabla V(x) + l(x)$ with the quasipotential

$$V(x) = (1 - x_1^2)^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \tag{15}$$

and the rotational component

$$l(x) = \big(-(x_2 + x_3 + x_4 + x_5), 2(x_1^3 - x_1), 2(x_1^3 - x_1), 2(x_1^3 - x_1), 2(x_1^3 - x_1)\big)^{\mathrm{T}}. \tag{16}$$

Note that

$$\nabla V(x) = \left(4(x_1^3 - x_1), 2x_2, 2x_3, 2x_4, 2x_5\right)^{\mathrm{T}}.$$

According to Eqs. (14) and (15), it can be derived that the model $\dot{x} = b(x)$ has three fixed points SN1 $= (-1,0,0,0,0)^{\mathrm{T}}$, SN2 $= (1,0,0,0,0)^{\mathrm{T}}$ and US $= (0,0,0,0,0)^{\mathrm{T}}$. The Jacobian matrix is given by

$$J = \begin{pmatrix} -2(3x_1^2-1) & -1 & -1 & -1 & -1 \\ 2(3x_1^2-1) & -1 & 0 & 0 & 0 \\ 2(3x_1^2-1) & 0 & -1 & 0 & 0 \\ 2(3x_1^2-1) & 0 & 0 & -1 & 0 \\ 2(3x_1^2-1) & 0 & 0 & 0 & -1 \end{pmatrix}.$$

We linearize at SN1 and SN2; hence

$$J_{(\pm1,0,0,0,0)^{\mathrm{T}}} = \begin{pmatrix} -4 & -1 & -1 & -1 & -1 \\ 4 & -1 & 0 & 0 & 0 \\ 4 & 0 & -1 & 0 & 0 \\ 4 & 0 & 0 & -1 & 0 \\ 4 & 0 & 0 & 0 & -1 \end{pmatrix}.$$

The real parts of either of five eigenvalues $(1/2)(-5+\mathrm{i}\sqrt{55}), (1/2)(-5-\mathrm{i}\sqrt{55}), -1, -1, -1$ of the Jacobian matrix $J_{(\pm1,0,0,0,0)^{\mathrm{T}}}$ are negative. Thus the fixed points SN1 and SN2 are asymptotically stable. For the other fixed point,

$$J_{(0,0,0,0,0)^{\mathrm{T}}} = \begin{pmatrix} 2 & -1 & -1 & -1 & -1 \\ -2 & -1 & 0 & 0 & 0 \\ -2 & 0 & -1 & 0 & 0 \\ -2 & 0 & 0 & -1 & 0 \\ -2 & 0 & 0 & 0 & -1 \end{pmatrix}.$$

There is one positive eigenvalue $(1/2)(1+\sqrt{41})$ and four negative eigenvalues $(1/2)(1-\sqrt{41})$, -1, -1, -1, and so the fixed point US is a saddle point. The basins of attraction for two stable fixed points SN1 and SN2 are separated by the hyperplane $x_1 = 0$, which is the stable manifold of the saddle point US. Considering the symmetry, we only handle the exit problem from the basin of attraction of SN1.

We randomly select 100000 points in the domain $[-1.5,0]\times[-1,1]\times[-1,1]\times[-1,1]\times[-1,1]$ to train our neural network. After the training process, we compare the quasipotential and rotational components of deep learning and the true results, and show them in Fig. 9. Since it is impossible to plot the five-dimensional function, we display the results as the function of x_1 and x_3 in the plane $x_2 = 0$, $x_4 = 0$, $x_5 = 0$. It is observed in Fig. 9 that the learned and true functions agree well. We can also compute the approximation errors for the quasipotential $e_V = 0.0421\%$ and the rotational component $e_l = 0.3344\%$, which are both sufficiently small.

Case A Non-characteristic boundary.

We define the hyperplane $x_1 = -0.5$ as the non-characteristic boundary for the basin of attraction of SN1. Then the exit point $x^* = (-0.5, 0.0005, -0.0009, -0.0001, 0)^{\mathrm{T}}$ is computed as the location of the minimum quasipotential on the boundary. As shown in Fig. 10, we numerically integrate the most probable paths from the point x^* using the results of the neural network and the true system, respectively. We find that the two paths almost coincide.

Based on the results for the most probable path, the approximate rotational component $l_\theta(x)$ and the matrix $\bar{H} = \nabla^2 V(\bar{x})$, we estimate the exponential prefactor $L_D^\varepsilon \sim 1.1750\sqrt{\varepsilon}$, which is close to the accurate solution $L_D^\varepsilon \sim 1.1816\sqrt{\varepsilon}$. Substituting the prefactor $L_D^\varepsilon \sim 1.1750\sqrt{\varepsilon}$ and the quasipotential value $V_\theta(x^*) = 0.5596$ into the expression for mean exit time, we obtain the function relation between the mean exit time and the noise intensity, and compare it with Monte Carlo results. As we see in Fig. 11, they are consistent with each other, which confirms the accuracy of the algorithm.

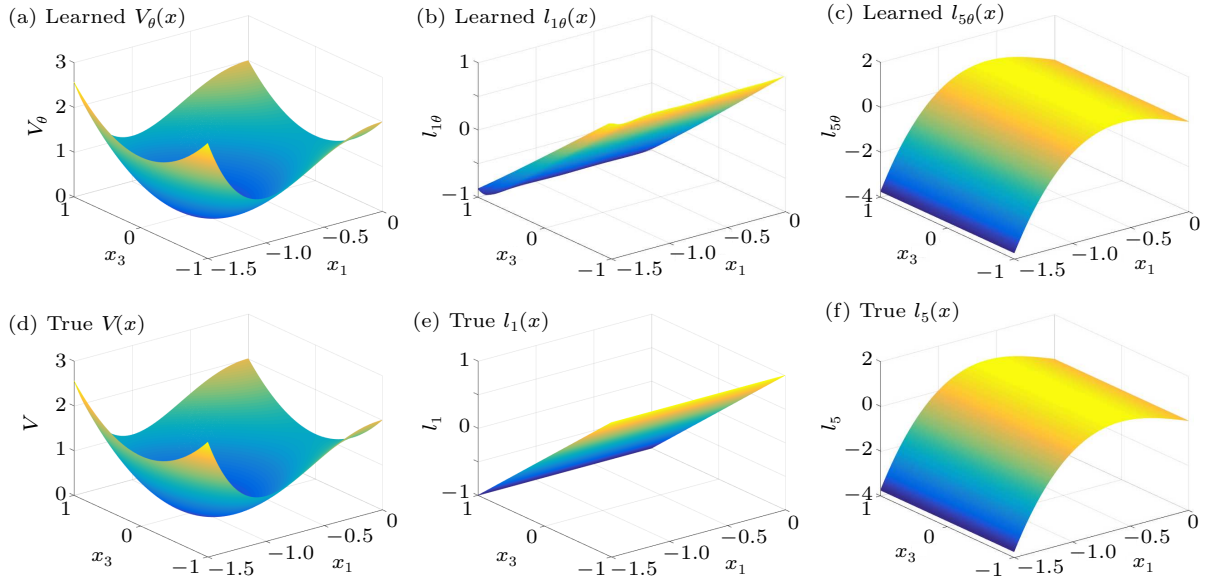


Fig. 9. Comparison between learned and true results of quasipotential $V(x)$ and rotational components $l_1(x)$ and $l_5(x)$ as functions of x_1 and x_3 in the plane $x_2 = 0$, $x_4 = 0$, $x_5 = 0$ for the attractive domain of the left fixed point SN1 $= (-1,0,0,0,0)^{\mathrm{T}}$.

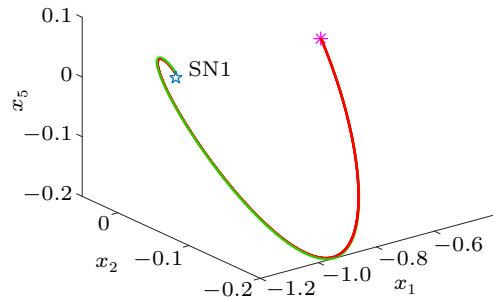


Fig. 10. Comparison between the most probable exit paths for Case A of the five-dimensional system computed by machine learning and the true system, denoted by red and green curves, respectively. The pink star indicates the exit point x^*.

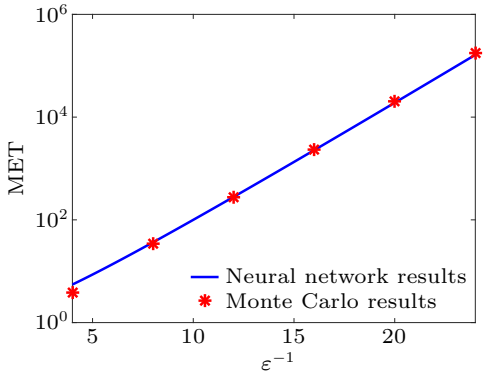


Fig. 11. Comparison between mean exit times versus reciprocal of noise intensity for Case A of the five-dimensional system via machine learning and Monte Carlo simulation, denoted by a blue curve and red star, respectively.

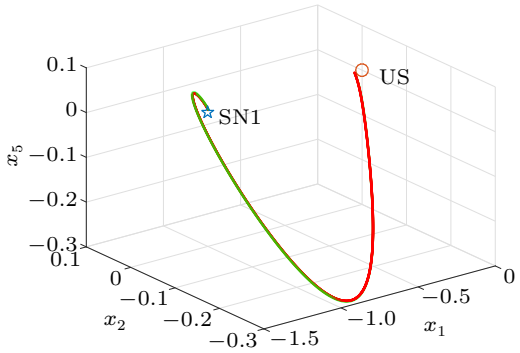


Fig. 12. Comparison between the most probable exit paths for Case B of the five-dimensional system computed by machine learning and the true system, denoted by red and green curves, respectively.

Case B Characteristic boundary.

Note that the hyperplane $x_1 = 0$, i.e., the stable manifold of the saddle point US, naturally serves as the characteristic boundary of the basin of attraction of SN1. According to the machine learning results for the quasipotential, we integrate the most probable path connecting SN1 and US. It is seen from Fig. 12 that the path almost coincides with that obtained with the true model, which verifies the effectiveness of the algorithm.

In addition, we estimate the prefactor $L_D^\varepsilon \sim 2.1828$ by utilizing the results of the most probable path, the rotational component $l_\theta(x)$ and the matrix $H^* = \nabla^2 V(\mathrm{US})$. It accords with the true result $L_D^\varepsilon \sim 2.2214$. Then the mean exit time can be approximated by the prefactor $L_D^\varepsilon \sim 2.1828$ and the quasipotential value $V_\theta(\mathrm{US}) = 0.9947$. The results computed by machine learning and Monte Carlo simulation are illustrated in Fig. 13. Note that the slight difference between the two results stems from the approximation error of the perturbation expression for mean exit time itself, since the machine learning results for quasipotential and prefactor are sufficiently accurate compared with the true values.

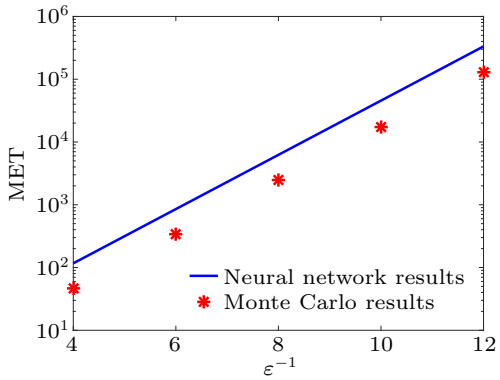


Fig. 13. Comparison between mean exit times versus reciprocal of noise intensity for Case B of the five-dimensional system via machine learning and Monte Carlo simulation, denoted by a blue curve and red star, respectively.

6. Conclusion and perspective

In this work we have developed a machine learning method to establish the large deviation prefactors of stochastic dynamical systems. In particular, we first introduced large deviation theory and the results of prefactors of mean exit time for non-characteristic and characteristic boundaries. Then we provided a new machine learning method to compute the prefactors based on orthogonal decomposition of the vector field. The successful application of the algorithm to two toy models illustrated its effectiveness in tackling mean exit time accurately.

Our neural networks and algorithm can be effectively extended to more sophisticated high-dimensional circumstances with computational complexity and models with physical significance. We use toy models to test our algorithm in this work since they have explicit quasipotential as ground truth to verify the learning results. They can also be generalized to deal with other types of rare events such as stationary probability distribution and escape probability. More importantly, our innovative approach may bring forth new ideas for improving the performance of searching for the most probable path by machine learning or data mining.

Data availability statement

The data that support the findings of this study are openly available in GitHub https://github.com/liyangnuaa/Computing-large-deviation-prefactors.

Acknowledgments

Project supported by the Natural Science Foundation of Jiangsu Province (Grant No. BK20220917) and the National Natural Science Foundation of China (Grant Nos. 12001213 and 12302035).

References

[1] Ma J, Xu Y, Li Y, Tian R, Ma S and Kurths J 2021 *Appl. Math. Mech.* **42** 65
[2] Zheng Y, Yang F, Duan J, Sun X, Fu L and Kurths J 2020 *Chaos* **30** 013132
[3] Scheffer M, Bascompte J, Brock W A, Brovkin V, Carpenter S R, Dakos V, Held H, Van Nes E H, Rietkerk M and Sugihara G 2009 *Nature* **461** 53
[4] Dykman M I, Mori E, Ross J and Hunt P 1994 *J. Chem. Phys.* **100** 5735
[5] Yuan S, Li Y and Zeng Z 2022 *Math. Model. Nat. Pheno.* **17** 34
[6] Yuan S, Zeng Z and Duan J 2021 *J. Stat. Mech. Theory E* **2021** 033204
[7] Zhu W and Wu Y 2003 *Nonlinear Dyn.* **32** 291
[8] Zhang Y, Jin Y, Xu P and Xiao S 2020 *Nonlinear Dyn.* **99** 879
[9] Freidlin M I and Wentzell A D 2012 *Random Perturbations of Dynamical Systems* (Berlin: Springer)
[10] Grafke T, Schäfer T and Vanden-Eijnden E 2021 arXiv: 2103.04837[cond-mat.stat-mech]
[11] Schorlepp T, Grafke T and Grauer R 2021 *J. Phys. A: Math. Theor.* **54** 235003
[12] Bouchet F and Reygner J 2022 *J. Stat. Phys.* **189** 21
[13] Schorlepp T, Grafke T and Grauer R 2023 *J. Stat. Phys.* **190** 50
[14] Naeh T, Klosek M, Matkowsky B and Schuss Z 1990 *SIAM J. Appl. Math.* **50** 595
[15] Matkowsky B, Schuss Z and Tier C 1983 *SIAM J. Appl. Math.* **43** 673
[16] Matkowsky B and Schuss Z 1982 *SIAM J. Appl. Math.* **42** 822
[17] Roy R V 1997 *Int. J. Nonlin. Mech.* **32** 173
[18] Maier R S and Stein D L 1997 *SIAM J. Appl. Math.* **57** 752
[19] Beri S, Mannella R, Luchinsky D G, Silchenko A and McClintock P V 2005 *Phys. Rev. E* **72** 036131
[20] Weinan E 2017 *Commun. Math. Stat.* **5** 1
[21] Li Y and Duan J 2021 *Physica D* **417** 132830
[22] Li Y and Duan J 2022 *J. Stat. Phys.* **186** 30
[23] Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S and Yang L 2021 *Nat. Rev. Phys.* **3** 422
[24] Rotskoff G and Vanden-Eijnden E 2018 *NIPS* **31** 7146
[25] Opper M 2019 *Annalen der Physik* **531** 1800233
[26] Li Y, Duan J and Liu X 2021 *Phys. Rev. E* **103** 012124
[27] Wei W, Gao T, Chen X and Duan J 2022 *Chaos* **32** 051102
[28] Li Y, Xu S, Duan J, Liu X and Chu Y 2022 *Nonlinear Dyn.* **109** 1877
[29] Lin B, Li Q and Ren W 2021 *PMLR* **145** 652
[30] Li Y, Yuan S and Xu S 2023 *Commun. Nonlinear Sci. Numer. Simul.* **126** 107425
[31] Xu Y, Zhang H, Li Y, Zhou K, Liu Q and Kurths J 2020 *Chaos* **30** 013133
[32] Bouchet F and Reygner J 2016 *Ann. Henri Poincaré* **17** 3499