Journal of Computational Physics 488 (2023) 112152

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Predicting rare events using neural networks and short-trajectory data

John Strahan [a], Justin Finkel [b,d], Aaron R. Dinner [a,b,*], Jonathan Weare [c,*]

[a] Department of Chemistry and James Franck Institute, the University of Chicago, Chicago, IL 60637, United States of America
[b] Committee on Computational and Applied Mathematics, the University of Chicago, Chicago, IL 60637, United States of America
[c] Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, United States of America
[d] Department of Earth, Atmospheric, and Planetary Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139, United States of America

ARTICLE INFO

Article history:
Received 1 August 2022
Received in revised form 8 February 2023
Accepted 14 April 2023
Available online 9 May 2023

Keywords:
Neural network
Rare event
Feynman-Kac equation
High-dimensional PDE
Adaptive sampling
Holton-Mass model

ABSTRACT

Estimating the likelihood, timing, and nature of events is a major goal of modeling stochastic dynamical systems. When the event is rare in comparison with the timescales of simulation and/or measurement needed to resolve the elemental dynamics, accurate prediction from direct observations becomes challenging. In such cases a more effective approach is to cast statistics of interest as solutions to Feynman-Kac equations (partial differential equations). Here, we develop an approach to solve Feynman-Kac equations by training neural networks on short-trajectory data. Our approach is based on a Markov approximation but otherwise avoids assumptions about the underlying model and dynamics. This makes it applicable to treating complex computational models and observational data. We illustrate the advantages of our method using a low-dimensional model that facilitates visualization, and this analysis motivates an adaptive sampling strategy that allows on-the-fly identification of and addition of data to regions important for predicting the statistics of interest. Finally, we demonstrate that we can compute accurate statistics for a 75-dimensional model of sudden stratospheric warming. This system provides a stringent test bed for our method.

1. Introduction

In many complex dynamical systems, behaviors of strong interest occur infrequently compared to the system's fastest timescale phenomena. For example, most climate-related destruction is due to extreme weather events (e.g., hurricanes, heat waves, flooding) [1–5]. More broadly, fluid turbulence in both natural and engineered systems produces intermittent, damaging extreme events [6]. In the molecular sciences, chemical reactions and molecular rearrangements occur on timescales many orders of magnitude longer than the timescale of individual bond vibrations [7,8]. In the biomedical sciences, it may take many mutations before a virulent strain of a pathogen emerges [9], or many heart beats before a cardiac arrhythmia become life-threatening [10,11].

Among the most common computational tasks related to these rare events is prediction—assessing the likelihood and extent of an event (i.e., the risk and cost in the case of a deleterious event)—before it occurs. When the event is not too rare,

* Corresponding authors.
E-mail addresses: dinner@uchicago.edu (A.R. Dinner), weare@nyu.edu (J. Weare).

https://doi.org/10.1016/j.jcp.2023.112152
0021-9991/

it can often be predicted with sufficient accuracy by direct forward-in-time integration of a computer model as is frequently done, for example, in weather prediction. However, when the event is very rare, direct forward-in-time integration becomes prohibitively expensive because many simulated model trajectories are required to observe even one instance of the event, leave alone compute accurate statistics. The computational cost increases further when the goal is to gain an understanding of how the rare event develops, which requires predictions generated from many initial conditions.

One common approach to this problem is to construct a "coarse-grained" model, in which some details of the system are treated implicitly [12–14]. One example is a Markov State Model (MSM), in which one groups the states of the full system into discrete sets and then evolves the system between these sets according to transition probabilities that are estimated from trajectories of the full system [15–18]. A variety of machine learning approaches that instead yield continuous coarse-grained representations of systems have come to be known as "equation discovery" [19–24]. When an accurate coarse-grained model can be constructed, it can be simulated extensively to make predictions with statistical confidence. However, building an accurate coarse-grained model can be challenging, in particular, because it is often not clear a priori which features must be included. The construction of coarse-grained models thus remains a subject of intense inquiry.

Here, we pursue an alternative approach: directly estimating conditional expectations of a Markov process as a function of initial condition. We term these conditional expectations "prediction functions." Prediction functions can be used to reveal how a rare event develops in remarkable detail. For example, the committor (also known as the splitting probability)—the probability that a process proceeds to a set of target states before a competing set of states—can be used to define the transition state ensemble of a molecular rearrangement [25,26], as well as the pathways that lead between the reactant and product states [27,28]. Prediction functions can also provide important information for decision making. For example, the committor can be used by energy, transportation, and financial sectors to measure risk due to extreme weather and allocate resources accordingly [29]. Committor estimation is a growing research focus in meteorology [30–33]. In real-time settings, the lead time—the expected time until onset of the event given that it occurs—is also essential to know [34,35,33].

Prediction functions satisfy Feynman-Kac equations, linear equations of the operator that describes the evolution of expectations of functions of a process, the transition operator (also known as the Koopman operator [36]) and its infinitesimal generator [37, Chapter 3]. Feynman-Kac equations cannot be solved by conventional discretization approaches because they involve a high-dimensional independent variable (the state of the underlying process). Moreover, the form of the transition operator is generally not known. Nonetheless, we showed recently that Feynman-Kac equations can be solved approximately by a basis expansion in which inner products of basis functions are estimated from a data set of short trajectories [27,38].

While this approach has been successfully applied to such diverse processes as protein folding [27], molecular dissociation [28], and sudden stratospheric warming [34], it relies on identifying an effective basis set. One choice is to use a basis of indicator functions for discrete sets, in which case the approach reduces to construction of an MSM (but with appropriate boundary conditions for the prediction function). However, just as it can be challenging to group states into sets that satisfy the Markov assumption in construction of an MSM [18,39], the choice of basis set is not always straightforward.

Here, we address this issue through a neural network ansatz for prediction functions. Our work builds on recent studies, which showed that a neural network ansatz can be used to solve for the committor if one assumes particular, explicit forms for the dynamical operator [40–43] (and see [44] for a closely related approach using tensor network approximation). Similar neural-network techniques have been devised to solve a wide variety of partial differential equations [45–50]. Because we work directly with a data set of short trajectories, our approach is free of restrictive assumptions about the dynamics (e.g., microscopic reversibility) and does not require explicit knowledge of a model generating the data, opening the door to treating high-fidelity models, and even experimental and observational data [33], without simplifying assumptions.

In Section 2, we review prediction functions and the Feynman-Kac equation that we need to solve to estimate them. In Section 3 we introduce our neural network approach to solving Feynman-Kac equations using a data set of paired trajectories. In Section 4, we compare with Galerkin methods and explore the role of the lag time and the distribution of trajectory initial conditions on performance. In Section 5 we introduce an adaptive sampling method that enriches the data set based on the current neural network approximation. Finally, in Section 6 we apply our algorithm to estimating the probability of onset and the lead time of a sudden stratospheric warming event.

2. Prediction functions and their Feynman-Kac equations

We consider events defined by a set of target states B; often, there is also a competing set of states A. For example, if we want to estimate the probability that a moderate storm develops into an intense hurricane before dissipating, we would take B to include all weather states consistent with an intense hurricane and A to include all quiescent states. The initial moderate storm would be a state in the domain $D = (A \cup B)^c$. Mathematically, we select states in B with the indicator function

$$\mathbb{1}_B(x) = \begin{cases} 1, & x \in B \\ 0, & x \notin B, \end{cases} \tag{1}$$

where x denotes a particular state of the system. We define analogous indicator functions for other sets.

We assume the dynamics of the system can be described by a Markov process X_t. In the example above, $X_0 = x$ is a moderate storm state, and the probability that it develops into an intense hurricane before the weather returns to a quiescent state is the committor:

$$q(x) = \mathbb{P}_x[X_T \in B] = \mathbb{E}_x[\mathbb{1}_B(X_T)], \tag{2}$$

where the subscripted x indicates the initial condition $X_0 = x$, and $T = \min\{t > 0 : X_t \in A \cup B\}$ is the stopping time, i.e., when the process leaves the domain $D = (A \cup B)^c$.

Continuing the example above, we may also want to compute the lead time, i.e., the average time until a moderate storm develops into an intense hurricane, given that the intense hurricane occurs (B occurs before A). The lead time tells us how much time we have to prepare for the worst case; by definition, it is shorter than the average time until an intense hurricane develops, which can be misleadingly large if the storm has a high probability of dissipating (A occurs before B). Mathematically, the lead time is

$$m_{AB}(x) = \frac{\mathbb{E}_x[T\,\mathbb{1}_B(X_T)]}{\mathbb{E}_x[\mathbb{1}_B(X_T)]}. \tag{3}$$

When the event of interest is rare, computing $q(x)$ or $m_{AB}(x)$ by direct forward-in-time simulation is difficult. It involves repeatedly simulating X_t starting in a selected initial condition x and running until either A or B is reached (which defines the stopping time T), and then assembling a sample average. This approach has significant drawbacks: first, when the time T is very large, generation of a single sample trajectory may be prohibitively computationally expensive, and second, when $q(x)$ is small, many sample trajectories will be required to observe a single trajectory reaching B. For example, starting from a typical weather state, the expected time to the next extreme event may be years, and the probability that it occurs on a much shorter time scale may be very small.

In this paper we estimate prediction functions by solving operator equations for them approximately. In the case of the committor, the operator equation takes the form

$$(\mathcal{T}^{\tau}_{D^c} - \mathcal{I})[q](x) = 0 \text{ with } q(x) = 0 \text{ for } x \in A \text{ and } q(x) = 1 \text{ for } x \in B, \tag{4}$$

where τ is a time interval known as the lag time and $\mathcal{I}$ is the identity operator. Here we focus on finite τ; the case of infinitesimal τ is discussed in Section 4.4. Above, the stopped transition operator $\mathcal{T}^{s}_{D^c}$ encodes the full dynamics of the system when it is in D; it is defined by its action on an arbitrary test function f:

$$\mathcal{T}^{\tau}_{D^c}[f](x) = \mathbb{E}_x\left[f(X_{\tau\wedge T})\right], \tag{5}$$

where $\tau \wedge T = \min\{\tau, T\}$. Physically, (4) reflects the fact that the average probability that B occurs before A after time τ over all trajectories emanating from $X_0 = x$ is the same as the probability that B occurs before A starting from x. Similarly, the lead time satisfies

$$(\mathcal{T}^{\tau}_{D^c} - \mathcal{I})[m_{AB}q](x) = -\mathbb{E}_x\left[\int_0^{\tau\wedge T} q(X_s)ds\right] \text{ with } m_{AB}q = 0 \text{ for } x \in A \cup B. \tag{6}$$

In this case, the right hand side accumulates the likelihood of reaching B before A until $A \cup B$ is reached or time τ, whichever occurs first.

Eqs. (4) and (6) are examples of Feynman-Kac equations [37, Chapter 3], which can take more general forms, such as

$$(\mathcal{T}^{\tau}_{D^c} - \mathcal{I})[u](x) = -\mathbb{E}_x\left[\int_0^{\tau\wedge T} h(X_s)ds\right] \text{ with } u(x) = g(x) \text{ for } x \notin D \tag{7}$$

which is solved by the prediction function

$$u(x) = \mathbb{E}_x\left[g(X_T) + \int_0^{T} h(X_s)ds\right]. \tag{8}$$

We recover (4) by setting $h(x) = 0$ and $g(x) = \mathbb{1}_B(x)$ and (6) by setting $h(x) = q(x)$ and $g(x) = 0$; the latter case yields $[m_{AB}q](x)$, and we must solve separately for $q(x)$ and divide by it to obtain $m_{AB}(x)$. Crucially, (7) exactly characterizes u for any choice of $\tau > 0$. In particular, τ can be chosen much shorter than typical values of T.

On its own, (7) brings us no closer to a practically viable approximation of the prediction function. The independent variable x is typically high-dimensional, rendering useless any standard discretization approach to solving (7) for u. Instead, the current state-of-the-art approach involves expansion of u in a problem-dependent basis [27,28,38]. In the next section, we explore a potentially more flexible and automated approach to solving (7).

3. Solving Feynman-Kac equations with neural networks

The goal of the present study is to solve (7) by approximating u by a neural network u_θ with a vector of parameters θ. Specifically, we seek $\theta = \theta^*$ that minimizes a mean square difference between the left and right hand sides of the Feynman-Kac equation and boundary condition in (7):

$$\theta^* = \arg\min_\theta [C_{\mathrm{FKE}} + \lambda C_{\mathrm{BC}}] \tag{9}$$

with

$$C_{\mathrm{FKE}} = \left\| \left((\mathcal{T}^\tau_{D^c} - \mathcal{I}) u_\theta + \mathbb{E}_x \left[\int_0^{\tau \wedge T} h(X_s) ds \right] \right) \mathbb{1}_D \right\|^2_\mu \text{ and } C_{\mathrm{BC}} = \| (u_\theta - g) \mathbb{1}_{D^c} \|^2_\mu . \tag{10}$$

The norm that we use is the μ-weighted L^2 norm $\|f\|^2_\mu = \int |f(x)|^2 \mu(dx)$, where μ is the sampling distribution. Importantly, unlike many existing estimators [40–44,51–55], our data need not be generated from (or re-weighted according to) the invariant distribution of X_t, a feature that we exploit in Section 4.5. In (10), C_{FKE} and C_{BC} are both zero when u_θ equals the desired prediction function. The parameter λ controls the strength of the first norm, which enforces the Feynman-Kac equation, relative to the second norm, which enforces the boundary condition. Smaller values enforce the boundary conditions more strictly but can compromise the satisfaction of the Feynman-Kac equation. For our numerical tests below, we tuned λ by trial and error to the smallest value that still enforced the boundary conditions to the desired precision.

The gradient of C_{FKE} includes the integral of a product of two terms of the form $\mathcal{T}^\tau_{D^c} v$ with $v = u_\theta$ and $v = \partial_\theta u_\theta$, the gradient of u_θ with respect to the parameters θ. While we cannot hope to evaluate $\mathcal{T}^\tau_{D^c} v$ exactly for any non-trivial v, as long as we can evaluate v we have access to the random variable $v(X_{\tau\wedge T})$ whose expectation is $\mathcal{T}^\tau_{D^c} v$. With only one sample of $X_{\tau\wedge T}$ for each sample of X_0, we would not be able to build an unbiased estimate of the product of two terms of the form $\mathcal{T}^\tau_{D^c} v$. One approach, common in reinforcement learning applications, is to simply drop the term involving this product from the gradient [56]. However, given at least two independent samples of $X_{\tau\wedge T}$ for each sample of X_0, we can construct an unbiased estimator of the full gradient of C_{FKE} that converges to the exact gradient of C_{FKE} in the limit of many samples of X_0 (even when the number of independent samples of $X_{\tau\wedge T}$ for each sample of X_0 does not increase). Below we outline a procedure that constructs an unbiased estimate of the gradient of C_{FKE} given a data set of samples of X_0, together with $\ell \geq 2$ samples of $X_{\tau\wedge T}$ for each sample of X_0 (in tests of $2 \leq \ell \leq 10$, we found the results to be insensitive to the choice of ℓ, and we use $\ell = 2$ throughout).

Our procedure is as follows.

1. Select a set of n initial conditions X^i_0 from the sampling distribution μ.
2. From each X^i_0, launch ℓ independent unbiased simulations to generate trajectories $\{(X^i_0, X^{i,j}_\Delta, \ldots, X^{i,j}_{S\Delta})\}_{j=1}^\ell$. Here we assume that $\tau = S\Delta$.
3. For trajectory $j = 1, 2, \ldots, \ell$ with initial condition X^i_0, determine the index of its stopping time as $k^{i,j} = \min\{s' : X^{i,j}_{s'\Delta} \in (A \cup B) \text{ or } s' = S\}$.
4. Given the data set of grouped trajectories, approximate the first norm in (9) as

$$\begin{aligned} \bar{C}_{\mathrm{FKE}} = \frac{1}{n} \sum_{i=1}^n & \left(\frac{1}{|S_i|} \sum_{j \in S_i} u_\theta(X^{i,j}_{k^{i,j}\Delta}) - u_\theta(X^i_0) + \Delta \sum_{s=0}^{k^{i,j}-1} h(X^{i,j}_{s\Delta}) \right) \\ & \times \left(\frac{1}{|S'_i|} \sum_{j' \in S'_i} u_\theta(X^{i,j'}_{k^{i,j'}\Delta}) - u_\theta(X^i_0) + \Delta \sum_{s=0}^{k^{i,j'}-1} h(X^{i,j'}_{s\Delta}) \right) \mathbb{1}_D(X^i_0) \end{aligned} \tag{11}$$

   and the second norm in (9) as

$$\bar{C}_{\mathrm{BC}} = \frac{1}{n} \sum_{i=1}^n \left(u_\theta(X^i_0) - g(X^i_0) \right)^2 \mathbb{1}_{D^c}(X^i_0), \tag{12}$$

   where S_i and S'_i are randomly chosen index sets such that $S_i \cap S'_i = \emptyset$.
5. Compute the total approximate loss function as

$$\bar{C} = \bar{C}_{\mathrm{FKE}} + \lambda \bar{C}_{\mathrm{BC}}, \tag{13}$$

   which converges to the loss in (9) as n increases.
6. Adjust the parameters to minimize (13).

7. Check termination criteria and stop if met (discussed further below).
8. If adaptively sampling, apply the procedure in Section 5 and set n to the total number of initial conditions.
9. Go to step 3.

In principle, the loss can be minimized over any sufficiently flexible ansatz u_θ. In this work, u_θ is a fully connected feed-forward neural network, and we determine the optimal parameters via the Adam algorithm [57]. In the present study, we stop training (step 7) when the average loss for an epoch is less than zero. It is possible for $\bar{C}_{\mathrm{FKE}}$ to become negative because the two parenthetical factors in (11) are evaluated using independent samples of $X_{\tau\wedge T}$. While this could be avoided by choosing $S_i = S_i'$, the result would be a biased estimator of C_{FKE}. In the limit of large n, $\bar{C}_{\mathrm{FKE}}$ converges to C_{FKE}, which must be non-negative. When using any sample approximation of C_{FKE}, some regularization is required to avoid overfitting. We find early stopping at the first occurrence of a negative value of $\bar{C}_{\mathrm{FKE}}$ to be a natural and effective approach. Further details are given in conjunction with the numerical examples.

4. Illustration of numerical considerations

In this section, we use a model for which we are able to compute reference results to illustrate the advantages of our approach relative to existing ones. Specifically, we compare our approach with one that employs a basis expansion (a Markov State Model) and one that employs a neural network with an assumed form for the dynamical operator. Finally, we examine common choices for the sampling distribution. We show that an important practical advantage of our approach is the freedom to choose the sampling distribution μ with which to weight the norm in C_{FKE}.

4.1. Müller-Brown model

The system that we study is specified by the Müller-Brown potential [58], which is a sum of four Gaussian functions:

$$V_{\mathrm{MB}}(y,z) = \frac{1}{20}\sum_{i=1}^{4} C_i \exp[a_i(y-z_i)^2 + b_i(y-y_i)(z-z_i) + c_i(z-z_i)^2]. \tag{14}$$

For all results shown, we use $C_i = \{-200, -100, -170, 15\}$, $a_i = \{-1, -1, -6.5, 0.7\}$, $b_i = \{0, 0, 11, 0.6\}$, $c_i = \{-10, -10, -6.5, 0.7\}$, $y_i = \{1, -0.27, -0.5, -1\}$, $z_i = \{0, 0.5, 1.5, 1\}$. The potential is shown in Fig. 1(left).

We consider the overdamped Langevin dynamics associated with V_{MB}, discretized with the BAOAB algorithm [59]:

$$X_{t+dt} = X_t - \nabla V(X_t)dt + \sqrt{\frac{dt}{2\beta}}(Z_t + Z_{t-dt}) \tag{15}$$

where dt is the time step, β is the inverse temperature, $Z_t \sim N(0,1)$, $N(0,1)$ is the normal distribution with zero mean and unit standard deviation (i.e., Z_t is Gaussian noise), and $V = V_{\mathrm{MB}}$ with $\beta = 1$ unless otherwise specified. In practice, we use a time step of $dt = 0.001$, saving the configuration every time step, such that $\Delta = 0.001$ (cf. step 2 in Section 3). When the parameter β is large, X_t makes only very rare transitions between the local minima of V_{MB}.

We define states A and B as

$$\begin{aligned} A &= \{y,z : 6.5(y+0.5)^2 - 11(y+0.5)(z-1.5) + 6.5(z-1.5)^2 < 0.3\} \\ B &= \{y,z : (y-0.6)^2 + 5(z-0.02)^2 < 0.2\}, \end{aligned} \tag{16}$$

neighborhoods of two of the three local minima of V_{MB} (Fig. 1(left)).

The Müller-Brown model described above is commonly employed as a simple illustration of the features of molecular rearrangements [58,38,41–43]. The presence of local minima in addition to A and B, and the fact that, at low noise, the trajectories connecting the minima do not align with the coordinate axes are both features that can be challenging for algorithms that enhance the sampling of transitions between the reactant (A) and product (B) states. In our tests, we specifically focus on the committor. We compute a reference committor by the finite difference scheme outlined in the appendices of [38,60] with $\epsilon = 0.0125$. In all tests, we compare the estimated committor to the reference committor computed using the same potential energy function used to generate the data.

Finally, to represent the fact that one of the most challenging aspects of treating complex systems is that the manifold on which the dynamics take place is generally not known, we transform the trajectories and sets A and B to a new set of coordinates. Specifically, we map the two-dimensional system onto a Swiss roll (Fig. 1(right)):

$$\begin{bmatrix} y \\ z \end{bmatrix} \to \begin{bmatrix} (c+y)\cos((y+c)f) \\ z \\ (c+y)\sin((y+c)f) \end{bmatrix}, \tag{17}$$

where the parameter f controls how tightly the roll is wound, and c is an offset to ensure that the range of x is positive. Unless otherwise specified, we use $f = 3$ and $c = 1.8$. For the remainder of the tests based on the Müller-Brown potential,

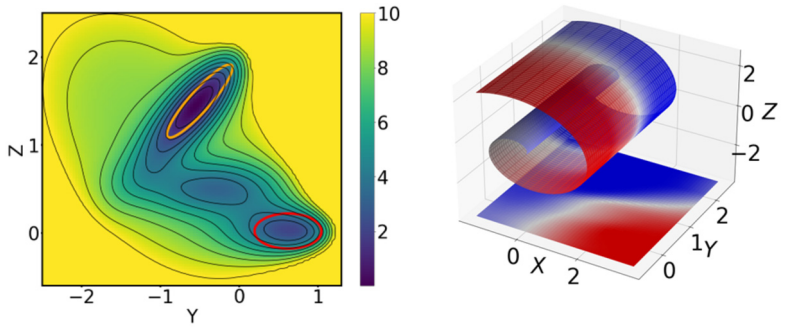

Fig. 1. The system used for the numerical experiments in Section 4. (left) Müller-Brown potential [58]. Sets A and B are marked by the orange and red ellipses, respectively, and contours are spaced at intervals of 1 in the units of (14). (right) Reference committor for the Müller-Brown dynamics mapped to the Swiss roll, and below on the two-dimensional surface. We compute the reference from a finite difference scheme [38,60] in two dimensions and then map it to the Swiss roll using (17). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

we use the three-dimensional coordinates as input features for all neural networks and k-means clustering. For clarity of visualization, we plot the estimated committors on the original two-dimensional coordinates. The error metric that we use is independent of coordinate system.

Unless otherwise specified, for our experiments with the Müller-Brown model below, we draw 30,000 initial conditions uniformly from the region:

$$\Omega = \{y, z : -1.5 < y < 1.0, -0.5 < z < 2.5, V(y, z) < 100\}. \tag{18}$$

Two independent trajectories of length τ (to be specified below) are then generated from each initial condition using (15).

4.2. Neural network details

For all the numerical experiments involving the Müller-Brown potential, we use fully connected feed-forward neural networks with three inputs, three hidden layers, each consisting of 30 sigmoid activation functions, and an output layer with a single sigmoid activation function. In all trials with fixed data sets, we trained for a maximum of 3000 epochs with a learning rate of 0.0005 and a batch size of 1500. Each epoch proceeds by drawing a permutation of the data set, then one step of Adam is performed using mini-batches of size 1500 (that is, 1500 pairs of trajectories) such that each trajectory pair is used exactly once per epoch (that is, the number of Adam steps is the data set size divided by the mini-batch size). The boundary term is computed with the same mini-batch as $\bar{C}_{\mathrm{FKE}}$; we use $\lambda = 1$ to weight the terms in the loss function. We also explored deeper networks with ReLU activation functions, and they performed comparably and generally required shorter training times (results not shown); we focus on the shallower networks with sigmoid activation functions because they allow a direct comparison with loss functions involving explicit derivatives of u_θ in Section 4.4.

4.3. Galerkin methods

As discussed in the Introduction, one of our main motivations in introducing an approach based on neural networks is that it can be difficult to identify basis functions for linear (e.g., Finite Element or other Galerkin) methods for solving Feynman-Kac equations. To illustrate this issue explicitly, we compare estimates for the committor from our approach with those obtained from dynamical Galerkin approximation [38,27] using a basis of indicator functions, which can be considered an MSM [38]. We do so as a function of the parameter f in (17) and generate data sets with $0 \le f \le 10$.

To construct an MSM, we clustered the configurations in each data set by the k-means algorithm (with k as specified below) applied to the three-dimensional coordinates of the model. The indicator functions of the set of points closest to each cluster centroid form a basis for a Galerkin approximation of the committor function. A matrix $\mathbf{T}$ of transition probabilities between clusters was constructed by counting transitions of the stopped process in our trajectory data set with a lag time of $\tau = 150\Delta$. Here we use the convention that the row and column indices are zero for A and their maximum values for B. The committor is then computed from $\mathbf{T}\mathbf{q}_+ = \mathbf{q}_+$ with the last component of the solution vector set to 1. The neural network and its training were as described in Section 4.2.

Fig. 2 shows the results. We see that as the roll is wound tighter (higher f), the MSM estimates, constructed with a constant 300 clusters, decrease in accuracy, while the network estimates remain consistently good. In the right panel, we vary the number of clusters and report the number required to reach a root mean squared error threshold of 0.045. This threshold is chosen because it results in numbers of clusters in a range that is typical in MSM studies [27,34]. We increase the number of trajectories in proportion to the number of MSM clusters to ensure that each cluster is sampled a consistent amount. In this test, we see that large numbers of MSM clusters, and hence large amounts of data, are needed. Intuitively, the MSM encounters problems when a single cluster spans adjacent layers. Therefore, it is necessary to vary the size of the

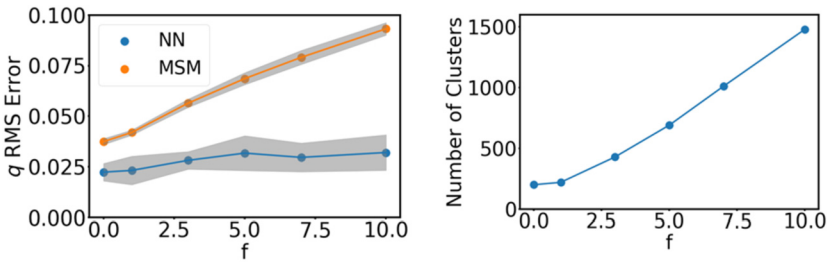

Fig. 2. Comparison with Galerkin methods. (left) For an MSM estimate with $k = 300$ clusters and neural network (NN) as described in the text, the root mean square (RMS) error in the committor for the Müller-Brown model as the Swiss roll is wound tighter (higher f in (17)). Shading shows the standard deviation in the error from training on ten independent data sets. (right) Number of MSM clusters needed to achieve an RMS error in the committor of less than 0.045.

clusters with the distance between layers of the Swiss roll, which is a linear function of $1/f$. Consistent with this idea, in Fig. 2 we find an approximately linear dependence of the number of clusters needed to achieve a certain error threshold.

We note that in practice MSMs are often constructed on coordinates obtained from a method for dimensionality reduction and/or manifold learning. With such pre-processing, linear methods can clearly be successful. However, kernel-based methods for dimensionality reduction (e.g., diffusion maps [61] or kernel time-lagged independent component analysis [62,63]) scale poorly with the size of the data set. A neural network (e.g., an autoencoder [64,65]) can be used for dimensionality reduction, but the approach presented here is simpler in that we go directly from model coordinates to prediction function estimates.

4.4. Lag time

As discussed in the Introduction, neural networks have been applied to estimating high-dimensional committors assuming a partial-differential form for the dynamical operator [40–42]. This form arises in the limit that one considers an infinitesimal lag time. In this case, one can write (7) as

$$\mathcal{L}[u](x) = h(x) \text{ for } x \in D \text{ and } u(x) = g(x) \text{ for } x \notin D, \tag{19}$$

where $\mathcal{L}$ is the infinitesimal generator:

$$\mathcal{L}[f](x) = \lim_{dt \to 0} \frac{\mathbb{E}_x[f(X_{dt})] - f(x)}{dt}. \tag{20}$$

For a diffusion process, $\mathcal{L}$ takes the form

$$\mathcal{L}f(x) = \sum_{i=1}^{\kappa} b_i(x) \frac{\partial f}{\partial x_i} + \frac{1}{2} \sum_{i,j=1}^{\kappa} (\sigma\sigma^{\mathrm{T}})_{ij}(x) \frac{\partial^2 f}{\partial x_i \partial x_j}, \tag{21}$$

where $b \in \mathbb{R}^{\kappa}$ and $\sigma \in \mathbb{R}^{\kappa \times \kappa}$ are the drift and diffusion coefficients that determine the evolution of X_t. In the limit of small dt, the dynamics in (15) correspond to a generator with $b = -\nabla V$ and $\sigma = \sqrt{2/\beta}\, I$. In this case, the loss function becomes

$$\bar{C}_{\mathrm{FKE}} = \sum_{i=1}^{n} \left(\sum_{j=1}^{\kappa} b_j(X_0^i) \frac{\partial u_\theta(X_0^i)}{\partial x_j} + \frac{1}{2} \sum_{j,l=1}^{\kappa} (\sigma\sigma^{\mathrm{T}})_{jl}(X_0^i) \frac{\partial^2 u_\theta(X_0^i)}{\partial x_j \partial x_l} - h(X_0^i) \right)^2 \mathbb{1}_D(X_0^i) \tag{22}$$

with an appropriate boundary condition term.

The loss function in (22) differs from the one used in many recent articles on the subject of committor estimation with neural-network (or recently tensor-network) approximations [40–44]. Those papers focus specifically on the case of reversible overdamped diffusive dynamics. In this case the committor can be found by minimizing a sample approximation of the loss function $\|\nabla q\|_\pi^2$ (for constant, isotropic diffusion coefficient) where π is the invariant distribution of the dynamics [55]. Relatedly, despite a resemblance to (10), the estimator $\|(q_+(X_\tau) - q_+(X_0))\|_\pi^2$ that appears in [51–54] is, in fact, a small τ approximation of $\|\nabla q\|_\pi^2$.

We stress that (22) is only appropriate for diffusion processes and requires working with the full set of variables in which the dynamics are formulated. Importantly, one generally analyzes only functions of a subset of the variables (termed collective variables or order parameters) [27,28,33,38]. For example, in a molecular simulation of a solute in solvent, one may include only the dihedral angles of the solute. In a weather simulation, one may focus on the wind speed and geopotential height at particular altitudes. When working with observational data, one only has access to the features that were measured. Even when the tracked variables can be described by an accurate coarse-grained model, that model is not known

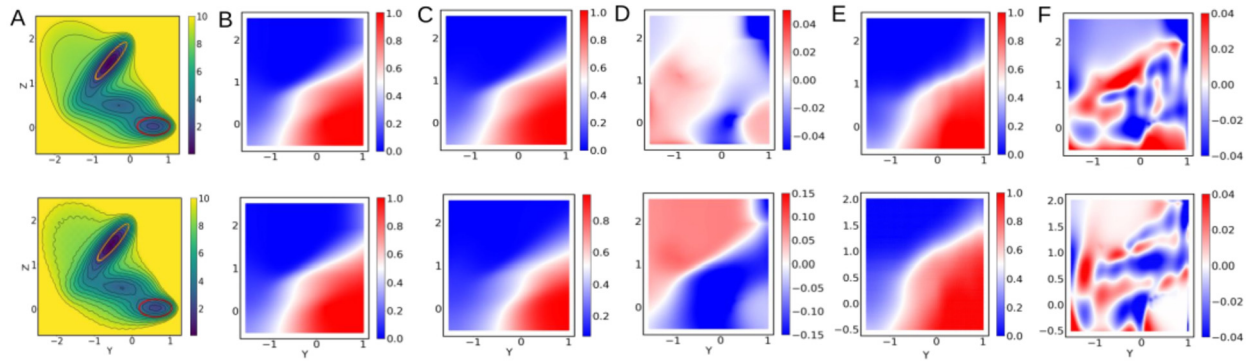

Fig. 3. Effect of potential roughness on the performance when using a loss function based on the infinitesimal lag time limit. Results shown are obtained with (22) and (23) with $\omega = 0$ (top row), and $\omega = 10$ (bottom row). (A) The potentials. (B) Reference committors obtained from the finite-difference scheme in [38,60]. (C) Neural network prediction of the committors. (D) Differences between the references and the predictions. (E,F) Same as columns C and D, except for a lag time of 100 steps. Note the different scales on the difference maps in columns D and F.

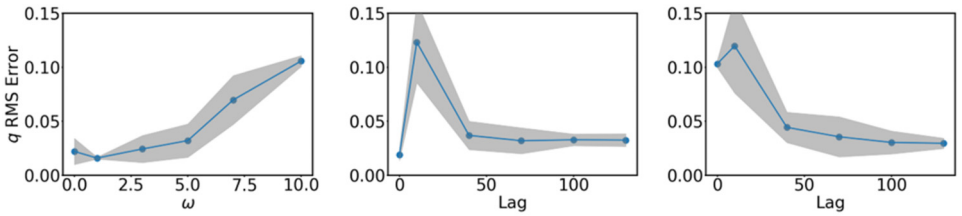

Fig. 4. Comparison of infinitesimal and finite lag time loss functions. (left) RMS error in the committor for the Müller-Brown dynamics mapped to the Swiss roll obtained with the infinitesimal lag time loss function in (22) as the frequency of the sinusoidal perturbation, ω, is increased. The other panels show the error as the lag time is increased with the frequency fixed at $\omega = 0$ (center) or $\omega = 10$ (right). Shading shows the standard deviation in the error from training on ten independent data sets.

explicitly and is difficult to identify from data. These considerations make minimization of any loss function explicitly involving (21) impossible for many practical applications.

Nonetheless, the loss function in (22) is appealing because it involves only a single time point, so no trajectories need to be generated if explicit forms for the drift and diffusion coefficients are known. While this would appear to be an advantage, we show in this section that, even when the dynamics can be reasonably described by (21), it can be preferable to work with finite lag times.

To make this point, we consider dynamics governed by the Müller-Brown potential with a small oscillating perturbation (Fig. 3A):

$$V(x) = V_{\mathrm{MB}}(x) + 0.1 \sin(2\pi\omega x) \sin(2\pi\omega y), \tag{23}$$

where ω controls the spatial frequency of the perturbation. Again we represent the data on the Swiss roll as described in Section 4.1. As shown in Fig. 3B, the perturbation is sufficiently small that it makes no qualitative change to the committor.

Given this data set, we train neural networks to minimize the loss function in (13), using either (11) or (22) for $\bar{C}_{\mathrm{FKE}}$, with $h(x) = 0$ and $g(x) = \mathbb{1}_B(x)$, corresponding to the committor. The network architecture was the same as above: i.e., fully connected feed forward with two inputs, 30 activation functions per hidden layer, and one output. The neural network and its training were as described in Section 4.2.

Typical results are shown in Fig. 3C and D, and the error in the committor is quantified in Fig. 4. As the frequency of the perturbation increases, the drift becomes large, with rapid sign changes, and the training of the infinitesimal lag time network tends to get stuck at poor estimates of the committor (Fig. 4(left)). By contrast, finite lag time networks consistently achieve low errors at longer lag times (Fig. 4(center and right)). This presumably results from averaging over values of the drift. Interestingly, we found that when the potential is smooth (Fig. 4(center)), slightly lower errors can be obtained using the zero lag time approach. However, in the presence of even such a small amount of roughness that the committor is qualitatively unchanged (Fig. 3A and B), our finite lag time approach performs better (Fig. 4(right)). We expect the latter case to be more relevant in many practical applications.

It may be tempting to assume that the zero lag time approach has lower computational cost since there is no need to actually simulate the stochastic differential equation (here, (15)). This is not necessarily the case. With the infinitesimal lag time loss function, the drift needs to be evaluated for each data point for every pass over the data set (one epoch). By contrast, the finite lag time loss function introduced here does not require evaluation of the drift once the data set is generated. Therefore, if the number of epochs needed to train the zero lag time network is comparable to the number of time steps used to generate the data set for the two-trajectory method, the finite lag time method will be less computationally costly.

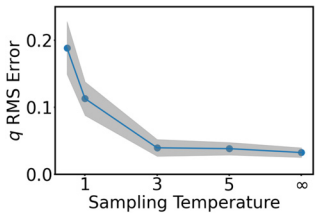

Fig. 5. RMS error of the committor as the sampling temperature, $1/\beta_s$, is increased. The point at $1/\beta_s = 1$ corresponds to the stationary distribution for the Müller-Brown model (in the small dt limit) at the temperature of the dynamics. Increasing the temperature makes the distribution more uniform. The point labeled ∞ is uniform. Shading shows the standard deviation in the error from training on ten independent data sets.

4.5. Sampling distribution

In this section, we investigate the role of the choice of sampling distribution. Following generation of a data set as described in Section 4.1, we selected initial points from the region specified in (18) with weight $\mu(x) \propto \exp(-\beta_s V(x))$ (β_s need not be the same as β) and trained over a range of β_s values. To a good approximation when dt is small, the invariant density of (15) is μ with $\beta_s = \beta$. When β_s is large, the data set of initial conditions concentrates at the local minima of $V(x)$. As β_s becomes small, the distribution approaches uniform. While this parametric form for the sampling distribution is convenient for the tests performed in this section, we emphasize that, unlike many existing schemes [40–42,53,54], our algorithm does not require explicit knowledge of the invariant density.

We trained ten networks on each pair of sampling distribution and lag time following the procedure in Section 4.2, and the resulting errors and their standard deviations are plotted as a function of $1/\beta_s$ in Fig. 5. At low $1/\beta_s$, which concentrates the initial points in the minima, the network is unable to find a good solution at any lag time. As $1/\beta_s$ increases and the distribution becomes more uniform, the solution improves significantly. This suggests that it is important to have the regions between minima well-represented in the data set, which is consistent with previous observations [66,38,27,34,43]. In high-dimensional examples, sampling the transition regions is not straightforward, and we present a solution to this problem in the next section.

5. Adaptive sampling

As we showed in Section 4.5, the choice of sampling distribution is important. In this section, we propose a simple method for adding data as the training proceeds. Since the approach depends on constructing a spatial grid we must first select a low-dimensional (e.g., two-dimensional) set of (possibly non-linear) coordinates $\xi(x)$ which, as noted above, we term collective variables. We then partition the space of possible ξ values into bins of equal volume labeled $S_1, S_2, \ldots, S_m$, and estimate

$$P_\eta = \frac{\int \left((\mathcal{T}_{D^c}^{\tau} - \mathcal{I}) u_\theta + \mathbb{E}_x \left[\int_0^{\tau \wedge T} h(X_s) ds \right] \right)^2 \mathbb{1}_{S_\eta}(\xi(x)) \mu(x) dx}{\int \mathbb{1}_{S_\eta}(\xi(x)) \mu(x) dx} \tag{24}$$

for each bin. The weights P_η are then used to select bins, and new initial points are then sampled from the selected bins with uniform probability. The essential idea is that we add data to the regions (bins) that contribute most to C_{FKE}.

When adaptively sampling to learn the committor we approximate (24) by

$$\hat{P}_\eta = \frac{\sum_{i=1}^{n} \left(\frac{1}{|S_i|} \sum_{j \in S_i} (u_\theta(X_{k^{i,j}}^{i,j}) - u_\theta(X_0^{i,j})) \right) \left(\frac{1}{|S_i'|} \sum_{j \in S_i'} (u_\theta(X_{k^{i,j}}^{i,j}) - u_\theta(X_0^{i,j})) \right) \mathbb{1}_{S_\eta}(\xi(X_0^{i,1}))}{\sum_{i=1}^{n} \mathbb{1}_{S_\eta}(\xi(X_0^{i,1}))}. \tag{25}$$

We compute (25) for each bin, select N bins with probability proportional to $\hat{P}_\eta$ with replacement, and sample a single additional initial point from each selected bin. From each of the N new initial points we generate ℓ independent trajectories. In practice, we observed that (25) can become negative for some bins in the same way that $\bar{C}_{\mathrm{FKE}}$ can become negative. In this case, we set all negative probabilities to zero.

The success of our adaptive sampling approach depends on the choice of ξ. In the absence of other knowledge, a reasonable choice is the current estimate of the committor function itself. We adopt this choice to test our adaptive sampling procedure on the Müller-Brown model. A related adaptive sampling approach using stratified sampling [67,68] based on a current committor estimate is proposed in [43].

The simulation and Swiss roll parameters, as well as neural network and training parameters are the same as above. We initially train with 10,000 pairs of trajectories drawn uniformly from the region in (18) for 1000 epochs. Then we alternate between adding $N = 5000$ new pairs of trajectories and training for 500 epochs, for four cycles. We compare to 30,000 trajectory pairs drawn uniformly from the region in (18). Results are presented in Fig. 6. We find that the adaptive sampling and uniform sampling perform similarly at long lag times, although the adaptive procedure gives more reproducible results

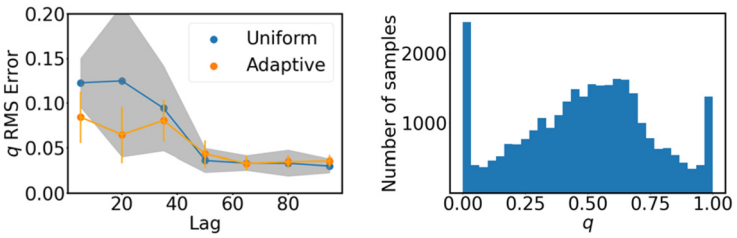

Fig. 6. Adaptive sampling scheme applied to the Müller-Brown dynamics mapped to the Swiss roll. (left) Comparison of uniform and adaptive sampling. Shading and error bars show the standard deviation in the error from training on ten independent data sets. (right) Histogram of the final data set as a function of the committor from training a neural network with a lag time of 100.

as shown by the smaller error bars. At short lag times the average error is lower as well. The adaptive sampling procedure concentrates sampling in the transition region, that is, near $q_+ = 0.5$. In the next section, we demonstrate the adaptive sampling procedure on our atmospheric model, and we again see that sampling is effectively directed to the transition region. For low noise diffusions, the transition region becomes narrower, and this is reflected by a sharper peak than in Fig. 6. In our testing, our adaptive sampling scheme remains effective, although more data are required at lower noise. We find that our method works for barriers $< 10/\beta$.

6. Predicting an atmospheric transition

As a demanding test of our method, we compute the committor and lead time for a model of sudden stratospheric warming (SSW), aiming to improve upon the benchmarks computed in [34]. Like other models of geophysical flows, the dynamics are irreversible and the stationary distribution is unknown. As a consequence, many competing approaches for computing the committor (e.g., [40–42,53,54]) are not applicable.

Typical winter conditions in the Northern Hemisphere stratosphere support a strong polar vortex, fueled by a large equator-to-pole temperature gradient. Approximately once every two years, planetary waves rising from the troposphere impart a disturbance strong enough to weaken and destabilize the vortex, in some cases splitting it in half. Such events cause stratospheric temperatures to rise by about 50 °C over several days, affecting surface weather conditions for up to three months. The polar vortex is dynamically coupled to the midlatitude (tropospheric) jetstream, which sometimes weakens in response to SSW. This can engulf the midlatitudes in Arctic air and alter storm tracks, bringing severe weather conditions to unprepared locations. Predicting SSW events is therefore a prime objective in subseasonal-to-seasonal weather prediction, but their abruptness poses a real challenge. For a review of SSW observations, predictability and modeling, see [69] and references therein.

We consider the Holton-Mass model [70], augmented by time-dependent stochastic forcing as in [34] to represent unresolved processes and excite transitions between the strong- and weak-vortex conditions. Despite the simplicity of the model relative to state-of-the-art climate models, these transitions capture essential features of SSW such as the rapid upward burst of wave activity mediated by the "preconditioned" vertical structure of zonal-mean flow [71,72]. We briefly describe the model here, but refer the interested reader to [34,73,70,74] for additional background and details.

The model domain is the region of the atmosphere north of 30° and above the altitude of $z \approx 10$ km (the tropopause). The Holton-Mass model describes stratospheric flow in terms of a wave-mean flow interaction between two physical fields. The mean flow refers to the zonal-mean zonal wind $\overline{u}(y,z,t)$: the horizontal wind velocity component in the east-west (zonal) direction, averaged over a ring of constant latitude (zonal-mean, denoted by the overbar). The spatial coordinate y denotes the north-south (meridional) distance from the latitude line $\phi_0 = 60°$, i.e., $y = a(\phi - \phi_0)$, where a is the Earth's radius and ϕ is the latitude. The wave refers to the perturbation streamfunction $\psi'(x,y,z,t)$: the deviation from zonal mean (denoted by a prime symbol) of the geostrophic streamfunction, which is proportional to the potential energy of a given air parcel. Holton and Mass worked with the following ansatz for the interaction:

$$\begin{aligned} \overline{u}(y,z,t) &= U(z,t)\sin(\ell y) \\ \psi'(x,y,z,t) &= \mathrm{Re}\{\Psi(z,t)e^{ikx}\}e^{z/2H}\sin(\ell y) \end{aligned} \tag{26}$$

where $k = 2/(a\cos 60°)$ and $\ell = 3/a$ are zonal and meridional wavenumbers, and $H = 7$ km is a scale height. The equations in (26) prescribe the horizontal structure entirely, so the model state space consists of $U(z,t)$ and $\Psi(z,t)$, the latter being complex-valued. Insertion of (26) into the quasigeostrophic potential voriticity equation yields a system of two coupled PDEs. Following [34,70,73,74], we discretize the PDEs along the z dimension in 27 layers. After enforcing boundary conditions, this results in a 75-dimensional state space:

$$\begin{aligned} X_t = [&\mathrm{Re}\{\Psi(\Delta z,t)\},\ldots,\mathrm{Re}\{\Psi(25\Delta z,t)\}, \\ &\mathrm{Im}\{\Psi(\Delta z,t)\},\ldots,\mathrm{Im}\{\Psi(25\Delta z,t)\}, \\ &U(\Delta z,t),\ldots,U(25\Delta z,t)]. \end{aligned} \tag{27}$$

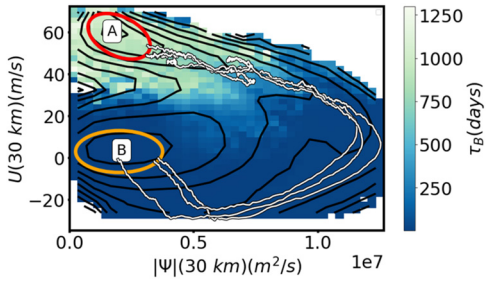

Fig. 7. Illustration of some key properties of the Holton-Mass model relevant to the prediction problems considered here. Red and yellow ellipses approximately mark the projections of states A and B, respectively, on the collective variables. The background color shows the average time to hit state B, clipped to a maximum of 1300 days to show detail. Black contours show the negative logarithm of the stationary density marginalized on these collective variables. Three transition paths harvested from a long simulation are shown in white.

The two states of interest in this model are a strong polar vortex, with large positive $U(z,t)$ (meaning eastward wind, marked as state A in Fig. 7), and a weak polar vortex, with a weak wind profile in which $U(z,t)$ sometimes dips negative (marked as state B in Fig. 7). Specifically, we define A and B as spheres centered on the model's two stable fixed points $(\Psi_{\mathbf{a}}, U_{\mathbf{a}})$ and $(\Psi_{\mathbf{b}}, U_{\mathbf{b}})$ in the 75-dimensional state space. The two spheres have radii of 8 and 20 respectively, with distances measured in the non-dimensionalized state space specified in [34]. In physical units, these correspond to the ellipsoids

$$A = \left\{ \Psi, U : \frac{\|\Psi - \Psi_{\mathbf{a}}\|^2}{(7.2 \times 10^5 \text{ m}^2/\text{s})^2} + \frac{\|U - U_{\mathbf{a}}\|^2}{(2.9 \text{ m/s})^2} \leq 8^2 \right\} \tag{28}$$

$$B = \left\{ \Psi, U : \frac{\|\Psi - \Psi_{\mathbf{b}}\|^2}{(7.2 \times 10^5 \text{ m}^2/\text{s})^2} + \frac{\|U - U_{\mathbf{b}}\|^2}{(2.9 \text{ m/s})^2} \leq 20^2 \right\} \tag{29}$$

where $\|\cdot\|$ is the complex vector 2-norm.

Fig. 7 illustrates the key features of this model relevant to the prediction problems we consider here. We see that the average time to reach B starting from A is over 1000 days, which is substantially longer than the longest lag times we consider here ($\leq$ 10 days). We can also see that the transition paths do not proceed through the saddlepoint of the effective free energy (i.e., the negative logarithm of the stationary density, marked by the contours), indicating that dynamical, non-diffusive, irreversible dynamics are important. Specifically, the transition path can be roughly divided into two stages: a "preconditioning" phase, in which the vortex gradually weakens, followed by an upward burst of wave activity that rips the vortex apart. Most of the committor's increase happens during the preconditioning phase, which siphons enstrophy (that is, squared vorticity, a measure of vortex strength that is conserved in the absence of dissipation) away from the mean flow and into the wave activity. The wave eventually dissipates, but only after its magnitude $|\Psi|$ bypasses the saddlepoint (Fig. 7). See [35,75] for further discussion.

To generate an initial data set, we sampled 30,000 points uniformly in $U(30 \text{ km})$ and $|\Psi|(30 \text{ km})$ from a long (50,000 days) trajectory and ran two ten-day trajectories from each starting point. Simulation details are reported in [34]. We simulated with a time step of 0.005 days, and saved the state of the system every 0.1 days. To validate our neural network results, we use a long trajectory of 500,000 days to compute

$$\langle q(s) \rangle = \mathbb{E}[\mathbb{1}_B(X(\tau)) | u_{\theta^*}(X(0)) \in [s, s + \Delta s]] \text{ for } s \in [0,1], \tag{30}$$

where θ^* are the parameters obtained from solving (9). This is the mean reference committor over the isocommittor surfaces from the neural network function. A perfect prediction corresponds to $\langle q(s) \rangle = s$. We use a similar construction for $m_{AB}q$, which we denote $\langle [m_{AB}q](s) \rangle$. For the committor, we take the overall error to be

$$\text{Error} = \sqrt{\int_0^1 (\langle q(s) \rangle - s)^2 ds}. \tag{31}$$

Because the lead time does not have a fixed range and scales exponentially with the noise, for it, we instead compute the relative error

$$\sqrt{\int_0^{40} (\langle [m_{AB}q](s) \rangle - s)^2 / s^2 ds}. \tag{32}$$

Fig. 8 shows the reference committor and lead time projected onto the collective variables.

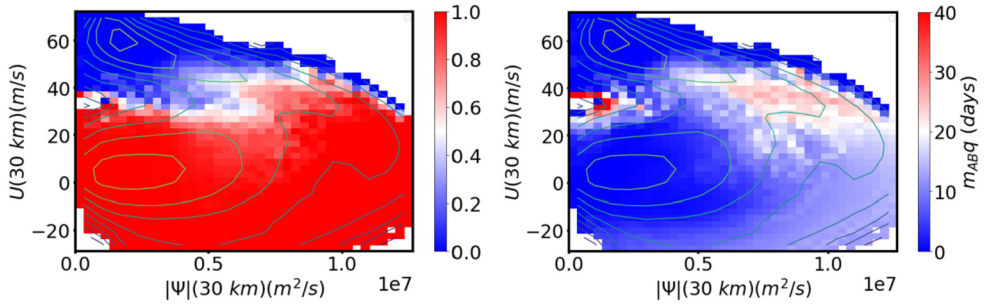

Fig. 8. Reference statistics for the Holton-Mass model. (left) Committor and (right) lead time computed from a long trajectory and projected onto $U(30\text{ km})$ and $|\Psi|(30\text{ km})$. Colors show reference statistics, and contours show the effective free energy.

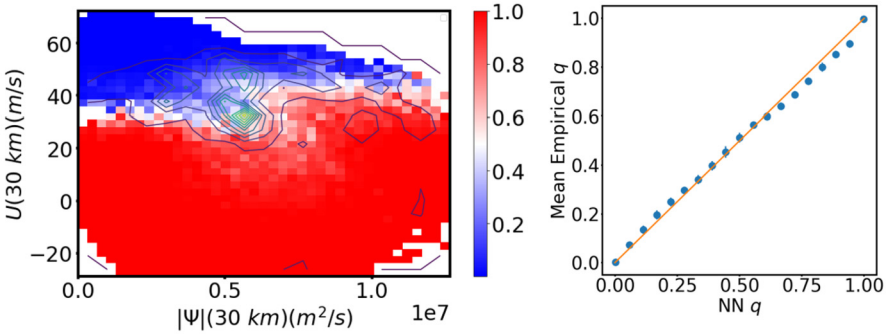

Fig. 9. Committor for the Holton-Mass model. (left) Colors show predictions, and contour lines show the density of added points from the adaptive sampling scheme described in Section 5. (right) Comparison of predicted and reference committors. Symbols show (30). Error bars show the standard deviation from networks trained on ten separate data sets resulting from the adaptive sampling scheme.

Fig. 9(left) shows results for the committor obtained with the adaptive sampling method. As collective variables in the adaptive sampling scheme we use $\xi = (U(30\text{ km}), |\Psi|(30\text{ km}))$. The space between $U(30\text{ km}) = -29$ and 72.5 m/s and between $|\Psi|(30\text{ km}) = 0$ and 1.26×10^7 m/s^2 is partitioned into a 20×20 grid of bins. We choose this collective variable space because it is physically intuitive, coming directly from the model's state space, and because it resolves SSW events well. Physically, U measures the strength of the vortex while $|\Psi|$ measures the strength of the disruptive wave. Their coupling is key to the nonlinear dynamics of the model. We begin with 50,000 pairs of short trajectories and add 22,000 new pairs of ten-day trajectories every 100 epochs for a total of 10 cycles. Thus the final number of trajectory pairs is 270,000. We take $\lambda = 10$ in (13). The network architecture is a fully connected feed forward network with 75 inputs, 10 hidden layers of width 50, with ReLU activation functions, and an output layer with a single sigmoid activation function. We stop training between each addition of data whenever the loss goes below zero (Fig. 10). Networks for the lead time have the same structure, except that they have a quadratic output layer. The contour lines in Fig. 9(left) indicate the density produced at the end of the training by the adaptive sampling procedure. The method concentrates new samples in the transition region without being given any information about its location. The method identifies the transition region on the fly.

To validate the results, we trained ten networks on the data set produced by the adaptive sampling method and computed (30) (Fig. 9(right)). The error bars show the standard deviation in $\langle q(s)\rangle$. We see that the training is robust and consistently able to produce good estimates of the committor. We used the data set obtained from the adaptive sampling scheme for the committor to train the neural network to predict the lead time (Fig. 11). Once again, we find that the method consistently produces good results compared with estimates from a long trajectory. We expect the errors in Fig. 11 to be larger than those in Fig. 9 because the estimated committor is used in the loss function for the lead time (as discussed below (7)), allowing errors to compound.

Finally, we determined how the performance of our method depended on key hyperparameters. To elucidate trends, we trained 10 networks on the data set produced by our adaptive sampling method. Fig. 12 shows the error in our scheme as the lag time is increased. As we observed in the case of the rugged Müller-Brown potential, the error decreases as the lag time increases. We note that as the lag time goes to infinity, all trajectories reach $A \cup B$, and the algorithm reduces to nonlinear regression of point estimates of the conditional expectation of being in state B (see (2)). We also investigated the dependence of the performance on the network depth, as shown in the right panel of Fig. 12. We found that deeper networks were able to achieve low errors at intermediate lag times, although there was relatively little sensitivity to this hyperparameter at short and long lag times.

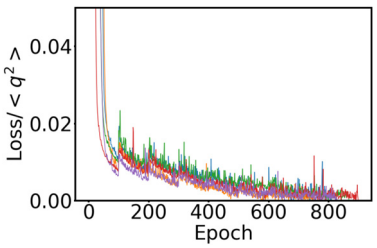

Fig. 10. The value of the loss as the training progresses for several replicates. We add data adaptively every 100 epochs and halt training when the loss goes below zero. Synchronized spikes in the loss function result from adding data where the loss is high.

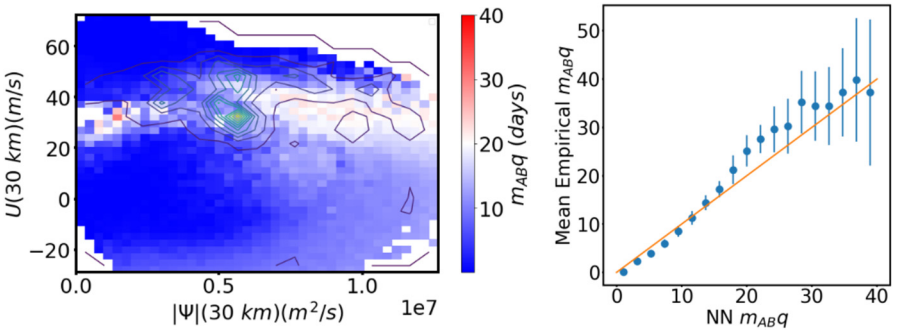

Fig. 11. Lead time for the Holton-Mass model. (left) Colors show predictions, and contour lines show the density of points in the data set obtained from the adaptive sampling scheme for the committor. (right) Comparison of predicted and reference lead times. Symbols show $\langle [m_{ABq}](s) \rangle$. Error bars show the standard deviation from networks trained on ten independent data sets resulting from the adaptive sampling scheme used to train the committor.

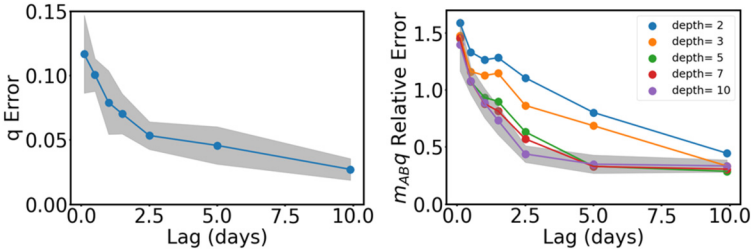

Fig. 12. Dependence of performance as key hyperparameters are varied. (left) RMS error of the committor as a function of lag time. (right) Relative error in the product used to solve for the lead time as a function of the network depth and lag time. Shading shows the standard deviation from networks trained on ten independent data sets resulting from the adaptive sampling scheme used to train the committor; on the right, shading is only shown for the deepest network for clarity.

7. Conclusions

In this work, we have proposed a machine learning method for solving prediction problems given a data set of short trajectories. By computing conditional expectations that solve Feynman-Kac equations rather than trying to learn the full dynamical law, we reduce the scope of the problem and hence render it more tractable. Our method has a number of advantages over existing ones:

- it allows computation of any statistic that can be cast in Feynman-Kac form;
- it does not require explicit knowledge of either the model underlying the data or its dynamics (e.g., the form of the generator and its parameters, such as the diffusion tensor);
- it allows for use of arbitrary lag times;
- it allows use of an arbitrary sampling distribution;
- it does not require microscopic reversibility.

We illustrate these advantages using two numerical examples. Using a three-dimensional model for which we can compute an accurate reference solution, we show that our method using short trajectory data is often more robust than related methods that instead use the differential operator form of the Feynman-Kac equation [40–43]. With the same model, we demonstrate the importance of having data in the low probability regions between metastable states and adequately weighting it against the data in the high probability regions. We propose a simple adaptive sampling scheme that allows us to

add data so as to target the largest contributions to the loss during training. Finally, we show that we can compute key statistics for a 75-dimensional model of SSW events (not just the committor but also the lead time) from trajectories that are significantly shorter than the times between events.

Our method opens new possibilities for the study of rare events using experimental and observational data. For example, data sets of short trajectories generated by weather forecasting centers can be analyzed by our method to study extreme weather and climate events [33]. However, the requirement that two trajectories be generated from each initial condition poses an obstacle to application of our method to many other data sets. Future work will focus on relaxing this restriction.

CRediT authorship contribution statement

John Strahan: Conceptualization, Writing, Investigation; **Justin Finkel:** Conceptualization, Writing; **Aaron Dinner:** Conceptualization, Writing; **Jonathan Weare:** Conceptualization, Writing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

We wish to thank Adam Antoszewski, Michael Lindsey, Chatipat Lorpaiboon, and Robert Webber for useful discussions and the Research Computing Center at the University of Chicago for computational resources. This work was supported by National Institutes of Health award R35 GM136381 and National Science Foundation award DMS-2054306. J.F. was supported at the time of writing by the U.S. DOE Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0019323.

References

[1] D.R. Easterling, G.A. Meehl, C. Parmesan, S.A. Changnon, T.R. Karl, L.O. Mearns, Climate extremes: observations, modeling, and impacts, Science 289 (5487) (2000) 2068–2074.
[2] A. AghaKouchak, L. Cheng, O. Mazdiyasni, A. Farahmand, Global warming and changes in risk of concurrent climate extremes: insights from the 2014 California drought, Geophys. Res. Lett. 41 (24) (2014) 8847–8852.
[3] C. Lesk, P. Rowhani, N. Ramankutty, Influence of extreme weather disasters on global crop production, Nature 529 (7584) (2016) 84–87.
[4] M.E. Mann, S. Rahmstorf, K. Kornhuber, B.A. Steinman, S.K. Miller, D. Coumou, Influence of anthropogenic climate change on planetary wave resonance and extreme weather events, Sci. Rep. 7 (1) (2017) 45242.
[5] D.J. Frame, S.M. Rosier, I. Noy, L.J. Harrington, T. Carey-Smith, S.N. Sparrow, D.A. Stone, S.M. Dean, Climate change attribution and the economic costs of extreme weather events: a study on damages from extreme rainfall and drought, Clim. Change 162 (2) (2020) 781–797.
[6] T.P. Sapsis, Statistics of extreme events in fluid flows and waves, Annu. Rev. Fluid Mech. 53 (1) (2021) 85–111.
[7] C.L. Brooks III, M. Karplus, B.M. Pettitt, Proteins, John Wiley & Sons, New York, 1988.
[8] M.C. Zwier, L.T. Chong, Reaching biological timescales with all-atom molecular dynamics simulations, Curr. Opin. Pharmacol. 10 (6) (2010) 745–752.
[9] M.S. Sohail, R.H. Louie, M.R. McKay, J.P. Barton, MPL resolves genetic linkage in fitness inference from complex evolutionary histories, Nat. Biotechnol. 39 (4) (2021) 472–479.
[10] G.R. Mirams, M.R. Davies, Y. Cui, P. Kohl, D. Noble, Application of cardiac electrophysiology simulations to pro-arrhythmic safety testing, Br. J. Pharmacol. 167 (5) (2012) 932–945.
[11] W. Liu, T.Y. Kim, X. Huang, M.B. Liu, G. Koren, B.-R. Choi, Z. Qu, Mechanisms linking T-wave alternans to spontaneous initiation of ventricular arrhythmias in rabbit models of long QT syndrome, J. Physiol. 596 (8) (2018) 1341–1355.
[12] S.J. Marrink, H.J. Risselada, S. Yefimov, D.P. Tieleman, A.H. De Vries, The Martini force field: coarse grained model for biomolecular simulations, J. Phys. Chem. B 111 (27) (2007) 7812–7824.
[13] M.G. Saunders, G.A. Voth, Coarse-graining methods for computational biology, Annu. Rev. Biophys. 42 (2013) 73–93.
[14] J.M. Jumper, N.F. Faruk, K.F. Freed, T.R. Sosnick, Trajectory-based training enables protein simulations with accurate folding and Boltzmann ensembles in cpu-hours, PLoS Comput. Biol. 14 (12) (2018) e1006578.
[15] R. Zwanzig, From classical dynamics to continuous time random walks, J. Stat. Phys. 30 (2) (1983) 255–262.
[16] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, T.R. Weikl, Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations, Proc. Natl. Acad. Sci. 106 (45) (2009) 19011–19016.
[17] G.R. Bowman, V.S. Pande, F. Noé, An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation, vol. 797, Springer Science & Business Media, 2013.
[18] B.E. Husic, V.S. Pande, Markov state models: from an art to a science, J. Am. Chem. Soc. 140 (7) (2018) 2386–2396.
[19] J.N. Kutz, Deep learning in fluid dynamics, J. Fluid Mech. 814 (2017) 1–4.
[20] S.H. Rudy, S.L. Brunton, J.L. Proctor, J.N. Kutz, Data-driven discovery of partial differential equations, Sci. Adv. 3 (4) (2017) e1602614.
[21] P.A. O'Gorman, J.G. Dwyer, Using machine learning to parameterize moist convection: potential for modeling of climate, climate change, and extreme events, J. Adv. Model. Earth Syst. 10 (10) (2018) 2548–2563.
[22] L. Zanna, T. Bolton, Data-driven equation discovery of ocean mesoscale closures, Geophys. Res. Lett. 47 (17) (2020) e2020GL088376.
[23] A. Chattopadhyay, P. Hassanzadeh, D. Subramanian, Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network, Nonlinear Process. Geophys. 27 (3) (2020) 373–389.

[24] K. Kashinath, M. Mustafa, A. Albert, J.-L. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, A. Manepalli, D. Chirila, R. Yu, R. Walters, B. White, H. Xiao, H.A. Tchelepi, P. Marcus, A. Anandkumar, P. Hassanzadeh Prabhat, Physics-informed machine learning: case studies for weather and climate modelling, Philos. Trans. R. Soc., Math. Phys. Eng. Sci. 379 (2194) (2021) 20200093.
[25] R. Du, V.S. Pande, A.Y. Grosberg, T. Tanaka, E.S. Shakhnovich, On the transition coordinate for protein folding, J. Chem. Phys. 108 (1) (1998) 334–350.
[26] P.G. Bolhuis, D. Chandler, C. Dellago, P.L. Geissler, Transition path sampling: throwing ropes over mountain passes in the dark, Annu. Rev. Phys. Chem. 53 (2002) 291–318.
[27] J. Strahan, A. Antoszewski, C. Lorpaiboon, B.P. Vani, J. Weare, A.R. Dinner, Long-time-scale predictions from short-trajectory data: a benchmark analysis of the trp-cage miniprotein, J. Chem. Theory Comput. 17 (5) (2021) 2948–2963.
[28] A. Antoszewski, C. Lorpaiboon, J. Strahan, A.R. Dinner, Kinetics of phenol escape from the insulin R_6 hexamer, J. Phys. Chem. B 125 (42) (2021) 11637–11649.
[29] H.C. Bloomfield, D.J. Brayshaw, P.L.M. Gonzalez, A. Charlton-Perez, Sub-seasonal forecasts of demand and wind power and solar power generation for 28 European countries, Earth Syst. Sci. Data 13 (5) (2021) 2259–2274.
[30] A. Tantet, F.R. van der Burgt, H.A. Dijkstra, An early warning indicator for atmospheric blocking events using transfer operators, Chaos, Interdiscip. J. Nonlinear Sci. 25 (3) (2015) 036406.
[31] D. Lucente, J. Rolland, C. Herbert, F. Bouchet, Coupling rare event algorithms with data-based learned committor functions using the analogue Markov chain, arXiv preprint, arXiv:2110.05050, 2021.
[32] D. Lucente, C. Herbert, F. Bouchet, Committor functions for climate phenomena at the predictability margin: the example of El Niño southern oscillation in the Jin and Timmermann model, J. Atmos. Sci. (2022).
[33] J. Finkel, E.P. Gerber, D.S. Abbot, J. Weare, Revealing the statistics of extreme events hidden in short weather forecast data, AGU Adv. 4 (2023) e2023AV000881.
[34] J. Finkel, R.J. Webber, E.P. Gerber, D.S. Abbot, J. Weare, Learning forecasts of rare stratospheric transitions from short simulations, Mon. Weather Rev. 149 (11) (2021) 3647–3669.
[35] J. Finkel, R.J. Webber, E.P. Gerber, D.S. Abbot, J. Weare, Exploring stratospheric rare events with transition path theory and short simulations, arXiv preprint, arXiv:2108.12727, 2021.
[36] M.O. Williams, I.G. Kevrekidis, C.W. Rowley, A data–driven approximation of the Koopman operator: extending dynamic mode decomposition, J. Nonlinear Sci. 25 (6) (2015) 1307–1346.
[37] G.A. Pavliotis, Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations, Texts in Applied Mathematics, vol. 60, Springer New York, New York, NY, 2014.
[38] E.H. Thiede, D. Giannakis, A.R. Dinner, J. Weare, Galerkin approximation of dynamical quantities using trajectory data, J. Chem. Phys. 150 (24) (2019) 244111.
[39] H. Sidky, W. Chen, A.L. Ferguson, High-resolution Markov state models for the dynamics of Trp-cage miniprotein constructed over slow folding modes identified by state-free reversible VAMPnets, J. Phys. Chem. B 123 (38) (2019) 7999–8009.
[40] Q. Li, B. Lin, W. Ren, Computing committor functions for the study of rare events using deep learning, J. Chem. Phys. 151 (5) (2019) 054112.
[41] Y. Khoo, J. Lu, L. Ying, Solving for high-dimensional committor functions using artificial neural networks, Res. Math. Sci. 6 (1) (2019) 1–13.
[42] H. Li, Y. Khoo, Y. Ren, L. Ying, A semigroup method for high dimensional committor functions based on neural network, in: Mathematical and Scientific Machine Learning, PMLR, 2022, pp. 598–618.
[43] G.M. Rotskoff, A.R. Mitchell, E. Vanden-Eijnden, Active importance sampling for variational objectives dominated by rare events: consequences for optimization and generalization, in: Mathematical and Scientific Machine Learning, PMLR, 2022, pp. 757–780.
[44] Y. Chen, J. Hoskins, Y. Khoo, M. Lindsey, Committor functions via tensor networks, J. Comput. Phys. 472 (2023) 111646.
[45] J. Han, A. Jentzen, W. E, Solving high-dimensional partial differential equations using deep learning, Proc. Natl. Acad. Sci. 115 (34) (2018) 8505–8510.
[46] J. Han, J. Lu, M. Zhou, Solving high-dimensional eigenvalue problems using deep neural networks: a diffusion Monte Carlo like approach, J. Comput. Phys. 423 (2020) 109792.
[47] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (6) (2021) 422–440.
[48] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[49] F. Chen, J. Huang, C. Wang, H. Yang, Friedrichs learning: weak solutions of partial differential equations via deep learning, arXiv preprint, arXiv:2012.08023, 2020.
[50] Q. Zeng, S.H. Bryngelson, F. Schäfer, Competitive physics informed networks, arXiv preprint, arXiv:2204.11144, 2022.
[51] P.V. Banushkina, S.V. Krivov, Nonparametric variational optimization of reaction coordinates, J. Chem. Phys. 143 (18) (2015) 11B607_1.
[52] S.V. Krivov, Blind analysis of molecular dynamics, J. Chem. Theory Comput. 17 (5) (2021) 2725–2736.
[53] B. Roux, String method with swarms-of-trajectories, mean drifts, lag time, and committor, J. Phys. Chem. A 125 (34) (2021) 7558–7571.
[54] B. Roux, Transition rate theory, spectral analysis, and reactive paths, J. Chem. Phys. 156 (13) (2022) 134111.
[55] W. E, W. Ren, E. Vanden-Eijnden, Transition pathways in complex systems: reaction coordinates, isocommittor surfaces, and transition tubes, Chem. Phys. Lett. 413 (1–3) (2005) 242–247.
[56] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[57] D.P. Kingma, J. Ba Adam, A method for stochastic optimization, arXiv:1412.6980, 2017.
[58] K. Müller, L.D. Brown, Location of saddle points and minimum energy paths by a constrained simplex optimization procedure, Theor. Chem. Acc. 53 (1) (1979) 75–93.
[59] B. Leimkuhler, C. Matthews, Rational construction of stochastic numerical methods for molecular sampling, Appl. Math. Res. Express 2013 (1) (2013) 34–56.
[60] C. Lorpaiboon, J. Weare, A.R. Dinner, Augmented transition path theory for sequences of events, arXiv preprint, arXiv:2205.05067, 2022.
[61] R.R. Coifman, S. Lafon, Diffusion maps, Appl. Comput. Harmon. Anal. 21 (1) (2006) 5–30.
[62] C.R. Schwantes, V.S. Pande, Modeling molecular kinetics with tICA and the kernel trick, J. Chem. Theory Comput. 11 (2) (2015) 600–608.
[63] A. Bittracher, S. Klus, B. Hamzi, P. Koltai, C. Schütte, Dimensionality reduction of complex metastable systems via kernel embeddings of transition manifolds, J. Nonlinear Sci. 31 (1) (2020) 3.
[64] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, arXiv preprint, arXiv:1312.6114, May 2014.
[65] C. Wehmeyer, F. Noé, Time-lagged autoencoders: deep learning of slow collective variables for molecular kinetics, J. Chem. Phys. 148 (24) (2018) 241703.
[66] A. Ma, A.R. Dinner, Automatic method for identifying reaction coordinates in complex systems, J. Phys. Chem. B 109 (14) (2005) 6769–6779.
[67] E.H. Thiede, B. Van Koten, J. Weare, A.R. Dinner, Eigenvector method for umbrella sampling enables error analysis, J. Chem. Phys. 145 (8) (2016) 084115.
[68] A.R. Dinner, E.H. Thiede, B.V. Koten, J. Weare, Stratification as a general variance reduction method for Markov chain Monte Carlo, SIAM/ASA J. Uncertain. Quantificat. 8 (3) (2020) 1139–1188.
[69] M.P. Baldwin, B. Ayarzagüena, T. Birner, N. Butchart, A.H. Butler, A.J. Charlton-Perez, D.I.V. Domeisen, C.I. Garfinkel, H. Garny, E.P. Gerber, M.I. Hegglin, U. Langematz, N.M. Pedatella, Sudden stratospheric warmings, Rev. Geophys. 59 (1) (2021) e2020RG000708.

[70] J.R. Holton, C. Mass, Stratospheric vacillation cycles, J. Atmos. Sci. 33 (11) (1976) 2218–2225.
[71] J.P. Sjoberg, T. Birner, Stratospheric wave–mean flow feedbacks and sudden stratospheric warmings in a simple model forced by upward wave activity flux, J. Atmos. Sci. 71 (11) (2014) 4055–4071.
[72] P. Maher, E.P. Gerber, B. Medeiros, T.M. Merlis, S. Sherwood, A. Sheshadri, A.H. Sobel, G.K. Vallis, A. Voigt, P. Zurita-Gotor, Model hierarchies for understanding atmospheric circulation, Rev. Geophys. 57 (2) (2019) 250–280.
[73] J. Finkel, D.S. Abbot, J. Weare, Path properties of atmospheric transitions: illustration with a low-order sudden stratospheric warming model, J. Atmos. Sci. 77 (7) (2020) 2327–2347.
[74] B. Christiansen, Chaos, quasiperiodicity, and interannual variability: studies of a stratospheric vacillation model, J. Atmos. Sci. 57 (18) (2000) 3161–3173.
[75] S. Yoden, Dynamical aspects of stratospheric vacillations in a highly truncated model, J. Atmos. Sci. 44 (24) (1987) 3683–3695.