Neural network approach to time-dependent dividing surfaces in classical reaction dynamics

Philippe Schraft, Andrej Junginger, Matthias Feldmaier, Robin Bardakcioglu, Jörg Main, Günter Wunner, and Rigoberto Hernandez

Phys. Rev. E **97**, 042309 — Published 19 April 2018

DOI: 10.1103/PhysRevE.97.042309

Neural network approach to time-dependent dividing surfaces in classical reaction dynamics

Philippe Schraft,[1] Andrej Junginger,[1, *] Matthias Feldmaier,[1] Robin Bardakcioglu,[1] Jörg Main,[1] Günter Wunner,[1] and Rigoberto Hernandez[2]

[1]*Institut für Theoretische Physik 1, Universität Stuttgart, 70550 Stuttgart, Germany*
[2]*Department of Chemistry, The Johns Hopkins University, Baltimore, MD, USA*

(Dated: March 14, 2018)

In a dynamical system, the transition between reactants and products is typically mediated by an energy barrier whose properties determine the corresponding pathways and rates. The latter is the flux through a dividing surface (DS) between the two corresponding regions and it is exact only if it is free of recrossings. For time-independent barriers, the DS can be attached to the top of the corresponding saddle point of the potential energy surface, and in time-dependent systems, the DS is a moving object. The precise determination of these direct reaction rates, e. g. using transition state theory, requires the actual construction of a DS for a given saddle geometry which is in general a demanding methodical and computational task, especially in high-dimensional systems. In this paper, we demonstrate how such time-dependent, global, and recrossing-free DSs can be constructed using neural networks. In our approach, the neural network uses the bath coordinates and time as input and it is trained in a way that its output provides the position of the DS along the reaction coordinate. An advantage of this procedure is that, once the neural network is trained, the complete information about the dynamical phase space separation is stored in the network's parameters, and a precise distinction between reactants and products can be made for all possible system configurations, all times, and with little computational effort. We demonstrate this general method for two- and three-dimensional systems, and explain its straightforward extension to even more degrees of freedom.

I. INTRODUCTION

One of the grand challenges in reaction dynamics is the accurate determination of reaction pathways and rates. Theoretical and computational approaches require both a precise understanding of the underlying potential energy landscape describing the reactive system and the effect of possible external —that is, time-dependent— forces. Formally, all that then remains is the integration of the equations of motion for a sufficiently large ensemble of trajectories and long times to completely characterize the dynamics. In practice, however, the dimensionality of reactive systems is sufficiently large that such approaches are computationally prohibitive. The key to resolving this challenge has long been known to lie in the observation that reactions are typically mediated by an energy barrier separating reactant and product basins in the underlying state space.

Transition state theory (TST) [1–13] has been a powerful approach for obtaining chemical reactions rates approximately. TST rates are determined by the flux through a dividing surface (DS) separating reactants and products divided by the reactant population. These rates are exact if and only if the DS is free of recrossings, i. e. it is crossed by each reactive trajectory exactly once. It should be noted that the boundary conditions imposed on the reaction process are taken to be such that the particles that reach the reactant or product basins never return either because the potential is unbound, geminate recombination is quenched, or particles are not propagated beyond these basins. For the overall rate, one can sometimes combine the direct rates between basins as was recently done in Refs. [14, 15]. Many of the advances over the past century have hinged on the construction of DSs with decreasing degree of recrossing. Thus, a crucial step to calculate accurate TST rates is the precise determination of a nonrecrossing hypersurface, and this is a nontrivial task especially in high-dimensional and time-dependent systems.

In autonomous systems, a *local* DS can be constructed using the normally hyperbolic invariant manifold [16–24], e. g. using normal form transformations [25]. However, the situation quickly becomes daunting when the system is time-dependent or if one aims to obtain a *global* DS (at points far from the saddle region.) In the former case, the DS itself is a time-dependent object that is harder to calculate, whereas in the latter case, perturbative approaches break down because of a finite radius of convergence.

For systems with one and two degrees of freedom this problem can basically be solved using a general minimization procedure based on Lagrangian descriptors (LDs) [26, 27]. In the one-dimensional case, the moving DS is attached to the time-dependent transition state trajectory [28–34]. If additional bath degrees of freedom are taken into account, these approaches are formally extendable to higher dimensions, and some of us have demonstrated this with at least one additional bath mode [35]. Its structure is directly related to special

[*] Present address: Machine Learning Team at ETAS GmbH (Bosch Group).

unstable trajectories close to the saddle which are challenging to construct even in the one-dimensional case. Additional bath degrees of freedom lead to increasingly chaotic trajectories and increase the dimensionality of the time-dependent DS, making it increasingly challenging to obtain a single point of the DS. As such a process leads only to a finite number of points, one remains with the additional challenge of obtaining a continuous surface.

In this paper, we demonstrate how all these issues become manageable when neural networks (NNs) are trained to approximate the DS. Although NNs have been known for decades, they have regained intense attention over the recent years, mostly because of their large success in the fields of image recognition and classification and the increasing power of computers to handle large datasets and complex representations. Recently, Carpenter et al. [36] demonstrated the use of such machine learning techniques to obtain molecular dynamics trajectories from which one can extract phase space structures. Further, NNs and other machine learning methods are also used to construct high-dimensional potential energy surfaces and other quantities in theoretical chemistry [37–46]. Here we show that we can construct one such phase space structure —the DS— directly using a NN. Our method is based on the fact that *single* points on the DS can be calculated with reasonable effort as well as easily propagated in time: They are easy to obtain, because they are minima of the LD for a given time t and, in addition, their dynamics is readily obtained by numerically integrating the equations of motion. To obtain the DS over the entire domain, we train a NN on a set of such points. The input to the NN is the system's configuration and its output is the position of the DS along the reaction coordinate. From the machine learning perspective, what we present in this paper is a classical, multidimensional regression task that we solve using feed-forward NNs.

This paper is organized as follows: In Sec. II, we review the basic concepts of calculating nonperturbative DSs by minimizing LDs, and that of NNs. In Sec. III, we present the application of the method to two- and three-dimensional reactive systems.

II. THEORY

A. Time-dependent dividing surfaces in classical reaction dynamics

In dynamical systems where reactants and products are separated by an energetic barrier, each reactive particle has to overcome the barrier region to reach the corresponding opposite basin, e. g. the product basin and reactant basin, respectively. In accordance with TST, the local properties of the barrier determine both the reaction pathways and the corresponding rates. The stable and unstable manifolds $\mathcal{W}_{\mathrm{s,u}}$ play an important role in this picture. Here, the stable manifold $\mathcal{W}_{\mathrm{s}}$ is the set of points

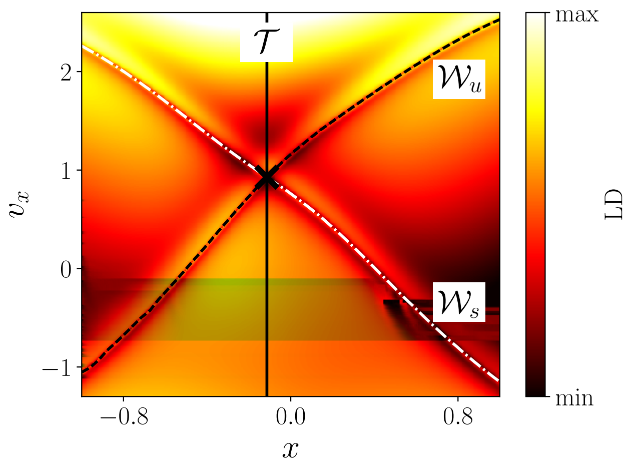

Figure 1. Plot of the LD (1) of the rank-1 saddle defined by the potential (9) for $y = z = v_y = v_z = t = 0$. The axes approximate the reaction coordinate x_{reac} and its associated velocity. The color coding represents the value of the LDs. The stable and unstable manifolds $\mathcal{W}_{\mathrm{s,u}}$ are visible as local minima of the LD. The stable manifold $\mathcal{W}_{\mathrm{s}}$ is highlighted by the white dash-dotted curve and the unstable manifold $\mathcal{W}_{\mathrm{u}}$ by the black dashed one. Their intersection is marked with a black cross ($\times$) and denotes one point of the NHIM of the corresponding saddle. The solid vertical black line is the DS, $\mathcal{T}$, attached to this point of the NHIM. The choice of the DS is, in general, not unique. We chose the most simple case, viz. a vertical line independent of v_x, where we assume that the DS does not cross any of the manifolds in an additional point. Such an intersection exists for each configuration of bath coordinates and for each time. See text for further explanations.

that exponentially approach a time-dependent normally hyperbolic invariant manifold (NHIM) and the unstable manifold $\mathcal{W}_{\mathrm{u}}$ are those points that exponentially depart from the NHIM. These manifolds have a crucial meaning because they separate reactive from non-reactive trajectories (see Fig. 1). The intersection of the closure of these manifolds —here meant in a geometric, not a dynamic sense— is the NHIM to which a recrossing-free DS can be attached that separates the reactant from the product basin.

The lowest-energy pathway crosses the barrier region at a rank-1 saddle point which locally defines a set of $d-1$ bath coordinates and a one-dimensional reaction coordinate that are parallel and perpendicular, respectively, to the associated $(2d-2)$-dimensional NHIM. The attached DS has an increased dimension $(2d-1)$ in respect to the NHIM, as can be seen in Fig. 1 and thus separates the $(2d)$-dimensional phase space. It also is useful for what follows to associate one degree of freedom x_{reac} as the reaction coordinate, and the remaining coordinates $\boldsymbol{x}_{\mathrm{bath}}$ as bath coordinates, as indicated by the subscripts. With these defined coordinates, for example, we can illustrate the parameterized hypersurfaces such as is shown in Fig. 2 (b).

Consideration of the barrier top dynamics in terms of the associated manifolds has the advantage that our geometric picture remains valid when the barrier is time-dependent. The most important difference in the time-independent case is that the position at which the manifolds intersect is not identical with the instantaneous barrier top. The manifold description also has the advantage that it lends itself to calculation through the machinery of dynamical systems theory including, for example, the very general method (cited in the introduction) minimizing the LD,

$$\mathcal{L}(\boldsymbol{x}_0, \boldsymbol{v}_0, t_0) = \int_{t_0-\tau}^{t_0+\tau} \|\boldsymbol{v}(t)\| \,\mathrm{d}t\,. \tag{1}$$

Due to the extremal property of trajectories on manifolds, the stable and unstable manifolds are related to the LD via

$$\mathcal{W}_\mathrm{s}(t_0) = \arg\min \mathcal{L}^{(f)}(\boldsymbol{x}_0, \boldsymbol{v}_0, t_0)\,, \tag{2a}$$
$$\mathcal{W}_\mathrm{u}(t_0) = \arg\min \mathcal{L}^{(b)}(\boldsymbol{x}_0, \boldsymbol{v}_0, t_0)\,, \tag{2b}$$

where (f, b) denote the forward (f: $t_0 \le t \le t_0 + \tau$) or the backward (b: $t_0 - \tau \le t \le t_0$) direction of time. Here, the operator arg min denotes the argument of the local minimum. The intersection of both manifolds including their closure

$$\mathcal{T}(t_0) = \mathcal{W}_\mathrm{s}(t_0) \cap \mathcal{W}_\mathrm{u}(t_0) \tag{3}$$

yields the NHIM at a given time t_0.

An essential challenge of exact reaction rate calculations is the tracking of a potentially cost-prohibitively large number of particles in the ensemble, and the determination of the exact time of each crossing through the DS. Further challenges when calculating such DSs are the numerical determination of the intersection of the manifolds and the high dimensionality $2d - 2$ of the NHIM, to which the DS is attached, as d becomes large. For a system with $d = 1$, the calculation is trivial because it is a single point. The case $d = 2$ is significantly more demanding leading to a two-dimensional surface embedded in full phase space which is numerically still manageable using e. g. cubic spline interpolation. For even higher dimensions, the effort to approximate the hypersurface, however, quickly becomes so large that following reactive particles becomes numerically intractable. With increasing dimensionality, one needs a method that interpolates smoothly and continuously across the surface. Such an interpolation is necessary because the reactive particles may cross the DS at an arbitrary phase space point for each initially unknown time.

It is the purpose of the remaining part of this paper to show that NNs are a useful tool for overcoming this dimensionality issue, and that they can make numerical treatments possible for systems with a high number of bath degrees of freedom.

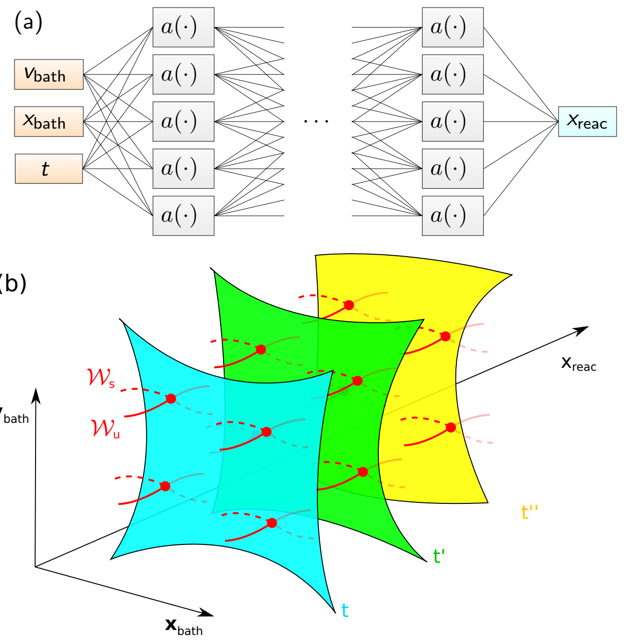

Figure 2. (a) General structure of the neuronal network: Time t, the bath coordinates $\boldsymbol{x}_\mathrm{bath}$, and their velocities $\boldsymbol{v}_\mathrm{bath}$ define the input layer (orange nodes) and the reaction coordinate x_reac is the output layer (cyan node). The gray nodes denote the hidden layers of adjustable depth and number of neurons, and $a(\cdot)$ is the respective activation function. (b) General scheme of the procedure: The time-dependent NHIM is defined by the intersection between the stable and unstable manifolds $\mathcal{W}_\mathrm{s,u}$ (solid and dashed red lines) for the respective, fixed values of the bath coordinates and for given time t. The phase space coordinates of such an intersection point serve as the training set of the neural net.

B. Feed-forward neural networks

In this section, we give a brief and basic overview of feed-forward NNs to set the stage for our implementation to construct the time-dependent DS. For details and more sophisticated types of NNs like recurrent or convolutional ones, we refer the reader to the extensive literature on this topic, such as e. g. Refs. [47, 48] and references therein.

In a feed-forward NN, information is processed consecutively by propagating it through a number of layers, each of which performs a nonlinear transformation of the input(s) resulting in the output(s). Such NNs typically consist of one input layer, one output layer as well as one or several in-between hidden layers, and each layer is made of several neurons, as illustrated in Fig. 2. Each neuron in the hidden layers is connected to the neurons of the previous and the subsequent layer.

The most basic component, the neuron, is responsible for the information processing which works as follows: The input Σ of a neuron i in layer l is a function of the information —that is, outputs $a_j^{(l-1)}$— from the previous layer $(l-1)$ and is given by

$$\Sigma_i^{(l)} = \left(\sum_{j=1}^{N} w_{ij}^{(l)} a_j^{(l-1)} \right) + b_i^{(l)} . \tag{4}$$

The weight $w_{ij}^{(l)}$ represents the connection between the neurons and reflect the importance of the information from the the previous neuron j to the neuron i. The value $b_i^{(l)}$ is a threshold which determines the area of activation and is also referred to as the bias. Of great importance for the functionality and learning behavior of the NN are the activation functions

$$a_i^{(l)} = a\left(\Sigma_i^{(l)}\right) . \tag{5}$$

They are generally nonlinear and their choice depends on the specific task of the NN. In this paper, we use the inverse tangent $a(\Sigma) = \arctan(\Sigma)$ as activation function for the hidden layers. For the input and output layer we use the identity matrix as the weight of the activation function.

The weights and biases in Eq. (4) serve as free parameters which are, during the network training, adjusted such that the NN performs the desired task. The intended behavior is trained by providing a cost function C which acts as a penalty term defining the network's deviation between its actual output $\tilde{\boldsymbol{y}}$ and the desired result $\boldsymbol{y}$. To train the neural network, the cost function is minimized, e. g. using back propagation and standard numerical procedures such as (batch) stochastic gradient descent. In this paper, we use the standard mean squared error cost function

$$C_{\boldsymbol{w},\boldsymbol{b}}(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = \frac{1}{2n} \sum_{i=1}^{n} \left\| \boldsymbol{y}_i - \tilde{\boldsymbol{y}}(\boldsymbol{x}_i) \right\|^2 , \tag{6}$$

where $\boldsymbol{x}_i$ is an input vector of the NN, $\boldsymbol{y}_i$ is the expected output, and n is the total number of training points or the batch size, respectively. The function $\tilde{\boldsymbol{y}}(\boldsymbol{x}_i)$ is the output of the NN for the given input $\boldsymbol{x}_i$ and the subscripts $\boldsymbol{w}, \boldsymbol{b}$ denote the dependence of the cost function on the network parameters.

Using gradient descent, the weights and biases are updated in each step according to

$$w_{ij}^{(l)} \to w_{ij}^{(l)} - \eta \frac{\partial C_{\boldsymbol{w},\boldsymbol{b}}}{\partial w_{ij}^{(l)}} , \tag{7a}$$

$$b_i^{(l)} \to b_i^{(l)} - \eta \frac{\partial C_{\boldsymbol{w},\boldsymbol{b}}}{\partial b_i^{(l)}} , \tag{7b}$$

where η is the learning rate determining the step size of the gradient descent method.

The NN as described above was sufficiently simple to be implemented without using special software packages. However, use of freely available NNs will allow for more efficient implementations and optimization of the training in the future.

C. Application of neural networks to time-dependent dividing surfaces

In this paper, our goal is to use NNs for a multidimensional regression task in order to approximate the phase space DS of a reactive system. The construction of the DS, compare Fig. 1, allows us to approximate the DS by only knowing the reaction coordinate of the NHIM for a given set of bath coordinates and a given time. More specifically, we want to learn the function [49] that maps the current configuration of the system, i. e. the values of the bath degrees of freedom $\boldsymbol{x}_{\text{bath}}, \boldsymbol{v}_{\text{bath}}$ and time t, to the value x_{reac} of the respective reaction coordinate,

$$x_{\text{reac}} = x_{\text{reac}}(\boldsymbol{x}_{\text{bath}}, \boldsymbol{v}_{\text{bath}}, t) . \tag{8}$$

This is, in general, a complicated, time-dependent and nonlinear relation, but a finite number of representatives can be generated readily using the method described in Sec. II A through minimization of the LD (3). NNs are an ideal approach for extending the information from this set of points—viz. the training set—to provide an output for the entire domain. Specifically, the NN is trained taking time t and bath coordinates $\boldsymbol{x}_{\text{bath}}, \boldsymbol{v}_{\text{bath}}$ as inputs to the first layer, that leads to an input dimension of $2d-1$, and the corresponding reaction coordinate x_{reac} as the outputs from the last layer.

D. Model system

To demonstrate the ability of NNs to approximate DSs, we use a higher-dimensional extension of the model reactive system already investigated in Ref. [35]:

$$\begin{aligned} V(x, y, z, t) = {} & E_{\text{b}} \exp\left(-a_{\text{b}} \left[x - x_{\text{b}}(t)\right]^2\right) \\ & + \frac{\omega_{\text{y}}^2}{2} \left[y - y_{\min}(x)\right]^2 + \frac{\omega_{\text{z}}^2}{2} \left[z - z_{\min}(x)\right]^2 . \end{aligned} \tag{9}$$

This potential contains a Gaussian barrier along the x-direction (which acts as x_{reac}), where E_{b} is the barrier height and a_{b} the width. The orthogonal degrees of freedom, x and y, act as the bath coordinates $\boldsymbol{x}_{\text{bath}}$. The barrier oscillates along the x-axis according to

$$x_{\text{b}}(t) = x_{\text{b},0} \sin\left(\omega_{x_{\text{b}}} t\right) , \tag{10}$$

where $x_{\text{b},0}$ is the amplitude and $\omega_{x_{\text{b}}}$ the frequency of the oscillation. Further, the potential includes harmonic potentials along the y- and z-axis with the frequencies $\omega_{y,z}$ and the degrees of freedom are nonlinearly coupled to the x-direction along the respective minimum energy valleys

$$y_{\min}(x) = z_{\min}(x) = \frac{2}{\pi} \arctan(2x) . \tag{11}$$

For simplicity, all variables are introduced in dimensionless, and are set to $E_{\text{b}} = 2$, $a_{\text{b}} = 1$, $\omega_{\text{y}} = 2$, $\omega_{\text{z}} = 2$,

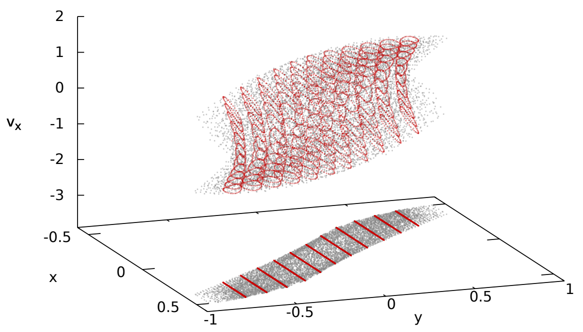

Figure 3. Extension of the training set in 2D. The red points are calculated using the LD minimization procedure according to Eq. (3). The gray points are afterwards obtained by using the red dots as initial values and propagating the respective particle according to the equations of motion. By this procedure the number of training points for the neural network is easily extended.

$x_{\mathrm{b},0} = 0.4$ and $\omega_{x_\mathrm{b}} = \pi$. Due to the time dependence of the saddle, the resulting DS is also be time-dependent. Note that the two-dimensional model system previously discussed in Ref. [35] is obtained by neglecting the z-degree of freedom in Eq. (9).

III. RESULTS AND DISCUSSION

In this section, we demonstrate several aspects of the use of NNs for obtaining DSs. The starting point for all of the implementations is a set of phase space points located on the NHIM that are precalculated according to Eq. (3) for the respective potential (9). See Fig. 2(b) for an illustration. Note that it is not within the scope of this paper to illustrate and explain the performed minimization process using Eq. (3) as that has already been confirmed and discussed in our recent work [35].

To obtain the training set, we perform the minimization (3) only for a limited number of points—e. g. on an equidistant grid in the bath coordinates and in time—highlighted in red in Fig. 3. Since all these points are, by construction, located on an invariant manifold, this structure remains valid as the respective particle is propagated in time. Thus, we can readily add new points to the training set as needed. For example, we can extend the training set size by a factor of ten simply by integrating five time steps in forward and backward time (if only a few steps are performed, there is no need to take special care of the instability of the trajectory). As a consequence, the newly calculated points are completely unordered but nevertheless associated with the appropriate corresponding time. Consequently, their use in training the NN, although formally out of order, introduces no error.

A. System with two degrees of freedom

First, we apply the NN to approximate a time-dependent DS for a system with two degrees of freedom. In this case, we neglect the z-degree of freedom in Eq. (9), and we use a NN with two hidden layers consisting of 40 and 10 neurons. Further, the NN is trained on 2400 points of the NHIM and the training is stopped after 10^6 training cycles. Figure 4 shows the result of the NN regression: snapshots of the NHIM at different times over one period. Specifically, the reaction coordinate x is shown over the domain of bath coordinates y and v_y (v_x is not shown). The surfaces displayed are the actual output of the neural network and the color code refers to the deviation compared to a spline interpolation of the underlying training set. The deviation between the training data points and the neural network prediction is small throughout. Its accuracy is also confirmed by the small value of the mean squared training error ($=2.95 \times 10^{-6}$) that we achieved at the end of the training procedure.

At some times and grid points, small areas show large deviations between the neural network output and the spline interpolation (cyan and yellow spots on the surface). However, as we have verified by a direct comparison with training data, these points are not erroneous predictions of the NN, but rather indicate grid points where the original training point has been poorly determined during the minimization procedure (3). The NN is thus smoothing out such numerical errors in the training set due to its global learning property (whereas spline methods just interpolate between points). This is another advantage for the use of NNs to numerically approximate DSs.

B. System with three degrees of freedom

We now take into account the full system (9) with three degrees of freedom, and calculate $10,000$ points of the NHIM as our training and verification sets ($4,000$ and $6,000$ points, respectively). The input layer of the NN now requires five neurons for all four bath coordinates, y, z, v_y, v_z, and one neuron for the time dependence of the NHIM. Here, we use two hidden layers with 40 and 10 neurons, respectively. The NN parameters are updated through each full loop — called an epoch — of the training algorithm across the complete training data set. The cost function of the evolving NN in terms of epochs shows the training progress in Fig. 5. Note that the cost for the training and verification sets nearly agree up to the first 10^3 epochs. For more epochs, the cost function of the verification set decreases less rapidly, indicating a saturation of the interpolation accuracy as is typical for NNs.

In order to verify the recrossing-free property of the DS generated by the NN, we regard a thermal ensemble of $250,000$ particles in the reactant well ($x < 0$) with a

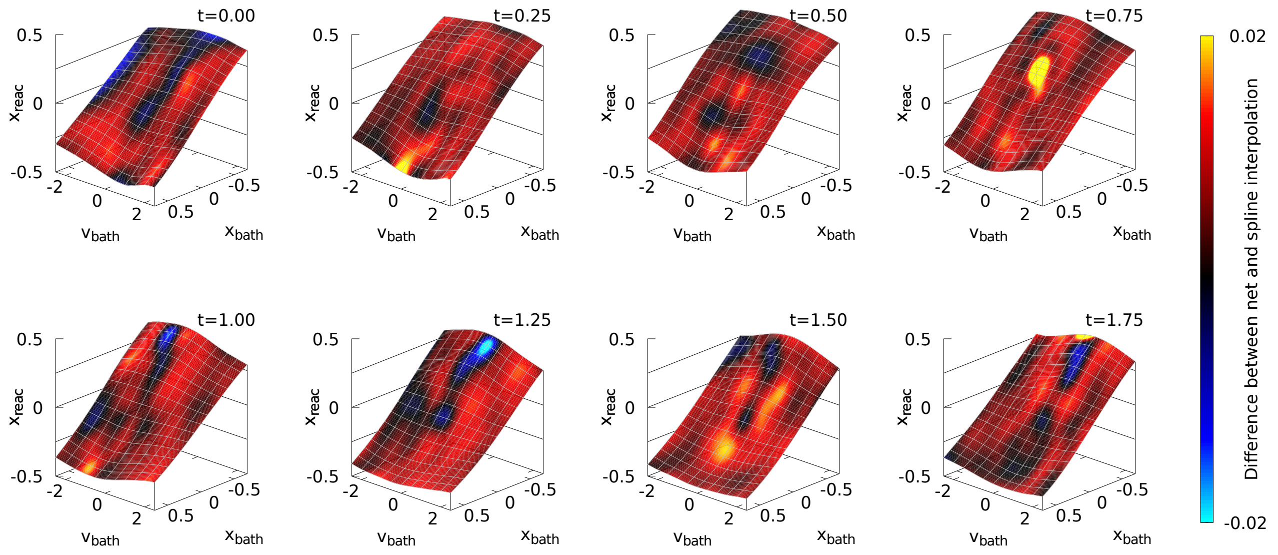

Figure 4. Time-dependent NHIM obtained from the NN in dependence of the bath coordinates. Note that x_{reac} refers to x and x_{bath} and v_{bath} to y and its velocity, respectively, in Eq. (9). The color of the surface indicates the difference between the NN's output and a spline interpolation of the data on which the NN is trained. Here, the smoothing ability of the neural network becomes visible.

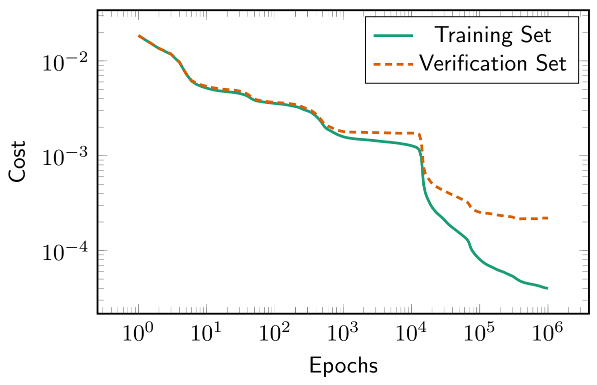

Figure 5. Cost functions of the data set for a DS of the system with three degrees of freedom. The cost function represents the difference between the output of the neural network and the actual points in the data set. The training set is a subset of the whole data set on which the neural network is trained and the verification set is a disjunct second subset on which the neural network is not trained. So the cost function of the verification set is a measure of how accurately the neural network interpolates. The training was stopped after 10^6 epochs.

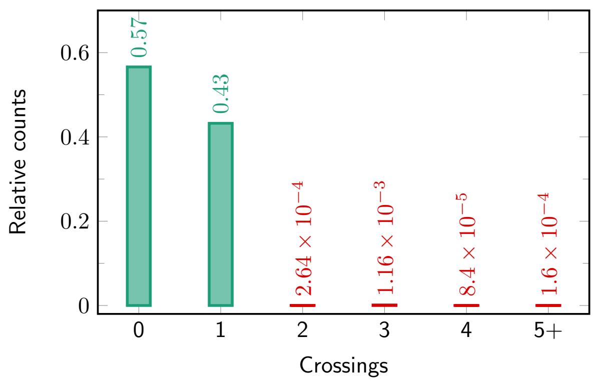

Figure 6. Counted crossings through the DS approximated by a neural network. The corresponding reaction curve can be seen in Fig. 7. The bars corresponding to two and more crossings are recrossings. Taking into account the small number of recrossings, the approximated DS is a recrossing-free DS with only tiny numerical errors.

density distribution

$$\rho(\boldsymbol{x}, \boldsymbol{v}) = \rho_{\mathrm{therm}} \delta(x+1) \Theta(v_x) , \tag{12}$$

where ρ_{therm} is a Boltzmann distribution, δ is the Dirac delta function and Θ is the Heaviside step function. We use these functions because the potential is unbound at $x \to \pm\infty$, and particles with outward velocities would not take part in the reaction. The particles of the generated thermal ensemble are propagated and any crossing through the DS is counted.

The relative counts of crossings are shown in Fig. 6. The two green bars for zero and one crossing do not violate the recrossing-free property of an exact DS while the red bars —for multiple crossings— violate it. Almost all particles either show no or one crossing, and only a negligible fraction (1.67×10^{-3}) of all particles show recrossings at all. Thus, the recrossing-free property is fulfilled to a high degree.

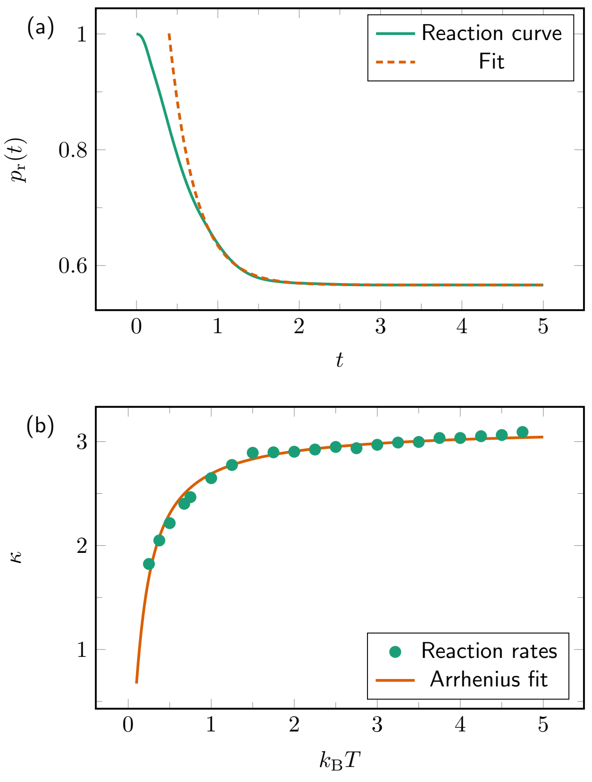

Figure 7. (a) Behavior of the reactant population over time of a thermal ensemble at $k_\mathrm{B}T = 4.75$ in units of $E_\mathrm{b}/2$. The population is normalized to the amount of particles propagated, which were 250,000 particles. There is also an exponential fit (dashed line) to this reaction curve as in Eq. (13) which yields a reaction rate of $\kappa = 3.09232$. (b) Reaction rates determined with fits as in subfigure (a) for different values of $k_\mathrm{B}T$. The fit corresponds to the Arrhenius equation (14).

The corresponding population decay of the initial ensemble is shown in Fig. 7(a), The decay exhibits an expected monotonous decrease and the subsequent saturation to a constant population. The dashed line is an exponential fit

$$p_\mathrm{r}(t) = p_{\mathrm{r},0}\mathrm{e}^{-\kappa t} + p_{\mathrm{r,c}} \tag{13}$$

to the long-time decay of the reactant population with the fit parameters $p_{\mathrm{r},0}$, κ and $p_{\mathrm{r,c}}$, where κ corresponds to the reaction rate. The population decay agrees well with the exponential fit as expected for a first-order transition. The reaction rate κ is further shown for ensembles of different temperature $k_\mathrm{B}T$ in Fig. 7(b) as green dots. The solid line refers to the Arrhenius' rate equation

$$\kappa(k_\mathrm{B}T) = \kappa_\infty \mathrm{e}^{-\Delta E_\mathrm{eff}/k_\mathrm{B}T}\,, \tag{14}$$

where κ_∞ is the high-temperature limit of the reaction rate and E_eff denotes the effective height of the energy barrier. We obtain $\kappa_\infty = 3.138$ and $E_\mathrm{eff} = 0.154$ for the fit parameters.

IV. CONCLUSION

In this paper, we have introduced neural networks as a tool for approximating high-dimensional, time-dependent DSs in classical reaction dynamics. We have shown that with rather small effort in calculating points on the NHIM as the training set for a NN, the resulting network predicts the DS accurately over a broader domain along a continuous times interval. This holds true even in the more challenging case of increased dimensionality of the bath. The accuracy and exactness of the resulting DS reproduced by the NN has been verified through the observation of negligible recrossings and calculated rates that are in good agreement with theory. We anticipate that DSs can be obtained using neural networks in systems of even more degrees of freedom, although such an approach is clearly more challenging as the dimensionality grows beyond the three evidenced here.

ACKNOWLEDGMENT

The German portion of this collaborative work was partially supported by DFG. The US portion was partially supported by the National Science Foundation (NSF) through Grant No. CHE 1112067. AJ acknowledges the Alexander von Humboldt Foundation, Germany, for support through a Feodor Lynen Fellowship. MF is grateful for support from the Landesgraduiertenförderung of the Land Baden-Württemberg. This collaboration has also benefited from support by the people mobility programs of the European Union, and the Horizon 2020 research and innovation programme under grant agreement No. 734557.

[1] K. S. Pitzer, F. T. Smith, and H. Eyring, *The Transition State*, Special Publ. (Chemical Society, London, 1962) p. 53.
[2] P. Pechukas, Annu. Rev. Phys. Chem. **32**, 159 (1981).
[3] B. C. Garrett and D. G. Truhlar, J. Phys. Chem. **83**, 1052 (1979).
[4] D. G. Truhlar, A. D. Issacson, and B. C. Garrett, "Theory of chemical reaction dynamics," (CRC Press, Boca Raton, FL, 1985) pp. 65–137.
[5] G. A. Natanson, B. C. Garrett, T. N. Truong, T. Joseph, and D. G. Truhlar, J. Chem. Phys. **94**, 7875 (1991).
[6] D. G. Truhlar, B. C. Garrett, and S. J. Klippenstein, J. Phys. Chem. **100**, 12771 (1996).
[7] D. G. Truhlar and B. C. Garrett, J. Phys. Chem. B **104**, 1069 (2000).
[8] T. Komatsuzaki and R. S. Berry, Proc. Natl. Acad. Sci. U.S.A. **98**, 7666 (2001).

[9] H. Waalkens, R. Schubert, and S. Wiggins, Nonlinearity **21**, R1 (2008).

[10] T. Bartsch, J. M. Moix, R. Hernandez, S. Kawai, and T. Uzer, Adv. Chem. Phys. **140**, 191 (2008).

[11] S. Kawai and T. Komatsuzaki, Phys. Rev. Lett. **105**, 048304 (2010).

[12] R. Hernandez, T. Bartsch, and T. Uzer, Chem. Phys. **370**, 270 (2010).

[13] O. Sharia and G. Henkelman, New J. Phys. **18**, 013023 (2016).

[14] A. Junginger, P. L. Garcia-Muller, F. Borondo, R. M. Benito, and R. Hernandez, J. Chem. Phys. **144**, 024104 (2016).

[15] A. Junginger, L. Duvenbeck, M. Feldmaier, J. Main, G. Wunner, and R. Hernandez, J. Chem. Phys. **147**, 06401 (2017).

[16] E. Pollak and P. Pechukas, J. Chem. Phys. **69**, 1218 (1978).

[17] P. Pechukas and E. Pollak, J. Chem. Phys. **71**, 2062 (1979).

[18] R. Hernandez and W. H. Miller, Chem. Phys. Lett. **214**, 129 (1993).

[19] R. Hernandez, J. Chem. Phys. **101**, 9534 (1994).

[20] T. Uzer, C. Jaffé, J. Palacián, P. Yanguas, and S. Wiggins, Nonlinearity **15**, 957 (2002).

[21] H. Teramoto, M. Toda, and T. Komatsuzaki, Phys. Rev. Lett. **106**, 054101 (2011).

[22] C.-B. Li, A. Shoujiguchi, M. Toda, and T. Komatsuzaki, Phys. Rev. Lett. **97**, 028302 (2006).

[23] H. Waalkens and S. Wiggins, J. Phys. A **37**, L435 (2004).

[24] U. Çiftçi and H. Waalkens, Phys. Rev. Lett. **110**, 233201 (2013).

[25] J. Murdock, *Normal Forms and Unfoldings for Local Dynamical Systems* (Springer, New York, 2002).

[26] C. Mendoza and A. M. Mancho, Phys. Rev. Lett. **105**, 038501 (2010).

[27] A. M. Mancho, S. Wiggins, J. Curbelo, and C. Mendoza, Commun. Nonlinear Sci. Numer. Simul. **18**, 3530 (2013).

[28] T. Bartsch, T. Uzer, and R. Hernandez, J. Chem. Phys. **123** (2005).

[29] T. Bartsch, R. Hernandez, and T. Uzer, Phys. Rev. Lett. **95** (2005).

[30] T. Bartsch, T. Uzer, J. M. Moix, and R. Hernandez, J. Chem. Phys. **124**, 244310 (2006).

[31] G. T. Craven, T. Bartsch, and R. Hernandez, Phys. Rev. E **89**, 040801(R) (2014).

[32] G. T. Craven, T. Bartsch, and R. Hernandez, J. Chem. Phys. **141**, 041106 (2014).

[33] G. T. Craven, T. Bartsch, and R. Hernandez, J. Chem. Phys. **142**, 074108 (2015).

[34] S. Kawai and T. Komatsuzaki, J. Chem. Phys. **131**, 224505(1) (2009).

[35] M. Feldmaier, A. Junginger, J. Main, G. Wunner, and R. Hernandez, Chem. Phys. Lett. **687**, 194 (2017).

[36] B. K. Carpenter, G. S. Ezra, S. C. Farantos, Z. C. Kramer, and S. Wiggins, J. Phys. Chem. B **XX**, XXXX (2018).

[37] T. B. Blank, S. D. Brown, A. W. Calhoun, and D. J. Doren, J. Chem. Phys. **103**, 4129 (1995).

[38] J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007).

[39] J. Behler, J. Chem. Phys. **134**, 074106 (2011).

[40] J. Cui and R. V. Krems, J. Phys. B **49**, 224001 (2016).

[41] R. A. Vargas-Hernández, Y. Guan, D. H. Zhang, and R. V. Krems, arXiv preprint arXiv:1711.06376 (2017).

[42] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld, Phys. Rev. Lett. **108**, 058301 (2012).

[43] J. Cui and R. V. Krems, Phys. Rev. Lett. **115**, 073202 (2015).

[44] J. Cui, Z. Li, and R. V. Krems, J. Chem. Phys. **143**, 154101 (2015).

[45] F. A. Faber, A. Lindmaa, O. A. Von Lilienfeld, and R. Armiento, Phys. Rev. Lett. **117**, 135502 (2016).

[46] B. Huang and O. A. von Lilienfeld, arXiv preprint arXiv:1707.04146 (2017).

[47] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016) `http://www.deeplearningbook.org`.

[48] M. A. Nielsen, *Neural Networks and Deep Learning* (Determination Press, 2015) `http://neuralnetworksanddeeplearning.com/`.

[49] K.-I. Funahashi, Neural Networks **2**, 183 (1989).