

# Invariant Manifolds and Rate Constants in Driven Chemical Reactions

Matthias Feldmaier,<sup>†</sup> Philippe Schraft,<sup>†</sup> Robin Bardakcioglu,<sup>†</sup> Johannes Reiff,<sup>†</sup>  
Melissa Lober,<sup>†</sup> Martin Tschöpe,<sup>†</sup> Andrej Junginger,<sup>†,§</sup> Jörg Main,<sup>†</sup> Thomas  
Bartsch,<sup>‡</sup> and Rigoberto Hernandez\*,<sup>¶</sup>

<sup>†</sup>*Institut für Theoretische Physik 1, Universität Stuttgart, 70550 Stuttgart, Germany*

<sup>‡</sup>*Centre for Nonlinear Mathematics and Applications, Department of Mathematical Sciences, Loughborough University, Loughborough LE11 3TU, United Kingdom*

<sup>¶</sup>*Department of Chemistry, Johns Hopkins University, Baltimore, Maryland 21218, USA*

<sup>§</sup>*Present address: Machine Learning Team at ETAS GmbH (Bosch Group).*

E-mail: r.hernandez@jhu.edu

## Abstract

Reaction rates of chemical reactions under nonequilibrium conditions can be determined through the construction of the normally hyperbolic invariant manifold (NHIM) [and moving dividing surface (DS)] associated with the transition state trajectory.

Here, we extend our recent methods by constructing points on the NHIM accurately even for multidimensional cases. We also advance the implementation of machine learning approaches to construct smooth versions of the NHIM from a known high-accuracy set of its points. That is, we expand on our earlier use of neural nets, and introduce the use of Gaussian process regression for the determination of the NHIM. Finally, we compare and contrast all of these methods for a challenging two-dimensional model barrier case so as to illustrate their accuracy and general applicability.

## 1 Introduction

The framework of transition state theory (TST) provides a powerful basis for the qualitative and quantitative description of chemical reactions. When their dynamics can be described

by a Born-Oppenheimer potential driven by a classical equation of motion, then TST reduces the dynamical calculation to a geometric one involving the identification of a barrier region that separates reactants and products. The position of the barrier is typically marked by a saddle point of rank 1 that has exactly one unstable direction which coincides with the *reaction coordinate* at that point. The remaining degrees of freedom are locally stable and are associated with the *bath coordinates*. In this paper, we aim to address activated processes with arbitrary dimensionality but restricted to reactions that can be characterized by a one-dimensional reaction coordinate. We do not solve this class of problems in fullest generality, but do make progress in treating systems with bath coordinates of dimensions higher than one, addressing both theoretical and numerical challenges to the computation of the dividing surface.

TST rests on the identification of a dividing surface (DS) in the barrier region which separates reactants from products.<sup>1–5</sup> It is exact, if the DS is crossed once and only once by each reactive trajectory. Recrossings of the DS lead to an overestimation of the rate. Advances in the determination of a recrossing-free DS will therefore impact a broad range of problems be-

yond those of chemical reactions in which the overall process can be described as an activated process that is primarily characterized by a single, albeit curved, reaction coordinate in the presence of a bath. Examples abound in atomic physics,<sup>6</sup> solid state physics,<sup>7</sup> cluster formation,<sup>8,9</sup> diffusion dynamics,<sup>10,11</sup> cosmology,<sup>12</sup> celestial mechanics,<sup>13,14</sup> and Bose-Einstein condensates,<sup>15–19</sup> to name a few.

Of course, the primary aim for the current work is the construction of accurate rates in chemical reactions that take place in solution, perhaps under nonequilibrium conditions, and for which we ultimately want to achieve some level of control. The basic (or naive) theory uses a planar dividing surface to determine the flux, but it is approximate because it is invariably recrossed by trajectories that go from reactants to products.<sup>1–4</sup> It is also an upper bound, and leads to a variational transition state theory<sup>20,21</sup> which optimizes the dividing surface by way of minimizing recrossings. The development of semiclassical transition state theory<sup>22–26</sup> taking advantage of Keck’s phase space representation of TST<sup>27</sup> hinged on the development of good action-angle variables associated with the DS. The use of perturbation theory to construct these objects was found in parallel to work in dynamical systems theory addressing activated escape.<sup>28,29</sup> The theory was then generalized for chemical reactions under time-dependent conditions —arising from driving, noise, or both.<sup>30,31</sup> Applications to chemical reactions have included  $\text{H} + \text{H}_2$ ,<sup>32,33</sup>  $\text{LiCN}$ ,<sup>34</sup> and ketene isomerization.<sup>35</sup> While these are interesting cases, they do not represent the bulk of chemical reactions which are generally higher dimensional and which take place in more complex environments. This work is thus focused on advancing the theoretical and computational machinery that have the potential of addressing such challenging systems.

In a multidimensional autonomous Hamiltonian system, a recrossing-free DS is attached to the normally hyperbolic invariant manifold (NHIM). The latter contains all trajectories that are trapped in the saddle region both forward and backward in time. It can be constructed using a normal form expan-

sion.<sup>13,24,32,36–42</sup> For a multidimensional time-dependent system that is driven by an external field or subject to thermal noise, the question arises whether the NHIM can be generalized in such a way that it can still be used to construct a recrossing-free DS. Recent successful constructions<sup>43–45</sup> of such a DS suggests that it is indeed possible to do so. The purpose of this paper is to clarify the geometry that underlies this construction, and to present a readily applicable procedure to calculate rate constants of time-dependent, driven and multidimensional systems with a rank-1 saddle and an arbitrary number of bath modes. As this work is focused on advancing the methods so as to ultimately treat higher dimensional chemical reactions, numerical results are restricted to model systems —inspired by chemical systems— useful for verifying the theory.

The computational task is challenging because of the dimensionality of the objects: In an  $n$ -dimensional system with one reaction coordinate and  $n-1$  bath coordinates, the phase space has dimension  $2n$ . The DS is a time-dependent and non-trivially curved hypersurface of dimension  $2n-1$ , and the NHIM is a surface of dimension  $2n-2$ . Previous work has shown that attaching a DS to the NHIM, whose dimension is lower, can yield a recrossing-free DS if the reaction coordinate is unbound.<sup>44,45</sup>

An individual point on the NHIM can be computed with the algorithm discussed in Ref. 46. However, to construct a DS that covers a sufficiently large region in phase space and time, a vast number of points is required. Even with an efficient algorithm, computing the required number of points on the NHIM is a daunting task. Consequently, a continuous interpolation between the finite collection of high-accuracy —and therefore computationally expensive— points on the DS in space and time is needed to determine the reactivity of a given trajectory. We present two tools developed in the context of machine learning to approximate any function between isolated points in arbitrary high dimensions. It is notable that Pozun *et al.*<sup>47</sup> earlier implemented a support-vector machine to identify a DS between reactants and products on a time-independent

potential energy surface. Here, we focus on time-dependent potentials, often coupled to a bath, in which the associated NHIM is consequently time-dependent. First, a neural network (NN)<sup>48</sup> can be trained on an arbitrary set of points located on the NHIM. Once trained, these networks provide fast access to any point needed for calculating rates in time-dependent driven systems.<sup>45</sup> We also present an alternative machine learning algorithm based on Gaussian process regression (GPR).<sup>49,50</sup> We compare these two algorithms and discuss the relative advantages and disadvantages in their application to TST.

Perturbative<sup>51–55</sup> and nonperturbative<sup>35,43,56–58</sup> constructions of a recrossing-free DS in time-dependent systems rely on the concept of the transition state (TS) trajectory:<sup>30,59</sup> a unique trajectory bound to the vicinity of the saddle for all time. In systems with more than one degree of freedom, such a TS trajectory is unique only if the system is dissipative with the oscillations of the bath modes damped asymptotically. If the reactive system is Hamiltonian—that is, when in the absence of damping,—all trajectories on the NHIM are trapped near the barrier for all time and can therefore serve as TS trajectories. The consequences of the non-uniqueness of the TS trajectory were avoided in Ref. 60 by an arbitrary choice of one of the trapped trajectories as “the” TS trajectory. The DS constructed there and in Refs. 44,45 also avoids such arbitrary choices because it considers the NHIM itself, rather than a particular trajectory on it. For multidimensional time-periodically driven Hamiltonian systems, we suggest identification of the TS trajectory to be restricted to the one trajectory, among all the bound trajectories on the NHIM, that is periodic. This reduces to the usual result in one dimension, and is no longer arbitrary in higher dimensions as it is now selected by a characteristic property of the system.

The outline of the paper is as follows: In Sec. 2, we start with a brief overview describing how rates in time-dependent systems with a rank-1 saddle are calculated using the NHIM and the associated DS. This suggests a restricted definition of the TS trajectory stated

above. In Sec. 3, we present a fast and robust algorithm for calculating single points on the NHIM. Sec. 4 summarizes two interpolation algorithms based on machine learning, and provides an analysis of the respective advantages and disadvantages. In Sec. 5, we apply both methods to calculate rate constants for a two-dimensional system.

## 2 TST in time-dependent systems

A possible first step in obtaining the time-dependent representation of the DS is the determination of individual points on this high-dimensional object. As outlined in Ref. 44, in time-dependent systems it suffices to calculate the *normally hyperbolic invariant manifold* (NHIM) because the DS in time-dependent systems is attached to it in a prescribed way. In this section, we describe the former as a fundamental geometric object and demonstrate how it can be used to obtain a DS. We start by considering a one-dimensional system in which there are no bath coordinates. We then present the *binary contraction method* for finding the NHIM based on the geometric properties of phase space in the vicinity of a rank-1 saddle. Furthermore, we describe how the insights gained in the one-dimensional setting can be generalized to higher dimension. Finally, we will discuss how a unique TS trajectory can be defined in a multidimensional system.

### 2.1 The role of the NHIM in TST

In a one-dimensional system, a rank-1 saddle is simply a maximum of the potential. The relevant phase space structures in its neighborhood are shown in Fig. 1: Trajectories that approach the energy barrier with sufficiently high energy (from either side) will cross the barrier, and others will not. Its phase space can be separated into four different regions as characterized by where particles came from and where they go to. The dividing lines between these regions are the stable  $\mathcal{W}_s$  and unstable  $\mathcal{W}_u$  manifolds. They contain trajectories that are trapped near

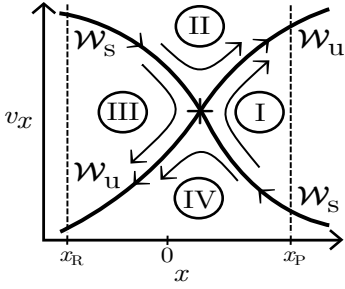


Figure 1: Phase space  $(x, v_x)$  close to the boundary point (+) of the stable  $\mathcal{W}_s$  and unstable  $\mathcal{W}_u$  manifold. The areas of non-reactive trajectories (I) and (III) are separated from the areas with reactive trajectories (II) and (IV) by the manifolds. A reaction can either take place from the reactant side  $x_R$  to the product side  $x_P$  or vice versa. The arrows illustrate the heading of the trajectories close to the boundary point.

**Table 1: Nomenclature of the different reactive (II, IV) and non-reactive (I, III) areas in phase space separated by the two manifolds. The particle leaves the neighborhood of the saddle at the exit point  $x_{\text{exit}}^{\text{b,f}}$  when propagated in backward (b) and forward (f) time, respectively.**

Area	$x_{\text{exit}}^{\text{b}}$	$x_{\text{exit}}^{\text{f}}$
I	$x_P$	$x_P$
II	$x_R$	$x_P$
III	$x_R$	$x_R$
IV	$x_P$	$x_R$

the saddle in forward ( $\mathcal{W}_s$ ) or backward ( $\mathcal{W}_u$ ) time, respectively.

A trajectory starting at a given phase space point is propagated in forward (f) and backward (b) time until it leaves the saddle region given by  $x_R \leq x \leq x_P$  with a suitably chosen  $x_R$  at the reactant side and  $x_P$  at the product side of the reaction coordinate. The initial point is then assigned to a region, depending on whether it leaves the saddle region at  $x_R$  or  $x_P$  in forward or backward time, as explained in Table 1. Note that numerically, the trapping cannot, in general, be observed because the dynamics near the saddle is unstable. Small numerical errors cause an otherwise trapped trajectory to leave the saddle region in either time

direction.

The closures of stable and unstable manifolds intersect in a point (see black cross in Fig. 1). When the potential is also time-independent, this intersection is a fixed point. In time-dependent (or driven) cases, it becomes a trajectory that is trapped in both forward and backward time. This trajectory is referred to as the TS trajectory.<sup>30,51–59</sup> In one-dimensional systems, it is unique. In multi-dimensional systems, the initial conditions of trajectories trapped in both forward and backward time form the time-dependent multidimensional NHIM.<sup>61–63</sup> The latter reduces to a single trajectory in one-dimensional systems, and so the unique TS trajectory in one dimension can also be called the NHIM.

For a time-independent, one-dimensional system, the NHIM is located exactly at the barrier top. In a time-dependent system, all structures shown in Fig. 1 depend on the initial time at which the trajectories are started, i.e., the stable and unstable manifolds, as well as the NHIM, are themselves time-dependent. The NHIM can be expected to reside close to the saddle, but it will in general not coincide with it. If the external driving is periodic in time, the NHIM is a periodically moving manifold with the same period as the external driving force (see blue, dashed circle in Fig. 3(d), calculated for the example of Eq. (3)).

So far, we have assumed open reactant and product basins. In the more typical case of closed basins —as in e.g. the model potential of Ref. 64, or potentials of real systems, for instance LiCN<sup>34,65,66</sup> or ketene<sup>35,67–69</sup>— the requirement for trajectories to be *recrossing-free* often needs to be restricted to mean *locally recrossing-free*, i.e., a particle that enters the neighborhood of the saddle will cross the DS no more than once before it leaves the saddle region. It may, however, reenter this region and recross the DS at a later time. For further information about these *global recrossings* and the degree to which TST can remain exact despite them, see Ref. 64.

## 2.2 Recrossing-free DS in time-dependent systems

For a system with  $n$  degrees of freedom, the separation of the  $2n$ -dimensional phase space into reactant and product regions requires a surface of dimension  $2n - 1$ . To distinguish reactant and product regions for any rank-1 saddle in an  $n$ -dimensional energy surface we use an appropriate coordinate system where  $x$  approximates the reaction coordinate, i.e., the unstable direction of the saddle, and  $\mathbf{y}$  the  $n - 1$  remaining (bath) coordinates, given by the stable directions of the saddle. The corresponding velocities are denoted by  $v_x$  and  $\mathbf{v}_y$ , respectively. We will construct a dividing surface at an instantaneous reaction coordinate  $x^{\text{DS}}(\mathbf{y}^{\text{p}}, \mathbf{v}_y^{\text{p}}, t)$  that depends on the bath coordinates, their velocities, and on time. A phase space point  $(x, \mathbf{y}, v_x, \mathbf{v}_y)$  is classified as lying on the reactant or product side of the barrier at time  $t$  depending on whether it fulfills

$$\begin{aligned} x < x^{\text{DS}}(\mathbf{y}^{\text{p}}, \mathbf{v}_y^{\text{p}}, t) &\rightarrow \text{reactant, or} \\ x > x^{\text{DS}}(\mathbf{y}^{\text{p}}, \mathbf{v}_y^{\text{p}}, t) &\rightarrow \text{product.} \end{aligned} \quad (1)$$

While propagating an ensemble of particles, each particle can be classified as reactant or product at any time according to Eq. (1). When it crosses the moving DS, it reacts and contributes to the reaction of the time-dependent system.

To satisfy the recrossing-free requirement, this high-dimensional DS is generally non-trivially curved in a manner that depends on the details of the dynamics. The representation (1) of the DS assumes that no curvature in the  $v_x$  direction is required. This assumption is inspired by the phase space plot in Fig. 3, where such choice of a DS appears as a vertical line going up from the NHIM. Reactive trajectories cross the DS transversely and exactly once, whereas non-reactive trajectories do not cross it. This statement is trivially true in a harmonic system. It describes a qualitative property that is robust under perturbations of either the system or the dividing surface. Many other surfaces that are sufficiently close to the vertical line will also be recrossing-free; the ex-

act choice of DS is therefore not critical. We take advantage of this non-uniqueness by postulating that the DS has the form (1) and that its location  $x^{\text{DS}} = x^{\text{NHIM}}$  coincides with that of the NHIM. These assumptions single out a unique DS and also provide a means to compute it because the NHIM is amenable to numerical computation, as discussed below in Sec. 3.

With this simplification, our choice of a recrossing-free DS, independent on the reactive velocity  $v_x$ , can be justified via a confirmation of the absence of unwanted recrossings in dynamical simulations, as shown in Sec. 5 for a particular two-dimensional model system. However, in more general cases when particles cross the DS far away from the NHIM, this simplification might lead to unwanted recrossings (see, e. g. Refs. 40,70–72), as the assumption embodied in Eq. (1) has its limits. There is no general proof that a recrossing-free DS of the form (1) should exist, and exceptions are known.<sup>70</sup> If the invariant manifolds are strongly deformed, e. g. such that they cross the chosen DS, then the fate of a trajectory starting on the DS  $x = x^{\text{DS}}(\mathbf{y}, \mathbf{v}_y, t)$  will depend on  $v_x$ , and the representation (1) becomes impossible. However, this is expected only for strongly non-linear systems or very high velocities, so that the representation (1) is adequate for many systems of practical relevance. For a given system, one can always numerically confirm whether a DS is truly recrossing-free. This may be done by propagating a large number of trajectories with initial conditions in the vicinity of the moving saddle, and subsequent monitoring of the crossings across the constructed DS, as has been done successfully for the model system discussed in Refs. 44,45.

If the underlying potential has open reactant and product basins, a particle that crosses a recrossing-free DS will never return because it escapes the saddle region forever. Assuming that our choice of a DS independent on the reactive velocity  $v_x$  is sufficiently accurate The propagation of a reacting particle can be stopped when a particle crosses a space determined by  $x = x^{\text{DS}}$  while calculating rates in open systems. In systems with closed reactant or product basins, particles can be re-

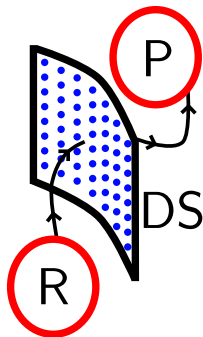


Figure 2: Simplified sketch of the phase space of a chemical reaction. The blue dots represent pre-calculated points on the DS. A reacting particle crosses the DS, a curved, co-dimension one manifold separating reactants (R) and products (P) in phase space. Here, we illustrate the problem, that such a reacting trajectory will in general not cross the DS at the pre-calculated points but somewhere in between. Note, that these manifolds are generally high-dimensional objects embedded in phase space and this figure has to be understood as highly schematic.

flected globally at the boundaries of the basins and may therefore cross the space determined by  $x = x^{\text{DS}}$  infinitely many times.<sup>34,35,64–69</sup> In this case, the propagation cannot be stopped and particles can react from one side of the saddle to the other multiple times. Still, the time-dependent classification in reactant and product is always given due to the local recrossing-free property of the DS presented here (see Sec. 2.2 for further discussion on how to construct the DS and its properties).

Reacting particles can, in principle, cross the DS at some previously unknown values of the bath coordinates and velocities in between the pre-calculated points, and at a previously unknown time  $t$  (schematically visualized as trajectory in Fig. 2), while the DS itself is a time-dependently moving object. To obtain a rate from the simulation of a large number of trajectories, one therefore needs to evaluate the location  $x^{\text{DS}}(\mathbf{y}, \mathbf{v}_y, t)$  of the DS for a wide range of the parameters  $\mathbf{y}$ ,  $\mathbf{v}_y$ ,  $t$ , and the parameter space will typically have high dimension. However, even the calculation of  $x^{\text{DS}}(\mathbf{y}, \mathbf{v}_y, t)$  at single points is challenging and numerically

expensive. Indeed, it is hopeless to calculate  $x^{\text{DS}}(\mathbf{y} = \mathbf{y}^p, \mathbf{v}_y = \mathbf{v}_y^p, t)$  from scratch for every position  $(\mathbf{y}^p, \mathbf{v}_y^p)$  on the trajectory of every particle  $p$  at the respective time  $t$ , while propagating a large ensemble. Consequently, it is critical to obtain a representation of the function  $x^{\text{DS}}(\mathbf{y}, \mathbf{v}_y, t)$ , that describes the dependence of the DS on all its arguments over a sufficiently large range, and which is easy to evaluate.

We obtain a computationally efficient representation of the DS in two steps: First, a high-accuracy algorithm is used to obtain a small, but incomplete, number of representative points on the DS. In Sec. 3, we present an algorithm that is able to efficiently calculate these points, based on the geometric properties of phase-space in the saddle region. Second, a smooth representation of the function  $x^{\text{DS}}(\mathbf{y}, \mathbf{v}_y, t)$  is obtained through an efficient extrapolation of the finite points obtained in the first step. This requirement can sometimes be achieved through interpolation and fitting of expected forms of the potential. To avoid such numerics and to take advantage of the limited data set presently available, we have recently implemented machine learning approaches which should, in principle, be applicable to arbitrarily high dimension.<sup>45</sup> The use of these methods to obtain a time-dependent description of the full DS is summarized, extended and assessed in Sec. 4.

### 2.3 Obtaining rate constants from reactant decay

Through TST one aims to acquire rate constants for a chemical reaction from a reactant state over an energy barrier to a product state. A typical approach for obtaining rate constants<sup>73</sup> is to propagate an ensemble of particles, identify only those trajectories from reactants to products that cross the DS such as that illustrated in Fig. 2, sum their flux, and divide by the population of reactant particles. In the TST approximation the reacting flux is obtained by summing the instantaneous flux across the entire DS. Rates thus obtained are exact if and only if the DS is free of recrossings.

To obtain rate constants, we monitor the

number of trajectories that remain reactants over time  $N_{\text{react}}(t)$  before crossing the moving DS. For large times, when all trajectories have left the barrier region, it approaches the number of non-reactive particles  $N_{\infty}$ . After the initial transient behavior has decayed, this limit is approached exponentially

$$N_{\text{react}}(t) - N_{\infty} \propto \exp(-kt), \quad (2)$$

where  $k$  is the rate constant of this reaction.<sup>3,74,75</sup> Deviations from exponential behavior also arise at long times when the number of remaining trajectories becomes too small to be statistically meaningful.

### 3 Obtaining single points on the NHIM

#### 3.1 Revealing phase space structures

The phase space structures discussed in Sec. 2.2 can be uncovered through an increasing array of tools, including the ones reviewed in this section. For simplicity, we will illustrate them using a one-dimensional model system for a simple chemical reaction with a periodically moving barrier and open reactant and product channels. Specifically, we consider the motion of a particle of unit mass under the influence of the time-dependent potential

$$V(x, t) = E_b \exp \left\{ -[x - \hat{x} \sin(\omega_x t)]^2 \right\}. \quad (3)$$

Here,  $E_b$  is the height of a Gaussian barrier oscillating along the  $x$  axis with frequency  $\omega_x$  and amplitude  $\hat{x}$ . We use dimensionless units in which we set  $E_b = 2$ ,  $\hat{x} = 0.4$ , and  $\omega_x = \pi$ .

A tool to resolve the manifolds in phase space is the Lagrangian descriptor (LD).<sup>76</sup> In the context of TST, the LD at position  $\mathbf{x}_0$  in the coordinate space, velocity  $\mathbf{v}_0$ , and time  $t_0$ , is defined as<sup>43,56,58,76</sup>

$$\mathcal{L}(\mathbf{x}_0, \mathbf{v}_0, t_0) = \int_{t_0 - \tau}^{t_0 + \tau} \|\mathbf{v}(t)\| dt. \quad (4)$$

The LD measures the arc length of the tra-

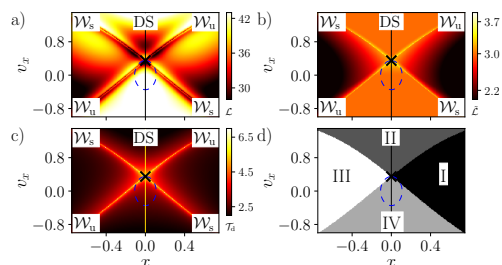


Figure 3: Phase space of the system introduced in Eq. (3). Several methods are visualized to reveal the stable ( $\mathcal{W}_s$ ) and unstable ( $\mathcal{W}_u$ ) manifolds attached to the saddle. As in Fig. 1, their intersection (the NHIM at a given time) is marked with a black cross, and the DS attached to it is a vertical solid line. The blue dashed curve is the union of all NHIMs points (and therefore the TS-trajectory) for the full time-dependence of Eq. (3). The contours in the panels correspond to (a)  $\mathcal{L}$  as defined in Eq. (4), (b)  $\tilde{\mathcal{L}}$  according to Eq. (5), (c) the time-descriptor  $\mathcal{T}_d$  (see Eq. (6)), and (d) reactive and non-reactive regions according to Table 1.

jectory  $\mathbf{x}(t)$  that passes through  $(\mathbf{x}_0, \mathbf{v}_0)$  at time  $t = t_0$ . The trajectory is considered forward and backward in time over the interval  $[t_0 - \tau; t_0 + \tau]$ , and the parameter  $\tau$  is chosen such that the relevant time scale of the system is covered. In the case of open reactant and product basins, the stable  $\mathcal{W}_s$  and unstable  $\mathcal{W}_u$  manifolds associated with the energy barrier should be revealed by local minima of the LD in phase space, but are sometimes obfuscated by ditch-like structures, as seen in Fig. 3(a). Trajectories starting near the stable and unstable manifolds generally have low LDs because they are bound to the saddle region for a finite time. One ditch—that is, a deep minimum—is found in the forward time component, indicating the location of the stable manifold  $\mathcal{W}_s$  and the other ditch is found in the backward time component, indicating the unstable manifold  $\mathcal{W}_u$ .

In the case of closed reactant and product basins, the LD defined in Eq. (4) results in a fractal-like structure in phase space that also contains both manifolds, but obfuscates them even more.<sup>64</sup> Nevertheless,  $\mathcal{W}_s$  and  $\mathcal{W}_u$  can be revealed if LDs are computed not for a fixed time  $\tau$ , but rather until they leave the saddle

region  $x_R \leq x \leq x_P$ :

$$\tilde{\mathcal{L}}(\mathbf{x}_0, \mathbf{v}_0, t_0) = \int_{t_b(\mathbf{x}_0, \mathbf{v}_0, t_0)}^{t_f(\mathbf{x}_0, \mathbf{v}_0, t_0)} \|\mathbf{v}(t)\| dt, \quad (5)$$

$$x(t_{b,f}(\mathbf{x}_0, \mathbf{v}_0, t_0)) = x_R \text{ or } x_P, \quad \text{for } t_b < t_0 < t_f.$$

As shown in Fig. 3(b), the manifolds are then clearly revealed by local maxima in  $\tilde{\mathcal{L}}$ .

The shift from minima in  $\mathcal{L}$  to maxima in the modified  $\tilde{\mathcal{L}}$  is attributed to the fact that trajectories close to the stable and unstable manifold are integrated over longer times, while the respective particles are moving in the saddle region. With this observation, one can find a simpler and arguably more reliable approach to reveal the manifolds. By simply tracking the time a particle spends in the saddle region, we alleviate the need to compute the arc length of the trajectory. Following the nomenclature of the LD, we refer to these times as time descriptor (TD):

$$\begin{aligned} \mathcal{T}_d(\mathbf{x}_0, \mathbf{v}_0, t_0) &\equiv t_f(\mathbf{x}_0, \mathbf{v}_0, t_0) - t_b(\mathbf{x}_0, \mathbf{v}_0, t_0), \\ \mathcal{T}_{d,f}(\mathbf{x}_0, \mathbf{v}_0, t_0) &\equiv t_f(\mathbf{x}_0, \mathbf{v}_0, t_0) - t_0, \\ \mathcal{T}_{d,b}(\mathbf{x}_0, \mathbf{v}_0, t_0) &\equiv t_0 - t_b(\mathbf{x}_0, \mathbf{v}_0, t_0). \end{aligned} \quad (6)$$

The closer a particle is to  $\mathcal{W}_s$  in forward time or to  $\mathcal{W}_u$  in backward time, the longer it will stay in the saddle region in the respective time component. Consequently, the stable and the unstable manifold are again revealed by the maxima of the TD, as seen in Fig. 3(c). Note that since trajectories on the stable and unstable manifolds are trapped in the barrier region for infinite time, both  $\tilde{\mathcal{L}}$  and  $\mathcal{T}_d$  may diverge, although it is unlikely when propagating the trajectories numerically.

To calculate the  $v_x$ -position of the two manifolds in phase space for a given  $x$ -position (reaction coordinate), one can search for the appropriate extremum of the LD, the modified LD, or the TD. Using what we refer to as the standard LD is perhaps the least robust of these methods because the LD has several local minima in addition to the primary minimum indicating the manifold, see Fig. 4(a). The modified LD and the TD do not show such substructure and have clear maxima for both stable and unstable manifolds, see Fig. 4(b) – (d). However, because of

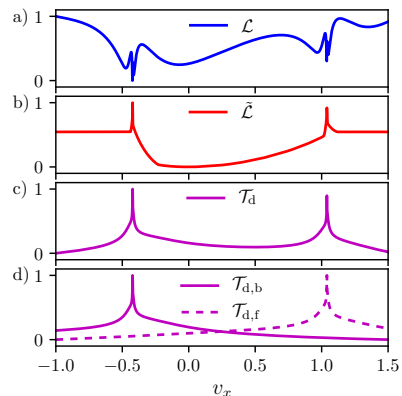


Figure 4: One-dimensional view along  $v_x$  of the data in Fig. 3 at  $x = -0.4$ . The LD without cutoff is displayed in (a), the LD with cutoff in (b) and the TD in (c). Subfigure (d) shows the individual parts of the TD for propagation solely in backward respectively forward time. All curves have been normalized.

its simpler definition the TD seems to be the method of choice as long as one can clearly define a saddle region. With all of these methods, the integration in forward and backward time can also be performed separately to reveal the two manifolds individually, see Fig. 4(d). Thus, numerical ambiguities between stable and unstable manifolds near their intersection can be avoided.

### 3.2 1D: Finding the NHIM

With the stable and unstable manifold revealed by the LDs or TDs, it is now possible to find the NHIM. Since the NHIM is given by the intersection of the stable and unstable manifold, it can be computed by finding coordinates where their distance is zero. This can be accomplished with a root search that finds the value of the reaction coordinate  $x$  for which the difference between the velocities  $v_x$  of the stable and the unstable manifold is zero. However, this procedure is very costly as it nests an extreme value search of the LDs or TDs within a root search for the distance between these extreme values. Moreover, the computation of each of the many LD or TD values underlying this search is itself expensive because it requires the integration of a full trajectory.

An efficient alternative for obtaining the



NHIM is the binary contraction method,<sup>46</sup> which is based on the classification of trajectories in Fig. 3(d). The algorithm starts with a quadrangle whose vertices represent the four different regions, see Fig. 5. A candidate vertex is then associated with the midpoint of one of the edges of the quadrangle, and a trajectory is initiated to determine which region it belongs to. The corresponding vertex is then replaced, and the area of the quadrangle is reduced. This process is repeated for all vertices until the quadrangle converges to the NHIM with the desired accuracy—that is, sufficiently small area. Additional details of the algorithm, including an explanation on how to find good initial conditions and the treatment of exceptional cases, can be found in Ref. 46.

Thus the binary contraction method can be used to find the NHIM (or the TS trajectory) with high precision, while propagating considerably fewer trajectories than the optimization method described above. For example, in the one-dimensional model potential of Eq. (3) with periodic external driving, this leads to a periodic orbit (the TS trajectory), as demonstrated by the blue dashed circles in Fig. 3.

### 3.3 The NHIM in higher dimensions

In systems with  $n > 1$  degrees of freedom, the situation is qualitatively similar, but it is more challenging because all phase space structures have higher dimensions. As already established in Sec. 2.2, the NHIM has dimension  $2n - 2$  in phase space. Consequently, its associated stable and unstable manifolds both have dimension  $2n - 1$ . The time-dependence in the potential, either because of the external driving or the noise, adds additional complexity which makes these manifolds time-dependent.

A point in phase space can be characterized by the reaction coordinates  $x$  and  $v_x$  and the bath coordinates  $\mathbf{y}$  and  $\mathbf{v}_y$ . For any fixed values of the bath coordinates  $\mathbf{y}$  and  $\mathbf{v}_y$ , we obtain a two-dimensional section through phase space with coordinates  $x$  and  $v_x$ . We assume that in each of these sections the stable and unstable manifold intersect as shown in Fig. 1. In partic-

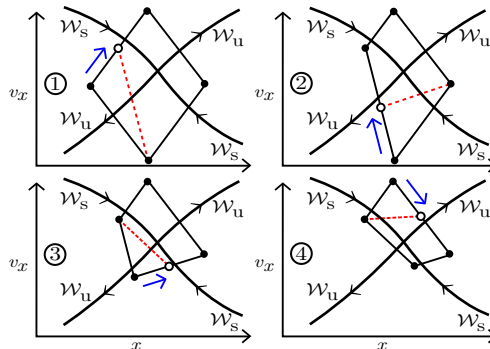


Figure 5: Illustration of the binary contraction method. The algorithm starts with a quadrangle having each of its vertices (black bullets) in one of the reactive and non-reactive areas described in Table 1. At each step, a new trajectory starting at the center between two of the vertices is calculated (white bullet) and classified to one of the areas. Afterwards the corresponding vertex of the quadrangle is replaced. Iteration of this procedure for each distance between adjacent vertices shrinks the quadrangle to the intersection of the stable  $\mathcal{W}_s$  and unstable  $\mathcal{W}_u$  manifold.

ular, each section contains a single point of the NHIM that lies at the intersection of the two reactive and two nonreactive regions. This condition provides a unique reaction coordinate and velocity  $(x^{\text{NHIM}}(\mathbf{y}, \mathbf{v}_y, t), v_x^{\text{NHIM}}(\mathbf{y}, \mathbf{v}_y, t))$  on the NHIM for each bath coordinate and time, which can be found by the binary contraction algorithm. Because these coordinates depend on  $2n - 2$  variables  $\mathbf{y}$  and  $\mathbf{v}_y$  in phase space, the NHIM has dimension  $2n - 2$ .

A possible counterexample arises in the simple limiting case of an uncoupled harmonic system. Therein, the point in the reactive phase space,  $x^{\text{NHIM}}$  and  $v_x^{\text{NHIM}}$  will be constant, independent of the bath coordinates  $\mathbf{y}$  and  $\mathbf{v}_y$ . In this case, it would not be possible to represent any of the bath modes as functions of  $x$  or  $v_x$ . Nevertheless, the construction above provides the correct phase space point because it does not rely on the association being injective.

A non-trivial multidimensional example is a two-dimensional model with a time-periodically moving barrier separating open reactant and product basins. It can be represented using an extension of the one-dimensional potential

of Eq. (3) to

$$V(x, y, t) = E_b \exp(-[x - \hat{x} \sin(\omega_x t)]^2) + \frac{\omega_y^2}{2} \left[ y - \frac{2}{\pi} \arctan(2x) \right]^2. \quad (7)$$

This potential describes a two-dimensional potential energy landscape that has minima in the bath coordinate along the curve  $y = (2/\pi) \arctan(2x)$ . A moving Gaussian barrier with height  $E_b$  is added over this stationary potential. It is oscillating along the  $x$  axis with frequency  $\omega_x$  and amplitude  $\hat{x}$ .  $\omega_y$  is the frequency of oscillations in the bath mode for a particle of unit mass. For simplicity, we again use dimensionless units in which the parameter are  $E_b = 2$ ,  $\omega_x = \pi$ ,  $\omega_y = 2$ , and  $\hat{x} = 0.4$ .

We choose  $x$  as the approximate reaction coordinate and  $y$  as the bath coordinate. Consequently, we calculate an  $(x, v_x)$ -slice for each set of the remaining coordinates  $(y, v_y)$  and for any time. Using the binary contraction method, we find the intersection  $(x, v_x)^{\text{NHIM}}(y, v_y, t)$ . Repeating this procedure for different values  $(y, v_y)$  on an equidistant grid for any time  $t$  and using an appropriate interpolation method (see below), we obtain the full NHIM of the system. It is a two-dimensional, time-dependently moving surface embedded in the four-dimensional phase space. A representation of this NHIM at  $t = 0.5$  is shown in Fig. 6 as a gray surface.

For certain points  $(y, v_y)$ , Fig. 6 shows sections of the stable and unstable manifolds that are attached to the NHIM. To obtain them, we consider a two-dimensional section of phase space with fixed values of  $y$  and  $v_x = v_x^{\text{NHIM}}(y, v_y)$ . The dynamical structure of this section is similar to that in Fig. 1, except that the axes are now  $x$  and  $v_y$ . The manifolds are obtained by finding the value of  $v_y$  at which, for given  $x$ , the TD is maximal. The stable and unstable manifolds intersect on the NHIM, as expected. Because they were obtained independently of the calculation that yields the NHIM, this observation also serves to confirm the reliability of the computation and the uniqueness of the NHIM.

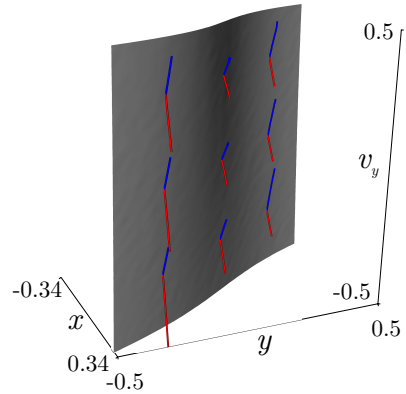


Figure 6: Gray surface:  $x^{\text{NHIM}}(y, v_y)$  for the potential (7) at  $t = 0.5$  with  $y$  and  $v_y$  varied on an equidistant grid in the plotted range. For  $(x, v_y)$ -slices at  $3 \times 3$  different values  $(y, v_x)$  the one-dimensional fiber of the stable manifold in each slice is given as a red line, the fiber of the unstable manifold as a blue line. Their intersections are located on the two-dimensional NHIM.

### 3.4 TS-trajectory in higher dimensions

In a one-dimensional system, the NHIM is a single trajectory that can be identified as the TS trajectory. A higher-dimensional NHIM, by contrast, is made up of infinitely many trajectories over time, all of which are trapped in the saddle region for all times. Thus a unique identification of the TS trajectory requires additional constraints beyond the minimal requirement that it is trapped within the saddle region.

In an autonomous system, the location of the DS is often marked by the naive TS—that is the saddle at the top of the barrier.

The TS is the point on some selected reaction path that crosses the DS and it is not necessarily the same as the naive TS but is often in its proximity. We further note that there exist exceptions to cases in which the TS is near the naive TS. This includes the TS structures far from the naive TS found by Gray and Davis<sup>77</sup> for the autonomous  $\text{HeI}_2$  system and, more recently, roaming reactions that are associated to a TS that is not necessarily close to a saddle.<sup>68,78–81</sup> Such cases are not addressed in the present work.

The TS trajectory is thus intended to gen-

eralize the fixed TS for driven systems. In a one-dimensional, time-periodically driven system, the TS trajectory is a periodic orbit. Following this property, we suggest that for a time-periodically driven, multidimensional Hamiltonian system the period-1 orbit trapped in the barrier region, if uniquely identified, should also be called the TS trajectory. Here, period-1 qualifies the orbit as one which is periodic and for which the ratio of its period to that of the external driving is 1.

To make this definition precise, we determine the Euclidean distance  $\Delta\gamma$  of two phase space coordinates  $\gamma = (\mathbf{x}, \mathbf{v})^T$  on the same trajectory, one time period  $T$  of the potential apart. In practice, it can be computed by propagating initial conditions given for a certain time  $t_0$  forward and backward in time by half a potential period  $T/2$ . We perform this computation in such a way to avoid errors accumulated by propagating trajectories near unstable manifolds over long times. A trajectory with  $\Delta\gamma = |\gamma(t_0 + T/2) - \gamma(t_0 - T/2)| = 0$  is a periodic orbit. We identify such a periodic orbit as the TS trajectory if it is the only period-1 orbit bound to the saddle region with respect to the periodicity of the external driving in a time-periodically driven system.

Again, the same assumptions as discussed in Sec. 2.2 are valid, meaning that the system should be not too non-linear, and the velocities of reacting particles not too high. Such high non-linearities lead in general to a breakdown of our assumptions as the motion on the NHIM of systems with many degrees of freedom becomes highly chaotic. Examples for such chaotic regimes can be found in the high-energy limit of autonomous systems described in Refs. 70,71. In such highly non-linear systems, the NHIM and the TS can even bifurcate, as found in Refs. 40,72.

The distance  $\Delta\gamma$  shown in Fig. 7 has been computed on the NHIM at  $t_0 = 0$  for the potential (7). The color of any point on the NHIM indicates the distance  $\Delta\gamma$  of the phase space vectors of two trajectories, integrated for  $T/2 = 1$  in forward and backward time. A pronounced minimum, and thus by our proposed definition, a point on the TS trajectory, can clearly be dis-

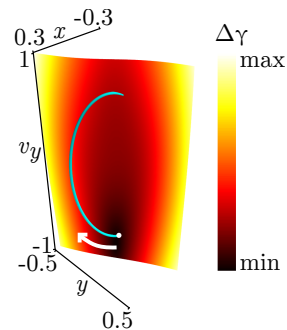


Figure 7: Surface: visualization of the NHIM for potential (7) at  $t = 0$ . Its color indicates the Euclidean distance of the phase space vectors  $\Delta\gamma = |\gamma(t_0 + T/2) - \gamma(t_0 - T/2)|$  for trajectories started at  $t_0 = 0$  at the respective point on the NHIM and propagated for half the period time of the external driving in potential (7) in forward and backward time. The point with  $\Delta\gamma = 0$  is marked by a white dot. It is the TS trajectory at time  $t = t_0$ . Propagating a trajectory starting at this specific point yields the TS trajectory displayed as a blue line (partially covered by the NHIM). The direction of movement on the TS trajectory is indicated by a white arrow. When calculated for different initial times  $t_0$  in a full period  $T$ , the union of all minima of  $\Delta\gamma$  on each NHIM at time  $t_0$  corresponds to the TS trajectory.

tinguished. Using this point as the initial condition for a trajectory propagation, the periodic TS trajectory can be obtained. Since this propagation will fail after a certain time because the NHIM is unstable, one cannot compute a fully periodic trajectory numerically. Nevertheless, with sufficiently accurate initial conditions on the NHIM, one can compute a periodic trajectory up to a certain error tolerance. Alternatively, various points on the TS trajectory can be computed from scratch searching for the minimum of  $\Delta\gamma$  on the NHIM calculated at different initial times  $t_0$ .

## 4 Machine-learning based description of the NHIM

Using the methods described in Sec. 3, a single point on the NHIM can be calculated for ar-

bitrary values of the bath modes and time, as is done here for the illustrative case of system (7). However, for a rate calculation, the position of the DS must be compared to the instantaneous position of each propagated particle at each time step. To compute the required reaction coordinate of the DS from scratch whenever it is needed would be prohibitively expensive because even the efficient binary contraction method described above requires the propagation of dozens of trajectories. To reduce the numerical effort, it is critical to obtain a continuous representation of the DS as a function of bath modes and time that is based on the knowledge of a comparatively small number of points on the NHIM.

Machine learning methods have already been applied within the field of theoretical chemistry for the interpolation of high-dimensional potential energy surfaces.<sup>82–92</sup> Following our recent work,<sup>45</sup> we use machine learning techniques—and specifically through the use of NNs or GPR—to characterize an entirely different set of surfaces: the NHIM and the DS attached to it. After training on an arbitrary set of points on the NHIM, a neural network yields a continuous representation that approximates the NHIM at the given points and elsewhere as shown in Sec. 4.1. In principle, the method can be applied for arbitrarily high dimension, though the complexity of the network may increase as the number of bath coordinates grows. Alternatively, the interpolation of a given set of points on the NHIM can be performed using GPR (see, e.g. Ref. 49 and references therein) as discussed in Sec. 4.2. This method requires *a priori* knowledge of the length and time scales along which the position of the NHIM is expected to vary. The comparison between these two methods and a discussion of their advantages and disadvantages in a given context will follow in Sec. 4.3.

## 4.1 Feed-forward neural networks

Neural networks (NNs) are a powerful tool for approximating functions of arbitrary complexity in any dimension.<sup>48</sup> Inspired by our current

understanding of NNs in the brain, an artificial NN consists of a set of model neurons with complex connections between them. It transforms a given input into some output. Typically, a NN is trained in the sense that a set of free parameters is adjusted on a set of input points for which the output is known. During *training*, the net *learns* to find patterns and dependencies in the given data without requiring any user input beyond the training data. When properly trained, the network will provide a good approximation to the correct output for input similar to the points it was trained on. In our case the training data consists of a set of points on the NHIM. By training on these points, we can create two separate NNs that represent the functions  $x^{\text{NHIM}}(\mathbf{y}, \mathbf{v}_y, t)$  and  $v_x^{\text{NHIM}}(\mathbf{y}, \mathbf{v}_y, t)$  separately, or use a more complex network to represent the combined function  $(x, v_x)^{\text{NHIM}}(\mathbf{y}, \mathbf{v}_y, t)$ . This representation will be reliable in the range of inputs that a suitably chosen set of training data covers, as long as the accuracy of the output was verified.

Here, we provide a brief overview on how we set up and use a NN to represent the DS.<sup>45</sup> For a more general discussion on NNs, we refer the interested reader to the literature, e.g. Refs. 93,94 and references therein. A feed-forward NN consists of several layers through which information is processed, see illustration in the schematic of Fig. 8. As we create a net to approximate  $(x, v_x)^{\text{NHIM}}(\mathbf{y}, \mathbf{v}_y, t)$ , the input layer is given by the bath coordinates  $\mathbf{y}$ , velocities  $\mathbf{v}_y$ , and time  $t$ . Based on its input, each neuron computes an output value that is passed to the next layer of the network, from that layer to the next, and so on until the final output is obtained.

In typical neural network applications, a user must decide on the number of hidden layers, the number of neurons in each layer, and the function by which a neuron converts its input into its output. Each neuron—such as the one shown in Fig. 8(b)—of a given layer receives inputs  $c_i$  from the  $N$  neurons of the previous layer. Its output,

$$c = a \left( \sum_{i=1}^N w_i c_i + b \right), \quad (8)$$

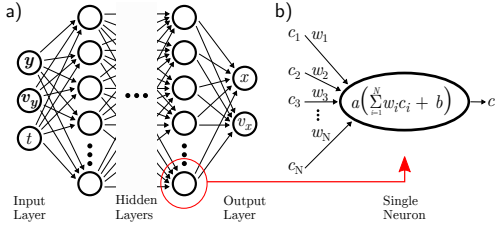


Figure 8: (a): A schematic for a NN approximating the function  $(x, v_x)^{\text{NHIM}}(\mathbf{y}, \mathbf{v}_y, t)$ . Using the  $(n - 1)$  bath coordinates  $\mathbf{y}$ , the respective velocities  $\mathbf{v}_y$  as well as a time  $t$  as input, the net processes this information layer by layer, passing the activation signals of the neurons from the previous layer to the next. The arbitrary number of layers between input layer and output layer are the hidden layers. Each layer has a number of neurons which interpret the incoming signals and pass them along to eventually calculate the reaction coordinates  $(x, v_x)^{\text{NHIM}}$  as results of the output layers. (b): A schematic of a neuron, indicating how its activation is computed. It sums up the input signals from the previous layer of neurons  $c_i$  weighted by  $w_i$  and adds an associated bias  $b$ , passing the results into the activation function  $a(x)$  to compute its output  $c$ .

depends on the input values and on the activation function  $a(x)$  whose nonlinearity ensures that the NN is not simply a linear transformation of the input variables and hence capable of capturing the necessary complexity. In this work, we choose the activation function  $a(x) = \tanh(x)$  for all neurons except those on the output layer as it is smooth and bound for  $x \rightarrow \pm\infty$  while providing the requisite nonlinearity. The different weights  $w_i$  that the neuron assigns to the output of each neuron of the previous layer and the bias  $b$  are optimized individually during the training process. We note, that although a single hidden layer NN is a universal approximator, it may not be the most efficient NN to a specific problem. In the following, we tried several different NN configurations, varying the number of hidden layers and neurons. The examples presented below provide the best results among the different approaches we sampled, but may not necessarily be fully optimized.

At the onset of training, a NN is initialized with the weights and biases of the neurons chosen randomly within a reasonable range of values. After successful training, a NN should return a good approximation of the outputs over the domain of inputs. A *cost function* is selected to measure the quality of the approximation. In the present application, we average the mean square difference between the output of the network and the known values over all training points. This mean square cost function has units of length squared whose values are denoted with respect to the dimensionless units introduced in Sec. 3.3. To minimize the cost function, there are several optimization routines that adjust the weights and biases of the neurons iteratively. This includes the *back propagation algorithm* which is a gradient-based method that successively adjusts all free parameters.<sup>45,95</sup> Usually, a NN is exposed to the entire set of training data many times before it is optimized, and each such exposure is called an *epoch*.

The training of a NN is usually not performed with all the available data. Instead, the data set is split into two disjoint sets: a *training set* and a *verification set*. The NN is trained only on the training set. The cost function on the verification set is used to assess the reliability of the NN for data that it was not trained for. If the cost function on the training set is significantly smaller than the cost function on the verification set, then the NN adapted very specifically to the given training set. This kind of overfitting is, in general, an undesirable effect as it indicates that the NN is no longer learning to make more accurate predictions.

To illustrate the ease in training a NN to approximate the position  $x^{\text{NHIM}}$  for the NHIM in the two-dimensional model system of Eq. (7), we wrote and tested a simple C++ code without using the more generally available libraries that we use in the production runs described below. The network was constructed with two hidden layers: the first and second with 40 and 10 neurons, respectively. It was trained on 2000 points randomly distributed over the NHIM and obtained using the binary contraction method. The remaining data points —

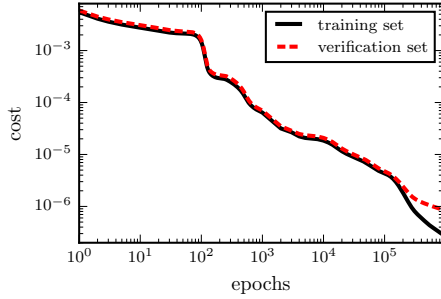


Figure 9: Training and verification costs of a NN for different epochs of training. The training is completed when the verification cost ceases to decrease, thereby preventing overfitting to the training set.

numbering approximately 8000— were used for verification. Fig. 9 shows the training and verification costs as a function of the epochs in the training. During training, both costs decrease monotonically at varying speeds and in concert with each other for most of the epochs. After more than about  $10^5$  epochs, the verification cost ceases to decrease while the training cost decays further. It is at this point (when additional training ceases to improve the verification) that the training of a NN should cease. Additional training simply provides better agreement with the training data while possibly distorting the general accuracy of the NN. In this example, we see that significant accuracy of the NHIM can be obtained (down to a cost of  $10^{-6}$ , measured in units of dimensionless length squared as described in Sec. 3.3), with a relatively small NN.

## 4.2 Gaussian process regression

Gaussian process regression (GPR) offers an alternative to the NN as a supervised machine learning approach for fitting a multidimensional hypersurface such as the NHIM to isolated training points. This approach interpolates given values of a function  $f(\mathbf{z})$  that depends on an arbitrary number of input variables  $\mathbf{z}$ . In our application, the input variables are the bath coordinates  $\mathbf{y}$ , velocities  $\mathbf{v}_y$  and time  $t$ . The unknown functions  $x^{\text{NHIM}}(\mathbf{y}, \mathbf{v}_y, t)$  and  $v_x^{\text{NHIM}}(\mathbf{y}, \mathbf{v}_y, t)$  will be interpolated independently.

GPR is based on the assumption that the values of a function  $f(\mathbf{z})$  are Gaussian random variables that are *a priori* unknown, apart from their statistical properties. After some values of the function have been *learned*, other values are inferred from the given information. For this inference to be possible, some *a priori* knowledge is required about the length scales along which the values of the function vary. Specifically, we assume that we know the *a priori* mean  $\mu(\mathbf{z})$  of the function  $f(\mathbf{z})$  and the covariance  $k(\mathbf{z}_1, \mathbf{z}_2)$  between the values  $f(\mathbf{z}_1)$  and  $f(\mathbf{z}_2)$ . In an unbiased *a priori* distribution these means would be set to zero, *i.e.*,  $\mu(\mathbf{z}) = 0$ . A typical choice for  $k(\mathbf{z}_1, \mathbf{z}_2)$  is the *squared exponential* kernel

$$k(\mathbf{z}_1, \mathbf{z}_2) = \exp\left(-\frac{|\mathbf{z}_1 - \mathbf{z}_2|^2}{2l^2}\right), \quad (9)$$

where the *hyper-parameter*  $l$  gives the typical length scale along which the values of  $f$  are expected to vary. It yields high covariance between closely located points  $\mathbf{z}_1$  and  $\mathbf{z}_2$  and a lower covariance between distant points. In an application to TST, the characteristic length  $l$  could differ for time and space coordinates, or even for different bath modes.

Assume now that the values  $\mathbf{f}_{\text{train}} = (f(\mathbf{z}_{\text{train}}^{(1)}), \dots, f(\mathbf{z}_{\text{train}}^{(n)}))^T$  of the function  $f$  are known in  $n$  points  $\mathbf{z}_{\text{train}}^{(i)}$  and the values  $\mathbf{f}_{\text{test}} = (f(\mathbf{z}_{\text{test}}^{(1)}), \dots, f(\mathbf{z}_{\text{test}}^{(m)}))^T$  in  $m$  other points  $\mathbf{z}_{\text{test}}^{(j)}$  are sought. Initially, the values  $\mathbf{f}_{\text{train}}$  and  $\mathbf{f}_{\text{test}}$  follow a multidimensional Gaussian distribution with zero mean and covariance matrix

$$\Sigma = \begin{pmatrix} K & N^T \\ N & M \end{pmatrix} \quad (10)$$

that consists of the blocks

$$K_{ij} = k(\mathbf{z}_{\text{train}}^{(i)}, \mathbf{z}_{\text{train}}^{(j)}) \quad (\text{of size } n \times n), \quad (11)$$

$$M_{ij} = k(\mathbf{z}_{\text{test}}^{(i)}, \mathbf{z}_{\text{test}}^{(j)}) \quad (\text{of size } m \times m), \quad (12)$$

$$N_{ij} = k(\mathbf{z}_{\text{test}}^{(i)}, \mathbf{z}_{\text{train}}^{(j)}) \quad (\text{of size } m \times n). \quad (13)$$

The *a posteriori* conditional distribution of the unknown values  $\mathbf{f}_{\text{test}}$ , given the known values

$\mathbf{f}_{\text{train}}$ , is a Gaussian function with mean<sup>49,50</sup>

$$\bar{\boldsymbol{\mu}} = NK^{-1}\mathbf{f}_{\text{train}} \quad (14)$$

and covariance matrix

$$\bar{\boldsymbol{\Sigma}} = M - NK^{-1}N^T. \quad (15)$$

The components of the conditional mean  $\bar{\boldsymbol{\mu}}$  are taken to be the interpolated values of the function. If desired, the diagonal elements of the conditional covariance matrix  $\bar{\boldsymbol{\Sigma}}$  can be used to provide an error estimate for the interpolation. If the standard deviation  $\sqrt{\bar{\Sigma}_{ii}}$  is sufficiently small, the interpolated value  $f(z_{\text{test}}^{(i)})$  is known essentially with certainty.

In practice, we compute a Cholesky decomposition  $K = LL^T$  of the matrix  $K$ , where  $L$  is a lower triangular matrix. It is then easy to pre-compute the vector  $K^{-1}\mathbf{f}_{\text{train}} = L^{-T}L^{-1}\mathbf{f}_{\text{train}}$  that depends only on the training points, but not on the points  $z_{\text{test}}^{(i)}$ . After that, the computation of one interpolated function value according to Eq. (14) requires the computation of the corresponding row of the matrix  $N$  and the scalar product with the precomputed vector. The effort required is proportional to the number  $n$  of training points. The conditional covariance matrix (15) can be written as

$$\Sigma = M - \tilde{N}\tilde{N}^T, \quad (16)$$

where the matrix  $\tilde{N}$  is the solution of the equation  $N = \tilde{N}L^T$ . Because  $L$  is triangular, the computation of one row of  $\tilde{N}$ , and therefore of an error estimate for one data point, requires an effort of order  $n^2$ .

The training and use of a GPR machine can be illustrated relative to a sine function shown in Fig. 10 (b). We use the squared exponential kernel (9) with a characteristic length scale  $l = 1$ . The conditional mean that serves as the interpolated function is indicated by the dashed red line, a 95 % confidence interval based on the conditional standard deviation by the gray area.

The difference between the GPR machine’s approximation and the original sine-function is small near the training points and increases elsewhere. Whereas for the prior, the mean  $\mu$  and the 95 % confidence region are independent

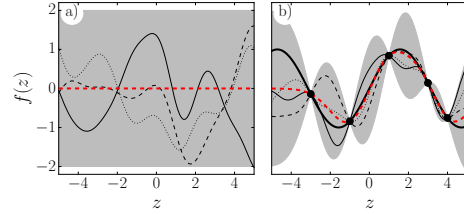


Figure 10: (a) A Gaussian prior with a mean of zero (thick red dashed line). The 95 % confidence interval is given by a gray area which covers two standard deviations from the mean. The lines show three random functions drawn from this prior. (b) Five points (●) located on a sine curve (thick black line) are used as training data. Learning these points turns the prior into the posterior distribution, changing its mean (thick red line) and its 95 % confidence interval. Thin curves show three random functions sampled from the posterior.

of  $z$ , the *learning* of several training points alters both. At the training points, this 95 % region is zero, as these points are assumed to be correct. The further away from training points, the more the confidence region grows. This increase happens on a length-scale compared to the characteristic length  $l = 1$  of the assumed squared exponential kernel (9). At points far from where no training points are given, e. g. for  $z < -3$ , the confidence region of the posterior realigns with the Gaussian prior. As a result, the accuracy of the approximation increases with the number of training points.

To illustrate that the posterior of the GPR machine gives a distribution over admissible functions, we plot three functions randomly sampled from the posterior distribution in Fig. 10 (b) and compare to three samples of the prior distribution in Fig. 10 (a). The samples of the posterior distribution match the training points and are mostly located within the altered 95 % confidence region around the approximation  $\mu(z)$  to the original function  $f(z)$ . Therefore, the posterior ‘chooses’ which functions of the prior are suitable to interpolate between the given training points.

### 4.3 NNs vs. GPR

We now compare the advantages and disadvantages of the two machine learning methods introduced in Secs. 4.1 and 4.2 for the interpolation of the NHIM and construction of the DS in the calculation of rate constants. The aim is to provide a guideline on when to use which method.

For the training of a NN on a given set of pre-calculated points of the NHIM, little previous knowledge on the interpolated object is needed. For example, in a typical NN that approximates  $(x, v_x)^{\text{NHIM}}(\mathbf{y}, \mathbf{v}_y, t)$ , the number of neurons in each of the input and output layers are fixed as illustrated in Fig. 8. The only information that is prescribed is the structure of the hidden layers, and it merely needs to contain as much complexity as the system that it is regenerating. Its efficacy is confirmed if the optimization of the neurons leads to weights and biases that are sufficiently small compared to the target accuracy with respect to the cost function.

When using GPR, the success and the speed of training highly depends on the chosen hyper-parameters of the kernel function. In choosing these parameters, one can include previous knowledge of the object to be interpolated, e. g. the expectation that the NHIM of a given system may be a smooth surface with small curvature. Consequently, the characteristic length in the direction of the bath coordinates can be higher than the characteristic length of the variation in time, if the external driving leads to a fast moving NHIM whose curvature does not vary heavily. If such previous knowledge is available, training can be accelerated significantly by choosing an appropriate kernel function or by a corresponding restriction of the parameter ranges in the optimization.

Another difference between both methods lies in the procedure of *training* itself. For NNs, training with the back propagation algorithm is an iterative process. It iterates a possibly high number of times—that is, epochs—over the given training set gradually adjusting weights and biases to achieve a certain level of convergence in the training and verification cost. This

procedure typically takes a lot of computation time. Using GPR, training can be effectively broken down to basic routines in linear algebra without need for iteration. Therefore, training to a certain accuracy in the context of GPR is usually much faster compared to the use of NNs.

We compare the training of the neural network already discussed in Sec. 4.1 to the training done by GPR on the NHIM of the same two-dimensional model potential according to Eq. (7). The NN is set up with two hidden layers with 40 and 10 neurons as before. For GPR, we use the *squared exponential* kernel (Eq. (9)) with  $l = 2.0$  as hyper-parameter. In the left panel of Fig. 11, the drop of the cost function in relation to the needed computational time is shown for the training for each of the NN and GPR machines. Whereas training a NN converges after several thousand seconds, the GPR approach reaches the same level of accuracy within seconds.

Another difference in the implementation of the two methods lies in the relative time each machine learning approach requires to evaluate the function  $(x, v_x)^{\text{NHIM}}(\mathbf{y}, \mathbf{v}_y, t)$  for some input of the bath coordinates. For simplicity, we report the internal run time reported by the same computer and call it the *function call time*. Precision for GPR is achieved by adding more training points and therefore decreasing the 95% region of the posterior. But adding more training points leads to an increasing dimension of the matrices discussed in Sec. 4.2. Consequently, the function call time required to evaluate Eq. (14) increases linearly with the number of training points as shown by the red curve in the right panel of Fig. 11. In contrast, increasing the number of training points does not increase the evaluation time for the NN, assuming its layer geometry does not change. Thus evaluating a small NN can be significantly faster than GPR, although it might have taken a long time to train the net.

Based on these three observations, we now have guidelines for choosing between the NN and GPR machines for continuously interpolating the time-dependent NHIM or the related DS: (i) In cases when the function call time ex-



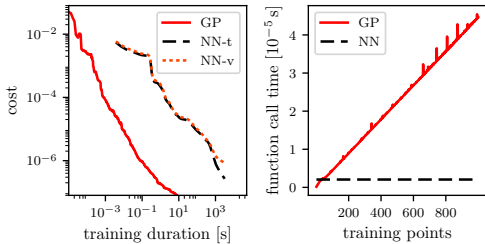


Figure 11: Comparison between NN and GPR: In the left panel, the cost function over the needed training time is shown for each implementation as a function of computing time. For GPR, successfully adding training points leads to a fast decrease of the training cost, whereas the same decrease using NNs is considerably slower while training with the back propagation algorithm. Here, both the training cost (NN-t), as well as the verification cost (NN-v) are shown. In the right panel, the function call time required for each method is shown relative to the number of training points. For GPR, function call time increases linearly with the number training points provided to it, and which are needed to decrease the cost function. Thus, high accuracy of GPR comes with the disadvantage of an increasing function call time. The black dashed line shows the function call time of a NN and it is independent of the number of training points because the cost of sampling across an epoch is dominated by the inversion of the NN parameters.

ceeds the additional training time of the neural network compared to GPR, it is computationally more efficient to use the neural network approach. A simple example would be a system with time-periodic external driving (and without friction). If the focus is on calculating rates for different initial ensembles of particles under the influence of the same external force, the NHIM and therefore the DS is always the same. Consequently, it is more efficient to use NNs so as to benefit from the short function call time. (ii) On the other hand, in cases when the same ensemble is used to calculate reaction rates for a system under influence of a different external driving or in cases with non-periodic driving, the training time for each NN can easily be higher than the over-

all function call time to propagate the ensemble and calculating reaction rates. It is then more efficient to use GPR. Another example is the future application of the methods presented here on thermal systems which experience non-trivial friction and fluctuations. In these systems, there will be a different NHIM (and DS) for each sequence of random numbers modeling the fluctuations in a Langevin-type approach, and each sequence will require separate training. The training time can then easily exceed the function call time. (iii) A third guideline can be formulated in terms of how many training points are needed for a given method. Using NNs, millions of training points can be fit easily, whereas with GPR, the function call time is rather high even for just thousands of training points, and its use to fit millions of training points is hopeless. This comparison can also be seen exactly in the opposite way, as prior knowledge on the object to be interpolated—here the NHIM—can be included in the GPR method by choosing an appropriate kernel function. This usually reduces the amount of training data needed significantly and provides an advantage of GPR over the NN approach. If only a few training points are available (for example if training data is hard to obtain), GPR is still capable of interpolating these few points when provided a suitable kernel function, at least one with respect to optimization of the cost function.

## 5 Application to a two-dimensional system

In this section, we assess the performance of the methods summarized above for the central application of this work. That is, the determination of the time-dependent NHIM and the associated rate for the driven model potential (7) for the same parameters as used in Sec. 3.3. In the previous section, we focused on the accuracy as determined by the cost function as one would typically use in training a NN or GPR machine. However, in our specific application, we also have an external criterion at our disposal that allows us to assess the accuracy of the

NHIM. Namely, the DS attached to the NHIM should by construction be recrossing-free. For a numerical test of this criterion, a large ensemble of trajectories can be launched in the neighborhood of the barrier and propagated forward in time until it leaves this neighborhood. In the process, the number of crossings and recrossings of the DS can be monitored. Beyond the decrease in the cost function, the demand of the DS to be as recrossing free as possible is another—and in our case more relevant—indicator of a successful training process.

To obtain a stringent test of the DS, it is advantageous to launch trajectories close to the stable manifolds. These trajectories will remain in the neighborhood of the barrier for a long time and therefore have plenty of opportunity to recross a DS of low quality. A detailed description of such tests for the NHIM constructed by other approaches, together with numerical results, can be found in Refs. 44,45 for various test systems. In what follows, we test the NN and GPR machines relative to this criteria.

## 5.1 Representation of the NHIM by NNs or GPR

For both NNs and GPR, training points are computed in the range  $y \in [-4, 4]$  and  $v_y \in [-8, 8]$ . The NN is trained on an equidistant grid of 25 points in  $y$ , 40 points in  $v_y$  range and 40 points in  $t$ , sampled over one period, i.e., in the interval  $[0, 2]$ . The structure of the net uses three input neurons for the variables  $y$ ,  $v_y$ , and  $t$ , four hidden layers with 40, 40, 40, and 10 neurons, respectively, and one output neuron representing  $x^{\text{NHIM}}$ . While the NN discussed in Sec. 4.3 was written from scratch in our group, here we resort to an efficient implementation using TensorFlow.<sup>96</sup> To improve the periodicity of the resulting function  $x^{\text{NHIM}}$ , the grid of training points is tripled to cover a full time span from  $t \in [0, 6]$ . All in all, we use 120 000 training points taking advantage of the periodicity of the driven system, so just 40 000 points have to be obtained using the binary contraction method explained in Sec. 3.3. After training for 50 000 epochs, the training cost has de-

creased to  $2.06 \times 10^{-7}$  and the verification cost to  $2.15 \times 10^{-7}$ . Although using so many training points might seem to be a bit conservative here, we did so as to ensure, that the likelihood of single, localized artifacts in the representation of the DS is reduced. Such artifacts, which are hidden by averaging over a large amount of accurately inferred data points in the loss function, will cause spurious recrossings to the trajectories. This would lead to errors in the corresponding rates which goes against our primary reason for calculating the non-recrossing DS. Thus the large number of points—though not fully optimized—was chosen in order to ensure not only the proper construction of the manifold but also to satisfy our more stringent criterion for a reduced number of recrossings.

For the GPR representation of the NHIM, we used 6000 training points in total on equidistant grids chosen at 30 time steps in the range  $t \in [-0.5, 2.4]$  and in the same  $y$  and  $v_y$  ranges as for the NN. The time range is more than the full period of  $T = 2$  of the time-dependent potential (7) in order to improve periodicity. Training was done using a squared exponential kernel like in Eq. (9), but with different characteristic length scales  $l$  in any direction  $(y, v_y, t)$ . Training was carried out for various values of the hyperparameters on a logarithmic grid, and the set that led to the smallest verification cost was retained. The final values of the hyperparameters that best reflect the length and time scales on which the NHIM moves, were  $\sigma_y \approx 0.705$ ,  $\sigma_{v_y} \approx 0.790$  and  $\sigma_t \approx 0.505$ . Note, that the equidistant grids are chosen in order to provide no initial bias, and thus focus on whether the different approaches are feasible. Choosing non-equidistant grids may be a subject for future optimization of the training process.

To assess the accuracy of the representation obtained from either the NN or the GPR, we computed the position of the NHIM at a fixed time for a grid of  $50 \times 50$  additional points in the ranges  $y \in [-4, 4]$  and  $v_y \in [-8, 8]$  with an accuracy of  $10^{-12}$  using the binary contraction method. Afterwards, the values  $x_{\text{NN}}(y, v_y, 0)$  and  $x_{\text{GPR}}(y, v_y, 0)$  were obtained for both trained machines of the NHIM at all

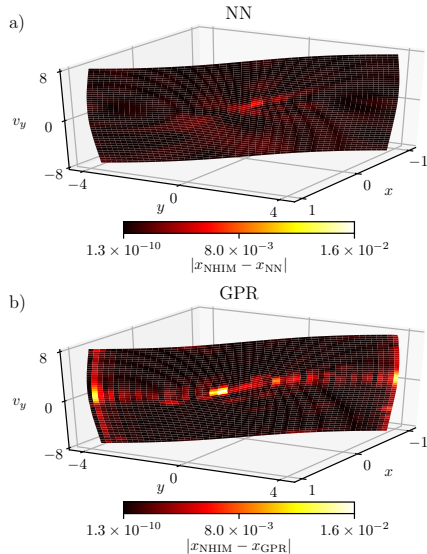


Figure 12: Estimation of the error of the NN (a) and GPR (b) machines in representing the NHIM. Here, the values  $x_{\text{NHIM}}(y, v_y, 0)$  for a grid of  $50 \times 50$  points in the ranges  $y \in [-4, 4]$  and  $v_y \in [-8, 8]$  are obtained with an accuracy of  $10^{-12}$  using the binary contraction method explained in Sec. 3.3. The color of the curved surface connecting these points represents the pointwise difference to the representation of the NHIM by one of the two interpolation methods discussed here.

of these  $50 \times 50$  verification points. Fig. 12 shows the location of the NHIM at time  $t = 0$  (in a three-dimensional projection of the four-dimensional phase space). Very similar figures have been obtained for different times. The color of the surfaces represents the error of the machine learning representation of the NHIM compared to the numerically exact result.

The maximum error of the NN representation is  $6.1 \times 10^{-3}$  as seen in Fig. 12(a). It is nearly a factor of three smaller than the maximum error for GPR in Fig. 12(b). Large errors of the NN occur just in a limited region near  $y = v_y = 0$ , whereas the GPR has large errors in a band-like structure near  $v_y = 0$ . The maximum difference could be lowered by e. g. using more training points. But this will come with the burden of longer training times for neural networks or longer function call times for GPR. A different approach, to be pursued in the future, might be the use of a non-equidistant grid of train-

**Table 2: Training and function call times for the use of NN and GPR to learn and access the same NHIM for the model problem discussed in the text.**

	NN	GPR
obtaining training data	3636 s	545 s
training	4220 s	9 s
optimize hyper-parameters	–	2238 s
total training time	7856 s	2792 s
function call time	14 $\mu$ s	253 $\mu$ s

ing points. By increasing the density of points in the regions where the quality of the representation is poor and decreasing it elsewhere, it should be possible to lower the maximum error while keeping the total number of training points constant.

As noted above, the GPR interpolates the given training points exactly, whereas the NN attempts to minimize the training cost, i.e., the mean square error, averaged over all training points, without bringing it to zero. It now appears that the NN produces a smoother representation of the NHIM, with moderate errors everywhere, whereas the GPR yields very small errors near the training points at the expense of larger errors elsewhere.

Note that a NN typically has intrinsic hyper-parameters like the activation function or the learning rate. Although adjusting these parameters is not free, we neglect this adjustment time in our discussion, since in our experience, these parameters have to be adjusted only once for the general problem, but can be kept constant while just varying potential parameters as, e. g., the amplitude of the moving saddle.

Finally, we comment on the numerical effort required by both methods, as represented by the computing times listed in Table 2. These times depend critically on, e. g., the hardware used or the desired accuracy, and should be understood as a rough guidance only. Computing all 40 000 training points for the neural network took 3636 s, whereas obtaining 6 000 points for the GPR took only 545 s. (This corresponds to 0.091 s on average per training point.) The pro-

cess of training the neural net for 50 000 epochs took 4220 s. GPR, by contrast, could be trained in only 9 s. However, this training had to be repeated many times in order to optimize the hyperparameters. In total, the process took 2238 s. In summary, the neural network was ready to use after 7856 s (about 131 min) the GPR about 2792 s (about 47 min). Afterwards, the evaluation of a single value of  $x^{\text{NHIM}}$  took  $14 \mu\text{s}$  for the neural network and  $253 \mu\text{s}$  for GPR. Thus, although training in GPR is much faster, its function call time is by a factor of 18 higher. As we concluded at the end of Sec. 4, depending on how many function calls are required after training, either method can be more computationally efficient: The NN becomes advantageous if more than about 21 million evaluations are made.

## 5.2 Relative accuracy of the rate constants: NNs vs. GPR

To obtain a rate constant for a reaction over the time-dependent barrier of potential (7), we propagate an ensemble of particles initialized on the reactant side and monitor the reaction of particles to the product side. The ensemble is set up with  $N_{\text{react}}(t=0) = 10^6$  particles at  $x_0 = -0.2$  with  $y$  on the minimum energy path of potential (7) at time  $t_0 = 0$ . The velocities  $(v_x, v_y)$  of these particles are sampled from a thermal distribution with  $k_B T = 0.5$ .

Propagating the full ensemble using a standard velocity-Verlet integrator, we monitor possible crossings of the DS and the related crossing times of each particle. Ideally, each trajectory should either descend on the reaction side of the barrier without crossing the DS or cross it at exactly once and then descend on the product side. In practice, because of inaccuracies in how the DS is represented, we expect a small number of trajectories to recross the DS.

A histogram with the number of crossings is shown in Fig. 13 (a) for the NN and in Fig. 13 (b) for the GPR approach. In both cases approx. 16% of all particles are reactive and cross the DS once from the reactant to the product side. Nearly all other particles are non-reactive. The number of trajectories that cross the DS

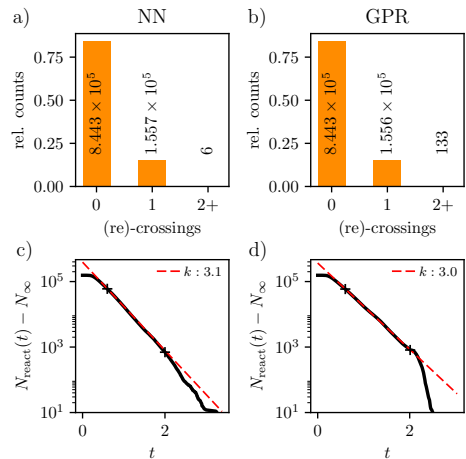


Figure 13: Number of trajectories in the reactive ensemble crossing the DS 0, 1 or more times, as determined using a NN and GPR in panels (a) and (b), respectively. As the associated DS should be recrossing free, the number of trajectories with more than one crossing (2+) gives an estimate of the error in the computation. Whereas the  $y$ -axis is in relative counts, the numbers on the bars are absolute numbers. The corresponding time-dependent traces of the reactant population  $N_{\text{react}}(t)$  shifted by its infinite time limit are shown in panels (c) and (d). The rate constant  $k$  given in units of a dimensionless inverse time can be found from the linear fit (shown with a dashed red line) in the steady-state regime according to Eq. (2).

more than once is very small in both cases, though the GPR still produces a factor 22 more recrossings than the NN. This difference arises because the NN represents the NHIM more accurately than GPR as shown in Fig. 12.

Rates are obtained in dimensionless units using the procedure described in Sec. 2.3. For the NN approach in Fig. 13 (c), we obtain a rate constant of  $k_{\text{NN}} = 3.1$  compared to the rate constant of  $k_{\text{GPR}} = 3.0$  for the GPR approach in Fig. 13 (d). As discussed in Secs. 2.3 and 3.3, these rate constants are given in units of a dimensionless inverse time. The time-scale is given by a period with  $T = 2$  of the oscillation with frequency  $\omega_x = \pi$  in Eq. (7). Although these constants are very similar,  $k_{\text{NN}}$  is presumed to be more precise because the DS of the NN approach exhibits fewer recrossings. Note

that differences between the reactant populations computed by the two methods arise once the populations have decayed to approximately the number of recrossing trajectories in GPR.

## 6 Conclusion and outlook

In this paper, we have presented several techniques to obtain the underlying geometric structures needed to obtain rate constants for reactions over time-dependent driven barriers in the context of TST. Specifically, we focused on the determination of the NHIM which is generally a multidimensional, time-dependent object located in phase space, to which a recrossing-free DS can be attached, and with respect to which the TST rate is obtained. As we restrict ourselves to rank-1 saddles, the procedures presented here can in principle be carried out for any multidimensional system with one unstable degree of freedom and an arbitrary number of stable bath degrees of freedom. Here, one needs to keep in mind that in higher-dimensional systems not only does the propagation of trajectories become more complex and computationally more challenging, but also the NHIM itself scales modestly with the dimensionality of the equations of motion. Depending on how one chooses to construct the DS, the number of points needed to characterize the underlying NHIM may scale exponentially with increasing dimensionality.

A central result of this work is the extension of recent approaches to address reactions of increasing dimensionality. We elaborate on the use of our recently developed *binary contraction method*<sup>46</sup> for the accurate non-perturbative determination of points on the NHIM. In the case of time-periodically driven Hamiltonian systems, this allowed us to extend the one-dimensional TS trajectory to multi dimensions by defining it to be the unique periodic orbit that remains in the vicinity of the saddle.

A second central result of this work is the elaboration and extension of our recent work on NNs.<sup>45</sup> Here we show the applicability of the NN to construct the NHIM and rates for a two-dimensional system. We also demonstrated

that a different machine learning approach, the GPR, is also effective. These two approaches have competing efficiencies in training and use. Together, they offer a powerful combination for obtaining high-accuracy smooth NHIMs for barriers with rank-1 saddles in higher dimensions. Indeed, Kamath et al<sup>97</sup> recently compared both of these methods for the first time in the context of constructing potential energy surfaces.<sup>92</sup> Although their work has a different objective than the NHIMs obtained here, we likewise obtained similar conclusions regarding the advantages and disadvantages of the two methods with respect to the determination of the target surface. Carrington and coworkers<sup>92</sup> employed a secondary criterion for evaluating the quality of their computed surface. Namely, the GPR outperforms NN when fitting potential energy surfaces to obtain vibrational spectra. In our case, we see in Fig. 13 that the total number of recrossings for the DS obtained by GPR is significantly higher than that found by the NN. Following this additional—and in our case more relevant—requirement to the DS that it be recrossing-free, we conclude that for our applications the NNs are more useful in the sense that they provide higher accuracy for an indirect metric not used as part of the cost function while training. We emphasize that we do not claim our setup to be fully optimized in terms of net structure or the training process, but it is the best we found using comparable approaches.

For the determination of the rate, the computation of the required reaction coordinates of the DS from scratch (using e. g. the binary contraction method) whenever it is needed would be prohibitively expensive, as the position of the DS must be compared to the instantaneous position of each propagated particle. Having a continuous representation of the DS as a function of bath modes and time reduces this numerical effort drastically. This is where the use of NN and GPR machine-learning techniques represent a key advance by way of providing a suitable *smooth* surfaces.

It remains to show that these approaches are effective for the determination of rate constants in systems with more complex energy land-

scapes or external potentials. This includes aperiodic external driving, potentials with multiple saddles, dissipative terms and thermal noise. In addition, although these methods are developed for in principle arbitrary dimensional problems, everything done so far was an application to two- and three-dimensional model systems.<sup>44,45</sup> So one aspect for a future work will be to test these methods on really high dimensional systems with tens of degrees of freedom to answer the question if for increasing dimensionality really an exponential increase in the obtained number of points on the NHIM is necessary or if these points can be chosen more efficiently by improving the machine learning procedures.

An application for future work will lead away from model systems towards real systems like the  $\text{LiCN} \leftrightarrow \text{LiNC}$  isomerization reaction, to which an analytical approximation to the molecular potential<sup>98</sup> as well as to the molecular dipole moment<sup>99</sup> is known. Another possible application includes the ketene isomerization<sup>67,100,101</sup> which has received increased attention since the initial work by some of us<sup>35</sup> using the Lagrangian descriptor to obtain the NHIM and the associated rate constants. The determination of the full-dimensional NHIM in both of these cases remains a challenge that may be addressable using the emerging approaches presented here.

**Acknowledgement** The German portion of this collaborative work was partially supported by Deutsche Forschungsgemeinschaft (DFG) through Grant No. MA1639/14-1. The US portion was partially supported by the National Science Foundation (NSF) through Grant No. CHE 1700749. 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 European Union’s Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie Grant Agreement No. 734557. Surface plots in Figs. 6 and 7 have been made with the *Mayavi*

software package.<sup>102</sup>

## References

- (1) Eyring, H. The Activated Complex in Chemical Reactions. *J. Chem. Phys.* **1935**, *3*, 107–115.
- (2) Wigner, E. P. Calculation of the Rate of Elementary Association Reactions. *J. Chem. Phys.* **1937**, *5*, 720–725.
- (3) Pechukas, P. Transition State Theory. *Annu. Rev. Phys. Chem.* **1981**, *32*, 159.
- (4) Truhlar, D. G.; Garrett, B. C.; Klippenstein, S. J. Current Status of Transition-State Theory. *J. Phys. Chem.* **1996**, *100*, 12771–12800.
- (5) Mullen, R. G.; Shea, J.-E.; Peters, B. Communication: An existence test for dividing surfaces without recrossing. *J. Chem. Phys.* **2014**, *140*, 041104.
- (6) Jaffé, C.; Farrelly, D.; Uzer, T. Transition State Theory without Time-Reversal Symmetry: Chaotic Ionization of the Hydrogen Atom. *Phys. Rev. Lett.* **2000**, *84*, 610–613.
- (7) Jacucci, G.; Toller, M.; DeLorenzi, G.; Flynn, C. P. Rate Theory, Return Jump Catastrophes, and the Center Manifold. *Phys. Rev. Lett.* **1984**, *52*, 295.
- (8) Komatsuzaki, T.; Berry, R. S. Regularity in chaotic reaction paths. I.  $\text{Ar}_6$ . *J. Chem. Phys.* **1999**, *110*, 9160–9173.
- (9) Komatsuzaki, T.; Berry, R. S. Chemical Reaction Dynamics: Many-Body Chaos and Regularity. *Adv. Chem. Phys.* **2002**, *123*, 79–152.
- (10) Toller, M.; Jacucci, G.; DeLorenzi, G.; Flynn, C. P. Theory of classical diffusion jumps in solids. *Phys. Rev. B* **1985**, *32*, 2082.

- (11) Voter, A. F.; Montalenti, F.; Germann, T. C. Extending the time scale in atomistic simulations of materials. *Annu. Rev. Mater. Res.* **2002**, *32*, 321–346.
- (12) de Oliveira, H. P.; Ozorio de Almeida, A. M.; Damião Soares, I.; Tonini, E. V. Homoclinic chaos in the dynamics of a general Bianchi type-IX model. *Phys. Rev. D* **2002**, *65*, 083511/1–9.
- (13) Jaffé, C.; Ross, S. D.; Lo, M. W.; Marsden, J.; Farrelly, D.; Uzer, T. Statistical Theory of Asteroid Escape Rates. *Phys. Rev. Lett.* **2002**, *89*, 011101.
- (14) H. Waalkens,; A. Burbanks,; S. Wiggins, Escape from planetary neighborhoods. *Mon. Not. R. Astron. Soc.* **2005**, *361*, 763.
- (15) Huepe, C.; Métens, S.; Dewel, G.; Borckmans, P.; Brachet, M. E. Decay rates in attractive Bose-Einstein condensates. *Phys. Rev. Lett.* **1999**, *82*, 1616.
- (16) Huepe, C.; Tuckerman, L. S.; Métens, S.; Brachet, M. E. Stability and decay rates of nonisotropic attractive Bose-Einstein condensates. *Phys. Rev. A* **2003**, *68*, 023609.
- (17) A. Junginger,; J. Main,; G. Wunner,; M. Dorwarth, Transition state theory for wave packet dynamics. I. Thermal decay in metastable Schrödinger systems. *J. Phys. A: Math. Theor.* **2012**, *45*, 155201.
- (18) A. Junginger,; M. Dorwarth,; J. Main,; G. Wunner, Transition state theory for wave packet dynamics. II. Thermal decay of Bose-Einstein condensates with long-range interaction. *J. Phys. A: Math. Theor.* **2012**, *45*, 155202.
- (19) A. Junginger,; M. Kreibich,; J. Main,; G. Wunner, Transition states and thermal collapse of dipolar Bose-Einstein condensates. *Phys. Rev. A* **2013**, *88*, 043617.
- (20) Truhlar, D. G.; Garrett, B. C. Variational Transition State Theory. *Annu. Rev. Phys. Chem.* **1984**, *35*, 159–189.
- (21) Vanden-Eijnden, E.; Tal, F. A. Transition state theory: Variational formulation, dynamical corrections, and error estimates. *J. Chem. Phys.* **2005**, *123*, 184103.
- (22) Miller, W. H. Semi-classical theory for non-separable systems: Construction of “good” action-angle variables for reaction rate constants. *Faraday Discuss. Chem. Soc.* **1977**, *62*, 40.
- (23) Miller, W. H.; Hernandez, R.; Handy, N. C.; Jayatilaka, D.; Willetts, A. Ab initio calculation of anharmonic constants for a transition state, with application to semiclassical transition state tunneling probabilities. *Chem. Phys. Lett.* **1990**, *172*, 62–68.
- (24) Hernandez, R.; Miller, W. H. Semiclassical Transition State Theory. A New Perspective. *Chem. Phys. Lett.* **1993**, *214*, 129–136.
- (25) Miller, W. H. Spiers Memorial Lecture Quantum and semiclassical theory of chemical reaction rates. *Faraday Discuss. Chem. Soc.* **1998**, *110*, 1.
- (26) Nguyen, T. L.; Stanton, J. F.; Barker, J. R. Ab Initio Reaction Rate Constants Computed Using Semiclassical Transition-State Theory:  $\text{HO} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$  and Isotopologues. *J. Phys. Chem. A* **2011**, *115*, 5118–5126.
- (27) Keck, J. C. Variational Theory of Reaction Rates. *Adv. Chem. Phys.* **1967**, *13*, 85–121.
- (28) Wiggins, S.; Wiesenfeld, L.; Jaffé, C.; Uzer, T. Impenetrable Barriers in Phase-Space. *Phys. Rev. Lett.* **2001**, *86*, 5478.
- (29) Uzer, T.; Jaffé, C.; Palacián, J.; Yanguas, P.; Wiggins, S. The geometry of reaction dynamics. *Nonlinearity* **2002**, *15*, 957.

- (30) Bartsch, T.; Hernandez, R.; Uzer, T. Transition state in a noisy environment. *Phys. Rev. Lett.* **2005**, *95*, 058301.
- (31) Bartsch, T.; Moix, J. M.; Hernandez, R.; Kawai, S.; Uzer, T. Time-dependent transition state theory. *Adv. Chem. Phys.* **2008**, *140*, 191–238.
- (32) Hernandez, R. A Combined Use of Perturbation Theory and Diagonalization: Application to Bound Energy Levels and Semiclassical Rate Theory. *J. Chem. Phys.* **1994**, *101*, 9534–9547.
- (33) Allahem, A.; Bartsch, T. Chaotic dynamics in multidimensional transition states. *J. Chem. Phys.* **2012**, *137*, 214310.
- (34) Junginger, A.; Garcia-Muller, P. L.; Borondo, F.; Benito, R. M.; Hernandez, R. Solvated molecular dynamics of LiCN isomerization: All-atom argon solvent versus a generalized Langevin bath. *J. Chem. Phys.* **2016**, *144*, 024104.
- (35) Craven, G. T.; Hernandez, R. Deconstructing field-induced ketene isomerization through Lagrangian descriptors. *Phys. Chem. Chem. Phys.* **2016**, *18*, 4008–4018.
- (36) Pollak, E.; Pechukas, P. Transition States, Trapped Trajectories, and Classical Bound States Embedded in the Continuum. *J. Chem. Phys.* **1978**, *69*, 1218–1226.
- (37) Pechukas, P.; Pollak, E. Classical Transition State Theory is Exact if the Transition State is Unique. *J. Chem. Phys.* **1979**, *71*, 2062–2068.
- (38) Uzer, T.; Jaffé, C.; Palacián, J.; Yanguas, P.; Wiggins, S. The Geometry of Reaction Dynamics. *Nonlinearity* **2002**, *15*, 957–992.
- (39) Teramoto, H.; Toda, M.; Komatsuzaki, T. Dynamical Switching of a Reaction Coordinate to Carry the System Through to a Different Product State at High Energies. *Phys. Rev. Lett.* **2011**, *106*, 054101.
- (40) Li, C.-B.; Shoujiguchi, A.; Toda, M.; Komatsuzaki, T. Definability of No-Return Transition States in the High-Energy Regime above the Reaction Threshold. *Phys. Rev. Lett.* **2006**, *97*, 028302.
- (41) Waalkens, H.; Wiggins, S. Direct construction of a dividing surface of minimal flux for multi-degree-of-freedom systems that cannot be recrossed. *J. Phys. A* **2004**, *37*, L435–L445.
- (42) Çiftçi, U.; Waalkens, H. Reaction Dynamics Through Kinetic Transition States. *Phys. Rev. Lett.* **2013**, *110*, 233201.
- (43) Craven, G. T.; Hernandez, R. Lagrangian descriptors of thermalized transition states on time-varying energy surfaces. *Phys. Rev. Lett.* **2015**, *115*, 148301.
- (44) Feldmaier, M.; Junginger, A.; Main, J.; Wunner, G.; Hernandez, R. Obtaining time-dependent multi-dimensional dividing surfaces using Lagrangian descriptors. *Chem. Phys. Lett.* **2017**, *687*, 194.
- (45) Schraft, P.; Junginger, A.; Feldmaier, M.; Bardakcioglu, R.; Main, J.; Wunner, G.; Hernandez, R. Neural network approach to time-dependent dividing surfaces in classical reaction dynamics. *Phys. Rev. E* **2018**, *97*, 042309.
- (46) Bardakcioglu, R.; Junginger, A.; Feldmaier, M.; Main, J.; Hernandez, R. Binary contraction method for the construction of time-dependent dividing surfaces in driven chemical reactions. *Phys. Rev. E* **2018**, *98*, 032204.
- (47) Pozun, Z. D.; Hansen, K.; Sheppard, D.; Rupp, M.; Muller, K.-R.; Henkelman, G. Optimizing transition states via kernel-based machine learning. *J. Chem. Phys.* **2012**, *136*, 174101.



- (48) Funahashi, K.-I. On the approximate realization of continuous mappings by neural networks. *Neural Networks* **1989**, *2*, 183 – 192.
- (49) Murphy, K. P. *Machine learning: a probabilistic perspective*; MIT press, 2012.
- (50) Rasmussen, C. E.; Williams, C. K. *Gaussian process for machine learning*; MIT press, 2006.
- (51) Bartsch, T.; Uzer, T.; Moix, J. M.; Hernandez, R. Identifying Reactive Trajectories Using a Moving Transition State. *J. Chem. Phys.* **2006**, *124*, 244310.
- (52) Kawai, S.; Komatsuzaki, T. Dynamic pathways to mediate reactions buried in thermal fluctuations. I. Time-dependent normal form theory for multidimensional Langevin equation. *J. Chem. Phys.* **2009**, *131*, 224505.
- (53) Craven, G. T.; Bartsch, T.; Hernandez, R. Persistence of transition state structure in chemical reactions driven by fields oscillating in time. *Phys. Rev. E* **2014**, *89*, 040801(R).
- (54) Craven, G. T.; Bartsch, T.; Hernandez, R. Communication: Transition State Trajectory Stability Determines Barrier Crossing Rates in Chemical Reactions Induced by Time-Dependent Oscillating Fields. *J. Chem. Phys.* **2014**, *141*, 041106.
- (55) Craven, G. T.; Bartsch, T.; Hernandez, R. Chemical reactions induced by oscillating external fields in weak thermal environments. *J. Chem. Phys.* **2015**, *142*, 074108.
- (56) Junginger, A.; Hernandez, R. Uncovering the geometry of barrierless reactions using Lagrangian descriptors. *J. Phys. Chem. B* **2016**, *120*, 1720.
- (57) Junginger, A.; Craven, G. T.; Bartsch, T.; Revuelta, F.; Borondo, F.; Benito, R. M.; Hernandez, R. Transition state geometry of driven chemical reactions on time-dependent double-well potentials. *Phys. Chem. Chem. Phys.* **2016**, *18*, 30270.
- (58) Junginger, A.; Hernandez, R. Lagrangian descriptors in dissipative systems. *Phys. Chem. Chem. Phys.* **2016**, *18*, 30282.
- (59) Bartsch, T.; Uzer, T.; Hernandez, R. Stochastic transition states: Reaction geometry amidst noise. *J. Chem. Phys.* **2005**, *123*, 204102.
- (60) Kawai, S.; Bandrauk, A. D.; Jaffé, C.; Bartsch, T.; Palacián, J.; Uzer, T. Transition state theory for laser-driven reactions. *J. Chem. Phys.* **2007**, *126*, 164306.
- (61) Lichtenberg, A. J.; Leibermann, M. A. *Regular and Stochastic Motion*; Springer: New York, 1982.
- (62) Ott, E. *Chaos in dynamical systems*, second edition ed.; Cambridge University Press: Cambridge, 2002.
- (63) Wiggins, S. *Normally hyperbolic invariant manifolds in dynamical systems*; Springer Science & Business Media, 2013; Vol. 105.
- (64) Junginger, A.; Duvenbeck, L.; Feldmaier, M.; Main, J.; Wunner, G.; Hernandez, R. Chemical dynamics between wells across a time-dependent barrier: Self-similarity in the Lagrangian descriptor and reactive basins. *J. Chem. Phys.* **2017**, *147*, 06401.
- (65) Garcia-Müller, P. L.; Hernandez, R.; Benito, R. M.; Borondo, F. The role of the CN vibration in the activated dynamics of LiNC LiCN isomerization in an argon solvent at high temperatures. *J. Chem. Phys.* **2014**, *141*, 074312.
- (66) García-Müller, P. L.; Hernandez, R.; Benito, R. M.; Borondo, F. Detailed study of the direct numerical observation of the Kramers turnover in

- the LiNC=LiCN isomerization rate. *J. Chem. Phys.* **2012**, *137*, 204301.
- (67) Ulusoy, I. S.; Stanton, J. F.; Hernandez, R. Effects of roaming trajectories on the transition state theory rates of a reduced-dimensional model of ketene isomerization. *J. Phys. Chem. A* **2013**, *117*, 7553–7560.
- (68) Ulusoy, I. S.; Stanton, J. F.; Hernandez, R. Correction to ‘Effects of roaming trajectories on the transition state theory rates of a reduced-dimensional model of ketene isomerization’. *J. Phys. Chem. A* **2013**, *117*, 10567.
- (69) Ulusoy, I. S.; Hernandez, R. Revisiting roaming trajectories in ketene isomerization at higher dimensionality. *Theor. Chem. Acc.* **2014**, *133*, 1528.
- (70) Komatsuzaki, T.; Berry, R. S. Regularity in chaotic reaction paths. II. Ar<sub>6</sub>. Energy dependence and visualization of the reaction bottleneck. *Phys. Chem. Chem. Phys.* **1999**, *1*, 1387–1397.
- (71) Li, C.-B.; Shoujiguchi, A.; Toda, M.; Komatsuzaki, T. Dynamical Hierarchy in Transition States in Reactions. *Few-Body Systems* **2006**, *38*, 173–179.
- (72) Li, C.-B.; Toda, M.; Komatsuzaki, T. Bifurcation of no-return transition states in many-body chemical reactions. *J. Chem. Phys.* **2009**, *130*, 124116.
- (73) P. Hänggi,; P. Talkner,; M. Borkovec, Reaction-rate theory: fifty years after Kramers. *Rev. Mod. Phys.* **1990**, *62*, 251.
- (74) Chandler, D. Statistical mechanics of isomerization dynamics in liquids and the transition state approximation. *J. Chem. Phys.* **1978**, *68*, 2959–2970.
- (75) Miller, W. H.; Schwartz, S. D.; Tromp, J. Quantum mechanical rate constants for biomolecular reactions. *jcp* **1983**, *79*, 4889.
- (76) Mancho, A. M.; Wiggins, S.; Curbelo, J.; Mendoza, C. Lagrangian Descriptors: A Method for Revealing Phase Space Structures of General Time Dependent Dynamical Systems. *Commun. Nonlinear Sci. Numer. Simul.* **2013**, *18*, 3530 – 3557.
- (77) Davis, M. J.; Gray, S. K. Unimolecular reactions and phase space bottlenecks. *J. Chem. Phys.* **1986**, *84*, 5389.
- (78) Townsend, D.; Lahankar, S. A.; Lee, S. K.; Chambreau, S. D.; Suits, A. G.; Zhang, X.; Rheinecker, J. L.; Harding, L. B.; Bowman, J. M. The Roaming Atom: Straying from the Reaction Path in Formaldehyde Decomposition. *Science* **2004**, *306*, 1158–1161.
- (79) Maugière, F. A. L.; Collins, P.; Ezra, G.; Farantos, S. C.; Wiggins, S. Multiple transition states and roaming in ion-molecule reactions: A phase space perspective. *Chem. Phys. Lett.* **2014**, *592*, 282–287.
- (80) Bowman, J. M.; Suits, A. G. Roaming Reactions: The Third Way. *Phys. Today* **2011**, *64*, 33–37.
- (81) Bowman, J. M. Roaming. *Mol. Phys.* **2014**, *112*, 2516–2528.
- (82) Blank, T. B.; Brown, S. D.; Calhoun, A. W.; Doren, D. J. Neural network models of potential energy surfaces. *J. Chem. Phys.* **1995**, *103*, 4129–4137.
- (83) Behler, J.; Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **2007**, *98*, 146401.
- (84) Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **2011**, *134*, 074106.
- (85) Cui, J.; Krems, R. V. Efficient non-parametric fitting of potential energy

- surfaces for polyatomic molecules with Gaussian processes. *J. Phys. B* **2016**, *49*, 224001.
- (86) Vargas-Hernández, R. A.; Guan, Y.; Zhang, D. H.; Krems, R. V. Bayesian optimization for the inverse scattering problem in quantum reaction dynamics. *arXiv preprint arXiv:1711.06376* **2017**,
- (87) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; Von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **2012**, *108*, 058301.
- (88) Cui, J.; Krems, R. V. Gaussian Process model for collision dynamics of complex molecules. *Phys. Rev. Lett.* **2015**, *115*, 073202.
- (89) Cui, J.; Li, Z.; Krems, R. V. Gaussian process model for extrapolation of scattering observables for complex molecules: From benzene to benzonitrile. *J. Chem. Phys.* **2015**, *143*, 154101.
- (90) Faber, F. A.; Lindmaa, A.; Von Lilienfeld, O. A.; Armiento, R. Machine Learning Energies of 2 Million Elpasolite (A B C 2 D 6) Crystals. *Phys. Rev. Lett.* **2016**, *117*, 135502.
- (91) Huang, B.; von Lilienfeld, O. A. The “DNA” of chemistry: Scalable quantum machine learning with “amons”. *arXiv preprint arXiv:1707.04146* **2017**,
- (92) Manzhos, S.; Carrington Jr., T. A random-sampling high dimensional model representation neural network for building potential energy surfaces. *J. Chem. Phys.* **2006**, *125*, 084109.
- (93) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016; <http://www.deeplearningbook.org>.
- (94) Nielsen, M. A. *Neural Networks and Deep Learning*; Determination Press, 2015; <http://neuralnetworksanddeeplearning.com/>.
- (95) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–538.
- (96) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. **2015**, Software available from tensorflow.org.
- (97) Kamath, A.; Vargas-Hernández, R. A.; Krems, R. V.; Carrington, Jr., T.; Manzhos, S. Neural networks vs Gaussian process regression for representing potential energy surfaces: A comparative study of fit quality and vibrational spectrum accuracy. *J. Chem. Phys.* **2018**, *148*, 241702.
- (98) Essers, R.; Tennyson, J.; Wormer, P. E. S. An SCF potential energy surface for Lithium Cyanide. *Chem. Phys. Lett.* **1982**, *89*, 223–227.
- (99) Brocks, G.; Tennyson, J.; van der Avoird, A. Ab initio dipole surfaces, vibrationally averaged dipole moments, and infrared transition intensities for KCN and LiCN. *J. Chem. Phys.* **1984**, *80*, 3223–3233.
- (100) Gezelter, J. D.; Miller, W. H. Resonant Features in the Energy Dependence of the Rate of Ketene Isomerization. *J. Chem. Phys.* **1995**, *103*, 7868–7876.
- (101) Maugière, F. A. L.; Collins, P.; Ezra, G.; Farantos, S. C.; Wiggins, S. Roaming dynamics in ketene isomerization. *Theor. Chem. Acta* **2014**, *133*, 1507.
- (102) Ramachandran, P.; Varoquaux, G. Mayavi: 3D Visualization of Scientific Data. *Computing in Science & Engineering* **2011**, *13*, 40–51.