

RESEARCH ARTICLE | JULY 16 2025

## Implicit neural representations for chemical reaction paths

Kalyan Ramakrishnan ; Lars L. Schaaf ; Chen Lin ; Guangrun Wang ; Philip Torr 



*J. Chem. Phys.* 163, 034109 (2025)

<https://doi.org/10.1063/5.0267023>



### Articles You May Be Interested In

Predicting solvation free energies with an implicit solvent machine learning potential

*J. Chem. Phys.* (December 2024)

Machine learning implicit solvation for molecular dynamics

*J. Chem. Phys.* (August 2021)

Implicit solvent approach based on generalized Born and transferable graph neural networks for molecular dynamics simulations

*J. Chem. Phys.* (May 2023)

## AIP Advances

### Why Publish With Us?

-  **21DAYS**  
average time to 1st decision
-  **OVER 4 MILLION**  
views in the last year
-  **INCLUSIVE**  
scope

[Learn More](#)



# Implicit neural representations for chemical reaction paths

Cite as: J. Chem. Phys. 163, 034109 (2025); doi: 10.1063/5.0267023

Submitted: 21 February 2025 • Accepted: 27 June 2025 •

Published Online: 16 July 2025



View Online



Export Citation



CrossMark

Kalyan Ramakrishnan,<sup>1,a)</sup> Lars L. Schaaf,<sup>2,3</sup> Chen Lin,<sup>1</sup> Guangrun Wang,<sup>1</sup> and Philip Torr<sup>1,b)</sup>

## AFFILIATIONS

<sup>1</sup> University of Oxford, Oxford, United Kingdom

<sup>2</sup> Engineering Laboratory, University of Cambridge, Cambridge CB2 1PZ, United Kingdom

<sup>3</sup> Cavendish Laboratory, University of Cambridge, Cambridge CB3 0HE, United Kingdom

<sup>a)</sup> Author to whom correspondence should be addressed: [kalyanr@robots.ox.ac.uk](mailto:kalyanr@robots.ox.ac.uk)

<sup>b)</sup> URL: <https://torrvision.com/>

## ABSTRACT

We show that neural networks can be optimized to represent minimum energy paths as continuous functions, offering a flexible alternative to discrete path-search methods such as Nudged Elastic Band (NEB). Our approach parameterizes reaction paths with a network trained on a loss function that discards tangential energy gradients and enables instant estimation of the transition state. We first validate the method on two-dimensional potentials and then demonstrate its advantages over NEB on challenging atomistic systems where (i) poor initial guesses yield unphysical paths, (ii) multiple competing paths exist, or (iii) the reaction follows a complex multi-step mechanism. Results highlight the versatility of the method: for instance, a simple adjustment to the sampling strategy during optimization can help escape local-minimum solutions. Finally, in a low-dimensional setting, we demonstrate that a single neural network can learn from existing paths and generalize to unseen systems, showing promise for a universal reaction path representation.

© 2025 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 International (CC BY-NC-ND) license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>). <https://doi.org/10.1063/5.0267023>

## I. INTRODUCTION

Understanding the mechanisms of chemical reactions and predicting their rates are central goals in computational chemistry, with wide-ranging applications in catalysis, biochemistry, and materials science.<sup>1,2</sup> A key concept is the minimum energy path (MEP), which traces a valley on the potential energy surface, connecting given stable configurations *A* and *B*. The MEP approximates the most probable route between the reactants and products of a chemical reaction.<sup>3</sup> The highest-energy configuration along this path, the transition state (TS), defines the energy barrier of the reaction and largely determines its kinetics via transition state theory.<sup>4,5</sup> Consequently, identifying MEPs and transition states is an essential step toward guiding the design of semiconductors,<sup>6</sup> catalysts,<sup>7–10</sup> and drugs.<sup>11</sup>

A well-established computational technique for MEP search is the Nudged Elastic Band (NEB),<sup>12</sup> which optimizes a discrete chain of configurations (called images) between the end points *A* and *B*. The images are connected by springs and repeatedly nudged in

directions perpendicular to the chain to minimize energy and converge on the MEP. A subsequent modification, known as climbing-image NEB,<sup>13</sup> additionally drives the highest-energy image up the energy surface to estimate the TS simultaneously. While these methods remain widely adopted, their reliance on a discrete representation of the underlying continuous reaction path can lead to difficulties.<sup>14,15</sup> NEB can produce unphysical configurations in the absence of a good initial guess,<sup>16,17</sup> can converge to suboptimal solutions during optimization,<sup>18</sup> and may fail to capture complex reaction paths.<sup>15</sup>

Inspired by the recent success of Implicit Neural Representations (INRs)<sup>19–22</sup> in modeling continuous functions, we parameterize reaction paths with a neural network representing a smooth function of a reaction coordinate. An INR is a neural network that parameterizes an implicitly defined function,<sup>20</sup> in our case, the MEP, defined by the constraint that the energy gradient remains aligned with the path tangent. The continuous formulation naturally supports automatic differentiation,<sup>23,24</sup> enabling precise computation of path tangents, curvatures, and parameter gradients while enjoying

the expressivity of neural networks. This allows us to incorporate physical priors into the training algorithm, such as the nudging and climbing mechanisms from NEB, without requiring a discrete representation. It also allows flexibility in the loss function and sampling strategy for optimization.

We demonstrate the capabilities of our implicit neural representation for MEP and TS search through experiments on systems ranging from simple two-dimensional potentials to challenging material and molecular reactions for which the conventional approach of NEB fails. Results show that the INR reliably locates transition states in these settings, overcoming the limitations of NEB described earlier. Finally, we show that, unlike existing methods that require a fresh optimization for each new system, an INR can learn from existing paths to instantly approximate MEPs for unseen systems, opening the door for a universal reaction path representation.

## II. RELATED WORK

We are not the first to utilize continuous functions for MEP or TS search. Below, we summarize some relevant work.

### A. Discrete representations

Several methods<sup>16,25–27</sup> optimize a discrete representation of the reaction path but repeatedly fit a continuous curve to estimate tangents or redistribute points along the path. Reference 16 fits cubic splines and approximates the TS via interpolation. Reference 26 also fits cubic splines but runs a separate optimization for TS search once an approximate MEP is found and does not discard tangential energy gradients. Reference 27 fits piecewise-linear curves within a variational formulation but does not focus on TS search. In contrast to these methods, we directly update a continuous neural network representation of the reaction path, combining MEP and TS search in a single optimization.

### B. Continuous representations

Other methods<sup>28–31</sup> explicitly work with a continuous representation of the reaction path. Reference 28 represents the path with a cubic spline but only optimizes it for TS search, and consequently, the path itself has no physical meaning. Reference 29 updates a B-spline curve representation and approximates the TS with the highest-energy sample observed on the path. While the curve may be fit to an improved initial guess,<sup>17</sup> the method has not been compared against NEB and has primarily demonstrated success on systems requiring very few optimization steps. Reference 30 adopts a variational framework and represents the path with a linear combination of basis functions. Despite involving a non-linear, constrained optimization, the method does not, in practice, find MEPs for even two-dimensional potentials,<sup>30</sup> limiting its applicability for path-search in atomistic settings. References 31 and 32 use neural networks to model a distribution over stochastic trajectories connecting the end states rather than optimizing for MEP or TS search.

To our knowledge, we are the first to use neural networks to represent MEPs and estimate transition states. Unlike prior work, we combine the nudging and climbing mechanisms in the same

optimization. Moreover, all previous methods lack the ability to condition the path representation on arbitrary end states for potential generalization across systems.

## III. METHOD

### A. Overview

For an  $N$ -atom system, let  $\vec{A} \in \mathbb{R}^{3N}$  and  $\vec{B} \in \mathbb{R}^{3N}$  represent the coordinates of all atoms in the initial and final states, respectively. We assume that the reaction path  $\vec{x}(t) \in \mathbb{R}^{3N}$ , with  $t \in [0, 1]$ , takes the form

$$\vec{x}_\theta(t) = \vec{b}(t) + t(1-t)\vec{g}_\theta(t), \quad (1)$$

where  $\vec{b}(t)$  is a *base* path and  $\vec{g}_\theta : [0, 1] \rightarrow \mathbb{R}^{3N}$  is a neural network with learnable parameters  $\theta$ . Note that  $\vec{x}_\theta(0) = \vec{A}$  and  $\vec{x}_\theta(1) = \vec{B}$  are satisfied by construction as long as  $\vec{b}_\theta(0) = \vec{A}$  and  $\vec{b}_\theta(1) = \vec{B}$ . Typically, we set the base path as the linear interpolation from  $\vec{A}$  to  $\vec{B}$ ,

$$\vec{b}(t) = (1-t)\vec{A} + t\vec{B}. \quad (2)$$

In general, given  $n+1$  points  $\{\vec{x}_0, \vec{x}_1, \dots, \vec{x}_n\}$  sampled from a reference path, we may set  $\vec{b}(t)$  as the Lagrange interpolating polynomial,

$$\vec{b}(t) = \sum_{j=0}^n \left[ \prod_{k \neq j} \left( \frac{t-t_k}{t_j-t_k} \right) \right] \vec{x}_j, \quad (3)$$

with  $t_j = j/n$  for  $j \in \{0, 1, \dots, n\}$ .

In either case, the path  $\vec{x}_\theta(t)$  is differentiable with respect to both  $t$  and  $\theta$ , allowing us to compute exact tangents, curvatures, and parameter gradients. For instance, we can differentiate through the potential energy  $U(\vec{x})$  of the system as follows, assuming access to its gradient  $\nabla_x U(\vec{x})$ :

$$\frac{\partial U(\vec{x}_\theta(t))}{\partial \theta} = \nabla_x U(\vec{x}_\theta(t))^T \frac{\partial \vec{x}_\theta(t)}{\partial \theta}. \quad (4)$$

One objective is to tune the parameters of the network to find the minimum energy path (MEP), which follows a valley on the potential energy surface, connecting the local minima  $\vec{A}$  and  $\vec{B}$ . This requires that the energy gradient  $\nabla_x U(\vec{x}_\theta(t))$  has no component orthogonal to the path tangent  $\vec{x}'_\theta(t) = d\vec{x}_\theta(t)/dt$  over  $t \in [0, 1]$ . More importantly, we are interested in the transition state (TS), the highest-energy point on the MEP from  $\vec{A}$  to  $\vec{B}$ . The energy at this point (relative to the starting point) is the barrier of the reaction. We approach this problem by sampling points along the path and minimizing a loss function over the parameters  $\theta$ , as described next.

### B. Loss function

A simple choice for the loss function is the expected energy along the path with an additional term to encourage a uniform *speed*, e.g.,

$$L(\theta) = \mathbb{E}_t[U(\vec{x}_\theta(t))] + \lambda_s \text{Var}_t[\|\vec{x}'_\theta(t)\|], \quad (5)$$

where  $\mathbb{E}_t$  and  $\text{Var}_t$  are the expectation and variance over some chosen distributions  $q_u(t)$  and  $q_s(t)$ , respectively, and  $\lambda_s > 0$  is a hyperparameter. A similar loss function was adopted previously<sup>29</sup> to optimize B-spline curves, using the uniform distribution  $u(0, 1)$  for both. However, since the gradient  $\nabla_\theta \tilde{L}(\theta)$  also includes components of  $\nabla_x U(\tilde{x}_\theta(t))$  tangential to the path, this loss encourages sampled points to depart from higher-energy regions on it, relying entirely on the second term to reduce this effect (see Appendix B for details).

Our loss function is inspired by the Nudged Elastic Band (NEB),<sup>12</sup> which optimizes a discrete set of points by nudging each point along the projection of  $-\nabla_x U(\tilde{x})$  orthogonal to the (estimated) path tangent. Specifically, we remove the tangential components using the stop gradient operator, which behaves as an identity function that has a zero gradient during automatic differentiation. We define the modified energy,

$$\tilde{U}(\tilde{x}_\theta(t)) := U(\tilde{x}_\theta(t)) - \text{Stop}[\nabla_x U(\tilde{x}_\theta(t))]_{\parallel}^T \tilde{x}_\theta(t), \quad (6)$$

where

$$\nabla_x U(\tilde{x}_\theta(t))_{\parallel} = \left( \frac{\nabla_x U(\tilde{x}_\theta(t))^T \tilde{x}'_\theta(t)}{\|\tilde{x}'_\theta(t)\|^2} \right) \tilde{x}'_\theta(t) \quad (7)$$

is the projection of  $\nabla_x U(\tilde{x}_\theta(t))$  tangential to the path. Using  $\tilde{U}(\tilde{x}_\theta(t))$  in Eq. (5) gives the updated loss function,

$$\tilde{L}(\theta) = \mathbb{E}_t[\tilde{U}(\tilde{x}_\theta(t))] + \lambda_s \text{Var}_t[\|\tilde{x}'_\theta(t)\|]. \quad (8)$$

As intended, the loss gradient  $\nabla_\theta \tilde{L}(\theta)$  now discards components of  $\nabla_x U(\tilde{x}_\theta(t))$  tangential to the path since

$$\begin{aligned} \nabla_x \tilde{U}(\tilde{x}_\theta(t)) &= \nabla_x U(\tilde{x}_\theta(t)) - \nabla_x U(\tilde{x}_\theta(t))_{\parallel} \\ &= \nabla_x U(\tilde{x}_\theta(t))_{\perp} \end{aligned} \quad (9)$$

is the projection of  $\nabla_x U(\tilde{x}_\theta(t))$  orthogonal to the path, producing nudging updates during optimization in our continuous setting. We theoretically justify the use of the modified energy  $\tilde{U}$  (vs  $U$ ) in Appendix C.

The stop gradient is a standard feature in popular deep learning libraries,<sup>24</sup> making it straightforward to use in a loss function for neural network training. Additional loss terms to exploit prior knowledge about the reaction or explicitly promote alignment with the energy gradient may further help, but they are beyond the scope of this paper.

Another advantage of our formulation is that it does not evaluate the energy Hessian,  $H_{pq} = \frac{\partial^2 U(\tilde{x})}{\partial x_p \partial x_q}$ , which is in general more expensive than the gradient by a factor proportional to the system size.<sup>33–36</sup>

### C. Improving TS localization

While the loss  $\tilde{L}(\theta)$  can be numerically minimized (by sampling from  $q_u$  and  $q_s$ ) to estimate the MEP, locating the TS would still

require a continuous search along the resulting path. Moreover, such an optimization would only passively improve the TS estimate (via improving the path). To avoid this, analogous to the climbing-image NEB,<sup>13</sup> we modify the loss so that at each update, the highest-energy sample is pushed up the energy surface, tangential to the path. Specifically, given samples  $\{t_i\}$  from  $q_u$ , if  $t_* = \arg \max_i U(\tilde{x}_\theta(t_i))$ , our climbing loss is

$$\tilde{L}_{\text{Cl}}(\theta) = \tilde{L}(\theta; \lambda_s) - \lambda_{\text{Cl}} \text{Stop}[\nabla_x U(\tilde{x}_\theta(t_*))]_{\parallel}^T \tilde{x}_\theta(t_*), \quad (10)$$

where  $\lambda_{\text{Cl}} > 0$  is a hyperparameter. The intuition for this choice is that  $\nabla_\theta \tilde{L}_{\text{Cl}}(\theta)$  essentially removes  $\nabla_x U(\tilde{x}_\theta(t_*))_{\parallel}$  twice, first zeroing it [see Eq. (9)] and then flipping its direction [due to the second term in Eq. (10)], thereby producing a climbing effect. After numerically minimizing  $\tilde{L}_{\text{Cl}}(\theta)$ , we locate the TS at  $\tilde{x}_\theta(t_*)$ , obtaining both the MEP and TS from a single optimization.

### D. Sampling strategies

Typically, we let  $q_u = q_s = u(0, 1)$  and compute the mean and variance in Eq. (8) with  $n + 1$  equidistant (“uniform”) samples  $t_i = i/n$ ,  $i \in \{0, 1, \dots, n\}$ . While this is a simple default, more complex sampling strategies may better suit the system or potential at hand, such as dynamically adapting  $q_u$  or varying the number of samples used.

We focus on one such strategy in this paper inspired by the growing string method (GSM),<sup>16</sup> which, conceptually, optimizes a path by gradually growing it from the end points  $(A, B)$  until the two ends merge. The idea is to update the path starting from where the solution is known/easier to find (near the ends) and gradually expand toward the center. In practice, GSM maintains a discrete, increasing set of points at each end and optimizes both sets iteratively with the following steps: (i) nudging the points (as in NEB), (ii) fitting a cubic spline, (iii) adding new points when appropriate, and (iv) repositioning points along the spline for a uniform spacing within each end. In contrast, our approach provides a simple and elegant implementation of this concept by just modifying the sampling strategy for Eq. (8). Specifically, for the energy loss (first term), rather than sampling uniformly from  $t \in [0, 1]$ , we could sample from the ends by setting

$$q_u(t) = \begin{cases} 1/\alpha_k & \text{if } t \in \left[0, \frac{\alpha_k}{2}\right] \cup \left[1 - \frac{\alpha_k}{2}, 1\right] \\ 0 & \text{if } t \in \left(\frac{\alpha_k}{2}, 1 - \frac{\alpha_k}{2}\right), \end{cases} \quad (11)$$

where  $\alpha_k \in (0, 1]$  is gradually increased with each iteration  $k$ . The simplest choice is to linearly expand the sampled region toward the center with  $\alpha_k = \frac{k}{N}$ , where  $N$  is the maximum number of iterations over which the loss function is optimized. Empirically, we find that growing sampling can improve the optimization trajectory to avoid local minimum paths, as we show in Sec. IV. Other sampling strategies are beyond the scope of this paper.

We summarize our overall training procedure in Algorithm 1 and provide Python code on GitHub.<sup>37</sup>

**ALGORITHM 1.** Training procedure for MEP and TS search.

---

```

1: for  $k = 1$  to  $n_{\text{iters}}$  do
2:   if uniform sampling then
3:      $\{t_i\} \leftarrow \text{Uniform}([0, 1], n_{\text{samples}})$  ▷  $n_{\text{samples}}$  equidistant inputs in  $[0, 1]$ 
4:   else if growing sampling then
5:      $f_k \leftarrow k/n_{\text{iters}}$ 
6:      $\{t_i^{(l)}\} \leftarrow \text{Uniform}([0, f_k/2], n_{\text{samples}}/2)$ 
7:      $\{t_i^{(r)}\} \leftarrow \text{Uniform}([1 - f_k/2, 1], n_{\text{samples}}/2)$ 
8:      $\{t_i\} \leftarrow \{t_i^{(l)}\} \cup \{t_i^{(r)}\}$  ▷  $n_{\text{samples}}$  inputs from the ends
9:   end if
10:   $\{\bar{x}_\theta(t_i)\}, \{\bar{x}'_\theta(t_i)\} \leftarrow \text{INR}_\theta\{t_i\}$ 
11:   $\{U(\bar{x}_\theta(t_i))\}, \{\nabla_x U(\bar{x}_\theta(t_i))\} \leftarrow \text{Potential}\{\bar{x}_\theta(t_i)\}$ 
12:   $\{\tilde{U}(\bar{x}_\theta(t_i))\} \leftarrow \{U(\bar{x}_\theta(t_i)) - \text{Stop}[\|\nabla_x U(\bar{x}_\theta(t_i))\|]^T \bar{x}_\theta(t_i)\}$  ▷ modified energies
13:   $t_* \leftarrow \arg \max_{t_i} U(\bar{x}_\theta(t_i))$ 
14:   $\text{Loss}(\theta) \leftarrow \text{mean}_i\{\tilde{U}(\bar{x}_\theta(t_i))\} + \lambda_s \text{var}_i\{\|\bar{x}'_\theta(t_i)\|\} - \lambda_{\text{CI}} \text{Stop}[\|\nabla_x U(\bar{x}_\theta(t_*))\|]^T \bar{x}_\theta(t_*)$ 
15:  Update  $\theta$  based on  $\frac{\partial \text{Loss}(\theta)}{\partial \theta}$ 
16: end for
17: MEP  $\leftarrow \bar{x}_\theta(\cdot)$ 
18: TS  $\leftarrow \bar{x}_\theta(t_*)$ 

```

---

**IV. EXPERIMENTS**

In this section, we first validate our method (INR) on two-dimensional systems and then focus on challenging high-dimensional, atomistic systems for which the traditional approach of NEB fails. We show that our method performs reliably in such settings.

**A. Implementation details**

We use the Atomic Simulation Library (ASE)<sup>38</sup> for the climbing NEBs, with the Fast Inertial Relaxation Engine (FIRE)<sup>39</sup> optimizer, 15 intermediate (17 total) images, and spring constant  $k = 0.1$ . We consistently observed faster convergence with FIRE than with other ASE optimizers. For the INRs, we use a three-layer feedforward neural network having 256 hidden features with the tanh activation and a linear output layer. To be consistent with the NEBs, we sample 17 inputs for training and inference, although a different number may be used for inference. By default, we set  $\lambda_s = 0$  and  $\lambda_{\text{CI}} = 1.0$  for uniform sampling (see Appendix B for details). For growing sampling (Sec. III D), we use  $\lambda_s = 0.1$  (and  $\lambda_{\text{CI}} = 0$ ) to maintain a balanced distribution of points throughout as the sampling region expands from the ends. The networks are trained with the Adam optimizer<sup>40</sup> and a default learning rate of  $1e-3$ . We use linear interpolation as the INR base path and NEB initialization and optimize both methods for a maximum of 500 iterations. The end states are first relaxed to equilibrium using L-BFGS<sup>41</sup> for atomistic systems. The potential energy function is a variant of MACE<sup>42</sup> specific to the system type, enabling fast evaluation of energies and gradients. To assess the practical cost of locating the TS, we wait until the root mean square loss gradient reaches  $\text{RMS}(\nabla_\theta L(\theta)) < 1e-3$  (for uniform sampling). We then tightly refine the TS estimate  $\bar{x}(t_*)$  using Sella,<sup>43</sup> a saddle point optimizer, until the largest per-atom  $\nabla_x U(\bar{x}(t_*))$  norm reaches  $F_{\text{max}} < 5e-4$ , for a maximum of 500 iterations. For growing

sampling, we always train the network up to a pre-defined iteration count (e.g., 200) since the sampled region expands according to this number. On average, refinement needed <3% of the method's energy evaluations to converge.

**B. Two-dimensional systems****1. Standard potentials**

Figures 1(a) and 2(a) show the path produced by the INR and the energy profiles of both methods for the standard London–Eyring–Polanyi–Sato (LEPS)<sup>12,44</sup> potential. Both methods converge to the same path and correct energy barrier of 1.34. In Fig. 13 (Appendix A), we plot the TS energy error  $|U(\bar{x}(t_*)) - U(\bar{x}_{\text{TS}})|$  vs iterations, showing that the methods approach the TS at a similar rate. We also show results for the popular Müller–Brown (MB)<sup>16,28,45</sup> potential in Figs. 1(b) and 2(b), where once again the INR matches the NEB path. The exact forms of the potentials, along with the initial and final minima, are provided in Appendix A.

**2. Global optimization**

We design a new potential energy surface to evaluate TS search methods on systems having multiple competing paths but only one with the lowest barrier. This surface tests the ability of methods to escape local minimum energy paths in favor of the global solution. The surface is periodic in the  $y$  direction, with adjacent pairs of minima connected by two MEPs: (i) the nearly straight-line path connecting the minima and (ii) a curved path with a smaller barrier. The potential takes the following form:

$$U_{\text{Sine}}(x, y) = \left(1 - e^{-6x^2 - \cos^2(\pi y)}\right) \times \left(1 - 0.1e^{-100x^2} - 0.6e^{-20(x+0.5 \cos(\pi y))^2}\right) + 0.1x^2. \quad (12)$$

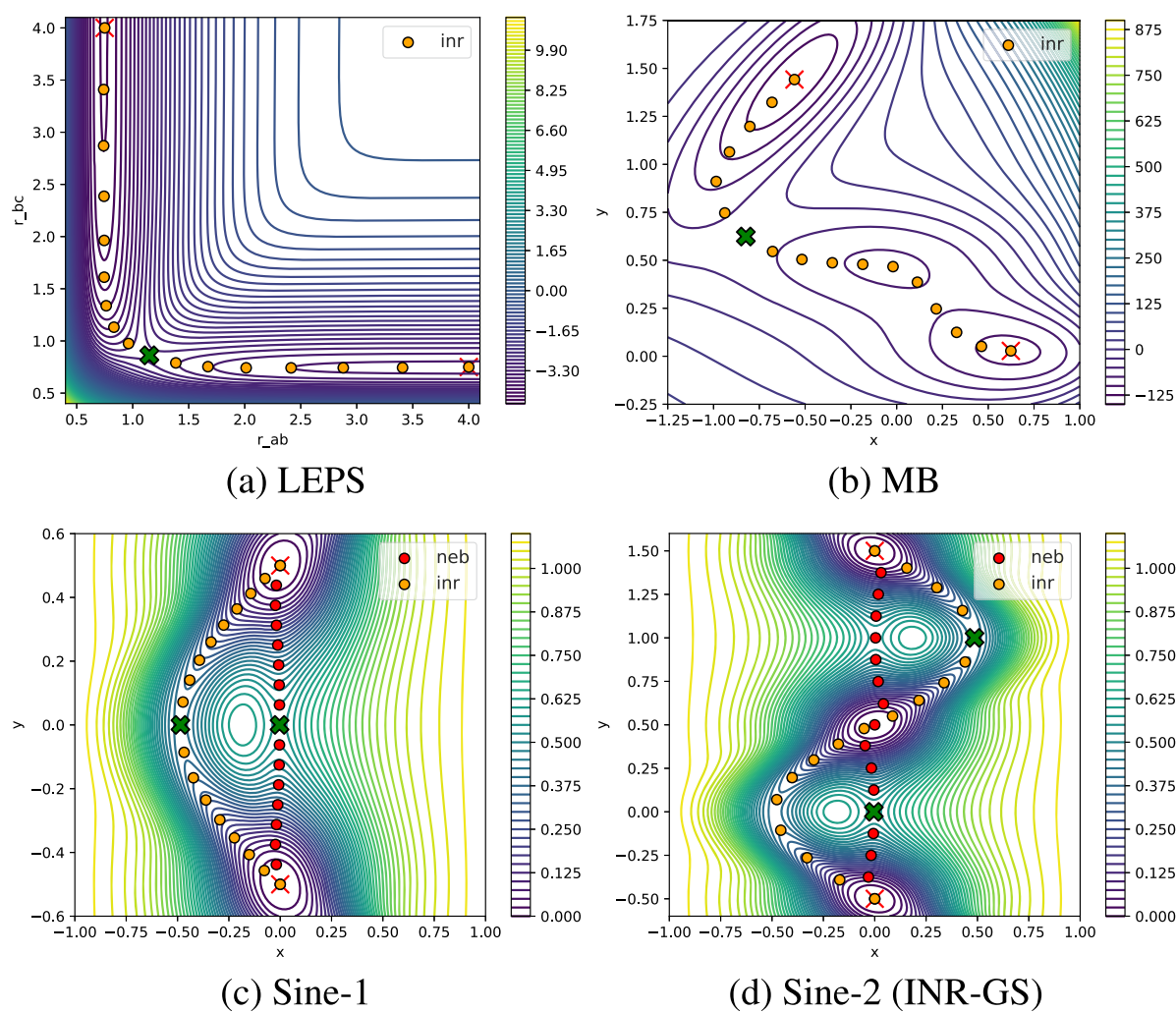


FIG. 1. INR paths (orange) for four 2D potentials [(a)–(d)]. Panels (c) and (d) show incorrect, higher-barrier paths (red) predicted by NEB.

The local minima are at  $\{(0, y_n) \mid y_n = n - \frac{1}{2}, n \in \mathbb{Z}\}$ , and the period along the  $y$  direction is 2.

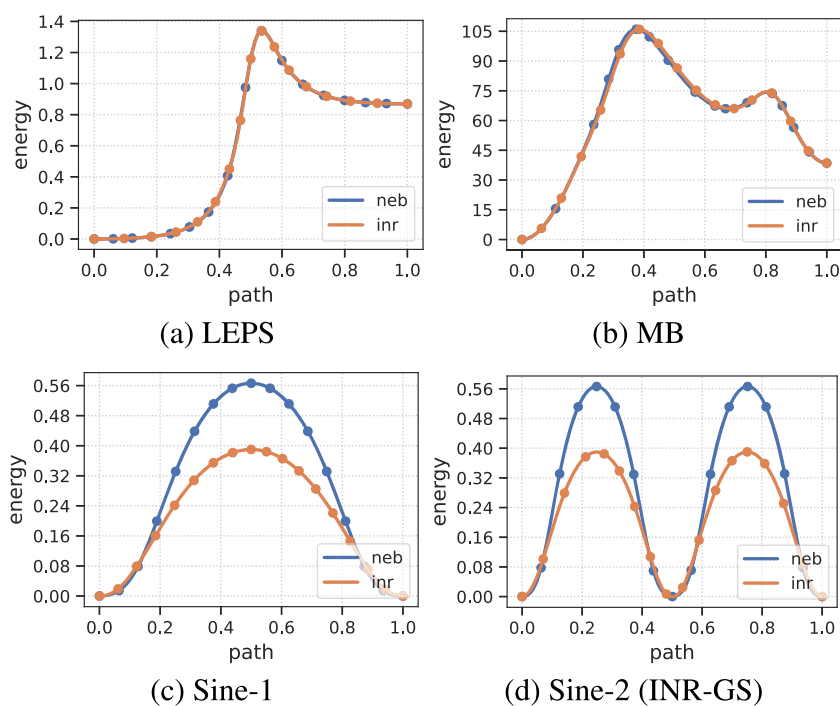
Figures 1(c) and 2(c) show NEB and INR results for the end points  $(0, y_0) \rightarrow (0, y_1)$ . We call this system sine-1. While NEB gets stuck in the direct path with a barrier of 0.566, the INR identifies the curved path with a smaller barrier of 0.39. For the more complex sine-2 system, having end points  $(0, y_0) \rightarrow (0, y_2)$ , neither method finds the curved path. However, by applying the growing sampling idea from Sec. III D, the INR (named INR-GS) finds the lower-barrier curved path, as shown in Figs. 1(d) and 2(d). We also present the evolution of this path over iterations and results for the sine-3 system in Appendix D, where similar behavior is observed.

### C. $\text{In}_2\text{O}_3$

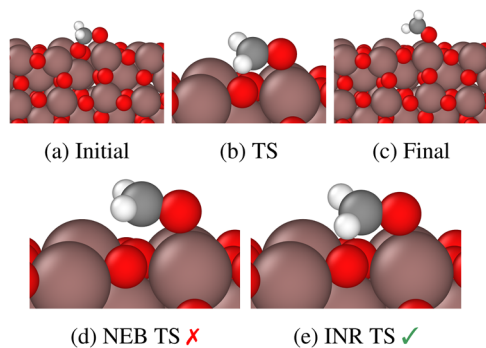
We begin by testing the methods on a basic chemical reaction step. Here, we consider the conversion of an adsorbed

dioxymethylene ( $\text{H}_2\text{CO}_2^*$ ) species into formaldehyde ( $\text{H}_2\text{CO}^*$ ) on an  $\text{In}_2\text{O}_3$  surface with an oxygen-vacant active site. This is a key step in the hydrogenation of carbon dioxide to methanol over an indium oxide catalyst.<sup>46,47</sup> Figures 3(a) and 3(c) depict the initial and final states of this reaction step. The system is periodic in the  $x$  and  $y$  directions, with the bottom two (of four)  $\text{In}_2\text{O}_3$  layers fixed, and is modeled with the MACE-MP<sup>48</sup> potential.

Figures 3(d) and 3(e) show the estimated TS from both methods, and Fig. 3(b) shows the reference TS obtained by refining either estimate with Sella. Visually, the INR estimates a more accurate TS with energy 1.2364 eV, close to the refined value 1.2362 eV (Table I). The TS energy error plot in Fig. 10(a) further shows that the INR approaches the TS faster than NEB. In Fig. 9(a), we plot the energy profiles of both methods and for NEB initialized with the INR path. The INR also finds a better approximation of the MEP, aligning closely with the NEB-refined path. However, the advantage for this



**FIG. 2.** Energy profiles for the 2D potentials. NEB (blue) and INR (orange) find identical paths for standard systems [(a) and (b)]. NEB gets stuck in higher-barrier solutions for some other systems [(c) and (d)]. (a) LEPS. (b) MB. (c) Sine-1. (d) Sine-2 (INR-GS).



**FIG. 3.** Conversion of a dioxymethylene species into formaldehyde on an  $\text{In}_2\text{O}_3$  surface. Compared to NEB, the INR identifies a more accurate TS in the same number of iterations. (a) Initial. (b) TS. (c) Final. (d) NEB TS  $\times$ . (e) INR TS  $\checkmark$ .

system is marginal in practice. Both methods find the correct TS within a similar number of energy evaluations when refined after around 50 iterations, as summarized in Table I.

#### D. Alanine dipeptide rotation

A key limitation of NEB is the need for a reasonable initial guess for the reaction path. In the worst case, a poor guess may lead to unphysical atomic configurations, causing the method, or the potential energy calculation, to diverge.<sup>16,17</sup> A well-known example is the 22-atom alanine dipeptide ( $\text{C}_6\text{H}_{12}\text{N}_2\text{O}_2$ ) reaction, which involves

two simultaneous dihedral rotations along the path between the initial and final states shown in Figs. 4(a) and 4(c). Here, the system is modeled with the MACE-OFF<sup>49</sup> potential.

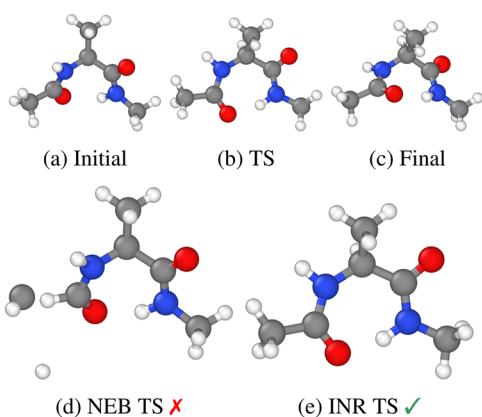
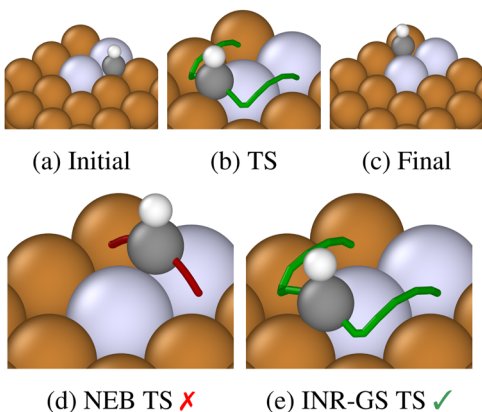
A linear interpolation of the Cartesian coordinates introduces steric clashes, leading to large energy gradients. Consequently, NEB predicts a TS with broken bonds, as shown in Fig. 4(d), and a discontinuous path, as evident from the energy profile in Fig. 9(b). Sella refinement fails to recover from this unphysical state (Table I). In contrast, the INR produces a physically meaningful path, preserving atomic connectivity through the dihedral rotations. The TS is shown in Fig. 4(e) and has energy 0.4544 eV, close to the refined value 0.4070 eV. The error curve in Fig. 10(b) reveals that the INR quickly recovers from the high energy, unnatural, initial guess and finds a reasonable estimate of the TS in around 100 iterations. Thus, unlike NEB, the INR does not rely on a specialized initialization; instead, it leverages its inherent smoothness to learn the reaction path robustly.

#### E. Methylidyne diffusion

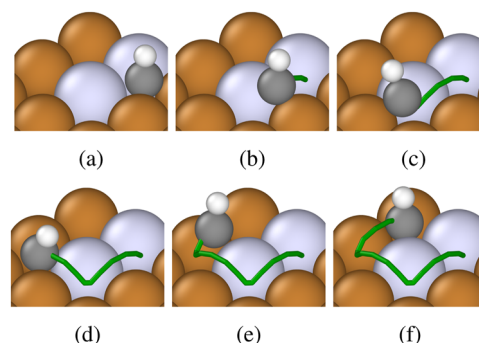
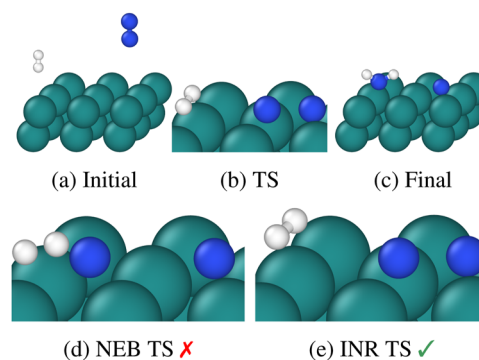
Another limitation of NEB is its tendency to become trapped in local minimum paths, illustrated earlier on the two-dimensional sine- $n$  systems (Sec. IV B). Here, we consider an atomistic setting where the same issue arises. The system involves the surface diffusion of an adsorbed methylidyne ( $^*\text{CH}$ ) species on a heterogeneous catalyst surface composed of Cu doped with Ag atoms, shown in Figs. 5(a) and 5(c). It is periodic in the  $x$  and  $y$  directions, with six fixed layers of Cu, and is modeled with the MACE-MP potential. A direct path crosses the Ag–Ag bridge site and is a valid MEP but encounters a larger barrier. An alternative, lower-barrier route avoids the Ag–Ag bridge, traversing the Cu–Ag sites instead.

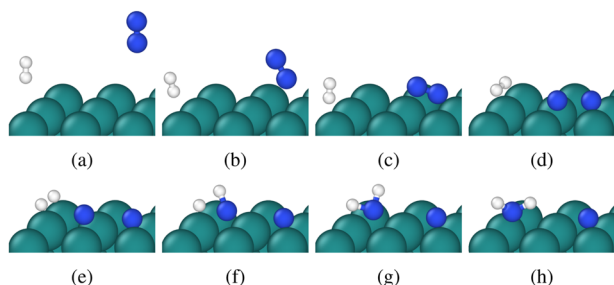
**TABLE I.** Quantitative results. “No refinement” runs just the method for 500 iterations (200 for INR-GS). Best energies are in bold.

System	Method	(No refinement)		(With refinement)		
		TS energy (eV)	Method iters	Sella energy evals	Total energy evals	TS energy (eV)
In <sub>2</sub> O <sub>3</sub>	NEB	1.2585	50	66	818	1.2362
	INR	<b>1.2364</b>	50	83	835	1.2362
Alanine dipeptide	NEB	11.5239	500	561	8063	11.0014
	INR	<b>0.4544</b>	200	44	3046	<b>0.4070</b>
Methyldiyne diffusion	NEB	2.3539	100	7	1509	2.3539
	INR-GS	<b>1.3720</b>	200	28	3030	<b>1.4142</b>
N <sub>2</sub> + H <sub>2</sub>	NEB	0.0533	500	38	7540	-0.4379
	INR	<b>0.4131</b>	200	64	3066	<b>0.3415</b>


**FIG. 4.** Dihedral rotation of alanine dipeptide. NEB produces an unphysical TS. The INR preserves bond integrity and estimates the TS reliably. Both methods start from a linear interpolation path. (a) Initial. (b) TS. (c) Final. (d) NEB TS  $\times$ . (e) INR TS  $\checkmark$ .

**FIG. 5.** Diffusion of a methyldiyne species on a Ag/Cu surface. NEB is trapped in the higher-barrier direct path. The INR (growing sampling) finds the lower-barrier route avoiding the Ag–Ag bridge. (a) Initial. (b) TS. (c) Final. (d) NEB TS  $\times$ . (e) INR-GS TS  $\checkmark$ .

NEB gets trapped in the direct path as indicated by the estimated TS in Fig. 5(d) and the energy profile in Fig. 9(c). On the other hand, applying the growing sampling idea from Sec. III D, the INR identifies the lower-barrier path with (refined) TS energy 1.4142 eV, compared to 2.3539 eV for the direct path. In Fig. 6, we show more


**FIG. 6.** Samples [(a)–(f)] along the INR-GS path for the methyldiyne diffusion system. The adsorbed species passes over the Cu–Ag sites, avoiding the higher-barrier Ag–Ag bridge.

**FIG. 7.** Reaction of N<sub>2</sub> and H<sub>2</sub> to form adsorbed \*NH<sub>2</sub> on a Ru surface. NEB fails to converge and produces an incorrect TS. The INR TS correctly represents the rate-limiting N<sub>2</sub> dissociation step. (a) Initial. (b) TS. (c) Final. (d) NEB TS  $\times$ . (e) INR TS  $\checkmark$ .



**FIG. 8.** Samples [(a)–(h)] along the INR path for the  $N_2 + H_2$  system, showing  $N_2$  dissociation (d),  $H_2$  dissociation (e), and  $^*N$  hydrogenation [(f) and (g)].

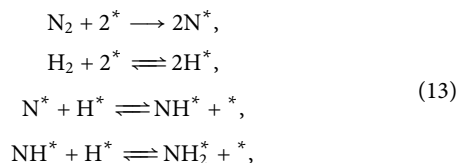
samples along the INR path, and in Appendix E, we present its evolution over iterations. Our results clearly highlight the versatility of the approach, showing how a simple adjustment to the sampling strategy can help escape local minima.

## F. $N_2 + H_2$

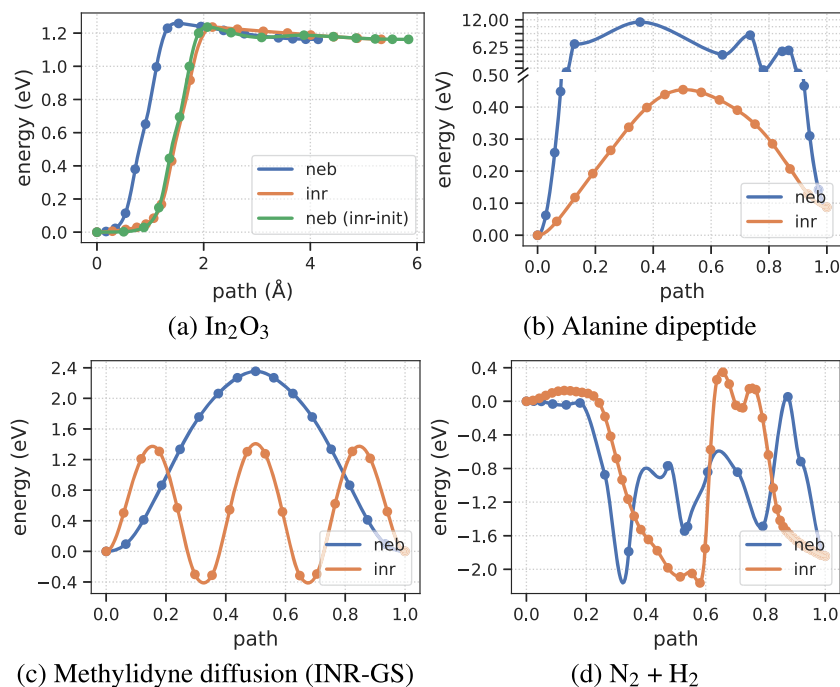
To test the ability of the INR to represent more complex reaction paths, we consider a reaction sequence in which  $N_2$  and  $H_2$  interact on a Ru catalyst surface, leading to the formation of an adsorbed  $^*NH_2$  intermediate. This marks the beginning of the dissociative pathway for ammonia synthesis, involving  $N_2$  and  $H_2$  dissociation followed by hydrogenation of the adsorbed  $^*N$  atoms.<sup>50</sup> The chosen initial and final states are shown in Figs. 7(a) and 7(c). The system is periodic in the  $x$  and  $y$  directions, with the bottom Ru

layer fixed. We model it with a MACE potential trained specifically for this system (see Appendix F for details).

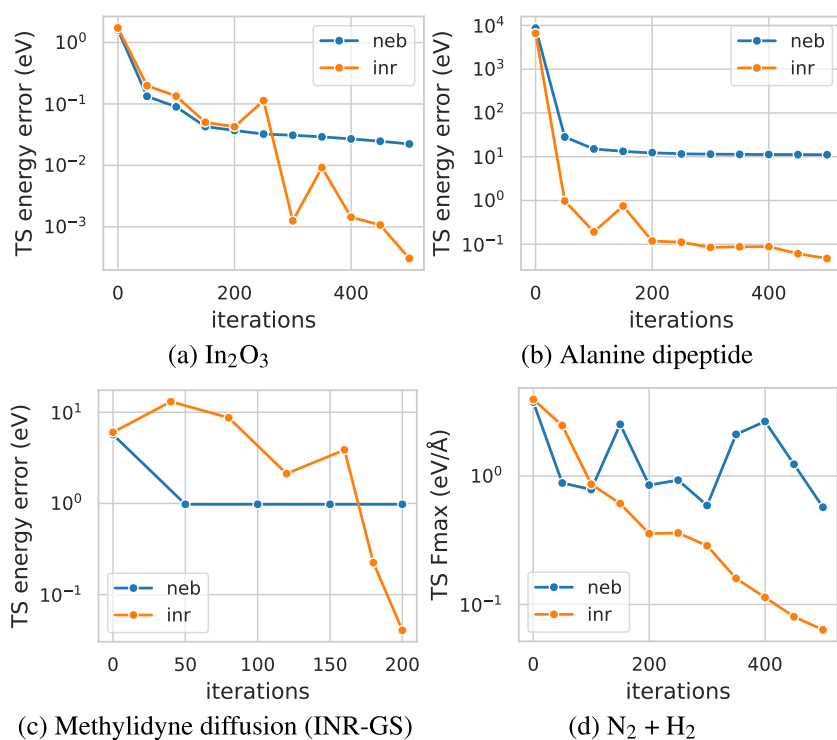
Figure 8 shows samples along the path learned by the INR. Qualitatively, the INR identifies the elementary steps of this complex reaction within a single optimization, consistent with the established mechanism (for an Fe surface<sup>50</sup>) given by



where  $^*$  indicates an empty site on the surface. The estimated TS, shown in Fig. 7(e), correctly corresponds to the rate-limiting<sup>50,51</sup>  $N_2$  dissociation step. This appears as the highest point on the energy profile in Fig. 9(d) (showing 50 samples for visualization only). The next peak (following the barrierless  $H_2$  dissociation<sup>50</sup>) visually corresponds to the first hydrogenation [Figs. 8(e) and 8(f)], while no separate peak is observed for the second hydrogenation. Note that we sample 15 inputs for training, prioritizing just the highest energy point [Eq. (10)]. A more refined sampling strategy may better capture additional complexities. In contrast, NEB fails to converge for this multi-step reaction, as evident from the  $F_{\max}$  plot in Fig. 10(d), resulting in an incorrect TS estimate. NEB would likely require a separate run for each manually defined elementary step.<sup>15</sup> Thus, our results highlight the INR's ability to automatically express a complex reaction path, significantly reducing the need for manual supervision.



**FIG. 9.** Energy profiles for chemical systems where the INR (a) better approximates an MEP, (b) is resilient to a poor initial guess, (c) finds a lower-barrier solution, and (d) captures a complex path. (a)  $In_2O_3$ . (b) Alanine dipeptide. (c) Methylidyne diffusion (INR-GS). (d)  $N_2 + H_2$ .



**FIG. 10.** Training progress for chemical systems where the INR (a) approaches a TS faster, (b) recovers from a bad start, (c) avoids a suboptimal path, and (d) predicts the TS for a complex system. (a)  $\text{In}_2\text{O}_3$ . (b) Alanine dipeptide. (c) Methylidyne diffusion (INR-GS). (d)  $\text{N}_2 + \text{H}_2$ .

## V. PATH GENERALIZATION

Section IV showed that neural networks can learn to represent reaction paths when optimized *separately* for individual systems. We now ask the following question: could a single neural network learn from existing MEPs to instantly predict paths for multiple *unseen* systems? This would represent a considerable leap over all existing methods, which start for each system from scratch. Assuming that a dataset of known MEPs for related systems is available, a neural network could be trained on this data to be applied to unseen systems where evaluating the potential energy surface is prohibitively expensive. Such a network must also take as inputs the initial and final states  $\{\vec{A}, \vec{B}\}$  and, optionally, parameters  $\phi$  characterizing the potential energy of the system. The reaction path would then take the following form:

$$\vec{x}_\theta(t; \vec{A}, \vec{B}, \phi) = \vec{b}(t; \vec{A}, \vec{B}) + t(1-t) \vec{g}_\theta(t; \vec{A}, \vec{B}, \phi). \quad (14)$$

The path and TS estimated by the network for an unseen system may serve as a quality initial guess for energy-based optimization. In this paper, we test this *path generalization* idea on a set of two-dimensional systems as a proof of concept, leaving the challenging extension to atomistic systems as future work. Our chosen potential energy surface has the following form, where  $\mathcal{N}(x, y; \vec{\mu}, \sigma)$  is the normal density with mean  $\vec{\mu}$  and standard deviation  $\sigma$ :

$$U_{\text{wells}}(x, y) = -\mathcal{N}(x, y; \vec{\mu} = (0, 0), \sigma = 0.4) - \mathcal{N}(x, y; \vec{\mu} = (0, 1), \sigma = 0.4) + c_p \mathcal{N}(x, y; \vec{\mu} = \vec{\mu}_p, \sigma = \sigma_p). \quad (15)$$

The first two terms are fixed negative wells centered at  $(0, 0)$  and  $(0, 1)$ , and the third term is a variable peak centered at  $\vec{\mu}_p$  with a scaling factor  $c_p$ . We consider the set of systems formed by varying the parameter vector  $\phi = (\mu_{px}, \mu_{py}, \sigma_p, c_p)$  over the following range, chosen so the peak has a meaningful effect on the MEP:  $\mu_{px} \in [0.15, 0.85]$ ,  $\mu_{py} \in [-0.15, 0.15]$ ,  $\sigma_p \in [0.05, 0.25]$ , and  $c_p \in [0.2, 1.0]$ . For each system, the end points  $\vec{A}$  and  $\vec{B}$  are the local minima of the potential near the wells  $(0, 0)$  and  $(0, 1)$ , respectively.

To create training and testing datasets, we first form a grid by discretizing each dimension of  $\phi$  into seven equidistant values and then randomly sample  $n_{\text{train}}$  and  $n_{\text{test}}$  disjoint systems from this grid. We use the L-BFGS optimizer for each system to locate the two end points, followed by climbing NEB with the FIRE optimizer to compute the MEP (see Sec. IV A for details). The neural network  $\vec{g}_\theta(t; \vec{A}, \vec{B}, \phi)$  inherits the architecture from Sec. IV A, with the addition of the system-specific inputs, and is trained to minimize the error between the predicted and actual paths (discretized) for each system. We use the following loss function at iteration  $k$ :

$$L_{\text{path}}^k(\theta) = \mathbb{E}_t \left[ \|\vec{x}_\theta(t; \vec{A}^k, \vec{B}^k, \phi^k) - \vec{x}^k(t)\|^2 \right], \quad (16)$$

where  $(\vec{A}^k, \vec{B}^k, \phi^k)$  specifies the  $k$ th training system,  $\vec{x}^k(t)$  is the true (NEB) path, and  $\mathbb{E}_t$  is the average over equidistant samples  $t_i = i/n$ ,  $i \in \{0, 1, \dots, n\}$ . We train the network for 200 epochs using the Adam optimizer with a  $5e-4$  learning rate. Our evaluation metric is the average TS energy error over the test systems.

Figure 11 plots the test error vs epochs for INRs trained on  $n_{\text{train}} = 8, 16,$  and  $32$  systems (with  $n_{\text{test}} = 100$ ). In each case, we compare with two baselines: (i) linear interpolation, which predicts a straight line connecting the end points of the test system, and (ii) nearest neighbor, which predicts a path by transforming the MEP of the training system having parameters closest to those of the test system. Specifically, for a test system specified by  $(\vec{A}, \vec{B}, \phi)$ , we first identify the *nearest* training system  $n = \text{argmin}_j \|\phi^j - \phi\|$  and its

MEP  $\vec{x}^n(t)$  and then affine-transform this path so its end points match  $(\vec{A}, \vec{B})$ ,

$$\vec{x}(t) = \vec{A} + \frac{\|\vec{B} - \vec{A}\|}{\|\vec{B}^n - \vec{A}^n\|} \mathbf{R} \cdot (\vec{x}^n(t) - \vec{A}^n), \quad (17)$$

where  $\mathbf{R}$  is the rotation matrix that aligns the direction  $(\vec{B}^n - \vec{A}^n)$  with  $(\vec{B} - \vec{A})$ . Note that  $\vec{x}(0) = \vec{A}$  and  $\vec{x}(1) = \vec{B}$  by construction. We

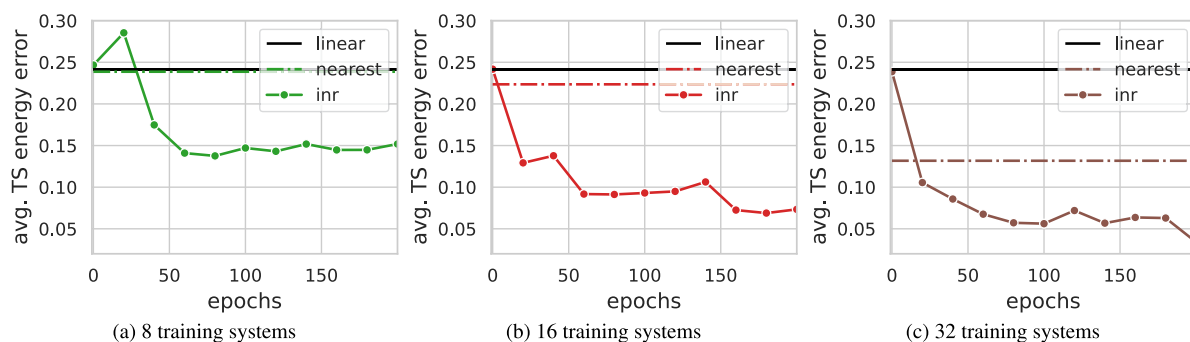


FIG. 11. Average TS energy error on unseen systems vs training epochs. The INR improves with more training examples and outperforms the nearest neighbor baseline, demonstrating its ability to generalize to new systems. (a) 8 training systems. (b) 16 training systems. (c) 32 training systems.

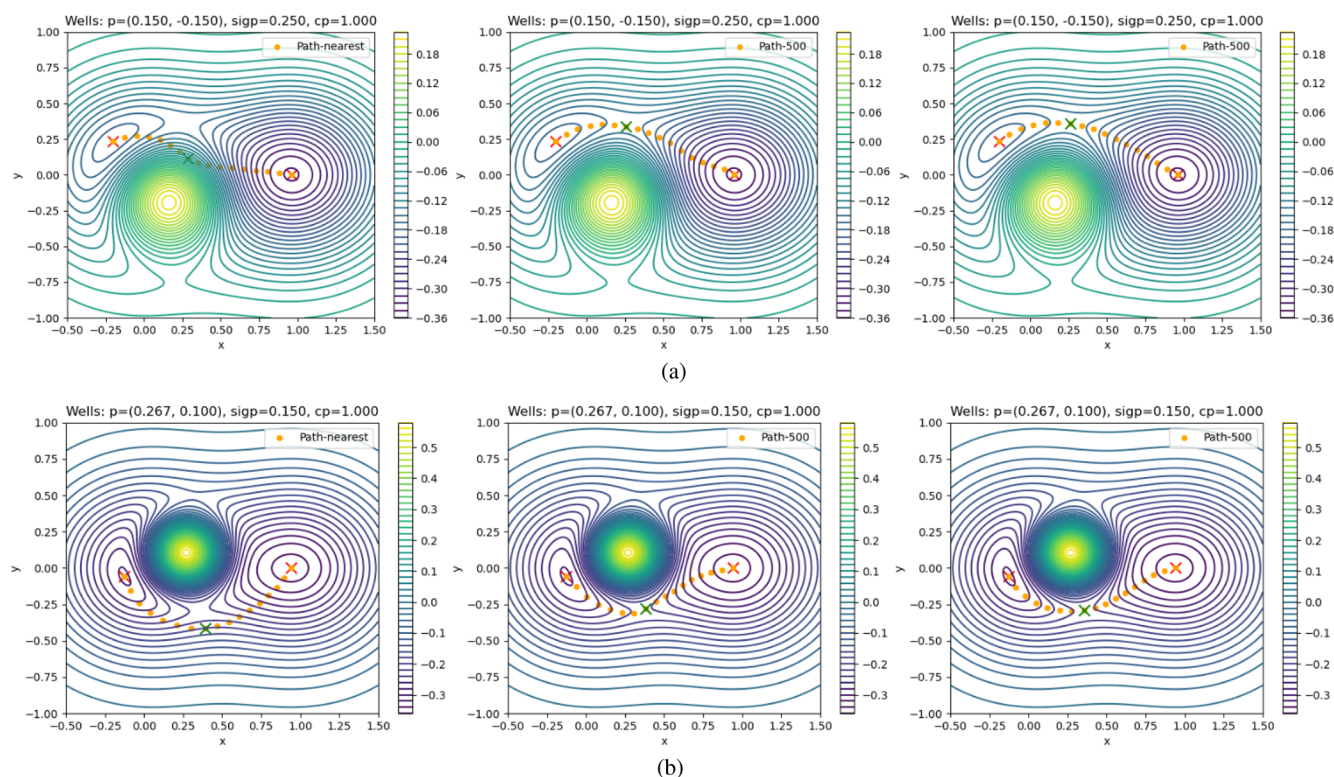


FIG. 12. Nearest neighbor (left), INR (middle), and true (right) paths for two unseen systems [(a) and (b)]. The INR learns from existing paths to instantly approximate MEPs for new systems, aligning more closely with the true path compared to the nearest neighbor baseline.

evaluate this baseline to ensure that the network's apparent generalization is not simply a result of memorizing the training data. From Fig. 11, we see that the INR outperforms both baselines at locating the TS, generalizing to new systems despite being trained on limited data (e.g., when  $n_{\text{train}} = 16$ ). While the INR's advantage over the nearest neighbor baseline reduces when provided with 32 vs 16 training systems in our low-dimensional setting, we anticipate a persistent improvement in high-dimensional, atomistic settings, where simple parameterizations of systems will not be feasible. Future work should investigate the potential benefits of neural networks in such scenarios and whether learning a universal path representation is possible. Figure 12 shows two example predicted paths on unseen systems (for  $n_{\text{train}} = 32$ ) along with the nearest neighbor and true paths, indicating that the INR learns to produce a good approximation of the MEP. More examples are in Appendix G.

## VI. CONCLUSION

We presented a method that optimizes neural networks to smoothly represent reaction paths while permitting direct estimation of the transition state. Through experiments on challenging material and molecular systems, we demonstrated that it overcomes some key limitations of Nudged Elastic Band (NEB). In particular, our approach (i) shows resilience to unnatural states arising from poor initial guesses, (ii) is easily adapted to escape local-minimum solutions, and (iii) automatically learns a complex multi-step reaction path with no manual supervision. We also showed that by conditioning the network on initial and final states, it has the potential to learn from existing paths and generalize to multiple unseen reactions, paving the way for a generalizable reaction path representation.

We end with some suggestions for future work. First, since our approach allows flexibility in the loss function, base path, and sampling strategy for optimization, different choices guided by chemical intuition may further improve performance. This equally applies to inductive biases in the neural network architecture. For example, instead of predicting the Cartesian coordinates of atoms, the network could be designed to operate in the internal coordinate<sup>52</sup> space. Second, while we used the same number of samples at training and inference to compare with NEB directly, one might improve efficiency by amortizing training with fewer, randomly sampled inputs and using stochastic gradient descent, following standard practice in deep learning. On a related note, in our growing sampling technique, the sampled region expands according to a pre-defined iteration count, limiting a full coverage of the input space to the last few iterations. This explains the slow initial progress in Fig. 10(c), suggesting room for improvement. Third, it would be helpful to understand the effect of the choice of optimizer to train the network. While we used Adam, a standard optimizer for neural networks, integrating ideas from physics-based<sup>39</sup> or ODE solver-based<sup>53</sup> optimizers is an interesting alternative. Finally, training networks that can directly predict paths and transition states for unseen systems in high-dimensional, atomistic settings could be an impactful direction for future work. While our preliminary evidence supports this idea, building a practical, generalizable reaction path predictor would require both a large-scale dataset<sup>54</sup> and an efficient model design that supports variable-sized outputs. Incorporating known symmetries into the network architecture through group equivariant graph

neural networks<sup>42,55</sup> represents a promising direction toward a scalable and universal reaction path representation.

## ACKNOWLEDGMENTS

K.R. is supported by the EPSRC Center for Doctoral Training in Autonomous Intelligent Machines and Systems EP/S024050/1. L.L.S. is supported by the EPSRC Center for Doctoral Training in Chemical Synthesis EP/S024220/1 and Wolfson College, University of Cambridge.

## AUTHOR DECLARATIONS

### Conflict of Interest

The authors have no conflicts to disclose.

## Author Contributions

**Kalyan Ramakrishnan:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Lars L. Schaaf:** Conceptualization (equal); Data curation (equal); Investigation (equal); Methodology (equal); Supervision (equal); Validation (equal); Visualization (equal); Writing – original draft (equal). **Chen Lin:** Conceptualization (supporting); Formal analysis (supporting); Investigation (supporting); Methodology (supporting); Supervision (supporting); Validation (supporting); Visualization (equal); Writing – review & editing (equal). **Guangrun Wang:** Conceptualization (supporting); Data curation (equal); Investigation (supporting); Supervision (supporting); Visualization (supporting); Writing – review & editing (supporting). **Philip Torr:** Resources (lead).

## DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## APPENDIX A: TWO-DIMENSIONAL POTENTIALS

Figure 13 plots the TS energy error  $|U(\vec{x}(t_*)) - U(\vec{x}_{\text{TS}})|$  vs iterations.

### 1. LEPS

$$U_{\text{LEPS}}(r_{ab}, r_{bc}) = E_{\text{Coulomb}} - E_{\text{exchange}}, \quad (\text{A1})$$

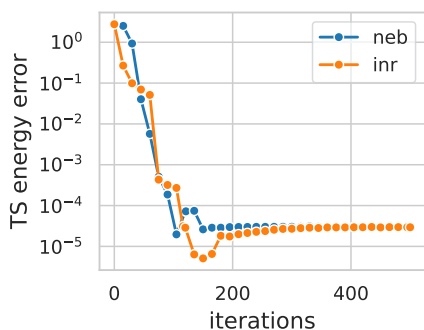
where

$$E_{\text{Coulomb}} = \frac{q_{ab}}{1+a} + \frac{q_{bc}}{1+b} + \frac{q_{ac}}{1+c}$$

and

$$E_{\text{exchange}}^2 = \frac{j_{ab}^2}{(1+a)^2} + \frac{j_{bc}^2}{(1+b)^2} + \frac{j_{ac}^2}{(1+c)^2} - \frac{j_{ab}j_{bc}}{(1+a)(1+b)} - \frac{j_{bc}j_{ac}}{(1+b)(1+c)} - \frac{j_{ab}j_{ac}}{(1+a)(1+c)}.$$

Here,  $r_{ac} = r_{ab} + r_{bc}$ , and the functions  $q = Q(r, d)$  and  $j = J(r, d)$  are defined as



**FIG. 13.** Training progress for the standard LEPS potential. Both methods approach the TS at a similar rate.

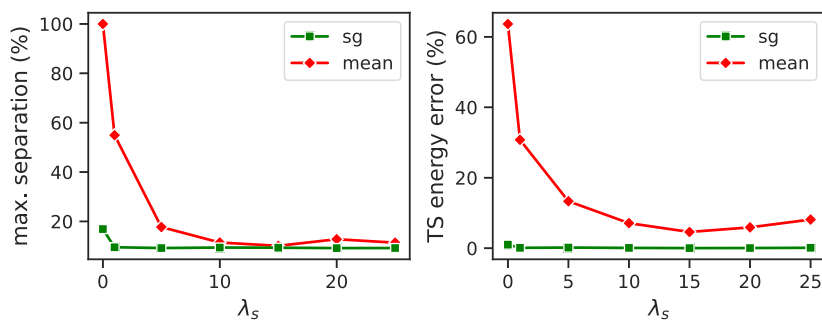
$$Q(r, d) = \frac{d}{2} \left( \frac{3}{2} e^{-2\alpha_0(r-r_0)} - e^{-\alpha_0(r-r_0)} \right),$$

$$J(r, d) = \frac{d}{4} \left( e^{-2\alpha_0(r-r_0)} - 6e^{-\alpha_0(r-r_0)} \right),$$

with parameters  $a = 0.05$ ,  $b = 0.3$ ,  $c = 0.05$ ,  $d_{ab} = 4.746$ ,  $d_{bc} = 4.746$ ,  $d_{ac} = 3.445$ ,  $r_0 = 0.742$ , and  $\alpha_0 = 1.942$ . The initial and final minima are  $(0.75, 4.0)$  and  $(4.0, 0.75)$ , respectively.

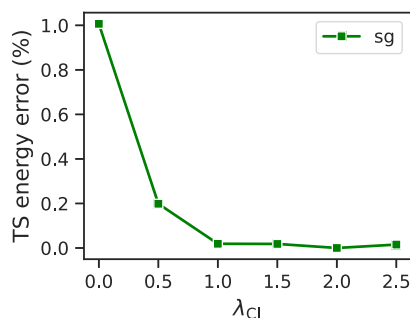
## 2. Müller-Brown

$$U_{\text{MB}}(x, y) = \sum_{i=1}^4 A_i e^{a_i(x-x_{0,i})^2 + b_i(x-x_{0,i})(y-y_{0,i}) + c_i(y-y_{0,i})^2}, \quad (\text{A2})$$



(a) Sample separation vs.  $\lambda_s$

(b) TS energy error vs.  $\lambda_s$



(c) TS energy error vs.  $\lambda_{\text{Cl}}$

with parameters

$$a = (-1, -1, -6.5, 0.7),$$

$$b = (0, 0, 11, 0.6),$$

$$c = (-10, -10, -6.5, 0.7),$$

$$x_0 = (1, 0, -0.5, -1),$$

$$y_0 = (0, 0.5, 1.5, 1),$$

$$A = (-200, -100, -170, 15).$$

The initial and final minima are  $(-0.5582, 1.4417)$  and  $(0.6235, 0.0280)$ , respectively.

## APPENDIX B: HYPERPARAMETERS

To better understand the effect of the loss function and hyperparameters ( $\lambda_s, \lambda_{\text{Cl}}$ ) on the path learned by a smooth representation like the INR, we run some experiments on the Müller-Brown potential.

First, we compare the mean-energy loss function [Eq. (5)], which retains the full energy gradient, against its stop-gradient counterpart [Eq. (8)], which discards tangential components. Specifically, we calculate the largest separation between adjacent sampled points  $\max_i \|\tilde{x}_\theta(t_{i+1}) - \tilde{x}_\theta(t_i)\|$  along the paths resulting from either loss. We plot this as a percentage of the overall displacement  $\|\vec{B} - \vec{A}\|$  in Fig. 14(a). The mean loss (“mean”) incurs a considerable separation for low values of the spacing loss coefficient  $\lambda_s$ . For  $\lambda_s = 0$ , the separation is at 100%, meaning that all samples have departed from the high energy interior, settling at one of the two end minima  $(\vec{A}, \vec{B})$ . While less severe at larger  $\lambda_s$ , this phenomenon stifles

**FIG. 14.** Loss function and hyperparameter analysis. The stop gradient loss is much less sensitive to  $\lambda_s$  with respect to sample spacing (a) and TS estimation (b). The climbing loss improves accuracy (c). (a) Sample separation vs.  $\lambda_s$ . (b) TS energy error vs.  $\lambda_s$ . (c) TS energy error vs.  $\lambda_{\text{Cl}}$ .

sampling-based learning and produces incorrect TS estimates, as evident from the TS energy error plot in Fig. 14(b). Moreover,  $\lambda_s$  must be re-tuned to neutralize tangential gradients for a different potential energy surface. In contrast, by discarding tangential components, the stop gradient loss (“sg”) is significantly less sensitive to  $\lambda_s$ , estimating the TS reliably. We next examine the climbing loss coefficient  $\lambda_{cl}$  in Eq. (10) by plotting the TS energy error (with  $\lambda_s = 0$ ) in Fig. 14(c). Setting  $\lambda_{cl}$  to around 1.0 or higher improves the accuracy of the TS while incurring no additional cost in the optimization. Thus, these simple experiments demonstrate the benefit of incorporating nudging and climbing mechanisms into the continuous representation.

### APPENDIX C: THEORETICALLY JUSTIFYING THE MODIFIED ENERGY $\tilde{U}$

To understand the effect of the stop gradient in theory, we analyze the loss functions involving  $\tilde{U}(\vec{x})$  vs  $U(\vec{x})$  using the Euler–Lagrange theorem, which we informally state here for completeness.

**Theorem:** (Euler–Lagrange). Let the functional

$$S[\vec{x}(\cdot)] = \int_a^b \mathcal{L}(t, \vec{x}(t), \vec{x}'(t)) dt$$

be defined for smooth functions  $\vec{x}(\cdot) : [a, b] \rightarrow \mathbb{R}^d$  satisfying  $\vec{x}(a) = \vec{A}$  and  $\vec{x}(b) = \vec{B}$ . If  $\vec{x}_*(t)$  is a minimizer of  $S$ , then

$$\delta S(\vec{x}_*(\cdot))(t) = 0 \quad \forall t \in [a, b],$$

where  $\delta S$  is the functional derivative of  $S$ ,

$$\delta S(\vec{x}(\cdot)) = \left. \frac{\partial \mathcal{L}}{\partial \vec{x}} - \frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \vec{x}'} \right) \right|_{\vec{x}(\cdot)}.$$

In our case, assuming uniform sampling and no spacing or climbing terms ( $\lambda_s = \lambda_{cl} = 0$ ), the stop gradient loss functional [Eq. (8)] is

$$S_{\tilde{U}}[\vec{x}(\cdot)] = \int_0^1 \tilde{U}(\vec{x}(t)) dt,$$

with functional derivative equal to the orthogonal energy gradient at  $\vec{x}(t)$  [as shown in Eq. (9)],

$$\delta S_{\tilde{U}}(\vec{x}(\cdot))(t) = \nabla_x \tilde{U}(\vec{x}(t)) = \nabla_x U(\vec{x}(t))_{\perp}.$$

This implies that the minimizer  $\vec{x}_*(\cdot)$  satisfies  $\nabla_x U(\vec{x}_*(t))_{\perp} = 0 \quad \forall t \in [0, 1]$ , which is precisely the definition of the MEP. Moreover, since  $\delta S_{\tilde{U}}$  is always orthogonal to the current path, analogous to NEB, gradient-based optimization nudges the path at each update (see Fig. 16 for a visual example).

On the other hand, the true energy loss functional [Eq. (5)] is

$$S_U[\vec{x}(\cdot)] = \int_0^1 U(\vec{x}(t)) dt,$$

with functional derivative equal to the full energy gradient at  $\vec{x}(t)$ ,

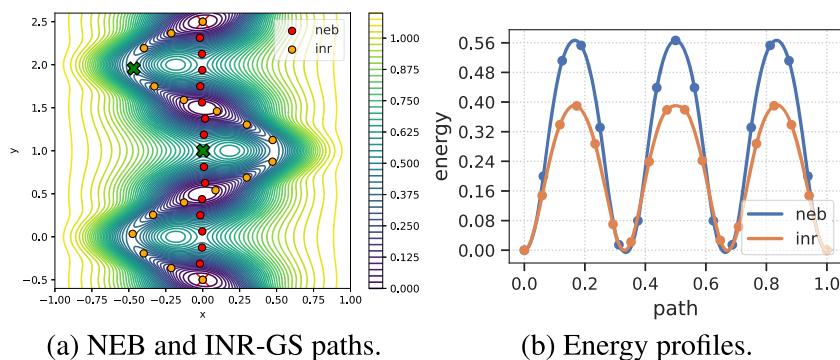
$$\delta S_U(\vec{x}(\cdot))(t) = \nabla_x U(\vec{x}(t)).$$

If a minimizer  $\vec{x}_*(\cdot)$  exists, it must satisfy  $\nabla_x U(\vec{x}_*(t)) = 0 \quad \forall t \in [0, 1]$ , which is not possible unless there is a constant-energy path connecting the end points  $\vec{A}$  and  $\vec{B}$ . Hence, while  $S_U$  does not have a minimizing path in general, it is easy in practice to *nearly* satisfy the zero-gradient condition by “wasting time” at local minima of  $U$ . An example path with this property (that wastes time at the end points) is

$$\vec{y}(t) = \begin{cases} \vec{A} & \text{if } t \in [0, t_1] \\ \vec{x}_c(t) & \text{if } t \in [t_1, t_1 + \varepsilon] \\ \vec{B} & \text{if } t \in [t_1 + \varepsilon, 1], \end{cases}$$

where  $t_1 \in (0, 1)$ ,  $0 < \varepsilon \ll 1$ , and  $\vec{x}_c(t)$  is a smooth sub-path connecting  $\vec{A}$  and  $\vec{B}$ . Empirically, a similar path results from optimizing  $S_U$  for the Müller–Brown potential, where we observe all sampled points to eventually settle at either  $\vec{A}$  or  $\vec{B}$  (see Appendix B for details). In this case, since  $\delta S_U$  includes components tangential to the current path, unlike NEB, gradient-based optimization attempts to avoid higher-energy regions on it at each update.

To summarize, (i) the stop gradient operator produces nudging updates in a continuous setting; (ii) with  $\tilde{U}(\vec{x})$ , the optimal path, in theory, has no orthogonal energy gradient and is thus an MEP; (iii) with  $U(\vec{x})$ , the resulting path in practice is generally not an MEP and may even converge to a degenerate solution as above; and (iv) the empirical benefit of  $\tilde{U}$  over  $U$  (despite balancing the latter objective with a spacing loss term) is further described in Appendix B.



**FIG. 15.** Sine-3 system results. NEB gets stuck in the direct path (red). The INR (growing sampling) finds the lower-barrier curved path (orange). (a) NEB and INR-GS paths. (b) Energy profiles.

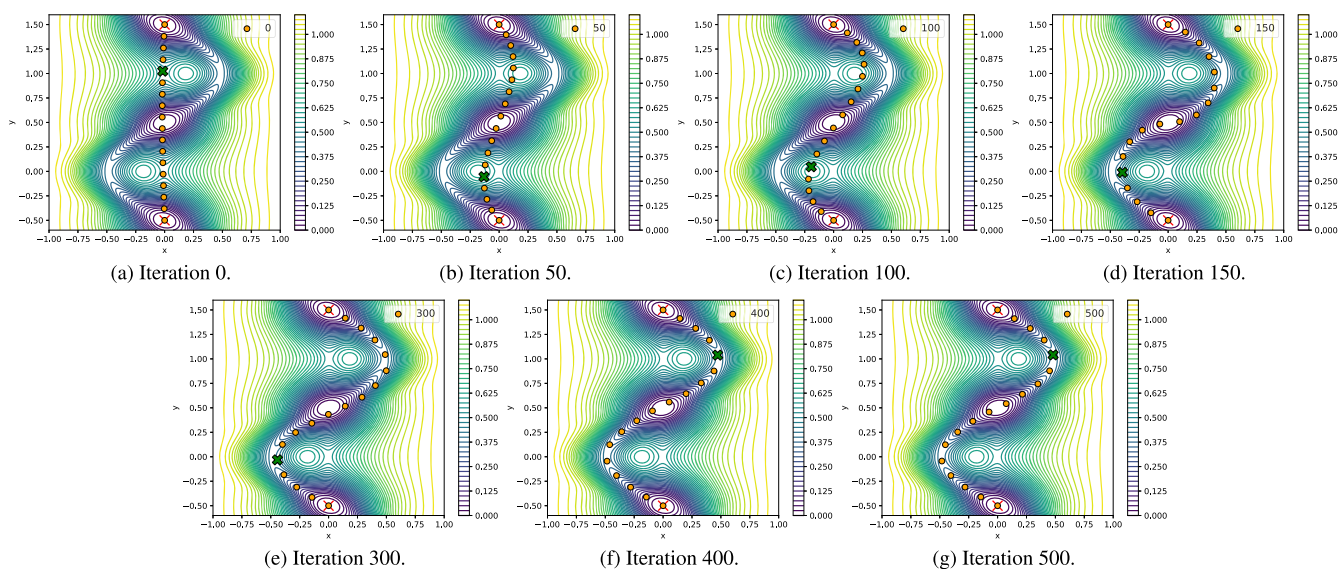


FIG. 16. INR-GS paths at different iterations for the sine-2 system. (a) Iteration 0. (b) Iteration 50. (c) Iteration 100. (d) Iteration 150. (e) Iteration 300. (f) Iteration 400. (g) Iteration 500.

#### APPENDIX D: SINE-N SYSTEMS

Figure 15 shows results for the sine-3 system. Figure 16 shows the evolution of the INR-GS path with iterations for sine-2.

#### APPENDIX E: METHYLIDYNE DIFFUSION SYSTEM

Figure 17 shows the evolution of the INR-GS path for the methylidyne diffusion system when optimized over 500 iterations.

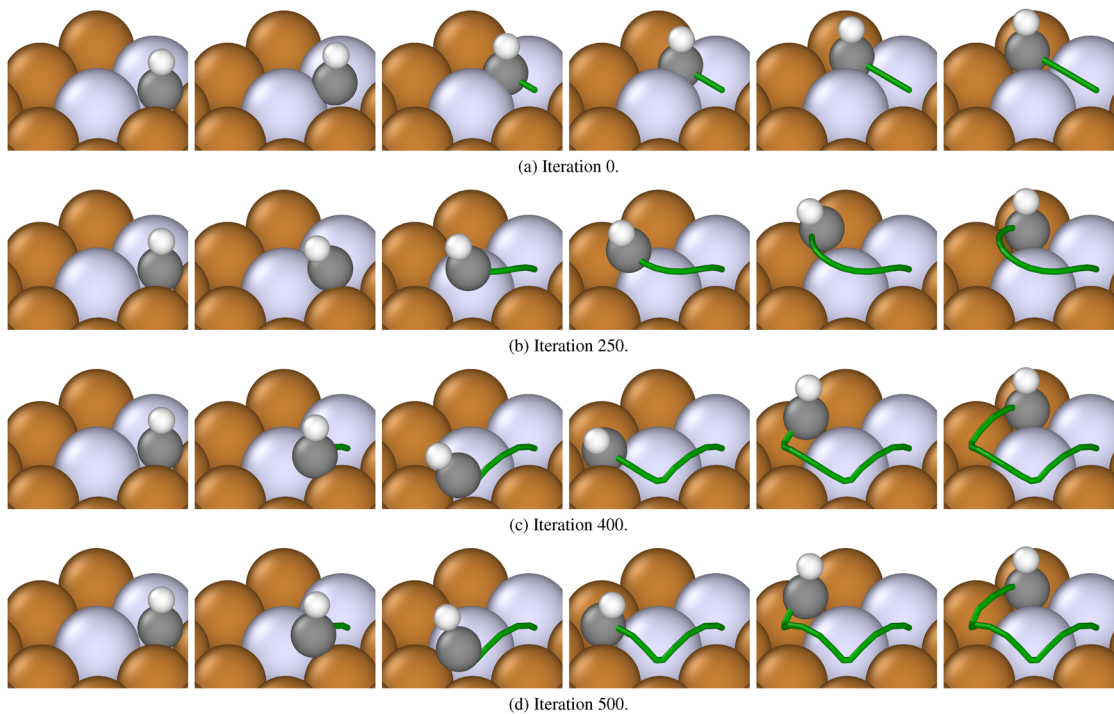


FIG. 17. INR-GS paths at different iterations for the methylidyne diffusion system. (a) Iteration 0. (b) Iteration 250. (c) Iteration 400. (d) Iteration 500.

APPENDIX F:  $\text{N}_2 + \text{H}_2$  SYSTEM

We train the MACE potential against energies and forces from Density Functional Theory (DFT) to improve its accuracy on the system. For this, we first construct a dataset of configurations by randomly perturbing points along the interpolated path connecting the initial and final states. The catalyst is not perturbed, and the interpolation is repeated for all permutations of like atoms (N and H only) in the initial state for a better coverage of the configuration space. Examples containing clashing atoms are filtered out. We then collect reference energies and per-atom forces for all examples using GPU-accelerated DFT calculations implemented in Quantum Espresso.<sup>56</sup> The self-consistent field (SCF) simulations use

a plane wave basis set with kinetic energy cutoff 600 eV. The Brillouin zone is sampled with a  $4 \times 4 \times 1$  Monkhorst–Pack grid, and the Perdew–Burke–Ernzerhof (PBE) exchange–correlation functional is adopted. The SCF procedure uses mixing parameter 0.7 and Davidson diagonalization and is run for a maximum of 300 iterations at an energy tolerance  $1e-6$  eV. All other parameters are set to their defaults. We source pseudopotentials from Materials Cloud.<sup>57,58</sup> The final dataset consists of 856 examples with converged energies and forces. We use these data points to train MACE over 1500 epochs with distance cutoff 6 Å, stochastic weight averaging<sup>59</sup> starting at 1200 epochs, and exponentially moving average decay 0.99. All other parameters are set to their defaults.

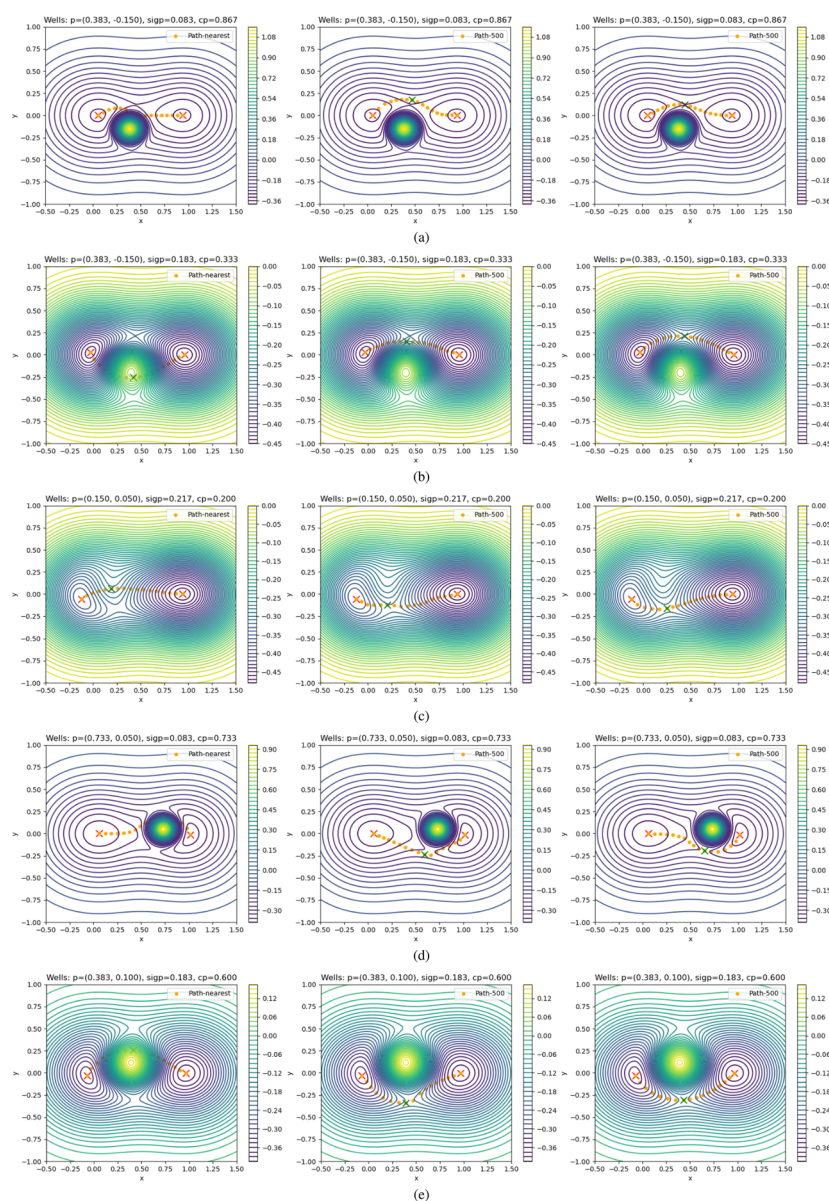


FIG. 18. Nearest neighbor (left), INR (middle), and true (right) paths for five unseen systems [(a)–(e)].

## APPENDIX G: PATH GENERALIZATION EXAMPLES

Figure 18 shows some example predicted paths on unseen systems along with the nearest neighbor and true paths.

## REFERENCES

- D. Wales, *Energy Landscapes: Applications to Clusters, Biomolecules and Glasses*, Cambridge Molecular Science (Cambridge University Press, 2004).
- F. Jensen, *Introduction to Computational Chemistry* (John Wiley & Sons, Inc., Hoboken, NJ, 2006).
- K. Fukui, "The path of chemical reactions—The IRC approach," *Acc. Chem. Res.* **14**, 363–368 (1981).
- H. Eyring, "The activated complex in chemical reactions," *J. Chem. Phys.* **3**, 107–115 (1935).
- M. G. Evans and M. Polanyi, "Some applications of the transition state method to the calculation of reaction velocities, especially in solution," *Trans. Faraday Soc.* **31**, 875–894 (1935).
- E. Maras, O. Trushin, A. Stukowski, T. Ala-Nissila, and H. Jónsson, "Global transition path search for dislocation formation in Ge on Si(001)," *Comput. Phys. Commun.* **205**, 13–21 (2016).
- J. K. Nørskov, T. Bligaard, J. Rossmeisl, and C. H. Christensen, "Towards the computational design of solid catalysts," *Nat. Chem.* **1**, 37–46 (2009).
- B. Hammer and J. K. Nørskov, "Electronic factors determining the reactivity of metal surfaces," *Surf. Sci.* **343**, 211–220 (1995).
- J. K. Nørskov, F. Abild-Pedersen, F. Studt, and T. Bligaard, "Density functional theory in surface chemistry and catalysis," *Proc. Natl. Acad. Sci. U. S. A.* **108**, 937–943 (2011).
- B. J. Rhodes, L. L. Schaaf, M. E. Zick, S. M. Pugh, J. S. Hilliard, S. Sharma, C. R. Wade, P. J. Milner, G. Csányi, and A. C. Forse, "<sup>17</sup>O NMR spectroscopy reveals CO<sub>2</sub> speciation and dynamics in hydroxide-based carbon capture materials," *ChemPhysChem* **26**(5), e202400941 (2025).
- A. Warshel and M. Levitt, "Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme," *J. Mol. Biol.* **103**, 227–249 (1976).
- H. Jónsson, G. Mills, and K. W. Jacobsen, "Nudged elastic band method for finding minimum energy paths of transitions," in *Classical and Quantum Dynamics in Condensed Phase Simulations*, edited by B. J. Berne, G. Ciccotti, and D. F. Coker (1998), pp. 385–404.
- G. Henkelman, B. P. Uberuaga, and H. Jónsson, "A climbing image nudged elastic band method for finding saddle points and minimum energy paths," *J. Chem. Phys.* **113**, 9901–9904 (2000).
- G. Henkelman and H. Jónsson, "Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points," *J. Chem. Phys.* **113**, 9978–9985 (2000).
- S. A. Trygubenko and D. J. Wales, "A doubly nudged elastic band method for finding transition states," *J. Chem. Phys.* **120**, 2082–2094 (2004).
- B. Peters, A. Heyden, A. T. Bell, and A. Chakraborty, "A growing string method for determining transition states: Comparison to the nudged elastic band and string methods," *J. Chem. Phys.* **120**, 7877–7886 (2004).
- S. Smidstrup, A. Pedersen, K. Stokbro, and H. Jónsson, "Improved initial guess for minimum energy path calculations," *J. Chem. Phys.* **140**, 214106 (2014).
- D. Sheppard and G. Henkelman, "Paths to which the nudged elastic band converges," *J. Comput. Chem.* **32**, 1769–1771 (2011).
- B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM* **65**, 99–106 (2021).
- V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Adv. Neural Inf. Process. Syst.* **33**, 7462–7473 (2020).
- J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2019), pp. 165–174.
- L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2019), pp. 4460–4470.
- A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.* **18**, 1–43 (2018).
- A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (Curran Associates, Inc., Red Hook, NY, 2019).
- W. E. W. Ren, and E. Vanden-Eijnden, "String method for the study of rare events," *Phys. Rev. B* **66**, 052301 (2002).
- W. E. W. Ren, and E. Vanden-Eijnden, "Simplified and improved string method for computing the minimum energy paths in barrier-crossing events," *J. Chem. Phys.* **126**, 164103 (2007).
- E. Vanden-Eijnden and M. Heymann, "The geometric minimum action method for computing minimum energy paths," *J. Chem. Phys.* **128**, 061103 (2008).
- R. Granot and R. Baer, "A spline for your saddle," *J. Chem. Phys.* **128**, 184111 (2008).
- A. C. Vaucher and M. Reiher, "Minimum energy paths and transition states by curve optimization," *J. Chem. Theory Comput.* **14**, 3091–3099 (2018).
- S.-i. Koda and S. Saito, "Locating transition states by variational reaction path optimization with an energy-derivative-free objective function," *J. Chem. Theory Comput.* **20**, 2798–2811 (2024).
- Y. Du, M. Plainer, R. Brekelmans, C. Duan, F. Noe, C. P. Gomes, A. Aspuru-Guzik, and K. Neklyudov, "Doob's Lagrangian: A sample-efficient variational approach to transition path sampling," in *Advances in Neural Information Processing Systems*, edited by A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Curran Associates, Inc., 2024), Vol. 37, pp. 65791–65822.
- S. Raja, M. Šipka, M. Psenka, T. Kreiman, M. Pavelka, and A. S. Krishnapriyan, "Action-minimization meets generative modeling: Efficient transition path sampling with the Onsager-Machlup functional," in *Proceedings of the 42nd International Conference on Machine Learning (ICML)* (PMLR, 2025).
- R. D. Amos and J. E. Rice, "Implementation of analytic derivative methods in quantum chemistry," *Comput. Phys. Rep.* **10**, 147–187 (1989).
- B. G. Johnson and M. J. Fisch, "An implementation of analytic second derivatives of the gradient-corrected density functional energy," *J. Chem. Phys.* **100**, 7429–7442 (1994).
- S. Mallikarjun Sharada, P. M. Zimmerman, A. T. Bell, and M. Head-Gordon, "Automated transition state searches without evaluating the Hessian," *J. Chem. Theory Comput.* **8**, 5166–5174 (2012).
- N. Gönheimer, K. Reuter, and J. T. Margraf, "Beyond numerical Hessians: Higher-order derivatives for machine learning interatomic potentials via automatic differentiation," *J. Chem. Theory Comput.* **21**, 4742–4752 (2025).
- See <https://github.com/Kalyan0821/INR-Path> for the Python code of our method.
- A. Hjorth Larsen, J. Jørgen Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Bjerre Jensen, J. Kermode, J. R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, "The atomic simulation environment—A Python library for working with atoms," *J. Phys.: Condens. Matter* **29**, 273002 (2017).
- E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch, "Structural relaxation made simple," *Phys. Rev. Lett.* **97**, 170201 (2006).
- D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization* (ICLR (Poster), 2015).
- D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.* **45**, 503–528 (1989).
- I. Batatia, D. P. Kovacs, G. Simm, C. Ortner, and G. Csányi, "MACE: Higher order equivariant message passing neural networks for fast and accurate force

- fields,” in *Advances in Neural Information Processing Systems*, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Curran Associates, Inc., 2022), Vol. 35, pp. 11423–11436.
- <sup>43</sup>E. D. Hermes, K. Sargsyan, H. N. Najm, and J. Zádor, “Sella, an open-source automation-friendly molecular saddle point optimizer,” *J. Chem. Theory Comput.* **18**, 6974–6988 (2022).
- <sup>44</sup>P. J. Kuntz, E. M. Nemeth, J. C. Polanyi, S. D. Rosner, and C. E. Young, “Energy distribution among products of exothermic reactions. II. Repulsive, mixed, and attractive energy release,” *J. Chem. Phys.* **44**, 1168–1184 (1966).
- <sup>45</sup>K. Müller and L. D. Brown, “Location of saddle points and minimum energy paths by a constrained simplex optimization procedure,” *Theor. Chim. Acta* **53**, 75–93 (1979).
- <sup>46</sup>S. Dang, B. Qin, Y. Yang, H. Wang, J. Cai, Y. Han, S. Li, P. Gao, and Y. Sun, “Rationally designed indium oxide catalysts for CO<sub>2</sub> hydrogenation to methanol with high activity and selectivity,” *Sci. Adv.* **6**, eaaz2060 (2020).
- <sup>47</sup>L. L. Schaaf, E. Fako, S. De, A. Schäfer, and G. Csányi, “Accurate energy barriers for catalytic reaction pathways: An automatic training protocol for machine learning force fields,” *npj Comput. Mater.* **9**, 180 (2023).
- <sup>48</sup>I. Batatia, P. Benner, Y. Chiang, A. M. Elena, D. P. Kovács, J. Riebesell, X. R. Advincula, M. Asta, W. J. Baldwin, N. Bernstein, A. Bhowmik, S. M. Blau, V. Cárare, J. P. Darby, S. De, F. D. Pia, V. L. Deringer, R. Elijošius, Z. El-Machachi, E. Fako, A. C. Ferrari, A. Genreith-Schriever, J. George, R. E. A. Goodall, C. P. Grey, S. Han, W. Handley, H. H. Heenen, K. Hermansson, C. Holm, J. Jaafar, S. Hofmann, K. S. Jakob, H. Jung, V. Kapil, A. D. Kaplan, N. Karimitari, N. Kroupa, J. Kullgren, M. C. Kuner, D. Kuryla, G. Liepuoniute, J. T. Margraf, I.-B. Magdáu, A. Michaelides, J. H. Moore, A. A. Naik, S. P. Niblett, S. W. Norwood, N. O’Neill, C. Ortner, K. A. Persson, K. Reuter, A. S. Rosen, L. L. Schaaf, C. Schran, E. Sivonxay, T. K. Stenczel, V. Svahn, C. Sutton, C. van der Oord, E. Varga-Umbrich, T. Vegge, M. Vondrák, Y. Wang, W. C. Witt, F. Zills, and G. Csányi, “A foundation model for atomistic materials chemistry,” [arXiv:2401.00096](https://arxiv.org/abs/2401.00096) [physics.chem-ph] (2023).
- <sup>49</sup>D. P. Kovács, J. H. Moore, N. J. Browning, I. Batatia, J. T. Horton, Y. Pu, V. Kapil, W. C. Witt, I.-B. Magdáu, D. J. Cole, and G. Csányi, “MACE-OFF: Short-range transferable machine learning force fields for organic molecules,” *J. Am. Chem. Soc.* **147**(21), 17598–17611 (2025).
- <sup>50</sup>Á. Logadóttir and J. K. Nørskov, “Ammonia synthesis over a Ru(0001) surface studied by density functional calculations,” *J. Catal.* **220**, 273–279 (2003).
- <sup>51</sup>K. Honkala, A. Hellman, I. N. Remediakis, A. Logadóttir, A. Carlsson, S. Dahl, C. H. Christensen, and J. K. Nørskov, “Ammonia synthesis from first-principles calculations,” *Science* **307**, 555–558 (2005).
- <sup>52</sup>J. Baker, A. Kessi, and B. Delley, “The generation and use of delocalized internal coordinates in geometry optimization,” *J. Chem. Phys.* **105**, 192–212 (1996).
- <sup>53</sup>S. Makri, C. Ortner, and J. R. Kermode, “A preconditioning scheme for minimum energy path finding methods,” *J. Chem. Phys.* **150**, 094109 (2019).
- <sup>54</sup>M. Schreiner, A. Bhowmik, T. Vegge, J. Busk, and O. Winther, “TransitionIx—A dataset for building generalizable reactive machine learning potentials,” *Sci. Data* **9**, 779 (2022).
- <sup>55</sup>V. G. Satorras, E. Hoogeboom, and M. Welling, “E(n) equivariant graph neural networks,” in *International Conference on Machine Learning* (PMLR, 2021), pp. 9323–9332.
- <sup>56</sup>P. Giannozzi, O. Baseggio, P. Bonfà, D. Brunato, R. Car, I. Carnimeo, C. Cavazzoni, S. de Gironcoli, P. Delugas, F. Ferrari Ruffino, A. Ferretti, N. Marzari, I. Timrov, A. Urru, and S. Baroni, “Quantum ESPRESSO toward the exascale,” *J. Chem. Phys.* **152**, 154105 (2020).
- <sup>57</sup>See <https://www.materialscloud.org/discover/sssp/table/efficiency> for pseudopotentials.
- <sup>58</sup>L. Talirz, S. Kumbhar, E. Passaro, A. V. Yakutovich, V. Granata, F. Gargiulo, M. Borelli, M. Uhrin, S. P. Huber, S. Zoupanos *et al.*, “Materials cloud, a platform for open computational science,” *Sci. Data* **7**, 299 (2020).
- <sup>59</sup>P. Izmailov, D. Podoprikhin, T. Garipov, D. P. Vetrov, and A. G. Wilson, “Averaging weights leads to wider optima and better generalization,” in *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6–10, 2018*, edited by A. Globerson and R. Silva (AUAI Press, 2018), pp. 876–885, <http://auai.org/uai2018/proceedings/papers/313.pdf>.