



# Predicting Chemical Reaction Barriers with a Machine Learning Model

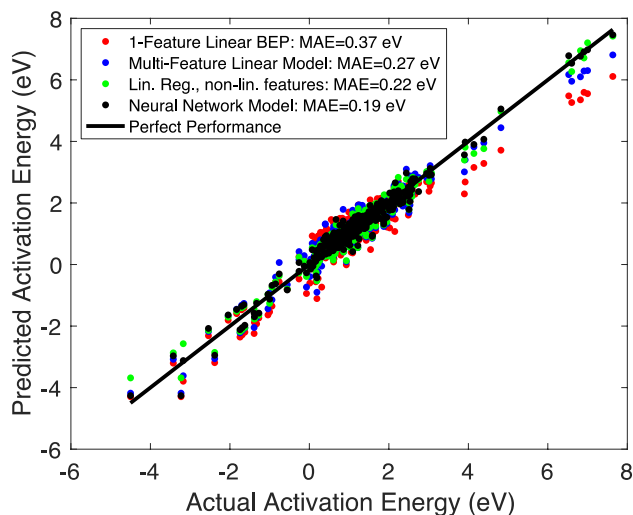
Aayush R. Singh<sup>1,2</sup> · Brian A. Rohr<sup>1,2</sup> · Joseph A. Gauthier<sup>1,2</sup> · Jens K. Nørskov<sup>1,2,3</sup>

Received: 24 November 2018 / Accepted: 4 February 2019 / Published online: 16 March 2019  
© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

In the past few decades, tremendous advances have been made in the understanding of catalysis at solid surfaces. Despite this, most discoveries of materials for improved catalytic performance are made by a slow trial and error process in an experimental laboratory. Computational simulations have begun to provide a way to rationally design materials for optimizing catalytic performance, but due to the high computational expense of calculating transition state energies, simulations cannot adequately screen the phase space of materials. In this work, we attempt to mitigate this expense by using a machine learning approach to predict the most expensive and most important parameter in a catalyst's affinity for a reaction: the reaction barrier. Previous methods which used the step reaction energy as the only parameter in a linear regression had a mean absolute error (MAE) on the order of 0.4 eV, too high to be used predictively. In our work, we achieve a MAE of about 0.22 eV, a marked improvement towards the goal of computational prediction of catalytic activity.

## Graphical Abstract



**Keywords** Density functional theory · Machine learning · Kinetic modeling

Aayush R. Singh, Brian A. Rohr and Joseph A. Gauthier have contributed equally to this work.

**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s10562-019-02705-x>) contains supplementary material, which is available to authorized users.

✉ Jens K. Nørskov  
norskov@stanford.edu

Extended author information available on the last page of the article

## 1 Introduction

Most industrial chemical processes reduce energy requirements by using catalysts to control reaction rate and selectivity. Many research efforts are directed at accelerating catalyst discovery. In the past two to three decades, computational efforts have shifted the efforts from a trial-and-error approach to one of rational design. Kinetic models based

on calculated reaction energies and barriers for elementary reaction steps allow for the identification of promising candidate materials. Reaction energies can be relatively cheaply calculated with methods such as density functional theory (DFT). Transition state energies, which are critical for determining the kinetic performance of a catalyst, are far more expensive to calculate. In practice they are estimated with a variety of saddle point search methods [1–3] in a high dimensional reaction coordinate, in combination with DFT. For a given catalytic process, many transition state energies for elementary reaction steps must be calculated, since for all but the simplest processes, there is a wide variety of chemistry that can occur on the surface.

For a given overall reaction, to rigorously determine the performance of a catalyst, microkinetic models are created by using DFT to calculate the reaction and transition state energies for each of the many elementary steps that make up the overall reaction, as well as those for competing reactions. Hence for a given catalyst, optimization can be intractably high dimensional, limiting even computational screening. To simplify the creation of the microkinetic model, linear scaling relations between the reaction energies and barriers of the different elementary steps can be created. Such relationships allow us to approximate all of the elementary reaction barriers after only explicitly calculating a few [4–6].

The use of the reaction energies as a descriptor for transition state energy has a meaningful basis in physics, as both the reaction energy and transition state energy for simple reactions correlate with the d-band center of a metal catalyst [7]. However, the reaction energy is not the only physically meaningful feature impacting the transition state energy. We expect that other easily determined features such as the geometry of the catalyst surface and the identity of the adsorbates involved in each reaction step could allow us to better describe the transition state energy without the need for more costly explicit DFT calculations. The question we set out to investigate in this paper is whether machine learning methods can be used to increase the accuracy of predictions of transition state energies for surface chemical reactions.

There is growing evidence that machine learning can be a useful tool in computational catalysis [8–11]. Researchers have previously used learning algorithms to reduce the number of DFT calculations needed for the construction of a surface phase diagram,[12] to predict the surface reactivity of metal alloys for carbon dioxide electro-reduction,[13] and to predict molecular atomization energies [14]. More recently, machine learning methods have been developed and implemented to augment and accelerate the calculation of energies and forces by DFT [13, 15–18].

In this work, we focus on reducing the error of transition state energy predictions for a range of chemical reactions. We use the data set generated in Ref. 5 where the plane-wave

DFT code DACAPO was used with a kinetic energy cutoff of 340 eV to describe the valence electrons, while the core electrons were described with Vanderbilt ultrasoft pseudopotentials [19]. We examine and compare several predictive methods, including linear and nonlinear regression, random forest, gaussian process, gradient boosted random forest, and neural networks of varying size. We employ multiple physically relevant and cheap to determine features, including the coordination number of the metal atom, the identity of the adsorbate, the number of bonds broken, the binding energy of the adsorbate, and polynomial combinations. Such a machine learning approach could significantly reduce the computational cost of screening for catalytic materials, allowing for a greatly expedited search of the vast phase space. The conclusion is two-fold. First, we find that the most important descriptor of transition state energies is indeed the surface bond energies of the atoms that interact with the surface in the transition state. Second, we find that the accuracy can be improved - typically the mean absolute error of predictions relative to the full DFT calculations can be reduced from 0.4 eV to 0.25 eV by adding up to 7 additional descriptors. Finally, we discuss the results in light of the inherent inaccuracies of the computational methods employed.

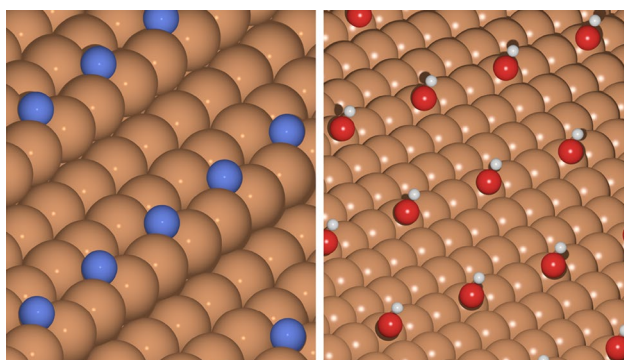
## 2 Methods

### 2.1 Dataset and Features

In order to develop a model to predict transition state energies, we must first have training examples where the transition state energy has already been calculated by a saddle point search method. Using a database from a previous work,[5] we have selected 315 examples of calculated transition state energies for dissociation reactions of an assortment of molecules on a variety of surfaces. Our data set consists of 236 dehydrogenation examples, 38 N<sub>2</sub> dissociation examples, and 41 O<sub>2</sub> dissociation examples. The data set used in this work is available digitally as supplementary material.

Beyond the traditionally used feature, reaction energy, we considered three new features in our model: (1) coordination of the surface, (2) the number of bonds broken between the initial and final state, and (3) the identity of the surface atom involved in bond breaking. Figure 1 illustrates two training examples with different values for each of the included features.

The catalyst geometry was treated as a binary variable, where the variable takes on a value of 1 for an under-coordinated step site, and 0 for a close-packed terrace. The identity of the surface atom involved in bond breaking was treated as a multinomial, where the variable was assigned 0 for hydrogen, 1 for carbon, 2 for oxygen, and 3 for nitrogen. Similarly,



**Fig. 1** (left) Illustration of an under-coordinated stepped surface with nitrogen atoms adsorbed to the surface, after undergoing a dissociation reaction in which three bonds were broken. (right) Illustration of a close-packed terrace surface with OH adsorbed

the number of bonds broken was also multinomial, with 1 assigned for dehydrogenation, 2 for O<sub>2</sub> dissociation, 3 for N<sub>2</sub> dissociation. We note that the numerical assignments given here are arbitrary. All of these features can be obtained from the atomic coordinates files, without the need for further DFT calculations.

## 2.2 Machine Learning Methods

First, we reserved 20% of our data at random as a test set to be used solely for evaluating the performance of our models. The remaining 80% of the data was our working data set used to train our models. At many points in our analysis, we divided the working set randomly into a training set (70% of the working set) and a validation set (30% of the working set). Overall, this led to an approximately 50–30–20 (training-validation-test) split of our data. In this work, the training error of a model refers to the error on the actual data points used to train the model. The validation error of a model refers to the model's error on the validation set-data that was in the working set but not used for the training of that particular model. Test error of a model refers to the model's error on the test set, which was never used at any point in the analysis leading to the generation of the model.

The forward search algorithm is a hold-out cross-validation “wrapper” method designed to select the best set of features for a particular model. It begins with all features in a set called the “out set.” The model is trained with each feature, one at a time. The feature that gives the lowest validation error is selected and moved from the out set to the model. The process is repeated *N* times, where *N* is the number of features. In each step, each remaining feature in the out set is added, and the one that gives the lowest validation error is selected and added to the model. At the end, the set of features that gives the lowest validation error is selected as the feature set for the model.

Inspired by the success of previous works using single feature linear regression, we first used the simple linear regression model with multiple features in an attempt to capture more of the information in the data set. The linear regression model is shown below in Eq. 1. Here *y* is the output variable (in this case transition state energy), *x<sub>i</sub>* is the value of feature *i*, and β<sub>*i*</sub> is the coefficient mapping *x<sub>i</sub>* to *y*, trained by linear least squares.

$$y = \sum_i x_i \beta_i \quad (1)$$

We do not expect the transition state energy to vary linearly with all of the features. For example, with all other values of features fixed, we expect the transition state energy of a species containing carbon, nitrogen, or oxygen to change non-linearly as the adsorbate is changed.

For this reason, we included non-linear (polynomial) terms with all second order combinations of the four features implemented in linear regression. Non-linear features were selected with the forward search method. The model for the linear regression with non-linear terms is identical to linear regression shown in Eq. 1; the key difference is that the features list contains non-linear terms. We searched a broader set of possible non-linear transformations of the four features using the SISO [20] package and found results that were similar to the other models reported in this study, but with features that are less interpretable.

Using the python package Scikit-learn,[21] we explore the effectiveness of the random forest method (both standard and gradient boosted) and gaussian process. In an effort to capture more complicated relationships between the inputs and the outputs, we fit the training data to a feed-forward neural network using Matlab. The network used a sigmoid activation function, and it was trained using the Levenberg–Marquardt back-propagation technique. This training technique uses the mean square error (MSE) as the loss function. Assuming a gaussian error distribution, this is equivalent to maximizing the likelihood of observing the data given the model parameters. We report the MAE as training error (not the loss function) because it is more readily interpreted.

## 3 Results and Discussion

### 3.1 Feature Selection

The results of the forward search algorithm (described in the [methods](#) section) for linear regression are found in Table 1.

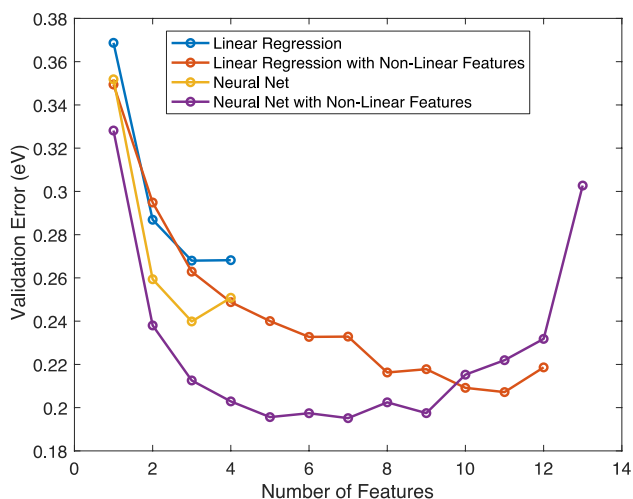
As anticipated, when considered alone, the single most important feature was the reaction energy. However, the fact that the validation error decreases as additional features are included (with each successive row of Table 1) indicates that

**Table 1** Forward search feature selection for linear regression

Feature	Training error (eV)	Validation error (eV)
Reaction energy	0.36	0.37
Identity of adsorbate	0.30	0.29
# Bonds broken	0.26	0.27
Surface geometry	0.26	0.27

the features added to the analysis are in fact physically meaningful and improve the predictive power of the model. After reaction energy, the model is most improved by including information regarding the identity of the adsorbate, which lowers the validation error to about 0.3 eV. Including the other two features, number of bonds broken and the surface geometry, results in further marginal improvements on the validation error. In the remainder of our analysis, when training linear models, we used all four features as it gave the lowest validation error.

The full output of forward search for linear regression including non-linear (polynomial) features is summarized in Fig. 2, where the errors reported are the average of 25 iterations of the forward-search algorithm. Here, once again, the most important feature was the reaction energy, as expected. Following the reaction energy, polynomial combinations of the four original features were chosen by the forward search algorithm within the first four iterations. This again suggests that the features we added each contain unique and physically relevant information, since it results in a lower error. In the rest of our analysis, when training models with non-linear features, we used the eight features selected by forward search that gave the lowest validation error.

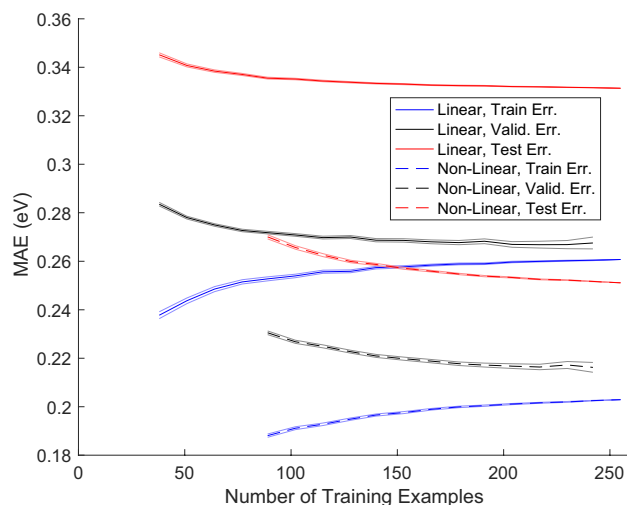
**Fig. 2** Plot of the output of the forward search algorithm for each of the models

We repeated the forward search procedure for the neural network using just the original four features as well as the polynomial terms. As seen with linear regression, the addition of each of the four features improved the performance of the neural network model, with a small increase in error with the inclusion of the last feature. This suggests that adding a fifth unique feature would be beneficial. The inclusion of non-linear features further reduces validation error, until more than 7 features are included. At this point the model is likely being overfit, which would cause an increase in validation error as seen.

### 3.2 Bias-Variance Analysis

The forward search output (Fig. 2) also provides some insight into the bias-variance balance. The addition of each feature continued to reduce the validation error of the linear model. The fact that the validation error did not begin to rise means that our linear model is still underfit even with all four features included. It would therefore be beneficial to find more physically meaningful (and cheap to determine) features and add them to this model.

The learning curve, Fig. 3, can give insight into the bias/variance balance of the model. Here the shaded region surrounding each line corresponds to a 95% confidence interval, constructed by repeating the hold-out cross-validation process 1000 times. The learning curve shows that the linear model converges very quickly, in fewer than 50 training examples. This indicates that we achieve very little added performance out of training examples 50 through 250, and our linear model is significantly underfit (high bias). This further justifies the inclusion of non-linear features in the model.

**Fig. 3** Learning curves for the two linear models tested

In the forward search algorithm (Fig. 2), applied to the linear regression with non-linear terms, the validation error begins to rise as the last few features are added. This indicates that adding higher order terms (cubic and above) may not improve the performance of the model. However, it is possible that cubic terms in some features would be beneficial even though square terms in other features were shown to be deleterious. The learning curve shows that the model with non-linear features converges after about 100 training examples. This is slower than the linear model converged, but it still indicates that we are not leveraging all of our data. The model including polynomial features is therefore likely still underfit, although it is less underfit than the linear model. The addition of some cubic terms may improve this model, which would be found by the forward search algorithm.

A hyperparameter search was performed to determine the best number of neurons per layer and number of layers for the problem. This search is summarized seen in Fig. 4. Networks with one through twenty neurons per layer and one through five layers were trained on the training set, and their performance was evaluated on the validation set. The best performing network had one hidden layer consisting of twelve neurons. While there is some stochasticity in the performance of these networks, the heat map shows that the simplest models, shown in the leftmost column of the heat map (between 3 and 13 nodes), consistently performed poorly on the validation set. Similarly, the most complicated models, shown in the bottom-right region of the heatmap, consistently performed poorly on the validation set. The best performing models had intermediate complexity: one or two hidden layers with five to fifteen neurons in each layer. These results are consistent with the bias-variance trade-off. The heat map indicates that one neuron is too simple of a model (underfit), but the nature of our regression problem and size of our data set does not merit using a complex network with many hidden layers and many neurons per layer, as these networks would likely be overfit.

With the neural network, the validation error begins to rise as soon as the non-linear features are added even though the training error is low. This suggests that the neural net trained on non-linear features is overfit. However, when using the same neural network and training only on the linear features, the validation error does not increase for the four features available. This is summarized in Table 2.

### 3.3 Model Performance

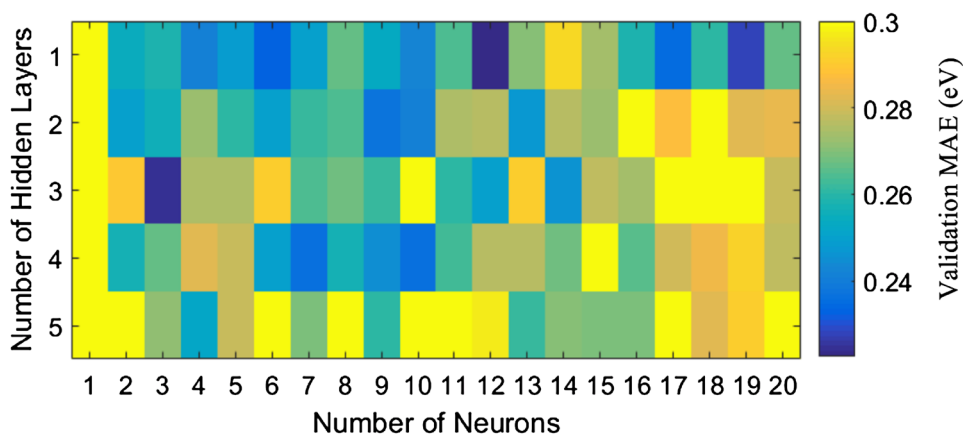
Linear transition state scaling models based on just one feature (reaction energy) typically have a training MAE of roughly 0.4 eV, with slight improvements for simpler, less general data sets. We were able to reproduce this error by performing one-feature linear regression on our data set; the test error for the traditional BEP relation was 0.40 eV. Moving to the multi-feature linear regression, we were able to decrease the test error to 0.33 eV, adding the best non-linear features decreased the test error to 0.25 eV. Fitting to the entire training set (i.e. without cross-validation), we find test errors for the random forest (both standard and gradient-boosted) and the gaussian process to be comparable to the linear regression with polynomial features. Moving to the optimal neural network decreased the test error slightly, to 0.22 eV. The results are summarized in Table 3.

Figure 5 illustrates the model performances in a parity plot, where it is clear that the models trained in this study out-perform the traditional single feature BEP relation. By

**Table 2** Training, validation, and test error for the neural network configurations

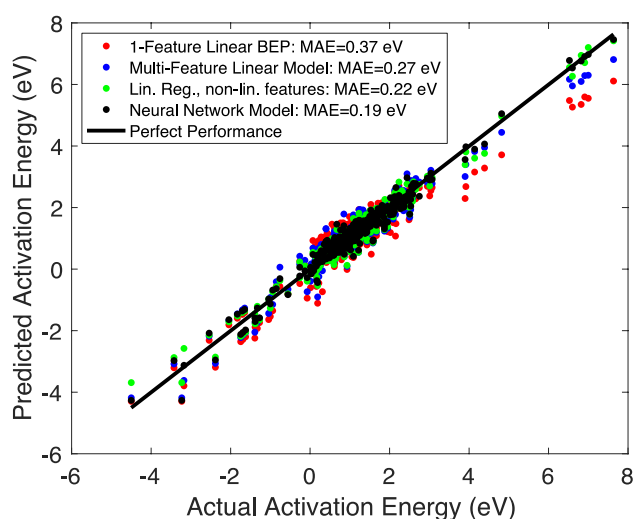
	Training error (eV)	Validation error (eV)	Test error (eV)
Under-fit NN (1,1)	0.30	0.30	0.40
Well-fit NN (1,12)	0.17	0.23	0.22
Over-fit NN (1,12)	0.16	0.20	1.83

**Fig. 4** Heat map used to optimize the size of the neural network



**Table 3** Training, validation, and test error for the models tested

	Training error (eV)	Validation Error (eV)	Test error (eV)
One feature linear BEP	0.36	0.37	0.40
Multi-feature linear regression	0.26	0.27	0.33
Lin. reg. with non-linear features	0.20	0.22	0.25
Random forest	0.11	--	0.31
Gaussian process	0.17	--	0.26
Gradient boosted random forest	0.18	--	0.29
Neural network	0.17	0.23	0.22

**Fig. 5** Plot of predicted transition state energy versus actual transition state energy for the original BEP scaling relation, the multi-feature linear model, and the neural network. The entire dataset is shown

using the neural network to improve the MAE from 0.40 to 0.22 eV, we can improve the accuracy of our predicted reaction rates by 2–3 orders of magnitude, since reaction rates depend exponentially on the reaction barrier (Eq. 2). Here the reaction barrier is given by  $\Delta G_a$ .

$$r = \frac{k_b T}{h} \exp \left[ -\frac{\Delta G_a}{k_b T} \right] \quad (2)$$

It is interesting to note that while the performance can be improved further with the use of a neural network compared to the linear regression with polynomial features, from a test MAE of 0.25 eV to 0.22 eV. This difference results in only a much smaller increase in confidence, less than one order of magnitude. Using a neural network also results in a vast increase in parameters required to train,

and with this comes an increased computational cost and a loss of understanding of the model.

## 4 Conclusions and Future Work

The work shown here illustrates a first step towards improving the existing single feature linear relationships used to predict transition state energies in complex chemical reactions. By predicting transition state energies from the simpler-to-determine reaction energy and other cheaply determined parameters, we can reduce the computational cost associated with screening a material's catalytic activity by several orders of magnitude. We show that the MAE can be reduced from 0.40 eV in the single feature linear regression (BEP) to 0.25 eV with linear regression including polynomial features, or 0.22 eV with a neural network. We hence improve the accuracy of our chemical rate calculations by 2–3 orders of magnitude at ambient temperatures, since chemical rates are proportional to the exponential of the activation energy. This represents a significant step towards the rapid computational screening of materials as a way to guide experiments. The use of the linear regression model with polynomial features may be preferred since it performs nearly as well as the neural network, while using far fewer parameters and hence allowing for an increased understanding of the model.

To further improve the models shown here, additional features could be added. This is important because we see that our test errors are not significantly higher than our training errors, indicating that we may still have high bias (underfitting) in our models. New features may include properties such as the adsorbate coordination number, charge delocalization across the system, and change in entropy across the reaction coordinate. The adsorbate coordination number in particular has been previously shown to influence both the reaction and transition state energy. The addition of new features would be especially important for the linear regression models, since these models were likely more underfit than the neural network models used in this work.

Additional data beyond just simple dissociation reactions could be collected to further train and test the model. These dissociation reactions are relatively straight-forward to calculate transition state energies for, which makes them ideal for generating test sets. But, the real power of a predictive model would be in assisting the calculation of harder-to-determine transition state energies.

Finally, the neural network model could be explored in greater depth. In this work, we used mostly the default parameters for a feed-forward neural network. It is possible that a feed-back neural network would have better performance for our system. If we were able to collect more data

on a wider variety of reactions, a more complex neural network may provide the most predictive power.

The failure of the neural network to significantly improve upon the polynomial linear regression, despite a large increase in the number of trainable parameters, indicates that there is likely significant unphysical uncertainty in the data that will not be captured by such a model. Future work will attempt to address this uncertainty by utilizing a training data set using higher order (non-GGA DFT) methods.

**Acknowledgements** Support from the U.S. Department of Energy, Office of Basic Energy Science, Chemical Sciences, Geosciences, and Biosciences Division, to the SUNCAT Center for Interface Science and Catalysis is gratefully acknowledged. B.A.R. acknowledges fellowship support from the National Science Foundation Graduate Research Fellowship (Grant No. DGE-114747). This work was supported by a research grant (9455) from VILLUM FONDEN.

## Compliance with Ethical Standards

**Conflict of interest** The authors have no conflict of interest to declare.

## References

1. Henkelman G, Jónsson H (1999) A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives. *J Chem Phys* 111:7010–7022. <https://doi.org/10.1063/1.480097>
2. Henkelman G, Uberuaga BP, Jónsson H (2000) Climbing image nudged elastic band method for finding saddle points and minimum energy paths. *J Chem Phys* 113:9901–9904. <https://doi.org/10.1063/1.1329672>
3. Henkelman G, Jónsson H (2000) Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points. *J Chem Phys* 113:9978–9985. <https://doi.org/10.1063/1.1323224>
4. Samuelson P, Thorp EO, Kassouf ST (1968) Beat the Market: A Scientific Stock Market System. *J Am Stat Assoc* 63:1049. <https://doi.org/10.1039/TF9363201333>
5. Wang S, Petzold V, Tripkovic V et al (2011) Universal transition state scaling relations for (de)hydrogenation over transition metals. *Phys Chem Chem Phys* 13:20760. <https://doi.org/10.1039/c1cp20547a>
6. Hammer B, Nørskov JK (2000) Theoretical surface science and catalysis—calculations and concepts. *Adv Catal* 45:71–129. [https://doi.org/10.1016/S0360-0564\(02\)45013-4](https://doi.org/10.1016/S0360-0564(02)45013-4)
7. Andersson MP, Bligaard T, Kustov A et al (2006) Toward computational screening in heterogeneous catalysis: Pareto-optimal methanation catalysts. *J Catal* 239:501–506. <https://doi.org/10.1016/j.jcat.2006.02.016>
8. Peterson AA, Christensen R, Khorshidi A (2017) Addressing uncertainty in atomistic machine learning. *Phys Chem Chem Phys* 19:10978–10985. <https://doi.org/10.1039/c7cp00375g>
9. Khorshidi A, Peterson AA (2016) Amp: A modular approach to machine learning in atomistic simulations. *Comput Phys Commun* 207:310–324. <https://doi.org/10.1016/j.cpc.2016.05.010>
10. Jørgensen PB, Jacobsen KW, Schmidt MN (2018) Neural message passing with edge updates for predicting properties of molecules and materials
11. Jørgensen PB, Mesta M, Shil S et al (2018) Machine learning-based screening of complex molecules for polymer solar cells. *J Chem Phys* doi. <https://doi.org/10.1063/1.5023563>
12. Ulissi ZW, Singh AR, Tsai C, Nørskov JK (2016) Automated discovery and construction of surface phase diagrams using machine learning. *J Phys Chem Lett* 7:3931–3935. <https://doi.org/10.1021/acs.jpcclett.6b01254>
13. Brockherde F, Vogt L, Li L et al (2016) By-passing the Kohn–Sham equations with machine learning. <https://doi.org/10.1038/s41467-017-00839-3>
14. Hansen K, Montavon G, Biegler F et al (2013) Assessment and validation of machine learning methods for predicting molecular atomization energies. *J Chem Theory Comput* 9:3404–3419. <https://doi.org/10.1021/ct400195d>
15. Schneider E, Dai L, Topper RQ et al (2017) Stochastic neural network approach for learning high-dimensional free energy surfaces. *Phys Rev Lett* 119:150601. <https://doi.org/10.1103/PhysRevLett.119.150601>
16. Ulissi ZW, Tang MT, Xiao J et al (2017) Machine-learning methods enable exhaustive searches for active bimetallic facets and reveal active site motifs for CO<sub>2</sub> reduction. *ACS Catal* 7: 6600–6608. <https://doi.org/10.1021/acscatal.7b01648>
17. Ulissi ZW, Medford AJ, Bligaard T et al (2017) To address surface reaction network complexity using scaling relations machine learning and DFT calculations. *Nat Commun* 8:14621. <https://doi.org/10.1038/ncomms14621>
18. Ma X, Li Z, Achenie LEK, Xin H (2015) Machine-learning-augmented chemisorption model for CO<sub>2</sub> electroreduction catalyst screening. *J Phys Chem Lett* 6:3528–3533. <https://doi.org/10.1021/acs.jpcclett.5b01660>
19. Vanderbilt D (1990) Soft self-consistent pseudopotentials in a generalized eigenvalue formalism. *Phys Rev B* 41:7892–7895. <https://doi.org/10.1103/PhysRevB.41.7892>
20. Ouyang R, Curtarolo S, Ahmetcik E et al (2018) SISO: A compressed-sensing method for identifying the best low-dimensional descriptor in an immensity of offered candidates. *Phys Rev Mater* doi. <https://doi.org/10.1103/PhysRevMaterials.2.083802>
21. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12:2825–2830

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

Aayush R. Singh<sup>1,2</sup> · Brian A. Rohr<sup>1,2</sup> · Joseph A. Gauthier<sup>1,2</sup> · Jens K. Nørskov<sup>1,2,3</sup>

<sup>1</sup> Department of Chemical Engineering, SUNCAT Center for Interface Science and Catalysis, Stanford University, Stanford, CA 94305, USA

<sup>2</sup> SLAC National Accelerator Laboratory, SUNCAT Center for Interface Science and Catalysis, 2575 Sand Hill Road, Menlo Park, CA 94025, USA

<sup>3</sup> Department of Physics, Denmark Technical University, 2800 Kgs. Lyngby, Denmark