Acceleration of saddle-point searches with machine learning

Andrew A. Peterson

Citation: The Journal of Chemical Physics **145**, 074106 (2016); doi: 10.1063/1.4960708

View online: http://dx.doi.org/10.1063/1.4960708

View Table of Contents: http://aip.scitation.org/toc/jcp/145/7

Published by the American Institute of Physics

Acceleration of saddle-point searches with machine learning

Andrew A. Peterson[a)]
School of Engineering, Brown University, Providence, Rhode Island 02912, USA

(Received 3 March 2016; accepted 27 July 2016; published online 16 August 2016)

In atomistic simulations, the location of the saddle point on the potential-energy surface (PES) gives important information on transitions between local minima, for example, via transition-state theory. However, the search for saddle points often involves hundreds or thousands of *ab initio* force calls, which are typically all done at full accuracy. This results in the vast majority of the computational effort being spent calculating the electronic structure of states not important to the researcher, and very little time performing the calculation of the saddle point state itself. In this work, we describe how machine learning (ML) can reduce the number of intermediate *ab initio* calculations needed to locate saddle points. Since machine-learning models can learn from, and thus mimic, atomistic simulations, the saddle-point search can be conducted rapidly in the machine-learning representation. The saddle-point prediction can then be verified by an *ab initio* calculation; if it is incorrect, this strategically has identified regions of the PES where the machine-learning representation has insufficient training data. When these training data are used to improve the machine-learning model, the estimates greatly improve. This approach can be systematized, and in two simple example problems we demonstrate a dramatic reduction in the number of *ab initio* force calls. We expect that this approach and future refinements will greatly accelerate searches for saddle points, as well as other searches on the potential energy surface, as machine-learning methods see greater adoption by the atomistics community. *Published by AIP Publishing.* [http://dx.doi.org/10.1063/1.4960708]

The potential-energy surface (PES) is the central abstraction of atomistic simulations. Regions surrounding local minima correspond to stable molecules or phases, while first-order saddle points provide information on transitions between these stable states. Equations based on transition-state theory can quantify a kinetic rate estimate for a chemical reaction;[1,2] these methods typically require knowledge of the saddle point location, which is then used to calculate the free energy change from the initial state to the saddle point. Thus, the location of saddle points—as well as the minimum-energy pathways (MEPs) that connect local minima via the saddle points—is one of the common, crucial tasks in atomistic simulations.

Numerous methods have been developed to systematically search for first-order saddle points. Of great popularity is the "nudged elastic band" (NEB) method,[3–5] which searches for the minimum energy pathway, the apex of which is the saddle point. The NEB constructs a virtual "band" of images between the initial and final states; forces in the directions between images are replaced with elastic (Hookean) forms that serve to space the images equally along the band. The band is subjected to a geometry optimization; as a result, the band is "nudged" towards the minimum energy path. This systematic approach and its variants[6,7] are well-suited to locating transition states, particularly in periodic density functional theory (DFT) codes.

Despite the robustness of this method, it is computationally expensive to run such calculations. The algorithm is iterative as it nudges the band towards the MEP; each of these iterations requires a set of *ab initio* calculations in order to calculate forces along the band. The result of the algorithm is either a band of a few images (the MEP) or a single image (the saddle point), which can be verified to be such with a single *ab initio* calculation per image. We assert that this final verification calculation is the *only* important *ab initio* calculation in the routine; that is, all other *ab initio* calculations only serve to find a path to this final result. Further, because the iterations must be performed sequentially, the routine does not parallelize well beyond the embarrassingly parallel division of computation among images. Therefore, if an algorithm can be developed which accelerates the intermediate steps of the routine but can still guarantee the final MEP / saddle point, it may drastically accelerate this search.

The properties of machine-learning (ML) algorithms may be well-suited to the solution of this problem. First, quantum mechanics tells us that the potential energy is a smooth, unique, and unknowable function of the atomic coordinates, $\vec{x}$; that is, $E(\vec{x})$ (under certain assumptions, such as in constant external fields). Methods such as DFT give a perfectly precise, or noiseless, approximation to this function. Many machine learning algorithms, given sufficient training data, have the flexibility to perfectly replicate any smooth functional form, as the model's parameter space (unboundedly) increases.[8] Thus, a machine-learning model should be capable of mimicking the potential energy of the system—or the derivative quantity, the forces on each atom—to arbitrary accuracy. Numerous demonstrations of machine-learning mimicry of the PES have been published in recent years,[9–13] and we have recently released an open-source software package, known as *Amp*,[14,15] that provides a toolbox of such techniques that integrates with

[a)]Electronic mail: andrew_peterson@brown.edu

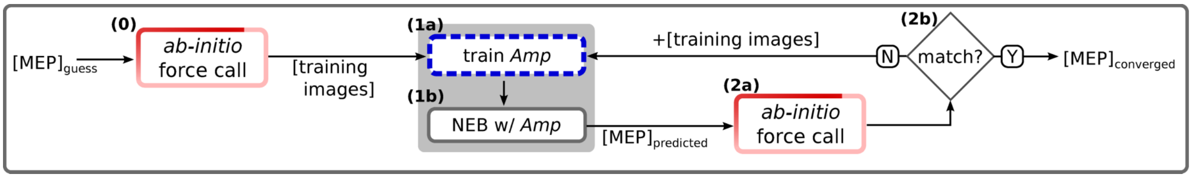

FIG. 1. Schematic of the proposed algorithm to accelerate saddle point and MEP searches.

popular atomistic and electronic-structure software. However, a machine-learning model is only as good as the region of data it is trained to, and extrapolation is generally quite poor. A general challenge in atomistic machine learning is finding relevant training data such that one can have confidence in predictions. So the question becomes: How can we identify relevant training data to enhance the representation in the region of interest—that is, near the minimum energy pathway—without *a priori* knowledge of the location of this region?

Here, we suggest a framework by which atomistic machine-learning tools can be used to accelerate tasks such as the search for MEPs and saddle points. The algorithm is outlined below and summarized in Figure 1. In step (**0**), an initial guess is made of the reaction path ("$[\text{MEP}]_{\text{guess}}$"), such as an interpolation between the initial- and final-state images, and evaluated with the *ab initio* calculator. This is identical to the beginning of the standard NEB routine. These *ab initio* calculated images become the training images for our machine learning calculator, *Amp*. In step (**1a**), *Amp* is trained to these images. In step (**1b**), *Amp* is used with the standard NEB routine to predict the minimum energy pathway; since these calculations are performed in the fast machine-learning code they can typically be performed orders of magnitude faster than in the *ab initio* code. The predicted band, $[\text{MEP}]_{\text{predicted}}$, is checked with *ab initio* calculations (step (**2a**)). Should the *ab initio* calculations match the *Amp* predictions in energy and forces (step (**2b**)), the routine is converged. If they do not match, these new images are added to the training set and *Amp* is retrained with the new data, cycling the technique through step (**1a**). This selectively improves the *Amp* representation in a region approaching the MEP by generating new *ab initio* training data increasingly close to the region of interest.

A key feature of machine learning is that the model improves with access to more experiences, without the user needing to interact to improve the model. We describe how this feature is used to accelerate our method. The initial training set is minimal and consists solely of the initial-guess band of images. We can expect that a machine-learning (ML) model that is trained to this small set will give a reasonably poor estimate of the true MEP; yet, we can expect that running a NEB with this ML model will move the band in the correct direction, towards of the MEP. The ML-predicted MEP is checked with *ab initio* calculations. Of course, if the prediction matches the *ab initio* calculations, the routine is stopped. However, if it does not match (and the ML calculator was properly trained), this can only mean we have encountered a region of the PES where we have insufficient training data. Thus, we have strategically identified new training points to improve our ML model. As long as these images are closer to the true MEP, our next guess should be substantially improved, and each subsequent "wrong" ML attempt results in new, relevant data, which improves the next attempt.

We demonstrate the feasibility of the routine with a simple example that we have constructed such that it can be represented in a two-dimensional PES, shown in Figure 2. This process is the diffusion of an Au atom across an Al surface substituted with Pt atoms to give the PES some asymmetry. The two independent dimensions of the PES are taken as the x and z coordinates of the diffusing Au atom. We used the EMT calculator[16] (available in Atomic Simulation Environment (ASE)[17]) to represent the energetics of the system, such that we could construct a high-resolution visualization of the "*ab initio*" PES, which is shown as the background color of the figure; full details are given in

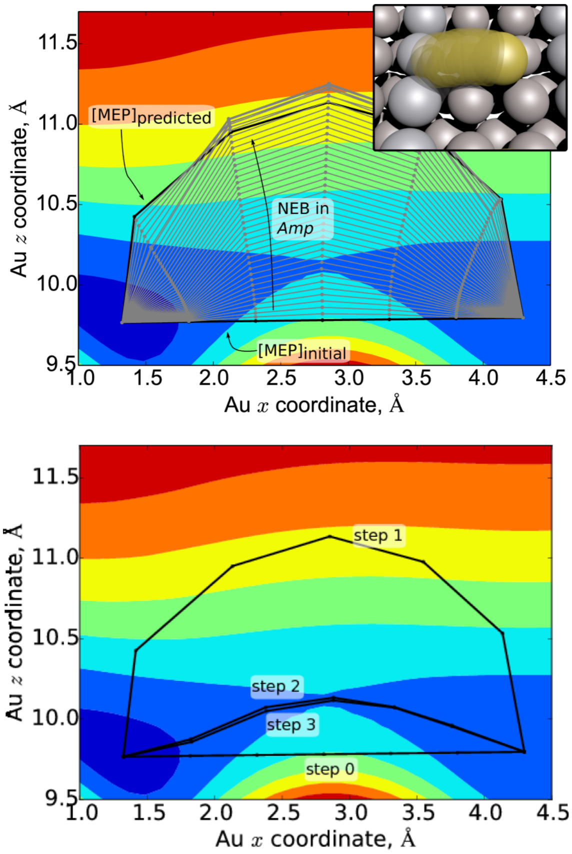

FIG. 2. The ML-NEB routine for diffusion of an Au atom on a fcc(100) Al surface with Pt alloyed. The top figure shows the first step of the algorithm, with black points representing the *ab initio* calculations and gray points the *Amp* calculations. The bottom figure shows the subsequent steps of the algorithm, which finds the true MEP in two more steps. On the colormap of the PES; dark blue is low energy and red is high energy. An analogous figure for the standard NEB is given in Figure S1 of the supplementary material.

Section S1 of the supplementary material. For the initial guess, we used a simple linear interpolation in atomic positions between the initial and final states; this initial band is calculated with the parent EMT calculator. The resulting 7 images are used as a small training set for the machine-learning calculator; in this case, we configured *Amp* to use the atom-centered Gaussian/neural-network (NN) scheme (i.e., the Behler–Parrinello scheme[10,18,19]) and trained *Amp* to reproduce both the energies and forces of the training set. The trained *Amp* instance is used to run a standard NEB calculation until it converges, as shown in the top PES of Figure 2. Here, we see the *Amp*-predicted MEP overshoots—that is, it makes a prediction across the PES from the true MEP; in practice this results in large discrepancies between the *Amp* predictions and the *ab initio* verifications of the predicted MEP. However, we can understand that when these images are added to the training set, the intermediate region near the true MEP is likely to become better represented—in this case, the job of machine learning becomes closer to interpolation than extrapolation. Indeed, after re-training the ML calculator and re-running the NEB, we see that a prediction is made that very closely approximates the true MEP. At this stage, only a single extra iteration is required to find the true MEP, taking a total of 4 band force calls to make an agreement that matches the true PES. In this simple case, the standard NEB takes only about 12 band force calls to find the MEP; although we see a significant computational savings, the benefits may be greater in more difficult-to-find MEPs.

To examine a more challenging system, we choose the example of a 120° C–C bond rotation in ethane; this is the same system chosen by Jónsson and co-workers[6] in their recent work on improved initial guesses for the NEB method. This system has a rather complex MEP to describe in unconstrained Cartesian coordinates: as the front methyl group rotates counterclockwise, the rear methyl group first follows the front in a counterclockwise direction, then reverses rotation such that the saddle point can be crossed, then the rear methyl group again follows the counterclockwise direction of the front methyl group. This is shown schematically in Figure 3; an animated GIF showing this is included as Figure S2 of the supplementary material. Thus, in net the rear hydrogens reverse direction twice during the course of the MEP; this is far away from an initial guess of a linear interpolation of positions between the initial and final states and provides a greater challenge to the algorithm. We used our ML-NEB algorithm with the same general procedure as in the previous example; full details are in Section S1 of the supplementary material. Importantly, we tested this with both an initial guess of straight-line interpolation between the initial state and final state, and using the improved methodology of Ref. 6, known as image-dependent pair potential (IDPP). We show the results in Figure 3. We see with the standard NEB implementation, 66 band force calls are required; this can be improved substantially to 48 force calls using the IDPP methodology to provide a better initial guess. (We note that we are using slightly different convergence criteria and calculators than Ref. 6, so the number of steps to convergence is different than their report, but in qualitative agreement.) Encouragingly, with our ML-NEB algorithm, this takes only 4 band force calls to successfully find the MEP when pairing with a straight interpolative initial guess; if we also employ IDPP, convergence took only three steps.

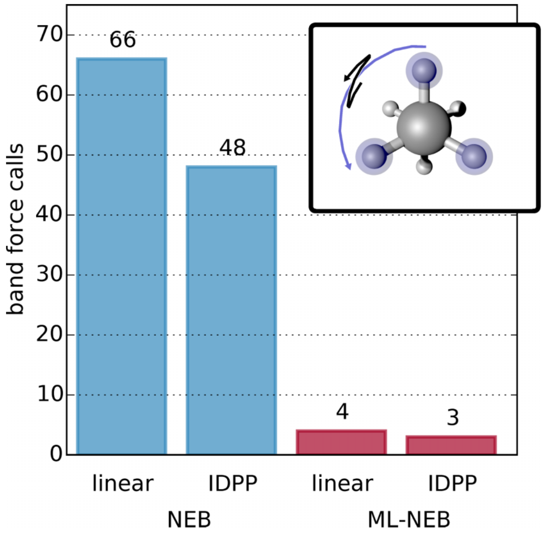

FIG. 3. The ML-NEB routine applied to finding the MEP for ethane rotation. The standard NEB calculation requires 66 band force calls, or 48 force calls when using the IDPP method[6] to improve the initial guess. The ML-NEB implementation offered substantial acceleration, taking 4 steps with a straight interpolative initial guess and three when paired with the IDPP approach.

Given these demonstrations, we discuss future prospects and note several subtleties of the methods discussed. First, the simple method described and demonstrated here only serves to demonstrate that such accelerations are possible; we expect that the methodology will be improved as it is applied to a greater number and variety of systems, including those of more significant complexity. Second, in the discussions above, we assumed the user had no prior data. Of course, in practice when a user reaches the point in their research of searching for a saddle point, they have likely already performed hundreds if not thousands of single-point *ab initio* calculations on their system or closely related ones. These images can be used as the initial training data, perhaps precluding the need for the initial band of *ab initio* calculations and almost certainly leading to a better representation of the PES, which should further speed convergence.

Third, in our examples above, we employ a neural-network (NN) model for the machine-learning algorithm. While NN models have some key advantages for large training data sets, they suffer from one key drawback: the cost function used to optimize the model parameters is not convex. In practice, this means that the best-fit parameters will depend upon the initial guess of NN parameters, which are typically initialized with random guesses. Often, it can occur that the NN cost function becomes "trapped" in a local minimum (in NN parameter space, although this is highly analogous to multiple local minima on a PES!) that does not adequately allow the model to represent the training data. When this occurs, it is not apparent whether it is due to a poor initial guess or an insufficiently sized model. This can lead to slow training of the machine-learning model, especially when not starting from a previously trained model. Other machine-learning models, such as support vector regression

(SVR), may perform better for this application. As compared to the NN model, SVR is deterministic so will always converge to a single, best solution. A disadvantage of SVR that makes it not suitable for general atomistic modeling is that the model size grows with the training set size; leading to a very large and slow model for very large training sets, such as tens of thousands of images. However, in the present application, the training sets are small, so an SVR model may offer superior characteristics to an NN model.

Fourth, we note that our approach is relatively agnostic to the choice of MEP (or saddle-point) search algorithm employed, as the search algorithm is done relatively quickly in the machine-learning representation. That is, as long as the search algorithm finds the MEP, the path to the MEP and the number of steps taken is of secondary importance. Therefore, our approach is not tied to the NEB method but should in principle pair well with any method that utilizes forces to converge to the true saddle point or MEP, including Hessian-free versions.[20]

Finally, in the current work, we have chosen to focus on the search for saddle points, as these searches are often the most computationally challenging of the routine tasks we encounter in our research and because the search for these barriers can be a severe bottleneck in applications such as electrochemistry. However, it is relatively straightforward to envision an adaptation of these methods to accelerate simpler stationary point searches, such as gradient-descent-type methods that converge into local minima on the PES. In fact, all types of stationary point searches should be ideal applications of approximate machine-learning models, as the single-point predictions can directly be checked with the *ab initio* calculator to see if the gradient is indeed zero at the predicted image; when it disagrees this data point will strategically improve the ML representation of the PES. With imagination, we expect that methods such as those described here can be used to accelerate many other routine tasks in atomistics, including global optimization schemes.

SUPPLEMENTARY MATERIAL

See supplementary material for a description of the methods and parameters used in both examples, a figure showing the full NEB for the Au diffusion example, and an animated GIF of ethane rotation.

ACKNOWLEDGMENTS

The author thanks Alireza Khorshidi (Brown) for vigorous discussions and acknowledges financial support from the Office of Naval Research through Award No. N00014-16-1-2355.

[1] 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).
[2] H. Eyring, "The activated complex in chemical reactions," J. Chem. Phys. **3**, 107–115 (1935).
[3] H. Jónsson, G. Mills, and K. W. Jacobsen, "Nudged elastic band method for finding minimum energy path of transitions," in *Classical and Quantum Dynamics in Condensed Phase Simulations*, edited by B. Berne, G. Ciccotti, and D. Coker (World Scientific, 1998), Vol. 385.
[4] 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).
[5] 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).
[6] S. Smidstrup, A. Pedersen, K. Stokbro, and H. Jónsson, "Improved initial guess for minimum energy path calculations," J. Chem. Phys. **140**, 214106 (2014).
[7] M. Melander, K. Laasonen, and H. Jónsson, "Removing external degrees of freedom from transition-state search methods using quaternions," J. Chem. Theory Comput. **11**, 1055–1062 (2015).
[8] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural Networks **2**, 359–366 (1989).
[9] S. Lorenz, A. Groß, and M. Scheffler, "Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks," Chem. Phys. Lett. **395**, 210–215 (2004).
[10] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," Phys. Rev. Lett. **98**, 146401 (2007).
[11] N. Artrith, T. Morawietz, and J. Behler, "High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide," Phys. Rev. B **83**, 153101 (2011).
[12] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," Phys. Rev. Lett. **104**, 136403 (2010).
[13] L. Raff, R. Komanduri, M. Hagan, and S. Bukkapatnam, *Neural Networks in Chemical Reaction Dynamics* (Oxford University Press, 2012).
[14] A. Khorshidi and A. A. Peterson, "Amp: A modular approach to machine learning in atomistic simulations," Computer Physics Communications (in press).
[15] A. Khorshidi and A. Peterson, Amp: Machine-learning for atomistics, v0.4, http://dx.doi.org/10.5281/zenodo.46737, Zenodo, 2016.
[16] J. K. Nørskov and N. D. Lang, "Effective-medium theory of chemical binding: Application to chemisorption," Phys. Rev. B **21**, 2131–2136 (1980).
[17] S. R. Bahn and K. W. Jacobsen, "An object-oriented scripting interface to a legacy electronic structure code," Comput. Sci. Eng. **4**, 56–66 (2002).
[18] N. Artrith and J. Behler, "High-dimensional neural network potentials for metal surfaces: A prototype study for copper," Phys. Rev. B **85**, 045439 (2012).
[19] N. Artrith and A. M. Kolpak, "Understanding the composition and activity of electrocatalytic nanoalloys in aqueous solvents: A combination of DFT and accurate neural network potentials," Nano Lett. **14**, 2670–2676 (2014).
[20] 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).