

OptiBoost: A Method for Choosing a Safe and Efficient Boost for the Bond-Boost Method in Accelerated Molecular Dynamics Simulations with Hyperdynamics

Jianming Cui¹ and Kristen A. Fichthorn^{1,2,*}

Department of Chemical Engineering¹ and Department of Physics²

The Pennsylvania State University

University Park, PA 16802 USA

*Corresponding Author: fichthorn@psu.edu

(Dated: 19 April 2022)

Accelerated molecular-dynamics (MD) simulations based on hyperdynamics (HD) can significantly improve the efficiency of MD simulations of condensed-phase systems that evolve via rare events. However, such simulations are not generally easy to apply since appropriate boosts are usually unknown. In this work, we developed a method called OptiBoost to adjust the value of the boost in HD simulations based on the bond-boost method. We demonstrated the OptiBoost method in simulations on a cosine potential and applied it in three different systems involving Ag diffusion on Ag(100) in vacuum and in ethylene-glycol solvent. In all cases, OptiBoost was able to predict safe and effective values of the boost, indicating the OptiBoost protocol is an effective way to advance the applicability of HD simulations.

This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.
PLEASE CITE THIS ARTICLE AS DOI:10.1063/5.0088521

I. INTRODUCTION

The temporal evolution of many materials systems is governed by rare events, where the system spends a relatively long time in one free-energy minimum before escaping and moving on to another one. While molecular dynamics (MD) simulations can, in principle, elucidate the atomic-scale processes and rates of rare events, the practical time scale is limited to the microsecond range. Accelerated MD methods were developed to address this issue.¹⁻⁸ In this paper, we focus on hyperdynamics (HD)^{1,2} – one accelerated MD technique that has been used in many studies.^{6,7,9-16} As we will elaborate below, a bias potential is added to the potential energy near the minima in HD. A careful design of the bias potential induces rapid transitions without affecting the relative transition frequency and a weighted time increment allows for long-time simulations that can exceed the microscale by many orders of magnitude.

One HD bias potential is given by the bond-boost method.^{17,18} In this method, the strain of the bond between atoms i and j is defined as

$$\varepsilon_{ij} = \frac{(R_{ij} - R_{ij}^0)}{R_{ij}^0}, \quad (1)$$

where R_{ij} is the current distance between atoms i and j , and R_{ij}^0 the distance when the potential energy is at a minimum. A bias potential, or “boost”, is applied to the maximally stretched bond – the bond that is most likely to “break” and result in a transition. For a fixed atom configuration $\mathbf{R}$, the boost energy ΔV of the maximally stretched bond, with ε_{max} is given by

$$\Delta V(\mathbf{R}) = \begin{cases} \Delta V_{max} \left(1 - \left(\frac{\varepsilon_{max}(\mathbf{R})}{q} \right)^2 \right), & |\varepsilon_{max}| < q \\ 0, & \text{else} \end{cases}, \quad (2)$$

where ΔV_{max} is a parameter that controls the magnitude of the boost and q is a cutoff parameter of the limiting strain at which the bias potential goes to zero. The q parameter should be set such that the boost is zero at transition states. MD trajectories are then run on a “boosted” potential-energy surface (PES) $V^*(\mathbf{R})$, where the potential energy is given by

$$V^*(\mathbf{R}) = V(\mathbf{R}) + \Delta V(\mathbf{R}) \quad . \quad (3)$$

There are two relevant times in HD simulations. The first is the MD time t_{MD} , which is given by

$$t_{MD} = N\Delta t \quad , \quad (4)$$

where N is the number of MD time steps and Δt is the MD time increment, typically on the order of fs. The physical time is given by

$$t = \sum_{i=1}^N \Delta t \exp \left[\frac{\Delta V(\varepsilon_i)}{kT} \right] \quad , \quad (5)$$

where ΔV_i is the boost applied at time step i . The time-boost factor B is used to assess the acceleration of HD simulations by the boost potential. It is given by the ratio of Eqs. (5) and (4), i.e.,

$$B = \frac{1}{N} \sum_{i=1}^N \exp \left[\frac{\Delta V_i}{kT} \right] \quad , \quad (6)$$

It is evident from Eq. (2) that the values of both q and ΔV_{max} control the time-boost factor in Eq. (6). In this work, we adopt the value of $q = 0.3$, which has been shown to be a safe value for many cases in previous studies,^{16, 17} and we focus our efforts on establishing an appropriate value of ΔV_{max} .

Figure 1 illustrates both the promise and the difficulty in using HD simulations with the bond-boost method. Figure 1 shows the escape rate from HD simulations as a function of ΔV_{max} run for motion on a one-dimensional cosine potential of the form

$$V(x) = \frac{1}{2} \cos 2\pi x - \frac{1}{2} \quad . \quad (7)$$

The boosted potential in these simulations is given by Eq. (3) and the bias potential is given by Eq. (2) with $q = 0.3$. The inset to Figure 1 shows the time-boost factor from Eq. (6) as a function of ΔV_{max} . For each value of ΔV_{max} , we ran 500 different trajectories on the cosine potential using Mathematica® to integrate the equation of motion with the Verlet algorithm and the Andersen thermostat. The unit of mass is m_0 , the unit of time is t_0 , the unit of length is x_0 , and the simulations were run at a temperature of $kT = 0.2 m_0 x_0^2 / t_0^2$. We calculated the rate as the reciprocal of the average physical time, given by Eq. (5), to exit a minimum after entering. The exact rate to move from one minimum to another on the cosine potential is given by

$$r = \frac{1}{2} \left(\frac{2kT}{\pi m} \right)^{\frac{1}{2}} \left(\frac{\exp \left[\frac{-V(0)}{kT} \right] + \exp \left[\frac{-V(1)}{kT} \right]}{\int_0^1 \exp \left[\frac{-V(x)}{kT} \right] dx} \right) \quad , \quad (8)$$

where m is the mass, which we take to be $1.0m_0$.

In Figure 1, we see that the simulated rate matches the exact rate given by Eq. (8) until $\Delta V_{max} \cong 1.2 m_0 x_0^2 / t_0^2$, then the simulated rate begins to deviate from the exact value and the difference between the exact and simulated rate increases with increasing ΔV_{max} . In the inset to Figure 1, we see the boost increases with increasing ΔV_{max} up to $\Delta V_{max} \cong 1.2 m_0 x_0^2 / t_0^2$. At $\Delta V_{max} = 1.2 m_0 x_0^2 / t_0^2$, the time-boost factor has reached a value of $B = 81$, meaning that the HD simulation is 81 times faster than regular MD. While the large efficiency afforded by HD simulations is promising, this can come at the expense of accuracy if ΔV_{max} is too large.

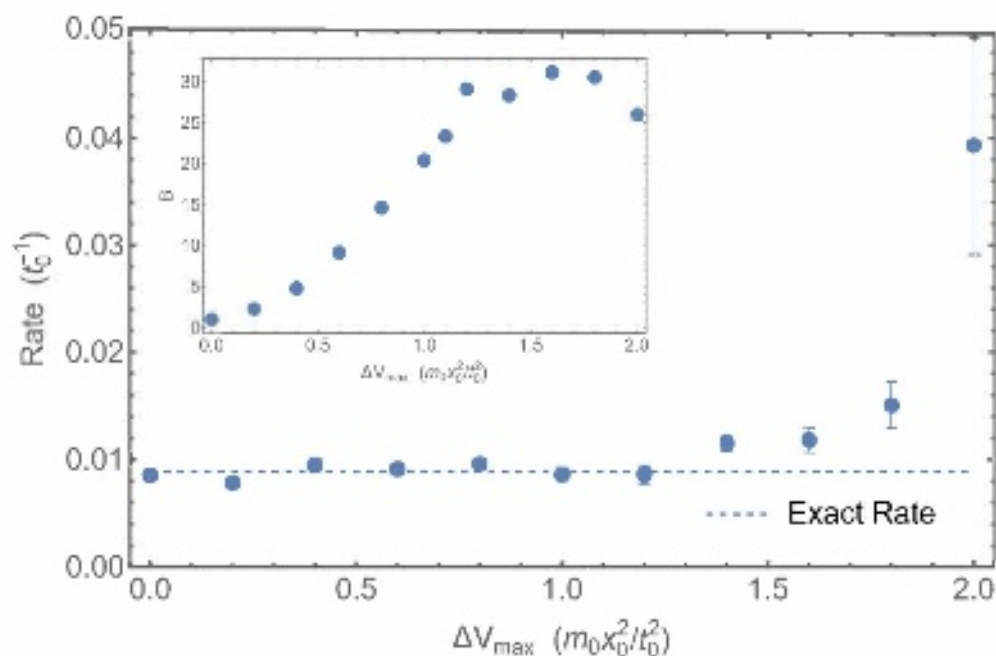


Figure 1. Plot of the rate obtained for escaping the minima of the cosine potential given by Eq. (7) as a function of ΔV_{max} . The exact rate from Eq. (8) is given by the dashed line. The inset shows the boost given by Eq. (6) as a function of ΔV_{max} .

The origins of the deviation of the rate from the exact value in Figure 1 can be seen from a plot of $V^*(x)$, given by Eq. (3) and shown in Figure 2 for the cosine potential with different values of ΔV_{max} . For $\Delta V_{max} = 0.2 \ m_0 x_0^2 / t_0^2$, the biased PES $V^*(x)$ retains the same shape as the original potential ($\Delta V_{max} = 0$). From the inset to Figure 1, the boost for $\Delta V_{max} = 0.2$ is around 2.3, which is non-negligible but modest in terms of what can be achieved. When ΔV_{max} increases to 0.8, the shape of $V^*(x)$ begins to deviate from that of the original potential. However, the shape deviation is not large and the rate on the boosted surface in Figure 1 is still the same as that on the original surface. From the inset to Figure 1, the boost for $\Delta V_{max} = 0.8$ is almost 15, which represents a significant increase in efficiency over $\Delta V_{max} = 0.2$. When we reach $\Delta V_{max} = 2.0$, the shape of V^* is significantly distorted from the original potential, as there is a maximum where a minimum

occurs on the original potential, and sub-minima appear near the transition states. Trajectories become trapped in the sub-minima on V^* and they cannot easily access the region of the original minimum. As a result, the rate increases on V^* compared to that on the original potential and becomes inaccurate.

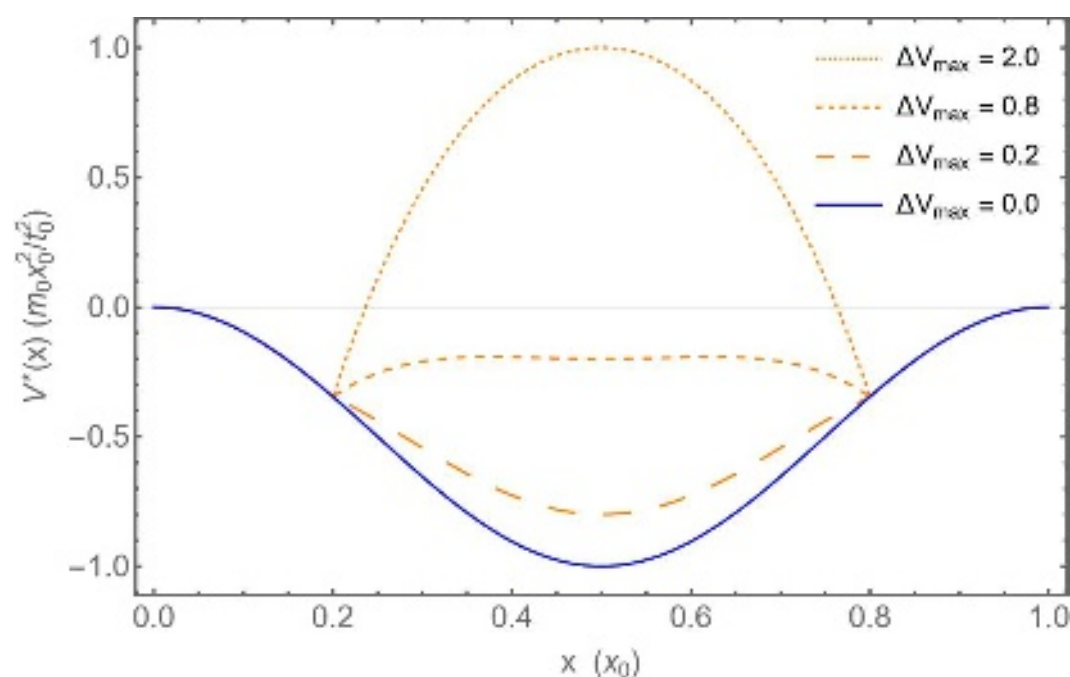


Figure 2. A plot of one period of the cosine potential given by Eq. (7) ($\Delta V_{max} = 0$), along with the boosted cosine potential from Eq. (3) for various boost magnitudes ΔV_{max} .

Thus, care needs to be taken in choosing the boost parameter for HD simulations based on the bond-boost method. As Figures 1 and 2 demonstrate, the magnitude of ΔV_{max} should not far exceed the value of the energy barriers in the system. In the current state of HD simulations, a safe and effective magnitude of ΔV_{max} is usually unknown and needs to be decided before running the simulation. In this work, we develop a method called OptiBoost to adjust the magnitude of the boost in HD simulations with the bond-boost method.

II. METHODS AND RESULTS

A. Cosine Potential

The main ideas behind the OptiBoost method are (1) The bond-boost potential in Eq. (2) affects the potential energy of a bond on an arbitrary 3N-dimensional PES in a similar way that it affects the potential energy of the cosine potential shown in Figure 2; (2) Dynamics on $V^*(x)$, but on the MD time scale, given by Eq. (4), displays a distinct signature of the boost potential. Regarding (1), the bond-boost potential is applied to bonds, whose potential energies as a function of strain are typically harmonic or slightly anharmonic around their minima and have a similar shape to the cosine potential. Regarding (2), the probability P that a trajectory originating in the potential well will escape to a neighboring well within a time t_{MD} is given by

$$P(t_{MD}) = 1 - e^{-r_{MD}t_{MD}}, \quad (9)$$

where r_{MD} is the escape rate on the MD time scale [i.e., not the physical time scale given by Eq. (5)]. For the “boosted” cosine potential in Figure 2, the escape rate on the MD time scale is given by

$$r_{MD} = \frac{1}{2} \left(\frac{2kT}{\pi m} \right)^{\frac{1}{2}} \left(\frac{\exp \left[\frac{-V^*(0)}{kT} \right] + \exp \left[\frac{-V^*(1)}{kT} \right]}{\int_0^1 \exp \left[\frac{-V^*(x)}{kT} \right] dx} \right). \quad (10)$$

Figure 3 shows a plot of Eq. (9) (dashed lines) as a function of ΔV_{max} for three different time intervals. From Figure 3, we see $P(\Delta V_{max})$ has an approximate sigmoid form in each case. This form can be seen to originate from Eq. (9), which we can write as

$$P(t_{MD}) = \frac{e^{r_{MD}t_{MD}} - 1}{e^{r_{MD}t_{MD}}} = \frac{r_{MD}t_{MD} + \frac{1}{2}(r_{MD}t_{MD})^2 + \dots}{1 + r_{MD}t_{MD} + \frac{1}{2}(r_{MD}t_{MD})^2 + \dots}, \quad (11)$$

If all the rates are small, as they are in this example (and as is generally the case for rare events), and the time intervals t_{MD} are sufficiently short, we have

$$P(t_{MD}) = \frac{r_{MD}t_{MD}}{1 + r_{MD}t_{MD}} \quad , \quad (12)$$

where we note that $r_{MD} = r_{MD}(\Delta V_{max})$. We note that Eq. (12) has a sigmoid shape. As we can anticipate from Figure 2, r_{MD} is a complicated function of ΔV_{max} , though we expect r_{MD} to increase with increasing ΔV_{max} and approach a constant value when ΔV_{max} is sufficiently large. Both features will promote a sigmoid curve for $P(t_{MD})$ with increasing ΔV_{max} , though not of the exact form as Eq. (12). Because we will not generally know the exact relationship between r_{MD} and ΔV_{max} for any arbitrary system, we can write P as

$$P(\Delta V_{max}) \cong \frac{A}{1 + B e^{-C \Delta V_{max}}} \quad . \quad (13)$$

This equation also has a sigmoid shape in which we can handle the fact that $r_{MD} = r_{MD}(\Delta V_{max})$ using three adjustable parameters (A, B, and C).

To test Eq. (13), we ran a series of HD simulations on the boosted cosine potential for various values of ΔV_{max} . We calculated the value of P as an average over 1000 trajectories, each with a duration of t_{MD} . Figure 3 shows the results from the simulations as points and a fit of the simulation results to the sigmoid curve in Eq. (13) as lines. It is evident there is excellent agreement between the exact values of P (dashed lines), the simulation values (points), and the fits to a sigmoid curve (solid lines).

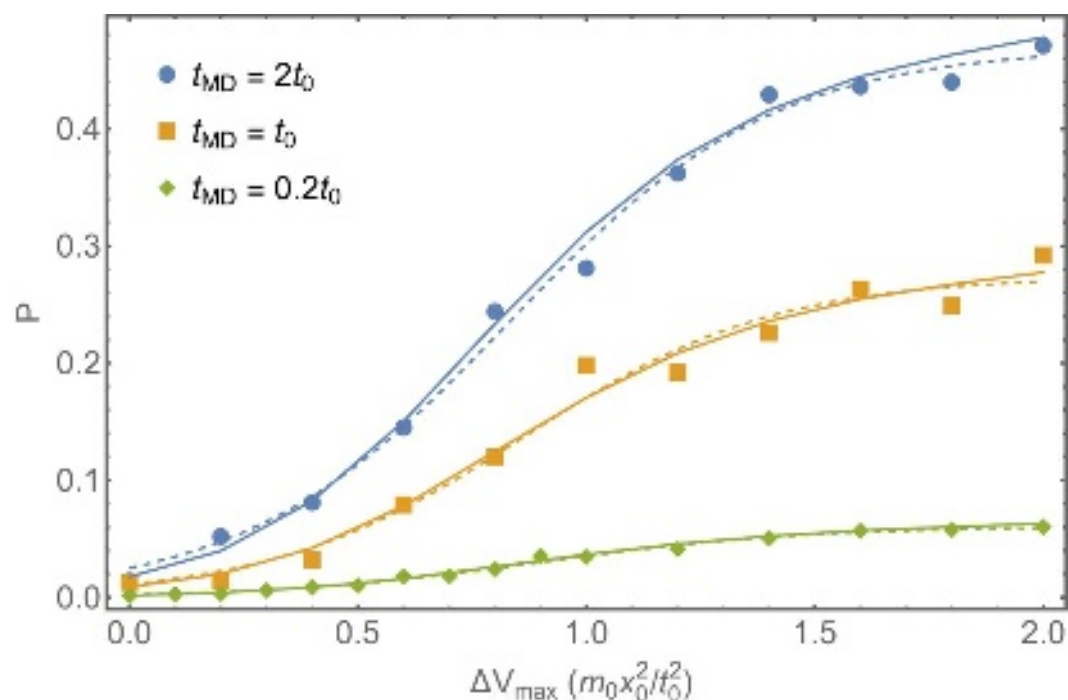


Figure 3. Plot of Eq. (9) (dashed lines) as a function of ΔV_{max} for short trajectories of various durations t_{MD} . Symbols are results from short MD trajectories and solid lines are fits of the MD results to the sigmoid function in Eq. (13).

The excellent fit of the sigmoid curve to the P data in Figure 3 is an opportunity to define an appropriate boost parameter. Namely, $P(\Delta V_{max})$ reaches an asymptotic value for large ΔV_{max} because $V^*(x)$ developed a substantial maximum where the original potential $V(x)$ had a minimum, resulting in sub-minima that trap the trajectory near the transition state. We can identify the value of ΔV_{max} where P begins to reach an asymptotic value from the minimum in $P''(\Delta V_{max})$. This value of ΔV_{max} should represent an aggressive boost. From this perspective, the inflection point in $P(\Delta V_{max})$, where $P''(\Delta V_{max}) = 0$ and $P'(\Delta V_{max})$ exhibits a maximum, is a well-defined quantity that should provide a safe estimate of ΔV_{max} .

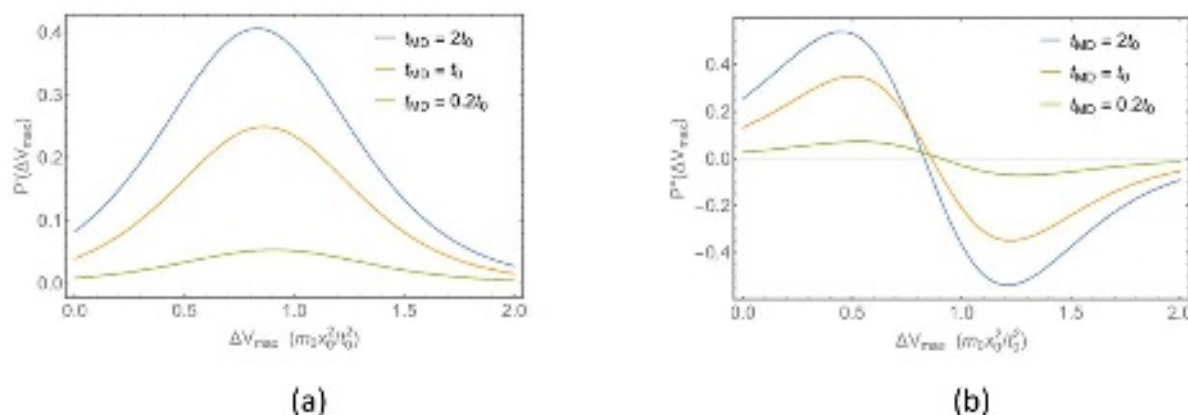


Figure 4. Plots of (a) $P'(\Delta V_{max})$ and (b) $P''(\Delta V_{max})$ for various simulation times t_{MD} on the cosine potential.

Figure 4(a) shows $P'(\Delta V_{max})$ and Figure 4(b) shows $P''(\Delta V_{max})$ for the simulations shown in Figure 3. The maxima in Figure 4(a) and the minima in Figure 4(b) are listed in Table I. It is evident from Figures 1 and 2 that the barrier of the cosine potential is $1.0 m_0 x_0^2 / t_0^2$ and that HD simulations become inaccurate when ΔV_{max} becomes much greater than the barrier. From Table I we see that ΔV_{max} implied from the maximum in $P'(\Delta V_{max})$ is less than the barrier and the point of minimum $P''(\Delta V_{max})$ is somewhat larger than the barrier. Though the values of ΔV_{max} vary for different time intervals, they are similar. From the viewpoint of computational efficiency, it is advisable to use the smallest time interval t_{MD} that will yield the sigmoid form – producing $P \cong 0$ for ΔV_{max} around zero and a non-zero plateau for large ΔV_{max} . From Figure 3, we see t_{MD} as small as $0.2t_0$ are acceptable for the cosine potential. Generally, the smallest acceptable time interval depends on the barrier for the transition under consideration.

From Table I, it is evident that maxima on $P'(\Delta V_{max})$ provide safe and conservative estimates for ΔV_{max} . In Figure 1, these values all yield the exact rate, with time-boost factors of $B \approx 15$. ΔV_{max} estimates from minima on $P''(\Delta V_{max})$ also yield exact rates in Figure 1, with time-boost

factors of $B \approx 30$ – however the HD rate begins to deviate from the exact value immediately beyond these estimates. Thus, ΔV_{max} estimates from $P'(\Delta V_{max})$ are safe and those from $P''(\Delta V_{max})$ are bold, but likely still satisfactory. In summary, the OptiBoost method involves (1) Obtaining escape probabilities P as a function of ΔV_{max} from a series of short trajectories; (2) Fitting $P(\Delta V_{max})$ to the sigmoid function in Eq. (13); (3) Delineating the boost range from the maximum in $P'(\Delta V_{max})$ and from the minimum in $P''(\Delta V_{max})$.

Table I. Values of optimal ΔV_{max} for various simulation times t obtained from the plots of $P'(\Delta V_{max})$ and $P''(\Delta V_{max})$ in Figure 4.

$t(t_0)$	Maximum in $P'(\Delta V_{max}) (m_0 x_0^2 / t_0^2)$	Minimum in $P''(\Delta V_{max})$ $(m_0 x_0^2 / t_0^2)$
$2t_0$	0.83	1.21
t_0	0.86	1.22
$0.2t_0$	0.91	1.28

B. Ag Diffusion on Ag(100)

To demonstrate that the results for the cosine potential can be observed for other systems, we applied OptiBoost to three different systems involving Ag atom diffusion on Ag(100): (1) An Ag atom in vacuum; (2) An Ag dimer in vacuum; and (3) An Ag atom in the presence of ethylene glycol (EG) solvent. The third case is relevant to understanding the growth of Ag nanocrystals in solution.¹⁹⁻²¹ We simulated these system using the LAMMPS code,²² version 29Oct2020, compiled with the ‘REPLICA’, ‘MANYBODY’, ‘MOLECULE’, ‘KSPACE’, ‘USER-MISC’ and ‘PYTHON’ packages. LAMMPS was used as a package to Python, for which we also imported

‘multiprocessing’, ‘os’, ‘random’, ‘shutil’, and ‘numpy’. We note that the parallel implementation of HD in LAMMPS has been described by Plimpton recently.¹⁶

We used an embedded-atom method (EAM) potential for Ag₂₃ and the MOMB force field for the Ag-EG and EG-EG interactions.²⁴⁻²⁷ The size of the Ag substrate in the vacuum systems was $8 \times 8 \times 6$ atoms and the substrate consisted of $6 \times 6 \times 6$ atoms with 48 EG molecules in the Ag-EG system. Prior to production runs, we performed equilibration in the NPT ensemble using the Nosé-Hoover thermostat and barostat for 10 ns to account for thermal expansion. All systems were equilibrated at 433 K and 1 bar. The molar mass of EG is 62.02 g/mol and the volume of EG in the system was $17.5 \times 17.5 \times 13 \text{ \AA}^3$ after NPT equilibration. Using 48 EG molecules, the density is 1.24 g/mL, a value somewhat higher than the experimental value of 1.1 g/mL.²⁸ Subsequent HD production runs were performed in the NVT ensemble using the Langevin thermostat. The MD time step was 1 fs in all runs.

For the HD simulations, we use $q = 0.3$ [see Eq. (2)]. There are several specific LAMMPS parameters for HD simulations.¹⁶ One is the bond cutoff D_{bond} , which we take to be $3.32 \text{ \AA}$. The bond cutoff is the distance over which LAMMPS defines bonds for the bond-boost method. As the simulation runs, a check is performed every 1000 time steps to determine whether an event has occurred. In the solvent system, a check is also performed every 500 time steps and the observed event rate did not change. The check consists of quenching the system using the ‘quickmin’ method, with dimensionless energy and force tolerances of 0.1, a maximum of 40 iterations, and 50 force evaluations. An event is said to have occurred if the displacement of the new quenched state from the current quenched state is greater than a distance of D_{event} , which we take to be $D_{event} = 1.2 \text{ \AA}$. We note that this value of D_{event} is approximately 70% of the nearest-neighbor distance.

In LAMMPS, our OptiBoost HD simulations are driven by Python, in a protocol where N short trajectories are launched in parallel for each value of ΔV_{max} . The outcome of each trajectory is either “yes” if an event occurred or “no” if no event occurred. We then determine the event probability $P(\Delta V_{max})$ as the number of “yes” trajectories (N_{yes}) divided by the total number of launched trajectories, i.e., $P(\Delta V_{max}) = N_{yes}/N$. All analysis of $P(\Delta V_{max})$ (i.e., curve-fitting to the sigmoid function and determination of appropriate values of ΔV_{max}) is done in Python.

1. Ag Diffusion on Ag(100) in Vacuum

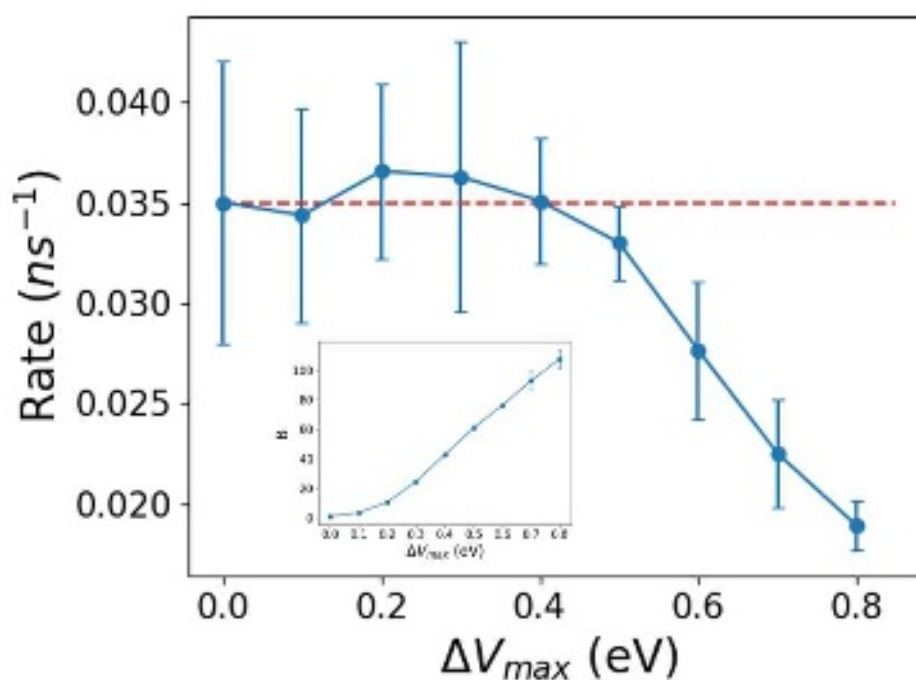


Figure 5. Observed surface diffusion rates in HD simulations of Ag adatom diffusion on Ag(100) in vacuum for different values of ΔV_{max} . The dashed line is the average rate for low values of ΔV_{max} . The inset shows the time-boost factor B for different ΔV_{max} .

We first performed HD simulations of a single Ag atom diffusing on Ag(100). Figure 5 shows the physical hopping rate as a function of ΔV_{max} . Each data point in Figure 5 is an average over 5 simulations, with each run covering 40~200 ns of MD time [t_{MD} in Eq. (4)]. The hop rate fluctuates around a value of 0.035 hops/ns for ΔV_{max} between 0.0-0.4 eV and then decreases for larger ΔV_{max} . This trend is opposite to that for the cosine potential in Figure 1 and the discrepancy can be attributed to the differences in way LAMMPS detects events and the way we checked for events in our Mathematica simulations.

In the Mathematica code for the cosine potential, the locations of the transition states are obvious, and we checked every time step for a transition-state crossing. In LAMMPS, an event check is performed every 1000 time steps and an event is detected when the current quenched state of the atom displaces longer than a distance of D_{event} from the previous one. The quench process in HD uses the ‘quickmin’ minimization routine without a strict tolerance, which implies the quenched state is not necessarily an energy minimum. When ΔV_{max} is large, the trajectory is confined near the transition state (as we see in Figure 2), with a small barrier such that the transition state could be crossed more than once before an event check occurs. Moreover, when a trajectory on the boosted surface is confined far from the minimum on the original potential, a relatively long distance needs to be travelled in ‘quickmin’ to reach the minimum. With loose tolerances, ‘quickmin’ times out, the minimization does not exactly reach the minimum, and the atom displacement is shorter than D_{event} . An event is not identified, and the net rate decreases. Though this may be seen as an impediment, we note that the HD simulations yield correct rates (i.e., the same rate as for diffusion with $\Delta V_{max}=0$) for a wide range of ΔV_{max} . Moreover, the value of ΔV_{max} where the boosted rate begins to deviate from the exact rate is around the value of the barrier for hopping in this system, which we earlier estimated from climbing-image nudged-elastic

bind (CI-NEB) calculations.²⁹ Additional simulation overhead is required for more frequent event checking and for more thorough optimization, so the loose parameters for these routines are justified for the present system. However, these parameters may not be satisfactory for all systems.

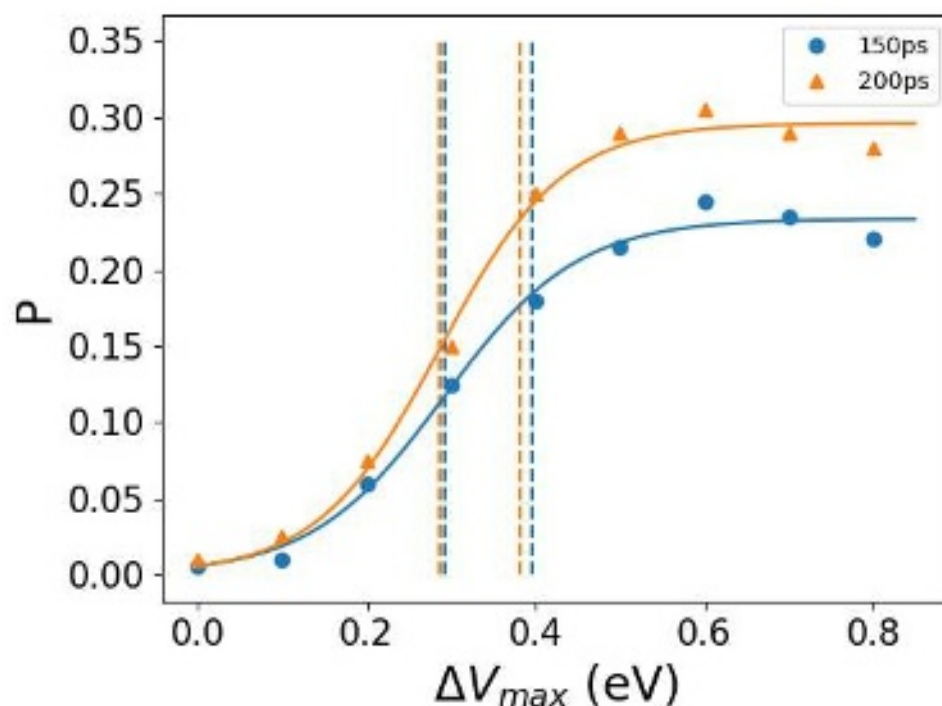


Figure 6. Event probabilities P for Ag atom surface diffusion on Ag(100) in vacuum as a function of ΔV_{max} . The lines are fits of the points to the sigmoid curve in Eq. (13). The vertical dashed lines indicate the range of safe and efficient ΔV_{max} summarized in Table II.

Event probabilities for single-adatom diffusion are shown as a function of ΔV_{max} for different short time intervals in Figure 6, along with their fits to the sigmoid curve in Eq. (13). Each data point in Figure 6 is an average over 200 simulations. Here, we see that P has a similar form to the cosine potential and it is evident that the sigmoid curve is an excellent fit to the data. As we discussed for the cosine potential, appropriate values of ΔV_{max} can be estimated from the maximum of $P'(\Delta V_{max})$ and/or the minimum of $P''(\Delta V_{max})$. Table II lists the optimal values of

ΔV_{max} obtained from these derivatives, along with the time-boost factor B for each of value of ΔV_{max} .

From Figure 5, we see that the ΔV_{max} predictions listed in Table II for $P'(\Delta V_{max})$ are safe, with values of less than half the barrier²⁹ (~0.5 eV) estimated from Climbing-Image Nudged Elastic Band (CI-NEB) calculations.³⁰ The ΔV_{max} predictions from $P''(\Delta V_{max})$ are also safe and are closer to the CI-NEB barrier. The time-boost factors for these two estimates are $B = 25$ from $P'(\Delta V_{max})$ and $B = 41-43$ from $P''(\Delta V_{max})$. To gauge the efficiency, we required 9 hours to obtain the 150 ps curve and the prediction range in Figure 6. The simulations to acquire P were run in parallel on 33 cores in several batches. For regular MD simulations, it will take 135 wall-clock hours to reach 1 microsecond in this system, running on 4 cores of Penn State's Roar cluster. With the time-boost factor range of 20-40 in Table II, a simulation with OptiBoost will require 12-14 hours (9 h for selecting a boost and 3-5 h for HD simulations running on 4 cores). It is evident that the computational effort has moved from executing the simulation to selecting the boost.

Table II. Values of optimal ΔV_{max} for various simulation times t_{MD} obtained from $P'(\Delta V_{max})$ and $P''(\Delta V_{max})$ for Ag adatom diffusion in Figure 6.

t (ps)	$P'(\Delta V_{max})$		$P''(\Delta V_{max})$	
	ΔV_{max}	B	ΔV_{max}	B
150	0.29	25	0.40	43
200	0.29	25	0.38	41

Even with the overhead of selecting an appropriate value of ΔV_{max} , HD simulations are an order of magnitude faster than regular MD. Here, we note that (1) the efficiency of selecting an appropriate value of ΔV_{max} , could be improved by more extensive, or complete parallelization; (2)

a boost can be selected “once and for all” to benefit future simulations, dramatically increasing the efficiency. Thus, once the boost(s) have been established for a particular system, HD simulations with the bond-boost method would be more than 20 times faster than regular MD.

2. Ag Dimer Diffusion on Ag(100) in Vacuum

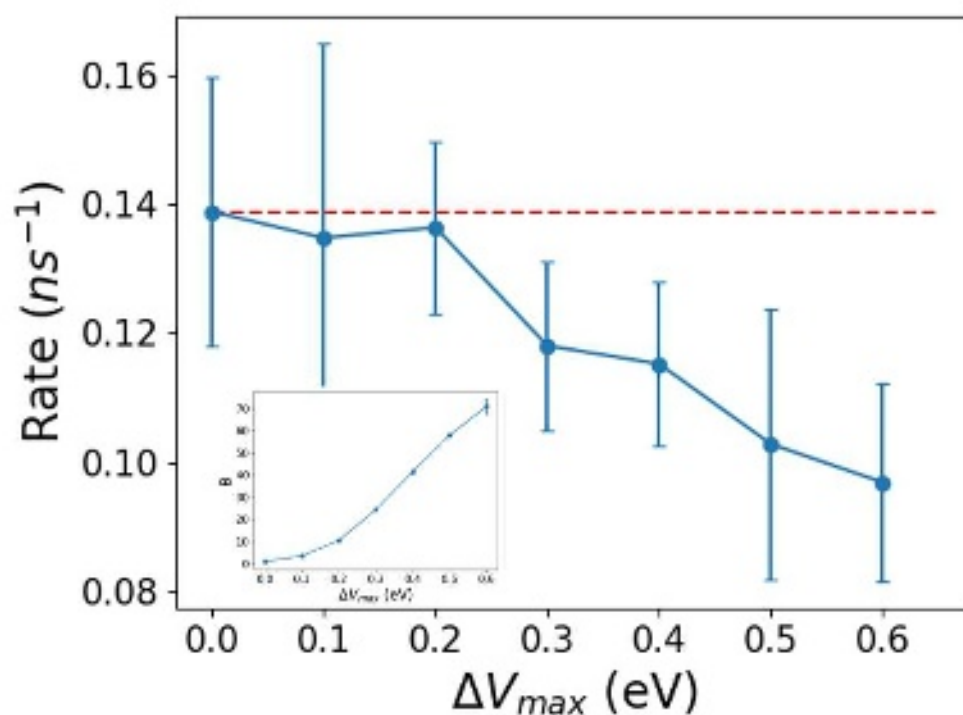


Figure 7. Net diffusion rate of Ag dimer on Ag(100) in vacuum as a function of ΔV_{max} . The dashed line is the average rate for low values of ΔV_{max} . The inset shows the time-boost factor as a function of ΔV_{max} .

Figure 7 shows the physical hopping rate as a function of ΔV_{max} for Ag dimer diffusion on Ag(100) in vacuum. Each data point in Figure 7 is an average of 5 simulations, with each run covering 40-100 ns of MD time [t_{MD} in Eq. (4)]. The observed dimer event rate fluctuates around 0.14 hops/ns for ΔV_{max} from 0.0-0.20 eV and then decreases. Unlike single-adatom hopping and

motion on the cosine potential, several different kinds of events can occur for dimer diffusion and the rate in Figure 7 reflects all these different events. The most frequent dimer motions are twirling dissociation and linear dissociation. We quantified these two events using CI-NEB calculations,³⁰ as shown in Figure 8. The smallest-barrier event is backward aggregation from twirling dissociation in Figure 8(a), which has a barrier of 0.24 eV and we note that the HD simulations in Figure 7 become inaccurate when the boost is larger than around 0.2 eV. The forward and backward barriers for linear dissociation, indicated in Figure 8(b), are both larger than this value.

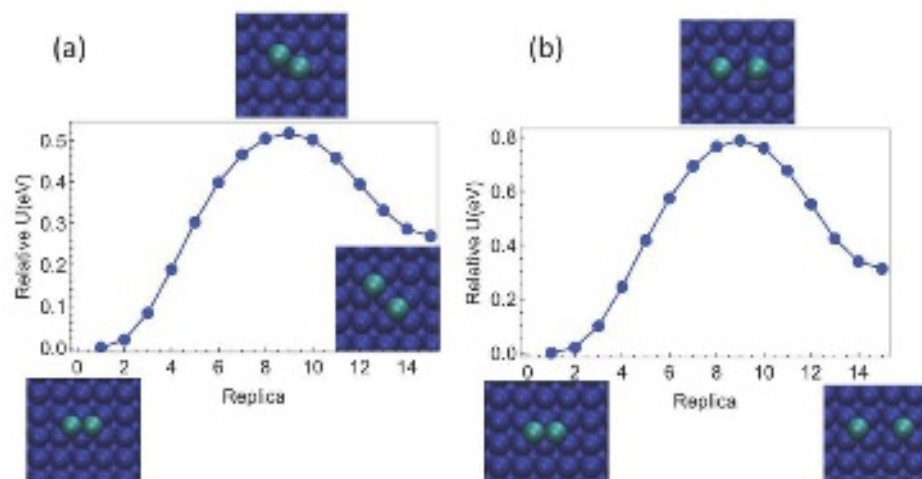


Figure 8. Potential-energy profiles obtained from 15 replicas in CI-NEB calculations for (a) twirling dimer dissociation, (b) linear dimer dissociation.

Event probabilities for dimer diffusion beginning with the intact dimer state are shown as a function of ΔV_{max} for different short time intervals in Figure 9 along with their fits to the sigmoid curve in Eq. (13). Each data point in Figure 9 is an average over 200 simulations. As for the previous cases, the sigmoid curve is an excellent fit to the data. Table III lists the values of ΔV_{max} obtained from $P'(\Delta V_{max})$ and $P''(\Delta V_{max})$, along with the time-boost factor B. While the ΔV_{max}

from $P'(\Delta V_{max})$ and $P''(\Delta V_{max})$ are smaller than both barriers to break the dimer from the intact configuration, these values are larger than the smallest barrier for dimer recombination after twirling dissociation in Figure 8(a) and larger than the barrier for which the physical rate deviates from the exact value in Figure 7.

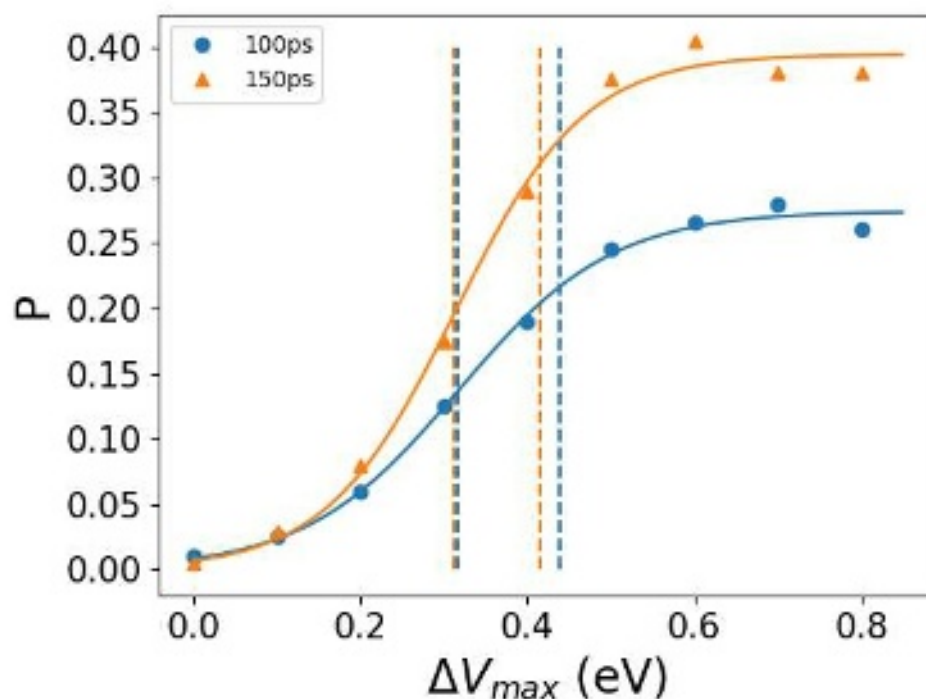


Figure 9. Event probabilities for dimer surface diffusion on Ag(100) in vacuum for different MD time intervals as a function of ΔV_{max} . The dashed lines indicate the values of ΔV_{max} summarized in Table III.

Table III. Values of optimal ΔV_{max} for various simulation times t_{MD} obtained from $P'(\Delta V_{max})$ and $P''(\Delta V_{max})$ for Ag dimer diffusion in Figure 9, along with the time-boost factors B .

t_{MD} (ps)	$P'(\Delta V_{max})$		$P''(\Delta V_{max})$	
	ΔV_{max}	B	ΔV_{max}	B
100	0.32	25	0.44	50
150	0.31	25	0.42	48

It is possible to isolate the recombination event in Figure 8(a) and determine an appropriate boost for this event. Beginning with the final state for twirling dissociation in Figure 8(a), we determined event probabilities for recombination of the dissociated dimer. The results are shown in Figure 10 where we again see an excellent fit to a sigmoid curve. The vertical lines in Figure 10 indicate the optimal values of ΔV_{max} obtained from $P'(\Delta V_{max})$ and $P''(\Delta V_{max})$. Based on the barrier for twirling dissociation in Figure 8(a) and the barrier for which the physical rate deviates from the exact value in Figure 7, we can see that these are safe values. Thus, applied to a single event, the bold boost estimates from the OptiBoost method can predict safe values for ΔV_{max} .

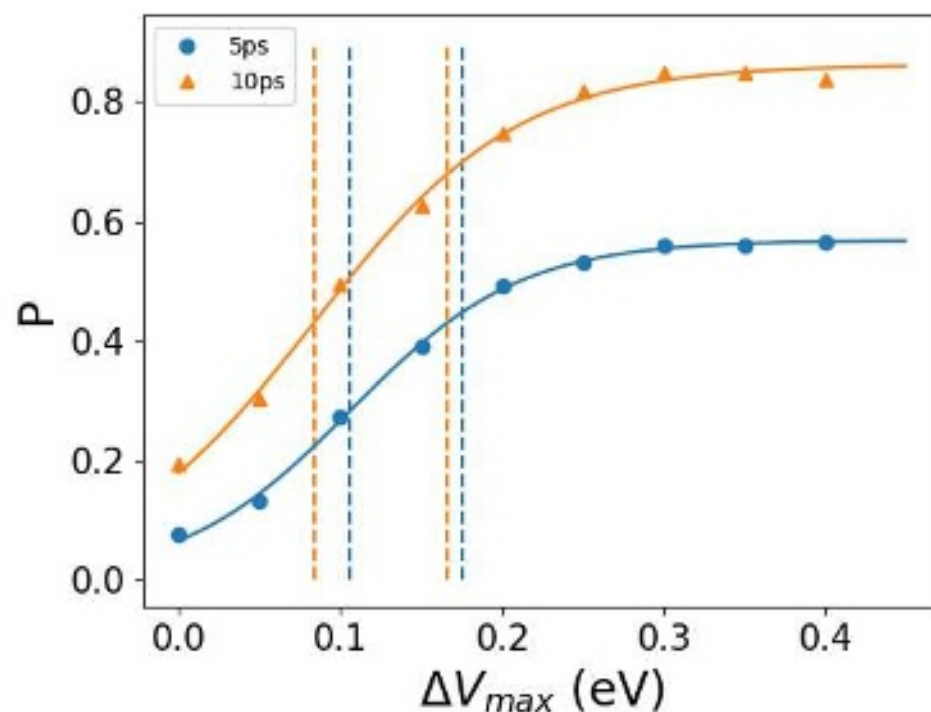


Figure 10. Event probabilities for twirling dimer dissociation on Ag(100) in vacuum for different MD time intervals as a function of ΔV_{max} . The dashed vertical lines delineate the boosts obtained from the maximum of $P'(\Delta V_{max})$ (0.08 and 0.11 eV) and from the minimum of $P''(\Delta V_{max})$ (0.17 and 0.18 eV).

For the dimer simulations, 10 min was required to obtain the 5 ps curve and the prediction range in Figure 10. The time-boost factors were $B = 3$ and $B = 10$ based on conservative and aggressive predictions, respectively. For regular MD simulations, it will take 129 hours to reach 1 microsecond in this system, running on 4 cores of our local facility. With the time-boost factor range of $B = 3$ -10 achieved in Figure 10, a one-microsecond simulation with OptiBoost will require 12-36 hours running on 4 cores using a boost that is safe.

Thus, HD simulations running with a minimal boost are up to an order of magnitude faster than regular MD. Here, we note that it is the current state of HD simulations to run with one boost that is set at the beginning of the simulation based on the lowest energy barrier. It is evident from the two examples for Ag in vacuum that such a mode of running would be sub-optimal since the boost determined for twirling dimer recombination (0.18 eV) is much less than that for single-atom diffusion (0.40 eV) or dimer dissociation (0.44 eV). Our calculations with the dimer indicate that it is possible to associate a particular boost with a particular local atomic environment. Thus, once the boosts have been established for various environments, HD simulations with the bond-boost method could run with high efficiency.

3. Ag Adatom Diffusion on Ag(100) in Solvent

Special concerns apply to HD simulations of Ag adatom diffusion in the system with EG solvent. Namely, the EG solvent is highly active so we must frequently update the bond list between Ag and EG and computational effort is wasted on quenching and re-bonding the system. If the bond list is not updated frequently, some EG bonded with the adatom will displace further than the bond cutoff and the bond will remain highly strained. This will result in a zero-bond bias and no acceleration ($B = 1$). Figure 11(a) depicts the typical way to group atoms in our HD simulations

of surfaces. In this set-up, the bottom two layers of Ag are fixed and the upper four layers of Ag along with the adatom(s) are simulated using HD. We used NVT simulations for EG solvent.

We ran test HD simulations with this protocol, using $\Delta V_{max} = 0.30$ eV and updating the bond list every 100 time steps. 20 simulations were run for 100 ps each and no events were detected. Moreover, 99% of the boosted bonds had zero boost and only 55% of the computational effort was spent on dynamics. Subsequently, 50 simulations were performed in which the bond list was updated every 10 time steps. Although no event was detected, the time-boost factor was $B \cong 10$, 40% of the boosted bonds had zero boost, but only 10% of the computational effort was on dynamics. By updating the bond list more frequently, larger boosts could be achieved, but a significant fraction of the computational effort was wasted on quenching and re-bonding.

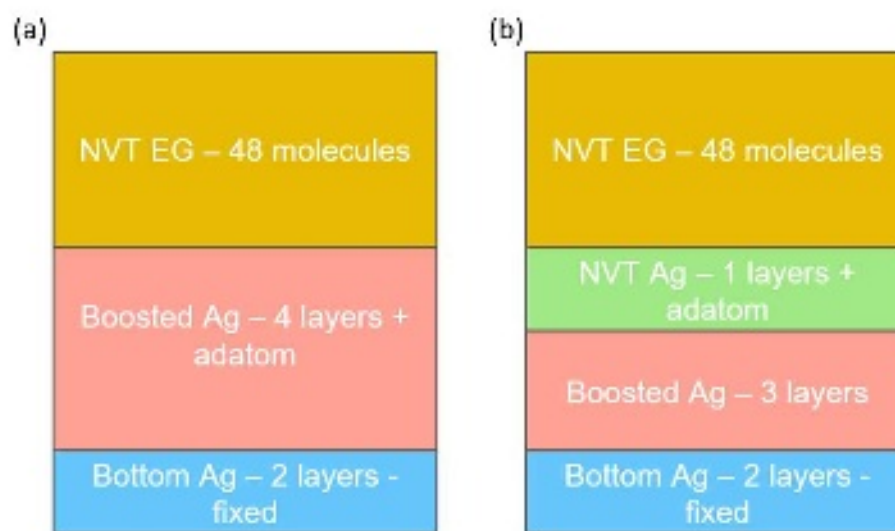


Figure 11. Two possible simulation protocols for HD simulations of surface events in EG solvent. (a) All the moving Ag atoms are simulated with HD and the solvent is simulated in the NVT ensemble. (b) Three layers of Ag atoms are simulated with HD, while the top Ag layer and the EG solvent are simulated in the NVT ensemble.

As an alternative to our standard protocol in Figure 11(a), we implemented the method in Figure 11(b). In this protocol, the top Ag layer and the adatom are run with NVT simulations (together with the EG), so that Ag-EG bonds are mostly excluded from the HD bond list. 200 HD simulations were performed for 100 ps each using the modified group method with $\Delta V_{max} = 0.30$ eV. The HD bond list was updated every 1000 time steps. In this protocol, an event occurred in one simulation out of nine. Only 10% of the boosted bonds had zero boost and over 90% of computational effort was spent on dynamics. With time-boost factors of $B \approx 80$, the modified group method can be applied to systems with a boundary between active and relatively inactive parts.

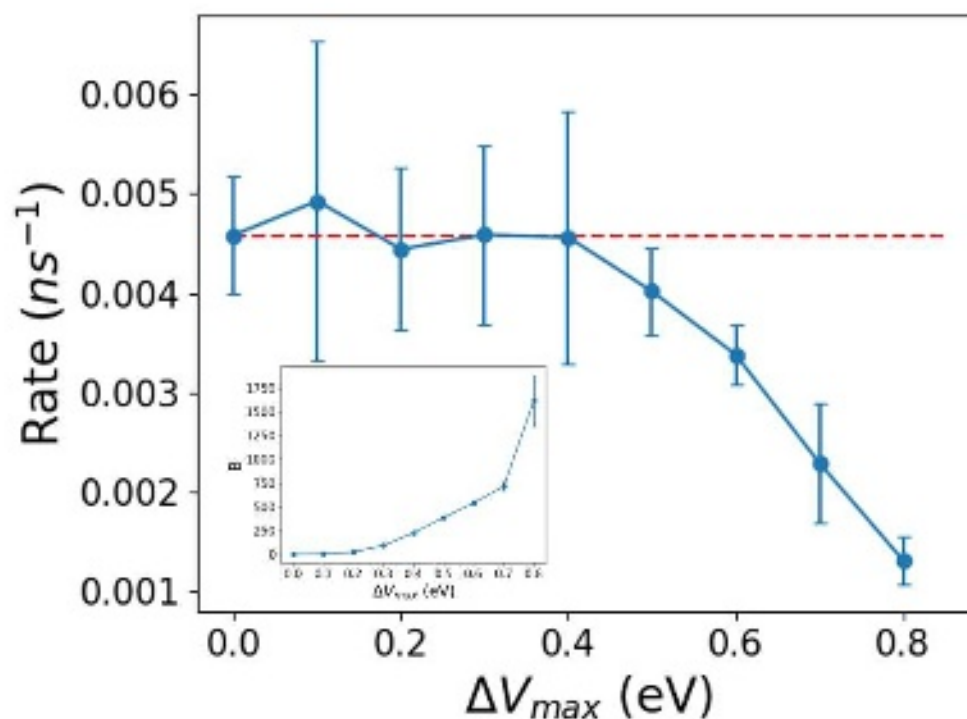


Figure 12. Net diffusion rate of Ag atom Ag(100) in EG solvent as a function of ΔV_{max} . The dashed line is the average rate for low values of ΔV_{max} . The inset shows the time-boost factor B as a function of ΔV_{max} .

In long-time simulations based on the method in Figure 11(b), the observed event rate as a function of ΔV_{max} is shown in Figure 12. Each data point in Figure 12 is an average over five simulations with times ranging from 40-200 ns. Interestingly, while single-atom hopping of the adatom on top of the surface occurred in vacuum, exchange diffusion, in which a surface atom and the adatom exchange places, was the preferred mechanism for a single Ag atom in solvent. We were able to determine that this was not a result of using the boosting protocol in Figure 11(b) because HD simulations of single-atom diffusion in vacuum still exhibited hopping using this protocol. Additionally, the lattice parameters were the same in vacuum and in solvent, so the exchange mechanism in the solvent systems did not result from strain. Comparing Figures 8 and 12, we see that Ag adatom diffusion in solvent environment is almost an order of magnitude slower than that in a vacuum environment. From Figure 12, we see the simulated rate becomes inaccurate for $\Delta V_{max} > 0.4$ eV.

To test the OptiBoost method, we conducted 200 HD simulations each for two different time intervals and various ΔV_{max} to obtain the event probabilities shown in Figure 13. In Figure 13, we see that the event probabilities for diffusion on a surface in EG solvent exhibit the same sigmoid form as the other systems. Using the OptiBoost analysis of $P'(\Delta V_{max})$ and $P''(\Delta V_{max})$, we find similar boost ranges as we saw previously: For the conservative analysis with $P'(\Delta V_{max})$, we find a safe boost of ~ 0.3 eV for both time intervals. Using the “bold” analysis from $P''(\Delta V_{max})$, we predict a safe, but aggressive boost of ~ 0.4 eV. From the inset to Figure 12, the time-boost factor for $\Delta V_{max} = 0.4$ eV is $B = 230$. Interestingly, even though the boosts are similar for Ag diffusion in a solvent and in a vacuum environment, B is about 40 times greater in the solvent environment. We attribute these differences to differences in the boosting mode for the solvent system [Figure 11(a) in vacuum vs. Figure 11(b) in solvent], which leads to a greater fraction of non-zero boosts.

Regarding the efficiency of these simulations, 80 h were required to obtain the 150 ps curve and the prediction range for ΔV_{max} . The time-boost factors were $B = 100$ and $B = 230$ based on conservative and aggressive predictions, respectively. For normal MD simulations, it would take 1316 hours to reach 1 microsecond in this system, running on 4 cores. With the time-boost factor range of 100-230 indicated by Figure 12, a simulation with OptiBoost will require 86-91 hours running on 4 cores using a safe boost – an acceleration of over an order of magnitude, even with the substantial overhead of establishing an appropriate value of ΔV_{max} . Without the overhead of determining ΔV_{max} , the acceleration could be two orders of magnitude.

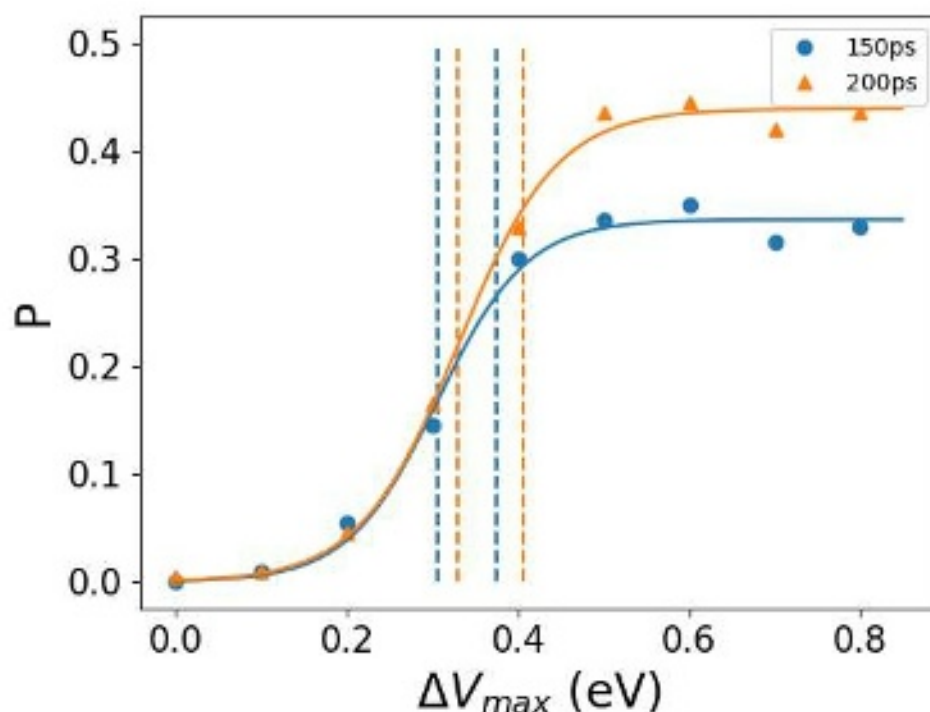


Figure 13. Event probabilities for Ag atom surface diffusion on Ag(100) in EG solvent for two different time intervals as a function of ΔV_{max} . The dashed vertical lines delineate the boost obtained from the maximum of $P'(\Delta V_{max})$ (0.31 and 0.33 eV) and the boost from the minimum of $P''(\Delta V_{max})$ (0.38 and 0.40 eV).

III. CONCLUSIONS

We developed a method called OptiBoost to adjust the value of ΔV_{max} in HD simulations based on the bond-boost method. The OptiBoost method involves (1) Selecting a value of ΔV_{max} and running a series of short trajectories to obtain the probability $P(\Delta V_{max})$ that an event occurred; (2) Fitting $P(\Delta V_{max})$ to a sigmoid function and; (3) Delineating the boost range from the maximum in $P'(\Delta V_{max})$ and from the minimum in $P''(\Delta V_{max})$.

We demonstrated the OptiBoost method in simulations on a cosine potential, where exact rates were known, and exact analysis could be performed. These results showed that the OptiBoost method could determine safe and aggressive boosts. We then implemented OptiBoost in three different simulations involving Ag atom diffusion on Ag(100): (1) An Ag adatom on Ag(100) in vacuum; (2) An Ag dimer on Ag(100) in vacuum and; (3) An Ag adatom on Ag(100) in EG solvent. We implemented the OptiBoost method for the Ag diffusion systems in the LAMMPS code, driven by a Python wrapper. All trajectory analysis for the Ag diffusion systems was performed in Python.

For all the Ag diffusion systems, we obtained similar results to the cosine potential, demonstrating the robust nature of OptiBoost. Even though OptiBoost incurs computational overhead, the increases in efficiency enabled by this approach were typically an order of magnitude. Since determination of the event probability is an “embarrassingly parallel” calculation, the overhead associated with this calculation would become negligible on a massively parallel architecture, allowing for more substantial acceleration. Additionally, there are several aspects of this algorithm that could be optimized, including the length of the run to determine the event probability and the range of ΔV_{max} values. While we did not optimize these aspects here, this would be a worthwhile future goal. Perhaps most importantly, our simulations with the dimer

showed that an appropriate boost can be established “once and for all” for each local atomic environment and recalled once the local configuration is re-visited, so that HD simulations can be run with multiple boosts, free of the overhead incurred to obtain the boost. In this mode of operation, HD simulations would be particularly efficient.

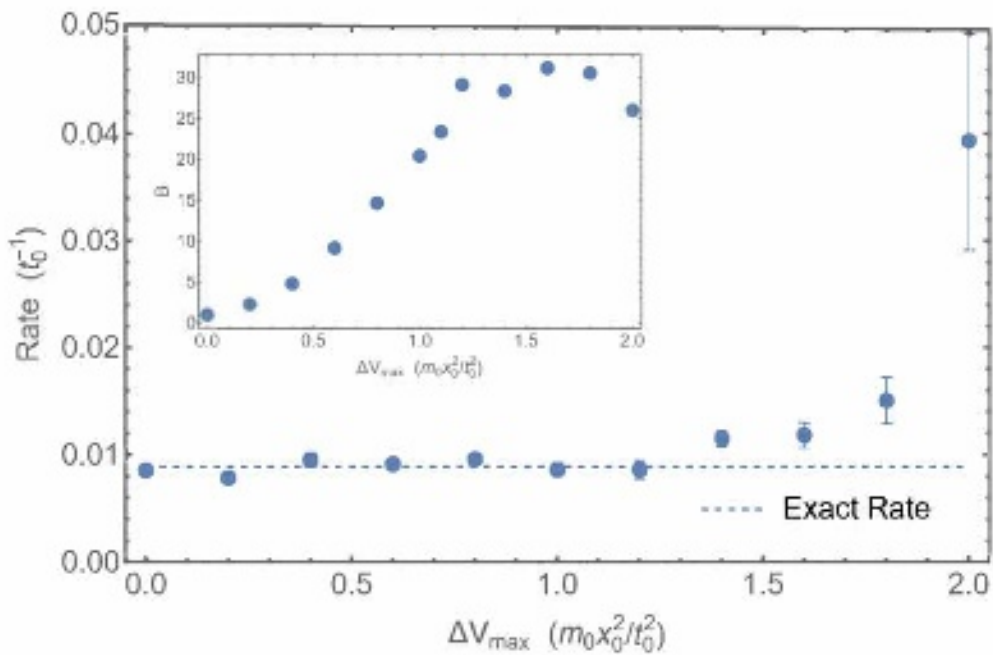
ACKNOWLEDGEMENTS

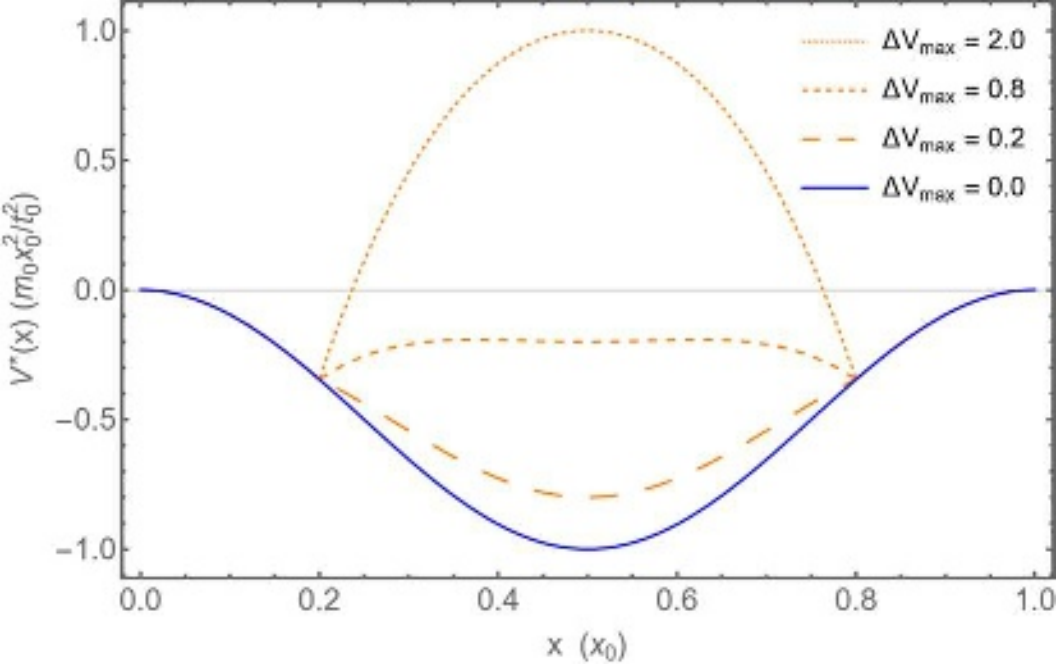
This work was funded by the National Science Foundation, Grant OAC-1835607.

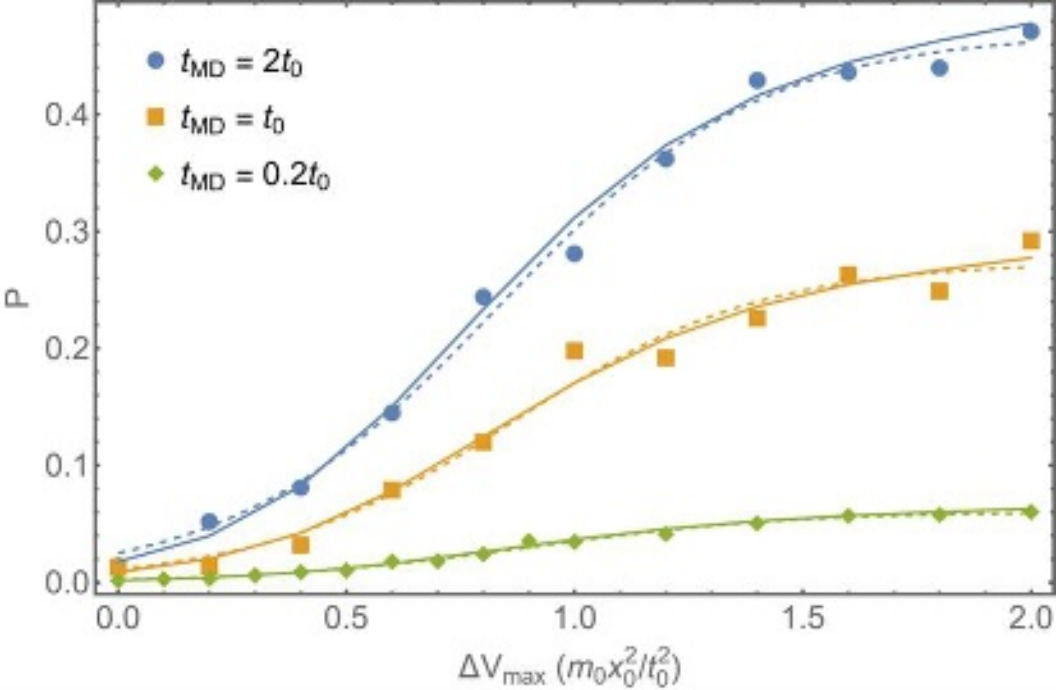
REFERENCES

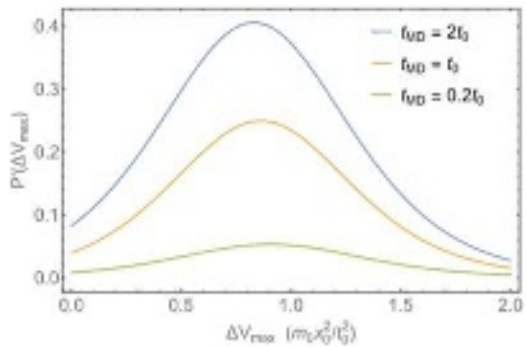
- 1 A. F. Voter, Journal of Chemical Physics 106 (1997) 4665.
- 2 A. F. Voter, Physical Review Letters 78 (1997) 3908.
- 3 M. R. Sorensen, and A. F. Voter, Journal of Chemical Physics 112 (2004) 9599.
- 4 A. F. Voter, Physical Review B 57 (1998) R13985.
- 5 D. Perez et al., Journal of Chemical Theory and Computation 12 (2016) 18.
- 6 D. Hamelberg, J. Mongan, and J. A. McCammon, Journal of Chemical Physics 120 (2004) 11919.
- 7 K. Ganeshan, M. J. Hossain, and A. C. T. van Duin, Molecular Simulation 45 (2019) 1265.
- 8 C. F. Abrams, and E. Vanden-Eijnden, Proceedings of the National Academy of Sciences 107 (2010) 4961.
- 9 R. A. Miron, and K. A. Fichthorn, Physical Review B 72 (2005)
- 10 K. E. Becker, M. H. Mignogna, and K. A. Fichthorn, Physical Review Letters 102 (2009) 046101.
- 11 R. A. Miron, and K. A. Fichthorn, Physical Review Letters 93 (2004) 128301.
- 12 K. A. Fichthorn et al., Journal of Physics-Condensed Matter 21 (2009) 084212.
- 13 C. Huang, D. Perez, and A. F. Voter, Journal of Chemical Physics 143 (2015) 074113.
- 14 S. Y. Kim, D. Perez, and A. F. Voter, Journal of Chemical Physics 139 (2013) 144110.
- 15 T. Cheng et al., Journal of the American Chemical Society 136 (2014) 9434.

- 16 S. J. Plimpton, D. Perez, and A. F. Voter, Journal of Chemical Physics 153 (2020) 054116.
- 17 R. A. Miron, and K. A. Fichthorn, Journal of Chemical Physics 119 (2003) 6210.
- 18 K. A. Fichthorn, and S. Mubin, Computational Materials Science 100 (2015) 104.
- 19 Y. Xia et al., Angewandte Chemie International Edition 48 (2009) 60.
- 20 X. Qi et al., Nano Letters 15 (2015) 7711.
- 21 X. Qi, and K. A. Fichthorn, Nanoscale 9 (2017) 15635.
- 22 S. J. Plimpton, Journal of Computational Physics 117 (1995) 1.
- 23 P. L. Williams, Y. Mishin, and J. C. Hamilton, Modelling and Simulation in Materials Science and Engineering 14 (2006) 817.
- 24 Y. Zhou, W. A. Saidi, and K. A. Fichthorn, The Journal of Physical Chemistry C 118 (2014) 3366.
- 25 O. Guvench et al., J Comput Chem 29 (2008) 2543.
- 26 K. Vanommeslaeghe et al., Journal of Computational Chemistry 31 (2010) 671.
- 27 I. Vorobyov et al., Journal of Chemical Theory and Computation 3 (2007) 1120.
- 28 Materials Handbook (Springer-Verlag, London, 2008), 2 edn.,
- 29 X. Qi et al., ACS Nano 13 (2019) 4647.
- 30 G. Henkelman, B. P. Uberuaga, and H. Jonsson, Journal of Chemical Physics 113 (2000) 9901.

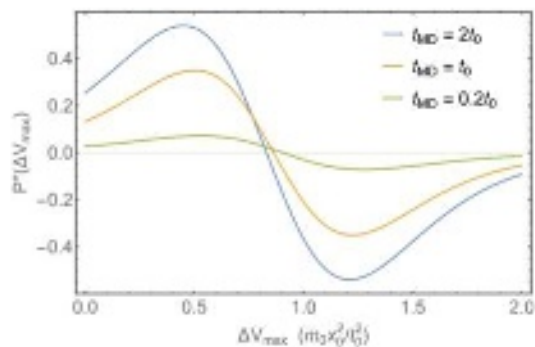




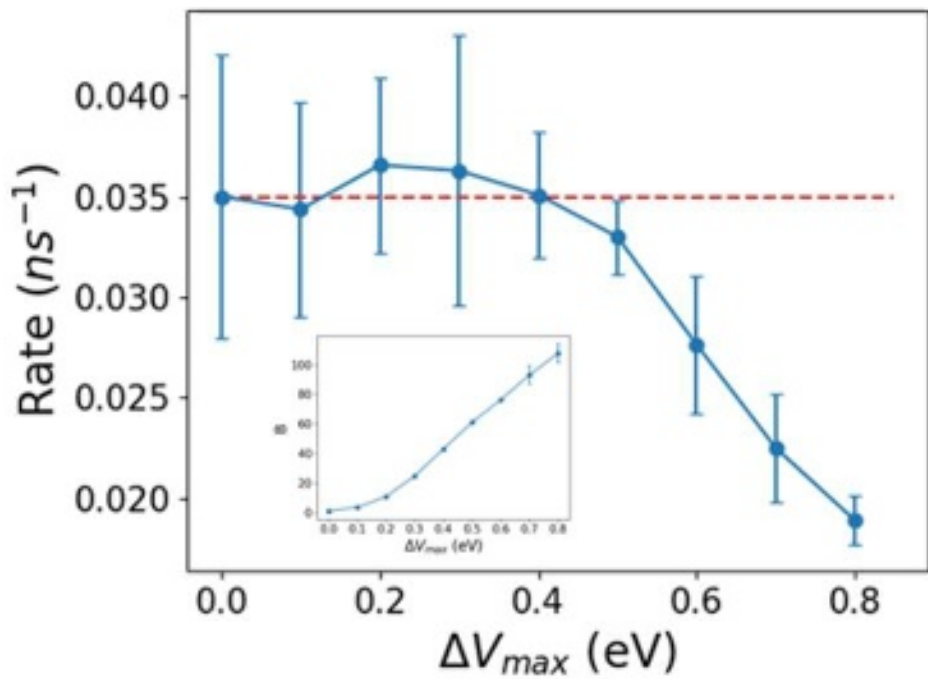


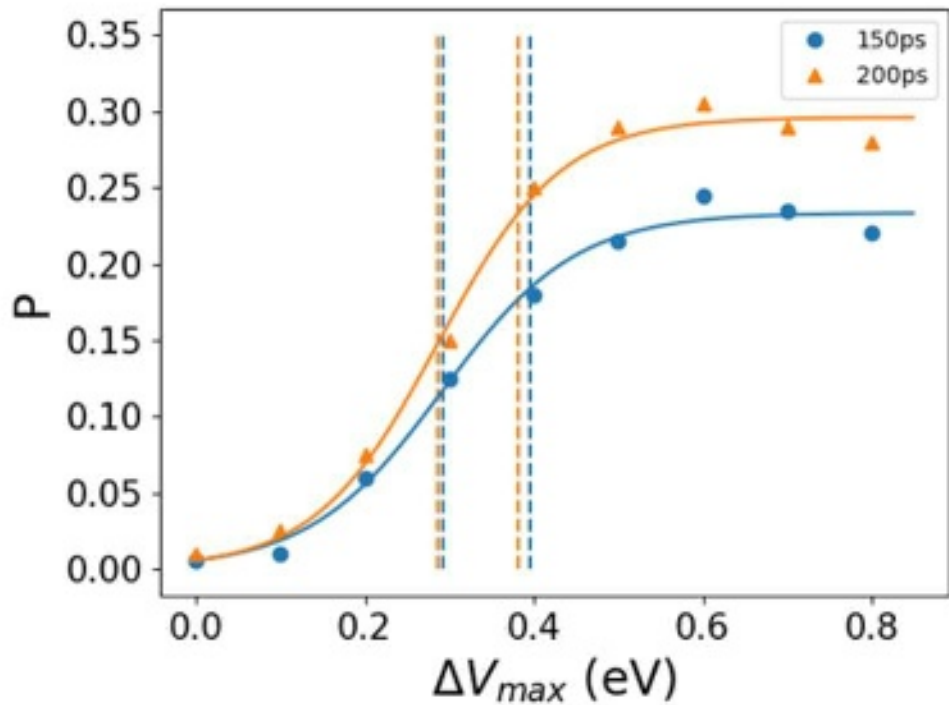


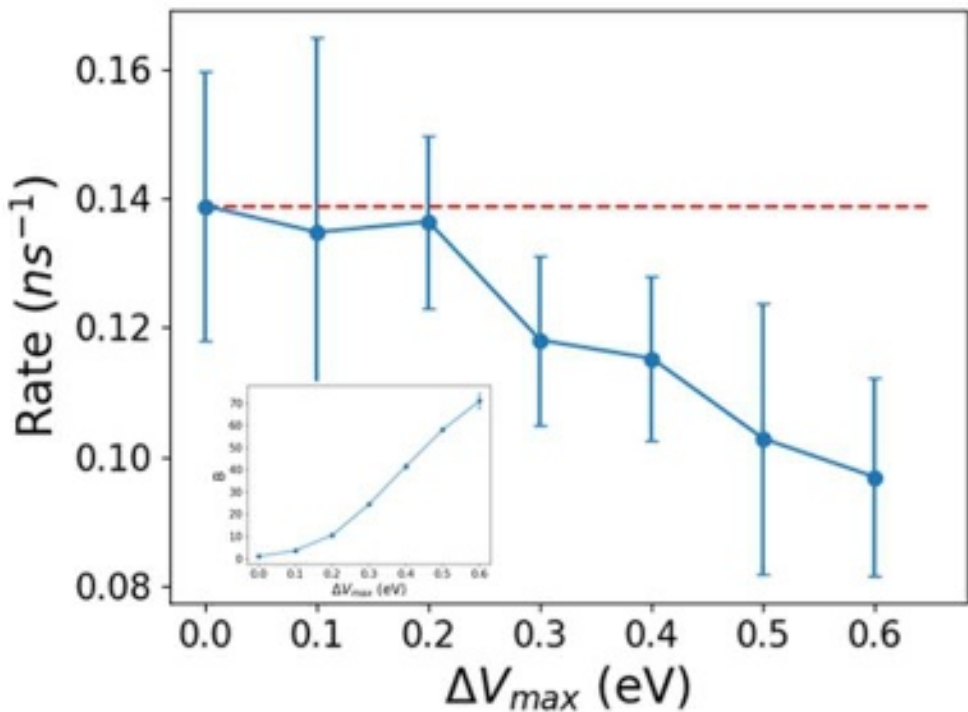
(a)



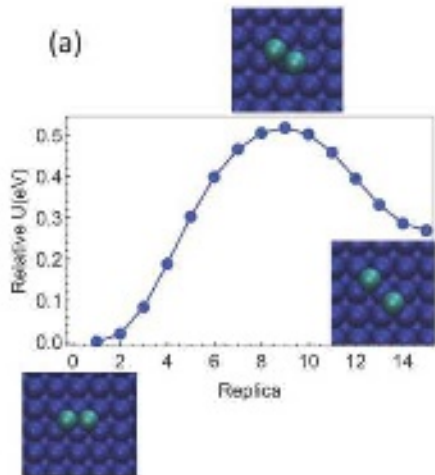
(b)



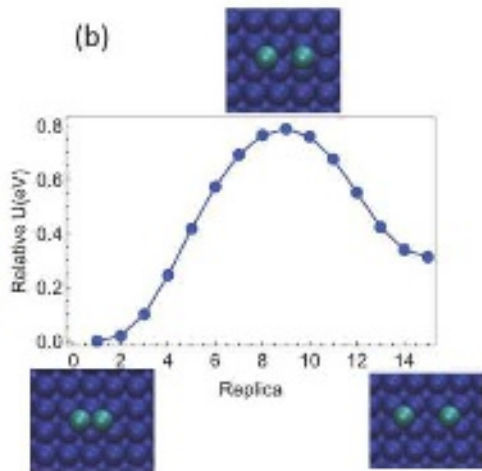


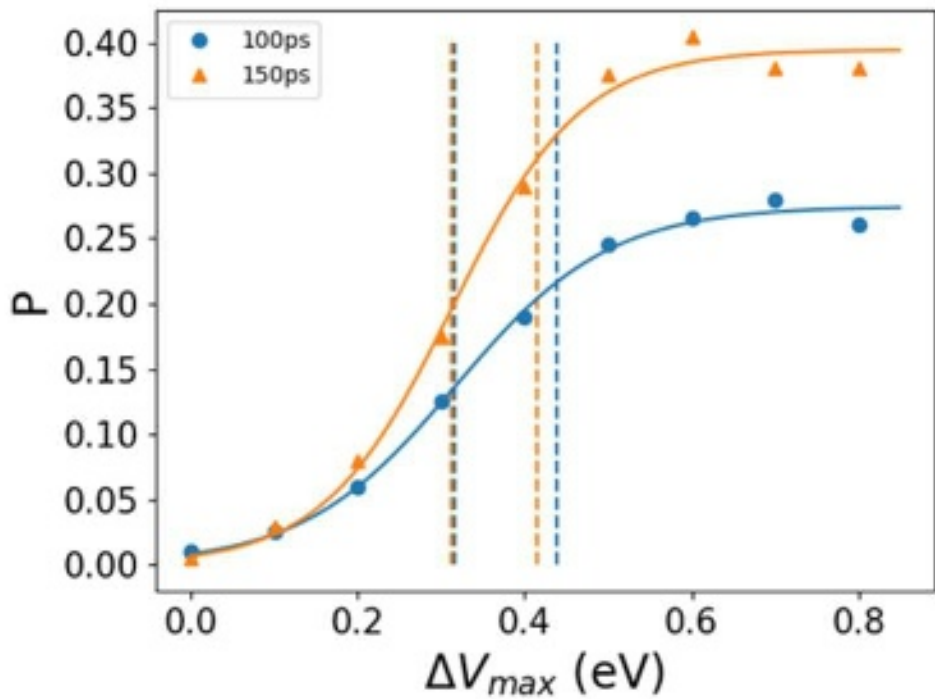


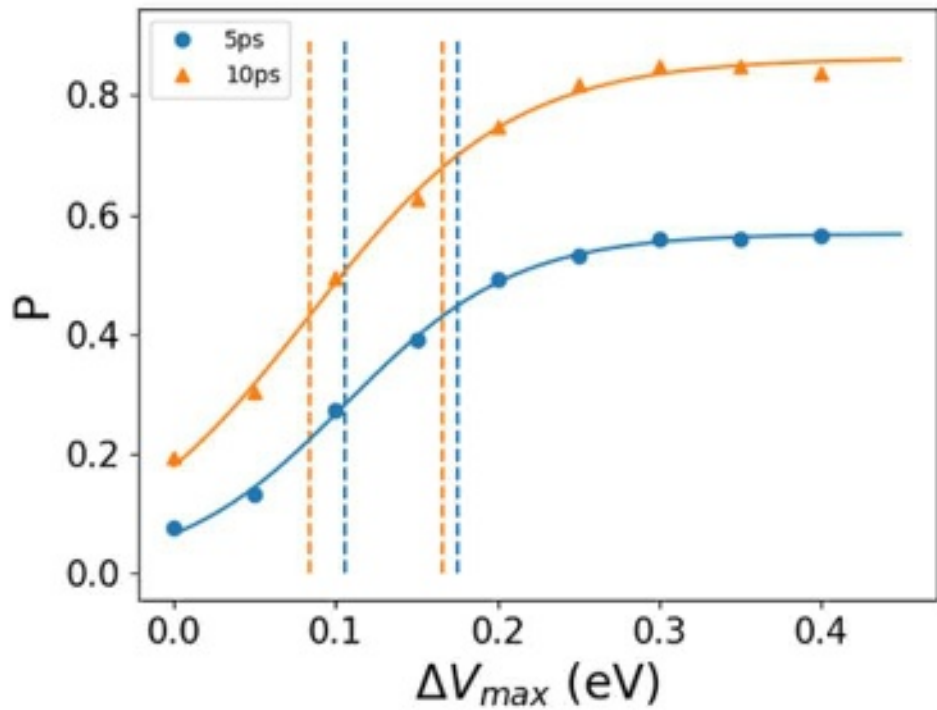
(a)



(b)







(a)



(b)



