ADVANCED SCIENCE Open Access
www.advancedscience.com

Reinforcement Learning-Guided Long-Timescale Simulation of Hydrogen Transport in Metals

*Hao Tang, Boning Li, Yixuan Song, Mengren Liu, Haowei Xu, Guoqing Wang, Heejung Chung, and Ju Li**

Diffusion in alloys is an important class of atomic processes. However, atomistic simulations of diffusion in chemically complex solids are confronted with the timescale problem: the accessible simulation time is usually far shorter than that of experimental interest. In this work, long-timescale simulation methods are developed using reinforcement learning (RL) that extends simulation capability to match the duration of experimental interest. Two special limits, RL transition kinetics simulator (TKS) and RL low-energy states sampler (LSS), are implemented and explained in detail, while the meaning of general RL are also discussed. As a testbed, hydrogen diffusivity is computed using RL TKS in pure metals and a medium entropy alloy, CrCoNi, and compared with experiments. The algorithm can produce counter-intuitive hydrogen-vacancy cooperative motion. We also demonstrate that RL LSS can accelerate the sampling of low-energy configurations compared to the Metropolis–Hastings algorithm, using hydrogen migration to copper (111) surface as an example.

1. Introduction

Diffusive atomic motion is an essential microscopic process in the kinetic theory of materials.[1,2] Various interesting phenomena and applications are rooted in diffusion-related processes, from the interdiffusion at metal interfaces, vacancy and void formation, to hydrogen embrittlement[3] and resistance switching in oxide memristors.[4] To investigate the diffusion process, atomic simulation[5,6] is often used to uncover atomic interactions behind a wide range of materials phenomena.[2,7] However, a critical challenge of atomistic simulation of diffusion-related process is the timescale problem:[8] the atomic vibration has a timescale of picoseconds, this limits the maximal time step that can be used in atomistic simulations; however, the diffusion-related transitions between adjacent energy minima have orders of magnitude longer timescale. That is because the energy barriers on the diffusion pathway slow down the diffusion process.[2] The timescale problem limits most of the straightforward molecular dynamics simulations to nanoseconds, which fall short of the timescales relevant to many diffusion-related phenomena.[8,9] Therefore, different methods are needed to deal with the long-timescale problem.[8]

Our work will be compared with one of the widely studied algorithms, the kinetic Monte Carlo (KMC) method,[10] where one directly works with diffusion timescale without explicitly showing the vibration timescale motion. Traditional KMC (in contrast with off-lattice KMC) requires energy minima and transition pathways (the so-called event table) as input, and the method chooses transition events according to the transition rates in the event table. However, as the diffusion pathway is sometimes counter-intuitive, correctly determining the necessary input information of KMC is not a trivial task.[10] To conduct a simulation without a known event table, the off-lattice KMC is developed.[11] The algorithm conducts saddle-point searches to obtain the diffusion pathways along with the KMC simulation. Another method reported to have advantageous efficiency is temperature accelerated dynamics (TAD), where the transition pathways are explored by high-temperature molecular dynamics.[12] In both methods, the transition pathway is explored by random sampling (random initial guess in the saddle-point search for off-lattice KMC, and random thermal motion for TAD). However, as the configuration space is high dimensional, it requires a large amount of random sampling to ensure that the correct transition pathway is obtained, which limits the simulation system size and accessible timescale.[11]

In this work we developed a reinforcement learning (RL) based method that guides the transition pathway sampling on

H. Tang, Y. Song, M. Liu, H. Chung, J. Li
Department of Materials Science and Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
E-mail: liju@mit.edu
B. Li, G. Wang
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
B. Li
Department of Physics
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
H. Xu, G. Wang, J. Li
Department of Nuclear Science and Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/advs.202304122

DOI: 10.1002/advs.202304122

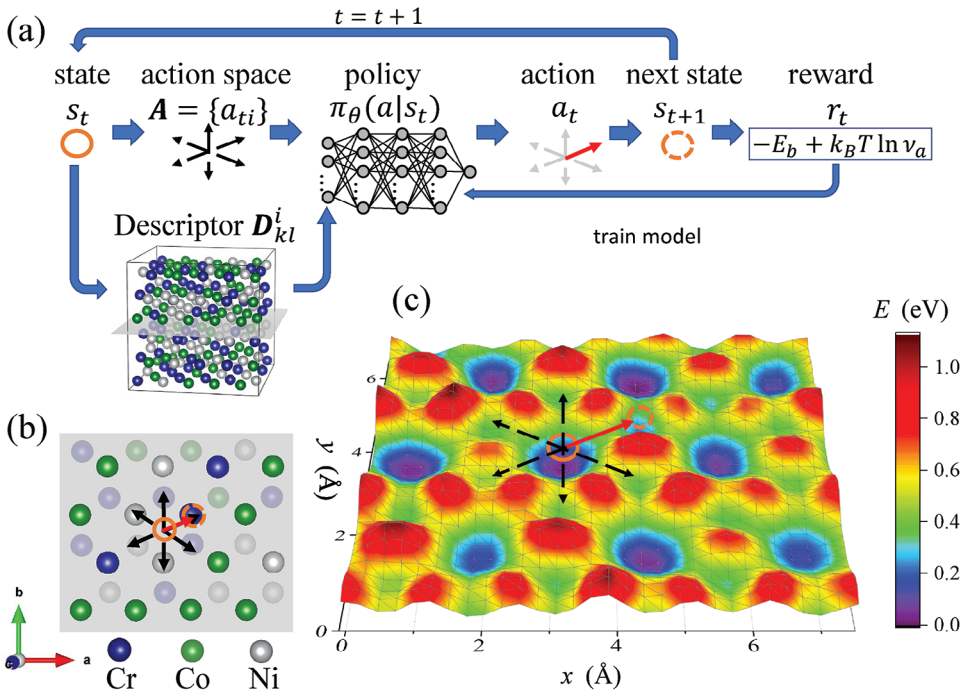

Figure 1. a) Computational workflow of the RL long-timescale method illustrated on b) hydrogen diffusion in CrCoNi medium entropy alloy. The blue, green, and grey spheres represent Cr, Co, and Ni atoms, respectively. The orange circle, black dashed arrow, and red arrow represent state, action space, and selected action, respectively. c) The potential energy landscape of a hydrogen atom on the grey planes in (a, b). When calculating the energy, surrounding atoms and the *z*-coordinate of the hydrogen atoms are relaxed.

chemically complex potential energy surface (PES). Instead of searching for all nearby saddle points along randomly sampled initial directions,[11] we use parameterized neural network model to predict the direction of atomic motion that yields the high-probability transition pathway, based on learning from the outcomes of rigorous PES minimum energy path (MEP) searches, resulting in a data superstructure of reduced-dimension "transition energy landscape" (TEL). The neural network based TEL avoids the repeated saddle-point searches, which is the most significant contributor to the computational cost of the off-lattice KMC. We demonstrate that our RL model can either simulate physical time diffusional trajectories (RL TKS) or sample low-energy configurations (RL LSS) efficiently, in various hydrogen diffusion phenomena in alloy bulk or near surfaces.

2. Results

2.1. General Framework

2.1.1. Computational Workflow

Our RL method is illustrated in **Figure 1a**. The PES has a large number of local minima separated by transition energy barriers. In this paper, we use hydrogen diffusion in face-centered cubic (FCC) alloys as an example, as shown in Figure 1b. In the local energy minimum configurations of FCC bulk structures, hydrogen atoms can reside in octahedral and tetrahedral interstitial sites shown as the deep blue and shallow green potential wells in Figure 1c, and the octahedral site corresponds to a lower-energy configuration. The energy landscape is provided by a universal neural network interatomic potential (which evaluates the total energy and forces on atoms for a given atomic configuration), the PreFerred Potential (PFP),[13,14] throughout this paper. Beginning from a given local energy minimum configuration or "state" $s_t \equiv (\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_N)$ (the orange circles in Figure 1, where $\vec{r}_i$ are the coordinates of the ith atom), a set of possible transition displacements $\{a_{ti}\}$ (also called "actions") are first identified. In our problem, we first identify the polyhedron formed by the nearest-neighbor metal atoms of each interstitial hydrogen. Possible actions are then defined by moving the hydrogen atom through the face centers of the polyhedron (See Section 4.1 for details).

In the next step, an action a_t is selected from the action space $\mathcal{A}_{s_t} \equiv \{a_{ti}\}$. The probability of selecting each action a given current state s_t is defined as $\pi_\theta(a|s_t)$, which is given by the Boltzmann policy based on a neural network value function $Q_\theta(s_t, a)$.[15]

$$\pi_\theta(a|s_t) = \frac{e^{Q_\theta(s_t,a)/k_B T}}{\sum_{a' \in \mathcal{A}_{s_t}} e^{Q_\theta(s_t,a')/k_B T}} \quad (1)$$

where $Q_\theta(s, a)$ represents the expected reward that can be obtained by taking a specific action a in a particular state s, and the actions with higher expected rewards are more likely to be selected (Equation (1)). The Boltzmann form reflects the randomness caused by temperature effects. θ represents the neural-network model parameters, k_B and T are the Boltzmann constant and physical temperature. We note that $Q_\theta(s, a)$ itself can also depend on T if the vibrational entropy contribution is considered, which will be discussed later. While Equation (1) looks very similar to the well-known KMC dynamics, the meaning of $Q_\theta(s_t, a)$ is quite different, in that $Q(s_t, a)$ reflects not only information about

the present (i.e., the present-step forward energy barrier), but also "future returns" that involve *future* energy barriers and/or *future* free-energy reductions, in a combination that is to be detailed later. Q_θ is neural network fitting of Q, where θ stands for the set of neural network parameters, which is the convention in this paper. Because of this conceptual distinction with KMC, the "dynamics" generated by Equation (1) is not guaranteed yet to be the real physical timescale according to transition-state theory (TST), as KMC aims to. And so the time label t in s_t, a_t above is an integer, and not the real time yet.

After selecting an action $a_t = (i, \vec{v})$ (both i and $\vec{v}$ are determined by a_t), the ith atom is displaced by vector $\vec{v}$ across the energy barrier. The system is then relaxed to the next state, s_{t+1}, using the MDMin algorithm implemented in the Atomistic Simulation Environment.[16] Parameters of the transition, including the transition energy barrier E_b^{NEB}, the attempt frequency ν_a^{label}, and the energy change after the transition ΔE, can then be computed using PFP. The transition saddle point is obtained by the nudged elastic band (NEB) method[17] by setting s_t and s_{t+1} as the initial and final points, to provide the ground truths for neural network training. The reward function of this transition step, r_t, the key concept in RL, can take different forms depending on the goal of the RL dynamics (specified in Equation (1)) which will be detailed later. The whole simulation trajectory is produced by repeating the above scheme that generates the next state according to the current state.

2.1.2. Model Architecture of $Q_\theta(s, a)$

The $Q_\theta(s, a)$ model is constructed based on the DeepPot-SE subnetworks.[18] As the atomic interaction in alloys is short ranged, we assume $Q_\theta(s, a = (i, \vec{v}))$ is a function of the atomic environment of the moved atom i and its displacement vector $\vec{v}$. The function $Q_\theta(s, a)$ should be invariant under translation, rotation, and permutation operations on the atomic system. Therefore, we define an atomic descriptor $\mathcal{D}^i$, which is a re-formalization of s and a, that is invariant under all symmetry operations. $\mathcal{D}^i$ can be realized by the following construction.

$$\tilde{R}^i = \begin{bmatrix} \hat{\vec{r}}_{i1}\cdot\hat{\vec{r}}_{i1} & \cdots & \hat{\vec{r}}_{i1}\cdot\hat{\vec{r}}_{iM} & \hat{\vec{r}}_{i1}\cdot\vec{v} \\ \hat{\vec{r}}_{i2}\cdot\hat{\vec{r}}_{i1} & \cdots & \hat{\vec{r}}_{i2}\cdot\hat{\vec{r}}_{iM} & \hat{\vec{r}}_{i2}\cdot\vec{v} \\ \vdots & \vdots & & \vdots \\ \hat{\vec{r}}_{iM}\cdot\hat{\vec{r}}_{i1} & \cdots & \hat{\vec{r}}_{iM}\cdot\hat{\vec{r}}_{iM} & \hat{\vec{r}}_{iM}\cdot\vec{v} \\ \vec{v}\cdot\hat{\vec{r}}_{i1} & \cdots & \vec{v}\cdot\hat{\vec{r}}_{iM} & |\vec{v}|^2 \end{bmatrix} \tag{2}$$

$$\mathcal{D}^i_{kl} = \sum_{m,n=1}^{M+1} G^1_k(f_c(r_{im}), c_m)\tilde{R}^i_{mn}G^2_l(f_c(r_{in}), c_n) \tag{3}$$

where $\hat{\vec{r}}_{ij} \equiv \frac{f_c(r_{ij})\vec{r}_{ij}}{r_{ij}}$, $\vec{r}_{ij} \equiv \vec{r}_j - \vec{r}_i$, $r_{ij} \equiv |\vec{r}_{ij}|$, $j = 1, 2, \cdots, M$ goes through all atoms around the ith atom within a cut-off radius r_c. The summation in Equation (3) goes from 1 to $M + 1$ to go through all rows/columns in the matrix in Equation (2). $f_c(r)$ is a cut-off function as defined in ref. [18], which goes smoothly to zero at a cut-off radius r_c, and G^1_k and G^2_l are embedding neural networks parametrized by θ_{emb}. $c_m (m = 1, 2, \cdots, M)$ are the atomic species of the mth atom. To embed the action (the last row and column in Equation (2)) into the descriptor, we set c_{M+1} as a unique "action species." The descriptor $\mathcal{D}^i$ is invariant under all symmetry operations. The descriptor is then flattened to a vector and passed to a multilayer perceptron (MLP) that outputs the Q function: $Q_\theta(s, a = (i, \vec{v})) = \mathrm{MLP}_{\theta_{fit}}(\mathcal{D}^i(\theta_{emb}))$, where the model parameters $\theta = (\theta_{fit}, \theta_{emb})$ include both parameters of the MLP θ_{fit} and that of the embedding network θ_{emb} (see Section S1, Supporting Information for detailed parameter settings).

By choosing different reward functions r_t (whose accumulated form becomes Q in standard RL formalism, for Equation (1)), our method has at least two working modes: transition kinetics simulator (TKS) and low-energy states sampler (LSS). RL TKS aims to simulate physical transition rates according to the HTST (thus is in principle identical to KMC, just with neural network estimators, and so the t label aims to lead to real physical time also), while the LSS aims to converge to global energy minimum configurations and is thus similar to an energy annealer, where the t is fictitious. In the next two sections we will introduce the two special options, RL TKS and RL LSS, separately. In Section 4.3, we will also discuss the mathematical and physical meaning of general RL (finite α, β, γ), away from the TKS ($\alpha = 1$, $\beta = 0$, $\gamma = 0$) and LSS ($\alpha = 0$, $\beta = 1$, $\gamma \approx 1$) corners in parameter space.

2.2. RL Transition Kinetics Simulator Option

TKS adopts the reward function of

$$r_t^{TKS} \equiv -E_b^{NEB} + k_B T \ln \nu_a^{label}, \quad \nu_a^{label} = \frac{\prod_{i=1}^{3M}\nu_i}{\prod_{j=1}^{3M-1}\nu_j^*} \tag{4}$$

where the E_b^{NEB} and ν_a^{label} are obtained using the PFP on the fly during the neural network training. The attempt frequency ν_a^{label} in our training data is evaluated using ν_i and ν_j^*, the ith normal mode vibrational frequency at state s_t and the jth positive vibrational frequency at the PES saddle point between s_t and s_{t+1}. (The first-order saddle point has one non-positive mode, so j only runs from 1 to $3M - 1$ excluding the non-positive mode, while i runs from 1 to $3M$.) We calculate ν_i and ν_j^* by diagonalizing the force constant matrix evaluated by the PFP for the M atoms within a cut-off radius $r_c = 4$ Å from the atom displaced by the action.

The model is trained as a contextual bandit problem,[19] where the value function $Q_\theta^{TKS}(s_t, a_t)$ is trained to fit the instantaneous reward r_t^{TKS} (minimizing $\langle(Q_\theta^{TKS}(s_t, a_t) - r_t^{TKS})^2\rangle$). This means that we have set the RL discount-rate $\gamma = 0$, so the RL value function Q is no longer cumulative or inclusive of the future rewards and cares only about the present-step immediate reward. Furthermore, this immediate reward is taken to be just the vibrational free-energy barrier of the forward transition in harmonic transition state theory (HTST), which means TKS is just a version of KMC, but with neural network learning for acceleration.

To associate an action with a physical time scale, the transition rate can be evaluated by Q_θ^{TKS} according to HTST:

$$\Gamma_{s_t a} = \nu_a^{label} e^{-E_b^{NEB}/k_B T} = e^{r_t^{TKS}/k_B T} \simeq e^{Q_\theta^{TKS}(s_t,a)/k_B T} \tag{5}$$

Equation (1) then gives the physical branching ratio (probability) $P(a|s_t) = \Gamma_{s_t a}/\sum_{a' \in \mathcal{A}_{s_t}} \Gamma_{s_t a'}$. The average residence time of the

system on the state s_t, $\langle \Delta\tau \rangle = (\sum_{a\in\mathcal{A}_{s_t}} \Gamma_{s_t a})^{-1}$, can be estimated using the Q_θ^{TKS} functions as

$$\tau = 1/\sum_{a\in\mathcal{A}_{s_t}} e^{Q_\theta^{\mathrm{TKS}}(s_t,a)} \tag{6}$$

Note that the t symbol is an integer in RL, counting the number of state transitions, so t is not the physical time, which are denoted by τ in this paper instead.

Then, we make the model applicable to different temperatures. Expressing the reward $r_t^{\mathrm{TKS}} = r_t^0 + r_t^1 T$ as a linear function of T, the zeroth-order term r_t^0 and the linear term r_t^1 can be fitted simultaneously by a two-component value function (Q_θ^0, Q_θ^1) in $Q_\theta^{\mathrm{TKS}} = Q_\theta^0 + Q_\theta^1 T$.

$$\theta \leftarrow \theta - \lambda\nabla_\theta \sum_t \left[(Q_\theta^0(s_t,a_t) - r_t^0)^2 + T_{\mathrm{tr}}^2 (Q_\theta^1(s_t,a_t) - r_t^1)^2\right] \tag{7}$$

where λ is the learning rate, and T_{tr}, the training temperature, is a hyperparameter that controls the relative importance of the two terms in the loss function (which does not need to be the physical temperature in simulations). By introducing temperature into Q_θ^{TKS}, Q_θ^0, and Q_θ^1 give neural network predictions for the energy barrier $E_{\mathrm{b}}^{\mathrm{NN}} \equiv -Q_\theta^0$ and attempt frequency $\log \nu_{\mathrm{a}}^{\mathrm{NN}} \equiv \frac{Q_\theta^1}{k_{\mathrm{B}}}$.

As a testbed, we first apply RL TKS to hydrogen diffusion in pure FCC Cu and Ni. The model is trained on a $4\times4\times4$ cubic supercell with four randomly sampled hydrogen sites. The model is then deployed to simulate a single hydrogen diffusion in a $3\times3\times3$ cubic supercell for 500 timesteps. This system is simulated by RL TKS at temperatures spanning 250 to 500 K with an interval of 50 K and repeated 50 times for each temperature. The time-dependent squared displacements $\Delta x_j^2(\tau)$ and temperature T_j of each (jth) simulation trajectory are recorded. The diffusivity $D(T)$ for each given temperature T is extracted from the linear fitting $\langle \Delta x_j^2(\tau)\rangle_{T_j=T} = 6D(T)\tau$. The two parameters D_0 and E_{a} in the Arrhenius form of diffusivity $D = D_0 e^{-E_{\mathrm{a}}/k_{\mathrm{B}}T}$ are derived from the $\ln D(T) = \ln D_0 - \frac{E_{\mathrm{a}}}{k_{\mathrm{B}}}\frac{1}{T}$ fit.

First, we checked that our RL TKS are consistent with the simulation results of traditional KMC using the same PFP interatomic potential, as shown in **Table 1**. This validates our RL methods in estimating hydrogen diffusivity in metals. The derived D_0 and E_{a} are also reasonably consistent with the related experimental measurements. The effective activation energy E_{a} in simulation tends to be slightly smaller than the experimental results for multiple reasons. First, the PFP machine learning potential we used tends to slightly underestimate the energy barrier. For example, in FCC copper, the diffusion energy barriers in O →T/T → O (O → T means from octahedral to tetrahedral, and T → O means the reverse process) transition are 0.32/0.12 eV using the PFP compared with 0.36/0.14 eV in the DFT calculations.[20] Second, the quantum tunneling effects in H diffusion can further influence the activation barrier, which is not considered in our classical dynamics calculations. The O → T activation barrier in FCC copper considering quantum tunneling is estimated as 0.40 eV at 300 K using the path-integral Monte Carlo method, 0.04 eV higher than that without considering the quantum tunneling effect.[20] Therefore, our calculation here slightly underestimate activation energies. As a PES transition dynamics sampling algorithm, our RL method is compatible with different methods to estimate activation barriers. Using the DFT or PIMD method to calculate activation barriers in our training dataset will improve the prediction accuracy.

Table 1. RL TKS hydrogen self-diffusion simulation results in pure copper, pure nickel, and CrCoNi medium entropy alloy. $\Delta E_{\mathrm{b}} \equiv \sqrt{\langle (E_{\mathrm{b}}^{\mathrm{NN}} - E_{\mathrm{b}}^{\mathrm{NEB}})^2\rangle}$ and $\Delta\nu_{\mathrm{a}} \equiv \sqrt{\langle (\ln \nu_{\mathrm{a}}^{\mathrm{NN}} - \ln \nu_{\mathrm{a}}^{\mathrm{label}})^2\rangle}$ are the validation error of model prediction on transition energy barrier and attempt frequency. The activation energy Q and coefficient D_0 are fitted by reinforcement-learning-simulated diffusivity $D = D_0 e^{-E_{\mathrm{a}}/k_{\mathrm{B}}T}$ using maximal-likelihood estimation, and D_0^{exp} and $E_{\mathrm{a}}^{\mathrm{exp}}$ are the values from previous experiments.

	Cu	Ni	CrCoNi
ΔE_{b} (eV)	0.020	0.022	0.037
$\Delta \ln \nu_{\mathrm{a}}$	0.09	0.12	0.12
D_0 (10^{-7} m^2/s)	3.6	3.1	5
E_{a} (eV)	0.30	0.33	0.43
D_0^{KMC} (10^{-7} m^2/s)	3.3	2.8	–
$E_{\mathrm{a}}^{\mathrm{KMC}}$ (eV)	0.31	0.32	–
D_0^{exp} (10^{-7} m^2/s)	3.69[21]	0.15–6.98[22]	–
	21.1[23]	1.1–6.87[24]	–
	17.4[25]		
$E_{\mathrm{a}}^{\mathrm{exp}}$ (eV)	0.38[21]	0.31–0.44[22]	–
	0.46[23]	0.37–0.44[24]	–
	0.435[25]		

To test the method's capability to capture chemical complexity, we train the RL model on equiatomic CrCoNi medium entropy alloy. The CrCoNi alloy has recently attracted interest because of its outstanding fracture toughness and ductility.[26] In the CrCoNi solid solution, each metal atom near the hydrogen can be of different atomic species, giving a complex atomistic state space. The predicted $E_{\mathrm{b}}^{\mathrm{NN}}$ and $\nu_{\mathrm{a}}^{\mathrm{NN}}$ are approximately consistent with the values in the training and testing datasets, as shown in **Figure 2**, where the data points are distributed close to the diagonal line in the wide range of observed quantities. The standard deviation errors of the model predictions are close in training and testing datasets, confirming that the training data is not overfitted despite the large number of neural network parameters.

The hydrogen self-diffusion in CrCoNi is simulated using RL TKS running on one hydrogen in a $4\times4\times4$ rhombohedral supercell with short-range ordering (SRO) obtained from ref. [27]. The hydrogen squared displacement as a function of the RL TKS simulation time is shown in **Figure 3**a under 300 K using 30 repetitions of μs long-timescale simulations. An approximate function form of $\langle \Delta x^2\rangle \propto \tau$ is shown by the blue line, and the diffusivity is estimated to be 2.84×10^{-14} m^2 s^{-1}. Similar simulations are implemented for different temperatures, as shown in Figure 3b. The Arrhenius plot shows a good linear relation. The estimated effective activation energy E_{a} equals 0.43 ± 0.01 eV, and the pre-exponential factor D_0 equals $5\pm2\times10^{-7}$ m^2 s^{-1}. To our knowledge, these parameters have not been provided in the literature, so we show these results as predictions of our method.

In CrCoNi, SRO has significant influences on various properties of the material ranging from hardness[30] and stacking fault energy[27] to magnetism.[31] We show that the SRO also has an ev-

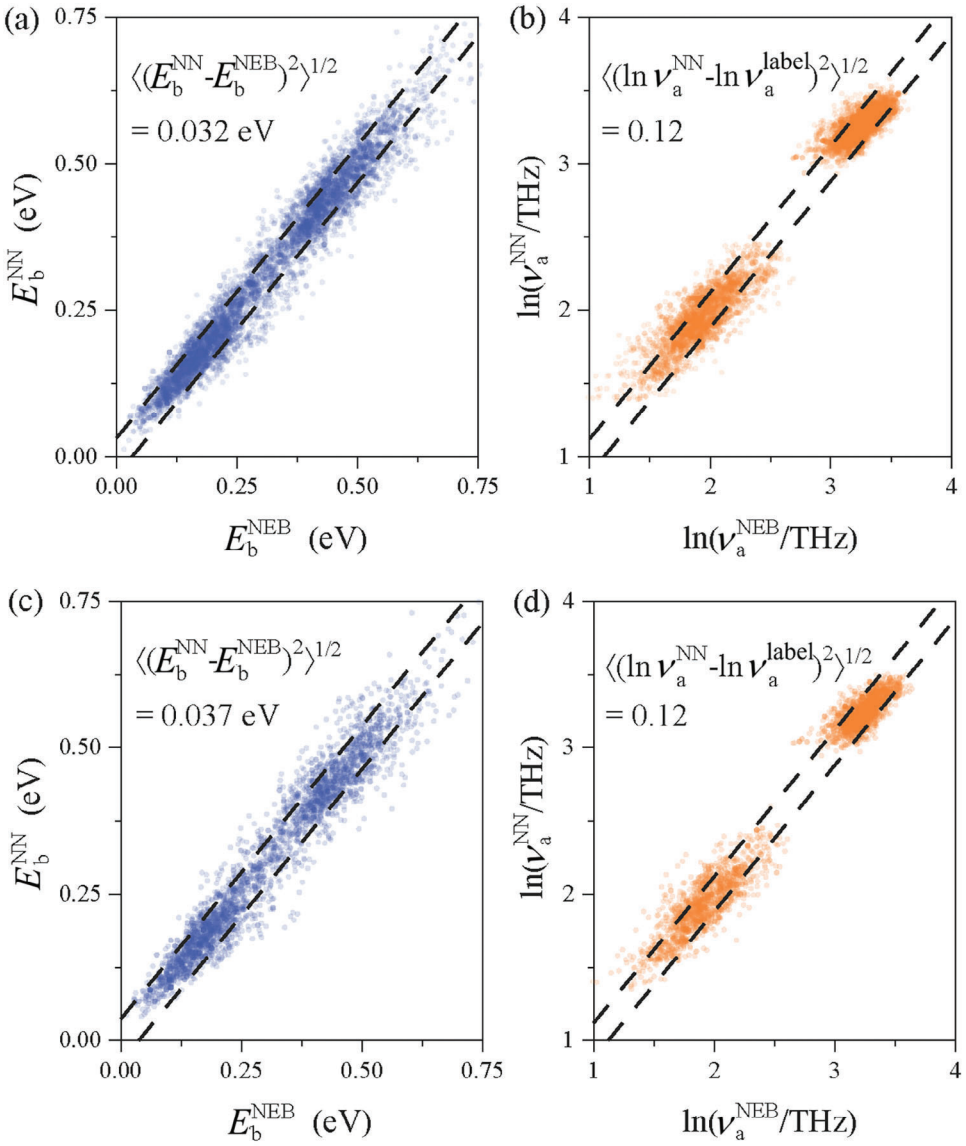

Figure 2. Comparison of the TEL neural network prediction of a) transition energy barriers E_b^{NN} and b) attempt frequency ν_a^{NN} with those calculated by the NEB method in the training dataset, E_b^{NEB} and ν_a^{label}. Here we show the data for hydrogen diffusion in equiatomic CrCoNi alloy. Validation on the testing dataset is shown in c) and d).

ident influence on the hydrogen diffusivity in CrCoNi, as shown in Figure 3c. The system with SRO under thermal equilibrium (SRO = 1) gives approximately double the hydrogen diffusivity of the fully random configuration (SRO = 0), showing that the SRO enhances hydrogen diffusion. This can be explained by the reduction of Cr–Cr bond concentration by the SRO,[27] as the hydrogen transition energy barriers proximate to the Cr–Cr bond are found to be higher than the average hydrogen transition energy barriers. Our results predict that the hydrogen diffusion behavior can also be tuned by the SRO in multi-principal element alloys.

RL TKS can also be used to discover geometrically surprising diffusion mechanisms, where the diffusion pathway can be counter-intuitive and involve cooperative motion of multiple atoms. We apply our method to the hydrogen-vacancy (HV) complexes diffusion in FCC copper.[32] The HV complex consists of a copper vacancy and a few hydrogen atoms adsorbed around the vacancy, as shown in **Figure** 4a. RL TKS is trained on a series of HV complexes containing one to eight adsorbed hydrogen atoms, providing a prediction accuracy of 28 meV for E_b (Figure 4b) and 0.10 for $\ln \nu_a$. We use RL TKS to simulate the HV complex diffusion, which can happen through multiple different transition pathways. Here, we present the transition pathway of one frequently appearing diffusion event, shown in Figure 4c. As the hydrogen distribution influences the vacancy transition rate, one hydrogen atom moves ahead to form a hydrogen arrangement that enhances the transition rate of the vacancy (step 0–3). The vacancy then follows the hydrogen (step 3–4). Finally, the hydrogen left behind follows the vacancy, completing the overall trans-

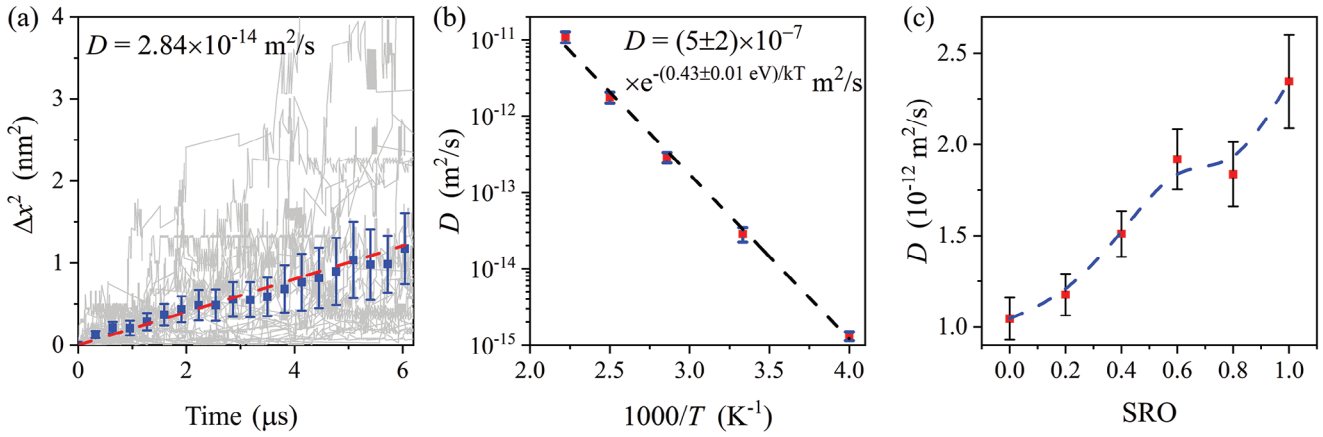

Figure 3. RL TKS hydrogen diffusion simulation in CrCoNi medium entropy alloy. a) Square hydrogen diffusion displacement Δx^2 (absolute value) as a function of time under 300 K. The grey lines show 30 trajectories; the blue squares and error bars are the mean square displacements and their error range (± one standard error); the red dashed line is the linear fitting of the blue dots. b) Arrhenius plot of hydrogen self-diffusivity under different temperatures. The blue caps show the error bar of calculated diffusivities, and the black dashed line is a linear fitting of log D versus $\frac{1000}{T}$. c) Hydrogen self-diffusivity at 400 K as a function of the short-range ordering parameter (the dashed line is a B-spline[28] connecting the data points). SRO = 0 corresponds to a fully random solid solution, SRO = 1 corresponds to WC parameters obtained from ref. [27], and intermediate values of SRO are linearly interpolated. The SRO is sampled using the OTIS code in ref. [29].

lation of the complex. Such a complex diffusion mechanism can hardly be conjectured by human, showing that our method can be applied to cases when the KMC event table is hard to construct without deep learning (in our case, the reduced-dimension "transition energy landscape" TEL is constructed based on Equations (2) and (3), and see also Section 4.1).

RL TKS provides evident speed-up compared to off-lattice KMC without RL acceleration, as shown in **Figure 5** in Section 4.2. Although implementing the TKS requires RL training in advance, their simulation runtime per transition step is about two orders of magnitudes smaller than off-lattice KMC. Therefore, the overall computational costs of the RL methods are smaller than the off-lattice KMC as long as the total number of simulation steps is larger than the threshold. We can see that our simulations in Figure 3 involve far more simulation steps than the threshold, demonstrating that RL methods provide significant acceleration. The acceleration is because to determine transition probabilities, one needs to do expensive saddle-point searches for all possible actions in the off-lattice KMC, while we just need to evaluate the Q_θ function in RL TKS, which is much faster.

2.3. RL Low-Energy States Sampler Option

The second option of our method, LSS, is a "true" RL method in that the discount-rate γ is generally taken to be finite and positive, so the RL value function Q is cumulative and inclusive of future rewards. That is to say, LSS looks long into the future event horizon. LSS also typically sets the reward function as the energy reduction after the transition:

$$r_t^{\mathrm{LSS}} \equiv E(s_t) - E(s_{t+1}) \tag{8}$$

($E(s)$ is the potential energy of state s). We may also include the vibrational free-energy contribution

$$r_t^{\mathrm{LSS}} \equiv F(s_t) - F(s_{t+1}) \tag{9}$$

if we choose to, which would involve the extra computational cost of computing and diagonalizing the Hessian matrices on the fly, or its learned version

$$r_t^{\mathrm{LSS}} \equiv F_\theta(s_t) - F_\theta(s_{t+1}). \tag{10}$$

The model is trained by the Deep Q-Network (DQN) algorithm,[33] which aims to maximize the total reward

$$R^{\mathrm{LSS}} \equiv \sum_{t=0}^{t_{\mathrm{horizon}}-1} \gamma^t r_t^{\mathrm{LSS}} \tag{11}$$

on a trajectory with a discount factor γ close to one (set as 0.8 in our calculation). The model parameters are updated according to the Bellman equation.[33]

$$\theta \leftarrow \theta - \lambda\nabla_\theta \sum_t \left(r_t^{\mathrm{LSS}} + \gamma \max_{a'} Q_{\theta^t}^{\mathrm{LSS}}(s_{t+1}, a') - Q_\theta^{\mathrm{LSS}}(s_t, a_t)\right)^2 \tag{12}$$

where θ^t is the target network that updates less frequently than θ, and then one intermittently assigns θ to θ^t to iterate. Such training gradually builds up the TEL neural network, in the reduced-dimension action space $\mathcal{A}(s) = \{a = (i, \vec{v})\}$ based on the atomic configuration s, which is much smaller than the $3N$-dimensional PEL and is highly adaptive, that is, it is trained to pay "attention" to only the small subspace of s that is likely to lead to large Q within time horizon t_{horizon}.

The converged $Q_\theta^{\mathrm{LSS}}(s_t, a_t)$ represents reduced-dimension TEL and fits the expected value of the maximal total rewards after timestep t, $\max_{(a_{t+1}, a_{t+2}, \ldots)} \sum_{t'=t}^{t_{\mathrm{horizon}}-1} \gamma^{t'-t} r_{t'}^{\mathrm{LSS}}$ (interested readers can find details about the DQN training algorithm and implementation in RL textbooks). As the Q^{LSS} function "foresees" the energy reduction of future steps and chooses actions that maximize long-term cumulative return, it is expected to converge to low-energy configurations faster than local strategies that only consider single-step energy terms. RL LSS is thus an efficient an-

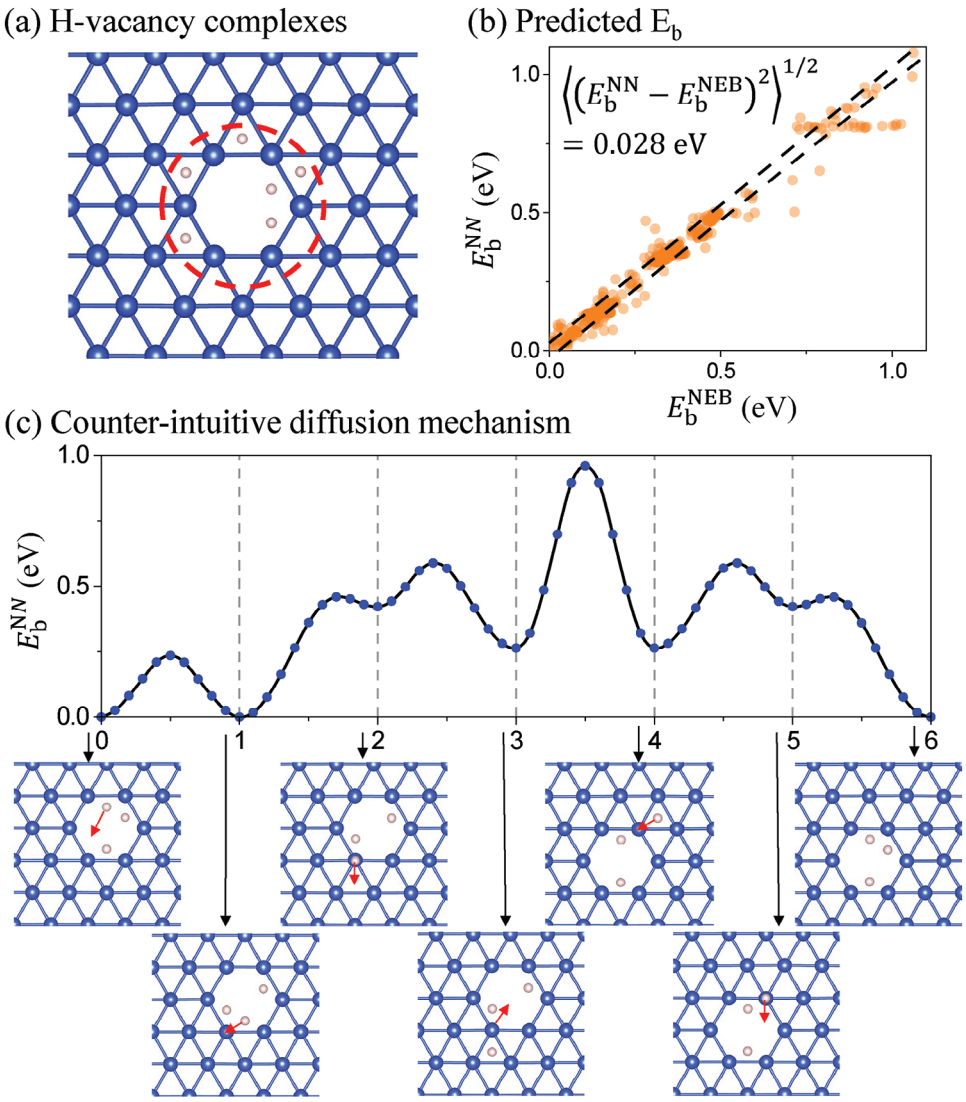

Figure 4. Hydrogen-vacancy complexes diffusion in FCC copper with RL TKS. a) Atomic configuration of a hydrogen-vacancy complex, where multiple hydrogen atoms are adsorbed around a vacancy site. The blue and pink spheres represent copper and hydrogen atoms, respectively. b) NN predicted energy barriers compared with the energy barriers calculated by the NEB method. c) A counter-intuitive diffusion mechanism identified from the RL TKS simulation.

nealer ("the end justifies the means"), which converges to a near-ground state with fewer timesteps than RL TKS ("procedural justice").

We demonstrate RL LSS's performance in simulating energy annealing by the process of hydrogen migration to copper (111) surface, as shown in Figure 5a. 4 × 4 × 3 hexagonal supercells are constructed with 10 randomly sampled hydrogen sites, and the (111) surface is created with a 15 Å vacuum layer. Hydrogen in the surface adsorption sites has lower energy than that in the bulk interstitial site, so the energy ground state is that all hydrogen atoms are on the surface adsorption sites. However, because of the energy difference between the octahedral sites and tetrahedral sites, the migration pathway involves multiple local energy minimums and low-energy barriers, making it challenging to sample the lowest-energy states.[34] After training, our RL policy gives the most likely action from each state, as shown in Figure 5a. Within the cut-off radius of 8.5Å in Equation (3) from the surface, the highest-probability actions (HPAs) from all sites are oriented toward the surface. The HPAs from surface adsorption sites point to neighbor surface sites. This policy provides orientation for the hydrogen atoms to migrate across the local energy barriers toward the surface sites. The HPAs from sites close to the surface have larger Q^{LSS} values than that far from the surface, as the discount factor reduces the contribution of long-term rewards to the Q^{LSS} function compared to short-term rewards.

We compare the annealing process using RL LSS and the Metropolis–Hastings algorithm,[35] as shown in Figure 5b,c. RL LSS leads all hydrogen atoms to surface adsorption sites and converges to the energy ground states in 200 timesteps in all 50 trajectories. From the grey lines, one can observe that the system moves across a large number of low-energy barriers and approaches the ground state. In comparison, the Metropolis–

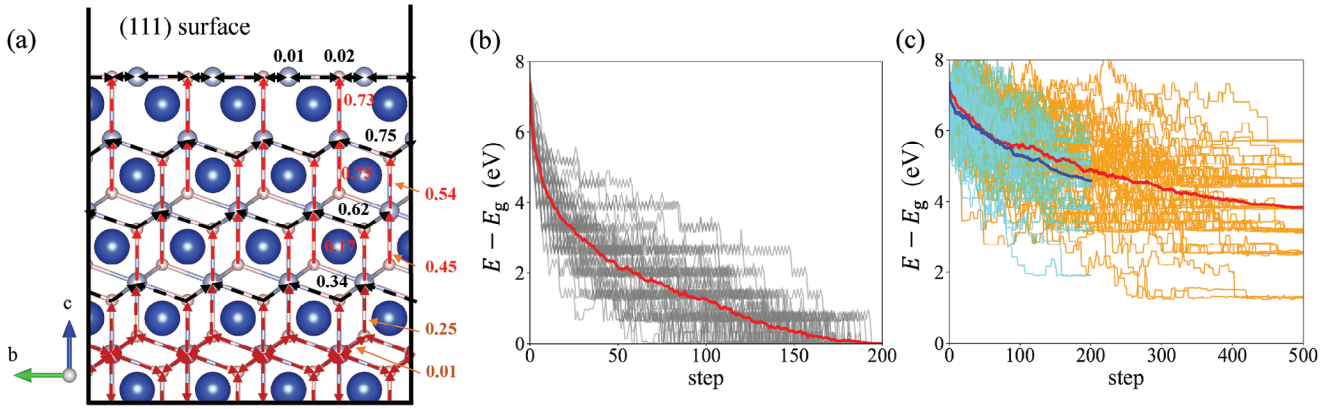

Figure 5. RL LSS sampling low-energy configurations of hydrogen migration to copper (111) surface. a) Highest probability actions (HPAs) and Q values of hydrogen atoms. The blue, silver, and pink spheres are copper atoms, octahedral interstitial sites, and tetrahedral interstitial sites. The HPAs (the actions with the highest probability according to the policy) are shown by arrows (red arrow: a unique HPA, black arrow: multiple [but not all] actions with equal probabilities, brown arrow: all actions have equal probabilities). The Q values of HPAs are denoted. Energy (using ground state energy as reference) versus simulation step under simulated annealing with b) $T = 1000 - 950\frac{t}{200}$ K using the trained policy and c) $T = 3000 - 2700\frac{t}{\tau_{\text{anneal}}}$ K (τ_{anneal}=200 for blue lines and τ_{anneal}=500 for red lines) using Metropolis–Hastings algorithm. The grey/cyan/orange thin lines are 50 simulation trajectories, and the thick red/blue lines are their average.

Hastings algorithm converges slowly. Less than half of the hydrogen migrates to the surface sites in 500 timesteps annealing, leaving ≈4eV energy above the ground state on average. These results demonstrate that the LSS can show advantageous performance in approaching low-energy configurations compared to straightforward Monte Carlo methods. A long lookahead t_{horizon} provides incentive for the hydrogen atoms stuck in the middle to move up, and the transition path networks self-assembled in Figure 5a look similar to the approach taken in the previous diffusive MD (DMD) algorithm.[2] DMD is however a much cruder dynamical simulator, without taking into account the correlations between adjacent atoms, which are now satisfactorily covered by the TEL neural network Q_θ.

3. Discussion and Conclusions

A major difference between the training schemes for TKS and LSS is that the TKS only learns the immediate reward: $Q^{\text{TKS}}(s_t, a_t) \to r_t$, while the LSS learns the cumulative future reward $Q^{\text{LSS}}(s_t, a_t) \to \mathbb{E}[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}r_\tau]$. The reason why we design the learning scheme in this way is that the TKS aims to reproduce the KMC transition probabilities, which depend only on the forward transition rates from the current states. Thus, RL TKS is a contextual bandit problem, and the Q function is trained as supervised learning using training dataset iteratively generated by sampling trajectories. In comparison, the LSS aims to solve a global minimization problem to find the low-energy states. The trajectory toward low-energy states involves a large number of future transitions (Figure 4), so the selection of action needs to consider which action is more promising in reducing energy in the long term. Optimizing the cumulative future rewards requires training algorithm beyond supervised learning, and the DQN of our choice is one of the well-developed RL methods to deal with cumulative rewards optimization.[33]

TKS and LSS can be viewed as two special limits of a unified RL DQN dynamics. The generalized present reward function can be written as a linear combination of the forward barriers and profits:

$$r_t = -\alpha(\tilde{F}(s_t^{\text{saddle}}) - F(s_t)) - \beta(F(s_{t+1}) - F(s_t)) \tag{13}$$

where

$$F(s) \equiv E(s) + k_B T \sum_{i=1}^{3M} \log \nu_i(s) + F_0 \tag{14}$$

is the vibrational free energy of state s, and

$$\tilde{F}(s^{\text{saddle}}) \equiv E(s^{\text{saddle}}) + k_B T \sum_{j=1}^{3M-1} \log \nu_j^*(s^{\text{saddle}}) + F_0 \tag{15}$$

is the effective free energy of the saddle point s^{saddle} (F_0 is a state-independent constant). There are three continuously tunable dimensionless parameters (α, β, γ) then in DQN dynamics Equation (1) using the generalized

$$Q \equiv \mathbb{E}\left[\sum_{t=0}^{t_{\text{horizon}}-1} \gamma^t r_t\right] \tag{16}$$

and its learning Q_θ. α, β, γ controls the importance assigned to reproducing the present-step transition probabilities, present-step potential energy reductions, and long-term lookahead of the model, respectively. TKS and LSS correspond to ($\alpha = 1$, $\beta = 0$, $\gamma = 0$) and ($\alpha = 0$, $\beta = 1$, $\gamma \simeq 1$) corners of the general (α, β, γ) parameter space, respectively. Other parametric settings, for which we are still seeking physical meaning in statistical physics (see Section 4.3), can be used to explore different aspects of PES with certain preferences. A probabilistic interpretation of this general DQN framework equations (1, 13–16) is discussed in Section 4.3, mapping each parameter set to a probability distribution function from which the trajectory is sampled.

Our method provides a general computational framework to simulate the long-timescale diffusion and annealing process. Although the simulations in this paper focus on hydrogen diffusion in metals, the method is actually applicable to diffusion processes in different materials and microstructures, given a specifically designed action space. This method can also bridge large length scales, by first training a model on varied small structures, then deploying the model to guide the long-timescale simulation in a large supercell that includes the complexity of all trained structures.

4. Method

4.1. Action Space Identification Algorithm

A big part of our computational saving comes from the learning of a reduced-dimension TEL, that is, the energetic forward barriers and profits for a given action. The action space $\mathcal{A}(s) \equiv \{a = (i, \vec{v})\}$ is identified based on the atomic configuration s. The ground truths for these energetic barriers and profits in the space of actions is computed based on the NEB or other rigorous algorithms navigating the $3N$-dimensional potential energy landscape (PES). But once learned, the "transition-energy landscape" is smaller in dimension ($\dim \mathcal{A}(s) \ll \dim s = 3N$) and much faster to evaluate than running NEB calculations on the fly. One can also think of $\mathcal{A}(s)$ as the equivalent of "attention" mechanism[36] in the atomic configuration space, focusing only on the small cluster of atoms that is likely to be altered at present in s. The reduced-dimension TEL is therefore an on-the-fly, adaptive data superstructure that are built on top of the well-known $3N$-dimensional PES, represented by our "forward barrier" (Equation (15)) and "profit" (Equation (9)) neural networks for evaluating Equation (13).

The algorithm first identifies all hydrogen atoms with indices $i_1, i_2, \cdots$. For each hydrogen atom i, the distance of all metal atoms j within a cut-off radius r_c is ranked

$$r_{ij_1} \le r_{ij_2} \le \cdots \le r_{ij_M} \tag{17}$$

where r_{ij_k} is the distance between atom i and atom j (the kth nearest neighbor of i). Then, we use all metal atoms j_k with a distance $r_{ij_k} < 1.2r_{ij_4}$ (we denote the largest k satisfying the condition as n) and the hydrogen atom i itself to construct a 3D convex hull including these atoms. If the hydrogen atom i is a corner of the convex hull, the hydrogen atom is on a surface adsorption site; if the hydrogen atom i is inside the convex hull, the hydrogen atom is a bulk interstitial site.

If the hydrogen atom is in a bulk interstitial site, we choose all face centers, $(\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_m)$, of the convex hull ($j_1$, $\cdots$, j_n). Then, the actions toward every face center $(i, \max(1.6(\vec{c}_k - \vec{r}_i), 1.2\forall\frac{\vec{c}_k - \vec{r}_i}{|\vec{c}_k - \vec{r}_i|})), k = 1, 2, \ldots, m$ are included into the action space, except there are "collisional" events. The "collisional" event is defined as, if the hydrogen atom i takes the action, it will have a smaller distance than 0.5 Å with at least one other atom. If the hydrogen atom "collides" with another hydrogen atom, the action is directly discarded. If the hydrogen atom "collides" with a metal atom, the metal atom will be added to reconstruct the convex hull, and actions toward face centers adjacent to the added atom will be included, except if it evokes another "collision." If that happens, the action will be directly discarded.

If the hydrogen atom is on the surface adsorption site, the convex hull is reconstructed using metal atoms j_k satisfying $r_{ij_k} < 1.2r_{ij_3}$. Atoms directly connected with the hydrogen atom, ($j_1, j_2, \cdots, j_n$), are identified as the adsorption site (we sort $(j_1, j_2, \cdots, j_n)$ to form a counter-clockwise loop). The adsorption site center is obtained as $\vec{c} = \frac{1}{n}\sum_k \vec{r}_{j_k}$. The adsorption site has n edges, and the sth edge center is $\vec{e}_s \equiv (\vec{r}_{j_s} + \vec{r}_{j_{s+1}})/2$. First, the surface diffusion actions $(i, 1.6(\vec{e}_s - \vec{c})), s = 1, 2, \ldots, n$ are included. Then, the action toward the bulk $(i, 3\forall\frac{\vec{c}_k - \vec{r}_i}{|\vec{c}_k - \vec{r}_i|})$ is included. If "collision" happens, the same procedure as the bulk interstitial site case is applied.

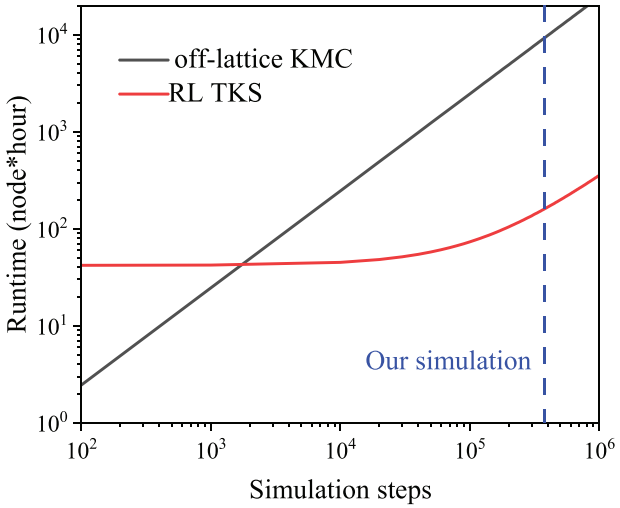

Figure 6. Computation costs estimation of our RL TKS method compared to off-lattice KMC method without deep learning in the hydrogen diffusion problem in equiatomic CrCoNi medium entropy alloy. RL TKS runtime includes both NN training and simulation running on the PFP online server.

4.2. Computational Costs Estimation

We estimate the computational costs of RL TKS compared to the off-lattice KMC for hydrogen diffusion in the equiatomic CrCoNi, as shown in **Figure 6**. The RL training time is 42 node hour, the simulation time per transition step is 1.13 and 88.5 node s for the RL and off-lattice KMC, respectively. The cross-point of the two curves is at 1730 simulation steps, and we have 375 000 steps to produce Figure 3. We note that RL TKS is fundamentally equivalent to KMC. The reason for the acceleration in the case of RL TKS is solely because we have used deep learning to construct a reduced-dimension "transition-energy landscape" that is super fast to evaluate.

4.3. Physical Interpretation of the General DQN Dynamics

The most general form, Equations (1) and (13–16), in the RL framework, does not produce dynamics identical to that of the physical dynamics of trapped metastable systems, often well approximated by Markovian network[37] + HTST in statistical kinetics. However, we feel this general dynamics parameterized by

continuous parameters (α, β, γ) should still have certain physical meaning. In this section, we will explore the possible conceptual explanations of RL dynamics, in different regimes of (α, β, γ).

By setting the parameters (α, β, γ), our method samples different time-dependent probability distributions. In physical reality, the transition rate is approximately determined by the HTST:.

$$\Gamma_{s_t a_t} = \nu_a e^{-(E(s_t^{\text{saddle}})-E(s_t))/k_B T} = e^{-(\tilde{F}(s_t^{\text{saddle}})-F(s_t))/k_B T} \tag{18}$$

If thermal equilibrium is reached (time-dependent → time-independent probability distribution), the probability distribution among different states in the state space $\mathcal{S}$ is

$$P(s) = \frac{1}{Z} e^{-F(s)/k_B T}, \quad Z = \sum_{s\in\mathcal{S}} e^{-F(s)/k_B T} \tag{19}$$

Below we analyze two different limits of RL discount-rate $\gamma = 0$, where only the forward barrier and profit at present are relevant, and $\gamma \approx 1$, where all future profits within the time horizon t_{horizon} are relevant.

4.3.1. $\gamma = 0$ and Modified Detailed Balance

If $\gamma = 0$, the value function cares only about the present, $Q^*(s_t, a_t) = r_t = -\alpha(\tilde{F}(s_t^{\text{saddle}}) - F(s_t)) - \beta(F(s_{t+1}) - F(s_t))$. The problem simplifies into choosing an action based on the next step reward, namely, a contextual bandit problem. If the parameterized $Q_\theta(s, a)$ properly reproduce the exact value function $Q^*(s, a)$, the policy gives

$$\pi_\theta(a|s) = \frac{(\Gamma_{sa})^\alpha P(s'_{sa})^\beta}{\sum_{a'\in\mathcal{A}_s}(\Gamma_{sa'})^\alpha P(s'_{sa'})^\beta} \tag{20}$$

where s'_{sa} is the next state after taking action a. For RL TKS that reproduces the transition rates of Equation (18), the coefficients are set as $\alpha = 1$, $\beta = 0$. The policy then gives

$$\pi_\theta(a|s) = \frac{\Gamma_{s_t a}}{\sum_{a'}\Gamma_{s_t a'}} \tag{21}$$

and the stationary time of the system at state s, $\tau(s)$, is evaluated as

$$\tau(s_t) = \frac{1}{\sum_a \Gamma_{sa}} = \frac{1}{\sum_a e^{Q^*(s,a)/k_B T}} \tag{22}$$

In certain scenarios, the goal is to sample thermal equilibrium distribution. The detailed balance principle (**Figure 7a**) states that if the following kinetic laws holds for arbitrary states s_1, s_2

$$\frac{1}{\tau(s_1)}\pi_\theta(a_{12}|s_1)P(s_1) = \frac{1}{\tau(s_1)}\pi_\theta(a_{21}|s_2)P(s_2) \tag{23}$$

then the sampled states will follow the thermal equilibrium distribution $P(s)$, as given in Equation (19). Here, a_{ij} means the action of transition from state s_i to state s_j. Equation (23) is equivalent to:

$$\exp\left\{\frac{Q^*(s_1,a_{12}) - Q^*(s_2,a_{21})}{k_B T}\right\} = \exp\left\{-\frac{F(s_2)-F(s_1)}{k_B T}\right\} \tag{24}$$

the well-known forward barrier–backward barrier–thermodynamics connection. As $Q^*(s_i, a_{ij}) = -\alpha(\tilde{F}(s^{\text{saddle}}) - F(s_i)) - \beta(F(s_j) - F(s_i))$, we have

$$Q^*(s_1,a_{12}) - Q^*(s_2,a_{21}) = (\alpha+2\beta)(F(s_1)-F(s_2)) \tag{25}$$

Then standard detailed balance Equation (24) would demand

$$\alpha + 2\beta = 1 \tag{26}$$

So we proved that the steady-state probability distribution can approach Equation (19) as long as $\alpha + 2\beta = 1$.

But interestingly, if $\alpha + 2\beta \neq 1$, an altered form of detailed balance still holds.

$$\exp\left\{\frac{Q^*(s_1,a_{12}) - Q^*(s_2,a_{21})}{k_B T}\right\} = \exp\left\{-\frac{F(s_2)-F(s_1)}{k_B T_{\text{eq}}}\right\} \tag{27}$$

just with

$$T_{\text{eq}} \equiv \frac{T}{\alpha+2\beta} \tag{28}$$

In other words, $\gamma = 0$ DQN can have two temperatures, a kinetic temperature T using which we run Equation (1), and a thermodynamic temperature T_{eq} where, when steady state is approached, the probability distribution still has a canonical form (Equation (19)) but with a rescaled temperature $T_{\text{eq}} = \frac{T}{\alpha+2\beta}$. So if $\alpha + 2\beta > 1$, the thermodynamic temperature is lower than the kinetic temperature, and vice versa.

Algorithmically we do not need to always keep a constant α, β. For example, one strategy could be while keeping T_{eq} constant, one varies β/α, and a larger β/α ratio could promote sampling more rare transition events while keeping the eventual thermodynamic properties correct. Indeed, if $\alpha \to 0$, we would be doing classical Metropolis Monte Carlo[38] (MC, not KMC) for sampling

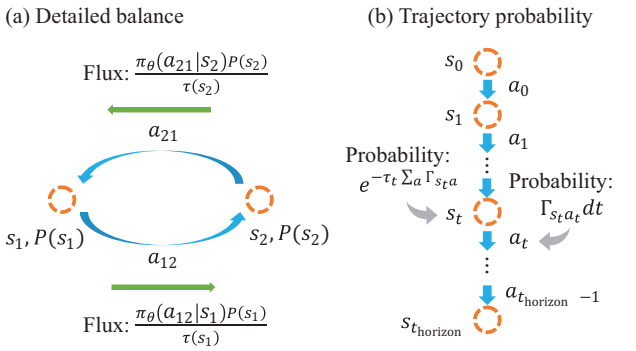

Figure 7. Illustration of the a) detailed balance and b) trajectory probabilities. Orange circles, blue arrows, and green arrows represent states, transitions, and probability flux, respectively.

the equilibrium thermodynamic distribution. Thus, β/α is a knob to smoothly tune from KMC to MC.

Because detailed balance is such an important concept in statistical kinetics (to prevent, e.g., infinite looping between a ring of states at thermodynamic equilibrium), it is important to discuss about the type of numerical errors that could break detailed balance. Due to the neural network prediction error, Q_θ and Q^* are not *exactly* the same, and may contain numerical noise. If we approximate Q^* by Q_θ in Equation (25), the detailed balance Equation (27) may not hold exactly. Therefore, the neural network prediction error will influence reaching thermodynamic equilibrium $T_{\rm eq}$.

If exact detailed balance is desirable, one can do the following "symmetrization procedure" with a higher computational cost: for each $Q_\theta(s_i, a_{ij})$, we apply the action and get the next state s_j, and calculate $F(s_j)$ and the backward $Q_\theta(s_j, a_{ji})$, returning to i. We then use the symmetrized value function

$$Q_\theta^{\rm corrected}(s_i, a_{ij}) \equiv Q_\theta(s_i, a_{ij}) + \frac{Q_\theta(s_j, a_{ji}) - Q_\theta(s_i, a_{ij}) - (\alpha + 2\beta)(F(s_j) - F(s_i))}{2} \tag{29}$$

always to sample the action. This ensures detailed balance and reaching thermodynamic equilibrium $T_{\rm eq}$ despite of neural network error.

4.3.2. $\gamma \sim 1$: Maximizing the Path Probability of a Trajectory

When we set $\gamma \sim 1$, the algorithm maximizes the total reward of the trajectory $\mathcal{J} \equiv (s_0, a_0, \tau_0, s_1, a_1, \tau_1, \ldots s_{t_{\rm horizon}})$ (Figure 7b), $R(\mathcal{J}) \simeq \sum_{t=0}^{t_{\rm horizon}} r_t$ ($t_{\rm horizon}$ is the time horizon of the trajectory. We consider setting γ slightly smaller than 1 as a convergence technique that leads to a small bias).

The physical probability of a trajectory $\mathcal{J}$ according to the conventional Markovian network[37] + HTST in standard statistical kinetics, given an initial state s_0, is a product of two factors: 1) the probability of staying in state s_t for physical time τ_t equals $e^{-\tau_t \sum_a \Gamma_{s_t a}}$, and 2) the probability of the transition from s_t to s_{t+1} in a small time interval $[\sum_{i=0}^{t} \tau_i,\ d\tau + \sum_{i=0}^{t} \tau_i]$ equals $\Gamma_{s_t a_t} d\tau$. We multiply all factors together to get the total probability of getting the trajectory $\mathcal{J}$ as

$$P(\mathcal{J}|s_0) = \prod_{t=0}^{t_{\rm horizon}-1} e^{-\tau_t \sum_a \Gamma_{s_t a}} \Gamma_{s_t a_t} d\tau \tag{30}$$

In any trajectory that our DQN method outputs, the expected stationary time $\tau_t = 1/\sum_a \Gamma_{s_t a}$, so $e^{-\tau_t \sum_a \Gamma_{s_t a}} = e^{-1}$ and the probability product becomes $P(\mathcal{J}|s_0) = \prod_{t=0}^{t_{\rm horizon}-1} \Gamma_{s_t a_t} (e^{-t_{\rm horizon}} d\tau^{t_{\rm horizon}})$. As $e^{-t_{\rm horizon}} d\tau^{t_{\rm horizon}}$ is a constant independent from the policy, we can write the path probability as

$$P(\mathcal{J}|s_0) \propto \exp\left\{ -\sum_{t=0}^{t_{\rm horizon}-1} \frac{\tilde{F}(s_t^{\rm saddle}) - F(s_t)}{k_B T} \right\} \tag{31}$$

according to the conventional Markovian chain + HTST. Equation (31) is in fact a path integral akin to the action integral of a trajectory in classical mechanics, and the "principle of least action" applies when we think about the most likely physical trajectory of a metastable system on a Markovian network.[37]

In contrast, in our general DQN trajectory, the total reward can be rewritten as:

$$\begin{aligned} R(\mathcal{J}) &= -\alpha \sum_{t=0}^{t_{\rm horizon}-1} (\tilde{F}(s_t^{\rm saddle}) - F(s_t)) \\ &\quad - \beta \sum_{t=0}^{t_{\rm horizon}-1} (F(s_{t+1}) - F(s_t)) \\ &= k_B T \left[\alpha \log P(\mathcal{J}|s_0) + \beta \log P(s_{t_{\rm horizon}})\right] + C_0 \end{aligned} \tag{32}$$

where C_0 is a constant independent from the policy. So we can see that maximizing the total rewards in DQN with $\gamma \sim 1$ is equivalent to maximizing

$$A \equiv \alpha \log P(\mathcal{J}|s_0) + \beta \log P(s_{t_{\rm horizon}}). \tag{33}$$

We can see right away that while this is different from the physical action integral (31), it does contain the path-integral contribution, while also mixing with the final energy drop $F(s_{t_{\rm horizon}}) - F(s_0)$. So the physical interpretation of α is an emphasis on "procedural justice," while β emphasizes "the end justifies the means" (consider that Metropolis MC[38] has only β and not α, while KMC requires only α and not β).

If $\alpha = 0$, $\beta = 1$, the method aims to sample the most probable final state $s_{t_{\rm horizon}}$ were the system at thermal equilibrium, corresponding to an annealing process that targets the ground state. If on the one hand $\alpha = 1$, $\beta = 0$, the method aims to sample the most probable trajectory based on transition kinetics. In the most general case, α and β can be tuned to balance "procedural justice" with "the end justifies the means."

Again, α, β and even γ only need to be piece-wise constant, and the relative emphasis on "procedural justice" versus "end justifies the means" may be tuned on the fly. For example, one could first use large β/α and γ to scope out the possible global direction of free-energy reduction (see Figure 5), perform on-the-fly training of the relevant forward barriers and profits, and then based on this experience, downtune the β/α as well as γ to get more and more realistic physical time estimation of the paths in this general direction. In other words, one may engineer "morphing" of the RL dynamics in (α, β, γ) parameter space, from the $(0, 1, 1^-)$ corner running long time horizon annealing, to the $(1^-, 0^+, 0^+)$ corner (KMC) running physical timescale kinetics.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

The authors thank Prof. Cathy Wu, Dr. Shen Shen, and Zhiyuan Shu for their insightful discussions. This work was supported by NSF DMR-1923976, CMMI-1922206, and Hydrogen in Energy and Information Sciences (HEISs), an Energy Frontier Research Center funded by the U.S.

Department of Energy (DOE), Office of Science, Basic Energy Sciences (BES), under Award DE-SC0023450. H. T. also acknowledges support from a Mathworks Engineering Fellowship. The calculations in this work were performed in part on the Matlantis high-speed universal atomistic simulator and the Texas Advanced Computing Center (TACC).

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

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

Keywords

hydrogen diffusion, long-timescale simulations, reinforcement learning

Received: June 21, 2023
Revised: October 29, 2023
Published online: December 7, 2023

[1] S. M. Allen, R. W. Balluffi, W. C. Carter, *Kinetics of Materials*, John Wiley & Sons, New York **2005**.
[2] J. Li, S. Sarkar, W. T. Cox, T. J. Lenosky, E. Bitzek, Y. Wang, *Phys. Rev. B* **2011**, *84*, 054103.
[3] S. K. Dwivedi, M. Vishwakarma, *Int. J. Hydrogen Energy* **2018**, *43*, 21603.
[4] S. Kumar, Z. Wang, X. Huang, N. Kumari, N. Davila, J. P. Strachan, D. Vine, A. D. Kilcoyne, Y. Nishi, R. S. Williams, *Appl. Phys. Lett.* **2017**, *110*, 103503.
[5] B. Uberuaga, R. Smith, A. Cleave, F. Montalenti, G. Henkelman, R. Grimes, A. Voter, K. Sickafus, *Phys. Rev. Lett.* **2004**, *92*, 115505.
[6] P. J. Feibelman, *Phys. Rev. Lett.* **1990**, *65*, 729.
[7] J. Li, K. J. Van Vliet, T. Zhu, S. Yip, S. Suresh, *Nature* **2002**, *418*, 307.
[8] B. P. Uberuaga, D. Perez, *Handbook of Materials Modeling: Methods: Theory and Modeling*, Springer Cham, Switzerland **2020**, p. 683.
[9] D. Perez, R. Huang, A. F. Voter, *J. Mater. Res.* **2018**, *33*, 813.
[10] A. F. Voter, in *Radiation Effects in Solids*, Springer, Berlin **2007**, pp. 1–23.
[11] M. Trochet, N. Mousseau, L. K. Béland, G. Henkelman, *Handbook of Materials Modeling: Methods: Theory and Modeling*, Springer Cham, Switzerland **2020**, p. 715.
[12] M. R. Sorensen, A. F. Voter, *J. Chem. Phys.* **2000**, *112*, 9599.
[13] S. Takamoto, S. Izumi, J. Li, *Comput. Mater. Sci.* **2022**, *207*, 111280.
[14] S. Takamoto, D. Okanohara, Q. Li, J. Li, *J. Materiomics* **2023**, *9*, 447.
[15] Z. Wang, Z. Shi, Y. Li, J. Tu, in *2013 IEEE international conference on robotics and biomimetics (ROBIO)*, IEEE, New York **2013**, pp. 1199–1204.
[16] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, et al., *J. Phys.: Condens. Matter* **2017**, *29*, 273002.
[17] H. Jónsson, G. Mills, K. W. Jacobsen, in *Classical and Quantum Dynamics in Condensed Phase Simulations*, World Scientific, Singapore **1998**, pp. 385–404.
[18] L. Zhang, J. Han, H. Wang, W. Saidi, R. Car, *Advances in Neural Information Processing Systems* **2018**, *31*.
[19] D. Bouneffouf, I. Rish, C. Aggarwal, in *2020 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, New York **2020**, pp. 1–8.
[20] H. Kimizuka, M. Shiga, *Phys. Rev. Mater.* **2021**, *5*, 065406.
[21] Y. Sakamoto, K. Takao, *J. Jpn. Inst. Met.* **1982**, *46*, 285.
[22] N. Ansari, R. Balasubramaniam, *Mater. Sci. Eng., A* **2000**, *293*, 292.
[23] T. Ishikawa, R. B. McLellan, *J. Phys. Chem. Solids* **1985**, *46*, 445.
[24] K. Lee, R. McLellan, *Scr. Metall.* **1984**, *18*, 859.
[25] H. Magnusson, K. Frisk, *J. Phase Equilib. Diffus.* **2017**, *38*, 65.
[26] D. Liu, Q. Yu, S. Kabra, M. Jiang, P. Forna-Kreutzer, R. Zhang, M. Payne, F. Walsh, B. Gludovatz, M. Asta, A. M. Minor, E. P. George, R. O. Ritchie, *Science* **2022**, *378*, 978.
[27] J. Ding, Q. Yu, M. Asta, R. O. Ritchie, *Proc. Natl. Acad. Sci.* **2018**, *115*, 8919.
[28] W. J. Gordon, R. F. Riesenfeld, in *Computer Aided Geometric Design*, Elsevier, Amsterdam **1974**, pp. 95–126.
[29] L. T. Fey, I. J. Beyerlein, *Integr. Mater. Manuf. Innov.* **2022**, *11*, 382.
[30] R. Zhang, S. Zhao, J. Ding, Y. Chong, T. Jia, C. Ophus, M. Asta, R. O. Ritchie, A. M. Minor, *Nature* **2020**, *581*, 283.
[31] F. Walsh, M. Asta, R. O. Ritchie, *Proc. Natl. Acad. Sci.* **2021**, *118*, e2020540118.
[32] J.-P. Du, W. Geng, K. Arakawa, J. Li, S. Ogata, *J. Phys. Chem. Lett.* **2020**, *11*, 7015.
[33] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, *Nature* **2015**, *518*, 529.
[34] G. Henkelman, H. Jónsson, T. Lelièvre, N. Mousseau, A. F. Voter, *Handbook of Materials Modeling* (Eds.: W. Andreoni, S. Yip), Springer, Berlin **2018**, pp. 1–10.
[35] W. K. Hastings, *Biometrika* **1970**, *57*, 97.
[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, *Advances in Neural Information Processing Systems* **2017**, *30*.
[37] J. Li, A. Kushima, J. Eapen, X. Lin, X. Qian, J. Mauro, P. Diep, S. Yip, *PLoS One* **2011**, *6*, e17909.
[38] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, *J. Chem. Phys.* **1953**, *21*, 1087.

ADVANCED SCIENCE
Open Access

Supporting Information

for *Adv. Sci.*, DOI 10.1002/advs.202304122

Reinforcement Learning-Guided Long-Timescale Simulation of Hydrogen Transport in Metals

*Hao Tang, Boning Li, Yixuan Song, Mengren Liu, Haowei Xu, Guoqing Wang, Heejung Chung and Ju Li**

Suporting Information: reinforcement learning-guided long-timescale simulation of hydrogen transport in metals

Hao Tang,[1] Boning Li,[2,3] Yixuan Song,[1] Mengren Liu,[1] Haowei Xu,[4] Guoqing Wang,[2,4] Heejung Chung,[1] and Ju Li[1,4,*]

[1]*Department of Materials Science and Engineering, Massachusetts Institute of Technology, MA 02139, USA*
[2]*Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*
[3]*Department of Physics, Massachusetts Institute of Technology, MA 02139, USA*
[4]*Department of Nuclear Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

(Dated: November 22, 2023)

S1. NUMERICAL DETAILS OF ATOMISTIC SIMULATION AND RL TRAINING

The model training on pure copper and nickel is conducted on $4 \times 4 \times 4$ cubic supercell of the FCC metals. 3 atomic configurations are generated for each metal, where 4 hydrogen atoms are randomly sampled in all octahedral and tetrahedral sites in each configuration. 20 and 40 trajectories are sampled for copper and nickel, respectively, with 30 timesteps in each. In the atomic relaxation and NEB calculations, all forces converge to 0.05 eV/Å under the PreFerred Potential (PFP) v4.0.0, which is used throughout this paper. The cut-off radius of the neural network model is 4 Å. The embedding network G_k^1 has one hidden layer and an output layer both with a size of 12. Throughout the paper, we take the first 1/4 columns of G_k^1 to form G_k^2, and the input layers of $G_k^{1,2}$ have a size of $N_c + 1$, where N_c is the number of chemical species. We define an element species list: $C = (C_1, C_2, \cdots, C_{N_c}, C_{N_c+1} = \text{action})$, where C_l is the lth element. For $G_k^{1,2}(f_c(r_{im}), c_m = C_l)$, the input layer takes the N_c+1 dimensional input vector whose lth component is $f_c(r_{im})$ and other components are zeros. The fitting network has two hidden layers with a size of 32. The maximum atom number (within the cut-off radius of each atom) of the "transition energy landscape" is set as 40, which has not been exceeded during the training. The training temperature is set as 1000 K throughout this paper. After including the nth trajectory, one randomly samples a trajectory from probability distribution $P_i = \frac{1-0.99}{1-0.99^n} 0.99^{n-i}$ (recent trajectory has larger probability) and train 20 gradient descend steps from the sampled trajectory, and repeat this for n times. The training algorithm is Adam throughout this paper, and the learning rate here is set as 10^{-3} in all online training. Offline training is conducted to further improve the model's accuracy. We separate the training data into the training dataset (2/3 of the data) and the testing dataset (1/3 of the data). 10000 full gradient descent is implemented on the training dataset. The learning rate changes from 10^{-3} to 10^{-5} that exponentially decays with timesteps in all offline training in this paper.

The model training on NiCrCo medium entropy alloy is conducted on $4 \times 4 \times 4$ cubic supercell of the FCC fully random solid solution. 9 atomic configurations are generated for each metal, where 4 hydrogen atoms are randomly sampled in all octahedral and tetrahedral sites in each configuration. 3 independent processes of training are conducted with 101 trajectories in each, and each trajectory contains 30 timesteps. In the atomic relaxation and NEB calculations, all forces converge to 0.05 and 0.07 eV/Å, respectively. The cut-off radius of the neural network model is 5 Å. The embedding network G_k^1 has one hidden layer and an output layer both with a size of 24. The fitting network has two hidden layers with a size of 128. The maximum atom number is set as 50, which was not exceeded during the training. The online training parameters are the same as pure metals. As to offline training, we separate the training data the same way as pure metals. Stochastic gradient descent is implemented with a minibatch size of 500 data points (one timestep is a data point). The minibatch is randomly sampled from all data points, and 10 gradient descent steps are applied to each minibatch. That is repeated for 20000 iterations. In order to avoid overfitting, a normalization term of $5 \times 10^{-6} \parallel \theta \parallel^2$ is added to the loss function.

The deep Q-network learning for copper (111) surface is conducted on $4 \times 4 \times 3$ hexagonal lattice of FCC copper (4 replications on a and b directions and 3 replications on c direction. c direction is along the 3-fold axis). A vacuum layer of 15 Å is included in the c direction. We implemented 7 independent training processes, 4 of them have only one randomly sampled hydrogen atom in the copper slab (12 configurations are sampled as starting points, and initial configurations are randomly selected from them), and the other 3 have 10 randomly sampled hydrogen atoms (10 configurations are sampled as starting points). 300 trajectories are sampled with 30 timesteps in each. In the atomic relaxation, all forces converge to 0.05 eV/Å. The cut-off radius of the neural network model is 8.5 Å, as the model needs more distant atomic information to foresee the long-term rewards. The embedding network G_k^1 has one hidden layer and an out-

* liju@mit.edu

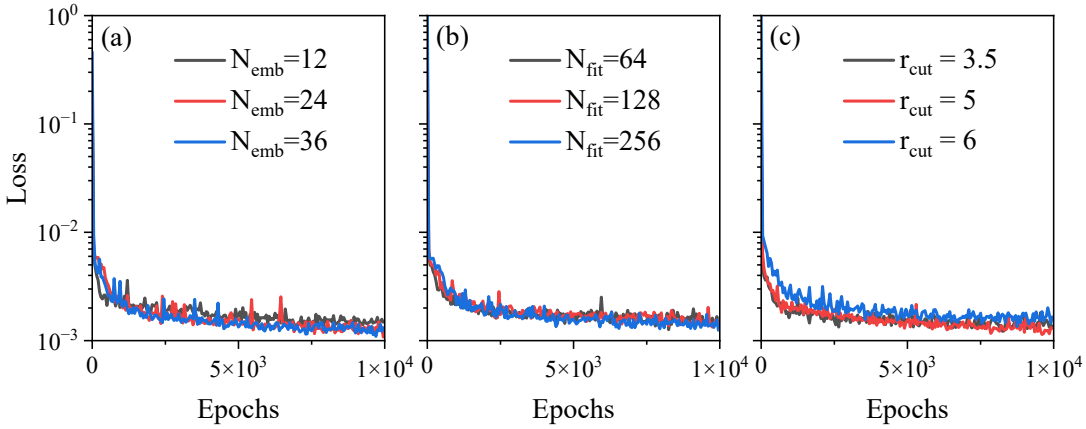

FIG. S1. Training curve (loss *vs* training epochs) of the TKS NN model for the hydrogen diffusion in equiatomic CrCoNi using different model hyperparameters. The $N_{\rm emb}$, $N_{\rm fit}$, and $r_{\rm cut}$ are the width of embedding network, fitting network, and the cut-off radius, respectively. We use $N_{\rm emb} = 24, N_{\rm fit} = 128, r_{\rm cut} = 5$ Å for unlabeled parameters in each panel.

put layer both with a size of 24. The fitting network has two hidden layers with a size of 128. The maximum atom number is set as 260, which has not been exceeded during the training. After including the nth trajectory, one randomly samples a trajectory and trains 5 gradient descent steps from the sampled trajectory, and repeats this for $\lceil n^{2/3} \rceil$ times. The offline training randomly samples a mini-batch with 10 trajectories and applies 10 steps of gradient descent at each iteration. There are 1010 iterations in the training process.

S2. NEURAL NETWORK TRAINING PARAMETERS

In this section, we compare the training curves using different neural network hyperparameters to validate our choice of the NN hyperparameter settings. We use hydrogen diffusion in equiatomic CrCiNi as an example, as shown in Fig. S1. For the embedding network width $N_{\rm emb}$, our choice $N_{\rm emb} = 24$ gives similar training loss with $N_{\rm emb} = 36$, slightly better than $N_{\rm emb} = 12$ (Fig. S1a). The fitting network width $N_{\rm fit} = 64, 128, 256$ gives similar training loss (Fig. S1b). For the cut-off radius $r_{\rm cut}$, our choice $r_{\rm cut} = 5$ gives the lowest training lost within $r_{\rm cut} = 3.5, 5, 6$ (Fig. S1c). In all plots, the loss function converges with respect to epochs. The tests validate that our NN settings give close-to-convergent model performance.

If one wants to further improve the model performance, a more sophisticated NN architecture design will be necessary. A promising choice is to use equivariant graph neural networks [1–3] to represent the $Q_\theta(s, a)$ function, where the state s is represented by a graph and the action a is represented by a vector input on nodes.

[1] S. Takamoto, S. Izumi, and J. Li, Comput. Mater. Sci. **207**, 111280 (2022).

[2] S. Takamoto, D. Okanohara, Q. Li, and J. Li, J. Materiomics **9**, 447 (2023).

[3] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, Nature communications **13**, 1 (2022).