

Prediction of vacancy defect diffusion paths in high entropy alloys via machine learning on molecular dynamics data

Cite as: J. Appl. Phys. 138, 074306 (2025); doi: [10.1063/5.0280842](https://doi.org/10.1063/5.0280842)

Submitted: 14 May 2025 · Accepted: 25 July 2025 ·

Published Online: 20 August 2025



View Online



Export Citation



CrossMark

C. Reimer,^{1,2} P. Saidi,³ C. Casert,⁴ C. Beeler,^{5,6} C. G. Tetsassi Feugmo,⁷ S. Whitelam,⁴ E. Mansouri,⁸ A. Martinez,^{8,9} L. Béland,⁸ and I. Tamblyn^{2,10,a)}

AFFILIATIONS

¹Security and Disruptive Technologies, National Research Council of Canada, Ottawa, Canada

²Department of Physics, University of Ottawa, Ottawa, Canada

³Canadian Nuclear Laboratories, Chalk River, Canada

⁴Molecular Foundry, Lawrence Berkeley National Laboratory, Berkeley, USA

⁵Department of Mathematics and Statistics, University of Ottawa, Ottawa, Canada

⁶Digital Technologies, National Research Council of Canada, Ottawa, Canada

⁷Department of Chemistry, University of Waterloo, Waterloo, Canada

⁸Department of Mechanical and Materials Engineering, Queen's University, Kingston, Canada

⁹Department of Physics, Universidad Nacional de Colombia, Bogotá, Colombia

¹⁰Vector Institute for Artificial Intelligence, Toronto, Canada

Note: This paper is part of the Special Topic on Integrating Data Science and Computational Materials Science.

a)Author to whom correspondence should be addressed: isaac.tamblyn@uottawa.ca

30 August 2025 06:05:26

ABSTRACT

Identifying the diffusion path of point defects is a critical step in understanding their evolution and the mechanisms of related phenomena. Defect diffusion occurs at small length and time scales, with impacts on material properties that may continue to evolve over ns to μ s, ms, and the continuum scale (s, min, etc., and cm, m, etc.). The time scale accessible to molecular dynamics (MD) simulations is limited by small step sizes, typically in the fs range. Thus, surrogate models of MD simulations through machine learning (ML)-based algorithms are of great interest, especially for complex systems such as high entropy alloys (HEAs). In this work, dynamics governing vacancy migration in HEA were approximated with graph convolutional network (GCN) models as ansatzes for kinetic Monte Carlo (KMC) rate catalogs. Network design considered that diffusion in crystalline solids generally depends on interactions between defects and their immediate neighbor atoms. Graphs represented the vacancy surroundings, MD-generated trajectories provided training and comparison datasets, and unsupervised GCN models approximated interatomic dynamics governing vacancy migration in HEAs as ansatzes for KMC. A proof-of-concept model trained on MD data for the Fe, Ni, Cr, Co, and Cu HEA environment was used with two different neighbor interactions to assess the feasibility of training a GCN to predict vacancy defect transition rates in the HEA environment. The resulting setup rapidly generated MD-formatted synthetic trajectories based on dynamics learned from the MD training set, with a time acceleration of roughly two orders of magnitude and a similar diffusion coefficient to MD observations. Additionally, Nudged Elastic Band (NEB) calculations were performed on randomly generated FeNiCrCoCu HEA structures to determine vacancy migration barriers across nearest-neighbor sites. Transition probabilities for each jump, categorized by atomic type, were extracted from these calculations. NEB-based and GCN-based approaches led to similar outcomes.

© 2025 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0280842>

I. INTRODUCTION

Understanding the migration of point defects over mesoscale times (between ~ 1 ns to 1 s) to sinks such as grain boundaries, impurities, or defect clusters is critical due to their role in degradation phenomena such as segregation and creep.

High entropy alloys (HEAs), concentrated (≥ 5 elements) mixtures of multiple elements, have received notable attention due to tailored compositions that offer properties such as stability at elevated temperatures, irradiation damage resilience, and altered defect diffusion.^{1,2} The role of chemical bias (the influence of composition in determining diffusion dynamics) in HEAs is vital to exploiting their future applications and is not presently well understood outside of simpler alloys, with some studies suggesting that particular elements have disproportionate influence irrespective of the remaining composition.³ The diversity of local atomic configurations in HEAs creates a rough potential energy landscape that has made theoretical and computational approaches to understand HEA diffusion difficult via common techniques such as density functional theory (DFT) or molecular dynamics (MD).^{2–6} Another phenomenon that impacts diffusion in HEAs is the interaction of solutes to form regions of chemical short-range order (SRO). Simulations have demonstrated that SRO can both reduce diffusion and trap vacancies due to local changes in the potential energy landscape.⁷

Significant advances in understanding the composition- and configuration-dependent properties of engineered materials have been achieved in recent years via the proliferation of ML algorithms.^{8–12} Of these, one of the more intriguing is crystal graph convolution (CGConv), a GCN architecture that allows for interpretability of atom-node-level contributions to properties based on crystal structure and elemental composition.¹² The success of graph neural networks can be partly attributed to the use of local descriptors for material representations (such as atomic position, number, and bond length/distance), as both molecules and crystalline structures map cleanly onto graphs, and periodic boundary conditions can be easily accommodated.^{13–15}

As a result of applications speculated within the HEA composition space, prediction and simulation of HEAs using ML techniques is a growing subject of interest.^{16–18} In addition, there has been substantial previous work using atomistic techniques such as KMC, MD, and DFT to study HEAs, which tend to focus on the behavior of lattice defects.^{19–32}

While much work has been conducted on predicting the basic properties of materials and screening them, attention has also been directed to the use of ML models as proxies or hybrid models to reduce the time expended on computer simulations.^{33–36} One approach is to use ML models to learn interatomic potentials for DFT or MD, with the aim of reducing reliance on first-principles calculations.^{15,37–41} An alternative technique to simplify the study of advanced materials involves capturing complex transition dynamics via ML models. Recently, Casert *et al.* used a transformer artificial neural network (ANN) architecture with a log-likelihood estimator (based on the Continuous Time Monte Carlo form of the log-likelihood expression) as a loss function in order to learn the unknown local and non-local dynamics in glassy systems, yielding a dynamics model that extrapolated emergent properties of the host systems that were not included in the training set.⁴²

With respect to simulation acceleration, the application of ML to make direct per-atom predictions of transition rates is particularly interesting for overcoming the limitations imposed on HEA research via MD. The area of defect migration in HEAs has specifically seen attention very recently, with ANNs being used to investigate the behaviour of the vacancy defect in hybrid ANN-KMC techniques.^{6,43,44}

In this work, a workflow was developed using ML and KMC techniques to generate accelerated predicted vacancy defect diffusion paths in HEAs. This was conducted in several stages including (a) generating and extracting vacancy states as well as transition atoms from MD trajectories; (b) training a GCN to predict transition rate constants on a per-atom basis using a log-likelihood loss function and local HEA atomic configurations as graph-based feature inputs; and (c) using trained models as part of a KMC-based algorithm to generate accelerated vacancy diffusion paths from the learned dynamics of the MD system.

II. METHODS

In the absence of external stresses, the vacancy defect diffusion path and diffusion rates depend on the type and configuration of the first and second graph neighbor atoms of the defect, the latter of which included the first several face centred cubic (FCC) nearest-neighbour sets of the vacancy.^{27,45} Following this assumption, MD trajectories were used to create a training dataset for a GCN, which, in turn, was used to rapidly generate synthetic trajectories that followed dynamics learned from the MD system (see Fig. 1 and Subsection II C).

In order to simplify the problem being studied, only the first graph nearest neighbour atoms were considered viable transition

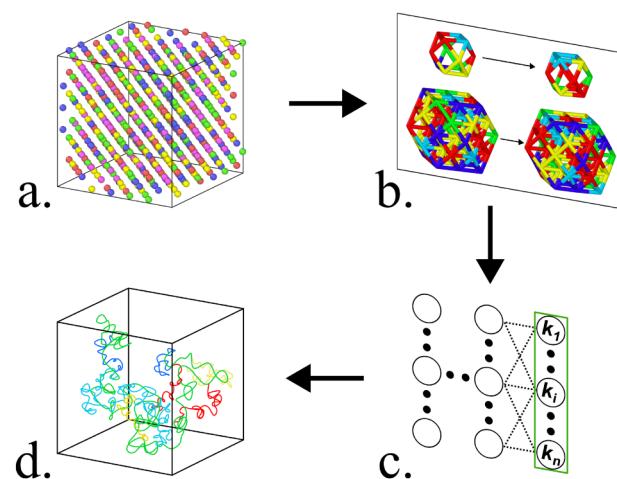


FIG. 1. The workflow used in this work: (a) An MD HEA system produced trajectories from which two vacancy surroundings were extracted to compare the influence of various neighbouring atoms on vacancy transitions (Appendix A), (b) vacancy region datasets were converted to graph format, (c) the resulting dataset of states and chosen transition atoms was fed through a GCN to construct a trained model, and (d) model learned dynamics were used to generate accelerated synthetic trajectories of vacancy defect diffusion in the HEA system.

30 August 2025 06:05:26

atoms to fill the vacancy. This simplification was justified by the small fraction (<1%) of transitions in the MD training simulations, which used atoms beyond the first neighbors of the vacancy site. Based on these assumptions, two model configurations were used for proof-of-concept models in this work: (i) first vacancy nearest neighbours with interaction (1-NNI), for which the atomic jump from the first nearest-neighbor of the vacancy to the empty lattice site depends only on the configuration and type of the first nearest-neighbour atoms of the vacancy, and (ii) first and second vacancy nearest neighbors with interaction (2-NNI), for which the atomic jump from the first nearest-neighbor of the vacancy to the empty lattice site depends on the configuration and type of both the first and second graph nearest-neighbor atoms of the vacancy.

In prior work, a proof-of-concept synthetic dataset was used to test the feasibility of different models under ideal assumptions, using graph templates extracted from MD data that matched the topologies of the first and second graph neighbors of the vacancy defect.⁴⁶ This work employed the same types of networks used for the proof-of-concept tasks, trained on MD data instead of topologically similar facsimiles. The MD-trained networks were used to generate vacancy defect trajectories in conjunction with *Python* code based on the KMC process, in a setup dubbed Evolved System via Machine Learning (EvoSys) (discussed in Subsection II F). The ML-driven trajectories were evaluated using several standard diffusion parameters as performance metrics, including jump rate and atomic squared displacement (ASD) (discussed in Sec. III).

A. Molecular dynamics

Defect simulations were performed using the large-scale atomic/molecular massively parallel simulator (LAMMPS) MD software⁴⁷ and processed using Open Visualization Tool (OVITO).⁴⁸ MD trajectories were processed through the workflow illustrated in Fig. 1. The LAMMPS simulations were conducted at a temperature of 1500 K in order to expedite the construction of the training set for the proof-of-concept network by increasing the rate of vacancy diffusion (to roughly 1×10^{-2} – 1×10^{-3} transitions/fs), which is normally a rare process. Atom types were assigned equiatomic proportions, and interatomic interactions were described using the embedded atom method (EAM). The interatomic potential was sourced from a model developed by Farkas *et al.* for solid solutions strongly resembling realistic FCC HEA behavior.⁴⁹

The MD setup on LAMMPS produced a FCC HEA system containing 864 atoms of five types (Fe, Ni, Cr, Co, and Cu). To create a vacancy, a single atom was deleted from the system, after which the system was allowed to evolve in steps of 1 fs. Data were exported at intervals of 100 fs. Training trajectories were extended incrementally to build up the datasets described in Table I. Baseline MD trajectories for comparison with EvoSys trajectories were run using a nearly identical setup to the training trajectories, only with different seeds for their initial construction. The training and baseline datasets were each composed of 10 trajectories started from different initial compositions. The simulation space in all cases was chosen to be 863 atoms in an FCC crystal environment containing a single vacancy defect.

TABLE I. Counts for initial frames present in each dataset, as well as defect states exported post-processing, and the unique compositions visited by the processed trajectories. For EvoSys results, the only reduction in size from post-processing was due to dropping the final frame of each trajectory. The final column indicates the rounded average of frames per transition event for each dataset.

| Dataset | MD train | MD base | EvoSys 1-NNI | EvoSys 2-NNI |
|--------------|-----------|-----------|--------------|--------------|
| Frames | 9 100 010 | 8 000 010 | 69 127 | 58 446 |
| Defects | 74 662 | 61 953 | 69 117 | 58 436 |
| Unique | 1 180 | 1 141 | 1 406 | 1 203 |
| Frames/Event | 122 | 129 | 1 | 1 |

All ML-driven MD trajectories were set to a maximum runtime of 1×10^8 fs (a value of 1×10^6 saved steps of 100 fs increments) in order to reasonably approximate the MD trajectory runtimes in terms of the total number of defect transitions captured.

In order to interactively view and post-process MD data, OVITO was used, producing the datasets detailed in Algorithm 3. In the conceptual case of multiple vacancies in the system, a *Cluster analysis* modifier would be appropriate as well to tag atoms belonging to each vacancy region-of-interest. The vacancy defects isolated from the MD data this way were converted into a graph representation for use with the neural network models.

B. Nudged elastic band (NEB) calculations

To complement the MD-based defect trajectories and provide an independent estimation of vacancy migration barriers, NEB calculations were performed in a simulation setup consistent with the Fe, Ni, Cr, Co, and Cu HEA system described in Sec. II A. A dataset of 5000 randomly generated atomic configurations was constructed to capture the statistical variability in local atomic environments characteristic of HEA systems. For each configuration, NEB calculations estimated migration energy barriers for vacancy transitions to each of the 12 first nearest neighbor sites, categorized by atom type. These calculations enabled extraction of atomic-type-dependent transition probabilities for vacancy migration.

To account for uncertainties arising from statistical sampling of the local environments, a bootstrap resampling method was applied⁵⁰ to the distribution of calculated barriers, providing symmetric upper and lower bounds through averaging. The resulting trends in barrier heights and derived transition probabilities were subsequently used for comparison with those generated by the GCN-based models.

C. Graph representation

The vacancy defect region-of-interest was represented by an undirected graph due to the intuitive visualization of the data and feature versatility of the graph format (i.e., graphs, edges, and nodes can store multiple feature types and be updated relatively easily). In addition, system-sized graphs have the benefit that periodic boundary conditions can be incorporated into them simply by adding the required edges to boundary nodes, looping connections over the simulation cell.

30 August 2025 06:05:26

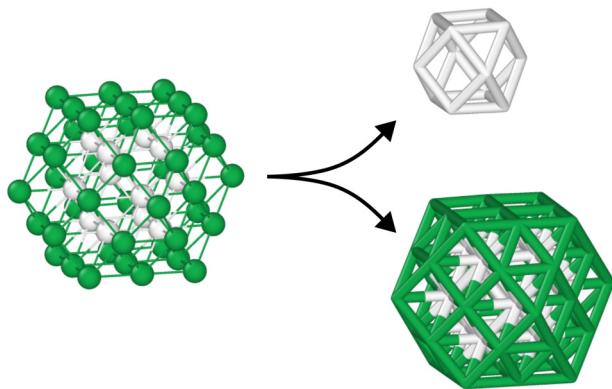


FIG. 2. Configurations of FCC regions of interest for the 1-NNI (upper right) and 2-NNI (lower right). In the 1-NNI model, the environment consists of 12 nearest-neighbor atoms of the vacancy, while in the 2-NNI model, the environment includes 54 atoms, incorporating both the first and second graph neighbors of the vacancy. The 12 allowed transition target atoms are colored in white.

To capture a larger environment around the vacancy defect for the 2-NNI setup, the initial neighbors of the vacancy were expanded by a spherical search space of 3 Å to include the second graph neighbors of the vacancy space. For more details on graph construction, see Fig. 2 and Appendix B.

When converted into graphs, the vacancy 1-NNI and 2-NNI setups had the topologies shown in Fig. 2, and each node featured a one-hot encoding [see Fig. 9(b) for details] indicating the element of the atom being represented from the original MD simulation (i.e., A Fe atom would convert to the node feature 1 0 0 0 0).

Vacancy defect atoms and topology were extracted from MD using OVITO modifiers, as shown in Fig. 2 and described in Appendix B, which created graph neural network-compatible graphs.

D. Neural networks

A GCN model with one *GraphConv* convolutional layer was implemented (Fig. 3). For the 2-NNI case, only the 12 core atoms around the vacancy contributed to k_{tot} , as per the assumptions made earlier in this work. Rate predictions for the rest were adjusted to zero to prohibit them as transition targets. This network was the same used in a proof-of-concept from related prior work.⁴⁶ The network had a depth of three layers in total and used the dynamics learner loss function based on log-likelihood estimation from Subsection II E.⁴² The convolutional layer *GraphConv* allowed emphasis to be placed on the relationship between the central host node and the peripheral nodes instead of between a mix of equivalent nodes.⁵¹ In this convolution, the central node \mathbf{x}_i and the neighborhood nodes \mathbf{x}_j with edge weights $e_{j,i}$ have their own weight matrices W_1 and W_2 , respectively [Eq. (1)].⁵¹ The physical interpretability demonstrated by the designers of the *CGConv* technique served as a guideline for obtaining network outputs that map output properties of a

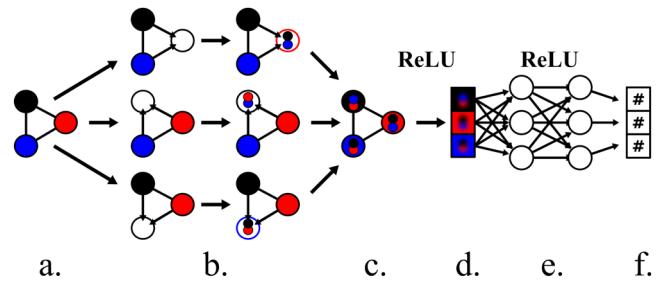


FIG. 3. Network structure used in this work: (a) A graph input, (b) a convolutional layer that collects information on the host and neighbor nodes [white and black/blue/red in (b) respectively], (c) the output graph features and ReLU activation, (d) flattened node features from graph, (e) a linear layer with ReLU activation, and (f) an output layer that returns a single float value prediction for the rate constant per input atom/node. The tricolored graph nodes are stand-ins for one-hot encoded atom features such as those shown in Fig. 9(b).

crystalline input graph to the local environment(s) of each node,¹²

$$\mathbf{x}'_i = \mathbf{W}_1 \mathbf{x}_i + \mathbf{W}_2 \sum_{j \in \mathcal{N}(i)} e_{j,i} \cdot \mathbf{x}_j. \quad (1)$$

E. Dynamics learner

To obtain the dynamics most likely to have generated observed trajectories, the training loss function was based on the work of Casert *et al.* and used the log-likelihood of a trajectory as a source of feedback to the ML model.⁴² This approach assumed the trajectories could be cast as continuous time Markov chains and was used with a few modifications (see Algorithm 1): (i) training was done on batches and did not include a fine-tuning stage over the full trajectory, and (ii) an approximation of the log-likelihood function was used in which the final time terms were omitted to reduce complexity caused by accommodating them as special cases in the training process. This resulted in deviation of less than 1% from the final training values for both networks in this work.

30 August 2025 06:05:26

ALGORITHM 1. The unsupervised dynamics learner loss function used in this work, adapted from Ref. 42. The log-likelihood U of a trajectory ω composed of K states C with transition rates W is given by $U_\omega = \sum_{k=0}^{K-1} (\ln W_{C_k \rightarrow C_{k+1}} - \Delta t_{C_k} R_{C_k})$, where R_{C_k} is the total escape rate, given by $R_{C_k} = \sum_{C \neq C_k} W_{C_k \rightarrow C}$. The adapted version required a trained *model*, *transition times*, the indices for each escape *idxs*, the graph node features x , graph edges, and the nodes per graph n .

```

1: logrates = model(x, edges)
2: rates =  $e^{\textit{logrates}}$ 
3: R =  $\sum$  rates
4: loglike =  $\sum$  (logrates at chosen idxs - (times  $\times$  R))
5: loss = -loglike
6: trainloss += loss
7: Backpropagation and optimization, repeat loop

```

F. Evolving systems via model (EvoSys)

Using an arbitrary trained model configured as shown in Fig. 3, the EvoSys algorithm was built to take predictions of transition and escape rates from a model and use them to produce ML-accelerated trajectories (see Algorithm 2). The EvoSys code used the centroid rather than the center of mass of a vacancy defect when making a transition.

The evolution of the system was set up using KMC rules. KMC has seen use in materials science for a variety of tasks, particularly simulating radiation damage,^{27,30,52} in part due to its applicability to bridge the time scale gap between methods such as MD and longer-term defect evolution (10^{-12} – 10^{-3} s).³⁰ A major assumption of KMC is an infrequent event system, in which the system retains no memory of the previous state after a transition due to remaining in each state for a long time relative to the lattice vibrations.²⁷ This makes the trajectory of states ω formed from KMC a valid Markov chain.

Each of the possible transitions from state i to state j is defined by a rate constant k_{ij} , and the summation for all j is called the *escape rate* k_{tot} . Typically, the rate constants are based on studies of the potential energy landscape between state i and possible state j . KMC is performed by taking a system and defining a finite list of states it could evolve into, as well as the rate at which it is likely to choose each of these states. Then, a Markov chain of states, a trajectory ω , is made by using independent uniform random values to pick a new state at each step based on k_{tot} . Additional details are available in Ref. 27.

Rate constants for different pathways/atoms were scaled to sum to unity by dividing by k_{tot} . The selection of the next state was performed by taking a uniformly distributed random number, and residence/escape time from the state was obtained by drawing an exponentially distributed random variable from a generator parameterized by k_{tot} .

ALGORITHM 2. EvoSys setup to generate synthetic trajectories in MD file format via a hybrid ML-KMC ansatz. Requires a *model*, a trained model state *trained*, a start state *initial_state*.

- 1: Load *model* with saved state *trained*
- 2: Create graph representation of system
- 3: Extract bond and atom data from *initial_state*
- 4: Add nodes and edges to blank graph G using OVITO data
- 5: Find the vacancy defect(s)
- 6: Apply modifiers from Appendix B Algorithm 5 steps 2–5
- 7: Use cluster analysis to find atoms and obtain centroid
- 8: Run *init_graph_vac()*
- 9: Create defect graph G_{defect} based on initial G
- 10: Delete G_{defect} nodes with atom IDs not in region
- 11: Split G_{defect} into graphs for each defect, *defect_graphs*
- 12: **for** defect region i in *defect_graphs* **do**
- 13: Feed i to *model* and produce prediction
- 14: **if** no collision with another defect is detected **then**
- 15: Run updates to i , G , and *initial_state*
- 16: **end if**
- 17: **end for**

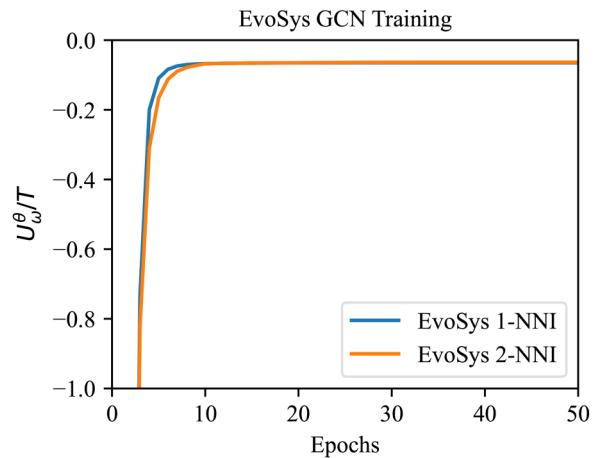


FIG. 4. Log-likelihood loss plots for both EvoSys training setups. Training runs were plotted together as the unknown true dynamics of the MD system were the same in both setups, enabling direct comparison.

For EvoSys trajectories, the minimization process parameters and force field were identical to those used in the MD simulations, minus the system construction steps. Instead, the initial system state for each trajectory came from the matching MD trajectory finalized files [step (C) in Fig. 1].

In this work, the use of only one defect in the system at a time was selected for the sake of easing data generation and training. Each EvoSys trajectory was initialized using an arbitrary start state taken from the beginning of an MD baseline dataset trajectory.

30 August 2025 06:05:26

III. RESULTS

A. Training and transition atom choices

The training log-likelihood loss plots for the 1-NNI and 2-NNI network setups (Fig. 4) were similar in their overall shape, with the EvoSys 2-NNI model taking longer to plateau than the 1-NNI model. This asymmetry in training loss is attributed to the larger graph size and complexity of the 2-NNI setup. The final log-likelihood loss values were -0.0649 for 1-NNI and -0.0634 for 2-NNI.

To compare the learned dynamics trajectories generated via ML, transition atom data were extracted from each trajectory dataset. These data are shown in Table II as the per-atom mean transition proportions for each setup. For information on the number of states in each dataset and the unique states present, refer to Table I.

The ML trajectories both generally matched the trends of the MD trajectories, with the 2-NNI setup being closer to the MD transition atom averages seen in Table II. The MD datasets both displayed a greater abundance of Cu transition events compared to the ML counterparts, which showed an elevated preference for each of the remaining atoms. The proportions of atoms chosen were relatively consistent between the MD baseline and training trajectories with small variations, attributed to the vacancy defects using the

TABLE II. Proportions (%) of atoms chosen in each trajectory dataset, rounded to two decimal places.

| Dataset | MD training | MD baseline | EvoSys 1-NNI | EvoSys 2-NNI | NEB |
|---------|------------------|------------------|------------------|------------------|------------------|
| Fe | 0.69 ± 0.18 | 0.84 ± 0.20 | 2.50 ± 0.30 | 1.52 ± 0.21 | 1.12 ± 0.14 |
| Ni | 4.87 ± 0.58 | 5.01 ± 0.97 | 7.37 ± 0.66 | 5.87 ± 0.91 | 3.77 ± 0.34 |
| Cr | 12.72 ± 1.27 | 13.44 ± 1.48 | 14.32 ± 0.96 | 15.07 ± 2.15 | 10.33 ± 0.95 |
| Co | 4.64 ± 0.72 | 4.91 ± 0.51 | 8.13 ± 0.71 | 6.78 ± 0.92 | 5.47 ± 0.47 |
| Cu | 77.07 ± 1.69 | 75.80 ± 1.88 | 67.68 ± 1.51 | 70.76 ± 3.40 | 79.31 ± 1.46 |

same simulation parameters aside from the seed values. The EvoSys 1-NNI trajectory displayed a noticeably increased global fraction of Fe atom transitions, roughly double or more the next leading value, which belonged to the EvoSys 2-NNI trajectory. The specific values obtained are presented in Table II. The EvoSys 2-NNI trajectory was the closest ML-driven comparison to the true MD data, being slightly reduced in Cu transitions and having an elevated proportion of each of the other elements.

The transition atom data are also presented in Fig. 5 as the proportion of each element selected during a transition with respect to their concentration in the 12-atom region of interest. This form was chosen in order to observe how the transitions of each element changed as their fraction in the composition of the local vacancy environment increased. The most common transition atom was Cu, with compositions containing only a single Cu atom choosing it in roughly 40%–50% of their transitions. As the local fraction of Cu increased, the transition proportion peaked fairly early in the EvoSys 1-NNI setup, with little change after 4 atoms of Cu were present. Though the general trend of the other plots was

similar, they retained a steeper slope with respect to Cu at high concentrations. Interestingly, all the plots indicate a slight decrease in the average transition preference for Cu at a fraction of 8/12, with the exception of the EvoSys 1-NNI setup, which lacked data on that composition space for Cu.

Even at high local concentrations, the other elements in the simulated HEA trajectories did not generally achieve this dominance over the diffusion trend, with Cr and Co being the next most common choices, respectively. The behavior of Cu in this system is as expected, given the strong affinity of copper for vacancies in other alloys.^{53,54}

The least common element as a transition atom was Fe, with the EvoSys 1-NNI and MD Baseline trajectories choosing it slightly more frequently compared to their counterpart datasets. For the EvoSys 1-NNI setup, Fe was encountered at local concentrations making up to 7/12 of the vacancy region, which matches the observed elevated Fe seen in Table II. The rate of change in Fe transitions was most similar in the MD training and EvoSys 2-NNI setups, and the EvoSys 1-NNI setup showed a noticeably

30 August 2025 06:05:26

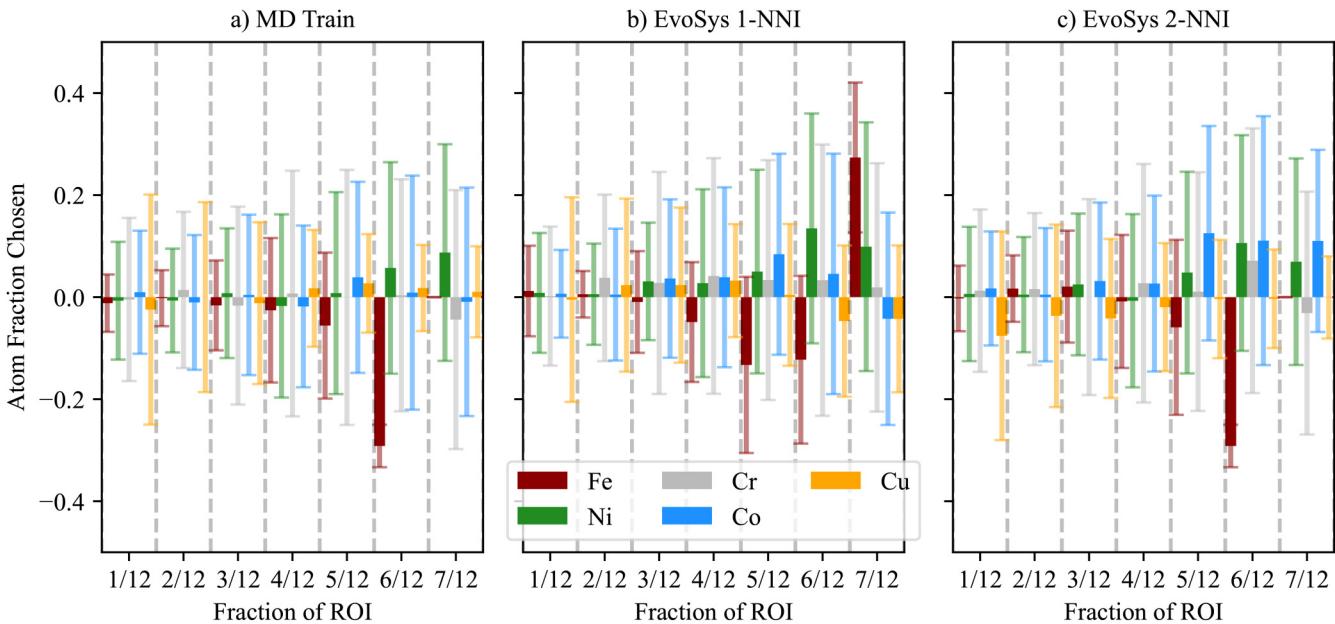


FIG. 5. Mean transition preferences per-atom as a function of their fraction of the vacancy 1-NNI composition for each dataset (i.e., how often element “X” jumped when it occupied “Y”/12 of the allowed target atoms). Showing data for (a) MD training, (b) EvoSys 1-NNI, and (c) EvoSys 2-NNI datasets. Values presented in terms of difference with the MD baseline dataset.

diminished slope compared to the other results. Additional details for the composition orderings are found in Appendix C.

B. Runtime comparisons

To determine the timing difference between trajectories produced from MD simulations and those produced by EvoSys, trajectory re-runs were obtained at set simulation timespans using the same seeds as their original counterparts. The resulting plots of the time scaling and number of transitions can be seen in Figs. 6 and 7. The final average data values for each figure are presented in Table III.

Realtime evaluation (Fig. 7) indicated that the EvoSys trajectories were both generated approximately two orders of magnitude faster than MD trajectories covering an equivalent simulated timespan. This result supported the initial hypothesis that the hybrid graph-ML EvoSys setups would be able to perform the vacancy defect diffusion at an accelerated pace compared to the conventional MD approach. The number of transitions generated per simulated timespan (Fig. 6) indicated that for a given timespan the MD results displayed greater variation and generally produced more transitions compared to EvoSys. Furthermore, the EvoSys results in the same plot showed that though the 1-NNI setup displayed the least variation of the three methods, the average number of transitions produced was closer to the average MD value than the 2-NNI result.

The MD trajectories included more transition events on average than either EvoSys setup, though there was overlap between all three ranges. This may be a consequence of the increased average affinity for non-Cu elements remarked in Subsection III A and the increased average residence times of the EvoSys models, particularly for the 2-NNI setup.

C. Vacancy diffusion parameters

Jump rates were extracted from the diffusion data, and the final cumulative averages for each setup are recorded in Table IV. Of the

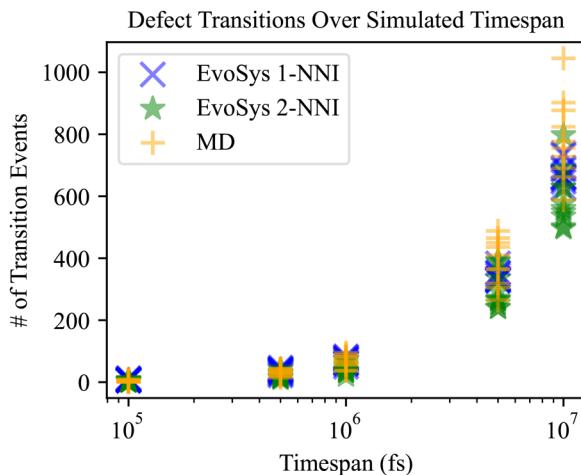


FIG. 6. Number of transition events detected as a function of simulated time for EvoSys 1-NNI and 2-NNI and MD Baseline setups. Each plot shows the results of 10 trajectories.

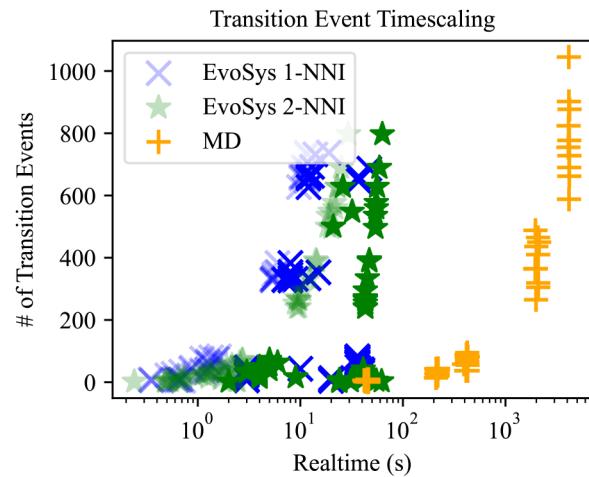


FIG. 7. Number of transition events detected as a function of simulation realtime for the EvoSys 1-NNI and 2-NNI and MD (Baseline) setups. Faded EvoSys points indicate timing reported via from inside the Python script, in contrast to times reported from the bash script, which are presented in solid points.

average jump rates, EvoSys 1-NNI showed the lowest deviation, and the mean was closest to the pair of MD data results. The reduced deviation indicated the possibility that the 1-NNI setup learned to mimic the mean dynamics rather than the true dynamics of the MD system. The EvoSys 2-NNI setup had a greater deviation compared to the MD results, and the lowest mean jump rate, with relatively little overlap with the MD trajectories. As with the transition atom proportions, the MD training and baseline data overlapped closely.

The atomic squared displacement (ASD) from the origin of the vacancy defect was also extracted and is plotted in Fig. 8. The EvoSys 1-NNI trajectory ASD showed a large standard deviation and diffusion value compared to all other trajectories. The variation seen in the 1-NNI plot may indicate a reduced capacity to learn the system dynamics compared to the 2-NNI setup.

The residence times of each dataset were similar between the two MD setups and the two EvoSys setups, with the MD times being smaller and more in agreement with each other. The EvoSys 2-NNI displayed the greatest average residence time, as seen in Table IV.

The diffusion coefficient D was estimated using a linear fit to the ASD data for each trajectory type in Fig. 8. The anomalous diffusion exponent α was also estimated via fitting the anomalous diffusion equation in 3D, $MSD = 6D_\alpha \tau^\alpha$ (obtained from Ref. 55). The fittings

TABLE III. Final timing-related averages with standard deviations for the performance of various models, rounded to the nearest integer value (as shown in Figs. 6 and 7). To generate trajectories for timing, the corresponding original seeds for each dataset were used.

| Dataset | Period (fs/event) | Transitions ^a | Realtime (s) |
|--------------|----------------------|--------------------------|-----------------|
| MD Baseline | $13\ 065 \pm 2\ 050$ | 785 ± 126 | $4\ 166 \pm 31$ |
| EvoSys 1-NNI | $14\ 827 \pm 728$ | 676 ± 34 | 11 ± 1 |
| EvoSys 2-NNI | $17\ 126 \pm 2\ 291$ | 596 ± 88 | 22 ± 3 |

^aRefers to transitions obtained over the longest testing time of 1×10^7 fs.

TABLE IV. Mean jump rate, ASD, and mean residence time performance metrics for each trajectory dataset. Reported as mean and standard deviation of 10 trajectories in each dataset.

| Dataset | Jump (10^{-1} fs $^{-1}$) ^a | ASD (10^4 Å 2) ^b | Time (10^2 fs $^{-1}$) ^c |
|--------------|---|---------------------------------------|--|
| MD Training | 8.37 ± 0.60 | 1.56 ± 1.06 | 119 ± 171 |
| MD Baseline | 7.91 ± 0.95 | 1.09 ± 0.50 | 126 ± 186 |
| EvoSys 1-NNI | 6.90 ± 0.14 | 2.56 ± 1.47 | 145 ± 155 |
| EvoSys 2-NNI | 5.84 ± 0.95 | 1.52 ± 1.02 | 171 ± 245 |

^aTaken as cumulative average with a step size of 10 ps.

^bTaken as cumulative average with a step size of 10 ps.

^cRounded to integer values in units of 100 fs due to the MD parameters detailed in Subsection II A.

returned values shown in Table V. The small values for D were in keeping with the rarity of transitions relative to the timespan of the MD simulations. From the fitted parameters, the EvoSys 1-NNI system showed the greatest diffusion coefficient while those for the other systems were smaller and similar to each other, as expected from the steepness of the curves in Fig. 8. The reduced variability of transition period and transition count for EvoSys 1-NNI seen in Table III, combined with the comparatively short period and a high average number of transitions, may explain the higher value of D as the result of many transitions at consistent intervals. This could indicate against a good fit of the EvoSys 1-NNI model to the host MD system.

Values from fitting α that are less than 1 indicate a subdiffusive trajectory, a form known to occur for continuous time random walks where crowded or trapping environments can be present.⁵⁵ This case was recorded for all the ASD data except the 2-NNI setup, which was fitted with an α value near 1, indicative of Brownian motion. This suggests that the EvoSys 2-NNI setup may not be making transition choices more erratically for one or more atom types than the true MD dynamics. Given the large and overlapping deviations of the single-particle trajectories in Fig. 8, additional trial trajectories

TABLE V. Estimated diffusion values for the ASD of each trajectory dataset. Values rounded to two decimal places.

| Dataset | D (10^{-5} Å 2 fs $^{-1}$) | α |
|--------------|-------------------------------------|----------|
| MD Training | 2.85 | 0.80 |
| MD Baseline | 2.17 | 0.58 |
| EvoSys 1-NNI | 5.13 | 0.85 |
| EvoSys 2-NNI | 2.91 | 1.06 |

may make the trends clearer in future analyses. Transition probabilities, derived from the NEB-calculated barriers for each atomic species, showed good agreement with those inferred from the MD-based trajectories and the EvoSys GCN predictions. In particular, the trends in species-specific vacancy migration, notably the relative preferences of Fe, Ni, and Cu atoms for vacancy transitions, were consistent between NEB-derived probabilities and those observed in the MD data. This agreement supports the reliability of the GCN-based approach in capturing atomistic dynamics governing diffusion in complex alloys. Further details of the NEB analysis and transition probability estimation are provided in Appendix E.

IV. CONCLUSION

In this work, a proof-of-concept, EvoSys, was demonstrated as a surrogate model of vacancy defect diffusion in HEAs, using two different interaction setups. Assessment of model performance was predicated on two main points: The accuracy of the trajectories generated via surrogate model setups and the acceleration of defect trajectories, both as compared to the MD technique.

Comparison of trajectory metrics with conventional MD indicated that both interaction modes, 1-NNI and 2-NNI, possessed reduced jump rates and residence times as well as a greater number of transitions involving elements other than Cu. However, the 2-NNI model setup showed a distribution of transition atom choices that was more in line with the MD trajectories. The trajectories produced via the 2-NNI model also demonstrated visually similar diffusion characteristics to the MD observations, unlike the 1-NNI setup. The fitted parameters for diffusion, D and α , suggested that while the diffusion coefficient D for 2-NNI was in agreement with the MD trajectories, the 2-NNI setup was more Brownian than the MD and 1-NNI systems, a subject for further investigation.

In terms of acceleration, the realtime required by the EvoSys surrogate model setups was found to be approximately two orders of magnitude less than the realtime required to simulate an equivalent amount of time using MD.

Determining the accuracy of rate constants predicted by the surrogate model setups was limited in this work by the absence of a set of analytical true values in MD, exacerbated by the configurational complexity of HEAs. To complement the MD-based observations, NEB calculations were performed to independently estimate vacancy migration barriers and associated transition probabilities. The agreement between the NEB-derived trends and those predicted by the surrogate model supports the reliability of the GCN-based approach in capturing the defect diffusion behavior in the HEA system. Summaries of defect diffusion paths proved the most viable alternative. Another consideration that would greatly

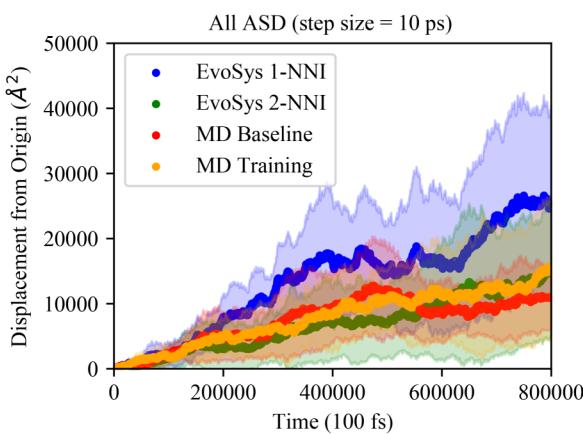


FIG. 8. ASD measured in 10 ps intervals for each dataset, averaged across 10 trajectories. Shaded regions indicate standard deviation. Values are tabulated in Table IV.

30 August 2025 06:05:26

expand the potential for future work with the fixed temperature proof-of-concept model is whether insights can be extrapolated to other temperatures, by using techniques based on temperature-accelerated molecular dynamics.⁵

In this work, MD simulations used a random solid solution setup for the HEA environment, a simplification of the complex reality of HEA composition. As the ML model setup implicitly learns entropic effects on diffusion from the training data, the model and associated workflow should, with sufficient training data and complexity, be able to account for the impact of ordered configurations of the region of interest (ROI) in an environment with SRO or other local effects that alter vacancy transport.

For future work, a closer examination of the surrogate predictions with the MD decisions would be helpful for improving model explainability. One approach that was considered briefly in this work was the use of repeated samplings of randomly perturbed MD states to build up an estimated set of transition rates for comparison. However, this was beyond the scope of this work. As the ML component of the workflow was largely modular, the applicability to defect migration with different networks and materials is an area that is open to additional exploration.

ACKNOWLEDGMENTS

This research was funded by the Natural Sciences and Engineering Research Council of Canada via the Discovery Grants program. This study was partially funded by Atomic Energy of Canada Limited, under the auspices of the Federal Nuclear Science and Technology Program. Work involving NEB calculations was performed with support from the Canada Excellence Research Chairs (CERC) program. Computing resources were provided via Digital Research Alliance of Canada. Work done at the Molecular Foundry was supported by the Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. C.C. was supported by a Francqui Fellowship of the Belgian American Educational Foundation and by the US DOE Office of Science Scientific User Facilities AI/ML project “A digital twin for spatiotemporally resolved experiments.”

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

C. Reimer: Data curation (lead); Investigation (lead); Software (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (lead). **P. Saidi:** Conceptualization (equal); Project administration (equal); Software (supporting); Visualization (supporting); Writing – review & editing (supporting). **C. Casert:** Conceptualization (equal); Writing – review & editing (supporting). **C. Beeler:** Conceptualization (equal); Project administration (equal); Software (supporting); Visualization (supporting); Writing – review & editing (supporting). **C. G. Tetsassi Feugmo:** Conceptualization (equal); Writing – review & editing (supporting). **S. Whitelam:** Conceptualization (equal); Writing – review & editing (supporting). **E. Mansouri:** Software (supporting); Writing – review & editing

(supporting). **A. Martinez:** Software (supporting); Visualization (supporting); Writing – review & editing (supporting). **L. Beland:** Writing – review & editing (supporting). **I. Tamblyn:** Conceptualization (equal); Funding acquisition (lead); Project administration (equal); Resources (lead); Supervision (lead).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request and are openly available in GitHub repository “EvoSys-Research-Data-Code,” No. 1025886460, Ref. 57.

APPENDIX A: DATA CLEANING AND EXTRACTION

The first task in this work after constructing MD simulations was the cleaning of MD files to produce datasets. This was performed using the process illustrated in Fig. 1 and detailed in Algorithm 3, which reduced the number of malformed graphs from roughly ~7% of the data investigated to $\ll 1\%$.

ALGORITHM 3. MD post-processing used to clean and identify each defect state. As long as the time t was less than the limit t_{limit} , MD data were saved every 100 fs, producing n saved files, where $n = t_{limit}/100\text{ fs}$. After minimization, any extracted defects that did not match the bond topology of the ideal vacancy, $template$, were subjected to a looped re-processing in which incrementally larger bond lengths, $bond_len$, were used up to a user-defined threshold value, $bond_limit$.

```
1: while  $t < t_{limit}$  do
2:   Evolve system in time,  $t$ 
3:   if every 100\,fs then
4:     Export MD system data to file
5:   end if
6: end while
7: convert all  $n$  exported files to data files
8: LAMMPS minimization of all  $n$  data files
9: extract final frame of all  $n$  minimizations
10: for  $i$  in range( $n$ ) finalized MD frames do
11:   Process  $i$  in OVITO to detect defect, timestep, frame num.
12:   if vacancy defect detected then
13:     if OVITO topology of  $i$  matches  $template$  then
14:       Process  $i$  in OVITO to get  $topology$ ,  $elements$ ,  $IDs$ 
15:     else
16:       file  $i$  listed as noncompliant
17:     end if
18:   end if
19: end for
20: while  $bond\_len < bond\_limit$  do
21:    $bond\_len$  increment by 0.1 Å
22:   for  $j$  in noncompliant do
23:     if OVITO topology of  $j$  matches  $template$  then
24:       Process  $j$  in OVITO to get  $topology$ ,  $elements$ ,  $IDs$ 
25:       remove  $j$  from noncompliant
26:     end if
27:   end for
28: end while
29: Export remaining noncompliant to .txt file
```

APPENDIX B: TOPOLOGY AND REPRESENTATION

The bond topology of the region of interest was similar to the example shown in Fig. 9(a) and was converted into NetworkX (NX) graph representation using Algorithms 4 and 5.

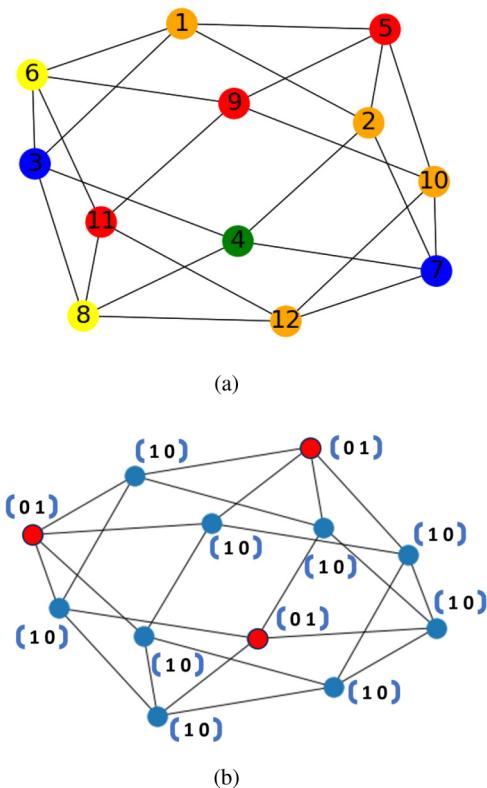


FIG. 9. NX graph representations of the vacancy defect 1-NNI environment: (a) A layout captured for an arbitrary vacancy state, showing node colors based on features (atom types) and node IDs from number tags assigned by OVITO; (b) an arbitrary vacancy graph layout for a binary alloy demonstrating the one-hot encoding scheme.

The graph structure of an arbitrary MD graph was taken as a template topology for both environments studied (see Fig. 2). In order to put similar MD graphs containing n nodes each with k node features and e edges with f edge features into the networks in Subsection II D, they were converted into $n \times n$ adjacency matrices, $n \times k$ node feature matrices, and $e \times f$ edge feature matrices. Node features were converted as indicated in Fig. 9. Graph construction is detailed in Algorithms 4 and 5.

For Algorithm 5, the OVITO modifiers for expanded selection range and universal bond cutoff used a value of 3 Å. The CNA modifier was used with the *Other* option selected in order to isolate atoms in the region of interest as they did not form an FCC crystal structure.

ALGORITHM 4. Construction of a list G_list of n graphs G from n MD vacancy states (data extracted with Algorithm 5) for GCN models, using Python OVITO and NX packages.

```

1: Extract Particle_identifier, Particle_type, Topology with OVITO
   via Algorithm 5
2: Initialize G_list to store graphs
3: for  $p$  in range( $n$ ) do
4:   initialize empty graph  $G$ 
5:   graph_data dict from Particle_identifier, Particle_type
6:   add nodes to  $G$  from graph_data
7:   extract edges from Topology[ $p$ ] to edge_data
8:   add edges from edge_data to  $G$ 
9:   save  $G$  in G_list
10: end for return G_list
```

ALGORITHM 5. Extraction of topological details of the i th of n vacancy defect environments, as performed in this work. Steps 2–5 are intended to be applied as an OVITO pipeline modifier stack. Images shown for reference in Fig. 2. Skipping step 3 allows extraction of just the 1-NNI region.

```

1: Open defect file in OVITO
2: CNA  $\xrightarrow{\text{Selecttype}(Other)}$ 
3: Expand selection
4: Invert selection
5: Delete selected
6: Apply bonds
7: Save Topology from Bond table
```

30 August 2025 06:05:26

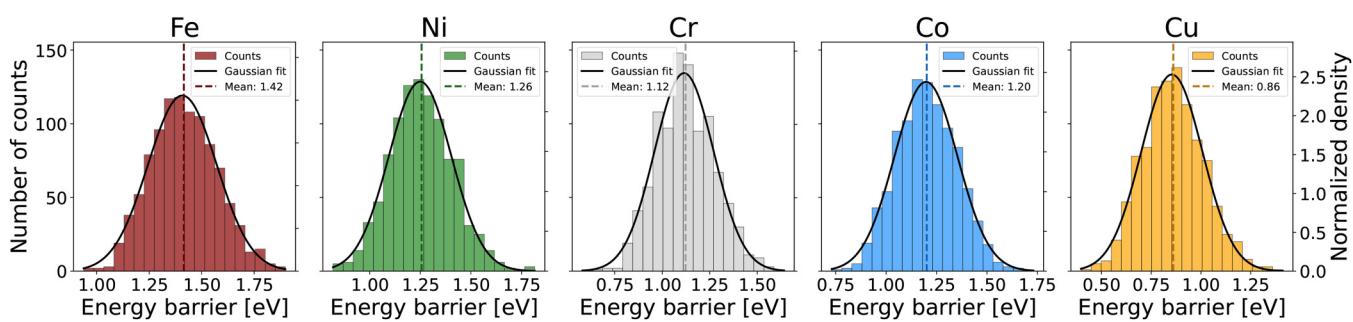


FIG. 10. Histograms of vacancy migration energy barriers for each elemental species (Fe, Ni, Cr, Co, and Cu) in the equiatomic FeNiCrCoCu HEA. The distributions represent the frequency of vacancy jump events as a function of energy barrier. Each histogram is fitted with a Gaussian distribution function, and the corresponding mean energy barrier for each element is reported.

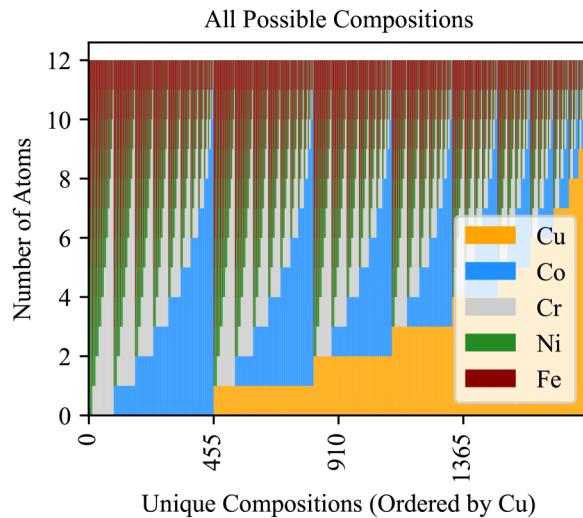


FIG. 11. All possible compositions for the 1-NNI region of 12 atoms surrounding the vacancy defect, sorted by Cu fraction first, then Co, etc.

APPENDIX C: ORDERING COMPOSITIONS

Mean transition preferences per-atom as a function of their fraction of the vacancy 1-NNI composition are shown in Fig. 5. For each unique composition, the sum of jump counts was normalized to unity and then ordered by increasing concentration in the 12 atom 1-NNI region. Finally, the mean was taken with respect to each element for transitions with the same elemental fraction. Proportions outside the range 1/12 to 8/12 were not reported, as these states were visited too infrequently to record meaningful statistics. Single element transitions and zero-valued entries were eliminated from the data. See Fig. 11 for an example of the mapping used to order compositions by increasing Cu content.

APPENDIX D: SIMULATION AND NETWORK SETTINGS

The LAMMPS simulations used the following settings: Units of ps, Å, and eV for time, space, and energy, respectively, as well as a lattice parameter of 3.552 Å, and cell size of 3 units (with periodic boundary conditions). The LAMMPS minimization command values for energy tolerance, force tolerance, maximum number of iterations, and the maximum number of evaluations were set to 1×10^{-4} , 1×10^{-7} eV/Å, 1000, and 10 000, respectively.⁵⁶ Trajectories were resumed from checkpoints using the *restart* function in LAMMPS in order to generate trajectories of approximately 8×10^5 data points each.

The GCN network training used hyperparameters of 50 epochs, a learning rate of 2.5×10^{-4} , and a batch size of 250. Seeds were fixed for training both the 1-NNI and 2-NNI versions of the GCN model.

APPENDIX E: NEB ENERGY BARRIERS

The probability of a vacancy jump involving an atom of species k was estimated by constructing a partition function where

the phase space is defined by all saddle point configurations associated to k . For a given set of migration barriers $E_m^{(i)}$ obtained from NEB calculations, as shown in Fig. 10, a partition function for species k jumps can be written as

$$Z_k = \sum_{i \in k} N_k e^{-\beta E_m^{(i)}}, \quad (\text{E1})$$

where N_k is the number of jumps available for species k (in our case, 1000 for each species), $\beta = 1/(k_B T)$, with k_B being the Boltzmann constant, and T is the absolute temperature.

The normalized probability ρ_k that a vacancy jump involves species k is then calculated as

$$\rho_k = \frac{Z_k}{\sum_{\ell} Z_{\ell}}, \quad (\text{E2})$$

where the sum runs over all atomic species ℓ in the system (in this case, 5).

REFERENCES

- ¹S. Zinkle, “Advanced irradiation-resistant materials for generation IV nuclear reactors,” in *Structural Materials for Generation IV Nuclear Reactors* (Elsevier Ltd, 2017), Chap. 16, pp. 569–594.
- ²E. J. Pickering, A. W. Carruthers, P. J. Barron, S. C. Middleburgh, D. E. J. Armstrong, and A. S. Gandy, “High-entropy alloys for advanced nuclear applications,” *Entropy* **23**, 98 (2021).
- ³Y. Osetsky, A. V. Barashev, and Y. Zhang, “Sluggish, chemical bias and percolation phenomena in atomic transport by vacancy and interstitial diffusion in Ni Fe alloys,” *Curr. Opin. Solid State Mater. Sci.* **25**, 100961 (2021).
- ⁴Z. H. Aitken, V. Sorkin, and Y.-W. Zhang, “Atomistic modeling of nanoscale plasticity in high-entropy alloys,” *J. Mater. Res.* **34**, 1509–1532 (2019).
- ⁵R. J. Zamora, D. Perez, and A. F. Voter, “Speculation and replication in temperature accelerated dynamics,” *J. Mater. Res.* **33**, 823–834 (2018).
- ⁶B. Xu, J. Zhang, S. Ma, Y. Xiong, S. Huang, J. Kai, and S. Zhao, “Revealing the crucial role of rough energy landscape on self-diffusion in high-entropy alloys based on machine learning and kinetic Monte Carlo,” *Acta Mater.* **234**, 118051 (2022).
- ⁷B. Xing, X. Wang, W. J. Bowman, and P. Cao, “Short-range order localizing diffusion in multi-principal element alloys,” *Scr. Mater.* **210**, 114450 (2022).
- ⁸G. R. Schleder, A. C. M. Padilha, C. M. Acosta, M. Costa, and A. Fazzio, “From DFT to machine learning: Recent approaches to materials science—a review,” *J. Phys. Mater.* **2**, 032001 (2019).
- ⁹J. K. Pedersen, T. A. A. Batchelor, A. Bagger, and J. Rossmeisl, “High-entropy alloys as catalysts for the CO₂ and CO reduction reactions,” *ACS Catal.* **10**, 2169–2176 (2020).
- ¹⁰T. A. A. Batchelor, J. K. Pedersen, S. H. Winther, I. E. Castelli, K. W. Jacobsen, and J. Rossmeisl, “High-entropy alloys as a discovery platform for electrocatalysis,” *Joule* **3**, 834–845 (2019).
- ¹¹C. Chen, Y. Zuo, W. Ye, X. Li, Z. Deng, and S. P. Ong, “A critical review of machine learning of energy materials,” *Adv. Energy Mater.* **10**(8), 1903242 (2020).
- ¹²T. Xie and J. C. Grossman, “Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties,” *Phys. Rev. Lett.* **120**, 145301 (2018).
- ¹³J. Damewood, J. Karaguesian, J. R. Lunger, A. R. Tan, M. Xie, J. Peng, and R. Gómez-Bombarelli, “Representations of materials for machine learning,” *Annu. Rev. Mater. Res.* **53**, 399–426 (2023).
- ¹⁴C. Li, W. Wei, J. Li, J. Yao, X. Zeng, and Z. Lv, “3DMol-Net: Learn 3D molecular representation using adaptive graph convolutional network based on rotation invariance,” *IEEE J. Biomed. Health* **26**, 5044–5054 (2022).

- ¹⁵L. Cian, G. Lancioni, L. Zhang, M. Ianese, N. Novelli, G. Serra, and F. Maresca, "Atomistic graph neural networks for metals: Application to BCC iron," *arXiv:2109.14012* [cond-mat.mtrl-sci] (2021).
- ¹⁶T. Zheng, X. Hu, F. He, Q. Wu, B. Han, D. Chen, J. Li, Z. Wang, J. Wang, J. jung Kai, Z. Xia, and C. Liu, "Tailoring nanoprecipitates for ultra-strong high-entropy alloys via machine learning and prestrain aging," *J. Mater. Sci. Technol.* **69**, 156–167 (2021).
- ¹⁷W. Huang, P. Martin, and H. L. Zhuang, "Machine-learning phase prediction of high-entropy alloys," *Acta Mater.* **169**, 225–236 (2019).
- ¹⁸C. G. Tetsassi Feugmo, K. Ryczko, A. Anand, C. V. Singh, and I. Tamblyn, "Neural evolution structure generation: High entropy alloys," *J. Chem. Phys.* **155**, 0 (2021).
- ¹⁹W.-M. Choi, Y. H. Jo, S. S. Sohn, S. Lee, and B.-J. Lee, "Understanding the physical metallurgy of the CoCrFeMnNi high-entropy alloy: An atomistic simulation study," *NPJ Comput. Mater.* **4**, 1 (2018).
- ²⁰L. Nibbelink, "Simulating vacancy formation and diffusion in NbMoTaW," M.S. thesis (University of California, San Diego, 2020).
- ²¹W. Yang and Y. Wu, "Effects of annealing on the micro-internal stress induced by interstitial defects in aluminum crystal by molecular dynamics simulations," *AIP Adv.* **12**, 025226 (2022).
- ²²S. A. Starikov, A. R. Kuznetsov, and V. V. Sagaradze, "Crowdion in deformed FCC metal: atomistic modeling," *Phys. Met. Metallogr.* **122**, 1207–1212 (2021).
- ²³S. Bukkuru, U. Bhardwaj, K. S. Rao, A. D. P. Rao, M. Warrier, and M. C. Valsakumar, "Kinetics of self-interstitial migration in BCC and FCC transition metals," *Mater. Res. Express* **5**, 035513 (2018).
- ²⁴S. P. Fitzgerald, "Structure and dynamics of crowdion defects in BCC metals," *J. Micromech. Mol. Phys.* **03**, 1840003 (2018).
- ²⁵D. R. Mason, A. E. Sand, and S. L. Dudarev, "Atomistic-object kinetic Monte Carlo simulations of irradiation damage in tungsten," *Model. Simul. Mater. Sci. Eng.* **27**, 055003 (2019).
- ²⁶M. Posselt, F. Gao, and H. Bracht, "Correlation between self-diffusion in Si and the migration mechanisms of vacancies and self-interstitials: An atomistic study," *Phys. Rev. B* **78**, 0 (2008).
- ²⁷A. E. Voter, "Introduction to the kinetic Monte Carlo method," in *Radiation Effects in Solids*, NATO Science Series, edited by K. E. Sickafus, E. A. Kotomin, and B. P. Uberuaga (Springer Netherlands, 2007), Chap. 1, pp. 1–23.
- ²⁸J.-W. Jang, B.-J. Lee, and J.-H. Hong, "Influence of Cu, Cr and C on the irradiation defect in Fe: A molecular dynamics simulation study," *J. Nucl. Mater.* **373**, 28–38 (2008).
- ²⁹A. Ervin and H. Xu, "Mesoscale simulations of radiation damage effects in materials: A SEAKMC perspective," *Comput. Mater. Sci.* **150**, 180–189 (2018).
- ³⁰B. W. C. S. Becquart, "Kinetic Monte Carlo simulations of irradiation effects," in *Comprehensive Nuclear Materials, Five Volume Set*, edited by R. J. Konings (Elsevier Science, 2012), Vol. 1, Chap. 14.
- ³¹L. Qian, H. Bao, R. Li, and Q. Peng, "Atomistic insights of a chemical complexity effect on the irradiation resistance of high entropy alloys," *Mater. Adv.* **3**, 1680–1686 (2022).
- ³²O. K. Orhan, M. Hendy, and M. Ponga, "Electronic effects on the radiation damage in high-entropy alloys," *Acta Mater.* **244**, 118511 (2023).
- ³³P. Saidi, H. Pircazi, M. Sanjari, S. Tamimi, M. Mohammadi, L. K. Béland, M. R. Daymond, and I. Tamblyn, "Deep learning and crystal plasticity: A preconditioning approach for accurate orientation evolution prediction," *Comp. Meth. Appl. Mech. Eng.* **389**, 114392 (2022).
- ³⁴Y. Xia, L. Wu, and G. Wang, "Rapid evaluation method for anisotropic growth of WS₂ monolayers by combining machine learning algorithms and kinetic Monte Carlo simulation data," *Comput. Mater. Sci.* **184**, 109922 (2020).
- ³⁵L. Scotti and A. Mottura, "Interstitial diffusion of O, N, and C in α -Ti from first-principles: Analytical model and kinetic monte carlo simulations," *J. Chem. Phys.* **144**, 084701 (2016).
- ³⁶M. Bracconi and M. Maestri, "Training set design for machine learning techniques applied to the approximation of computationally intensive first-principles kinetic models," *Chem. Eng. J.* **400**, 125469 (2020).
- ³⁷F. J. Domínguez-Gutiérrez, J. Byggmästar, K. Nordlund, F. Djurabekova, and U. von Toussaint, "Computational study of crystal defect formation in Mo by a machine learning molecular dynamics potential," *Model. Simul. Mater. Sci. Eng.* **29**, 055001 (2021).
- ³⁸J. Byggmästar, A. Hamedani, K. Nordlund, and F. Djurabekova, "Machine-learning interatomic potential for radiation damage and defects in tungsten," *Phys. Rev. B* **100**, 0 (2019).
- ³⁹M. Ceriotti, "Beyond potentials: Integrated machine learning models for materials," *MRS Bull.* **47**, 1045–1053 (2022).
- ⁴⁰K. Ryczko, D. A. Strubbe, and I. Tamblyn, "Deep learning and density-functional theory," *Phys. Rev. A* **100**, 022512 (2019).
- ⁴¹K. Ryczko, J. T. Krogel, and I. Tamblyn, "Machine learning diffusion Monte Carlo energies," *J. Chem. Theory Comput.* **18**, 7695–7701 (2022).
- ⁴²C. Casert, I. Tamblyn, and S. Whitelam, "Learning stochastic dynamics and predicting emergent behavior using transformers," *Nat. Commun.* **15**, 0 (2024).
- ⁴³G. D. Savva, R. L. Benson, I. A. Christidi, and M. Stamatakis, "Large-scale benchmarks of the time-warp/graph-theoretical kinetic Monte Carlo approach for distributed on-lattice simulations of catalytic kinetics," *Phys. Chem. Chem. Phys.* **25**, 5468–5478 (2023).
- ⁴⁴B. Xing, T. J. Rupert, X. Pan, and P. Cao, "Neural network kinetics for exploring diffusion multiplicity and chemical ordering in compositionally complex materials," *Nat. Commun.* **15**, 3879 (2024).
- ⁴⁵M. Nastar and F. Soisson, "Radiation-induced segregation," in *Comprehensive Nuclear Materials* (Elsevier Science, 2012), Chap. 18, Vol. 1, pp. 471–496.
- ⁴⁶C. Reimer and I. Tamblyn, "Characterizing structure of high entropy alloys (HEAs) using machine learning," M.S. thesis (University of Ottawa, 2023).
- ⁴⁷A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, "LAMMPS—A flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," *Comput. Phys. Commun.* **271**, 108171 (2022).
- ⁴⁸A. Stukowski, "visualization and analysis of atomistic simulation data with OVITO – the open Visualization tool," *Model. Simul. Mater. Sci. Eng.* **18**, 015012 (2010).
- ⁴⁹D. Farkas and A. Caro, "Model interatomic potentials and lattice strain in a high-entropy alloy," *J. Mater. Res.* **33**, 3218–3225 (2018).
- ⁵⁰A. Davison and D. Hinkley, *Bootstrap Methods and Their Application*, 1st ed. (Cambridge University Press, Cambridge, 1997).
- ⁵¹C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, "Weisfeiler and leman go neural: Higher-order graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence* (Association for the Advancement of Artificial Intelligence, 2019), Vol. 33, pp. 4602–4609.
- ⁵²K. Nordlund, S. Zinkle, A. Sand, F. Granberg, R. Averback, R. Stoller, T. Suzudo, L. Malerba, F. Banhart, W. Weber, F. Willaime, S. Dudarev, and D. Simeone, "Primary radiation damage: A review of current understanding and models," *J. Nucl. Mater.* **512**, 450 (2018).
- ⁵³L. Messina, T. Schuler, M. Nastar, M.-C. Marinica, and P. Olsson, "Solute diffusion by self-interstitial defects and radiation-induced segregation in ferritic Fe-X (X=Cr, Cu, Mn, Ni, P, Si) dilute alloys," *Acta Mater.* **191**, 166–185 (2020).
- ⁵⁴N. Castin, M. Pascuet, and L. Malerba, "Mobility and stability of large vacancy and vacancy–copper clusters in iron: An atomistic kinetic Monte Carlo study," *J. Nucl. Mater.* **429**, 315–324 (2012).
- ⁵⁵C. Schirripa Spagnolo and S. Luin, "Trajectory analysis in single-particle tracking: From mean squared displacement to machine learning approaches," *Int. J. Mol. Sci.* **25**, 8660 (2024).
- ⁵⁶LAMMPS, "minimize command," LAMMPS Documentation (2022).
- ⁵⁷C. Reimer, "EvoSys-Research-Data-Code," No. 1025886460, available at <https://github.com/CLEANit/EvoSys-Research-Data-Code> (2025).