The Generation of Machine-Learned Interatomic Potentials for the Acceleration of Materials Discovery

By

Siona Benjamin

Senior Honors Thesis

Chemistry

University of North Carolina at Chapel Hill

April 27th, 2023

Approved:

Scott Warren, Thesis Advisor

James Cahoon, Reader

Yosuke Kanai, Reader

Senior Honors Thesis – Department of Chemistry
Siona Benjamin

Abstract

The computational prediction and analysis of crystal structures is a vital aspect of materials science, allowing for the rapid discovery of useful compounds. Evolutionary algorithms are popular computational implementations of materials discovery. Often, these algorithms make use of ab initio methods such as density functional theory (DFT) to calculate structural properties and rank structures throughout the algorithm. DFT offers impeccable accuracy because it uses first-principle equations, but this high precision also means that calculations come at a greater cost in terms of computational power and time. To overcome this, machine-learned interatomic potentials have been proposed to analyze and rank structures throughout the materials discovery process. However, developing a model that can accurately analyze structures in an evolutionary search is challenging. To harness the power of machine learning, the deep potential platform DeePMD served as a user-friendly method to produce interatomic potentials. While being user-friendly, model creation still required significant user intervention, which motivated the encapsulation of the DeePMD training process in a series of automated workflows. These workflows, along with a program to randomly create crystal structures for training data, enable the rapid creation of widely applicable deep potential models for the intended use in evolutionary algorithms. Along with evolutionary searches, these workflows can be used to generate potentials for a variety of other calculations, – such as large-scale dynamics simulations, phonon calculations, or defect predictions. With the creation of machine-learned deep potential models, intense calculations such as those mentioned above become significantly more achievable while maintaining the desired levels of accuracy.

Introduction

Material and compound discovery is a fundamental aspect of science, leading to breakthroughs in the fields of pharmaceuticals, renewable energy, and more. However, traditional experimental materials research has been plagued with issues such as lengthy discovery timelines and low probability of success[1]. These issues, coupled with the sheer vastness of the chemical space of useful compounds, have led to the emergence of computational prediction and analysis techniques. Virtual screening (VS)[2] and evolutionary algorithms (EAs) are just two examples of computational methods for discovering promising candidate structures. While VS makes use of the information present in standard chemical databases and has made major contributions to the field of drug discovery, EAs are an example of popular de novo algorithms where novel chemical structures are assembled from scratch[1]. This method has proved especially beneficial in the discovery of new inorganic materials. Evolutionary algorithms are inspired by biological evolution, where top ranked members of a structure

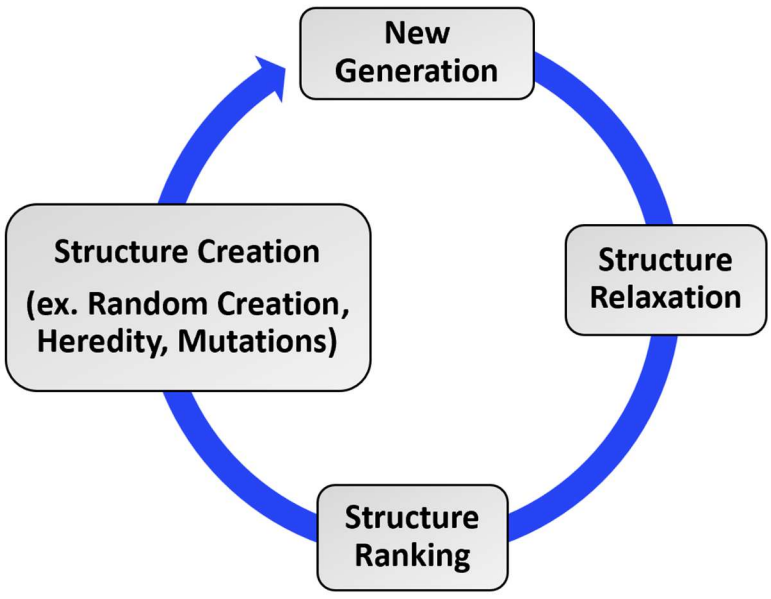


Figure 1: Simplified Representation of the Evolutionary Algorithm.

population will be mutated and proceed to the next generation until the best suited structures remain. Initial populations in an EA can be generated randomly, but it is also common to incorporate known structures in the search. Structures in initial and subsequent populations are ranked by a fitness function and the 'fittest' materials are selected from the population and mutated for next generations[3]. The fitness function used varies depending on the property being searched for, and can include structural energy, conductivity, and more.

Structural optimizations in an EA are crucial for selecting which structures make it to the next steps of the cycle. Not only are these optimizations commonly used as a fitness function to rank structures based on energy, but they also convert randomly created structures to more reasonable candidates. Like all material computational analyses, accurate insight into a material's properties is reliant on a sufficient model for the interatomic forces between atoms in the system[4]. This can be visualized as a potential energy surface (PES), where different points on the surface correspond to different structural configurations and therefore potential energies. One way to construct the PES of a structure is through quantum mechanical principles. Ab-initio methods such as density functional theory (DFT) and Hartree-Fock (HF) derive structural properties from first principles equations such as the Schrödinger equation. Quantum mechanical methods are therefore more accurate than alternative empirical or semiempirical methods. This increase in accuracy also comes with high computational costs, with calculations scaling by a factor of N^3 where N is the number of atoms[5]. Since using DFT geometry optimizations can lead to bottlenecks in an evolutionary search, the use of machine learned (ML) potentials is being explored to speed up the discovery of new materials[6, 7, 8]. Geometry optimizations using machine learned interatomic potentials can be used in place of DFT calculations to increase speed while maintaining DFT levels of accuracy.

The first example of an ML neural network used to describe a PES was developed by Blank et al, and has since led to a plethora of other ML methods, such as the Behler-Parrinello neural network[9], the Gaussian approximation potentials using stochastic processes[10], and Deep potentials (DP)[11]. Deep potentials are an example of neural network potentials and were implemented in the DeePMD-kit in 2017 by Han et al[12]. DeePMD makes use of TensorFlow, an open-source deep learning platform, for efficient training, testing, and evaluation of the PES. With DeePMD, atomic positions are mapped to 'descriptors' of the atom's chemical environment. Descriptors are ultimately constructed from shifting neighboring atoms and lead to the preservation of translational, rotational, and permutational symmetries on the PES. The mapping from atomic positions to their descriptors is denoted by Equation 1[12].

$$D_i = D_i\left(R_i, \{R_j \mid j \in N_{R_C}(i)\}\right) \quad \textbf{(1)}$$

Here, D_i is the descriptor, R_i is the position of the ith atom, and $\{R_j \mid j \in N_{RC}(i)\}$ represents the atom's neighbors. The deep neural network (DNN) that maps the descriptors to atomic energy is represented by Equation 2[12], where E_s represents the atomic energy, N_s represents the neural network, and D_i represents the descriptors.

$$E_{s(i)} = N_{s(i)}(D_i) \quad \textbf{(2)}$$

Figure 2 represents the architecture of the DNN used by DeePMD. The input layer includes the descriptors of each atom in a single system. Hidden layers of the DNN are where the descriptors are mapped to atomic properties before producing the local properties of each atomic site. The final output of the DNN are the overall properties of the system[11].

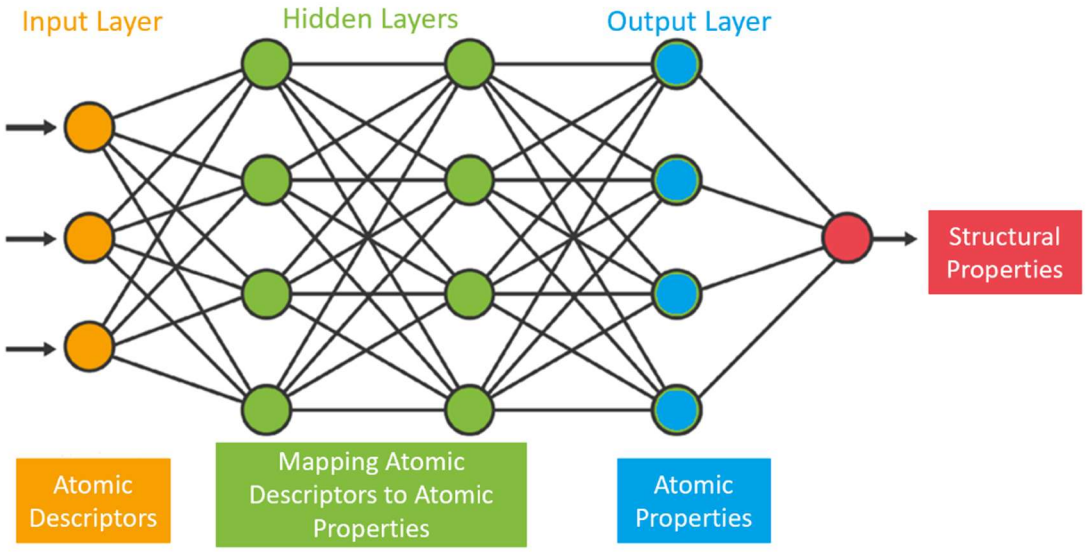


Figure 2: Representation of the deep neural network (DNN) employed by DeePMD.

Because DeePMD is easily implemented, the platform was used to build interatomic potentials for the use in structural relaxations and fitness functions of EAs. To be successful in an EA, the interatomic potential needs to provide an accurate representation of atomic interactions. This ensures that structures throughout the algorithm are properly relaxed and ranked. The interatomic potential also needs to cover a significant portion of the PES since random structures will most likely be high in energy but could theoretically be located anywhere on it. Manual testing of DeePMD was carried out to determine whether deep potentials could be used to represent atomic interactions. Automated workflows for the DeePMD training process were also created to speed up the process for model creation in hopes of allowing the widespread application of machine learned potentials in EAs. Additionally, a novel workflow for model generation is proposed, incorporating random structure generation as a means to create the optimal interatomic potential for use in EAs.

Methods

The function for random structure generation makes use of Python and was coded under the Simulated Materials Ecosystem (Simmate) framework[13].

This research project utilized the DeePMD-kit to generate machine-learned interatomic potentials. Unless otherwise specified, default settings were used with DeePMD during model training. Starting structures were obtained from the Materials Project database[14]. During manual testing of DeePMD, training structures were created with the PyMatGen[15] Monte Carlo Rattle perturbation function.

The machine learned potentials were interfaced with the Atomic Simulation Environment (ASE)[16]. With ASE, the potentials were used to instantaneously predict structural energy and forces, run molecular dynamics, and run geometry optimizations. LAMMPS[17] was also briefly used with machine-learned potentials to run molecular dynamics.

Automated workflows were coded under the Simmate framework. Simmate makes use of the Vienna Ab initio Simulation Package (VASP)[18] to carry out DFT calculations, and all code is written in Python. When testing workflows, initial structures were obtained from the Materials Project database.

Results and Discussion

Random Structure Creation

Along with structural relaxations, a crucial part of the evolutionary algorithm is the creation of new structures, especially for the initial population. While we can obtain this data from databases of existing chemical information, the advantage of EAs is that they use random

structure generation to introduce high energy structures to the search. To explore random structure generation, we developed a random unit cell generator for the intended use in an EA. While intended for an EA, this method of structure generation offers an interesting and potentially beneficial source of training data for DeePMD, especially when the intention is to use the DeePMD potential within an EA.

This random unit cell creator was named Random Symmetry Walk since structure sites are created by systematically 'walking' through Wyckoff sites until reaching the desired number of atomic sites. Beginning with a user-defined composition, the function iterates through each element and randomly selects a Wyckoff site from a randomly chosen space group. If the Wyckoff site can be used, a random coordinate for the site is created and checked to ensure that it is an acceptable distance from other sites in the unit cell. In this function, the distance between the elements at two sites must be greater than the combination of their atomic radii. If this

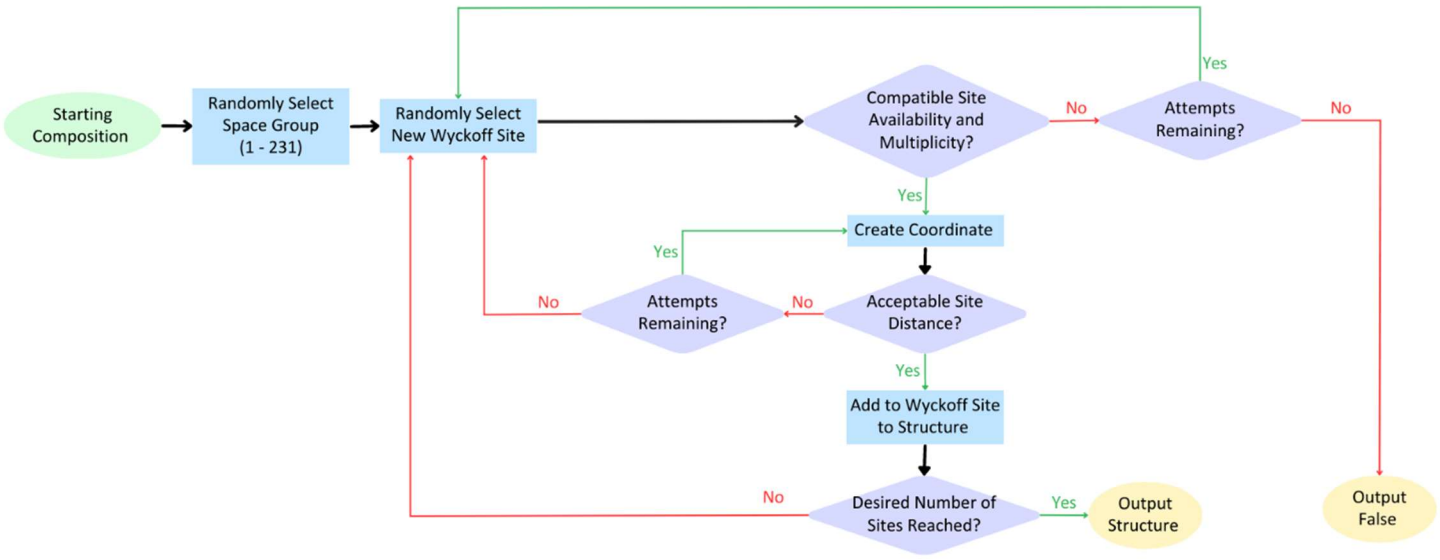


Figure 3: Flowchart representation of the Random Symmetry Walk Function.

distance check fails, then another coordinate is generated and checked. If too many attempts are made to generate a coordinate, a different Wyckoff site will be chosen instead. If the function

exceeds the set maximum number of attempts, then the entire structure fails for that space group, and the function returns False. The steps taken by the Random Symmetry Walk function are outlined in Figure 3. Examples of $MgSiO_3$ structures produced with the function can be seen in Figure 4.

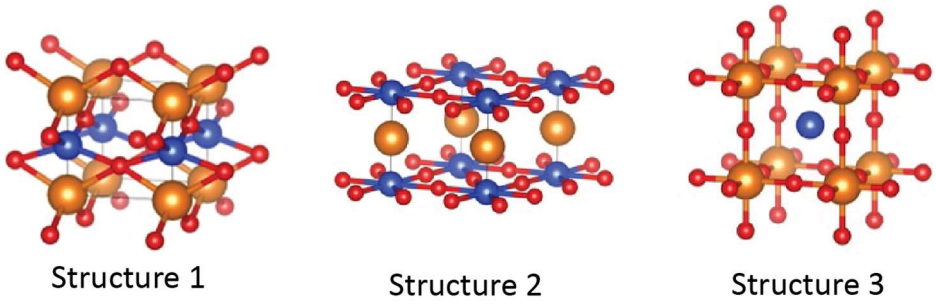


Figure 4: $MgSiO_3$ structures created with the Random Symmetry Walk function.

Random Symmetry Walk uses a series of randomized assignments and checks to generate a unit cell, leading to the creation of widely diverse structures for a single composition. In addition to the VESTA images seen in Figure 4, the diversity of these structures is shown by taking their fingerprint as seen in Figure 5. CrystalNN fingerprints from the MatMiner[19] package analyze how consistent a site is with a certain coordination environment and therefore arrangement of atoms. For each of the forty coordination environments analyzed with CrystalNN, there are five statistics resulting in a total of 200 points plotted along the x-axis. For example, the outlined portion of Structure 1's fingerprint represents coordination number 4 (Figure 5). The peaks in this section correspond to these statistics which describe how well the sites in the structure match the coordination environment. Therefore, each location along the horizontal axis of the fingerprints represents a different coordination environment, and we see that the fingerprints of structures 1-3 show peaks along different parts of the axis. The diversity between these structures can be analyzed quantitatively by calculating the difference between fingerprints (Table 1).

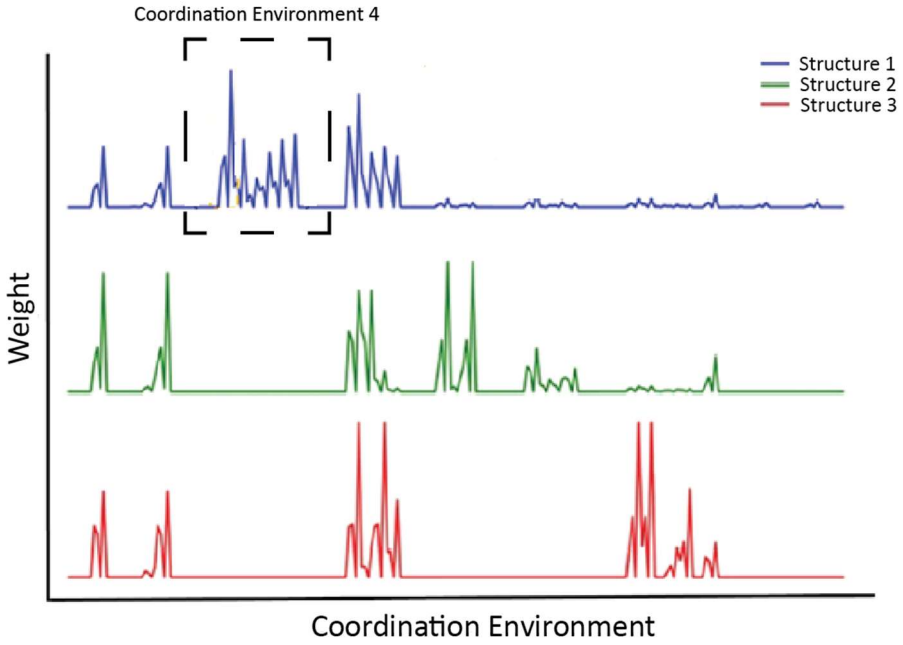


Figure 5: CrystalNN fingerprints of $MgSiO_3$ structures 1-3 created with the Random Symmetry Walk function.

Table 1: CrystalNN Fingerprint Distances Between Structures 1, 2, and 3

	Structure 1	**Structure 2**	**Structure 3**
Structure 1	0	2.08	2.31
Structure 2	2.08	0	2.51
Structure 3	2.31	2.51	0

To analyze how Random Symmetry Walk performed against widely used algorithms, the diversity of structures being produced by our function was compared to the diversity of structures produced by USPEX, a commonly used EA that utilizes random structure creation. When comparing the CrystalNN fingerprint distances of 250 C_4 structures produced by each method (Figure 6), it is seen that both cover a similar range of distances with maximum distance of approximately 3.5. This tells us that our creation function is a competitive alternative to the structure creation method used in the popular USPEX algorithm.

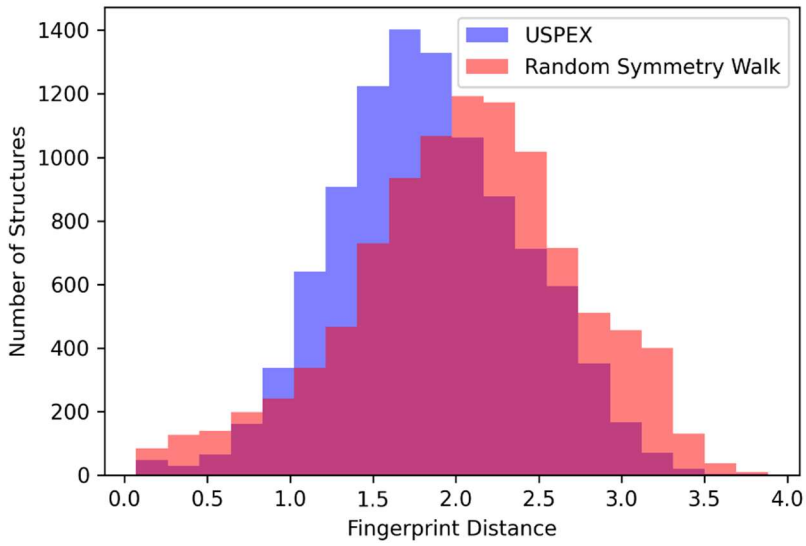


Figure 6: Comparison of fingerprint distances between Random Symmetry Walk and USPEX

The ability to produce diverse structures is essential for use in an EA, but it is also important to produce structures rapidly to create a fast algorithm. The Random Symmetry Walk function accomplishes this benchmark as well. To test the speed of the function, 10,000-unit cells of $MgSiO_3$ were generated with the total attempt counter set to 1,000 (Figure 7). Of these trials, 97.8% were successfully able to create a unit cell. Of the 10,000 trials, 99.8% of the unit cells took less than 1 second to generate. Although these time trials were conducted on small unit

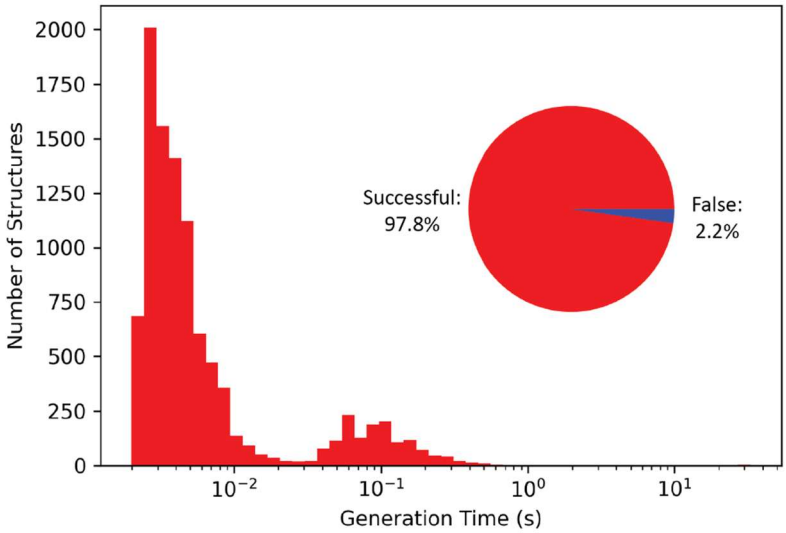


Figure 7: Results of time trials conducted on the creation of 10,000 $MgSiO_3$ structures.

cells (5 atoms), the time scales seen with this function indicate that it could be implemented to create a fast EA.

The creation times for the same C_4 structures represented in Figure 6 were again compared to creation times for C_4 structures produced with USPEX. Out of these 250 structures, the average time to produce a structure for C_4 with USPEX was 0.301 seconds while the average time for Random Symmetry Walk was 1.8 seconds. The higher average for Random Symmetry Walk can be attributed trials in which an incompatible space group was selected and the function continued to cycle through the total attempts given. Even though it had a higher average creation time, 88% of the structures created with Random Symmetry Walk were generated in less time than the USPEX average, hinting that our function could have improved performance over USPEX (Figure 8).

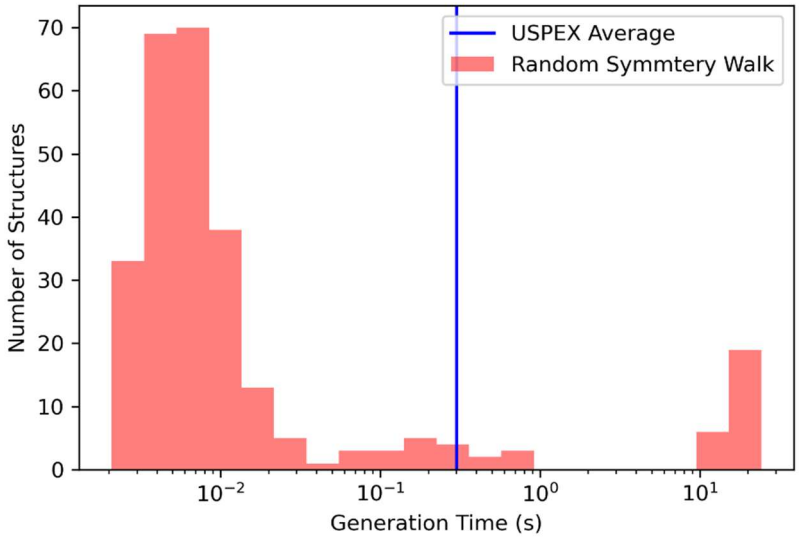


Figure 8: Comparison of C_4 generation times with Random Symmetry Walk vs. average generation time with USPEX.

The final attribute of Random Symmetry Walk that was compared to USPEX was the energy of structures being produced. For a subset of the 250 C_4 used before, the energies seen

with both methods were comparable, with the USPEX energy averaging to approximately -4.46 eV/atom and Random Symmetry Walk averaging to -4.48 eV/atom (Figure 9).

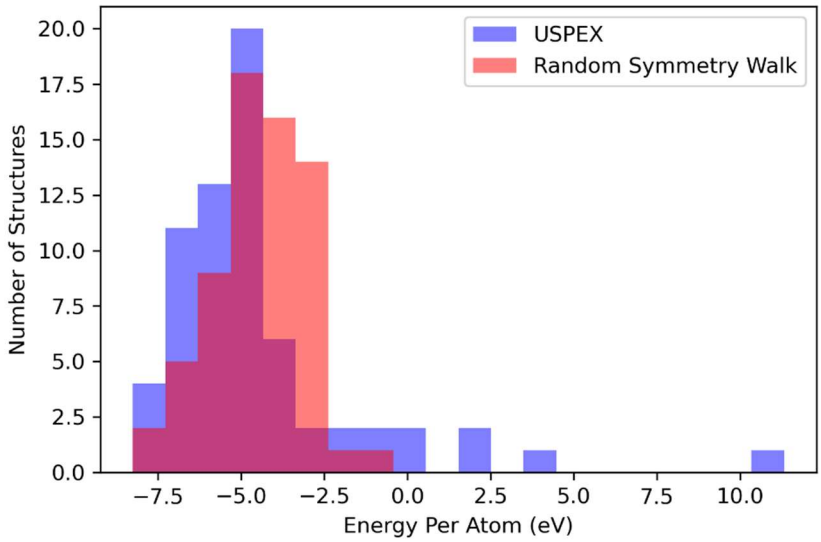


Figure 9: Comparison of the energies per atom in eV seen with Random Symmetry Walk vs. USPEX.

When comparing fingerprint distances, structure generation times, and structural energies of structures created by our Random Symmetry Walk function, we see that results are comparable with USPEX in each of these aspects. In the case of generation times, we see that our Random Symmetry Walk function may even outperform USPEX, but further testing is needed to make a concrete conclusion. These results indicate that Random Symmetry Walk is a viable method for structure creation.

Initial Testing of DeePMD

Initial testing of the DeePMD platform was carried out to test the validity of deep potentials and discover an optimal training procedure. To be applied to an evolutionary algorithm, an interatomic potential must be capable of predicting a significant portion of the PES since randomly created structures could be located anywhere on it. Additionally, the interatomic potential must be accurate so that it can properly rank structures. The process for creating a

machine learned model with DeePMD can be outlined with the following steps: creating datasets of structures whose energies and forces have been calculated with high-quality DFT methods, submitting these datasets to the DeePMD training process, and freezing the model for use in other applications. The structures used to make DeePMD datasets are crucial because they dictate which part of the PES the machine-learned potential will represent. Starting structures can be obtained from many sources, including molecular dynamics simulations and structure perturbations. Any source is valid as long as structural properties have been obtained from ab-initio methods. Throughout training, DeePMD calculates root mean squared errors (RMSEs) for forces and energies, allowing users to see how model training is progressing. RMSE values should decrease and converge during model training, with the RMSEs for energy and force at the end of training being reported as 2.8×10^{-2} eV and 2.4×10^{-2} eV/ Å[12] respectively in some applications. In DeePMD, model freezing simply means creating the output file with the interatomic potential. This file can then be used in third-party applications.

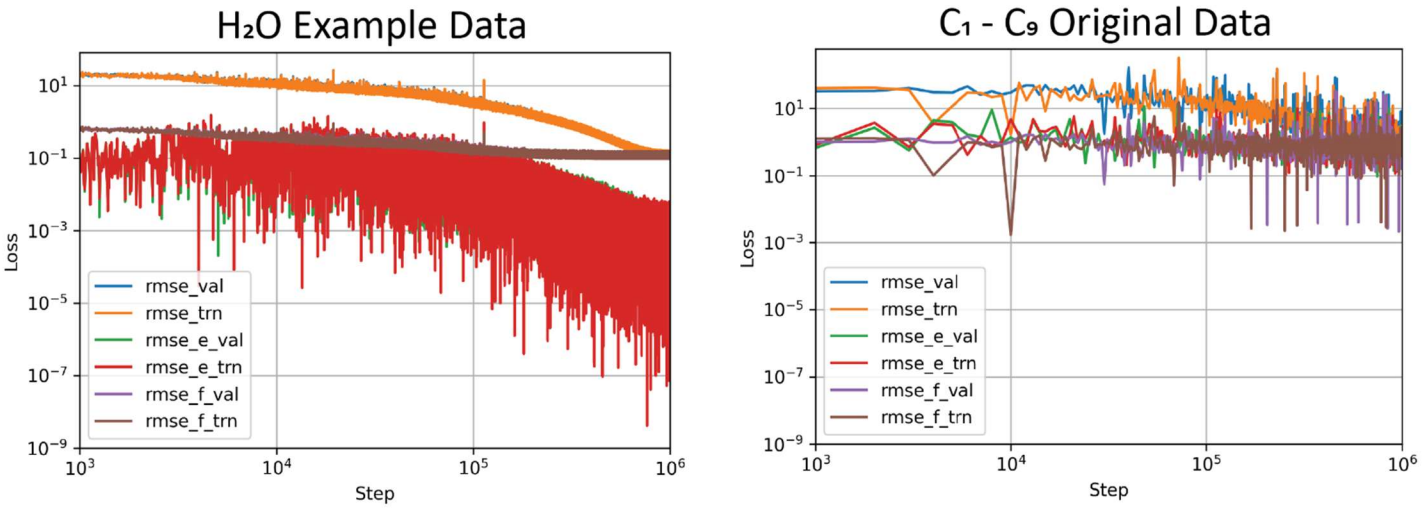


Figure 10: RMSE plots produced by DeePMD for H_2O training and carbon training. Errors are calculated by comparing training and testing data to DFT values. RMSE values for energy and force are reported in eV/atom and eV/ Å respectively.

Model creation was carried out briefly with premade H_2O datasets available through the DeePMD GitHub repository and more extensively with carbon structures. When comparing the RMSEs for water data to initial training carried out with carbon structures, the RMSE values for the water's interatomic potential showed greater convergence (Figure 10). The model trained with H_2O data had a final RMSE of 7.38×10^{-5} eV/atom for energy and 4.80×10^{-2} eV/ Å for forces, while the model created from carbon data showed an RMSE 1.05 eV/atom for energy and 0.894 eV/ Å for forces. When analyzing each dataset composition further, it was found that the water dataset was obtained from a single, low temperature MD simulation for a single composition with 40,000 total structures. In comparison our initial carbon datasets consisted of compositions ranging from C_1 to C_9, with over 100,000 structures. To replicate the dataset composition seen with H_2O data while training the model on a wider energy range of structures, we developed a new training procedure. Subsequent carbon training was carried out by iteratively training the model with separate datasets, with each dataset containing a greater spread of configurations and energies. A training procedure with a similar setup was found in the literature as well[20].

New datasets were made starting from a diamond structure of carbon from the Materials Project database (material-id: mp-66). The structure was made into a supercell of 16 carbon atoms before applying PyMatGen's Monte Carlo Rattle perturbation to create new structural configurations. DFT energies and forces for each structure were obtained with VASP MIT static energy calculations. Four datasets, each consisting of 200 structures, were made using this

procedure. Between each dataset, the rattle amplitude was systematically increased from 0.0125 to 0.1 so that each dataset encompassed a greater energy range (Figure 11).

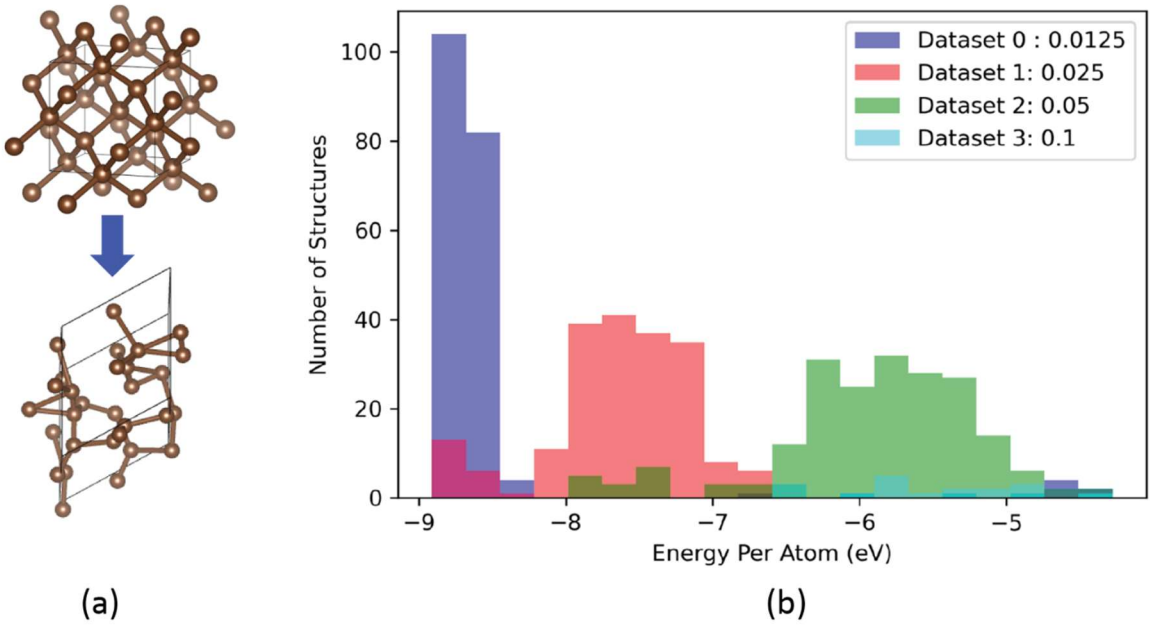


Figure 11: (a) Effect of applying Monte Carlo Rattle perturbation to diamond carbon structure with a rattle amplitude of 0.01. (b) Energy range of structures produced by PyMatGen's Monte Carlo structure perturbation. Dataset 0 corresponds to a rattle amplitude of 0.0125, Dataset 1 to 0.025, Dataset 2 to 0.05, and Dataset 3 to 0.1.

The easiest way to test the accuracy of DeePMD models is to interface them with ASE to predict the potential energies of structures and compare them to DFT calculations. As seen in Figure 12, as more datasets are introduced to the training process, the interatomic potential is better able to reproduce the values seen from DFT calculations. These results show that DeePMD can be used to accurately represent atomic systems. When looking at the RMSEs produced by the iterative training process (Figure 13), it appears that low-energy datasets show greater convergence than higher-energy datasets. This can also be seen when comparing the RMSE values for energy and force at the end of each training iteration as the values continue to increase throughout training (Table 2). Although the final RMSE of 1.24×10^{-2} eV/atom for energy and 8.21×10^{-1} eV/Å for force are higher than values seen with the H_2O datasets, these

errors are lower than those seen in the initial testing of carbon (Figure 10) and comparable to values seen in literature[12].

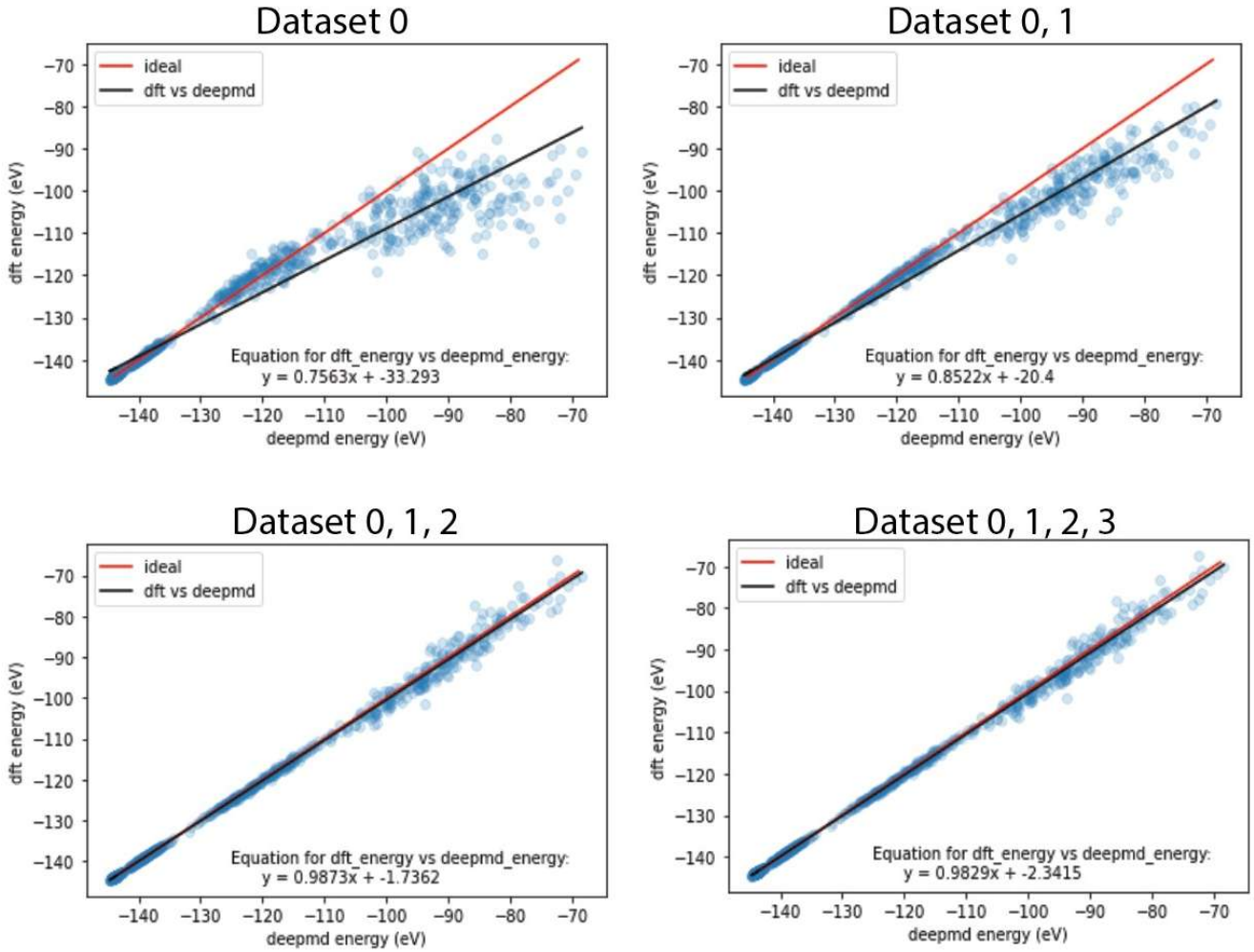


Figure 12: Progression of model accuracy with the iterative incorporation of C_{16} datasets 0-3. DFT energy values are plotted against DeePMD energy values, and an ideal fit is indicated by the relationship of these values obtaining a slope of 1.

Table 2: RMSE for Energy and Force Throughout Interative Training with C_{16} Datasets 0-3

Training Iteration	Energy RMSE (eV/atom)	Force RMSE (eV/ Å)
1	4.06×10^{-5}	5.02×10^{-3}
2	1.35×10^{-3}	4.98×10^{-2}
3	2.50×10^{-3}	7.34×10^{-2}
4	1.24×10^{-2}	8.21×10^{-1}

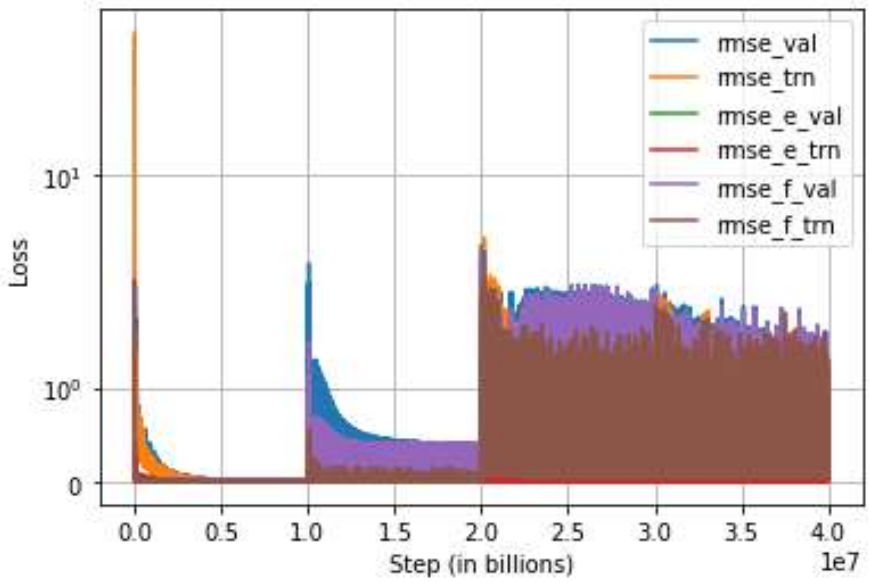


Figure 13: RMSE progression of iterative training with C_{16} datasets 0-3.

Another facet of DeePMD that was tested was different training parameters, as these could influence the accuracy of the potential created. These parameters include the size of the fitting net, number of neurons in each layer, number of nearest neighbours included, training steps and more. As it would be difficult to test all these parameters thoroughly, the fitting net was increased to test a different DNN procedure and the number of nearest neighbours was increased to test changing the descriptor for atoms. Datasets 0-2, whose energies are displayed in Figure 11, were used to train a model with increased parameters. The resulting model was used to predict the energy of structures in Datasets 0-3 and the percent error between DeePMD and DFT energies were calculated (Figure 14). When compared to default settings, using a larger fitting net and more neighboring atoms resulted in slight model improvement, reducing the average error by 0.17%. Increasing these parameters also resulted in longer training times, with 1000 steps taking approximately 20-30 seconds for default settings compared to 60-100 seconds for increased parameters. Given the dramatic increase in training times without a comparable

decrease in error, default settings were deemed adequate for creating interatomic potentials with DeePMD. Further analysis of Figure 14 shows the greatest differences in percent error for high-energy structures that were not included in the original trianing datasets, hinting that a larger fitting net and more nearest neighbours might be beneficial when extremely high energy structures are used for training.

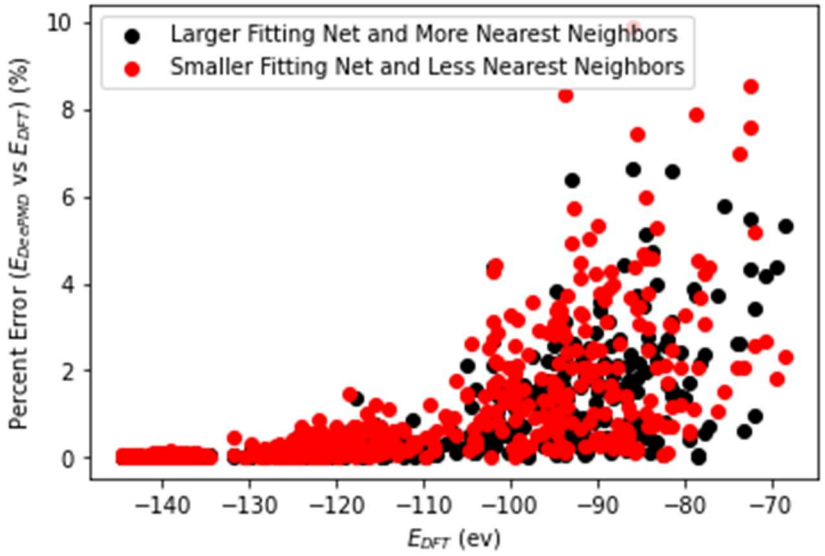


Figure 14: Percent error between DFT and DeePMD values for different fitting nets and number of nearest neighbors.

Initial testing of DeePMD informed many decisions for future model development with DeePMD. As seen in Figure 13, iterative training produced RMSE's that were lower than when training large volumes of data at once. Therefore, iterative training seems to be the most ideal way to train with a wide range of data. Even without complete RMSE convergence, the DeePMD model created for carbon showed high levels of accuracy when compared to DFT data as seen in Figure 12. Additionally, increasing the fitting net and number of nearest does not dramatically improve the results for the energy ranges used above. Although all training parameters could not be tested, it can be inferred that using default training settings for the most part will produce accurate interatomic potentials.

Creation of Automated Workflows

Our previous work showed that DeePMD could be used to create accurate models for interatomic potentials. It also showed that the process for creating such models requires significant user effort, especially in terms of collecting starting data. Therefore, ways to automate DeePMD training were explored since rapid model creation would allow for the widespread implementation of machine learning in EAs. While the DeePMD platform offers a method for automated model creation known as DP-GEN, initial testing of this package showed that it was difficult to implement. To simplify the training process, reduce the amount of user intervention, and provide an easy-to-use alternative to DP-GEN, several workflows informed by our manual testing of DeePMD were created to encapsulate the DeePMD training steps. These workflows were coded under the Simmate framework, making them available alongside commonly used calculations. Simmate also allows the workflows to easily make use of the DFT calculations needed to collect training data. The separate workflows offer unique ways of creating a model depending on the source of starting structures and method of training. When applicable, default settings for VASP and DeePMD are used but can always be modified by the user before starting a workflow. The workflows created are known as build-from-md and build-from-table.

The build-from-md method provides a way to develop models when a user has no starting data. A user only needs to provide a single starting structure for the desired composition. Training structures are then obtained from molecular dynamics simulations with the given structure under various temperatures. The switch from the structure perturbations used before to molecular dynamics was made for several reasons. First, molecular dynamics offers a way to easily access large amounts of data that includes different structural configurations. Molecular dynamics through VASP also automatically outputs the energy and force values needed for

DeePMD training, so additional calculations are not needed as they are when using structural perturbations. Using molecular dynamics as a source for starting structures is also a popular method in the literature[21, 22]. In this workflow, the number of temperatures submitted by the user dictates the number of datasets created (one for each temperature), and therefore the number of training iterations.

The initial goal when testing the build-from-md workflow was to ensure that each step was functioning properly. Therefore, parameters to complete the workflow quickly were chosen over those that would produce the most accurate model. In training a model for Si, the number of steps for each molecular dynamics run was decreased from 10,000 to 5,000, and the number of DeePMD training steps was reduced from the default of 1,000,000 to 5,000. Because of this, the models produced show poor correlation between DFT and DeePMD but still offer insight into the training process of interatomic potentials. The starting Si structure was obtained from the Materials Project database (material-id: mp-149) and made into a supercell with 64 atoms. Temperatures of 1800, 1600, 1400, and 1200 K were submitted to the build-from-md function. The resulting model was then used to predict the energy of 10,000 structures from a VASP MD simulation from 300-2000 K (Figure 15). Except for the last 500 structures of the MD simulation, there is a consistent difference of approximately 40 eV between the DFT calculated energy and DeePMD predicted energy. The dramatic reduction in training steps is thought to be responsible for the model's high rate of error, indicating that the default number of steps is necessary to create an accurate model and should not be significantly reduced. However, the consistency in the difference between the DFT and DeePMD values hints that the interatomic potential was able to fit the data to some degree.

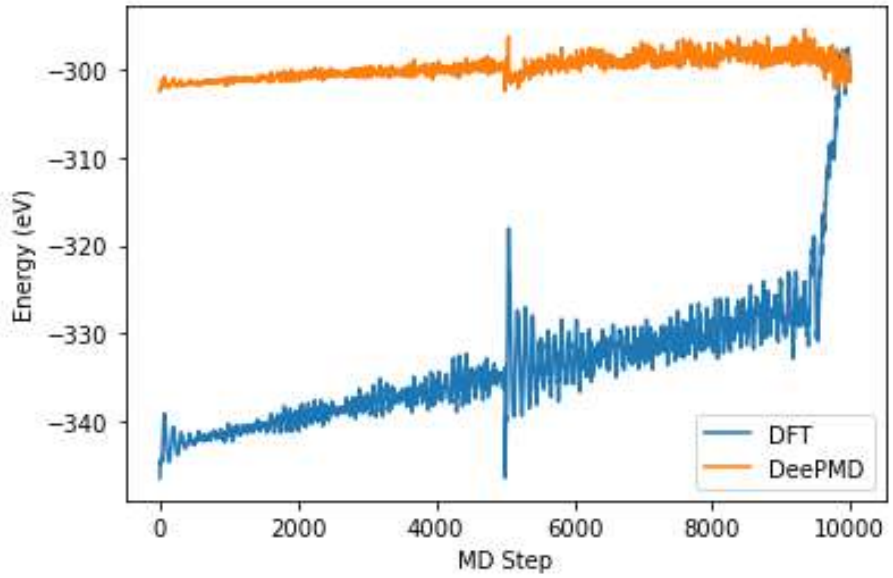


Figure 15: Comparison of energies predicted by DeePMD and energies calculated by DFT.

The build-from-table workflow was created as a way for users to make use of existing data. This workflow relies on Simmate database tables and utilizes data from previously run calculations, saving time and computational resources. In addition to molecular dynamics, DeePMD inputs can be collected from geometry optimizations and static energy calculations. The build-from-table method allows users to harness these sources simultaneously. In this workflow, the user specifies their desired composition and from which Simmate database tables they want to collect data from. Additional parameters such as the temperature of a simulation, the number of steps, and more can be provided to further refine queries. The resulting structures are then ordered by energy-per-atom and divided into equally sized datasets based on the number of user-specified training iterations. For example, if a user specifies three training iterations, then three equally sized datasets will be created. Proceeding model training and creation then follows the build-from-md method, where each dataset is introduced to the process in distinct stages.

Although this method provides less structure and standardization compared to the build-from-md workflow, it offers greater flexibility to the user. Now, training can include a greater variety of structures than those produced by molecular dynamics alone. Additionally, training would not have to be limited to structures with the same number of atoms which could lead to an interatomic potential that can be applied in more situations. All this considered, users must be mindful of if the structures being used for training make sense for the intended use of the interatomic potential and that they have enough data to train with. Among the benefits of workflow is the ability to quickly test different test parameters and training operations, which will allow us to further refine DeePMD training methods.

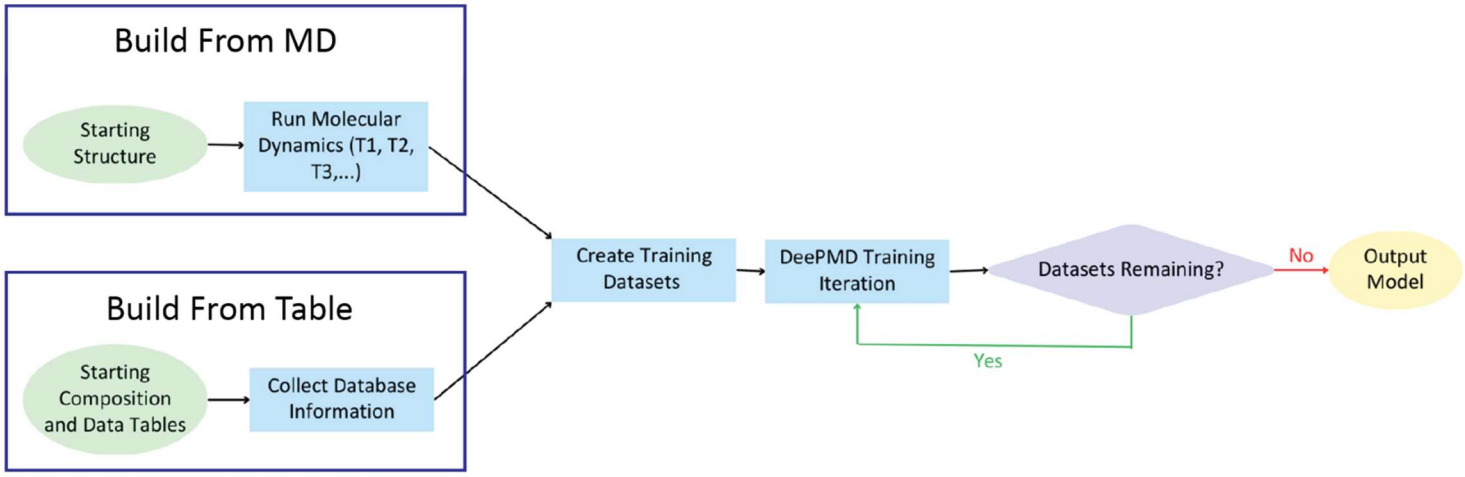


Figure 15: Flowchart representation of the build-from-md and build-from-table workflow. Each workflow differs in initial data collection but train models with the same process.

Incorporation of Random Symmetry Walk in DeePMD Training with Build-From-Parallel Workflow

The build-from-md and build-from-table workflows offer viable methods to produce interatomic potentials, but we are concerned that they do not cover the range of energies that would need to be represented in an EA. The build-from-parallel method is proposed to offer a more rigorous method to validate the accuracy of models and cover a wider range of the PES.

This workflow follows aspects of the DP-GEN process but is easier to use and takes advantage of being part of the Simmate environment. It also incorporates training with random structures instead of gaining new test data from MD simulations as DP-GEN does. In this workflow, multiple models are created simultaneously with either the build-from-md or build-from-table method. A collection of random structures is then produced using the Random Symmetry Walk function described above. The energy of each structure is then predicted by each of the models, and the structures whose energy shows a certain level of disagreement between each model are collected into a dataset and used for consecutive training iterations. This process continues until all models reach a certain level of agreement for a collection of random structures.

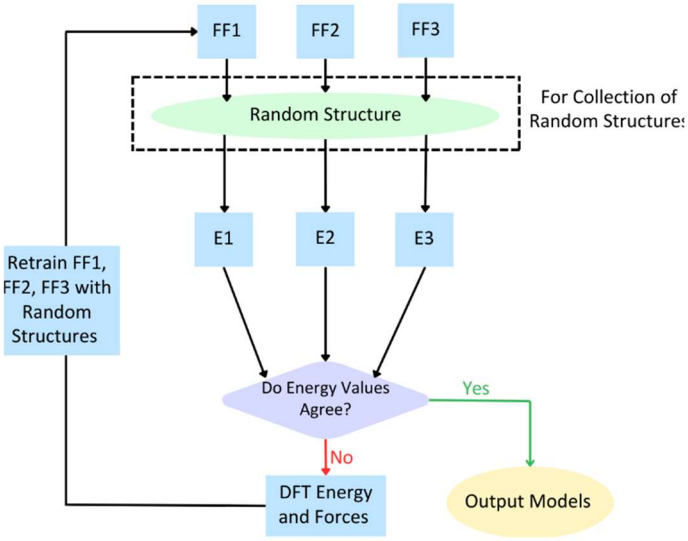


Figure 16: Flowchart representation of the build-from-parallel workflow. FF stands for force field which is synonymous with a potential model.

Obtaining a model with build-from-md or build-from-table first will create a potential that covers lower energy regions of the PES before introducing higher energy structures with randomly created ones. Training multiple models at once offers additional benefits, as it can account for the differences in model accuracy that may arise due to the random selection of specific DeePMD training parameters (i.e. the 'seed' parameters that randomly select which structure to start from). Additionally, comparing the performance of multiple models acts as a quick way to check model accuracy throughout model training while also systematically deciding which test structures to

incorporate in consecutive iterations, both of which will lead to the increased accuracy of interatomic potentials.

Conclusion and Next Steps

In hopes of increasing the speed and efficiency of EAs and therefore materials discovery, we tested random structure creation and machine learned potentials made with DeePMD. The random unit cell generator Random Symmetry Walk was shown to rapidly produce diverse structures, which is beneficial in both EAs and in creating DeePMD models that can be used to replace the fitness function in an EA. The creation of interatomic potentials with DeePMD was also validated. A potential for C_{16} was created and showed correlating values between DeePMD predicted energies and DFT calculated energies, showing that DeePMD has the potential to create accurate models. The extensive user intervention needed to produce an interatomic potential with DeePMD motivated the creation of several automated workflows, called build-from-md, build-from-table, and build-from-parallel. These workflows offer an efficient way to create models from a variety of data collection sources. The build-from-parallel workflow in particular is proposed as a way to incorporate random structure creation in model training, theoretically creating the ideal interatomic model for accurate use in an EA.

Many questions remain about random structure generation and DeePMD training. Rapid generation times were seen with the random symmetry walk method, but further testing is needed to see how the function performs when generating larger unit cells and if steps must be taken to speed up the function. For DeePMD training, the automated workflows bring us a step closer to the rapid generation of DeePMD potentials. A more systematic study of how parameters, especially the number of training steps, affect model accuracy could also help inform the best ways to orchestrate DeePMD training within workflows. As higher energy structures are

introduced to the training process, it may also be necessary to increase the dimensions of the neural network, such as fitting net and number of neurons, as this may improve model fitting to high energy data.

To successfully incorporate DeePMD potentials into an EA, much work remains to be done. Future work will include more rigorous testing of the build-from-md and build-from-table methods to ensure that they can be used to generate accurate potentials. Additionally, a workflow for the build-from-parallel method has been coded and remains to be tested. Once an accurate potential that can predict a wide range of energies is produced, it can be used to run geometry optimizations in an EA in place of DFT calculations. The training methods described in this research would be best applied to fixed composition evolutionary searches. When considering the use of a DeePMD potential for evolutionary searches including varied compositions (i.e, Y_2C, Y_4C_2, etc.), training data might need to include a variety of compositions and structure types. The most efficient way to train a DeePMD model then might be to train on-the-fly as the evolutionary search is ongoing. Although several steps remain, the incorporation of random structure creation and DeePMD in evolutionary searches is possible and has the ability to dramatically improve the process of materials discovery.

References

1. Menon, D.; Ranganathan, R. A Generative Approach to Materials Discovery, Design, and Optimization. *ACS Omega* 2022, 7 (30), 25958–25973.
2. Maia, E. H.; Assis, L. C.; de Oliveira, T. A.; da Silva, A. M.; Taranto, A. G. Structure-Based Virtual Screening: From Classical to Artificial Intelligence. *Font. Chem.* 2020, 8.
3. Le, T. C.; Winkler, D. A. Discovery and Optimization of Materials Using Evolutionary Approaches. *Chem. Rev.* 2016, 116 (10), 6107–6132.
4. González, M. A. Force Fields and Molecular Dynamics Simulations. École thématique de la Société Française de la Neutronique 2011, 12, 169–200.
5. Wen, T.; Zhang, L.; Wang, H.; E, W.; Srolovitz, D. J. Deep Potentials for Materials Science. Materials Futures 2022, 1 (2), 022601.
6. Jennings, P. C.; Lysgaard, S.; Hummelshøj, J. S.; Vegge, T.; Bligaard, T. Genetic Algorithms for Computational Materials Discovery Accelerated by Machine Learning. *Npj Comput. Mater.* 2019, 5 (1).
7. Oliveira, O. N.; Oliveira, M. C. Materials Discovery with Machine Learning and Knowledge Discovery. *Front. Chem.* 2022, 10.
8. Menon, D.; Ranganathan, R. A Generative Approach to Materials Discovery, Design, and Optimization. *ACS Omega* 2022, 7 (30), 25958–25973.
9. Blank, T. B.; Brown, S. D.; Calhoun, A. W.; Doren, D. J. Neural Network Models of Potential Energy Surfaces. *J. Chem. Phys.* 1995, 103 (10), 4129–4137.
10. Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G. Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *PRL* 2010, 104 (13).

11. Han, J.; Zhang, L.; Car, R.; E, W. Deep Potential: A General Representation of a Many-Body Potential Energy Surface. *CiCP* 2017, 23 (3).
12. Wang, H.; Zhang, L.; Han, J.; E, W. DeePMD-Kit: A Deep Learning Package for Many-Body Potential Energy Representation and Molecular Dynamics. *Comput. Phys. Commun.* 2017, 228, 178–184.
13. Sundberg, J. D.; Benjamin, S. S.; McRae, L. M.; Warren, S. C. Simmate: A Framework for Materials Science. *JOSS* 2022, 7 (75), 4364.
14. Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; Persson, K. A. Commentary: The Materials Project: A Materials Genome Approach to Accelerating Materials Innovation. *APL Mater.* 2013, 1 (1), 011002.
15. Ong, S. P.; Richards, W. D.; Jain, A.; Hautier, G.; Kocher, M.; Cholia, S.; Gunter, D.; Chevrier, V. L.; Persson, K. A.; Ceder, G. Python Materials Genomics (Pymatgen): A Robust, Open-Source Python Library for Materials Analysis. *Comput. Mater. Sci.* 2012, 68, 314–319.
16. Hjorth Larsen, A.; Jørgen Mortensen, J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dułak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C.; Hermes, E. D.; Jennings, P. C.; Bjerre Jensen, P.; Kermode, J.; Kitchin, J. R.; Leonhard Kolsbjerg, E.; Kubal, J.; Kaasbjerg, K.; Lysgaard, S.; Bergmann Maronsson, J.; Maxson, T.; Olsen, T.; Pastewka, L.; Peterson, A.; Rostgaard, C.; Schiøtz, J.; Schütt, O.; Strange, M.; Thygesen, K. S.; Vegge, T.; Vilhelmsen, L.; Walter, M.; Zeng, Z.; Jacobsen, K. W. The Atomic Simulation Environment—a Python Library for Working with Atoms. *J. Phys. Condens. Matter* 7 2017, 29 (27), 273002.

17. Thompson, A. P.; Aktulga, H. M.; Berger, R.; Bolintineanu, D. S.; Brown, W. M.; Crozier, P. S.; in 't Veld, P. J.; Kohlmeyer, A.; Moore, S. G.; Nguyen, T. D.; Shan, R.; Stevens, M. J.; Tranchida, J.; Trott, C.; Plimpton, S. J. LAMMPS - a Flexible Simulation Tool for Particle-Based Materials Modeling at the Atomic, Meso, and Continuum Scales. *Comput. Phys. Commun.* 2021, 271, 108171.
18. Kresse, G.; Furthmüller, J. Efficient Iterative Schemes for Ab InitioTotal-Energy Calculations Using a Plane-Wave Basis Set. *PRB* 1996, 54 (16), 11169–11186.
19. Ward, L.; Dunn, A.; Faghaninia, A.; Zimmermann, N. E. R.; Bajaj, S.; Wang, Q.; Montoya, J.; Chen, J.; Bystrom, K.; Dylla, M.; Chard, K.; Asta, M.; Persson, K. A.; Snyder, G. J.; Foster, I.; Jain, A. Matminer: An Open Source Toolkit for Materials Data Mining. *Comput. Mater. Sci.* 2018, 152, 60–69.
20. Botu, V.; Batra, R.; Chapman, J.; Ramprasad, R. Machine Learning Force Fields: Construction, Validation, and Outlook. The Journal of Physical Chemistry C 2016, 121 (1), 511–522.
21. Sours, T. G.; Kulkarni, A. R. Predicting Structural Properties of Pure Silica Zeolites Using Deep Neural Network Potentials. *J. Phys. Chem. C* 2023, 127 (3), 1455–1463.
22. Wen, T.; Wang, R.; Zhu, L.; Zhang, L.; Wang, H.; Srolovitz, D. J.; Wu, Z. Specialising Neural Network Potentials for Accurate Properties and Application to the Mechanical Response of Titanium. *Npj Comput. Mater*. 2021, 7 (1).