EVALUATING MACHINE LEARNED POTENTIALS IN DISORDERED ROCKSALTS

THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE (RESEARCH)

BY

NIDHISH SAGAR

UNDERGRADUATE PROGRAMME

INDIAN INSTITUTE OF SCIENCE

THESIS SUPERVISOR

SAI GAUTAM GOPALAKRISHNAN

ASSISTANT PROFESSOR

DEPARTMENT OF MATERIALS ENGINEERING

INDIAN INSTITUTE OF SCIENCE

ABSTRACT

Most commercial high-energy density cathodes for rechargeable lithium batteries till date use ordered, layered materials in which lithium and redox-active transition-metal cations occupy distinct sites. In recent years, it has been shown that cation-disordered materials, which can exhibit a large diversity in their compositions, allow facile lithium movement and can yield high capacities, resulting in storage systems with high energy densities. However, disordered materials are hard to accurately model using conventional methods like density functional theory (DFT) because of their size and complexity: experimentally synthesizing disordered structures is often easier than theoretically simulating them. Hence, the aim of this work is to study the use of different machine learning interatomic potentials (ML-IAPs) on a dataset of multi transition-metal disordered rocksalts, which can yield robust predictions on novel compositions as battery electrodes. By learning the DFT-calculated local environments available in "smaller" structures, ML-IAPs can accurately extrapolate properties to larger structures (and configurational spaces) and save orders of magnitude of computational cost. The predictions will also guide experimentalists to prepare new compositions in the lab, that can be practical battery electrode materials with high energy densities. By comparing the performance of different ML-IAPs, we will be able to recommend good ML-IAPs for this particular family of compounds for any future work.

CONTENTS

FRONT COVER 1
ABSTRACT 2
CONTENTS 3
LIST OF FIGURES 4
LIST OF TABLES 6
ACKNOWLEDGEMENTS 7
CHAPTER 1: INTRODUCTION 9
CHAPTER 2: LITERATURE SURVEY 11
CHAPTER 3: METHODS 21
CHAPTER 4: RESULTS 30
CHAPTER 5: DISCUSSION 38
CHAPTER 6: WORK PLANNED 39
CHAPTER 7: CHALLENGES 40
CHAPTER 8: CONCLUSIONS & FUTURE WORK 41
REFERENCES 42
APPENDICES 46

LIST OF FIGURES

FIGURE 1: SPECIFIC POWER AND ENERGY DENSITIES FOR DIFFERENT TYPES OF RECHARGEABLE BATTERIES PRESENT AND BEING INVESTIGATED[2] ... 9

FIGURE 2:SCHEMATIC OF THE MACHINE LEARNING POTENTIAL APPROACH. IN CONTRAST TO CONVENTIONAL ATOMIC INTERACTION POTENTIALS THAT ARE APPROXIMATE DESCRIPTIONS OF PHYSICAL LAWS, MACHINE LEARNING POTENTIALS ARE BLACK BOXES WHOSE FUNCTIONAL FORMS HAVE NO DIRECT PHYSICAL INTERPRETATION [9] 11

FIGURE 3 (A) GRAPH REPRESENTATION OF A MULTILAYER PERCEPTRON ARTIFICIAL NEURAL NETWORK. THE NODES IN THE INPUT LAYER (BLUE) REPRESENT THE ARGUMENT VECTOR (X=X_0) OF THE NETWORK FUNCTION, AND THE NODES IN THE OUTPUT LAYER (GREEN) CORRESPOND TO THE FUNCTION VALUE (Y=X_3.) BIAS NEURONS ARE LABELLED WITH 1 AND HAVE A DOTTED RED BORDER. THE NODES OF THE FIRST NETWORK LAYER X_1 AND THE EDGES CORRESPONDING TO THE THIRD TRANSFER MATRIX W_3 ARE HIGHLIGHTED IN LIGHT GREY SHADE BOX. (B) A SINGLE ARTIFICIAL NEURON. EACH NODE xi,j OF THE ARTIFICIAL NEURAL NETWORK CORRESPONDS TO A SINGLE ARTIFICIAL NEURON THAT SUMS UP INPUT SIGNALS AND TRANSMITS THE OUTPUT OF THE ACTIVATION FUNCTION fai,j. BIAS NEURONS TRANSMIT CONSTANT SIGNALS [9]. 13

FIGURE 4 SHOWS A) AN ORDERED $LICOO_2$ STRUCTURE; B) DISORDERED $LICOO_2$ WITH LI AND CO MIXED (NO DISTINGUISHABLE LAYERS); C) MULTI-SPECIES DISORDERED $LITMO_2$, WHERE TM CONSISTS OF A COMBINATION OF DIFFERENT 3D ELEMENTS, EACH OF WHICH IS REPRESENTED BY A UNIQUE COLOR (EXCEPT GREEN). D) MULTI-SPECIES DISORDERED TMO_2 WITHOUT LI. 22

FIGURE 5 SCHEMATIC SHOWS A TYPICAL ML-IAP TRAINING PROCESS[43] 25

FIGURE 6: FLOWCHART OF A TYPICAL ANN CONSTRUCTION AND PREDICTION USING THE AENET PACKAGE[9]. 26

FIGURE 7: PLOTS FOR DFT V/S ML-IAP ENERGY. THE PARITY LINE IS SHOWN IN BLACK; THE BLUE DOTS REPRESENT THE TEST DATA POINTS. THE DISTANCE OF THE BLUE DOTS FROM THE PARITY LINE IS USED TO CALCULATE MEAN ABSOLUTE ERRORS (MAE). 31

FIGURE 8: PLOTS FOR DFT V/S ML-IAP ENERGY. THE PARITY LINE IS SHOWN IN BLACK; THE BLUE DOTS REPRESENT THE TEST DATA POINTS. THE

DISTANCE OF THE BLUE DOTS FROM THE PARITY LINE IS USED TO CALCULATE MEAN ABSOLUTE ERRORS (MAE). 32

FIGURE 9: PLOTS FOR DFT V/S ML-IAP FORCES. THE PARITY LINE IS SHOWN IN BLACK; THE BLUE DOTS REPRESENT THE TEST DATA POINTS. THE DISTANCE OF THE BLUE DOTS FROM THE PARITY LINE IS USED TO CALCULATE MEAN ABSOLUTE ERRORS (MAE). 33

FIGURE 10: PLOTS FOR DFT V/S ML-IAP ENERGY. THE PARITY LINE IS SHOWN IN BLACK; THE BLUE DOTS REPRESENT THE TEST DATA POINTS. THE DISTANCE OF THE BLUE DOTS FROM THE PARITY LINE IS USED TO CALCULATE MEAN ABSOLUTE ERRORS (MAE). 34

FIGURE 11: PLOTS FOR DFT V/S ML-IAP ENERGY. THE PARITY LINE IS SHOWN IN BLACK; THE BLUE DOTS REPRESENT THE TEST DATA POINTS. THE DISTANCE OF THE BLUE DOTS FROM THE PARITY LINE IS USED TO CALCULATE MEAN ABSOLUTE ERRORS (MAE). 35

FIGURE 12: PLOT OF AENET PREDICTED V/S DFT ENERGY. THE PARITY LINE IS SHOWN IN BLACK; THE BLUE DOTS REPRESENT THE TEST DATA POINTS. THE DISTANCE OF THE BLUE DOTS FROM THE PARITY LINE IS USED TO CALCULATE MEAN ABSOLUTE ERRORS (MAE). 36

FIGURE 13: VALUES OF MAE FOR “N” TM STRUCTURES (N VARYING FROM 1 TO 8) 37

LIST OF TABLES

TABLE 1: SUMMARY OF THE DIFFERENT ML-IAP FRAMEWORKS AVAILABLE. 20

TABLE 2: HUBBARD U CORRECTION VALUE TABLE 24

TABLE 3: ATOMIC ENERGY TABLE...24

TABLE 4: COLUMN 1 IS THE ML-IAP AND COLUMN 2 SHOWS THE CORRESPONDING VALUE OF RANGE NORMALIZED ERROR IN MEV/ATOM 31

TABLE 5: COLUMN 1 IS THE ML-IAP AND COLUMN 2 SHOWS THE CORRESPONDING VALUE OF RANGE NORMALIZED ERROR IN MEV/ATOM 32

TABLE 6: COLUMN 1 IS THE ML-IAP AND COLUMN 2 SHOWS THE CORRESPONDING VALUE OF RANGE NORMALIZED ERROR IN MEV/ATOM 34

TABLE 7: COLUMN 1 IS THE ML-IAP AND COLUMN 2 SHOWS THE CORRESPONDING VALUE OF RANGE NORMALIZED ERROR IN MEV/ATOM 35

TABLE 8: TIME TAKEN TO TRAIN ML-IAP METHODS OVER DIFFERENT DATASET COMPLEXITIES 38

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Prof. Sai Gautam Gopalakrishnan, for his continuous support and guidance. He is my role model and I have always considered myself lucky to be a student (in fact, the 1st student) in his lab. There are tons of things (both academic and non-academic) that I get to learn from him on a daily basis which has cumulatively helped me become a better person.

Second, I must thank IISc for giving me such an amazing platform to grow from a school kid in 2017 to a fairly mature 21 year old today. Hostel life has been vital for that change. My smart undergrad friends have done me a world of good by making me realise and accept that there is a person better than me in everything. Thanks Aman for being the nicest friend ever, will miss you loads as you move to Europe in the coming months. Yash, thanks for being the good samaritan who lends notes and gives great food parties. Special shoutout to Nabarun and Niket for all the C-mess entertaining conversations. You guys will definitely land girlfriends wherever you go for PhD, don't worry. Girish, for being the kindest hostel neighbour I have had till date. Thanks Saravana, for sharing the first bench with me throughout the IISc classes, as you move to New York in the coming months, looking forward to seeing great photos of the skyline.

Next, I would like to thank all my department professors who have taught me: beginning with Prof. Abinandanan (Abi) for all the interesting conversations we have had and also for introducing me to Prof. Gautam; Prof. Karthikeyan for his brilliant course lectures and timely advice which has helped me a lot (in hindsight); Prof. Ravishankar for all the lectures and life lessons at times; Prof. Abhishek Singh, for the Computational Modelling course, which fueled my interest in Computational Materials and gave me a headstart into my thesis project.

Thanks to Prof. Vikram Jayaram(VJ), whom I like to call "The Encyclopedia of Materials Science", for the most practical and useful course on Functional Materials. He made the class see the world like a materials scientist. A big shoutout to Dr. Dheepa Srinivasan, a senior scientist and a materials industry veteran, who took me as an intern in 1st year and motivated me to major in Materials Science. She is always there to help young students like me, give advice and shape our careers.

Thanks to Prof. Balaram Sahoo, for a well designed UG course which helped me, because I paid attention in class; Prof. Kaushik Chatterjee for that amazing first course on Materials Science and supporting me for keeping the Amalgam club alive; Prof. Bose (Surjo) for being a caring hostel warden and for teaching me an advanced course on polymers in the most intuitive way possible, even though I hadn't done the basic course (because PCR didn't offer it that year) ; Prof. Satyam Suwas for being the nicest professor in UG Materials who shows exceptional compassion and care; Prof. Praveen Kumar (PK), Prof. Chandan Srivastava and Prof. Aloke Paul for their effort in teaching the UG core courses.

A big thank you to Prof. CNR Rao, who took me as a summer intern in my 2nd year at JNCASR. He is a scientific celebrity, yet he took out time to discuss with me and think about my well being. He wanted paper-worthy work from me in 3 months and I am glad, I was able to deliver.

Looking back, coming to IISc and choosing a Materials major was definitely one of the best decisions I have made.

Finally, I would like to thank my family: my dad for being most considerate and progressive, my mom for always being concerned about my well-being, my elder sister for all the care and support she has shown, my grandpa who calls me everyday to hear my voice, my grandma who has been the backbone of my childhood, when my mom used to be away on work. I am here today only because of all of your selfless contributions.

Being well versed in Sanskrit, I would like to end with an old adage

गते शोको न कर्तव्यो भविष्यं नैव चिन्तयेत् ।
वर्तमानेन कालेन वर्तयन्ति विचक्षणाः ॥

The past is not worth mourning, don't worry for the future

The wise advice us to live by the present

CHAPTER 1: INTRODUCTION

Lithium-ion batteries are becoming ever more important to the society. They are being used in portable electronics, grid storage for renewables and electric vehicles. The demand for high performance Li-ion batteries has been increasing and various compositions and structures have been and are being investigated [1]. **Figure 1** summarizes the various battery frameworks that have working prototypes (and/or have commercial applications), with their range of specific power (which affects how quickly energy can be stored or delivered) and specific energy (how much energy can actually be stored or delivered).

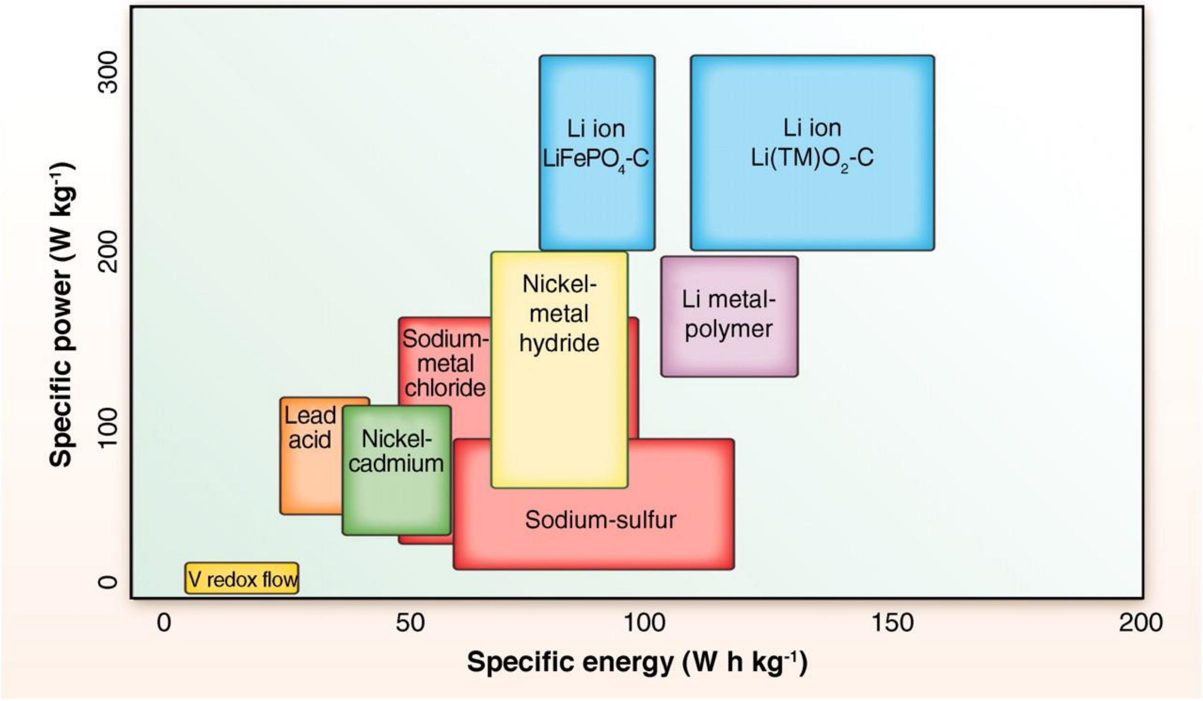

Figure 1: Specific Power and Energy densities for different types of rechargeable batteries present and being investigated[2]

The current structure of Li-ion batteries consists of so-called intercalation electrodes that allow the flow of lithium ions into and out of the a host framework, which is typically an ordered and layered structure, i.e., lithium and other cations occupy distinct sites on distinct layers of the structure [1,3,4]. In the past, cation-disordered structures were generally disregarded as potential positive electrode (or cathode) materials and received less attention [5] because lithium diffusion was believed to be limited by cation-disordering. Recently, Lee et al. [6] showed transport pathways which can allow Li diffusion in Li-excess disordered rocksalt (DRX) materials, through the formation of a percolation network connected by fast Li migration channels, which demonstrated their potential to be candidate battery cathodes that have deliver high energy densities. Note that the rocksalt structure is the combination of two face-centered cubic (FCC) sublattices, which are in the case of lithium transition metal (TM) oxides, occupied by oxygen and the cations (Li, TM), respectively. Since the oxygen sublattice is independent of the cation ordering, the configurational space of rocksalt-type oxides can be reduced to the cation (FCC) sublattice.

While layered $Li_xTM_yO_2$ compounds have distinct cation sublattice sites being occupied by Li and TM, i.e., a unique ordering of Li and TM atoms across the cation sublattice, a DRX has no such ordering, especially at long length scales. This cation disorder is found to enhance structural stability upon Li extraction, which makes it possible to achieve high reversible capacities and reduce the overall volume change with varying lithium content [7]. Minimizing volume fluctuations is beneficial for all electrodes and is especially important for solid-state batteries in order to prevent fracturing of the solid-solid, electrode||electrolyte interfaces. DRX structure, by definition, is isotropic, allowing for Li diffusion in all 3 directions unlike 2 in layered structures.

With these advantages in mind, we proceed to find new DRX compositions that can be worthy candidates for battery electrodes. But modelling these structures using computational methods, such as density functional theory (DFT) calculations, is expensive, given that a disordered structure typically needs a large supercell containing hundreds of atoms with several thousands-to-millions of possible configurations. Hence, we decide to use Machine Learning (ML) to learn local atomic environments in terms of an interatomic potential (IAP), based on smaller-scale DFT calculations, and use the generated potentials to model larger systems by extrapolation. Thus, the ML-IAPs are a mathematical approximation of the potential energy surface [8] that governs the DRX structure, and in turn its electrochemical properties, given a composition of Li and TMs.

CHAPTER 2: LITERATURE SURVEY

ML-IAP Formalism

Potential fitting, i.e., the (non-)linear regression of the potential energy surface of atomic systems, is an example of a supervised learning problem, as the model learns the relationship between atomic structure and energy (and/or forces) from a database of reference structures and energies (and/or forces). IAPs are usually atom-centered, i.e., the energy of a structure is given as a weighted summation of the energies contributed by individual atoms present in the structure. The energies of individual atoms, in turn, are governed by their local environments. Once the model successfully learns the structure-energy relationship, it can be used as a "black box" to predict energies of previously unknown structures, provided that the new structures are sufficiently similar to the ones in the trained reference set or have similar local chemical environments, as illustrated in **Figure 2**. While classical potential fitting has relied heavily on well-established mathematical functional forms and human ingenuinity, ML-IAPs typically retain a large degree of flexibility in their potential's functional forms and can rapidly learn on available data. Also, since machine learning models typically consist of differentiable mathematical functions, expressions for atomic forces can be obtained as negative gradient of the energy. Thus, ML-IAPs are powerful theoretical frameworks for coarse-graining a diverse configurational space from well-known local structural features.

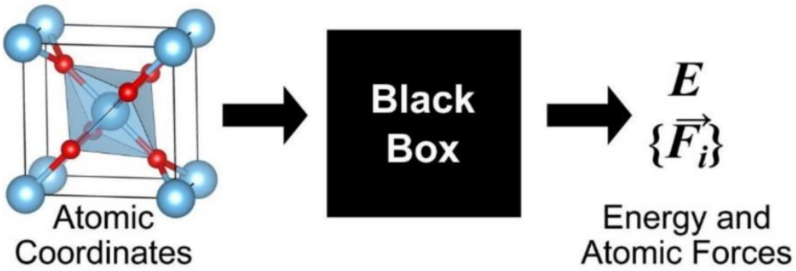

Figure 2:Schematic of the machine learning potential approach. In contrast to conventional atomic interaction potentials that are approximate descriptions of physical laws, machine learning potentials are black boxes whose functional forms have no direct physical interpretation [9]

ML-IAP models

Table 1 provides a summary of the different ML-IAP models considered in this work. In this section, we explain each of the models considered, and their mathematical basis, sequentially.

Artificial Neural Networks (ANN)

In ANNs, unknown/arbitrary mathematical functions are represented in the form of graph networks that are similar to the network of neurons in our brains, resulting in the terminology of an artificial neural network. In biology, each neuron in a network receives an electrical input and transmits a response (or output) whenever the total input exceeds a certain activation threshold. Each neuron can thus be pictured as a signal processor with a Heaviside step activation function. In the case of ANNs, an artificial neuron is placed within a graph network and transmits an output depending on the received input and a mathematical activation function, which is typically one of the functions illustrated in **Eqs. I-IV**. For numerical reasons, the discontinuity of a step function is avoided in ANNs [9].

Linear Function $$f_a^1(x) = x \qquad \text{I}$$

Hyperbolic tangent $$f_a^2(x) = tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \qquad \text{II}$$

Logistic Function $$f_a^3(x) = \frac{1}{1+e^{-x}} \qquad \text{III}$$

tanh with linear twisting $$f_a^4(x) = 1.7159\, tanh\left(\frac{2}{3}x\right) + ax \qquad \text{IV}$$

ANNs that connect input nodes with output nodes in a linear fashion, without cyclic connections between neurons, are called feed-forward neural networks [10], an example of which is displayed in **Figure 3a**. In multilayer perceptrons (MLPs), the class of ANNs used in this project, the artificial neurons are additionally organized in layers, and only neurons (black outlined circles in Figure 3a) in adjacent layers are connected. The graph representation of a single neuron is shown in Figure 3b. The algebraic expression of the **jth** artificial neuron in the **ith** layer of an MLP with N layers is:

$$x_{i,j}(\{x_{i-1,k}\}) = f_a^{i,j}\left(\sum_k w_{kj}^i x_{i-1,k}\right) \text{ with } i = 1, \ldots, (N-1) \qquad \text{V}$$

where $\{x_{i-1,k}\}$ are the input signals from the previous layer $(i-1)$, w_{kj}^{i} is the weight of the signal from the k^{th} neuron in layer $(i-1)$ and $f_a^{i,j}$ is the activation function. Equivalently, the vector of all neurons in layer i is given by,

$$\boldsymbol{x}_i(\boldsymbol{x}_{i-1}) = f_a^i(\boldsymbol{W}_i\boldsymbol{x}_{i-1})\ with\ i = 1, \dots, (N-1) \qquad \text{VI}$$

where $\mathbf{W}_i$ is the weight matrix for signal transfer from layer $(i-1)$ to layer i with $(\mathbf{W}_i)_{k,j} = w_{kj}^{i}$. In **Eq. VI**, the activation function acts on each component of the argument vector. The algebraic expression (network function) of the MLP in Fig.3a is

$$\mathcal{N}(\boldsymbol{x}_0; \{\boldsymbol{W}_i\}) = f_a^3\{\boldsymbol{W}_3 f_a^2[\boldsymbol{W}_2 f_a^1(\boldsymbol{W}_1\boldsymbol{x}_0)]\} = \boldsymbol{x}_3 \qquad \text{VII}$$

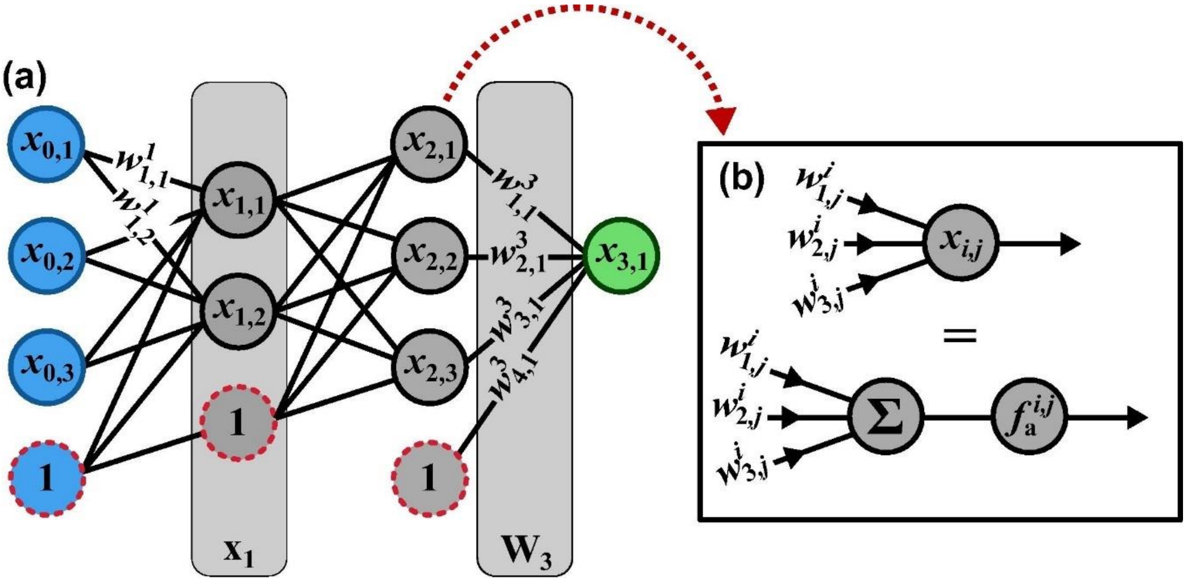

Figure 3 (a) Graph representation of a multilayer perceptron artificial neural network. The nodes in the input layer (blue) represent the argument vector ($x=x_0$) of the network function, and the nodes in the output layer (green) correspond to the function value ($y=x_3$.) Bias neurons are labelled with 1 and have a dotted red border. The nodes of the first network layer x_1 and the edges corresponding to the third transfer matrix W_3 are highlighted in light grey shade box. (b) A single artificial neuron. Each node $x_{i,j}$ of the artificial neural network corresponds to a single artificial neuron that sums up input signals and transmits the output of the activation function $f_a^{i,j}$. Bias neurons transmit constant signals [9].

It is evident from **Eq. VIII** that vector x_0 is the argument (input layer, blue nodes in Fig. 3a) of the network function N, and x3 is the function value (output layer, green nodes in Fig. 3a). The layers between input and output layers (grey nodes in Fig. 3a) are called hidden layers, as they have no intuitive interpretation. The components of the weight matrices W_l, the weight parameters, are the model parameters that must be determined during the learning process. The number of layers in an MLP and the number of neurons per layer define the network architecture. The sample MLP in Fig. 3a has a 3-2-3-1 architecture. In general, the number of output nodes is arbitrary, however, for the representation of potential energy surfaces in this project we consider a scalar output corresponding to the total energy. The input layer consists of structural inputs in the form of descriptors, which need to be invariant with respect to translation and rotation of the structure and the exchange of equivalent atoms.

We use a combined set of structural ($c_\alpha^{(2)}$) and compositional ($c_\alpha^{(3)}$) descriptors [11], which are expansion coefficients of the atom centred radial distribution function (RDF) and angular distribution function (ADF), written in a complete basis set ϕ_α, as shown in **Eq. IX**.

$$\begin{aligned} c_\alpha^{(2)} &= \textstyle\sum_{\boldsymbol{R}_j \in \sigma_i^{R_c}} \phi_\alpha(R_{ij}) f_c(R_{ij}) w_{t_j} \\ c_\alpha^{(3)} &= \textstyle\sum_{\boldsymbol{R}_j, \boldsymbol{R}_k \in \sigma_i^{R_c}} \phi_\alpha(\theta_{ijk}) f_c(R_{ij}) f_c(R_{ik}) w_{t_j} w_{t_k} \end{aligned} \tag{IX}$$

where f_c is a cutoff function that smoothly goes to zero at R_c. We use,

$$f_c(r) = 0.5[cos\,(r \cdot \pi/R_c) - 1] \tag{X}$$

The weights w_{t_j}, w_{t_k} are 1 for the structural descriptor and they take on species dependent values for the compositional descriptor. The basis functions $\{\phi_\alpha\}$ are made up of Chebyshev polynomials $\{T_\alpha\}$, such that

$$\phi_\alpha(r) = \frac{k}{2\pi\sqrt{\frac{r}{R_c} - \frac{r^2}{R_c^2}}} T_\alpha\left(\frac{2r}{R_c} - 1\right) \tag{XI}$$

where T_α is the αth order of the Chebyshev polynomial.

Another commonly-used descriptor set is the radial and angular symmetry functions, developed by Behler and Parinello [12], which are referred to as atom centred symmetry functions (ACSF). The ACSF converts the local atomic environment to numeric vectors that fulfil rotational and translational invariance. For atom i in the structure, the radial ACSF, which provides information about pair correlations between the atoms, is given as:

$$G_i^{atom,rad} = \sum_{j\neq i}^{N_{atom}} e^{-\eta\left(R_{ij}-R_s\right)^2} \cdot f_c\left(R_{ij}\right) \qquad XII$$

where η determines the width of the Gaussian basis, and R_s is the position shift over all neighboring atoms within the cutoff radius R_c. The cutoff function f_c ensures a smooth decay in value and slope at cutoff radius R_c and is given by:

$$f_c\left(R_{ij}\right) = \begin{cases} 0.5 \cdot \left[cos\left(\frac{\pi R_{ij}}{R_c}\right) + 1\right], & \text{for } R_{ij} \leq R_c \\ 0.0, & \text{for } R_{ij} > R_c \end{cases} \qquad \text{XIII}$$

The angular ACSF takes the form,

$$G_i^{atom\,,\,ang} = 2^{1-\zeta} \sum_{j,k\neq i}^{N_{atom}} \left(1+\lambda cos\ \theta_{ijk}\right)^{\zeta} \cdot e^{-\eta'\left(R_{ij}^2+R_{ik}^2+R_{jk}^2\right)} \cdot f_c\left(R_{ij}\right) \cdot f_c\left(R_{ik}\right) \cdot f_c\left(R_{jk}\right) \qquad XIV$$

where the summation loops over neighbours j and k, θ_{ijk} is the angle, and ζ, η' and λ are three parameters that determine the shape of angular symmetry functions. Symmetry functions determined by an appropriate choice of hyperparameters (η, R_s, etc.) should reflect the effective relationship between atomic representations and corresponding properties (e.g., energy and force components). Both ACSF and Chebychev-polynomial-based descriptors are implemented in the atomic energy network (AENET) package [9,11,13].

Gaussian Approximation Potential (GAP)

GAP, implemented in the quantum mechanics and interatomic potentials (QUIP) package [14], applies Gaussian Process Regression (GPR) to interpolate the atomic energy based on the spatial distribution of neighboring atoms. The energy for a given atom is given as:

$$\epsilon(\boldsymbol{R}) = \sum_k b_k \boldsymbol{K}(\boldsymbol{R}, \boldsymbol{R}_k) \qquad \text{XV}$$

where $\boldsymbol{R}$ represents the geometry of neighbouring atoms within the cut-off radius R_c, and k indexes a set of reference data points $\boldsymbol{R}_k$ that serve as the basis on which the atomic energy is expanded. $\boldsymbol{K}$ is the kernel function that captures the variation of the energy in terms of changing local configurations. $\mathrm{b_k}$ are the coefficients of the kernel fitted during the training of the potential. The kernel function used in this project is the smooth overlap of atomic positions (SOAP), which represents the rotationally integrated overlap of neighbour densities. The atomic configuration of a central atom i is represented by its neighbour density (ρ_i) as a sum of Gaussians centred over all neighbouring atoms j within a cutoff distance R_c, as in **Eq. XVI**.

$$\rho_i(\boldsymbol{R}) = \sum_j f_c(R_{ij}) \cdot exp\left(-\frac{|R-\boldsymbol{R}_{ij}|^2}{2\sigma_{atom}^2}\right) \qquad XVI$$

f_c in **Eq. XVI** is a cutoff function which ensures smooth decay in value and slope at cutoff radius R_c, and σ_{atom} atom is a smearing parameter. This neighbour density is then expanded in terms of a basis of spherical harmonics $Y_{lm}(\widehat{\boldsymbol{R}})$, and radial functions $g_n(R)$:

$$\rho_i(\boldsymbol{R}) = \sum_{nlm} c_{nlm} g_n(R) Y_{lm}(\widehat{\boldsymbol{R}}) \qquad XVII$$

In turn, the rotationally invariant spherical power spectrum of atom i is expressed in terms of expansion coefficients c_{nlm} as follows:

$$p_{n_1 n_2 l}(\boldsymbol{R}_i) = \sum_{m=-l}^{l} c_{n_1 lm}^{*} c_{n_2 lm} \qquad XVIII$$

Finally, the SOAP kernel is written as a dot product of power spectrums raised to a small integer power (ζ) as follows:

$$\boldsymbol{K}(\boldsymbol{R}, \boldsymbol{R}') = \sum_{n_1 n_2 l} \left(p_{n_1 n_2 l}(\boldsymbol{R}) p_{n_1 n_2 l}(\boldsymbol{R}')\right)^{\zeta} \qquad XIX$$

Normalization is then carried out to ensure that the kernel of each atomic environment with itself is unity. The total energy of the structure is the sum of all individual kernels each multiplied by a coefficient $\mathrm{b_k}$ as shown in **Equation XX.** ζ, R_c, σ_{atom} are some of the hyperparamters used in GAP. Setting initial values for these terms determines the accuracy of our fitted potentials.

Spectral Neighbour Analysis Potential (SNAP)

SNAP uses the bispectrum formalism to project 3D local atomic neighbour density into a set of coefficients that satisfy the required invariant properties by expansion on a spherical harmonic basis [15]. To begin, the atomic neighbour density around a central atom i at position **R** is defined as follows:

$$\rho_i(\boldsymbol{R}) = \delta(\boldsymbol{R}) + \sum_{R_{ij}<R_c} f_c(R_{ij}) \cdot \omega_j \cdot \delta(\boldsymbol{R} - \boldsymbol{R}_{ij}) \qquad XXI$$

where $\mathbf{R}_{ij}$ is the position of neighbor atom j relative to i, and ω_j is the dimensionless weight to differentiate atom types. The cutoff function f_c ensures that the neighbor atomic density decreases smoothly to zero at the cutoff radius R_c. The angular distribution of neighbour density function can be projected onto spherical harmonic functions $Y_m^l(\theta, \phi)$. In the bispectrum approach, the radial distribution is converted into an additional polar angle θ_0 defined by:

$$\theta_0 = \theta_0^{max}\frac{r}{R_c} \qquad \text{XXII}$$

Thus, the density function can be represented in 3-sphere coordinates (θ, ϕ, θ_0) instead of the typical spherical coordinate system of $(\theta, \phi, \mathrm{r})$. The density function on 3-sphere can then be expanded with 4-dimensional hyperspherical harmonics $U_{m,m'}^{j}$ as:

$$\rho_i(\boldsymbol{R}) = \sum_{j=0}^{\infty} \sum_{m,m'=-j}^{j} u_{m,m'}^{j} U_{m,m'}^{j} \qquad \text{XXIII}$$

where the coefficients $u_{m,m'}^{j}$ are obtained as the inner product of the neighbour density function with the basic function, as below:

$$u_{m,m'}^{j} = U_{m,m'}^{j}(0,0,0) + \sum_{R_{ij}<R_c} f_c(R_{ij}) \cdot \omega_j \cdot U_{m,m'}^{j}(\theta, \phi, \theta_0) \qquad \mathit{XXIV}$$

The bispectrum components $B_{j_1,j_2,j}$ can then obtained via following:

$$B_{j_1,j_2,j} = \sum_{m_1,m_1'=-j_1}^{j_1} \sum_{m_2,m_2'=-j_2}^{j2} \sum_{m,m'=-j}^{j} \left(u_{m,m'}^{j}\right)^{*} \times H \begin{matrix} j & m & m' \\ j_1 & m_1 & m_1' \\ j_2 & m_2 & m_2' \end{matrix} u_{m_1,m_1'}^{j_1} u_{m_2,m_2'}^{j_2} \qquad \text{XXV}$$

where constants $H \begin{matrix} j & m & m' \\ j_1 & m_1 & m_1' \\ j_2 & m_2 & m_2' \end{matrix}$ are the coupling coefficients that satisfy the condition, $\|j_1 - j_2\| \leq j \leq \| j_1 + j_2 \|$.

In SNAP formalism, the total energy of a given structure (E_{SNAP}), force on atom j (F_{SNAP}^{j}) and stress tensor of the structure ($\sigma_{\mathrm{SNAP}}^{j}$) are expressed as linear functions of the K-vector of bispectrum components of each i atom (B^i) as follows:

$$\begin{aligned} E_{SNAP} &= \beta_0 N + \beta \cdot \sum_{i=1}^{N} B^i \\ F_{SNAP}^{j} &= -\beta \cdot \sum_{i=1}^{N} \frac{\partial B^i}{\partial r_j} \\ \sigma_{SNAP}^{j} &= -\beta \cdot \sum_{i=1}^{N} r_j \otimes \sum_{i=1}^{N} \frac{\partial B^i}{\partial r_j}, \end{aligned} \qquad \text{XXVI}$$

where β_0 and the vector β are the coefficients derived from fitting with the database of DFT calculations. Hyperparameters involved in SNAP include cutoff radius, element weights, "rcutfac" (the scale factor applied to all cutoff radii) and "twojmax", which is the band limit for bispectrum components.

qSNAP: Quadratic Extension of SNAP

In SNAP, the linear relationship between the potentially energy surface (PES) and bispectrum components limits the complexity of energy functions to a maximum of a four-body effect, which may have an impact on its predictive power. Hence, a quadratic extension of SNAP (qSNAP) approach was proposed by Wood and Thompson [16]. The quadratic contributions to the energy can be viewed as a kind of embedding energy in analogy with the embedded atom method (EAM) [17,18]. In qSNAP, the SNAP potential is extended via the addition of the embedding energy term as follows:

$$E^i_{SNAP} = \boldsymbol{\beta} \cdot \boldsymbol{B}^i + F(\rho_i) \qquad XXVII$$

where $F(\rho_i)$ represents the energy of embedding atom i into the electron density contributed by its neighbouring atoms. The "host" electron density of embedding atom i can be expressed as a linear function of the bispectrum components as follows:

$$\rho_i = \boldsymbol{a} \cdot \boldsymbol{B}^i \qquad XXVIII$$

and the embedding energy can be expressed as a Taylor expansion based on a reference structure with density ρ_0 as follows:

$$F(\rho) = F_0 + (\rho - \rho_0)F' + \frac{1}{2}(\rho - \rho_0)^2 F'' + \cdots \qquad XXIX$$

Thus, the modified SNAP energy is given by the following expression:

$$\begin{aligned} E^i_{SNAP} &= \boldsymbol{\beta} \cdot \boldsymbol{B}^i + \frac{1}{2}F''\left(\boldsymbol{a} \cdot \boldsymbol{B}^i\right)^2 \\ &= \boldsymbol{\beta} \cdot \boldsymbol{B}^i + \frac{1}{2}\left(\boldsymbol{B}^i\right)^T \cdot \boldsymbol{\alpha} \cdot \boldsymbol{B}^i \end{aligned} \qquad XXX$$

where $\boldsymbol{\alpha} = F'' \boldsymbol{a} \otimes \boldsymbol{a}$ is a symmetric K x K matrix. Essentially the quadratic extension carries all distinct pairwise products of bispectrum components and expands the maximum complexity of energy functions to seven-body effects.

Moment Tensor Potential (MTP)

MTP constructs a contracted rotationally invariant representation of the local atomic environment using tensors [19]. A linear correlation between potential energy and atomic representation is built based on the assumption that the total energy can be partitioned into individual atomic environment contributions[20]. The potential energy of the atomic environment of a central atom i can be linearly expanded on a set of basis functions $B(\boldsymbol{R})$,

$$V_i(\boldsymbol{R}) = \sum_l \beta_l B(\boldsymbol{R}) \qquad \text{XXXI}$$

The basis functions $B(\boldsymbol{R})$, in turn, depend on a series of moment tensor descriptors over all neighbour atoms j,

$$M_{\mu,\nu}(\boldsymbol{R}) = \sum_j f_\mu\left(R_{ij}\right) \underbrace{\boldsymbol{R}_{ij} \otimes \cdots \otimes \boldsymbol{R}_{ij}}_{\nu\ times},$$

XXXII

where the functions f_μ are the radial distribution of atomic configuration and the terms $\boldsymbol{R}_{ij} \otimes \cdots \otimes \boldsymbol{R}_{ij}$ are tensors of rank ν entailing angular information about the atomic configuration. The hyperparameters in MTP training include radial cutoff value, number of radial basis functions, number of maximum iterations and weight of energies and forces.

Table 1: Summary of the different ML-IAP frameworks available.

Potential	ANN	GAP	SNAP	qSNAP	MTP
Idea	Feed Forward Neural Network represents the potential energy surface	Neighbour density is the sum of Gaussians centred over atoms	Project 3D atomic neighbour density into coefficients of a spherical harmonic basis functions	Addition of embedding energy term to SNAP	Rotationally invariant tensors to describe potential energy
Descriptor	Coefficients of distribution functions (radial and angular)	SOAP kernel	Coefficients of the bispectrum of atomic neighbour density functions	Embedding energy is expresssed by a Taylor expansion	Moment Tensors
Training Algorithm	Non linear ANN regression	Gaussian Process Regression (Bayesian Approach)	Linear Regression of bispectrum components	Linear Regression	Linear Regression
Basis Functions	Chebyshev polynomials	Spherical harmonics	Hyperspherical harmonic	Hyperspherical harmonics	Radial distribution functions * Tensors entailing angular information

CHAPTER 3: METHODS

Structure dataset

The $LiTMO_2$ DRX structure data set was generated by enumerating lithium and transition metal configurations on the cation sublattice of the rocksalt structure (see schematic in **Figure 4** and description below). Enumeration is the generation of all possible combinations (or orderings) of a given structure at a given composition with any relevant constraints on site occupancies. Specifically, we used the structures enumerated by Artrith et al. [11] with various cation arrangements up to a total of 18 cation sites using the enumeration approach of Hart et al. [21–23]. Thus, Artrith et al. generated a total of 10046 structures based on 9 TMs (i.e., TM = Sc, Ti, V, Cr, Mn, Fe, Co, Ni, and Cu). Subsequently, we generated the specific combinations of transition metals that were missing from Artrith et al.'s dataset (e.g. Li_9 (Sc Mn V Cr Fe Co Cu Ni) O_{18} and structures that had all nine TMs) using pymatgen's advanced transformations module [24], resulting in an additional 400 structures. We also generated structures without lithium ions to include lithium deficient conditions that are encountered at the top of charge (TOC). We chose a total of 500 structures randomly from each of the n-TM combinations (n = 1 to 9) and removed all Li atoms from such structures, bringing our final dataset to a total of ~11000 structures.

Figure 4a shows an ordered lithium transition-metal oxide structure ($LiCoO_2$ shown as a reference) with distinguishable green Li polyhedra layer, blue Co polyhedra layer and oxygen represented by red spheres. In Figure 4b we see a disordered rocksalt (i.e., disordered $LiCoO_2$) with lithium and multiple transition metals interspersed (no distinguishable Li/TM layers). Figure 4c represents a multi transition metal disordered cation structure (nominal composition of $LiTMO_2$) with different coloured polyhedra (except green) representing the different transition metals. When Li is stripped from Figure 4c, we obtain Figure 4d, a representation of a lithium-deficient disordered transition metal oxide (TMO_2).

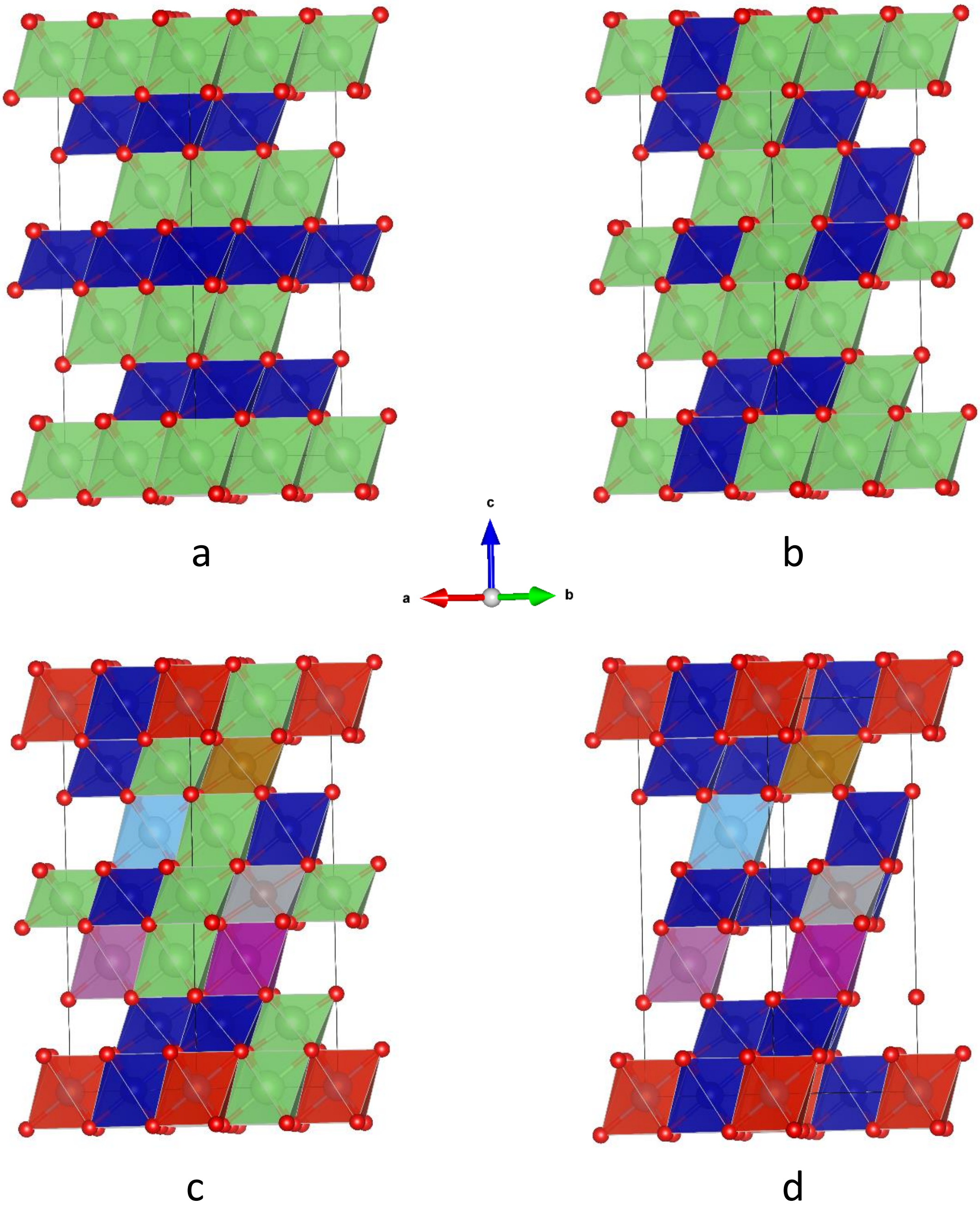

Figure 4 shows a) An ordered $LiCoO_2$ structure; b) Disordered $LiCoO_2$ with Li and Co mixed (no distinguishable layers); c) Multi-species disordered $LiTMO_2$, where TM consists of a combination of different 3d elements, each of which is represented by a unique color (except green). d) Multi-species disordered TMO_2 without Li.

Computational methods

Hubbard *U* corrected[25] DFT[26,27] calculations using the Perdew-Burke-Ernzerhof (PBE)[28] functionalization of the Generalized Gradient Approximation (GGA+*U*) were done for all the 11000 structures in the dataset. [28] projector-augmented wave potentials [29], as implemented in the Vienna Ab-initio Simulation Package (VASP) [30,31] and performed calculations till total energies and atomic forces converged to within 0.01 meV and 30 meV/Å, respectively. Gamma centered *k*-point meshes with a density of 1000 divided by the number of atoms were used, in accordance with the work of Artrith et al [11]. We set the plane wave kinetic energy cut-off to 520 eV and the VASP input files were generated using pymatgen software [32] with computational parameters compatible with the Materials Project [33]. For the reference set of atomic energies that are required by the AENET package, we calculated the energy of isolated Li, TM, and O atoms by constructing a unit cell of dimension 18x19x20 (length in Å) and placing 1 atom of the element at the origin.[33] Approximately 1.2 million core hours of supercomputing power was used for all the DFT calculations combined. List of Hubbard *U* values used and the absolute atomic energies calculated are shown in **Table 2** and **Table 3**, respectively. We used the materials machine learning (MAML) python package to implement GAP, SNAP, qSNAP and MTP code.

Table 2: Hubbard *U* correction value table

Element	U-value (eV)
Li	0
O	0
Mn	3.9
Ni	6.2
Sc	0
Ti	0
V	3.25
Cr	3.7
Fe	5.3
Co	3.32
Cu	0

Table 3: Atomic Energy table

Element	Energy (eV)
Li	-0.3
O	-1.7
Mn	-4.6
Ni*	-3.0
Sc	-2.1
Ti	-2.4
V	-2.7
Cr	-4.5
Fe	-2.2
Co	-1.1
Cu	-0.2

*Ni atom energy obtained from MAML

Constructing ML-IAPs

For all ML-IAPs except ANNs, the input files were prepared in an identical way as described: We used the OUTCAR file generated from the successful DFT calculations to read converged energies, forces and positions of the relaxed structures using pymatgen objects. Since there are around 11000 structures, we used a loop script to write all these data serially into a JSON format, which stores the information in a concise, yet human-readable form.

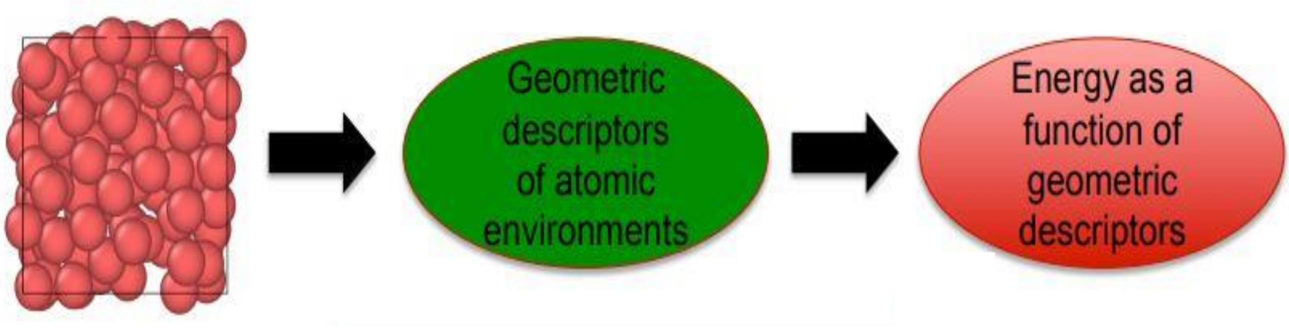
Geometric descriptors of atomic environments
Energy as a function of geometric descriptors

Figure 5 Schematic shows a typical ML-IAP training process[43]

1. **ANN**

   First, we collect suitable reference structures and their DFT calculated energies. For the input and output of atomic structures, energies and force data, AENET uses XCrySDen Structure Format (XSF) [34]. The energy of the structure required by the ANN potential training occurs as a comment at the top of the XSF file, which allows the file to be of a valid XSF format.

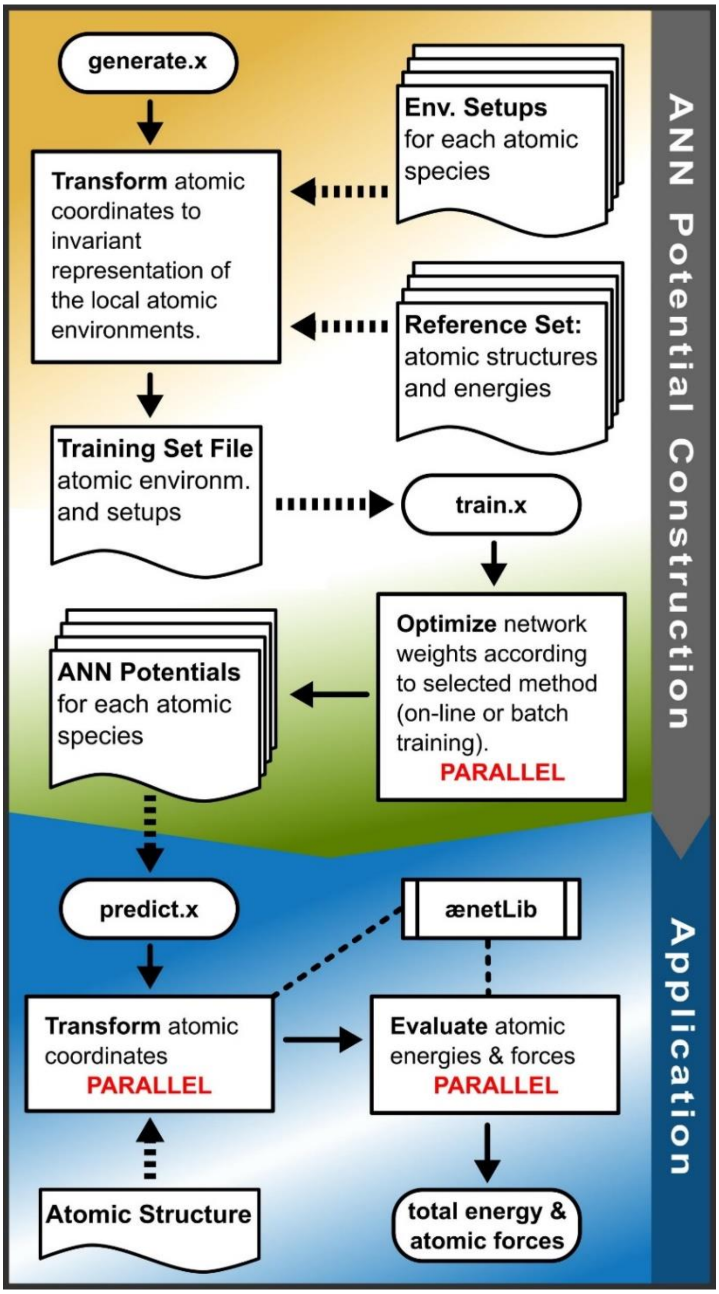
ANN Potential Construction
generate.x
Env. Setups for each atomic species
Transform atomic coordinates to invariant representation of the local atomic environments.
Reference Set: atomic structures and energies
Training Set File atomic environm. and setups
train.x
Optimize network weights according to selected method (on-line or batch training).
PARALLEL
ANN Potentials for each atomic species
Application
predict.x
ænetLib
Transform atomic coordinates
PARALLEL
Evaluate atomic energies & forces
PARALLEL
Atomic Structure
total energy & atomic forces

Figure 6: Flowchart of a typical ANN construction and prediction using the AENET package[9].

Training set generation

The cartesian atomic coordinates of the structures are transformed into an invariant, atom-centered basis using the Artrith-Urban-Ceder descriptor[11] based on a Chebyshev polynomials. Each atomic species in our dataset needs a structural fingerprint file which specifies the possible atom environment and parameters for the basis set. The executable `generate.x` was used to iterate over the list of structures in our dataset and transform each structure's coordinates using the method specified in the structural fingerprint file.

Training with `train.x`

Upon generation of invariant representations for our dataset, a training set file is generated. Using this, our ANN potentials can be fitted with the `train.x` executable. We specify the ANN architectures for all 11 chemical species and activation function types for the hidden layers. For the weight optimization, we use the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) training algorithm[35–38]. 10% of the dataset was set aside for testing. AENET calculates MAE and RMSE on the training and validation set after each training iteration. We used 300 epochs to train the data.

Prediction of energy and forces

Upon training, the ANN potential is constructed and stored in separate .nn files for each atomic species. Now, we use the `predict.x` tool to calculate the energies and forces on structures by using the structures as the input for our constructed ANN potential.

2. GAP

GAP implemented in the QUIP package [14] was used as part of the larger MAML package [39]. The GAPotential class in MAML was used to implement the SOAP kernels. The <train> method was used to train the data with GPR. The choice of hyperparameters used has a significant influence on the performance of GAP models[40,41] and we used the following:

`l_max` (int) = 6; The band limit of spherical harmonics basis functions

`n_max` (int) = 6; The number of radial basis functions

`atom_sigma` (float)= 0.5; The width of gaussian atomic density

`zeta` (float): = 4.0; Positive integer power of the SOAP kernel

`cutoff` (float) = 5; The cutoff radius (Å)

`delta` (float) = 1; Parameter to configure sparsification; the signal variance of noise n_sparse (int) = 200; Number of sparse points

covariance_type (str): = dot_product; The type of convariance function

sparse_method (str)= ‘cur_points’; Method to perform clustering in sparsification

3. MTP

Machine Learning Interatomic Potentials (MLIP) is the software package implementing MTP and can be integrated with MAML[39].

The MTPotential class in MAML was used to convert atomic positions into rotationally invariant basis functions by using rotationally covariant tensors. Linear regression is used to correlate energies with the basis functions. Methods used in the MTPotential class were:

convert-cfg: for converting VASP input/output files to the internal .cfg format of atomic configuration

train: training MTPs on .cfg datasets

mindist: computing minimal distances in configurations

calc-efs: To evaluate energy, forces, and stresses on a dataset.

We used the following hyperparameters with MTP:

max_dist (float): 5Å; The actual radial cutoff

radial_basis_size (int): 8; Number of radial basis function

max_iter (int): 500; The number of maximum iterations

energy_weight (float): 1; The weight of energy

force_weight (float): 0.01; The weight of forces.

4. SNAP

SNAP uses bispectrum components of atomic neighbor density functions as descriptors. Energy and forces are linear regression of the bispectrum components. The SNAP code comes along with the LAMMPS[42] installation and can be used through the MAML interface[39]. We initialize the SNAPotential class with atomic descriptors and model parameters, which are used to generate the bispectrum coefficients features for structures and to train the parameters. We used the following hyperparameters:

`Cutoff radius` (for each element): 4Å

`Element Weight`: 1

`rcutfac`: 0.5; Scale factor applied to all cutoff radii

`twojmax`: 6; Band limit for Bispectrum components

5. qSNAP

Formalism and implementation is similar to SNAP. The energy and forces have a quadratic dependence on the bispectrum components via the addition of an embedding energy term as discussed in **Chapter 2**. Implementation and hyperparameters are identical to SNAP, with an extra keyword, namely “quadratic=True”, being added to the SNAPotential class[39] of MAML.

CHAPTER 4: RESULTS

We have compiled plots for predictions using different ML-IAPs from **Figures 7-13**. We first train on a small subset of the entire ~11000 structure dataset to ensure that all the codes work appropriately and do a preliminary optimization of the hyperparamters. Also, progressively training on increasing dataset size gives an indication of the learning rate of the models used in this work. Hence, we choose a small dataset containing 364 structures with the composition $LiMn_{0.5}Ni_{0.5}O_2$ for training and 21 structures of the same composition for testing.

Mean Absolute Error (MAE) is calculated using the formula

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \qquad \text{XXXIII}$$

where y_i = prediction value, x_i = true value, n = total number of data points

Range Normalized MAE is a useful value to compare MAE across different dataset sizes and ML-IAP models. It was calculated using the formula

Range Normalized MAE = MAE/(Range of predicted energies) XXXIV

364 Training Data (4 atomic species): **Energy Plots on Train Data**

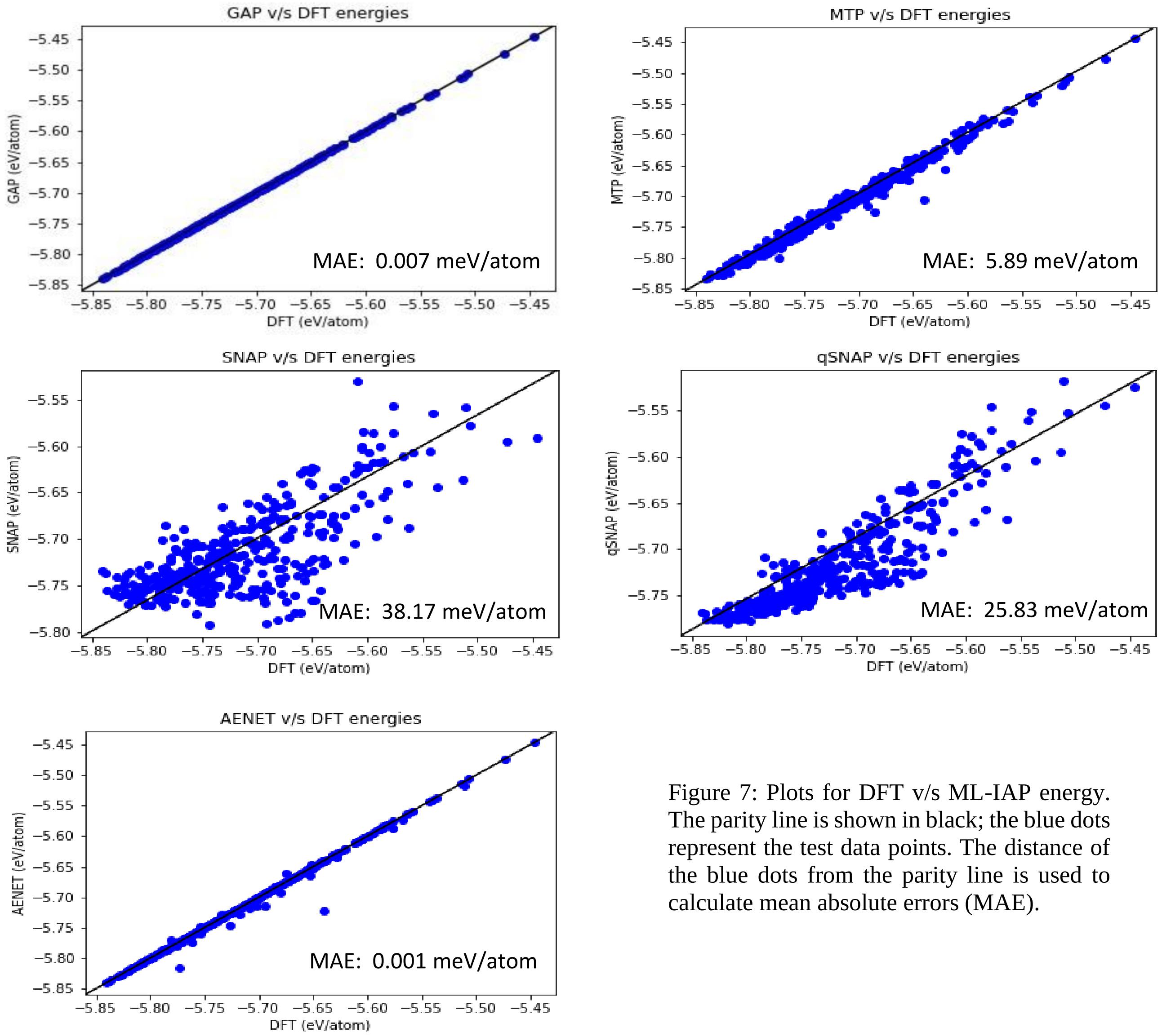

Figure 7: Plots for DFT v/s ML-IAP energy. The parity line is shown in black; the blue dots represent the test data points. The distance of the blue dots from the parity line is used to calculate mean absolute errors (MAE).

Figure 7 shows comparison of different ML-IAP performance on the same training data. We find AENET and GAP to be extremely accurate on predicting energies on the trained data, while MTP is relatively better than SNAP/qSNAP. The range normalized MAE for these were

GAP	0.000018
MTP	0.014725
SNAP	0.141370
qSNAP	0.107625
AENET	0.000002

Table 4: Column 1 is the ML-IAP and Column 2 shows the corresponding value of range normalized error in meV/atom

364 Training Data, 21 Test Data (4 atomic species): **Energy Plots on Test Data**

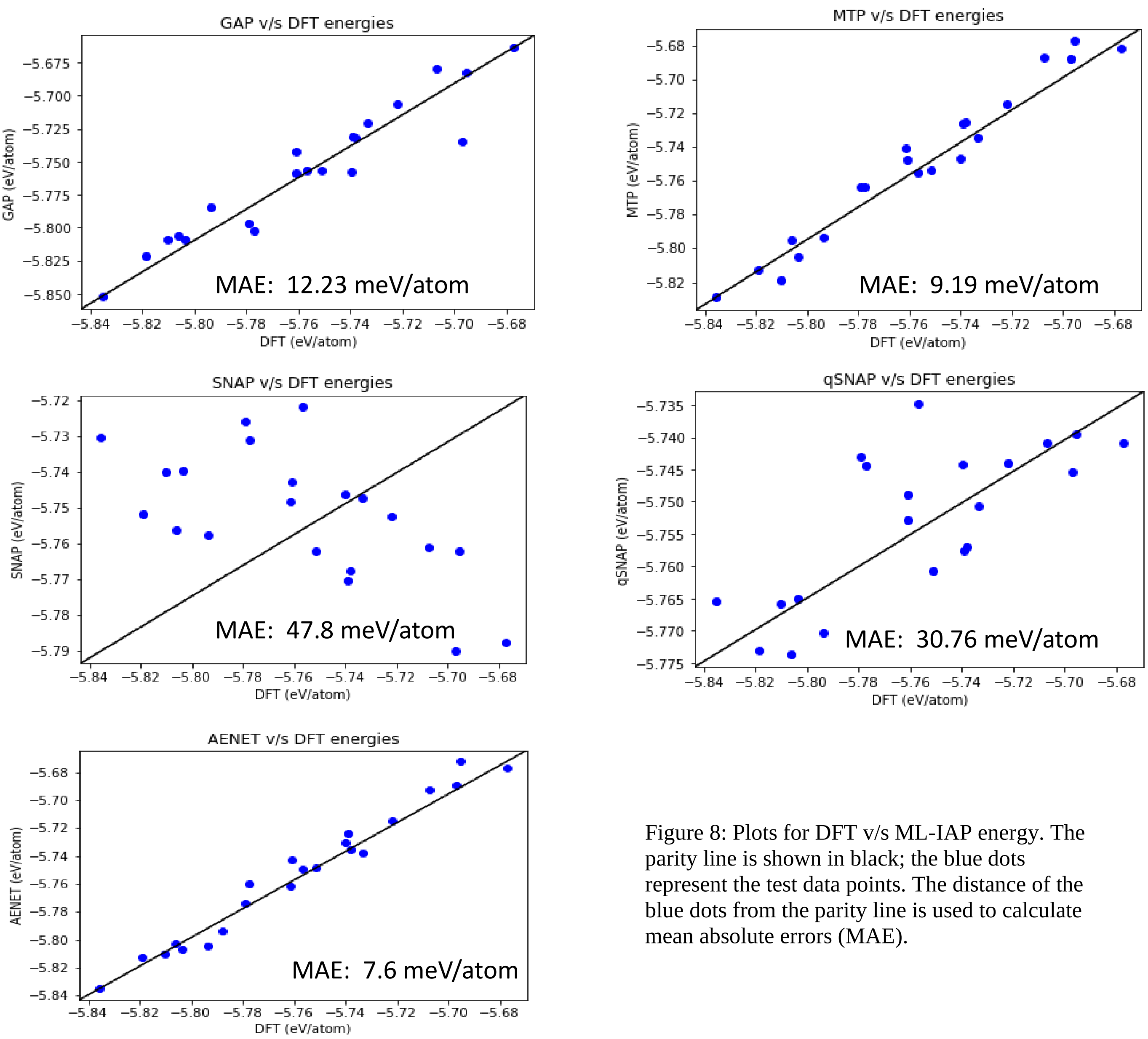

Figure 8: Plots for DFT v/s ML-IAP energy. The parity line is shown in black; the blue dots represent the test data points. The distance of the blue dots from the parity line is used to calculate mean absolute errors (MAE).

Figure 8 shows comparison of different ML-IAP performance on a test dataset. We find AENET, MTP and GAP to be fairly accurate on predicting energies on this unseen test data, while SNAP/qSNAP do not have reasonable fits. The range normalized MAE for these were

GAP	0.072
MTP	0.061
SNAP	0.683
qSNAP	0.769
AENET	0.048

Table 5: Column 1 is the ML-IAP and Column 2 shows the corresponding value of range normalized error in meV/atom

364 Training Data, 21 Test Data (4 atomic species): **Force Plots on Test Data**

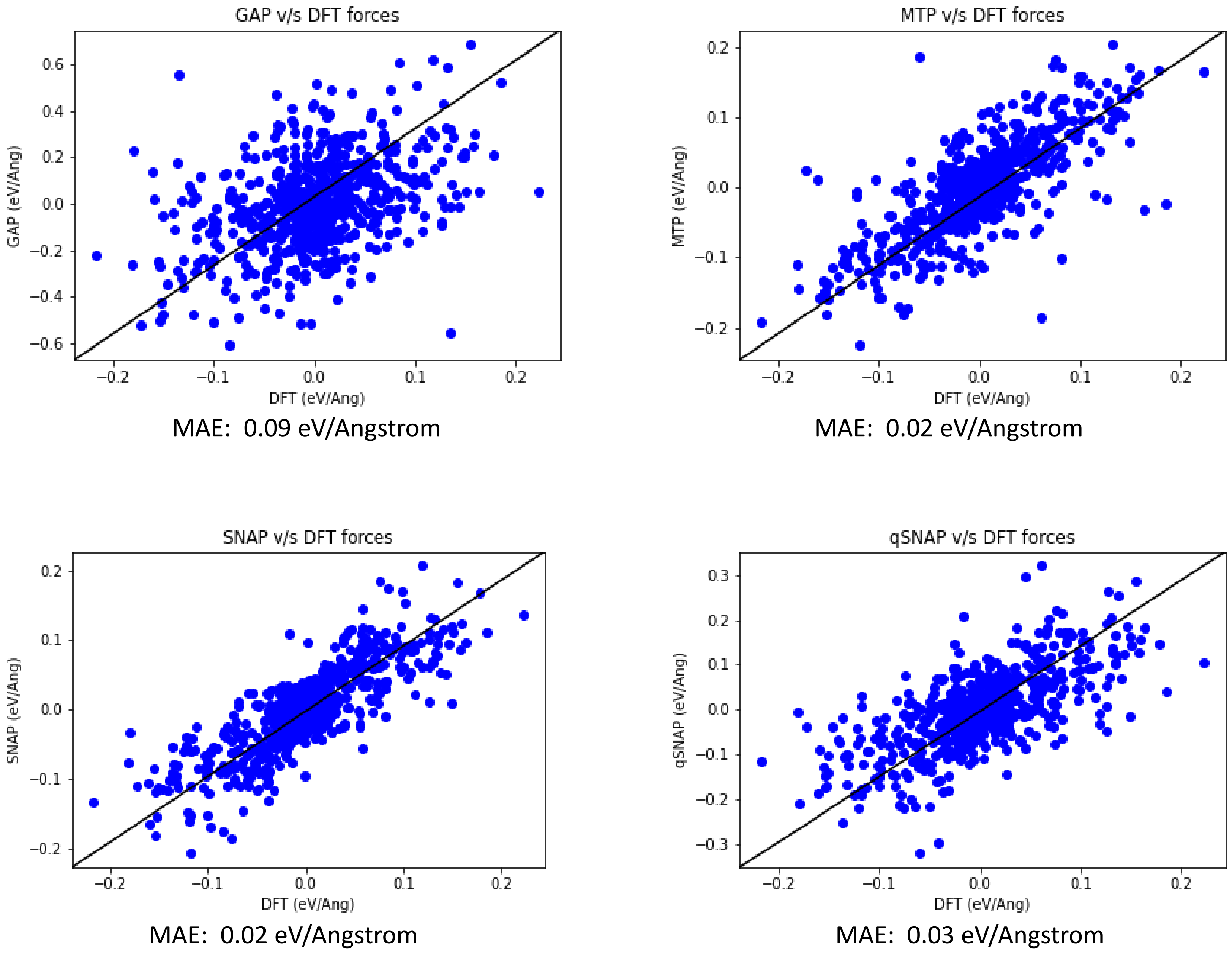

Figure 9: Plots for DFT v/s ML-IAP forces. The parity line is shown in black; the blue dots represent the test data points. The distance of the blue dots from the parity line is used to calculate mean absolute errors (MAE).

Figure 9 shows comparison of different ML-IAP performance on a test dataset. We find MTP and SNAP/qSNAP are fairly accurate on predicting forces on this unseen test data, while GAP has a larger error. AENET has not been used for force prediction yet.

Next, a random number generator was used to select 10% of structures from each n-transition metal(n-TM) system. Around 1005 structures were obtained, sampling all types of n-TM systems. This was divided into 900 + 105 for train + test respectively. The 900 dataset samples all possible combinations and is a good random representation of our entire dataset at (1/10)th of the size.

900 Train Data (11 atomic species) **Energy Plots on Train Data**

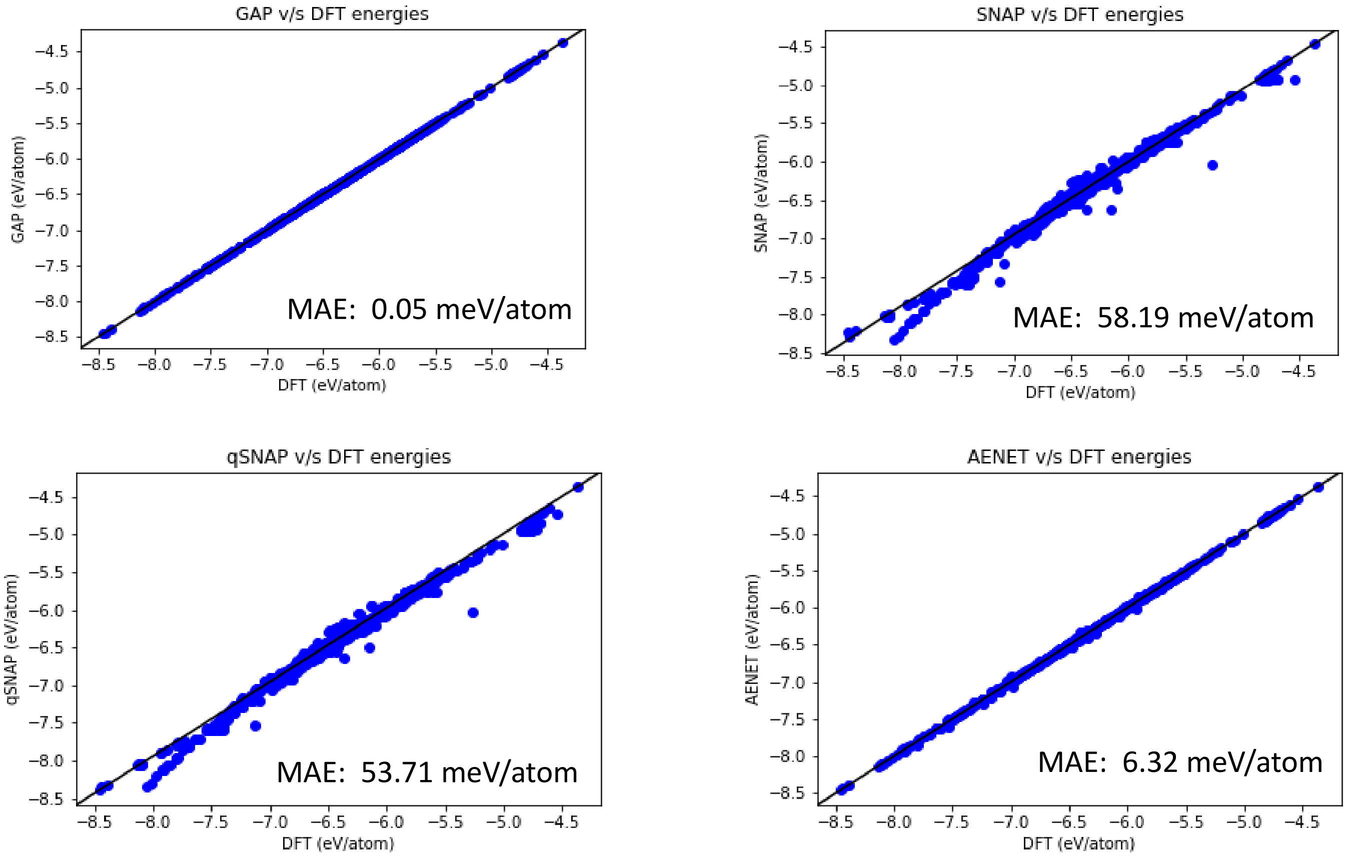

Figure 10: Plots for DFT v/s ML-IAP energy. The parity line is shown in black; the blue dots represent the test data points. The distance of the blue dots from the parity line is used to calculate mean absolute errors (MAE).

Figure 10 shows comparison of different ML-IAP performance on the same training dataset. We find GAP and AENET are fairly accurate on predicting energies, while SNAP/qSNAP show large errors, but notably having lower range normalized error compared to the 364 dataset. The range normalized MAE for these were

Table 6: Column 1 is the ML-IAP and Column 2 shows the corresponding value of range normalized error in meV/atom

GAP	0.00001
SNAP	0.01454
qSNAP	0.01342
AENET	0.00158

900 Training Data, 105 Test Data (11 atomic species) **Energy Plots on Test Data**

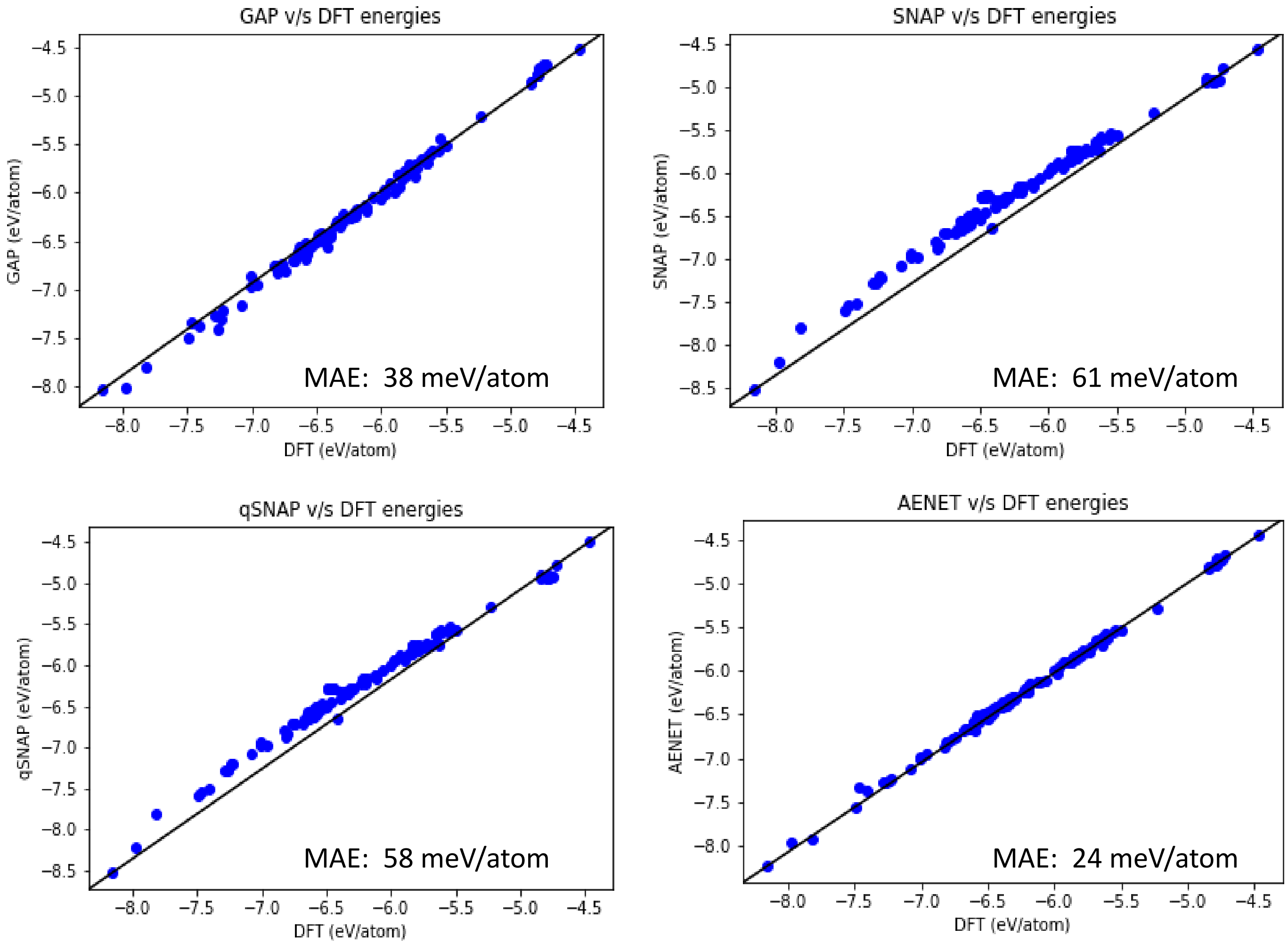

Figure 11: Plots for DFT v/s ML-IAP energy. The parity line is shown in black; the blue dots represent the test data points. The distance of the blue dots from the parity line is used to calculate mean absolute errors (MAE).

Figure 11 shows comparison of different ML-IAP performance on a unseen test dataset. We find AENET and GAP do better than SNAP/qSNAP in terms of accuracy, but SNAP and qSNAP have notably lower range normalized error compared to the 364 dataset. The range normalized MAE for these were

Table 7: Column 1 is the ML-IAP and Column 2 shows the corresponding value of range normalized error in meV/atom

GAP	0.010
SNAP	0.015
qSNAP	0.014
AENET	0.006

We find that training time varies non-linearly with number of TM species and number of structures being trained. Directly training on all the data (10k+ structures) lead to applications crashing out.

AENET using ANN is the only potential which we have been able to train and test on our entire 10k+ structure dataset as yet.

10046 train data (11 atomic species) **Energy Plot on Train Data**

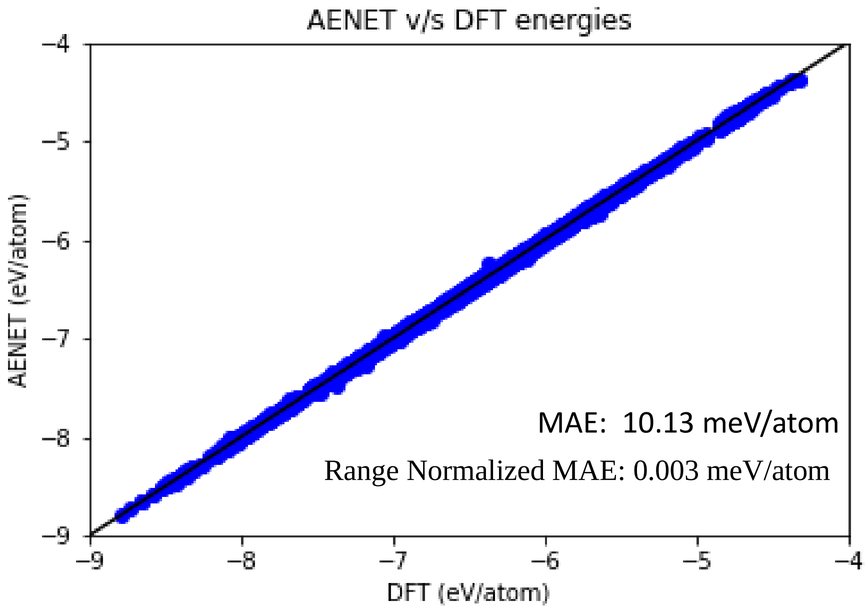

Figure 12: Plot of AENET predicted v/s DFT energy. The parity line is shown in black; the blue dots represent the test data points. The distance of the blue dots from the parity line is used to calculate mean absolute errors (MAE).

Figure 12 shows AENET having extremely high accuracy by the means of lower range normalized error compared to our previous smaller datasets.

To test the AENET potential (trained on 10k+ data) performance against varying complexities of test datasets, we chose 10 of each “n” transition metal structures as the test set and predicted their energies. The MAE plot is shown below

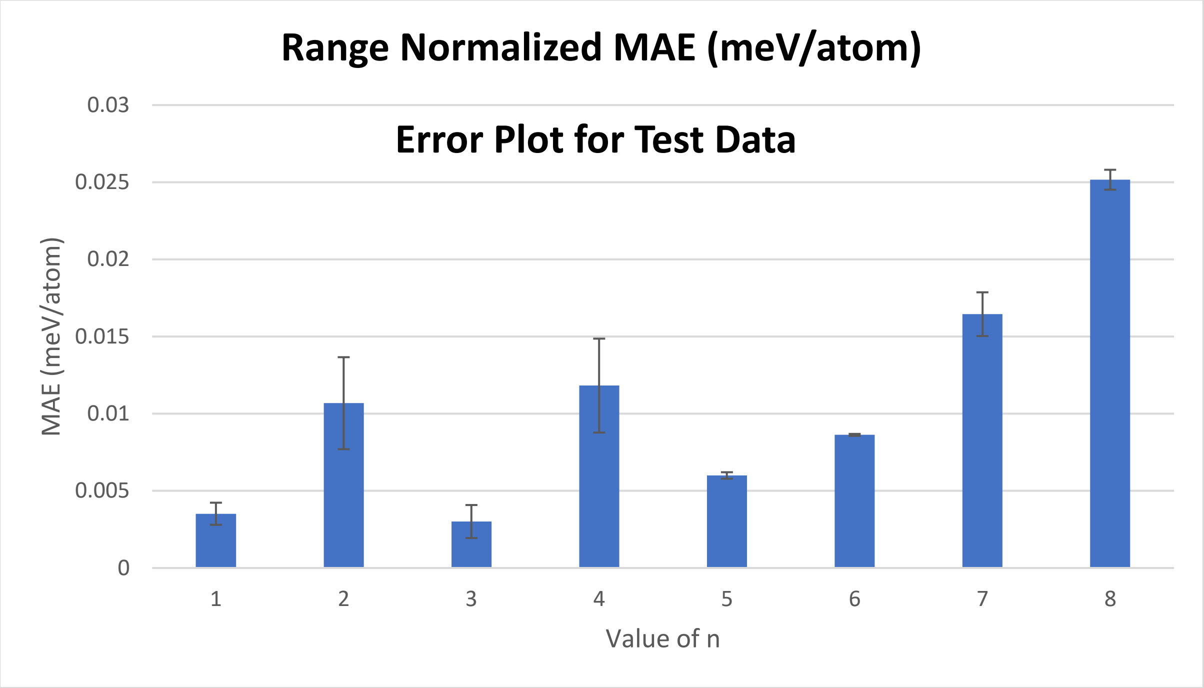

Figure 13: Values of MAE for “n” TM structures (n varying from 1 to 8)

Figure 13 shows the values of MAE for each n (1 to 8). It indicates that with our current model, the prediction works best for species with1 and 3 transition metals and worst for species with 7 and 8 transition metals.

CHAPTER 5: DISCUSSION

From the results shown in **Chapter 4**, we see that AENET and GAP give us consistently more accurate predictions on energy data compared to the other ML-IAPs. Note that MTP is currently not working on 11 atomic species structures, hence we are unable to use it for comparison yet. SNAP and qSNAP show accurate force predictions on the dataset trained, but do have significant errors on the energy predictions, especially on small datasets. The accuracy gets better with increasing training dataset size across the ML-IAP models. Importantly, AENET is able to train on large number of datapoints much faster than any other method, enabling us to train our entire dataset using it. With the initial hyper-parameters, our models are doing reasonably well.

Table 8: Time taken to train ML-IAP methods over different dataset complexities

ML-IAP method	Number of training data	Number of TM chemical species	Range Normalized energy MAE	Time for training
GAP	300	4	0.072	4 min
	900	11	0.010	4.5 hrs
SNAP, qSNAP	300	4	0.683, 0.769	10 min
	900	11	0.015, 0.014	30 min
MTP	300	4	0.061	12 min
AENET	300	4	0.048	15 min
	900	11	0.006	25 min
	10046	11	0.003	4.5 hrs

Table 9 shows estimated training time for the best models we have currently. Note that training time will vary significantly with the values of hyperparameters used. We observe that the range normalized MAEs reduce significantly as we increase number of training set structures for all models (except MTP, where we couldn't train on larger datasets). Time for prediction varies from a few seconds to a few minutes depending on the size of the test set. We observe that GAP is accurate and trains quickly on small datasets, while AENET is the fastest and most accurate on large datasets.

The training times taken by all our models are orders of magnitude faster than DFT calculations on all test structures, indicating the usefulness of ML-IAPs.

From **Figure 13**, we observe that our dataset lacks adequate number of structures containing 7 and 8 transition metal species, hence we need to fill this gap to build a better performing model over wider range of compositions.

CHAPTER 6: WORK PLANNED

If things had been normal (without Covid-19), the plan was to complete hyperparmater optimization using grid search by the time this thesis is written and use the optimum hyperparamters to get low MAE's (of the order of a few meV/atom). We also thought of training on cohesive energies instead of total energies to make the dataset normalized while training which might lower the resulting MAEs. Next we want to generate voltages as a function of various DRX compositions to predict potential candidates for battery electrode materials.

CHAPTER 7: CHALLENGES

Learning and becoming familiar with ML tools for this project was particularly challenging. Different IAPs use different file formats to represent their I/O and adjusting to them was a great learning. It takes more time to get the dataset ready in the right file format and specifications than to do the actual training. Being part of a new lab and figuring out most things from scratch was challenging, yet my best experience. Running calculations on smaller machines and keeping track of many systems in parallel due to harsh SERC rules on parallel computing jobs. File management of 11000 structures was immensely time consuming even though I wrote python codes to automate parts of it. This is because often, VASP calculations do not converge (and needs to be rerun with small fixes to input parameters) and keeping track of them amongst this huge dataset is not trivial.

ML-IAPs do not have provisions to update potentials on new datasets as yet. This implies that I need to retrain the entire dataset if I add new data, which is time consuming. Also, I was in the lab physically only for 3 months. The lab PC is much faster and better than our average laptops. Spending more time on the lab PC physically would have got the work done faster compared to remote computing. Our lab cluster is quite powerful and I could have finished my calculations in half the time that I actually took if its installation was on time. However, the cluster installation was delayed by a few months due to Covid-19.

CHAPTER 8: CONCLUSIONS & FUTURE WORK

We studied different ML-IAPs on disordered rocksalts, which are relevant to battery electrode applications, to model the highly complex configurational landscape. We enumerated disordered rocksalts, sampled all possible combinations and made a robust training dataset using GGA+U calculations, which gave us relaxed structures, energies, and forces. These were input into various machine learning IAPs using their respective descriptors. Potentials were trained and predictions were made. Comparisons on the efficacies of the different potentials were made and preliminary fits and conclusions were drawn. Structures with 1 or 3 transition metals have the most accurate predictions of energies using our potentials. Also, AENET predicts most accurately and quickly over large datasets, while GAP shows good accuracy over smaller datasets. SNAP and qSNAP have very good force predictions, and their MAE's of energies reduce considerably with increasing dataset size.

In the immediate future, we intend to do hyperparmater optimization using a grid search approach to reduce the MAEs. Later, we plan to generate voltages as a function of the 10 different (Li+9TM) compositions. We also want to find ways to speed up the training process, for example, by building a parallel GAP executable using openMPI. An important task is to get MTP working on 11 chemical species by debugging its code. We are working with the MAML code developers to resolve this issue. Finally, we intend to publish a research paper and communicate our work to the scientific community.

REFERENCES

[1] K. Kang, Ying Shirley Meng, Julien Bréger, Clare P Grey, Gerbrand Ceder, Electrodes with High Power and High Capacity for Rechargeable Lithium Batteries, Science. 311 (2006). https://doi.org/10.1126/science.1122152.

[2] B. Dunn, H. Kamath, J.-M. Tarascon, Electrical Energy Storage for the Grid: A Battery of Choices, Science. 334 (2011). https://doi.org/10.1126/science.1212741.

[3] T. Ohzuku, A. Ueda, M. Nagayama, Electrochemistry and Structural Chemistry of LiNiO2 (R3m) for 4 Volt Secondary Lithium Cells, Journal of The Electrochemical Society. 140 (1993). https://doi.org/10.1149/1.2220730.

[4] K. Mizushima, P.C. Jones, P.J. Wiseman, J.B. Goodenough, LixCoO2 (0&lt;x&lt;-1): A new cathode material for batteries of high energy density, Materials Research Bulletin. 15 (1980). https://doi.org/10.1016/0025-5408(80)90012-4.

[5] M. Obrovac, O. Mao, J.R. Dahn, Structure and electrochemistry of LiMO2 (M=Ti, Mn, Fe, Co, Ni) prepared by mechanochemical synthesis, Solid State Ionics. 112 (1998). https://doi.org/10.1016/S0167-2738(98)00225-2.

[6] J. Lee, A. Urban, X. Li, D. Su, G. Hautier, G. Ceder, Unlocking the Potential of Cation-Disordered Oxides for Rechargeable Lithium Batteries, Science. 343 (2014). https://doi.org/10.1126/science.1246432.

[7] R.J. Clément, Z. Lun, G. Ceder, Cation-disordered rocksalt transition metal oxides and oxyfluorides for high energy lithium-ion cathodes, Energy & Environmental Science. 13 (2020). https://doi.org/10.1039/C9EE02803J.

[8] V.L. Deringer, M.A. Caro, G. Csányi, Machine Learning Interatomic Potentials as Emerging Tools for Materials Science, Advanced Materials. 31 (2019). https://doi.org/10.1002/adma.201902765.

[9] N. Artrith, A. Urban, An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for TiO2, Computational Materials Science. 114 (2016). https://doi.org/10.1016/j.commatsci.2015.11.047.

[10] G. Montavon, G.B. Orr, K.-R. Müller, eds., Neural Networks: Tricks of the Trade, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. https://doi.org/10.1007/978-3-642-35289-8.

[11] N. Artrith, A. Urban, G. Ceder, Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species, Physical Review B. 96 (2017). https://doi.org/10.1103/PhysRevB.96.014112.

[12] J. Behler, Atom-centered symmetry functions for constructing high-dimensional neural network potentials, Journal of Chemical Physics. 134 (2011). https://doi.org/10.1063/1.3553717.

[13] A.M. Cooper, J. Kästner, A. Urban, N. Artrith, Efficient training of ANN potentials by including atomic forces via Taylor expansion and application to water and a transition-metal oxide, Npj Computational Materials. 6 (2020). https://doi.org/10.1038/s41524-020-0323-8.

[14] Bartok, A. P.; Csanyi, G. http://www.libatoms.org, (n.d.).

[15] A.P. Thompson, L.P. Swiler, C.R. Trott, S.M. Foiles, G.J. Tucker, Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials, Journal of Computational Physics. 285 (2015). https://doi.org/10.1016/j.jcp.2014.12.018.

[16] M.A. Wood, A.P. Thompson, Extending the accuracy of the SNAP interatomic potential form, The Journal of Chemical Physics. 148 (2018). https://doi.org/10.1063/1.5017641.

[17] S.M. Foiles, M.I. Baskes, M.S. Daw, Embedded-atom-method functions for the fcc metals Cu, Ag, Au, Ni, Pd, Pt, and their alloys, Physical Review B. 33 (1986). https://doi.org/10.1103/PhysRevB.33.7983.

[18] M.S. Daw, M.I. Baskes, Semiempirical, Quantum Mechanical Calculation of Hydrogen Embrittlement in Metals, Physical Review Letters. 50 (1983). https://doi.org/10.1103/PhysRevLett.50.1285.

[19] A. v. Shapeev, Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials, Multiscale Modeling & Simulation. 14 (2016). https://doi.org/10.1137/15M1054183.

[20] E. v. Podryabinkin, A. v. Shapeev, Active learning of linearly parametrized interatomic potentials, Computational Materials Science. 140 (2017). https://doi.org/10.1016/j.commatsci.2017.08.031.

[21] G.L.W. Hart, R.W. Forcade, Algorithm for generating derivative structures, Physical Review B. 77 (2008). https://doi.org/10.1103/PhysRevB.77.224115.

[22] G.L.W. Hart, R.W. Forcade, Generating derivative structures from multilattices: Algorithm and application to hcp alloys, Physical Review B. 80 (2009). https://doi.org/10.1103/PhysRevB.80.014120.

[23] G.L.W. Hart, L.J. Nelson, R.W. Forcade, Generating derivative structures at a fixed concentration, Computational Materials Science. 59 (2012). https://doi.org/10.1016/j.commatsci.2012.02.015.

[24] S.P. Ong, W.D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V.L. Chevrier, K.A. Persson, G. Ceder, Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis, Computational Materials Science. 68 (2013). https://doi.org/10.1016/j.commatsci.2012.10.028.

[25] V.I. Anisimov, J. Zaanen, O.K. Andersen, Band theory and Mott insulators: Hubbard *U* instead of Stoner *I*, Physical Review B. 44 (1991). https://doi.org/10.1103/PhysRevB.44.943.

[26] W. Kohn, L.J. Sham, Self-Consistent Equations Including Exchange and Correlation Effects, Physical Review. 140 (1965). https://doi.org/10.1103/PhysRev.140.A1133.

[27] P. Hohenberg, W. Kohn, Inhomogeneous Electron Gas, Physical Review. 136 (1964). https://doi.org/10.1103/PhysRev.136.B864.

[28] J.P. Perdew, K. Burke, M. Ernzerhof, Generalized Gradient Approximation Made Simple, Physical Review Letters. 77 (1996). https://doi.org/10.1103/PhysRevLett.77.3865.

[29] P.E. Blöchl, Projector augmented-wave method, Physical Review B. 50 (1994). https://doi.org/10.1103/PhysRevB.50.17953.

[30] G. Kresse, J. Furthmüller, Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set, Computational Materials Science. 6 (1996). https://doi.org/10.1016/0927-0256(96)00008-0.

[31] G. Kresse, J. Furthmüller, Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set, Physical Review B. 54 (1996). https://doi.org/10.1103/PhysRevB.54.11169.

[32] S.P. Ong, W.D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V.L. Chevrier, K.A. Persson, G. Ceder, Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis, Computational Materials Science. 68 (2013). https://doi.org/10.1016/j.commatsci.2012.10.028.

[33] A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K.A. Persson, Commentary: The Materials Project: A materials genome approach to accelerating materials innovation, APL Materials. 1 (2013). https://doi.org/10.1063/1.4812323.

[34] A. Kokalj, The XSF Format Specification, 2009. <http://www.xcrysden.org/doc/ XSF.html>, (n.d.).

[35] C.G. BROYDEN, The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations, IMA Journal of Applied Mathematics. 6 (1970). https://doi.org/10.1093/imamat/6.1.76.

[36] R. Fletcher, A new approach to variable metric algorithms, The Computer Journal. 13 (1970). https://doi.org/10.1093/comjnl/13.3.317.

[37] D. Goldfarb, A family of variable-metric methods derived by variational means, Mathematics of Computation. 24 (1970). https://doi.org/10.1090/S0025-5718-1970-0258249-6.

[38] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization, Mathematics of Computation. 24 (1970). https://doi.org/10.1090/S0025-5718-1970-0274029-X.

[39] Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen, J. Behler, G. Csányi, A. v. Shapeev, A.P. Thompson, M.A. Wood, S.P. Ong, Performance and Cost Assessment of Machine Learning Interatomic Potentials, The Journal of Physical Chemistry A. 124 (2020). https://doi.org/10.1021/acs.jpca.9b08723.

[40] V.L. Deringer, G. Csányi, Machine learning based interatomic potential for amorphous carbon, Physical Review B. 95 (2017). https://doi.org/10.1103/PhysRevB.95.094203.

[41] W.J. Szlachta, A.P. Bartók, G. Csányi, Accuracy and transferability of Gaussian approximation potential models for tungsten, Physical Review B. 90 (2014). https://doi.org/10.1103/PhysRevB.90.104108.

[42] S. Plimpton, Fast Parallel Algorithms for Short-Range Molecular Dynamics, Journal of Computational Physics. 117 (1995). https://doi.org/10.1006/jcph.1995.1039.

[43] ML-IAP Training Process, (n.d.).

APPENDICES

Hundreds of lines of python code were written for this project. The Jupyter Notebooks used are uploaded in this OneDrive link for ready reference

https://indianinstituteofscience-my.sharepoint.com/:u:/g/personal/nidhishsagar_iisc_ac_in/EQqIinX5PgFHji02Ap6VSwcBlq6uNrofWzUVuCl_48RJGw?e=4DrRxB