NORTHWESTERN UNIVERSITY

Deep Learning Methodologies for Scientific Knowledge Discovery

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Engineering

By

Dipendra Jha

EVANSTON, ILLINOIS

March 2020

ABSTRACT

Deep learning is a new area of machine learning research that allows deep neural networks composed of multiple processing layers to learn representations of data with multiple levels of abstraction. Deep learning has helped in achieving the objective of pushing machine learning closer to one of its original goals of artificial intelligence. It has become state-of-the-art machine learning technique in the fields of computer vision, speech recognition and text processing. Although it has enjoyed great success in the fields of computer science, its application in scientific fields has been very limited. This is mainly due to the scarcity of and complex nature of scientific datasets since they are collected from expensive and time-consuming scientific experiments and computations. This thesis explores how to design and build novel deep neural network architectures that can handle the challenges associated with such datasets and automatically learn the underlying science behind those scientific phenomena using deep learning, for the advancement of the overall process of scientific knowledge discovery.

Acknowledgements

First, I would like to thank my advisors- Prof. Alok Choudhary, Prof, Ankit Agrawal and Prof. Wei-keng Liao, for continuously advising me and for their help, support and guidance during my Ph.D. at Northwestern University. Next, I would like to thank my first academic advisor at Northwestern University- Prof. Fabian Bustamante, for providing me the opportunity to pursue my Ph.D. at Northwestern University. This Ph.D. thesis involved several collaborations with scientists and researchers from Northwestern University, University of Chicago, Argonne National Lab, Carnegie Mellon University and National Institute of Standards and Technology (NIST). I would like to thank all my collaborators, especially Logan Ward, for their continuous help, support and advice, during our project collaborations. Next, I would take the opportunity to thank the funding agencies for financially supporting my Ph.D. work. This dissertation was supported by the NIST awards 70NANB14H012 and 70NANB19H005 from U.S. Department of Commerce, National Institute of Standards and Technology as part of the Center for Hierarchical Materials Design (CHiMaD), AFOSR MURI award FA9550-12-1-0458, NSF award CCF-1409601, and DOE awards DE-SC0007456, DE-SC0014330 and DE-SC0019358.

Pursuing a Ph.D. degree at Northwestern University has been a long and challenging endeavor that required a lot of passion, hard work, dedication and patience. It would not have been possible without the help and support from my family and close friends. I would like to thank my parents and brother for their continuous support throughout this

journey. I feel really grateful to my close friends, especially Rishabh Gemawat, Sabita Acharya, Arindam Paul, Reda Al-Bahrani, Mas-ud Hussain, Amar Krishna, Rosalia Rizo Ortega, Prakash Paudyal, Prakash Shrestha and Anastasiia Tsarenko, for continuously supporting me and listening to my frustrations and grievances. Last but not least, I would like to thank all my teachers, classmates, labmates, roommates, and colleagues from my primary school to Northwestern, who have always believed in me and encouraged me to accomplish this goal.

Table of Contents

ABSTRACT 3

Acknowledgements 4

Table of Contents 6

List of Tables 9

List of Figures 13

Chapter 1. Introduction 24

1.1. Challenges 27

1.2. Problem Statement 29

1.3. Thesis Organization 29

Chapter 2. Background 31

2.1. Deep Learning 31

2.2. Deep Neural Networks 31

2.3. Stochastic Gradient Descent 33

2.4. Machine learning for scientific discovery 34

2.5. Deep learning for scientific discovery 35

Chapter 3. ElemNet: Deep Learning the Chemistry of Materials From Only Elemental Composition 37

3.1. Introduction 37

3.2. Methods 41

3.3. Experimental Results 45

3.4. Discussion 67

Chapter 4. IRNet: A General Purpose Deep Residual Regression Framework for Materials Discovery 69

4.1. Introduction 69

4.2. Background 73

4.3. Design 75

4.4. Experimental Results 80

4.5. Conclusion and Future Work 93

Chapter 5. Enhancing Materials Property Prediction by Leveraging Computational and Experimental Data using Deep Transfer Learning 95

5.1. Introduction 95

5.2. Methods 100

5.3. Experimental Results 102

5.4. Discussion 120

Chapter 6. Extracting Grain Orientation from EBSD Patterns of Polycrystalline Materials using Convolutional Neural Networks 124

6.1. Introduction 124

6.2. Background 130

6.3. Design 132

6.4. Experimental Results 137

6.5. Conclusion 140

Chapter 7. Peak Area Detection Network for Directly Learning Phase Regions from Raw X-ray Diffraction Patterns 142

7.1. Introduction 142

7.2. Background 145

7.3. Design 147

7.4. Experimental Results 156

7.5. Conclusion 161

Chapter 8. Conclusion and Future Works 165

8.1. Conclusion 165

8.2. Future Works 166

References 169

List of Tables

3.1 *ElemNet* Architecture. Considering the Input as the 0th layer, types and positions of different types of fully connected and dropouts are shown below. Dropout layers are used to prevent overfitting and they are not counted as a separate layer. We used ReLU as the activation function. 43

3.2 Benchmarking our deep learning model – *ElemNet* – against conventional machine learning approaches. We trained several conventional ML models such as Linear Regression, SGDRegression, ElasticNet, AdaBoost, Ridge, RBFSVM, DecisionTree, ExtraTrees, Bagging and Random Forest. Out of them, Random Forest performed the best with and without using physical attributes. Here, we show the results from our deep learning model and the best conventional ML model- Random Forest, in our study for both types of model inputs (with and without physical attributes), along with the type of input used, mean absolute error (MAE) on the test set, training time on the training set, and prediction time on the entire test set (25,662 entries). All the models are trained and tested using a ten-fold cross validation. All timings are on a single (logical) CPU core of an NVIDIA DIGITS DevBox with

a Core i7-5930K 6 Core 3.5GHz desktop processor with 64GB DDR4 RAM and 4 TITAN X GPUs with 12GB of memory per GPU, except the deep learning models. 48

3.3 Subset of Potential Stable Compounds Predicted using *ElemNet*. Out of the 450M predictions, we determined the number of systems where *ElemNet* identifies at least one new potential stable compound. We list the number of binary, ternary, and quaternary systems for several categories of compounds along with the two most stable predictions. We validated some of the these compounds- NaY_2F_7 and KY_2F_7 using DFT computations by leveraging crystal structures of existing materials with similar stoi-chemistry; we found them to be stable using DFT, further literature search revealed that they have already been synthesized recently. Our model predicts Cu_2O as the only new binary oxide which is a known compound but was not in our training set. 66

4.1 Detailed configurations for different depths of network architecture. The notation [...] represents a stack of model components, comprising a single (FC: fully connected layer, BN: batch normalization, Re: ReLU activation function) sequence in the case of IRNet and multiple such sequences in the case of SRNet. Each such stack is followed by a shortcut connection. 77

4.2 Performance of deeper residual networks for the *design problem*. Test errors are MAE in eV/atom. Increased depth of residual network

architectures leads to improved performance for both stacked and individual residual networks. The individual residual network (IRNet) clearly outperforms the stacked residual network (SRNet), achieving significantly lower MAE. 86

4.3 Performance of Traditional ML Approaches for the *design problem*. We performed extensive grid search for hyperparameter tuning for all the listed ML models. Test errors are MAE in eV/atom. 87

4.4 Performance on OQMD-C and MP-C datasets of our DNN models vs. 10 traditional ML approaches for regression problems: Linear Regression, Lasso, Ridge, Decision Tree, Adaboost, KNeighbors, ElasticNet, SGD Regression, Random Forest and Support Vector, with extensive grid search used to tune hyperparameters for each. Test errors are MAE in eV/atom. 91

4.5 Performance from combinatorial search. Our 17-layer IRNet, when trained on OQMD-SC-ICSD, predicts formation enthalpy (stability) more accurately than Random Forest for all three types of crystal structures considered. 92

5.1 *ElemNet* model architecture used for training different models. 102

5.2 Performance of the ElemNet models from ten-fold cross-validation in MAE (eV/atom). 105

5.3 Holdout test set performance of the ElemNet models in MAE (eV/atom). 105

5.4 Performance of ElemNet models on the experimental data in MAE (eV/atom). 116

6.1 Mean Disorientation Error (MDE) and mean symmetrically equivalent orientation absolute error (MSEAE) using different models and loss functions. 138

List of Figures

3.1 Comparison of deep learning approach with conventional ML approach for prediction of materials properties. The conventional ML approach for predictive modeling of materials properties involve representing the material composition in the model input format, manual feature engineering and selection by incorporating the required domain knowledge and human intuition by computing the important chemical and physical attributes of the constituent elements, and applying ML techniques to construct the predictive models. Our deep learning based predictive approach directly learns to predict properties of materials such as the formation enthalpy from their elemental compositions with better accuracy and speed than conventional ML approaches. 38

3.2 Performance of deep learning models of different depths in model architecture. The models are trained and tested on the lowest DFT-computed formation enthalpy of 256,622 compounds. Here, we present the impact of depth of architecture for one sample split from our ten-fold cross validation. (a) shows the mean absolute error (MAE) on the test dataset of 25,662 compounds with unique compositions at different epochs for one split from the cross validation. The DNN models keep

learning new features from the training dataset with the increase in the number of layers up to 17 layers, after which they begin to slowly overfit to the training data. (b) shows the MAE for different depths of deep learning model architectures and also illustrates mean absolute error of the best performing conventional ML model trained using physical attributes computed on the same training and test sets. The deep learning model start outperforming the best performing conventional ML model with an architecture depth of 10 layers, achieving the best performance at 17 layers, we refer to the best performing DNN model as *ElemNet*. The detailed architecture for *ElemNet* is available in the Method section. 46

3.3 Impact of training dataset size on the prediction accuracy of *ElemNet* (DNN model) using elemental compositions only and the best conventional ML model, Random Forest, with either raw elemental compositions (RF-Comp) and physical attributes (RF-Phys). The training and test sets are created during the ten-fold cross validation from the OQMD; different random subsets of the training set with sizes ranging from 464 to $230,960$ are created using a logarithmic spacing for this analysis. Training dataset size has more impact on ElemNet (deep learning model) compared to Random Forest models, but *ElemNet* performs better than Random Forest for all size greater than $4k$. 50

3.4 Error analysis of the predictions using *ElemNet* of a test set containing 25,662 compounds from our ten-fold cross validation. The left side

shows that the predicted values are very close to the DFT-computed values. The right side illustrates the cumulative distribution function (CDF) of the prediction errors for *ElemNet* and Random Forest (the best performing conventional ML model) with elemental fractions (RF-Comp) and physical attributes(RF-Phys). Our error analysis demonstrates that the deep learning performs very well, achieving an MAE of 0.050 ± 0.000 eV/atom; predicting with an absolute error of less than 0.120 eV/atom for 90% of the compounds in our test set (right). 53

3.5 Predicted phase diagrams from the hold-out test. These charts show the convex hulls predicted for the (a) Ti-O binary and (b) Na-Mn-O from ML models that were trained without any data from each system in their training set. We compare the performance of a Random Forest model trained using only element fractions (RF-Comp), RF trained using physical features (RF-Phys) and a deep learning model (*ElemNet*). Each vertex on the convex hull corresponds to the composition of a stable compound. The black lines on each chart show the OQMD convex hull. We find that the deep learning model has the fewest predictions outside the regions where compounds are known to form, for both the Ti-O and Na-Mn-O phase diagrams. 55

3.6 Visualization of the activations of different materials in *ElemNet*. Each frame shows a 2D projection (using PCA) of the activations of different materials in several layers of *ElemNet*, which shows which materials have similar representations. The upper row shows the activations of

different elements, where each point is a different element and is colored by the group number. The second row shows the activations of AB compounds formed of group I and II metals combined with S (group VI) or Cl (group VII). We note that elements from the same group in the periodic table, such as alkali metals, are clustered together in the early layers of the network, and that later layers reflect properties related to combinations of elements (e.g., charge balance). 59

4.1 Three types of 17-layer networks. Each "layer" is a fully connected neural network layer with size as described in Table 4.1; all but the last are followed by batch normalization and ReLU. A *plain network* simply connects the output of each layer to the input of the next. A *stacked residual network* (SRNet) places a shortcut connection after groups of layers called stacks. An *individual residual network* (IRNet) places a shortcut connection after every layer. 76

4.2 Test error curve for various plain networks for the *design problem*. Batch normalization before activation function (FC+BN+ReLU) improves performance significantly. 83

4.3 Test error curve for deeper plain networks for the *design problem*. Performance degrades with network depth, even in the presence of batch normalization. 84

4.4 Impact on residual learning for the *design problem*. Both residual networks outperform the plain network, and the individual network

outperforms the stacked network for all depths of network. We observe similar trends even in the case of training error curves for all types of networks of all depths; the IRNet converges faster than the SRNet and Plain Network for all depths. 85

4.5 Cumulative distribution function (CDF) of the prediction errors for the *design problem*. Deep learning (IRNet) performs significantly better than the traditional ML approach, Random Forest, achieving a 90th percentile MAE of 0.081 eV/atom vs. 0.158 eV/atom for Random Forest. 88

5.1 DFT-computation error analysis of different DFT-computed datasets against the experimental observations. We compared the experimental formation energies of 463 materials against their corresponding formation energies from OQMD (a), Materials Project (b) and JARVIS (c) datasets available in Matminer [**1**]. The MAE in OQMD, Materials Project and JARVIS for formation energies against experimental observations are 0.083 eV/atom, 0.078 eV/atom and 0.095 eV/atom respectively. (d) The 50th percentile and 90th percentile MAE for OQMD, Materials Project and JARVIS are 0.057 eV/atom and 0.201 eV/atom, 0.055 eV/atom and 0.171 eV/atom, and 0.068 eV/atom and 0.190 eV/atom, respectively. 98

5.2 Proposed approach of deep transfer learning. First, a deep neural network architecture (ElemNet) is trained from scratch, by initializing

model parameters randomly from a uniform distribution, on a big DFT-computed source dataset (OQMD). This allows the model to learn the input data representation and capture the essential chemistry from the big source training data. Since this model is trained from scratch on OQMD, we refer to this as OQMD-SC model. Next, we train a deep neural network architecture (ElemNet) on other smaller target dataset, such as experimental dataset, using transfer learning. Here, the model parameters are initialized using the values from OQMD-SC, and then fine-tuned using the corresponding target dataset. 100

5.3 Impact of training data size on performance of models trained from scratch and using transfer learning (mean and s.d.). The models are trained on the experimental dataset and the results are aggregated from a ten-fold cross-validation. For each cross validation, first we split the complete dataset randomly into training and test (validation) set in the ratio of 9 : 1. Next, we fixed the test (validation) set and changed the size of the training set from 10% to 100%. OQMD-SC represents the model trained from scratch on OQMD dataset, EXP-SC represents the prediction error of the model trained from scratch, and EXP-TL represents the prediction error using transfer learning from the OQMD-SC model. 109

5.4 Prediction error analysis of OQMD-SC model using a test set containing $34,145$ samples from a 9:1 random split of OQMD. OQMD-SC model is trained from scratch (with random weight initialization from a

uniform distribution) using a 9:1 random split of training and test set from the OQMD. Since the dataset is large, the model is able to automatically capture the essential chemical and physical interactions between different elements; hence, providing robust predictions while compared against OQMD. 112

5.5 Prediction error analysis using OQMD-SC model on the other three datasets. The OQMD-SC model is trained from scratch (with random weight initialization from a uniform distribution) using a 9:1 random split of training and test set from the OQMD. Although OQMD-SC model has low prediction error against the test set from OQMD, the prediction error is high if we compare against other datasets. This is because of the difference in the DFT-computations used in JARIVS and Materials Project, and OQMD. Since DFT-computations from the OQMD has an error of around 0.1 eV/atom against experimental observations, this error is inherent in the OQMD-SC model leading to higher prediction errors. 113

5.6 Prediction error analysis on the test (validation) sets from the ten-fold cross validation (except for OQMD-SC). For OQMD-SC, ElemNet model is trained from scratch using a 9:1 random split of training and test (validation) set from the OQMD. For other datasets, we aggregate the prediction errors on the test (validation) sets from the ten-fold cross-validation for each model. The four rows represent the four datasets- (a-c) JARVIS (JAR), (d-f) Materials Project (MP), (g-i)

OQMD and (j-l) the experimental observations (EXP); first (a, d, g and j) and second (b, e, h and k) columns of each row show the predictions using the model trained on the particular dataset from scratch (SC) and using transfer learning (TL), respectively, the third column (c, f, i and l) shows the respective CDF of the prediction errors using models trained from scratch (SC) and using transfer learning (TL), trained on the particular dataset. 114

5.7 Prediction error analysis for the ElemNet architecture trained using OQMD and evaluated on the experimental data containing $1,963$ observations. When training from scratch, the weights are initialized randomly from a normal distribution; for transfer learning, the model is first trained on the OQMD dataset and then fine-tuned using the corresponding dataset. 115

5.8 Prediction error analysis on the experimental dataset containing $1,963$ observations using different models. For the models trained using experimental dataset, the predictions on the test sets are aggregated from validation sets using ten-fold cross-validation. For the models trained using JARVIS and Materials Project, since we have ten models from the ten-fold cross-validation during training, we take the mean of their prediction for each data point in the experimental dataset. For OQMD-SC, we make ten predictions on each point in the experimental dataset and take the mean. The four rows represent the four datasets- (a-c) JARVIS (JAR), (d-f) Materials Project (MP), (g-i) OQMD and

(j-l) the experimental observations (EXP); first (a, d, g and j) and second (b, e, h and k) columns of each row show the predictions using the model trained on the particular dataset from scratch (SC) and using transfer learning (TL), respectively, the third column (c, f, i and l) shows the respective CDF of the prediction errors using models trained from scratch (SC) and using transfer learning (TL), trained on the particular dataset. 117

5.9 Analysis of the activations from the first hidden layer of the ElemNet architecture for the magnetic vs non-magnetic class (1 and 0) from JARVIS dataset. The four columns represent the models trained using four different datasets- (a, e and i) JARVIS (JAR), (b, f and j) Materials Project (MP), (c, g and k) OQMD and (d, h and l) the experimental observations (EXP); the first (a-d) and second (e-h) rows represent the models trained from scratch (SC) and using transfer learning (TL), while the third row (i-l) represents the ROC curves from the Logistic Regression model trained using all activations from the same hidden layer (the corresponding AUC values are shown in brackets) on the respective datasets. The scatter plots demonstrate the first two principal components of the activations using principal component analysis (PCA) technique. 121

6.1 Schematic of the EBSD geometry. 125

6.2 EBSD pattern from Iron (a) Experimental and (b) Simulation. 126

6.3 OMNet: CNN architectures for learning multiple outputs. 136

6.4 Loss and mean disorientation error using different loss functions. (a) shows the training loss and mean disorientation error (MDE) on training set and test set for MDE as the loss function. (b) shows the loss and MDE for the hybrid loss function of the sum of mean absolute error (MAE) and MDE. 137

7.1 1D XRD Patterns from SLAC and NIST. The XRD patterns from SLAC contains highly irregular noise while the noise in the case of XRD patterns from NIST is a constant function of 2θ. 148

7.2 Distribution of class labels for the two XRD datasets. XRD Patterns are collected for the same composition space of Sn-Ti-Zn-O from both NIST and SLAC; hence, they refer to same samples. 149

7.3 PADNet model architectures for the XRD patterns from SLAC and NIST. Since both datasets refer to the same composition space of Sn-Ti-Zn-O and have same samples, we constrained both models to have same number of model parameters and same architecture. PADNet for NIST is composed of two convolutional graphs to handle the two XRD patterns compared to the PADNet for SLAC having one convolutional graph since SLAC outputs one XRD image. 151

7.4 Peak Area Detection Component with Slope Filters: This component contains slope filters which help in peak detection by measuring the difference in slope across different symmetries. The slope filters has two

regions: blue representing -1 and red representing 1. Since they are symmetric, they can effectively detect the high slope areas containing peaks. 151

7.5 Convolutional Graph: The component of the CNN network for the raw 2D XRD pattern in the input. 151

7.6 Background and processed XRD images: I is the original XRD pattern on the left of (a) and (b). The top row of subfigures represent the background (MF, CS) and the bottom row of subfigures shows the XRD patterns after background removal using the two techniques (I-MF and I-CS). We used a filter size of 200 for both the minimum filter and the convolutional mean filter for both cases. For SLAC, the raw XRD pattern is similar to the background images using the two techniques; this illustrates that SLAC image contains high background noise. For NIST, the pattern after background removal look similar to the raw pattern since the background is very small. This concurs with the domain expertise, thereby suggesting that the proposed background detection module is working as expected. 163

7.7 Performance of PADNet using a ten-fold cross validation (mean and standard deviation). The uniform performance across all test pattern types exhibits the efficacy of PADNet for phase region classification directly from raw 2D XRD pattern. 164

CHAPTER 1

Introduction

Deep learning is a new area of machine learning research that has helped in achieving the objective of pushing it closer to one of its original goals of artificial intelligence [**2**]. Deep learning allows deep neural network (DNN) models composed of multiple processing layers (deep neural network architecture) to learn representations of data with multiple levels of abstraction [**3**]. Deep learning has gained significant attention in the field of computer science with breakthrough results in computer vision, speech recognition and text processing [**4, 5, 6, 7**]. Although the basic algorithmic approach in deep learning remains the same, such progress has been possible mainly due to availability of large datasets and increase in computational power for training. Due to their flexible DNN architecture with large parameter set, deep learning models can automatically capture the high level non-linear mappings between input features and output values through multi-layered feature abstraction. They are now the state-of-the-art models in both supervised and unsupervised object detection and recognition tasks [**4, 5, 6, 7**]. Although it has enjoyed great success in the fields of computer science, its application in scientific fields has been limited. This thesis study how to design and develop deep learning models for the advancement of knowledge discovery from scientific datasets for accelerating scientific design and discovery.

Many of the technical advances we see today from cellphones, laptops, supercomputers to supersonic airplanes and interstellar rockets, all are the results of advancements

of scientific knowledge discovery. Most of the scientific problems we face today are due to the limitation of current scientific knowledge discovery which in turn impacts the advancements in scientific discovery such as new materials design. One of the main branch of scientific discovery is materials science and engineering, which focuses on how to design new materials as well as optimize the properties of existing engineering materials; such optimizations include increasing efficiency by reducing the material utilization and cost, and improving safety. Materials design is systematically carried out using the so called *materials paradigm*, which is the study of the inter-relationship between processing, structure, property, and performance of a material system [**8, 9**]. This paradigm is used to advance the understanding in a variety of research fields such as nanotechnology, biomaterials and metallurgy. Materials characterization, a collection of experimental techniques to determine structure and chemistry across multiple length and time scales, lies at the heart of the materials paradigm and is routinely used to study these inter-relationships. A few important techniques include optical and electron microscopy [**10**], x-ray diffraction [**11**] and spectroscopy [**12**]. The interpretation of such large scale multi-modal data requires development of accurate physics-based predictive machine learning models for direct comparison with experiments.

Significant advancement has been made in assisting the process of scientific discovery using machine learning (ML) techniques [**13, 14, 15, 16**]. Some machine learning (ML) approaches used are clustering [**14**], linear regression [**17**], polynomial regression [**17**], decision trees [**18**], support vector regression (SVR) [**15**], kernel ridge regression (KRR) [**16**] and support vector machines (SVM) [**18, 17**]. Most current ML approaches require incorporation of domain knowledge in some form in the model inputs. They have shallow model

architectures that limits their learning capability. Most applications of neural networks in scientific domains have been limited to models with small number of processing layers with limited learning and prediction capabilities [**19, 20, 21, 22, 23, 24**]. Deep learning [**3**] can offer an alternative route for accelerating the creation of models by overcoming these challenges in building predictive models for scientific datasets.

Although deep learning has enjoyed great success in the fields of computer science, its application in scientific fields has been very limited. Since scientific datasets are collected from expensive and time-consuming scientific experiments and computations, they are many challenges associated with them due to the scarcity and complex nature. Some these challenges includes complexity due to presence of background noise in experiments, heterogeneity in data due to use of multiple instruments with different output formats for same experiment, and heterogeneity due to presence of multiple data types in input. Developing machine learning models to handle these challenges requires novel DNN architectures that can automatically learn the underlying science behind the scientific phenomena using artificial intelligence (deep learning). It includes learning from heterogeneous inputs collected from different instruments, predicting multiple outputs using a single deep neural network architecture, optimizing for domain specific loss functions and incorporation of domain knowledge and human intuition in the model inputs to develop accurate physics-based predictive models. This thesis presents different deep learning methodologies to overcome the challenges associated in building predictive machine learning models by leveraging deep learning. In particular, this thesis presents the design, implementation and evaluations of DNN models that can automatically handle

different challenges associated with scientific datasets and learn the underlying scientific phenomena for the advancement of scientific knowledge discovery.

1.1. Challenges

There are many challenges in building a data-driven machine learning based predictive modeling for the advancement of scientific knowledge discovery. Scientists generally rely on experiments and simulations to understand scientific phenomena, which has lead to collection of datasets over time that can used for building data-driven prediction models using machine learning techniques. The scientific datasets are complex and high dimensional, and using them to build predictive models for accelerating scientific discovery involves careful consideration and understanding of the nature of scientific phenomena governing the experiments and data collection in collaboration with domain scientists. Here, I present some of the challenges involved in the process of developing deep learning methodologies for scientific knowledge discovery.

1.1.1. Limited Availability

Scientific datasets are either collected from experiments or simulation. While experiments are expensive and time consuming, simulation datasets are much more simpler and still time consuming even on the modern era supercomputers. Hence, the size of scientific datasets are small compared to typical datasets used for deep learning. For instance, the in field of materials science and engineering, the size of experimental datasets are in the range of 10^2 to 10^3, while the size of computational datasets are in the range of 10^4 to

10^5. This has discouraged many scientist from applying deep learning methodologies to build predictive models for their domains.

1.1.2. Feature Engineering for the Incorporation of Domain Knowledge in Model Inputs

The current methodologies for predictive modeling based on machine learning involves incorporation of domain knowledge and human intuition in the model inputs by feature engineering [**14, 13**]. While this approach may have helped in improving robustness of models, there is no clear consensus on which attributes are important and how much domain knowledge is sufficient for a given prediction problem.

1.1.3. Presence of Irregular Background Noise and Heterogeneity in Nature

Experimental datasets are collected from scientific instruments such as Linear Accelerators and Electron Diffractometers under different experimental settings and environments. Use of different instruments and different experimental settings result in heterogeneous samples. While experimental settings can be controlled by scientists, there are several environmental factors during experiment which impacts the instrument and hence, the experimental observations, that are beyond scientist's control. This leads to irregular background noise in the collected samples that are hard to process manually or using existing techniques. Also, the datasets are themselves composed of multiple types such as image, composition, experimental parameters, these lead to another type of heterogeneity in the model inputs for predictive modeling.

1.1.4. Optimizing for Domain-specific Loss Function

In computer science fields like computer vision and natural language processing, there are simple and well-defined loss functions that are used for optimization during training a deep neural network or machine learning model. However, in scientific fields such as chemistry, physics and materials science, the loss function one wants to optimize for can be really complicated and non differential. Developing predictive modeling for such tasks involves optimizing for loss functions that incorporate the domain specific loss function which can be more complicated than the deep neural network architecture itself.

1.2. Problem Statement

The problem statement for this thesis proposal is ***"How to develop data-driven machine learning based predictive models using deep neural network architectures that can automatically handle different challenges associated with scientific datasets and learn the underlying scientific phenomena by leveraging artificial intelligence (deep learning) for the advancement of the overall process of scientific knowledge discovery."***

1.3. Thesis Organization

In this thesis, I present how to develop data-driven machine learning based predictive models for advancing the overall process of scientific knowledge discovery. This thesis builds by first discussing about background and related works on deep learning, deep neural networks and machine learning for scientific discovery in Chapter 2. Next, the thesis presents different methodologies for designing deep neural network architectures

for handling the complexity and challenges associated with scientific datasets in building predictive models for scientific applications.

Chapter 3 presents the methodology for designing a deep neural network that can directly learn the chemistry of materials from their raw elemental fractions without any need for feature engineering for the incorporation of domain knowledge in the model inputs. Chapter 4 presents a general deep neural network architecture framework for building predictive models for vector inputs that can be composed attributes derived from materials crystal structure and/or composition using domain knowledge. Chapter 5 builds on Chapter 3 by leveraging together computational and experimental datasets to build more robust predictive models with high accuracy such that prediction accuracy is comparable to that of computational datasets and more closer to the true experimental observations.

Chapter 6 presents methodology for designing a convolutional neural network that can optimize for a hybrid loss function of mean absolute error and a domain specific loss function- "disorientation" to accurately predict crystal orientations from electron back-scatter diffraction patterns. In Chapter 7, we present a methodology for designing a convolutional neural network that can directly predict the phases of material alloy sample from its raw 2D X-ray diffraction by automatically handling the background noise using specially designed peak area detection network and slope filters for background removal.

Finally, the thesis concludes by discussing future directions in developing deep learning based methodologies for scientific knowledge discovery in Chapter 8.

CHAPTER 2

Background

2.1. Deep Learning

Deep learning is a new area of machine learning field of learning representations of data. Deep learning allows deep neural network (DNN) models composed of multiple processing layers (deep neural network architecture) to learn representations of data with multiple levels of abstraction [**3**]. They are exceptionally effective in learning patterns and helped in achieving the objective of pushing it closer to one of its original goals of artificial intelligence [**2**]. Deep learning models learns to understand the information present in big training datasets and learns to respond in useful ways. Deep learning has gained significant attention in the field of computer science with breakthrough results in computer vision, speech recognition and text processing [**4, 5, 6, 7**]. Although the basic algorithmic approach in deep learning remains the same, such progress has been possible mainly due to availability of large datasets and increase in computational power for training. Due to their flexible DNN architecture with large parameter set, deep learning models can automatically capture the high level non-linear mappings between input features and output values through multi-layered feature abstraction.

2.2. Deep Neural Networks

Deep neural networks (DNN) are model architectures inspired by human brain; they are composed of multiple processing layers that learn the representations present in the

input data with multiple levels of abstraction. The input data is fed into the first layer; the deep neural network learns information present in the input of a layer by mapping it to higher-level more abstract features; the final layer predicts the output. Depending on the type of input, the DNN can be composed of several types of layers such as convolutional layers and max pooling, recurrent neural networks, and fully connected layers. The DNN architecture search involves exploring large search space with models of different depths and individual layer breadths. Here are some of the terminologies used in deep learning:

- **Feedforward Neural Networks** These are networks composed of neuron layers that are fully connected with each other; each neuron in one layer is fully connected to each neuron in another. These networks are generally used for vector inputs. Fully connected layers are used at the end of all types of deep neural networks to make the classification or regression outputs.
- **Convolutional Networks** Convolutional neural networks are class of feedforward neural networks that have become state-of-the-art method for learning from input images. Convolutional neural networks are class of feedforward neural networks that composed of convolutional layers, generally followed by pooling layers. The hierarchy of convolutional layers work in similar was as our visual cortex; the first few layers captures the edges, the next few layers captures the shapes and the final layers capture the face or objects. The convolution operation involved are computationally very expensive; each output features is a convolution of the feature vector on a local region in the input. They are very effective for image inputs and are the state-of-the-art models in computer vision.

- **Dropout** Dropout is a concept of randomly dropping some features present in the layer input; generally a fraction of the inputs are made zero. This serves two purposes. First, the dropout helps the model from overfitting [**25**]. Overfitting is the problem of model learning the training data very well but performing very poorly on the test data. Second, the dropout makes the model as powerful as an ensemble of similar smaller models; the smaller models can be thought of as being formed by dropping the neurons, and hence, their connections, from the original network by making them zeros.
- **Pooling** Pooling is a technique of reducing the number of features to reduce the amount of computations involved in next layers. Pooling are generally used in all convolutional neural networks, they are inspired by human brain. Generally, two kinds of pooling are used- max-pooling and average-pooling. As the name suggest, max pooling involves taking the max out of a region of features while average pooling takes the average of all the features in a region.
- **Learning rate** Learning rate determines the rate of adjusting the model parameters using the gradients during back propagation. During stochastic gradient descent, the gradients are multiplied by the learning rate before applying them to the model parameters. Higher values of learning rates results in high oscillations while lower values of learning rates can result in longer training times.

2.3. Stochastic Gradient Descent

Stochastic gradient descent is the optimization algorithm to train deep learning models [**26**]. It is different from the batch gradient descent in the sense that batch gradient

descent involves going through whole training dataset before performing back propagation while stochastic gradient descent chooses a random sample of training data during each pass to optimize the model parameters using back propagation. It is consists of two parts: forward pass and back propagation. During forward pass, the model computes the output for the given input data. During back propagation, first the model computes the loss of the predicted outputs with respect to the true outputs. Next, the gradients of the loss are computed with respect to each model parameters and the parameters are updated using them.

2.4. Machine learning for scientific discovery

There have been many initiatives to computationally assist scientific discovery using machine learning techniques [**13, 14, 15, 16, 17, 18**]. Most works involve developing prediction system for properties of either organic and inorganic materials for materials search and design. Ward et al. [**14**] used random forest (RF) and clustering for discovering new photo-voltaic materials and metallic glass alloys. Agrawal et al. [**17**] used linear, polynomial and support vector machine (SVM) regression for the prediction of fatigue strength of steels from their composition and processing parameters. Liu et al. [**18**] used decision trees and SVMs for reducing the search space using feature selection for microstructure optimization. Xue et al. [**15**] used support vector regression (SVR) to infer the thermal hysteresis of NiTi-based shape memory alloys to accelerate search for materials with low thermal hysteresis properties. Kernel ridge regression (KRR) is used by Xue et al. [**15**] to transform the input fingerprint of a material into a higher dimensional space

to establish linear relation between the transformed fingerprint and the property of interest. Meredig, Agrawal et.al [**13**] used a combination of rotation forest based ensemble approach [**27**] and heuristic based modeling, using around 15,000 DFT calculations on various materials in the Inorganic Crystal Structure Database (ICSD), to predict formation energy of compounds. They obtained a mean absolute error (MAE) of 0.16 eV/atom and 0.12 eV/atom using their machine learning and heuristic models respectively. Agrawal et al. [**28**] used the same model, but trained on around 100,000 DFT computed formation energy values, and obtained an MAE of 0.1343 eV/atom. Their model was limited to binary and ternary compounds. Recently, Ward et al. [**28**] presented similar approach based on machine learning approach trained on 228,676 compounds from OQMD. They used 145 descriptive attributes as inputs to include domain knowledge, obtaining an MAE of 0.0882 eV/atom for formation energy prediction. All these current data-driven approaches depend on inclusion of domain knowledge in inputs or heuristic modeling.

2.5. Deep learning for scientific discovery

Recently, advancements in deep learning have opened up a new era of technical advancements in computer science. Deep neural networks (DNN) have achieved state-of-the-art results in computer vision [**4, 29, 30**], speech recognition [**31**] and text processing [**32**]. Neural networks have also been used in the field of material science [**20, 21, 22, 23, 24**]. In the Harvard Energy Clean Project, Pyzer et al. [**20**] used a multi-layered perceptron (MLP) of just 3 layers for predicting power conversion efficiency of organic photo-voltaic materials. Pyzer et al. [**21**] used a bayesian approach to calibration of quantum chemical calculations to experiment, implemented as a Gaussian process with a prior based

upon relevant experimental observations. Montavon et al. [**22**] trained a 4 layered MLP on a database of around 7,000 organic compounds to predict multiple electronic ground-state and excited-state properties. Some other application of neural networks to materials science works on spectroscopy classification and structural identification [**23**] and characterizing constitutive relationship for alloys [**24**]. Most of the networks used in such works have relatively shallow networks and used small dataset size.

CHAPTER 3

ElemNet: Deep Learning the Chemistry of Materials From Only Elemental Composition

3.1. Introduction

Materials scientists, condensed matter physicists and solid-state chemists rely on data generated by experiments and simulation-based models to discover new materials and understand their characteristics. For the major part of the history of materials science, experimental observations have been the primary means to know the various chemical and physical properties of materials [**33, 34, 35, 36, 37, 38**]. Nevertheless, experimentation of all possible combinations of material composition and crystal structures is not feasible as that would be very expensive and time-consuming, and the composition space is practically infinite. Computational methods, such as Density Functional Theory (DFT) [**39**], offer a less expensive means to predict many material properties and processes on the atomic level [**40**]. DFT calculations have offered opportunities for large-scale data collection such as the Open Quantum Materials Database (OQMD) [**41, 42**], the Automatic Flow of Materials Discovery Library (AFLOWLIB) [**43**], the Materials Project [**44**], and the Novel Materials Discovery (NoMaD) [**45**] ; they contain DFT computed properties of $\sim 10^4 - 10^6$ of experimentally-observed and hypothetical materials. In the past few decades, such materials datasets have led to the new data-driven paradigm of materials informatics [**9, 46, 47, 48, 49, 50**]. The availability of such large data resources has

spurred the interest of researchers in applying advanced data-driven based machine learning (ML) techniques for accelerated discovery and design of new materials with select engineering properties [**51, 52, 17, 13, 53, 54, 55, 18, 15, 56, 57, 58, 14, 59, 60, 61, 62, 50, 63, 64, 65**].

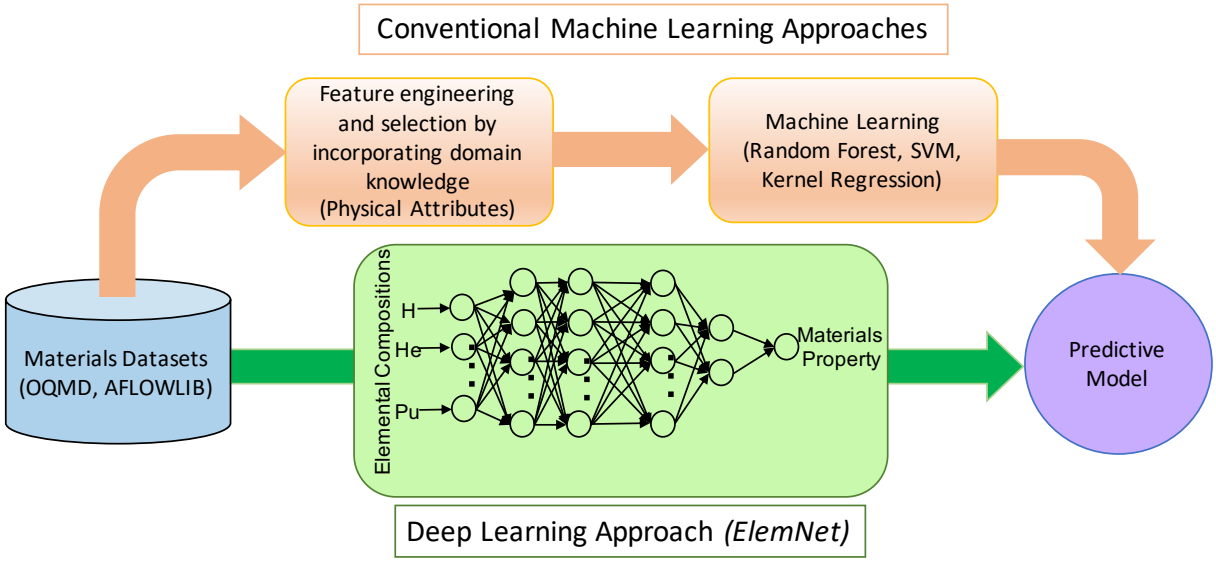


Figure 3.1. Comparison of deep learning approach with conventional ML approach for prediction of materials properties. The conventional ML approach for predictive modeling of materials properties involve representing the material composition in the model input format, manual feature engineering and selection by incorporating the required domain knowledge and human intuition by computing the important chemical and physical attributes of the constituent elements, and applying ML techniques to construct the predictive models. Our deep learning based predictive approach directly learns to predict properties of materials such as the formation enthalpy from their elemental compositions with better accuracy and speed than conventional ML approaches.

Conventionally, constructing an effective ML model requires first developing a suitable representation for the input data. As has been discussed in several recent works, the best representations are those that encode knowledge about the physics of the underlying

problem. To that end, there have been many distinct approaches for encoding information regarding the composition[**13, 14**] or crystal structure[**63, 66, 60, 67**] of a material. For instance, Ward *et al.* developed a set of attributes based on the composition of a material that can be useful for problems including predicting formation enthalpies of crystalline materials and glass-forming ability of metal alloys. [**14**] Ghiringhelli et al. [**68**] analyzed the tendency for materials to form different crystal structures using thousands of descriptors. Developing ML models based on intuitive representations is evidently successful given the large number and growing rate of ML models constructed over the past several years using this approach [**50, 49, 69**]. However, the prediction accuracy for these problems is limited by our ability to feature engineer the materials representation to incorporate all the domain knowledge required to make correct predictions. Given that one of the major use cases of ML is for problems where the physics driving behavior is yet to be understood,[**50**] this limit could be a significant impediment to the use of ML. A better approach would be to construct a system that can automatically learn the optimal representation.

Deep learning [**3**] offers an alternative route for accelerating the creation of predictive models by reducing the need for designing physically-relevant features. It makes use of deep neural network (DNN) models composed of multiple processing layers (network architecture) to learn representations of data with multiple levels of abstraction [**3**]. DNN models can learn from input representations such as numerical encoding of texts, color pixels of images, etc., without any need to first compute application-specific descriptors [**70, 71, 72**] thereby eliminating the manual step of feature engineering and representation required in conventional ML. Due to this powerful advantage, deep learning has gained significant attention in the field of computer science with breakthrough results in

computer vision [**4, 73**], speech recognition [**74, 75**] and text processing [**76**]. Although deep learning models have enjoyed great success in the above applications, implementation of deep learning systems in materials science is in its early stages - mainly due to scarcity of big training datasets. Nevertheless, they have already shown some promise in materials science. Convolutional Neural Networks (CNN) have been used for building models from microstructural data and improving characterization methods,[**77, 78, 79**] and deep neural networks have been shown to be useful for predicting properties of crystal structures and molecules [**80, 81, 82**].

Our goal in this work is to leverage the power and elegance of deep learning to directly learn the properties of materials from their elemental compositions, eliminating the limitations of current ML approaches that require manual feature engineering. We design a deep neural network model that we refer to as *ElemNet*, which takes only the elemental compositions as inputs and leverages artificial intelligence to automatically capture the essential chemistry to predict materials properties. Here, we evaluate the effectiveness of this approach by revisiting a commonly-studied challenge in materials informatics: predicting whether a crystal structure will be stable given its composition. [**13, 14, 83, 84, 85**] We adopt the approach of Meredig et al. [**13**] and Ward et al. [**14**], and train *ElemNet* on the DFT-computed formation enthalpies (the energy of forming a compound from its constituent elements) of $275,759$ compounds with unique elemental compositions from the OQMD. As demonstrated by Meredig et al., the formation energy predicted using this model can be compared to the formation energies of existing compounds in order to identify compositions where there is likely a yet-undiscovered compounds. In contrast to these previous papers which relied on physics-informed features to train a model, we

approach this material prediction problem without using any domain knowledge about materials stability and rely purely on representation learning.

We find that *ElemNet* is able to automatically learn the chemical interactions and similarities between different elements which allows it to even predict the phase diagrams of chemical systems absent from the training dataset more accurately than conventional ML models based on physical attributes leveraging domain knowledge. We compared the performance of our deep learning model to a recent conventional ML approach that used engineered features [**14**] on the OQMD; using a ten-fold cross validation, we find that *ElemNet* outperforms the conventional ML models both in terms of speed and accuracy for all training data size exceeding 4000 compounds. As deep learning frameworks support execution on Graphics Processing Units (GPUs), *ElemNet* can make predictions at two orders of magnitude faster than the physical attributes based ML models running on CPUs. The improved accuracy and higher speed of the model can allow us to perform combinatorial screening for new material candidates. As a case study, we perform a combinatorial screening in a huge composition space of around half a billion compounds, and find that our model successfully identifies compounds not in our training set. We believe *ElemNet* opens a new direction for more robust and faster identification of promising materials and thus, can play a crucial role in accelerating the materials discovery process.

3.2. Methods

3.2.1. Data Cleaning

The data is composed of fixed size vectors containing raw elemental compositions in the compound as input and formation enthalpy in eV/atom as output labels. The input

vector has non-zero values for all the elements present in the compound and zero values for others. As most compounds are composed of fewer than five elements, the input vector is very sparse. The composition ratio is normalized so that the elements of the input vector sum to one. Two stages of data cleaning are performed to remove single element compounds and outliers. First, all single-element materials are removed as their formation energy is zero, by definition. Next, data entries with formation energy values outside of $\pm 5\sigma$ (σ is the standard deviation in the training data) are removed. Such outliers are discarded to prevent calculation errors undetected by strict value bounds. Further, the elements (attributes) that do not appear in the cleaned dataset are removed from the input attribute set. Out of 118 elements in the periodic table, 86 elements are present in our dataset. Our dataset contains 256,622 compounds after cleaning, out of which there are 16,339 binary compounds, 208,824 ternary compounds, and 31,459 compounds with between 4 and 7 constituent elements. The dataset (after cleaning) is randomly split into training and test sets using a ten-fold cross validation; each training set and test set contain $230,960$ compounds and $25,662$ compounds with unique compositions and their minimum formation enthalpies.

3.2.2. Model Architecture Search

Our deep learning model is based on a deep neural network (DNN) composed of multiple consecutive layers of neurons. To find the best model for the formation enthalpy prediction, we carry out an extensive search for the best DNN model architecture as well as in the hyper-parameters space. We performed a systematic search through a large neural network architecture space, starting from a two-layered architecture and incrementally

Table 3.1. *ElemNet* Architecture. Considering the Input as the 0th layer, types and positions of different types of fully connected and dropouts are shown below. Dropout layers are used to prevent overfitting and they are not counted as a separate layer. We used ReLU as the activation function.

Layer Types	**No. of units**	**Activation**	**Layer Positions**
Fully-connected Layer	1024	ReLU	First to 4th
Drop-out (0.8)	1024		After 4th
Fully-connected Layer	512	ReLU	5th to 7th
Drop-out (0.9)	512		After 7th
Fully-connected Layer	256	ReLU	8th to 10th
Drop-out (0.7)	256		After 10th
Fully-connected Layer	128	ReLU	11th to 13th
Drop-out (0.8)	128		After 13th
Fully-connected Layer	64	ReLU	14th to 15th
Fully-connected Layer	32	ReLU	16th
Fully-connected Layer	1	Linear	17th

increasing the depth to improve the learning capacity of our model until a saturation point is reached. We explored with different combinations of the number of neurons units per layer. A dropout [**86**] layer was added whenever the number of neurons between consecutive layers changed to avoid overfitting [**87**]. The test error started oscillating within small limits beyond 17-layered architecture. The architecture search was continued up to 24 layers DNN model where the test error remained same as the 17 layered network. We believe that the deep learning model already learned the necessary features it could find in the training dataset at this point, as increasing the depth did not improve the model performance any further. We also experimented with different types of activation functions, and ReLU (rectified linear unit) [**88**] was observed to perform the best.

3.2.3. Model Hyperparameter Search

We performed an extensive search to tune the model hyperparameters as recommended by Bengio et.al [**89**]. We started with a small range of values for each hyperparameter based on our intuition, rather than performing a grid search that would have been infeasible due to time and computational resource constraints. The hyperparameter search space comprised of different candidate values of momentum [**90**], learning rate [**91**], optimization algorithms, dropouts [**86**] and other hyperparameters. Learning rate was one of the most important DNN hyperparameters. Learning rates values from 0.1 to $1e^{-6}$ were tried, decreasing by a factor of 10. Dropouts [**86**] are known to have a great impact on decreasing the overfitting [**87**] of the model to training set [**25**]. A search for dropout values ranging from 0.5 to 0.9 (dropout value denotes the inputs retained, such as 0.7 means 30% input values are dropped and rest 70% are used) was carried for each of the four dropout layers used in our DNN models. Increasing dropout helped in improving prediction accuracy as it decreased overfitting the of model to the training dataset. For momentum, we experimented with values in the $[0.9, 0.95, 0.99]$; momentum value of 0.9 performed the best. Stochastic gradient descent (SGD) performed best among all optimization algorithms in our study. Similarly, we experimented with a range of values for other hyperparameters.

3.2.4. Machine Learning Parameter Search

We performed a thorough grid search for parameters of all ML models used in this study. For instance, we experimented Random Forest regression with a number of different combinations of estimators in $[50, 100, 150, 200]$, minimum samples splittings in $[5, 10, 15, 20]$, maximum features in $[0.25, 0.33]$ and maximum depths in $[10, 25]$.

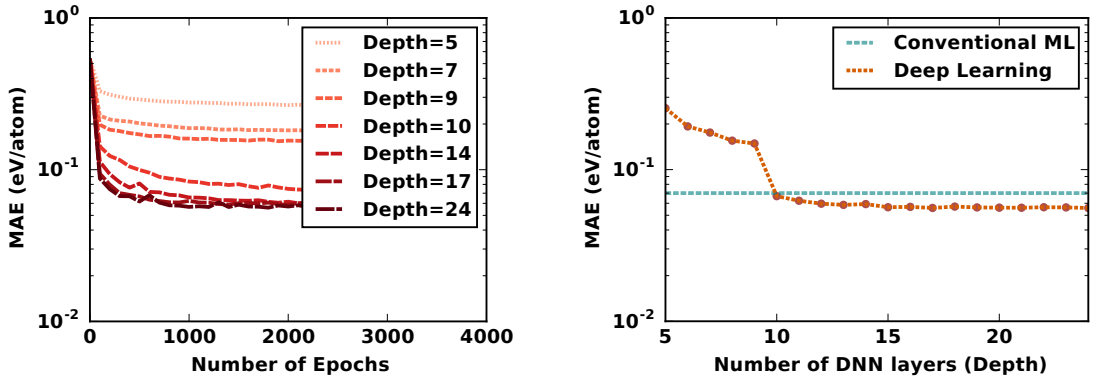
3.2.5. Experimental Settings and Tools Used

The deep learning models are implemented using Python 2.7, Theano [**92**] and TensorFlow [**93**] framework. For other ML models, implementations available in Scikit-learn [**94**] are used. All the models were trained and tested using NVIDIA DIGITS DevBox.

3.3. Experimental Results

3.3.1. Dataset

We used the OQMD [**42, 95**] for training and testing our proposed deep learning model. OQMD is an extensive high-throughput DFT database, consisting of DFT computed crystallographic parameters and formation enthalpies of experimentally observed compounds taken from the Inorganic Crystal Structure Database (ICSD) [**96**] and hypothetical structures created by decorating prototype structures from the ICSD with different compositions. OQMD is continually growing and, at the time of writing, contains 506,115 compounds at 275,778 unique compositions. We train our predictive models on the lowest formation enthalpy at each composition becauses they represent the most stable compounds, which causes our model to predict the energy of the ground-state structure given composition.



(a) Test Error

(b) Impact of depth

Figure 3.2. Performance of deep learning models of different depths in model architecture. The models are trained and tested on the lowest DFT-computed formation enthalpy of 256, 622 compounds. Here, we present the impact of depth of architecture for one sample split from our ten-fold cross validation. (a) shows the mean absolute error (MAE) on the test dataset of 25, 662 compounds with unique compositions at different epochs for one split from the cross validation. The DNN models keep learning new features from the training dataset with the increase in the number of layers up to 17 layers, after which they begin to slowly overfit to the training data. (b) shows the MAE for different depths of deep learning model architectures and also illustrates mean absolute error of the best performing conventional ML model trained using physical attributes computed on the same training and test sets. The deep learning model start outperforming the best performing conventional ML model with an architecture depth of 10 layers, achieving the best performance at 17 layers, we refer to the best performing DNN model as *ElemNet*. The detailed architecture for *ElemNet* is available in the Method section.

3.3.2. Design

We perform an extensive search for deep neural network (DNN) architectures and hyper-parameters (details in Method section). Figure 3.2 illustrates the improvement in DNN learning capacity with the increase in the number of layers for different training epochs. From the test error plot, it is obvious that the learning capacity of DNN models improves

with the increase in the depth of the network. The errors observed on training and test sets decrease rapidly up to 17 layers. After a certain depth, the improvement in learning of features by the DNN models starts plateauing. This plateauing effect can be a result of the features reaching the maximal extent of learning possible via our models. Figure 3.2(b) illustrates the overall comparison of the test errors of DNN models with different architecture depths. The best predictive model is a 17-layered DNN architecture (excluding four dropout layers) with tuned hyperparameters; we refer to this model as *ElemNet*. The model with 17 layers has the best accuracy of 0.050 ± 0.0007 eV/atom in 10-fold cross-validation, which is only 9% of the mean absolute deviation in the set (0.550 eV/atom). The detailed architecture of *ElemNet* is provided in the Method section. The results illustrate that deep neural networks can effectively learn the optimal feature representation from materials composition without any need for manual feature engineering using domain knowledge.

3.3.3. Deep Learning vs Physical-attributes-based Conventional ML Approach

Table 3.2. Benchmarking our deep learning model – *ElemNet* – against conventional machine learning approaches. We trained several conventional ML models such as Linear Regression, SGDRegression, ElasticNet, AdaBoost, Ridge, RBFSVM, DecisionTree, ExtraTrees, Bagging and Random Forest. Out of them, Random Forest performed the best with and without using physical attributes. Here, we show the results from our deep learning model and the best conventional ML model-Random Forest, in our study for both types of model inputs (with and without physical attributes), along with the type of input used, mean absolute error (MAE) on the test set, training time on the training set, and prediction time on the entire test set (25,662 entries). All the models are trained and tested using a ten-fold cross validation. All timings are on a single (logical) CPU core of an NVIDIA DIGITS DevBox with a Core i7-5930K 6 Core 3.5GHz desktop processor with 64GB DDR4 RAM and 4 TITAN X GPUs with 12GB of memory per GPU, except the deep learning models.

Model	**Input Type**	**MAE (eV/atom)**	**Training time (hour)**	**Prediction time (sec)**
RandomForest	Physical Attributes	0.071 ± 0.0006	1.5	14.80
RandomForest	Elemental Compositions	0.157 ± 0.0012	1.5	2.87
ElemNet	**Elemental Compositions**	$\mathbf{0.050 \pm 0.0007}$	**7 (GPU)**	**9.28 (CPU) & 0.08 (GPU)**

Our next step is to compare *ElemNet* against the current ML approach: conventional ML models that rely on the computation of physical attributes. We chose to compare *ElemNet* against the general-purpose approach of Ward *et al.*, which uses 145 physical attributes that fall into four different categories - stoichiometric attributes, elemental property statistics, electronic structure attributes and ionic compound attributes.[**14**] As shown in Table 3.2 and Figure 5.6b, the models created using conventional ML are better with the physical attributes than with only the element fractions using the same training and test sets. We also find that deep learning surpasses all the conventional ML models – whether with physical attributes or not – in accuracy by at least 30%. This improvement in accuracy is quite fascinating as it is achieved without encoding any domain knowledge into the inputs of the function – a finding that shows carefully-developed features are not critical for success in ML if sufficient training data is available. While adding more domain knowledge is certainly expected to improve a ML model, for some problems, it may not be straightforward or even feasible to come up with appropriate physical attributes due to lack of understanding of the underlying phenomena. It is thus quite encouraging to find that this step of incorporating domain knowledge might not always be necessary to achieve excellent performance.

3.3.4. Impact of Training Data Size

Deep learning models have enjoyed great success in many applications, and typically these were applications where the training data is relatively abundant [**3**]. The perceived need for large datasets has discouraged many researchers in the scientific community having access to only small datasets from leveraging deep learning. To understand what

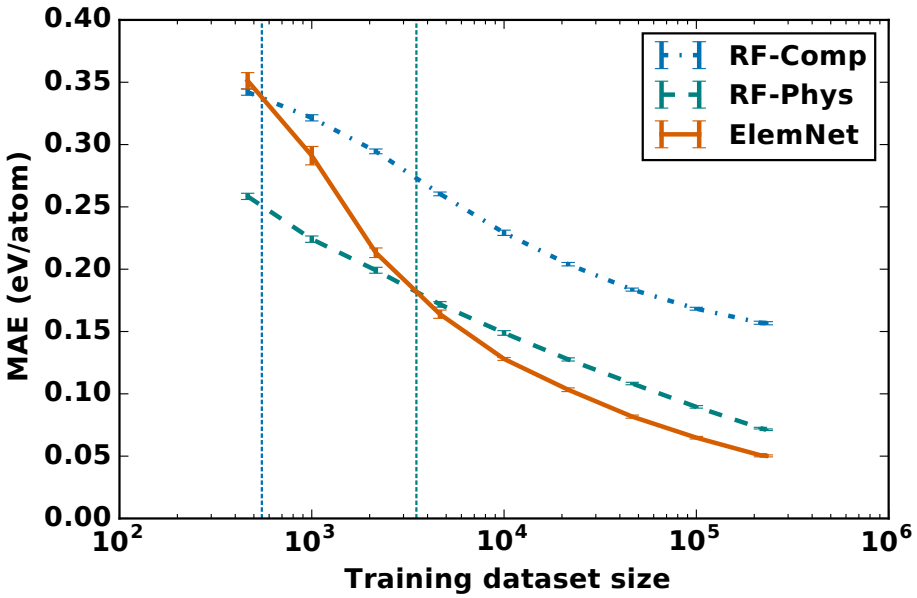


Figure 3.3. Impact of training dataset size on the prediction accuracy of *ElemNet* (DNN model) using elemental compositions only and the best conventional ML model, Random Forest, with either raw elemental compositions (RF-Comp) and physical attributes (RF-Phys). The training and test sets are created during the ten-fold cross validation from the OQMD; different random subsets of the training set with sizes ranging from 464 to $230,960$ are created using a logarithmic spacing for this analysis. Training dataset size has more impact on ElemNet (deep learning model) compared to Random Forest models, but *ElemNet* performs better than Random Forest for all size greater than $4k$.

the necessary dataset size is for deep learning to be effective for our application, we compared the effect of training dataset size on the accuracy of deep learning model and our best performing conventional ML model- Random Forest, with either the raw elemental compositions or the physical attributes as model inputs. We used different random subsets of the training dataset from the ten-fold cross validation with sizes ranging from 464 to 230,960 using a logarithmic spacing; the test set always contains 25,662 compounds. We used the same ten-fold training and test datasets for both *ElemNet* and Random Forest

models (both with and without physical attributes) to ensure a fair comparison between the various approaches.

As illustrated in Figure 3.3, our deep learning model achieves better accuracy than the best conventional ML approach based on physical attributes (manual feature engineering by incorporating domain knowledge) with only 2% of our training set. In general, *ElemNet* exhibits higher impact of training dataset size compared to the Random Forest models. The error curve has a steeper reduction in test error with the increase in training dataset size in the DNN model compared to Random Forest models. However, the important observation is that deep learning performs better than the Random Forest models even when the training dataset size is in $\sim 10^3 - 10^4$. It surpasses the accuracy of the Random Forest model with raw elemental compositions as input even at a training dataset size of 550, and the Random Forest model with physical attributes for all training dataset sizes exceeding 3500. Our results demonstrate that deep learning models can not only benefit more with an increase in dataset size compared to traditional ML models, but also deep learning can outperform them even at relatively smaller dataset size of around $4k$ samples. What the small training set requirement implies is that deep learning models such as *ElemNet* may be useful for building more accurate predictive models than conventional ML based models for many materials science datasets that are much smaller than the OQMD.

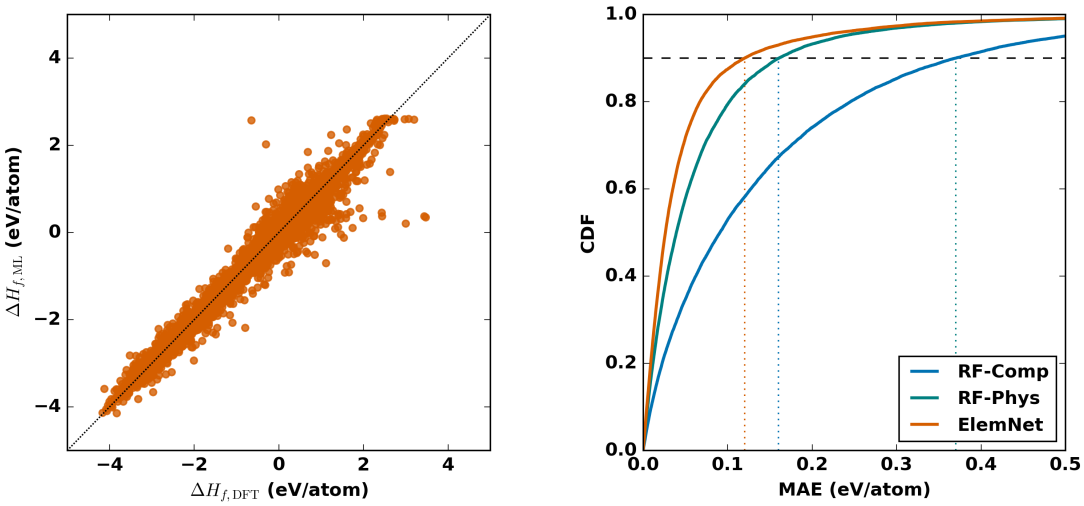
3.3.5. Prediction Time Analysis

ElemNet predicts the formation enthalpy with better accuracy and speed. Table 3.2 shows the time taken by different predictive models to train on the training set and predict the

formation enthalpy for the entire test set. All deep learning models are trained using GPUs and both the prediction time of deep learning using a single (logical) core of CPU as well as a GPU core are reported in Table 3.2. The prediction time of deep learning model is lower than the time required by the best conventional ML approach - Random Forest. Since deep neural networks mainly involve matrix multiplications, they are highly parallelizable compared to conventional ML methods such as Random Forest; hence, deep learning frameworks supports execution on GPUs. While running on GPUs, *ElemNet* can predict with two orders of magnitude faster than the current conventional ML models in practice. Our results illustrates that the proposed deep learning approach can predict with better accuracy as well as speed. It can, therefore, play a crucial role in accelerating the exploration of new composition spaces for materials discovery.

3.3.6. Assessing Accuracy of Model

Our deep learning model achieves strong performance across a broad range of materials. As shown in Figure 5.6b, *ElemNet* predicts the formation enthalpy of compounds in one of our test sets with a mean absolute error (MAE) of 0.055 eV/atom; predicting the formation enthalpy of 90% of compounds in our test set with an error of less than 0.120 eV/atom. To better understand how our model could be best used, we studied for which kinds of materials it performs the least accurately. The materials where our model has the largest errors typically have large, positive formation enthalpies (see the outliers in Figure 5.6a), which suggests our model performs the worst at trying to predict the formation enthalpy of highly unstable compounds. Only 59% of our test set has a positive formation enthalpy yet 67% of the entries with the largest errors (99% percentile



(a) Prediction vs DFT

(b) CDF of Prediction Errors

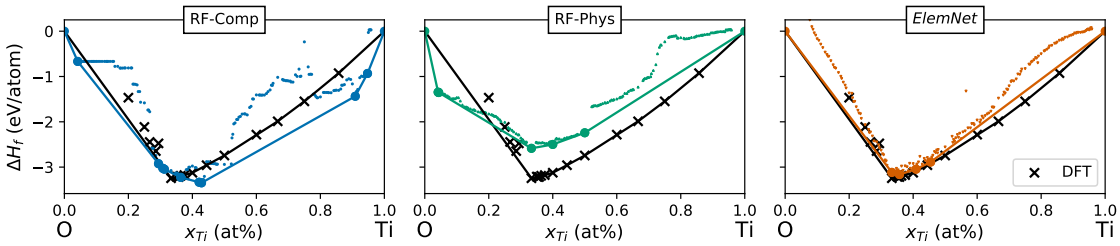
Figure 3.4. Error analysis of the predictions using *ElemNet* of a test set containing 25,662 compounds from our ten-fold cross validation. The left side shows that the predicted values are very close to the DFT-computed values. The right side illustrates the cumulative distribution function (CDF) of the prediction errors for *ElemNet* and Random Forest (the best performing conventional ML model) with elemental fractions (RF-Comp) and physical attributes(RF-Phys). Our error analysis demonstrates that the deep learning performs very well, achieving an MAE of 0.050 ± 0.000 eV/atom; predicting with an absolute error of less than 0.120 eV/atom for 90% of the compounds in our test set (right).

of absolute error) have positive formation enthalpies. These unstable compounds are arguably the least physically important part of the dataset, and therefore the inability of *ElemNet* to accurate predict these energies is not a significant drawback.

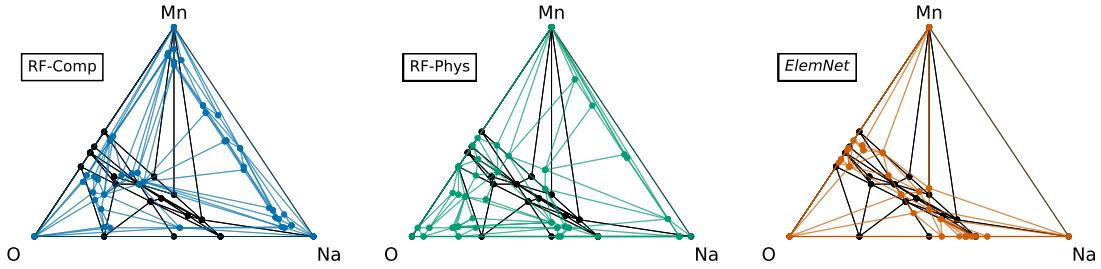
We also studied how *ElemNet* performs on different chemical classes of materials. The 25 entries with the highest errors include intermetallics (e.g., Cr_2Ni_3), metal/nonmetal compounds (e.g., Ho_2C, Sm_3AlN), and compounds with only non-metallic elements (e.g., BCl), so there does not seem to be a systematic problem with modeling a particular

material class. To further understand if certain chemistries have larger errors, we first grouped entries in the test set by whether they contained certain elements and then computed the Spearman rank correlation coefficient for each group. The elements that exhibit the lowest correlation coefficients are Pu (0.66), Np (0.86), C (0.87), and N (0.87). The Pu and Np compounds are likely to have the lowest performance because they have the fewest number of training points among metallic elements. C and N both appear much less frequently in our training set than any metallic element because they are not included in the combinatorial searches for intermetallics, whose results constitute the bulk of the OQMD. Among these elements which appear less often in the OQMD (Br, C, Cl, F, H, I, N, P, S, Se, Xe), C and N have the highest number of compounds with positive formation enthalpies in the test set. Consequently, we conclude the poor performance on C- and N-containing compounds is also a result of the poor performance of the model on unstable material and not because of a systematic issue with modeling certain elements.

The types of compounds where *ElemNet* performs best also line up with our expectations. The elements with the highest correlation coefficients are lanthanides and alkali metal compounds. Lanthanides display a strong degree of chemical similarity (e.g., all form trivalent cations), and so we would expect the properties of lanthanide compounds to be relatively easy to predict if our model can recognize the similarity between these elements. Additionally, alkali metals are most often observed in single oxidation state (1+), which makes their chemistry somewhat simpler than most transition metals. In terms of the nonmetals, our model has the best performance on Se-, F-, and Cl-containing compounds, which have the highest fraction of compounds with negative formation enthalpies. In general, we find that *ElemNet* has strong predictive performance across many classes



(a) Ti-O Holdout Test



(b) Na-Fe-Mn-O Holdout Test

Figure 3.5. Predicted phase diagrams from the hold-out test. These charts show the convex hulls predicted for the (a) Ti-O binary and (b) Na-Mn-O from ML models that were trained without any data from each system in their training set. We compare the performance of a Random Forest model trained using only element fractions (RF-Comp), RF trained using physical features (RF-Phys) and a deep learning model (*ElemNet*). Each vertex on the convex hull corresponds to the composition of a stable compound. The black lines on each chart show the OQMD convex hull. We find that the deep learning model has the fewest predictions outside the regions where compounds are known to form, for both the Ti-O and Na-Mn-O phase diagrams.

of materials and is most accurate for stable compounds that contain elements with fewer possible oxidation states.

3.3.7. Learning Interaction between Elements

Due to the absence of domain knowledge in materials representation for *ElemNet*, one potential issue that might arise is that it may have difficulty generalizing trends learned from one materials system to systems not included in the training set. When presented with an entry from a system that was not included in a training set, the inputs to *ElemNet* would be in a previously-unobserved portion of feature space. In contrast, models that rely on physical features suffer from this problem less. For example, consider a case where a training set contains no entries with both Ti and O together, and a ML model is tasked with predicting the formation enthalpy of TiO_2. A model trained on the features from Ward *et al.* [**14**] would be provided with useful information such as "TiO_2 is charge-balanced given the known oxidation states of Ti and O", and that "Ti_2O_3 has a similar difference in electronegativities to Al_2O_3". Without these physical features as guidance, the prediction task for *ElemNet* could potentially be more difficult.

To further test the predictive accuracy of *ElemNet* with respect to the above-described concern, we designed a holdout test where we withheld all training examples from several systems. We first analyzed the training set to determine that Ti-O is the binary chemical system with largest number of compositions in the training set and, similarly, that Na-Mn-O and Na-Fe-O are the two most common ternary chemical systems. Next, we created two separate training sets and test sets for two different holdout tests. For the first test, we withheld all entries that contain both Ti and O to use as a test set (561 entries) and used all other entries as a training set. For the second test, we withheld all entries from the Na-Fe-Mn-O quaternary phase diagram (i.e., any compound that contains exclusively Na, Mn, Fe, and O) - total of 96 entries. Each of these training/test splits provides a unique way for

evaluating whether a ML model can accurately assess previously-unobserved combinations of elements.

We found that *ElemNet* outperformed both Random-Forest-based models (with and without physical features) in both of these cross-validation tests. The RF model without physical features achieves an MAE of 0.323 eV/atom on the Ti-O holdout test, and a MAE of 0.405 eV/atom on the Na-Fe-Mn-O holdout test. The performance of this model is quite poor when considering that the mean absolute deviation of the test sets are 0.478 and 0.792 eV/atom for the Ti-O and Na-Fe-Mn-O tests, respectively. The RF model using physical attributes is significantly better with MAE of 0.198 and 0.179 eV/atom for each test, which again illustrates the importance of physical features for conventional machine learning models. We found that *ElemNet* achieves markedly better performance on both tests (MAE of 0.138 and 0.122 eV/atom), demonstrating that *ElemNet* can infer the properties of unobserved chemical systems better than existing machine learning models.

ElemNet having quantitatively better accuracy on the test sets is promising, but it still does not effectively capture whether this network is better at discovering stable compounds. To test the discovering potential of each model, we emulated searching for stable compounds by using each model to evaluate a large number of candidate materials from each of the systems held out from the training set. These systems are composed of commonly-occurring elements, for these tests we assume that they are well studied and that there are no yet-undiscovered compounds that are not included in the OQMD. Figure 3.5 illustrates the formation enthalpies and convex hull predicted by each of the ML models, compared to the known DFT result. We find that *ElemNet* reproduces the Ti-O and Na-Mn-O phase diagrams the most accurately. All three models correctly

identify that there should be a stable compound near TiO_2, and all miss the Ti-rich stable compounds (e.g., Ti_2O). This happens because the Ti-rich stable compounds have the Magneli phases which is specific to Ti-O system which are absent from training set; hence, they can not learn the specific behavior of Ti-rich compounds [**97, 98**]. However, both Random Forest models predict spurious minima near pure O, while *ElemNet* makes no spurious predictions. *ElemNet* also has the fewest number of spurious predictions in the Na-Mn-O system, where it captures that ternary compounds are only known to form in the region bounded by Na_2O, MnO_2, and MnO. In contrast, the two RF-based models predict many stable compounds in Na- and O-rich regions where no compounds are known to exist. Consequently, we conclude that our deep learning model achieves not only better accuracy on these holdout tests but it can also predict the locations of unknown, stable phases with much higher fidelity than current best ML based predictive techniques.

3.3.8. Chemistry Insights

ElemNet is evidently able to learn a useful representation of materials, given its strong prediction scores in the ten-fold cross validation and the hold-out tests. To understand how this network is performing so well, we studied the representation learned by the network. In deep neural networks, the inputs (known as activations) to each successive hidden layer become less related to the input data and more strongly related to the output. In our case, the activations for each layer are incrementally better representations of compositions for predicting formation enthalpy. We interrogated these representations by providing specific inputs to the network and measuring the activations of the network for

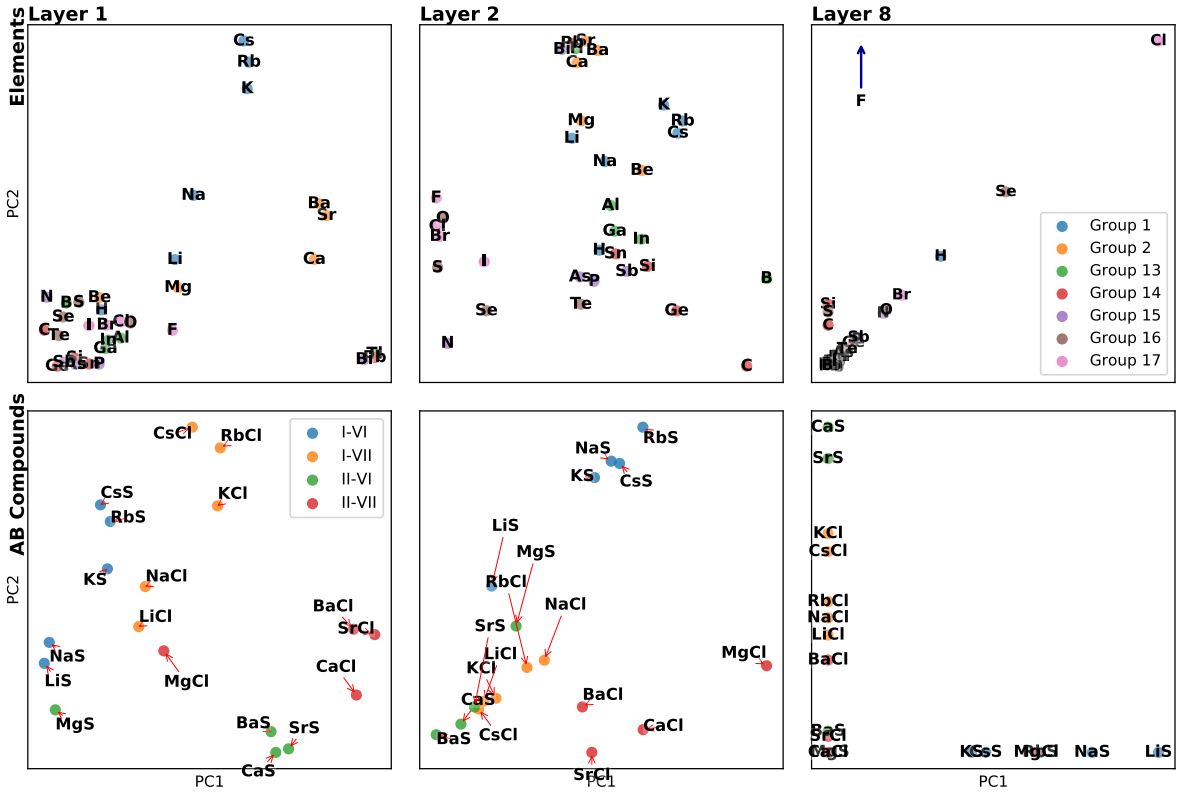


Figure 3.6. Visualization of the activations of different materials in *ElemNet*. Each frame shows a 2D projection (using PCA) of the activations of different materials in several layers of *ElemNet*, which shows which materials have similar representations. The upper row shows the activations of different elements, where each point is a different element and is colored by the group number. The second row shows the activations of AB compounds formed of group I and II metals combined with S (group VI) or Cl (group VII). We note that elements from the same group in the periodic table, such as alkali metals, are clustered together in the early layers of the network, and that later layers reflect properties related to combinations of elements (e.g., charge balance).

several hidden layers. We can then understand the behavior of the network by comparing how the activations change for different materials.

Specifically, we studied the activations of different main group elements and AB compounds that contain S or Cl paired with an Group I or Group II metal. Figure 3.6 shows the activations for each subset for the 1st, 2nd, and 8th layers of the network. As the hidden layers are composed of a large number of activations, we only considered the first two principal components of activations for this analysis. By projecting the activations down to a two-dimensional representation, we can view which compositions have similar representations and, with our knowledge of materials science, infer what kind of features the network is learning.

The 1st layer of the network exhibits clustering between elements based on their group number. The alkali and alkali earth metals, in particular, are easily identifiable and well-separated from the elements of other groups. Several groups of elements are also well-ordered by their period. The alkali metals group is ordered H, Li, Na, K, Rb, Cs from left to right and the halogens are ordered in a descending period. Elements groups are also separated where appropriate. Bi is clustered near Pb and Tl but not other chalcogens, which makes sense given that is the only metal in its group. B is also separated from the cluster containing Al, Ga, and In, which reflects that B is a metalloid unlike the other metallic elements in Group 13. Given the remarkably-clear periodic trends, it is worth emphasizing that no information about groups and periods of the periodic table was provided to *ElemNet*; all of these similarities are learned from the data.

The clustering of elements becomes less clear in later hidden layers in the network. Groups of elements are still clearly visible in Layer 2, although the ordering by period is less evident. By Layer 8, periodic trends are nearly unrecognizable in the activations of

each element. One possible explanation is that each layer of the network is gradually learning more complex features in a way similar to networks built for image classification.[**4, 3**] The early layers of the network are learning features based directly on the input values (i.e., presence of certain types of elements). Later layers in the network are learning more complex features of the compositions that have more to do with the interactions between elements than the types of elements present, which would explain why the similarity of elements becomes less visible in the activations.

To test our hypothesis that later layers in the model network capture features related to interactions between elements, we measured the activations AB compounds composed of alkali and alkaline earth metals combined with S or Cl. In the first layer, the compounds are clustered by similar groups and the distances between clusters are related to chemical similarity. The I-VII compounds (e.g., LiCl) are clustered together and closer to II-VII (for example, MgCl), which contain one element from the same group, than they are to II-VI compounds, which have no groups in common with I-VII compounds. Grouping based on similarity of element groups becomes less apparent in the second layer. I-VII compounds are now closer to II-VI compounds than any other group. We hypothesize that this change in the grouping is a result of both I-VII and II-VI compounds being charged balanced, which means they should have more negative formation enthalpies. The activations of the 8th layer show some of the I-VI and II-VI compounds together, though there are more violations of the rule (for example, BaS is far from CaS). The grouping based on charge balance is imperfect (Be-containing compounds from a separate cluster from the other group II compounds), but it is clear that the later layers are more related to interactions between elements than the presence of single elements. Overall,

the activations for both single elements and binary compounds demonstrate the power of deep learning networks to learn essential domain knowledge without specially-designed inputs.

3.3.9. Combinatorial Screening for New Materials Candidates

As our deep learning model can make robust and fast predictions, it can be used to perform combinatorial screening in huge composition space for discovery of new materials. As a case study, we conducted a combinatorial screening using our model in a huge composition space of around half a billion compounds to study if it can identify stable compounds which are not present in our training set. We first generate a list of about $450M$ hypothetical compounds of the form $\mathrm{A}_w\mathrm{B}_x\mathrm{C}_y\mathrm{D}_z$ where the elements (A-D) can be any of the 86 elements in the OQMD besides He, Ne and Ar, and w-z are positive integers where $w+x+y+z \leq 10$. The order of the elements are not fixed based on electronegativity. The compositions are unique in the sense that the ratio of constituent elements, i.e., we take AB and A_2B_2 as one composition AB since they have same composition ratio. Since we are taking the combination, there is no duplicate counting. We then evaluate the ΔH_f of these compositions using *ElemNet*. As *ElemNet* is two orders of magnitude faster than the current best ML based predictive models [**13, 14**], it allows extremely fast scanning for the discovery of new materials compared to the models in practice – we scan the entire composition space of $450M$ within few days of GPU time.

We identified compositions where it could be possible to form a new compound by identifying the compositions where *ElemNet* predicted a formation enthalpy much lower than the OQMD convex hull. Specifically, we computed the difference between the ΔH_f

predicted by *ElemNet* at each composition to the ΔH_f of the OQMD convex hull at that composition. Considering that 95% of the predictions on our test set had an error less than 0.2 eV/atom, we removed all predictions where this difference is smaller than 0.2 eV/atom to identify the predictions most likely to be correct. In total, we found 232 binary, 14,366 ternary, and 353,352 quaternary chemical systems out of the $4.3M$ compositions where the *ElemNet* ΔH_f is below the current OQMD hull by at least 0.2 eV/atom. The list of these binary and ternary compositions is available in its entirety in the Supplementary material (we could not upload the quaternary compositions due to space limit for Supplementary material) of [**99**].

Our first step for validating these predictions was to determine whether any compositions correspond to known compounds from the Inorganic Crystal Structure Database (ICSD) that are absent from the OQMD. These "missing" ICSD compounds are reasonable guesses for stable compounds, as many ICSD compounds are stable. We assembled a list of ICSD compounds not in the OQMD by first identifying all 92,756 unique compositions of compounds in the ICSD and then the 63,823 that are farther than 1% (measured using the L_2 distance) of an entry in our training set. If we restrict the prediction to be within 1% of the ICSD composition, the $4.3M$ predicted compositions includes 29 ICSD binary compounds not in the OQMD, 179 ternary compounds, and 80 quaternary compounds. If we decrease the tolerance to 10%, our model identifies 108 of the missing ICSD binary compounds, $1,121$ ternaries, and $1,087$ quaternaries. The number of ICSD compounds we find with our *ElemNet* model is small compared to the number of ICSD compounds not in the OQMD, but this is not unexpected. For one, we apply a large threshold for the hull distance (0.2 eV/atom), such that the compounds we find must be

very stable compared to compounds already in the OQMD. Finding some predictions from *ElemNet* that match up to ICSD entries shows *ElemNet* is at least identifying compounds that are reasonable to assume to be stable.

To further characterize the predictions of *ElemNet*, we analyzed the how the predictions are distributed across composition space. Over 20% of the systems predicted to contain new stable compounds include lanthanides or actinides, which is unsurprising given that compounds of these elements have not been studied as extensively as other elements. We, therefore, exclude actinide and lanthanide compounds from further analysis, and identify predictions from systems with more commonly occurring elements for further study, as shown in Table 3.3. The predictions for compounds that include Li, K, or Na are particularly illustrative. We note that our model predicts KF_6, NaF_8, OF_9 and SeF_9 to be stable, which is unlikely given the known oxidation states and suggests *ElemNet* underestimates the enthalpy of F-containing compounds, especially at high F-fractions. The predictions for the ternary compounds are interesting as they reflect realistic oxidation states of each element despite the model having no information about oxidation states in the input. Additionally, KY_2F_7 and NaY_2F_7 are reasonable predictions given that they have already been synthesized experimentally [**100**]. NaY_2F_7 is indeed stable in the OQMD and KY_2F_7 is only unstable by 50 meV/atom. The prediction of quaternary fluorides with Na and Cs are also reasonable, given their similar stoichiometries to many known Elpasolite phases[**101**]. Overall, the predictions for Li-, K-, or Na-containing compounds illustrates that *ElemNet* is making reasonable predictions. The few numbers of predictions of new $3d$ metals oxides are in agreement with our expectations, given how extensively these materials have been studied. The only new binary oxide we predicted

is Cu_2O, which is a known compound and appears in this list because *ElemNet* overestimates its formation enthalpy. We also predict $Zn_2Cu_3O_3$ to be stable, which is unlikely because ZnO-CuO is known to be phase separate.[**102**] These two unlikely predictions suggest that the formation enthalpies of Cu oxides may be generally overestimated by the models, which could be an effect of Cu_2O being in the test set for *ElemNet* rather than the training set. The quaternary prediction, $TiZnCrO_5$, is potentially interesting given that it is charged balanced and that there are already several known $ABCO_5$ oxides[**103, 104**]. Overall, these few subsets of compounds once again show that *ElemNet* is making reasonable predictions for new materials – an outstanding feat given how little knowledge of materials science was used to create it.

Table 3.3. Subset of Potential Stable Compounds Predicted using *ElemNet*. Out of the 450M predictions, we determined the number of systems where *ElemNet* identifies at least one new potential stable compound. We list the number of binary, ternary, and quaternary systems for several categories of compounds along with the two most stable predictions. We validated some of the these compounds- NaY_2F_7 and KY_2F_7 using DFT computations by leveraging crystal structures of existing materials with similar stoi-chemistry; we found them to be stable using DFT, further literature search revealed that they have already been synthesized recently. Our model predicts Cu_2O as the only new binary oxide which is a known compound but was not in our training set.

Category	**Binary**		**Ternary**		**Quaternary**	
	Count	Examples	Count	Examples	Count	Examples
[Li,K,Na]-Containing	4	KF_6 NaF_8	707	$\boldsymbol{NaY_2F_7}$ $\boldsymbol{KY_2F_7}$	18446	$CsNa_2CdF_4$ Na_2CrPbF_5
Chalco-/oxyhalides	5	OF_9 SeF_9	522	Y_2OF_6 Sc_2OF_7	17184	$Sr_3Cu_2IO_4$ Zr_6RhIO_2
Metal Oxides	1	$\boldsymbol{Cu_2O}$	81	KTi_4O_5 $ReAu_2O_5$	501	$YAlV_2O_6$ $Y_4FeBi_2O_3$
3*d* Metal Oxides	1	$\boldsymbol{Cu_2O}$	3	$Zn_2(CuO)_3$ Ti_5CuO_2	1	$TiZnCrO_5$
Intermetallics	11	Nb_5Sn_3 Al_5Ir_3	123	$HfAl_5Ir_3$ YAl_4Ir_3	425	Sc_5NiSn_3Mo $ZrAl_5OsRh$
Intermetallics $HHI_p < 2500$	0		0		1	$NaMn_2AlAu_6$

3.4. Discussion

Conventional predictive ML modeling approaches require manual feature engineering of materials representation to incorporate domain knowledge in the model inputs. However, there is no consensus among researchers on how many and which physical attributes to include into the model inputs, such that they incorporate all the important domain knowledge required to make accurate predictions. Here, we demonstrated that the need to engineer features for materials can be bypassed by leveraging a deep learning approach. A deep learning model can learn the optimal materials representation required for the prediction task by automatically capturing the chemical interactions between different elements from the training dataset using artificial intelligence, without any need for manual feature engineering, domain knowledge or human intuition; which can allow it to make better prediction for chemical systems absent in the training set than the conventional ML models.

The general belief in scientific community is that deep learning techniques require big training datasets [**3**] to perform well; however, we demonstrate that *ElemNet* can perform better than conventional ML models by leveraging only 2% of the OQMD dataset for training, which shows that deep learning can be used to build predictive models on relatively smaller materials and scientific datasets such as of size $4k$. Our results provide a stimulus for researchers to use DNN based approaches for building predictive models on their datasets. Since the proposed deep learning approach yielded the highest accuracy to date, it provides a new direction for more robust and fast predictions to identify composition regions containing materials with strong-negative formation enthalpies for discovery. We scanned around 450 million candidate compositions for novel ternary and

quaternary compounds, and predicted that new stable compounds could be found in about $368k$ different chemical systems. The entire list is made available in the Supplementary Material of [**99**] to facilitate further research and analysis for accelerating the process of new materials design and discovery. We have added *ElemNet* to our existing online formation enthalpy calculator [**28, 13**] publicly available at `http://info.eecs.northwestern.edu/FEpredictor` so that researchers can publicly access and evaluate its predictions. The model is also available at `https://github.com/dipendra009/ElemNet` with the trained weights and sample code to demonstrate how to load and use the model for making predictions and performing combinatorial screening for new materials discovery. We plan to keep refining the model by training on larger datasets as they become available in future which will help in further improvement in the prediction results.

CHAPTER 4

IRNet: A General Purpose Deep Residual Regression Framework for Materials Discovery

4.1. Introduction

Materials discovery plays an important role in many domains of science and engineering [**105, 106**]. The slow pace of development and deployment of new/improved materials is a major bottleneck in the innovation cycles of emerging technologies [**107**]. Collection of large scale datasets through experiments and first-principle computations such as high throughput density functional theory (DFT) calculations [**108, 109, 42**] and the emergence of integrated data collections and registries [**110, 111**] have spurred the interest of materials scientists in applying machine learning (ML) models to understand materials and predict their properties [**15, 16, 14, 56, 50, 18, 63, 20, 22**], leading to the novel paradigm of materials informatics [**9, 47, 112, 50**]. Such interests have been supported by government initiatives such as the Materials Genome Initiative (MGI) [**113**].

Predictive modeling tasks in materials science are generally regression problems where we need to predict materials properties from an input vector composed of numerical features derived from their composition and/or crystal structures by incorporating domain knowledge [**15, 16, 14, 56, 50, 63, 99**]. Since the model input contains vector of independent features, the neural network models used for such tasks are composed of fully connected layers. Vanishing gradient and performance degradation issues that arise

when using deeper architectures have caused the neural network architectures used for such prediction modeling to be limited in their depth [**22, 20, 114, 99, 115**]. For instance, Montavon et al. [**22**] trained a four-layer network on a database of around 7000 organic compounds to predict multiple electronic ground-state and excited-state properties. In the Harvard Energy Clean Project, Pyzer-Knapp et al. [**20**] used a three-layer network for predicting power conversion efficiency of organic photo-voltaic materials. Zhou et al. [**115**] used a fully connected network with single hidden layer to predict formation energy from high-dimensional vectors learned using Atom2Vec. ElemNet [**99**] used a 17-layered architecture to learn formation energy from elemental composition, but experienced performance degradation beyond that depth. Hence, domain scientists have mainly used traditional ML techniques such as Random Forest, Kernel Ridge Regression, Lasso, and Support Vector Machines for materials prediction tasks [**68, 67, 13, 14**].

Recently, several projects have used domain knowledge-based model engineering within a deep learning context for predictive modeling in materials science [**81, 116, 117**]. Deep learning was used for directly predicting the crystal orientations of polycrystalline materials from their electron back-scatter diffraction patterns [**116**]. SchNet [**81**] used continuous filter convolutional layers to model quantum interactions in molecules for the total energy and interatomic forces that follows fundamental quantum chemical principles. Boomsma and Frellsen [**118**] introduced the idea of spherical convolution in the context of molecular modelling, by considering structural environments within proteins. Smiles2Vec [**117**] and CheMixNet [**114**] have applied deep learning methods to learn molecular properties from the molecular structures of organic materials.

Our goal here is to design a general purpose deep regression network for predicting the properties of inorganic materials from their compositions and/or crystal structures, without using any domain knowledge-based model engineering. We introduce the idea of residual learning to deep regression networks composed of fully connected layers. In a fully connected network, the number of parameters is directly proportional to the product of the number of inputs and the number of output units. Several works have dealt with the performance degradation issue due to vanishing or exploding gradients for other types of data mining problems [**119, 120, 121**]. Srivastava et al. [**119**] introduced an LSTM-inspired adaptive gating mechanism that allowed information to flow across layers without attenuation; the gating mechanism required more model parameters. They designed highway networks composed of up to 100 layers that could be optimized. A highway network [**119**] uses gated connections, which double the number of parameters in a fully connected network. In a DenseNet [**121**], all previous inputs are combined before being fed into the current layer. For a fully connected network, this approach results in a tremendous increase in the number of model parameters, a particular problem when working with limited GPU memory. He et al. [**120**] introduced the idea of residual learning, in which a stack of layers learns the residual mapping between the output and input; they built deep CNN models composed of 152 layers for image classification problem. Since the input is added to the residual output, the number of required parameters for residual learning was lower than that in Srivastava et al. [**119**]. This technique has been used in several CNN and LSTM architectures, with shortcut connections being placed after a stack of multiple CNN or LSTM layers to build deeper networks for better performance [**73, 122, 123**]. For a fully connected network, an elegant approach is to use the

residual mapping approach used in ResNet [**120**]. However, although residual learning has been widely used in classification networks, no previous work leverages residual learning for building deep regression networks composed of fully connected layers for numerical vector inputs.

In this work, we study and propose design principles for building deep residual regression networks composed of fully connected layers for data mining problems with numerical vectors as inputs. We introduce a novel deep regression network architecture with individual residual learning (IRNet), in which shortcut connections are placed after each layer such that each layer learns only the residual mapping between its output and input vectors. We compare IRNet against two baseline deep regression networks: and a stacked residual network (SRNet) with shortcut connections after stack of multiple layers. We focus on the *design problem* of learning the formation enthalpy of inorganic materials from an input vector composed of 126 features representing their crystal structure, and another 145 composition-based physical attributes from the OQMD-SC dataset. OQMD-SC contains $435,582$ materials with their composition and crystal structure from the Open Quantum Materials Database (OQMD) [**42**].

Our proposed 48-layered IRNet achieves significantly better performance than does the best state-of-the-art ML approach, Random Forest: a mean absolute error (MAE) of 0.038 eV/atom compared to 0.072 eV/atom on the OQMD-SC dataset. IRNet also performed significantly better than both the plain network and SRNet. The use of individual residual learning (IRNet) led to faster convergence compared to the existing approach of residual learning in SRNet, while maintaining the same number of parameters. We also evaluated IRNet performance for learning materials properties with 145

composition-based physical attributes in two other datasets: OQMD-C (341, 443 data points) and MP-C (83, 989) [**109**]. IRNet significantly outperformed the plain network and the traditional ML approach on the new prediction tasks; the deeper models performing better in case of larger dataset (OQMD-C). We performed a combinatorial search for materials discovery using the proposed models. The models were trained on 3.2111×10^4 entries in OQMD-SC-ICSD dataset. The evaluation was performed by searching for stable materials with specific crystal structures. The proposed model provided significantly more accurate predictions compared to the traditional ML approach (Random Forest).

4.2. Background

4.2.1. Property Prediction

The prediction of chemical properties from material crystal structure and composition is strongly related to the discovery of new materials. One important material property is formation enthalpy: the change in energy when one mole of a substance in the standard state (1 atm of pressure and 298.15 K) is formed from its pure elements under the same conditions [**124**]. In other words, it is the energy released when forming a material (chemical compound) from the constituent elements. By knowing the formation enthalpy, one can know whether the material is stable and thus feasible to experimentally synthesize in laboratory. The more negative the formation enthalpy, the more stable the compound. Materials properties also contain various other properties [**42, 109**].

4.2.2. Materials Representation

Most ML approaches require manual feature engineering and a representation that incorporates domain knowledge into model inputs. They thus take composition-based physical attributes and/or crystal structure as the input. Recently, Ward et al. [**14**] presented a ML framework for formation energy prediction that used an input vector with 145 features computed from composition; stoichiometric attributes, elemental property statistics, electronic structure attributes, and ionic compound attributes. We leverage this approach to compute the 145 physical attributes used in our datasets.

The crystal structure of a material is defined by the shape of the unit cell and associated atom positions, which together define the repeat pattern of the atomic structures that form the material. It is possible to represent the unit cell shape and atom positions as a vector of $3+3N$ features (where N is the number of atoms), but this representation is not suitable for ML. The atomic coordinates are not unique—rotating or translating the coordinate system does not change the material—and they do not readily reflect important features of the material (e.g., bond lengths). Many crystal structure representations, such as "bag of bonds" [**125**] and histograms of bond distances [**66**], have been developed to address this problem. We use the representation developed by Ward et al. [**59**], which uses 126 features derived from the Voronoi tessellation of a material. The Voronoi tessellation of a crystal structure provides a clear description of the local environment of each atom, which is used to compute features such as the difference in elemental properties (e.g., molar mass) between an atom and its neighbor [**59**].

4.3. Design

We next describe how we build deep residual regression models, composed of multiple fully connected layers, for data mining problems with numerical vectors as inputs. We first introduce a plain network without any residual learning. Next, we build a stacked residual network by introducing shortcut connections for residual learning after each of a number of stacks, each composed of one or more layers with the same configuration. Finally, we introduce our novel individual residual learning approach, in which shortcut connections are used after every layer. We use the plain network and stacked networks later as baseline models for comparison against the individual residual network.

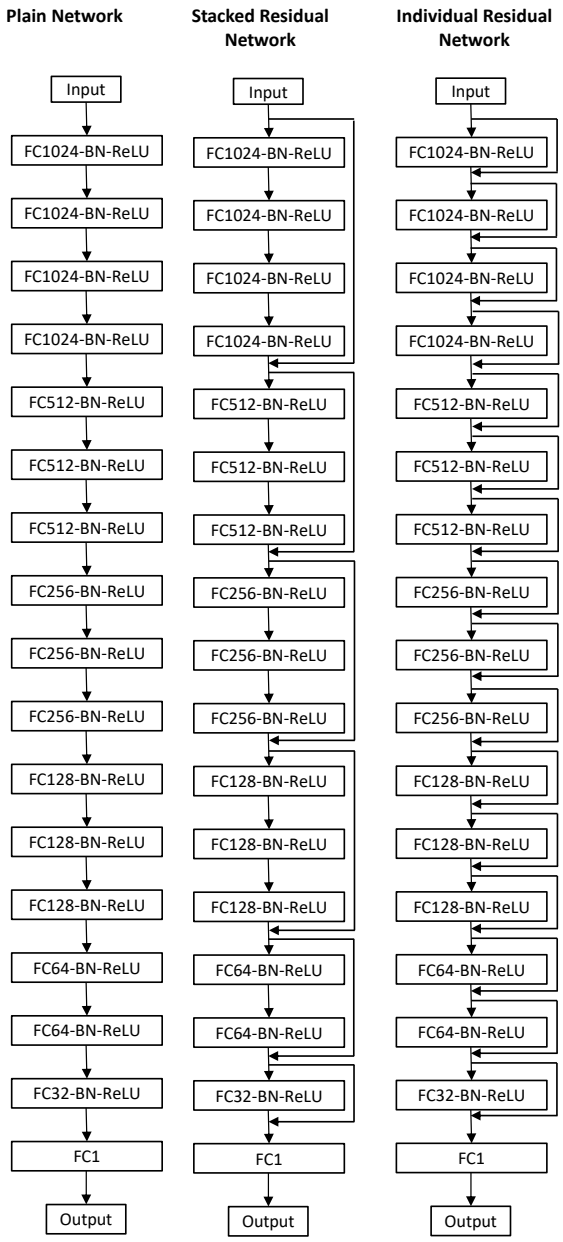


Figure 4.1. Three types of 17-layer networks. Each "layer" is a fully connected neural network layer with size as described in Table 4.1; all but the last are followed by batch normalization and ReLU. A *plain network* simply connects the output of each layer to the input of the next. A *stacked residual network* (SRNet) places a shortcut connection after groups of layers called stacks. An *individual residual network* (IRNet) places a shortcut connection after every layer.

Table 4.1. Detailed configurations for different depths of network architecture. The notation [...] represents a stack of model components, comprising a single (FC: fully connected layer, BN: batch normalization, Re: ReLU activation function) sequence in the case of IRNet and multiple such sequences in the case of SRNet. Each such stack is followed by a shortcut connection.

Output	**17-layer SRNet**	**17-layer IRNet**	**24-layer SRNet**	**24-layer IRNet**	**48-layer SRNet**	**48-layer IRNet**
1024	[FC1024-BN-Re x 4]	[FC1024-BN-Re] x 4	[FC1024-BN-Re x 4]	[FC1024-BN-Re] x 4	[FC1024-BN-Re x 4] x 2	[FC1024-BN-Re] x 8
512	[FC512-BN-Re x 3]	[FC512-BN-Re] x 3	[FC512-BN-Re x 4]	[FC512-BN-Re] x 4	[FC512-BN-Re x 4] x 2	[FC512-BN-Re] x 8
256	[FC256-BN-Re x 3]	[FC256-BN-Re] x 3	[FC256-BN-Re x 4]	[FC256-BN-Re] x 4	[FC256-BN-Re x 4] x 2	[FC256-BN-Re] x 8
128	[FC128-BN-Re x 3]	[FC128-BN-Re] x 3	[FC128-BN-Re x 4]	[FC128-BN-Re] x 4	[FC128-BN-Re x 4] x 2	[FC128-BN-Re] x 8
64	[FC64-BN-Re x 2]	[FC64-BN-Re] x 2	[FC64-BN-Re x 3]	[FC64-BN-Re] x 3	[FC64-BN-Re x 4] x 2	[FC64-BN-Re] x 8
32	[FC32-BN-Re]	[FC32-BN-Re]	[FC32-BN-Re x 2]	[FC32-BN-Re] x 2	[FC32-BN-Re x 4]	[FC32-BN-Re] x 4
16			[FC16-BN-Re x 2]	[FC16-BN-Re] x 2	[FC16-BN-Re x 3]	[FC16-BN-Re] x 3
1	FC1					

4.3.1. Plain Network

The model architecture is formed by putting together a series of stacks, each composed of one or more sequences of three basic components with the same configuration. Since the input is a numerical vector, the model uses a fully connected layer as the initial layer in each sequence. Next, to reduce the internal covariance drift for proper gradient flow during back propagation for faster convergence, a batch normalization layer is placed after the fully connected layer [**126**]. Finally, ReLU [**88**] is used as the activation function after the batch normalization.

The simplest instantiation of this architecture adds no shortcut connections and thus learns simply the approximate mapping from input to output. We refer to this network as a *plain network*.

4.3.2. Stacked Residual Learning

Deep neural networks suffer from the vanishing or exploding gradient problem [**127, 128**], which hampers convergence, and also from the degradation problem: as network depth increases, accuracy becomes saturated and then degrades rapidly. One approach to dealing with these issues is to use shortcut connections for residual learning [**120, 121, 119**].

Here, we introduce the idea of residual learning to deep regression networks composed of fully connected layers. In a fully connected network the number of parameters is directly proportional to the product of the number of inputs and the number of output units. The gated connection approach from the highway network and the use of all previous inputs from DenseNet [**121**] would result in a huge increase in model parameters that would

not fit in GPU memory. Hence, for a fully connected deep neural network, the residual learning from He et al. [**120**] is the most elegant approach.

We use stacks of consecutive layers with the same configuration, with the first stack composed of four sequence of layers and the final stack of two sequences. Instead of directly fitting the underlying mapping, the stacked layers explicitly learn the residual mapping. If the underlying mapping is denoted by $H(\vec{x})$, the stacked layers fit the residual mapping of $F(\vec{x}) = H(\vec{x}) - \vec{x}$. If the input and output of a stack have the same dimensions, they can be added by using a shortcut connection for residual learning. If the output of a layer, $F(\vec{x})$, has a different dimension than the input $\vec{x}$, we perform a linear projection W_s to match the dimensions before adding:

$$\vec{y} = F(\vec{x}) + W_s\vec{x}, \tag{4.1}$$

where $\vec{x}$ and $F(\vec{x})$ are the input and output to the stack of layers, respectively. W_s acts as a dimension reduction agent. We refer to such a network with shortcut connections across each stack as a *stacked residual network* (SRNet).

4.3.3. Individual Residual Learning

He et al. [**120**] introduced the idea of using shortcut connections after a stack composed of multiple convolutional layers. The latest Inception-ResNet [**73**] architecture for image classification follows a similar approach, with shortcut connections used between stack of Inception-ResNet blocks, where each block is composed of multiple convolutional layers followed by 1×1 convolutional filters for dimension matching. In our case, the stacks are composed of up to four sequences, with each sequence containing a fully connected layer,

a batch normalization, and ReLU. Our stacks are comparably more complex and highly non linear when compared to those used in CNN models for image classification. Also, learning the residual regression mapping from input to output vector is comparatively harder than the residual learning for classification task; the activations and gradients can vanish within the stacks.

To solve this issue, we introduce a novel technique of individual residual learning for sequences containing a fully connected layer with batch normalization and non linear activation. We place a shortcut connection after every sequence, so that each sequence needs only to learn the residual mapping between its input and output. This innovation has the effect of making the regression learning task easy. As each "stack" now comprises a single sequence, shortcut connections across each sequence provide a smooth flow of gradients between layers. We refer to such a deep regression network with individual residual learning capability as an *individual residual network* (IRNet).

The detailed architectures for networks with different depths are illustrated in Figure 4.1 and Table 4.1. There are several deep network design techniques based on advanced branching techniques such as Inception [**30, 73**] and ResNext [**129**], but here our goal is to design a general purpose deep regression network framework rather than optimizing for a specific prediction task. We will explore branching techniques in future work.

4.4. Experimental Results

We now present a detailed analysis of the design and evaluation of our deep regression networks with residual learning. We proceed in three stages. First, we present our evaluation of the proposed deep regression model (IRNet) for the *design problem* and compare

its performance with the plain network, SRNet, and traditional ML approaches when applied to the OQMD-SC dataset. Next, we evaluate the proposed model architecture by learning materials properties from physical attributes for compounds in the OQMD-C and MP-C datasets. Finally, we perform a combinatorial search for materials discovery by training on the OQMD-SC-ICSD dataset. Before presenting our evaluation, we discuss the experimental settings and datasets that we use in this work.

Experimental Settings. We implement the deep learning models with Python and TensorFlow [**93**]. We performed extensive architecture search and hyperparameter tuning for all deep learning and other ML models used in this study. For deep learning models, we experimented with different activation functions: sigmoid, tanh, and ReLU, both for the intermediate layers and for the final regression layer. We explored learning rates in [1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6]; StochasticGradientDescent, MomentumOptimizer, Adam, and RMSProp optimizers; and mini-batch sizes in [32, 64, 128]. Since we are dealing with regression output, we experimented with mean squared error and mean absolute error as the loss functions. We found the best hyperparameters to be are Adam [**130**] as the optimizer with a mini batch size of 64, learning rate of 0.0001, mean absolute error as loss function, and ReLU as activation function, with the final regression layer having no activation function. Rather than training the model for a specific number of epochs, we used early stopping with a patience of 200, meaning that we stopped training when the performance did not improve in 200 epochs. For traditional ML models, we used Scikit-learn [**94**] implementations and employed mean absolute error (MAE) as loss function and error metric.

Datasets. We used four datasets to evaluate our models: OQMD-SC, OQMD-C, MP-C, and OQMD-SC-ICSD. OQMD-SC is composed of $4.355\,82 \times 10^5$ unique compounds (unique combination of composition and crystal structure) with their DFT-computed formation enthalpy from the Open Quantum Database (OQMD) [**42**]; this is used for the *design problem*. It is composed of 271 attributes: 125 derived to represent crystal structure using Voronoi tesselations and another 145 physical attributes derived from composition using domain knowledge, as in Ward et al. [**14**]. OQMD-C is composed of $3.414\,43 \times 10^5$ compounds with the materials properties from OQMD as of May 2018. MP-C is composed of 8.3989×10^4 inorganic compounds from the Materials Project database [**109**] with a set of materials properties as of September 2018. OQMD-C and MP-C contain composition only (no structure information); we compute 145 physical attributes from the composition using Ward et al.'s methods [**14**]. OQMD-SC-ICSD is composed of entries from the Inorganic Crystal Structure Database (ICSD) [**96**] present in OQMD-SC. The datasets are randomly split into training and test sets in the ratio of 9:1.

4.4.1. Design Problem

First, we analyze the impact of different design choices by evaluating the proposed models on the design problem. The design problem involves learning to predict formation enthalpy from input vector composed of 126 attributes to represent crystal structure and 145 physical attributes in OQMD-SC dataset. An extensive architecture search and hyperparameter tuning is performed to search for the best deep regression model for the design problem.

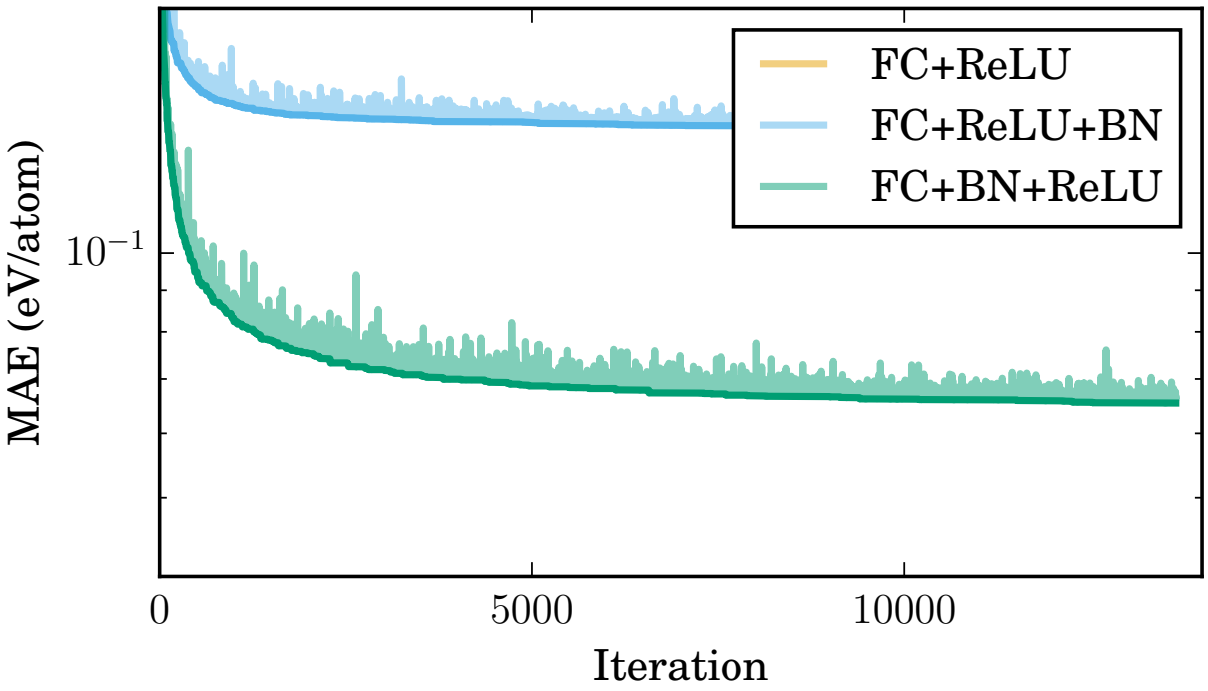


Figure 4.2. Test error curve for various plain networks for the *design problem*. Batch normalization before activation function (FC+BN+ReLU) improves performance significantly.

4.4.1.1. Basic Components. We experimented with different patterns of use of our basic components—fully connected layer, batch normalization, activation function, and dropout—within the plain network. Use of batch normalization resulted in significant reduction in errors, as seen in Figure 4.2. Batch normalization can be used either before (FC+BN+ReLU) or after the activation function (FC+ReLU+BN). For our regression problem, using batch normalization before ReLU (FC+BN+ReLU) worked better; the original work also used it before the activation function for image classification problem [**126**]. Since ReLU truncates all negative activations to zero, applying batch normalization on ReLU outputs leads to changes in the activation distribution; since the

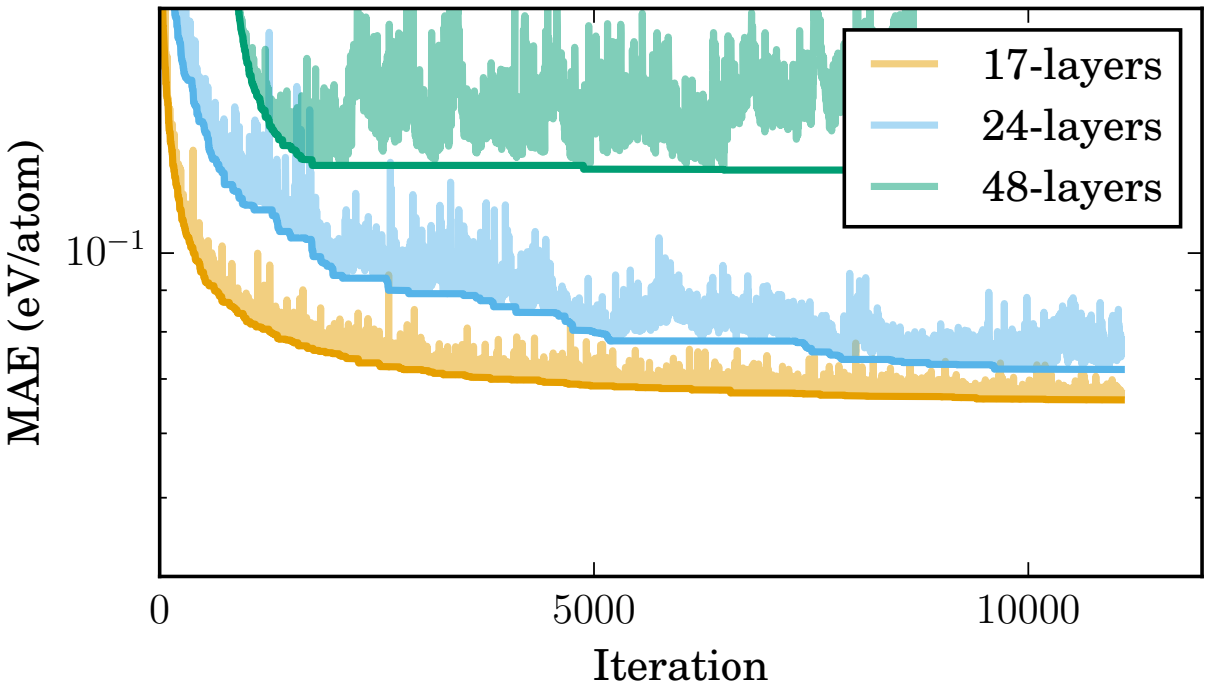
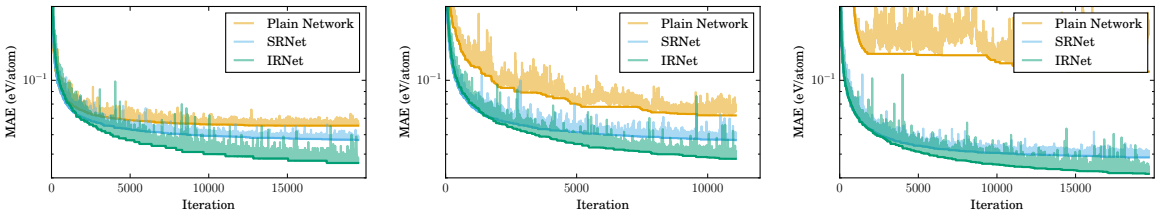


Figure 4.3. Test error curve for deeper plain networks for the *design problem*. Performance degrades with network depth, even in the presence of batch normalization.

regression output is dependent on all activations, batch normalization after ReLU leads to higher oscillations and poor convergence.

We also experimented with using dropouts after the first four stacks for better generalization; however, dropouts resulted in slight degradation in the performance. The best plain network architecture for our design problem is composed of 17 sequences containing a fully connected layer, a batch normalization and a ReLU; we refer to this as the *17-layer plain network*. as shown in Figure 4.1.

4.4.1.2. Residual Learning. Figure 4.3 shows how performance can degrade with increased depth for plain networks. This happens mainly because of the vanishing gradient problem. To solve this issue, we introduced residual learning to create SRNet and IRNet,



(a) 17-layers (b) 24-layers (c) 48-layers

Figure 4.4. Impact on residual learning for the *design problem*. Both residual networks outperform the plain network, and the individual network outperforms the stacked network for all depths of network. We observe similar trends even in the case of training error curves for all types of networks of all depths; the IRNet converges faster than the SRNet and Plain Network for all depths.

as discussed earlier. We see in Table 4.2 and Figure 4.4 that the introduction of shortcut connections to enable residual learning significantly improved model performance, presumably by helping with the smooth flow of gradients from output to input. We compared the individual residual learning in IRNet with the existing approach of use of shortcut connections after stacks of multiple layers in SRNet. The stacks are formed by putting the consecutive layers with equal number of output units in a stack.

We observe a significant benefit from the novel approach of using shortcut connections for individual residual learning in IRNet; the mean absolute error significantly decreased compared to SRNet as seen in Figure 4.4 and Table 4.2. Both the training and test error curves in the case of IRNet exhibits better convergence than both SRNet and plain network during the training.We conjecture that learning the residual between the output and the input vector of the sequence is better compared to learning the more complex residual mapping in the case of stacked residual network in SRNet. Also, if the identity mapping using shortcut connections are optimal, the residuals would be pushed to zero and hence, better suited for batch normalization to learn our regression output. This illustrates the

Table 4.2. Performance of deeper residual networks for the *design problem*. Test errors are MAE in eV/atom. Increased depth of residual network architectures leads to improved performance for both stacked and individual residual networks. The individual residual network (IRNet) clearly outperforms the stacked residual network (SRNet), achieving significantly lower MAE.

Model Type	**Plain Network**	**SRNet**	**IRNet**
17-layer	0.0653	0.0551	**0.0411**
24-layer	0.0719	0.0546	**0.0403**
48-layer	0.1085	0.0471	**0.0382**

advantage of using individual residual learning for deep regression networks composed of fully connected layers for vector inputs.

4.4.1.3. Deeper Architectures. Next, we experimented with deeper architectures composed of 24 and 48 sequences of layers for all types of deep regression networks: plain network, SRNet, and IRNet. From Figure 4.3, we can clearly observe the performance degradation issue in plain networks that do not leverage any shortcut connections for residual network. Figure 4.4 illustrates the trend in error curves. Although both types of residual networks exhibit reduced test error with increased depth, the rate of reduction for IRNet is significantly better than that for SRNet. To prevent overfitting of such deep models with large numbers of parameters to the training dataset, we used early stopping with a patience of 200. Table 4.2 shows the final MAE for all types of networks with different depths. Our results illustrates the efficiency of using individual residual learning with deeper architectures.

4.4.1.4. Comparison with Other ML Approaches. Next, we compared the performance of the proposed deep learning model with traditional ML models: see Table 4.3. We performed an extensive hyperparameter search to find the best hyperparameters for all ML models. For instance, for Random Forest model, we used a minimum sample split

Table 4.3. Performance of Traditional ML Approaches for the *design problem*. We performed extensive grid search for hyperparameter tuning for all the listed ML models. Test errors are MAE in eV/atom.

ML Approach	**Test Error**
AdaBoost	0.479
ElasticNet	0.384
LinearRegression	0.261
Ridge	0.261
SVR	0.243
KNeighbors	0.154
DecisionTree	0.104
Bagging	0.078
RandomForest	**0.072**

from [5, 10, 15, 20], number of estimators from [100,150,200], maximum features from [0.25, 0.33] and maximum depth from [10,25]. Similarly, extensive grid search for optimization of hyperparameters for other ML models are used. Among all of the traditional ML approaches considered, Random Forest achieved the best MAE of 0.072 eV/atom. By comparison, the 48-layer IRNet achieved an MAE of 0.038 eV/atom, significantly outperforming Random Forest for the design problem. Figure 4.5 illustrates the comparison of the prediction errors for the test set. Deep learning provides a more accurate and robust prediction model than does the state-of-the-art ML approach, Random Forest, predicting the formation enthalpy of 90% of the compounds in the test set with half the error of Random Forest. These results demonstrate that deep learning in general, and IRNet in particular, can help construct a robust model for predicting formation enthalpy from materials crystal structure and composition.

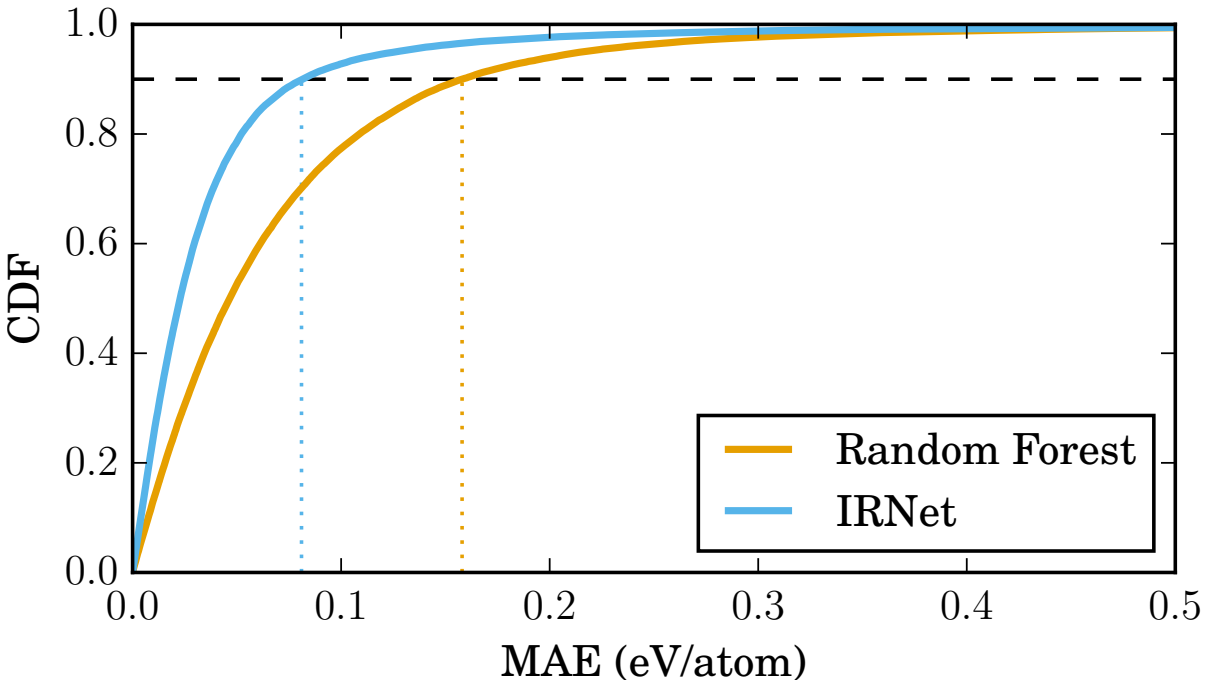


Figure 4.5. Cumulative distribution function (CDF) of the prediction errors for the *design problem*. Deep learning (IRNet) performs significantly better than the traditional ML approach, Random Forest, achieving a 90th percentile MAE of 0.081 eV/atom vs. 0.158 eV/atom for Random Forest.

4.4.1.5. Summary of design insights. We draw the following lessons from our experiments with building deep regression networks for learning regression output from numerical vector inputs.

(1) **Batch Normalization** Batch normalization works better in deep regression networks if used before ReLU. Otherwise, ReLU truncates all negative values to zero, which makes learning the regression output hard. Dropout with batch normalization slightly worsens performance.

(2) **Residual Learning** Residual learning in deep regression always performs better compared to directly learning to fit the underlying mapping from input vector to the regression output.

(3) **Individual Residual Learning** Putting a shortcut connection after each sequence of layers (IRNet) works significantly better than the conventional way of putting the shortcut connection after each stack of multiple layers (SRNet).

The presented architecture can be applied to other data mining problems with vector inputs in scientific domains; they can provide more robust and accurate predictive modeling than the existing ones based on traditional ML approach. The same architecture can be also applied to classification problem by adding a $softmax$ activation at the last layer and using *cross entropy* as the loss function.

4.4.2. Other Datasets

We evaluated the proposed deep regression architecture on learning materials properties present in two other datasets, OQMD-C and MP-C. OQMD-C is composed of $3.414\,43 \times 10^5$ samples while MP-C has 8.3989×10^4 samples; they contain the materials properties with their composition. For comparison, we used the 17-layered plain network and ten other traditional ML approaches. We did not perform hyperparameter tuning and architecture search for deep learning models for these tasks, to illustrate the general purpose use of the proposed deep regression model. The deep regression networks designed for the *design problem* were trained on an input vector containing 145 physical attributes derived from composition; they were trained from scratch using random weights initialization. For the traditional ML models, we performed an extensive grid search for hyperparameter optimization as in the previous case for the design problem.

We can observe three things from the results in Table 4.4. First, the deep learning network almost always outperforms the traditional ML approaches. Second, the proposed

network with individual residual learning performs better than the plain network in all cases. Third, deeper networks worked better in case of OQMD-C while they did not help in case of MP-C, suggesting that deeper networks work better when the dataset size is larger (OQMD-C vs MP-C). This agrees with the fact that deep neural networks perform better with big data. The results demonstrate that although the proposed model was originally designed for a different *design problem*, they almost always outperform the plain network and the traditional ML approaches used by domain scientists. We also experimented with SRNet from design problem for these prediction problems, SRNet performed better than the plain network but worse than the IRNet, similar to the results for the design problem. This illustrate that IRNet can serve as a general purpose deep learning model for different predictive modeling tasks where we need to learn the regression output from an input vector composed of materials composition and/or crystal structures.

Table 4.4. Performance on OQMD-C and MP-C datasets of our DNN models vs. 10 traditional ML approaches for regression problems: Linear Regression, Lasso, Ridge, Decision Tree, Adaboost, KNeighbors, ElasticNet, SGD Regression, Random Forest and Support Vector, with extensive grid search used to tune hyperparameters for each. Test errors are MAE in eV/atom.

Dataset	**Property**	**Best of 10 ML**	**17-layer Plain Network**	**17-layer IRNet**	**48-layer IRNet**
OQMD-C	Formation Enthalpy	0.077	0.072	0.054	**0.048**
	Bandgap	**0.047**	0.052	0.051	**0.047**
	Energy_per_atom	0.1139	0.0939	**0.0696**	-
	Volume_pa	0.473	0.0.483	0.415	**0.394**
MP-C	Bandgap	0.4788	0.396	**0.363**	0.364
	Density	0.5052	0.401	**0.348**	0.386
	Energy_above_hull	0.1184	0.098	**0.091**	0.0944
	Energy_per_atom	0.2999	0.175	**0.143**	-
	Total_magnetization	3.232	3.0897	**3.005**	-
	Volume	225.671	219.439	**215.037**	-

Table 4.5. Performance from combinatorial search. Our 17-layer IRNet, when trained on OQMD-SC-ICSD, predicts formation enthalpy (stability) more accurately than Random Forest for all three types of crystal structures considered.

Crystal Structure	**Random Forest MAE (eV/atom)**	**17-layers IRNet MAE (eV/atom)**
B2	0.5114	**0.4780**
$L1_0$	0.4793	**0.4419**
Perovskite	0.6166	**0.3693**

4.4.3. Application for Materials Discovery

Since the proposed model achieved a significant reduction in prediction error for formation enthalpy compared to state-of-the-art approach, it can be applied for high throughput materials discovery. To test the ability of the proposed method to identify new materials, we emulated a common approach in computational materials science, namely combinatorial search . A combinatorial search involves first enumerating all possible combinations of different elements on a specific crystal structure prototype, and then evaluating the stability of each resultant structure with DFT to find which are stable. We performed a combinatorial search using the evaluation settings based on the combinatorial search analysis from [**59**]. OQMD-SC-ICSD, used as a training set by Ward et al. [**59**], comprises 3.2111×10^4 entries in OQMD-SC that correspond to known, experimentally-synthesized materials in ICSD [**96**]. The proposed IRNet is trained using the OQMD-SC-ICSD dataset and evaluated by predicting the formation enthalpy (stability) of materials with crystal structures from three different, commonly occurring crystal structure types: B2, $L1_0$, and orthorhombically-distorted perovskite. These three structure types were chosen to sample structures with different kinds of bonding environments and that are stable with different types of chemistry (e.g., metals vs. oxides).

We show in Table 4.5 the deep learning model's prediction error for each type of crystal structures. To compare the performance of our deep learning model, we also trained a Random Forest model (the best traditional ML approach from previous analysis) on OQMD-SC-ICSD, with extensive hyperparameter search. Our results demonstrate that our models perform better on the evaluation candidates than does the Random Forest model. Although we do not repeat the entire combinatorial search workflow here with the proposed models, more accurate predictions on the discoveries from Ward et al. [**59**] suggest that the proposed IRNet model can improve the quality and robustness of the combinatorial search workflow. Despite a small training data size, the IRNet model provides a more robust method for performing combinatorial search for high-throughput materials discovery.

4.5. Conclusion and Future Work

In this work, we studied and proposed the design principles for building deep regression networks composed of fully connected layers for data mining problems with numerical vector input. We introduced the use of residual learning in deep regression network; we proposed a deep regression network (IRNet) that leveraged individual residual learning in each layer. The proposed IRNet outperformed the plain network (without residual learning) and traditional machine learning approaches in learning different materials properties from different size of datasets and input vector. For the *design problem* of predicting formation enthalpy from crystal structures and composition, the proposed IRNet significantly reduced the MAE from 0.072 eV/atom to 0.038 eV/atom. We were able to converge the deep regression networks with up to 48 layers, performance increasing with

greater depth. Since IRNet kept improving performance with increased depth, we plan to explore deeper IRNet architectures to study their impact on model performance and convergence, and to apply the resulting networks to data mining problems from other scientific domains. It will also be interesting to see how this model performs on experimental datasets using transfer learning from larger simulation datasets. The proposed deep learning model and design insights gained from this work can be used in building predictive models for other applications with vector inputs. The code repository is available at `https://github.com/dipendra009/IRNet`; we also plan to make the models described in this work available via DLHub [**131**].

CHAPTER 5

Enhancing Materials Property Prediction by Leveraging Computational and Experimental Data using Deep Transfer Learning

5.1. Introduction

Experimental observations have been the primary means to learn and understand various chemical and physical properties of materials [**33, 34, 35, 36, 37, 38**]. Nevertheless, since experiments are expensive and time-consuming, materials scientists have been relying on computational methods such as Density Functional Theory (DFT) [**39**] to compute materials properties and model processes at the atomic level to help guide experiments [**40**]. DFT has enabled the creation of high-throughput atomistic calculation frameworks for accurately computing (predicting) the electronic-scale properties of a crystalline solid using first principles, which can be expensive to measure experimentally. Over the years, such DFT-computations have led to a number of large datasets like the Open Quantum Materials Database (OQMD) [**41, 42**], the Automatic Flow of Materials Discovery Library (AFLOWLIB) [**43**], the Materials Project [**44, 132, 133**], Joint Automated Repository for Various Integrated Simulations (JARVIS) [**134, 135, 136, 137**], and the Novel Materials Discovery (NoMaD) [**45**]. They contain DFT-computed properties of $\sim 10^4 - 10^6$ materials which are either experimentally-observed [**138**] or hypothetical materials. The availability of such large DFT-computed datasets has spurred

the interest of materials scientists to apply advanced data-driven machine learning (ML) techniques to accelerate the discovery/design of new materials with select engineering properties [**51, 52, 17, 13, 53, 54, 55, 15, 56, 57, 58, 14, 59, 60, 61, 116, 62, 50, 63, 64, 65, 99, 139, 140, 141**]. Such predictive models enable reducing the size of the search space for material candidates and help in prioritizing which DFT simulations and, possibly, experiments, to perform. Training data sizes can have significant impact on the quality of prediction performance in machine learning, and particularly in deep learning [**3**]. This has also been proven specifically for the case of the prediction of material properties [**59, 99**]. Since experimental data are limited in materials science, ML models are mostly trained using DFT-computational datasets [**14, 59, 99, 13, 80, 81, 82**].

Some recent works compare the DFT-computed formation energies with experimental observations [**42, 142, 143**]. For instance, Kirklin et al. compared the DFT-computed formation energy with experimental measurements of $1,670$ materials and found that the mean absolute error (MAE) to vary from 0.096 to 0.136 eV/atom for OQMD [**42**]. Jain et al. [**143**] reports the MAE of the Materials Project as 0.172 eV/atom, while in Kirklin et al. [**42**], the MAE of the Materials Project is reported as 0.133 eV/atom. We also performed an analysis to compare the experimental formation energies of 463 materials against their corresponding formation energies from OQMD, the Materials Project and JARVIS datasets available in Matminer (an open-source materials data mining toolkit) [**1**]. A scatter plot of the comparison of different DFT-computed datasets against the experimental observations is illustrated in Figure 5.1. We find the MAEs in OQMD, Materials Project and JARVIS are 0.083 eV/atom, 0.078 eV/atom and 0.095 eV/atom respectively,

against experimental formation energies. In this paper, we will refer to this as the "discrepancy" between DFT computation and experiments, in order to distinguish it from the "error" of the ML-based predictive models built on top of DFT/experimental datasets. As DFT calculations are performed at 0 K and experimental formation energies are typically measured at room temperature, the two formation energies could be different [**42, 142**]. However, such a difference is very small except for the materials that undergo phase transformation between 0 K and 300 K; these elements include Ce, Na, Li, Ti and Sn [**144**]. DFT databases, such as OQMD and the Materials Project, reduce this systematic error by chemical potential fitting procedures for the constituent elements having phase transformations between 0 K and 300 K [**42**]. For instance, Kim et al. [**142**] performed a comparison between the experimental and the DFT-computed formation energy of such compounds containing constituent elements having phase transformation at low temperature, and reported an average discrepancy of about 0.1 eV/atom in both the Materials Project and OQMD; the average uncertainty of the experimental standard formation energy was one order of magnitude lower. Unlike OQMD and Materials Project, JARVIS does not apply any empirical corrections on formation energies to match experiments. As a consequence, such models trained on DFT-computed datasets automatically inherit the underlying discrepancies between the DFT-computations and the experimental observations, in addition to the prediction error with respect to DFT-computations used for training. The discrepancy between DFT-computation and experiments serves as the lower bound of the prediction errors that can be achieved by the ML models with respect to experiments. Due to this issue, potential material candidates identified by such

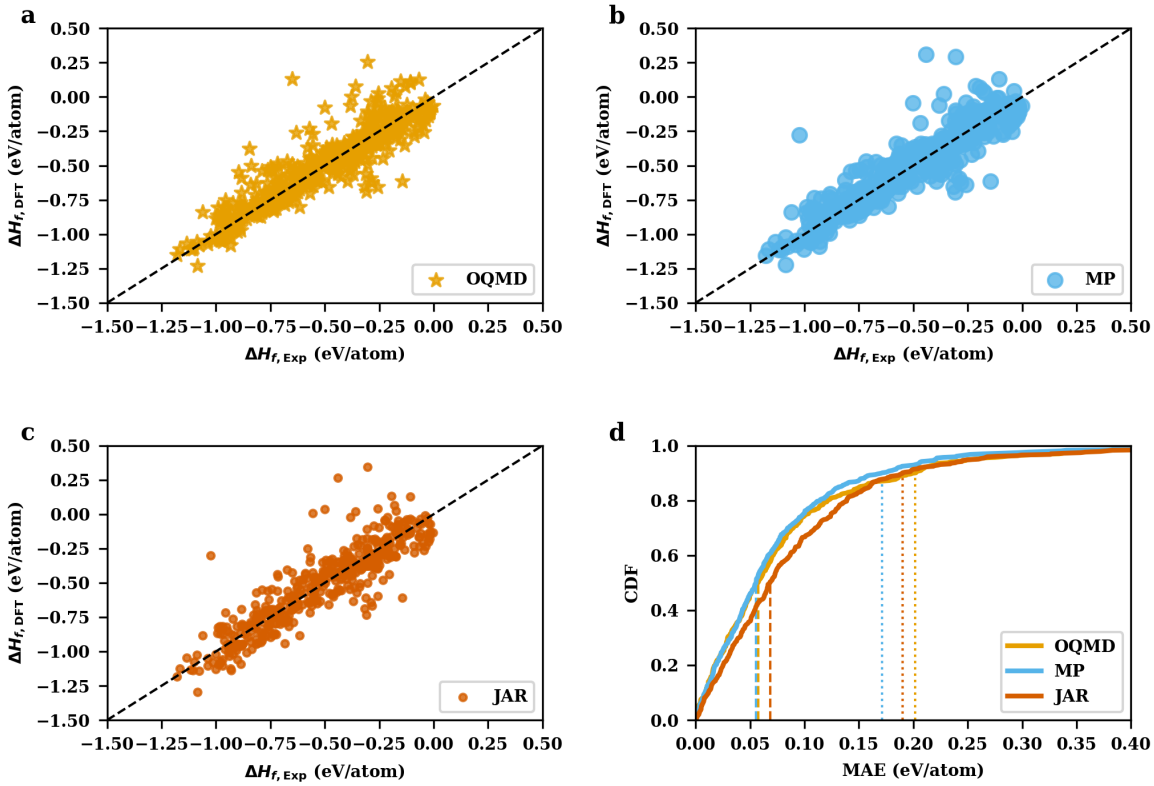


Figure 5.1. DFT-computation error analysis of different DFT-computed datasets against the experimental observations. We compared the experimental formation energies of 463 materials against their corresponding formation energies from OQMD (a), Materials Project (b) and JARVIS (c) datasets available in Matminer [**1**]. The MAE in OQMD, Materials Project and JARVIS for formation energies against experimental observations are 0.083 eV/atom, 0.078 eV/atom and 0.095 eV/atom respectively. (d) The 50th percentile and 90th percentile MAE for OQMD, Materials Project and JARVIS are 0.057 eV/atom and 0.201 eV/atom, 0.055 eV/atom and 0.171 eV/atom, and 0.068 eV/atom and 0.190 eV/atom, respectively.

ML screening could be incorrect and disagree with intuition from domain knowledge and experiments [**13, 99, 14**].

In this work, we demonstrate that it is possible to predict material properties closer to the true experimental observations using deep learning models that can leverage the

existing large DFT-computational datasets together with available experimental observations and other smaller DFT-computed datasets. Deep learning [**3**] enables us to perform transfer learning from large datasets to smaller datasets between similar domains. The transfer learning approach works by first training a deep neural network (DNN) model on the source domain with a large dataset and then, fine-tuning the trained model parameters by training on the target domain with a relatively smaller dataset as shown in Figure 5.2 [**145, 146**]. Since the model is first trained on a large dataset, it identifies a rich set of features from the input data representation, and this simplifies the task of learning features present in the smaller dataset, on which the model is subsequently fine-tuned. Specifically, here we evaluate the effectiveness of the proposed approach by revisiting a commonly-studied challenge in materials informatics: predicting whether a crystal structure will be stable (formation energy) given its composition [**13, 14, 83, 84, 85**]. We leverage the recent deep neural network architecture- ElemNet [**99**]; ElemNet enables us to perform transfer learning from OQMD (a large dataset containing DFT-computed materials properties for $\sim 341K$ materials) to two other DFT-databases (JARVIS and the Materials Project) and an experimental dataset containing 1,963 samples from the SGTE Solid SUBstance (SSUB) database. Our results demonstrate a significant benefit from the use of deep transfer learning; in particular, the proposed approach enables us to achieve an MAE of 0.06 eV/atom against an experimental dataset containing $1,963$ observations, which is significantly better than the mean absolute discrepancy of around 0.1 eV/atom of the DFT-computational datasets compared against experiments, and MAE of around 0.15 eV/atom of the predictive models trained from scratch (without using transfer learning) on either experimental dataset or DFT-computed datasets.

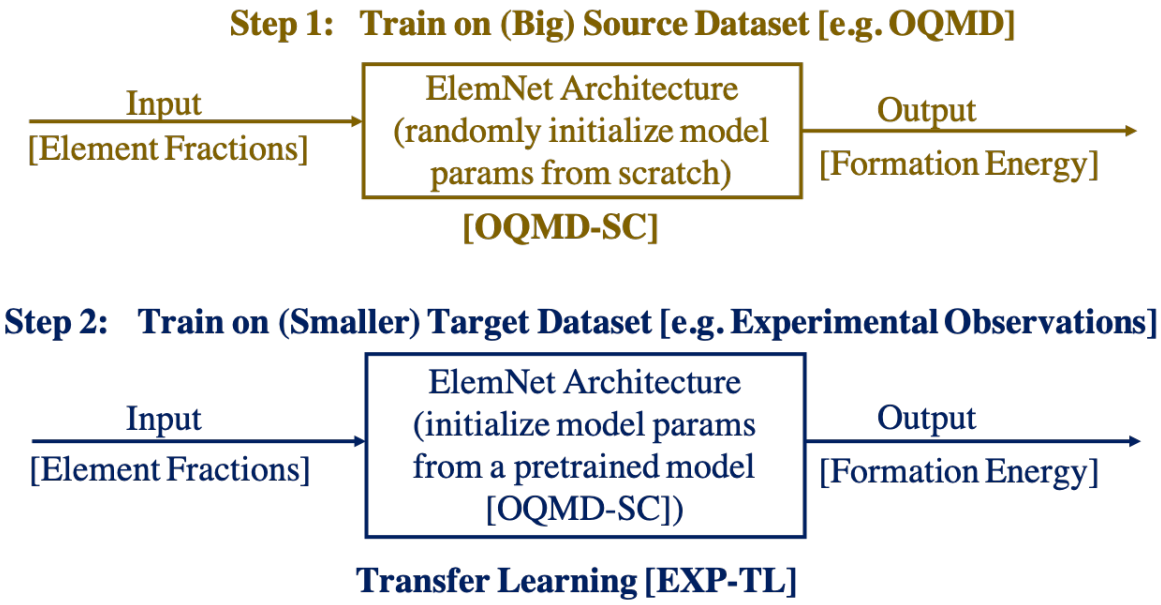


Figure 5.2. Proposed approach of deep transfer learning. First, a deep neural network architecture (ElemNet) is trained from scratch, by initializing model parameters randomly from a uniform distribution, on a big DFT-computed source dataset (OQMD). This allows the model to learn the input data representation and capture the essential chemistry from the big source training data. Since this model is trained from scratch on OQMD, we refer to this as OQMD-SC model. Next, we train a deep neural network architecture (ElemNet) on other smaller target dataset, such as experimental dataset, using transfer learning. Here, the model parameters are initialized using the values from OQMD-SC, and then fine-tuned using the corresponding target dataset.

5.2. Methods

5.2.1. Data Cleaning

The input data is composed of fixed size vectors containing raw elemental compositions as the input and formation enthalpy in eV/atom as the output labels. The input vector is composed of non-zero values for all the elements present in the compound and zero values for others; the composition fractions are normalized to one. We perform two stages of

data cleaning to remove single elements and outliers. The single elements are removed since their formation energy is zero. The samples with formation energy outside of $\pm 5\sigma$ (σ is the standard deviation in the training set) are removed. Further, the elements not appearing in the training datasets after cleaning are removed from the input attribute set. Out of 118 elements in the periodic table, our dataset contains the following 86 elements- [H, Li, Be, B, C, N, O, F, Na, Mg, Al, Si, P, S, Cl, K, Ca, Sc, Ti, V, Cr, Mn, Fe, Co, Ni, Cu, Zn, Ga, Ge, As, Se, Br, Kr, Rb, Sr, Y, Zr, Nb, Mo, Tc, Ru, Rh, Pd, Ag, Cd, In, Sn, Sb, Te, I, Xe, Cs, Ba, La, Ce, Pr, Nd, Pm, Sm, Eu, Gd, Tb, Dy, Ho, Er, Tm, Yb, Lu, Hf, Ta, W, Re, Os, Ir, Pt, Au, Hg, Tl, Pb, Bi, Ac, Th, Pa, U, Np, and Pu].

5.2.2. Experimental Settings and Tools Used

We have used the ElemNet [**99**] model architecture shown in Table 5.1 implemented using Python and TensorFlow [**93**] framework. ElemNet is a 17-layered fully connected deep neural network architecture that is designed to predict the formation energy from elemental fractions without any manual feature engineering [**99**]. The input for ElemNet is composed of a set of 86 elements in our dataset, from Hydrogen to Plutonium except for Helium, Neon, Argon, Polonium, Astatine, Radon, Francium and Radium. These 86 elements form the materials in most of the current DFT-computed datasets such as OQMD, JARVIS, and the Materials Project. ElemNet model is trained on each dataset with/without using transfer learning using ten-fold cross-validation except when training from scratch on OQMD; in the case of OQMD, ElemNet model is trained using a 9:1 random split into train and test(validation) sets, this is referred as OQMD-SC. OQMD-SC model is used for transfer learning in this work. We train for 1000 epochs with a

Table 5.1. *ElemNet* model architecture used for training different models.

Layer Types	**No. of units**	**Activation**	**Layer Positions**
Fully-connected Layer	1024	ReLU	First to 4th
Drop-out (0.8)	1024		After 4th
Fully-connected Layer	512	ReLU	5th to 7th
Drop-out (0.9)	512		After 7th
Fully-connected Layer	256	ReLU	8th to 10th
Drop-out (0.7)	256		After 10th
Fully-connected Layer	128	ReLU	11th to 13th
Drop-out (0.8)	128		After 13th
Fully-connected Layer	64	ReLU	14th to 15th
Fully-connected Layer	32	ReLU	16th
Fully-connected Layer	1	Linear	17th

learning rate of 0.0001 and mini batch size of 32 using Adam [**130**] optimizer. A patience of 200 minibatch iterations is used to avoid overfitting to the training dataset; if there is no improvement in validation error for 200 minibatch iterations, the training is stopped. Dropout [**86**] layers are leveraged to prevent overfitting and they are not counted as a separate layer. We used ReLU [**88**] as the activation function. We have used the Matplotlib library in Python to plot the figures used in this manuscript. All the models are trained and tested using Titan X GPUs on NVIDIA DIGITS DevBox. The training curves of the ElemNet models trained from scratch and using transfer learning on the experimental dataset are available in Supplementary Figure 3 of [**147**].

5.3. Experimental Results

5.3.1. Datasets

We use three datasets of DFT-computed properties: OQMD, the Materials Project and JARVIS, and one experimental dataset. Among other properties, these databases report the composition of material compounds along with their lowest formation energy in

eV/atom, hence identifying their most stable structure. OQMD contains composition and formation energies for $\sim 341K$ material compounds that can be either stable or unstable. We selected $11,050$ stable materials from JARVIS and $23,641$ stable materials from Materials Project. Note that the total number of materials in JARVIS and Materials Project is on the order of $30,000$ and $70,000$, respectively. However, for the present work, only materials present on the convex hull (energy above convex hull=0) were selected. In the case of material compounds with multiple crystal structures, the minimum formation energy for the given material composition is used since it represents the most stable crystal structure. For the experimental dataset, we use the experimental formation energy from the SGTE Solid SUBstance (SSUB) database; they are collected by international scientists [**148**] and contain a single value of the experimental formation enthalpy which should represent the average of formation enthalpy observed during multiple experiments, and do not contain error bars. It is curated and used by Kirklin et al. in their study of assessing the accuracy of DFT formation energies in OQMD [**42**]. It is composed of $1,963$ formation energies at 298.15 K, and contains many oxides, nitrides, hydrides, halides, and some intermetallics, all being stable compounds.

5.3.2. Training from Scratch

First, we discuss our results when training ElemNet model architecture on each dataset from scratch. While training from scratch, the model parameters are initialized randomly from a uniform distribution. Since the model parameters are initialized randomly, all the features are learned from the input training data. The input vector contains the elemental fractions normalized to one, and the regression output gives the formation energy. The

models learn to capture the required chemistry from the input training data. We report the results of a ten-fold cross-validation (except OQMD) performed on the four datasets in Table 6.1 (for OQMD, we used a 9:1 random split into train and test (validation) sets for this analysis, and the same model is used ten times to get the predictions on the test set since the model predictions changes for same input due to use of Dropout [**86**]). We also report performance of our models on a separate holdout test set using two different training:test set splits in Table 5.3. For holdout test, we split the datasets into training and test sets in the ratio of 9:1 and 8:2 and train ElemNet model architecture on the training sets using a ten-fold cross-validation, and report the performance of the best model from the ten-fold cross-validation on the holdout test set. Our results demonstrate that the size of training dataset has a significant impact on the model performance, which is in agreement with similar analyses from past studies [**99, 59**]. Despite the smaller training dataset size, the ElemNet model trained using the Materials Project has slightly better performance compared to the models trained using OQMD. This may be attributed to the inherent formation energy data in Materials Project for which several empirical fittings were applied. The impact of training dataset is most evident in the case of the experimental dataset, where the training data for each fold of the ten-fold cross validation contains only $\sim 1,767$ observations and each test set contains ~ 196 samples. The higher error in the case of the experimental dataset is due to its limited size and clearly illustrates the impact of the training data size on the performance of predictive models.

Table 5.2. Performance of the ElemNet models from ten-fold cross-validation in MAE (eV/atom).

Dataset	**Size**	**Scratch [SC]**	**OQMD-SC**	**Transfer Learning [TL]**
OQMD	$341,000$	0.0417 ± 0.0000	–	–
JARVIS	$11,050$	0.0546 ± 0.0019	0.0821 ± 0.0000	0.0311 ± 0.0012
Materials Project	$23,641$	0.0326 ± 0.0009	0.1084 ± 0.0000	0.0248 ± 0.0006
Experimental	$1,963$	0.1299 ± 0.0136	0.1354 ± 0.0000	0.0642 ± 0.0061

Table 5.3. Holdout test set performance of the ElemNet models in MAE (eV/atom).

Dataset	**Size**	**Train:Test Split Ratio**	**Scratch [SC]**	**Transfer Learning [TL]**
OQMD	$341,000$	$8:2$	0.0471	–
OQMD	$341,000$	$9:1$	0.0437	–
JARVIS	$11,050$	$8:2$	0.0593	0.0324
JARVIS	$11,050$	$9:1$	0.0568	0.0312
Materials Project	$23,641$	$8:2$	0.0347	0.0251
Materials Project	$23,641$	$9:1$	0.0327	0.0247
Experimental	$1,963$	$8:2$	0.1388	0.0660
Experimental	$1,963$	$9:1$	0.1460	0.0608

5.3.3. Prediction using OQMD-SC model

Since OQMD is the largest dataset used for training our models, we evaluated the ElemNet model trained on OQMD from scratch for making predictions on different datasets. We refer to this as the OQMD-SC. As shown in Table 6.1, we observe that although the OQMD-SC model has a low prediction error with an MAE of 0.0417 eV/atom against OQMD, it exhibits significantly higher error when evaluated against other datasets, regardless of whether they are DFT-computed or experimental. Although JARVIS, the Materials Project and OQMD are all DFT-computed datasets, they differ in their underlying approach for DFT-computations. Note that the OQMD-SC model is trained using only OQMD, our goal in this evaluation is to illustrate the underlying difference in different DFT datasets and the discrepancy between OQMD and the experimental observations. When the OQMD-SC model is evaluated against JARVIS and the Materials

Project, which are different from the training dataset OQMD, the underlying difference in DFT-computations between OQMD and the test datasets becomes obvious. This problem is exacerbated when the OQMD-SC model is evaluated on the experimental observations. Since the DFT-computations for the formation energy in the QOMD have a significant discrepancy (an MAE of around 0.1 eV/atom) against experimental observations, this adds up with the prediction error of the OQMD-SC model against the OQMD dataset itself. If we compare the prediction errors using the OQMD-SC model on different datasets against the error of the models trained from scratch on them, we find that prediction errors are in the same order of magnitude. The evaluation error for the Materials Project dataset using OQMD-SC model is three times greater compared to the ElemNet model trained from scratch using the Materials Project. Since the empirical shifts applied in the Materials Project are not performed for OQMD, the OQMD-SC model can not learn about them and performs poorly when evaluated on the Materials Project dataset (which is different from the training dataset - OQMD). Especially in the case of the experimental dataset, where the training sets in the ten-fold cross-validation contains only around 1770 compositions, the prediction error of the OQMD-SC model is very close to the model trained from scratch using the experimental dataset. Such observations suggest the research question of whether using an existing model trained on large DFT-computed datasets is better than using a prediction model trained from scratch on relatively smaller datasets such as ones from experimental observations containing $\sim$ 1000s samples.

5.3.4. Impact of Transfer Learning

Since the prediction error of both the model trained from scratch on the experimental dataset and the OQMD-SC model (which is trained from scratch on largest DFT-comptued dataset- OQMD) against the experimental observations is poor, we decided to leverage the concept of deep transfer learning as it enables to transfer the feature representations learned for a particular predictive modeling task from a big source dataset to other smaller target datasets in similar domains. For the task of transfer learning, we chose the OQMD-SC model which is trained from scratch on OQMD using a 9:1 random split for training and the test (validation) sets. We chose the OQMD-SC model due to two reasons. First, OQMD-SC model is trained on OQMD, which is the largest dataset in our study- containing around $341K$ samples. Second, the OQMD-SC model learns the required physical and chemical interactions and similarities between different elements better than other models trained from scratch, which is again due to the large dataset used for training (more on this later). The use of transfer learning helps us in leveraging these chemical and physical interactions and similarities between elements learned by the OQMD-SC model in training models for the other relatively smaller datasets. Unlike in the case of training from scratch, where the model parameters are initialized randomly, here the model parameters are initialized using the ones from the OQMD-SC model. Next, they are fine-tuned during the new training process, to learn the data representation from the smaller target dataset.

We find that the prediction error significantly drops after using transfer learning from OQMD-SC model. As seen in Table 6.1 and Table 5.3, the prediction error for the experimental data model almost halves. Interestingly, the error of the model trained

using transfer learning from OQMD-SC model on JARVIS and the Materials Project achieves even smaller error than that of the prediction error of the OQMD-SC model itself against the OQMD dataset. Since the JARVIS and Materials Project datasets are larger than the experimental dataset, we observe better performance for JARVIS and Materials Project. The use of transfer learning is very effective in the case of the models trained using experimental observations. We find that the use of transfer learning from the OQMD-SC model moves the predictions closer to the true experimental observations. The prediction error of the model trained on the experimental dataset using transfer learning from OQMD-SC model is also comparable to the prediction error of the OQMD-SC model itself against the OQMD dataset. We expect the benefit of using deep transfer learning to improve with the increase in the availability of experimental observations for fine-tuning (as discussed next). We believe that an MAE of 0.06 eV/atom by a prediction model against experimental observations is a remarkable feat since this is comparable to and slightly better than the existing discrepancy of DFT computations themselves against experimental observations [**42**].

5.3.5. Impact of Training Data Size on Transfer Learning

The success of deep learning in many applications is mostly attributed to the availability of large training datasets, which has discouraged many researchers in the scientific community having access to only small datasets from leveraging deep learning in their research. In our previous work [**99**], we demonstrated how deep learning can be used even with small datasets (in the order of 1000s) to build more robust predictive models than the ones using traditional machine learning approaches like Random Forest. Here,

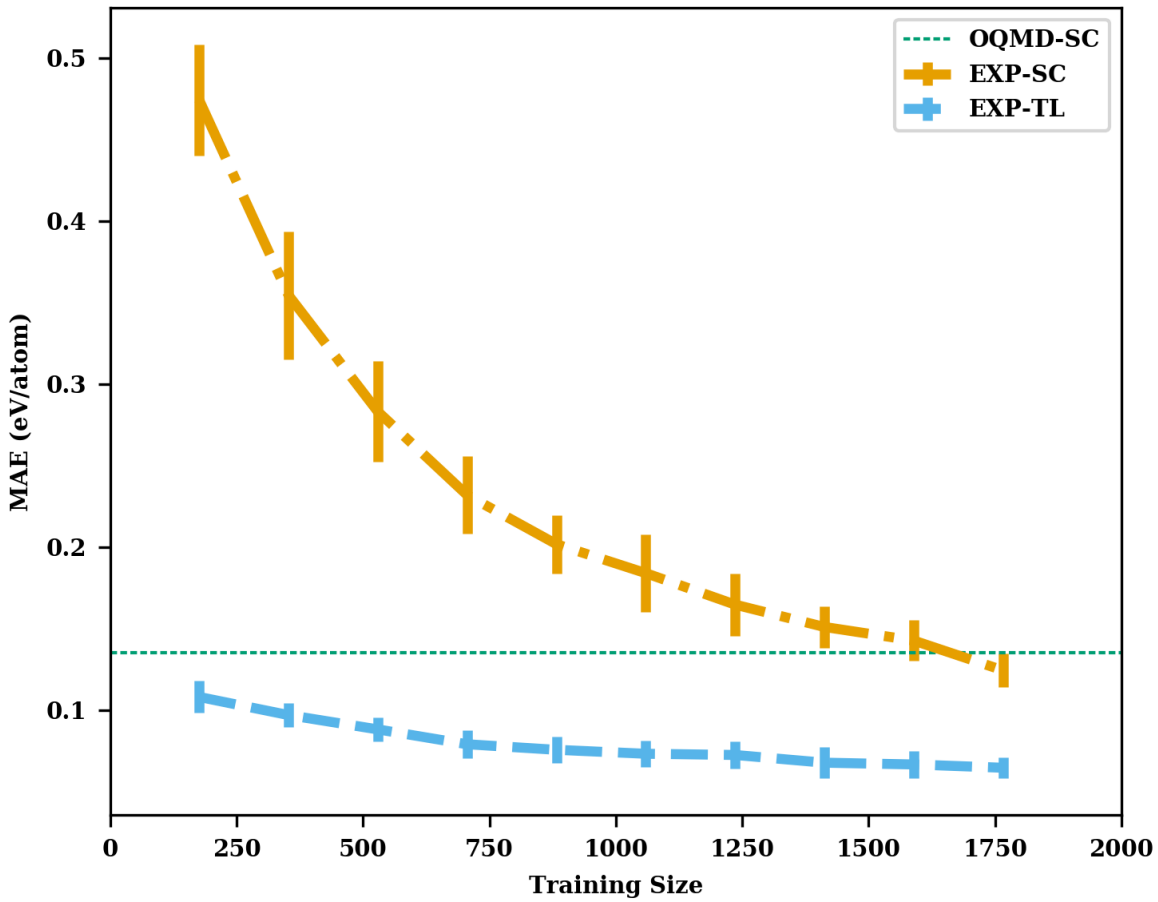


Figure 5.3. Impact of training data size on performance of models trained from scratch and using transfer learning (mean and s.d.). The models are trained on the experimental dataset and the results are aggregated from a ten-fold cross-validation. For each cross validation, first we split the complete dataset randomly into training and test (validation) set in the ratio of 9 : 1. Next, we fixed the test (validation) set and changed the size of the training set from 10% to 100%. OQMD-SC represents the model trained from scratch on OQMD dataset, EXP-SC represents the prediction error of the model trained from scratch, and EXP-TL represents the prediction error using transfer learning from the OQMD-SC model.

we demonstrate how transfer learning can be leveraged even if the target dataset is very small (in the order of 100s). We demonstrate this for the experimental dataset by fixing

the test (validation) set and changing the size of the training dataset from 10% to 100% with an increment of 10%, for each fold in the ten-fold cross-validation. We trained the ElemNet model from scratch - EXP-SC, and also using transfer learning from OQMD-SC model - EXP-TL, on training data with varying size, as illustrated in Figure 5.3. For EXP-SC, we observe a large impact of the training dataset size as the MAE decreased from 0.474 eV/atom to 0.124 eV/atom as the training data size increased from 10% to 100%. However, the impact of training dataset size is significantly lower in the case of transfer learning in the case of EXP-TL; the MAE changes gradually from 0.108 eV/atom to 0.064 eV/atom, as the training data size changes from 10% to 100%. This illustrates that the proposed approach of deep transfer learning can be leveraged even in the case of significantly smaller datasets having around $100s$ of samples for fine-tuning provided there exists a bigger source dataset for transfer learning.

5.3.6. Prediction Error Analysis

Next, we analyzed the distribution of prediction error of all ElemNet models: the model trained from scratch (denoted by EXP-SC, JAR-SC, MP-SC), and the model trained using transfer learning from OQMD-SC model (denoted by EXP-TL, JAR-TL, MP-TL). Figure 5.6 illustrates the scatter plot and CDF of the ElemNet models trained from scratch and using transfer learning on different datasets; they contain the test predictions gathered using ten-fold cross validation in different cases. We find that the use of transfer learning leads to significant improvement in the prediction of formation energy; the predicted values move closer to the DFT-computed or the experimental values. The benefit of the use of transfer learning is most significant in the case of experimental data; the predicted

formation energies are mostly concentrated along the diagonal (hence, closer to the values from actual experimental observations). A glimpse of the CDF of the model trained using experimental data shows the same benefit in terms of percentiles; both the 50th and 90th percentiles of prediction error reduced by almost half. We observe a similar trend in case of JARVIS and Materials Project; although the distributions look similar, there is a clear reduction in prediction error as predicted values become more concentrated along the diagonal of the scatter plot in both cases. The third row in Figure 5.6 illustrates the scatter plot and cumulative distribution function (CDF) of the OQMD-SC model against a test set containing $34,145$ materials from the OQMD. Although the scatter plot appears to have a widespread in the prediction error, most of the predictions are very close to the diagonal. This is evident from the CDF plot, which illustrates that the 50th percentile error is around 0.015 eV/atom and the 90th percentile error is around 0.08 eV/atom. Hence, the OQMD-SC model predicts the formation energy of most of the compounds with high precision when compared against OQMD itself. However, OQMD-SC model has significantly worse error distribution when compared against other three datasets - broader spread in the scattter plot and lower slopes for the CDF curves (Supplementary Figure 1 of [**147**]), which illustrates that although the OQMD-SC model is trained on the big DFT-computed OQMD dataset, it does not always make robust predictions against datasets computed/collected using other techniques. A thorough analysis of the input elements present in the set of compounds having more than 98th percentile error is available in the Supplementary Discussion of [**147**].

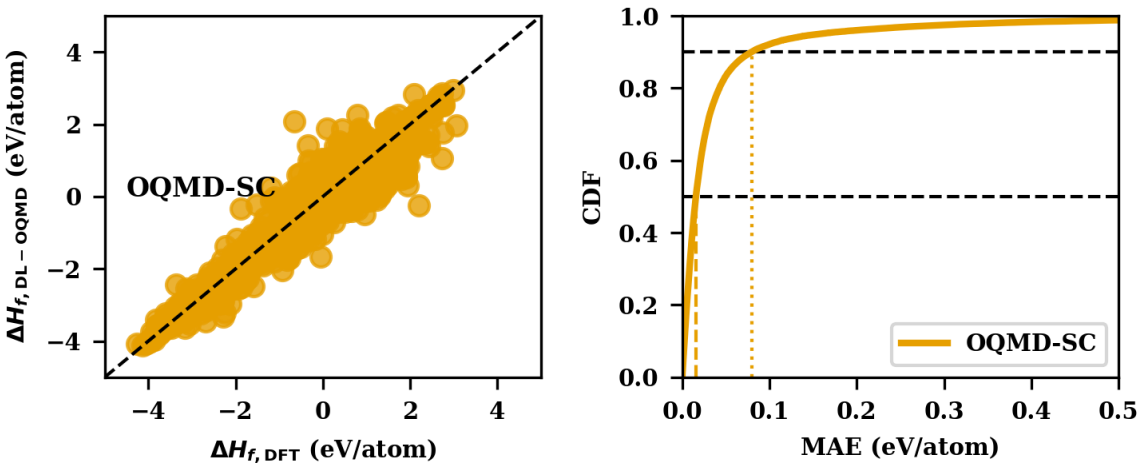


Figure 5.4. Prediction error analysis of OQMD-SC model using a test set containing 34,145 samples from a 9:1 random split of OQMD. OQMD-SC model is trained from scratch (with random weight initialization from a uniform distribution) using a 9:1 random split of training and test set from the OQMD. Since the dataset is large, the model is able to automatically capture the essential chemical and physical interactions between different elements; hence, providing robust predictions while compared against OQMD.

5.3.7. Performance on Experimental Data

Next, we analyze the performance of the prediction models trained on different DFT-computed datasets (both trained from scratch and with transfer learning from the OQMD-SC model), by evaluating their performance on the experimental observations containing 1,963 samples. The performance of different models on the experimental dataset is shown in Table 5.4. For models trained on experimental data, we report the performance on test (validation) sets from the ten-fold cross-validation. For JARVIS and the Materials Project, we report the mean and standard deviation of the predictions using ten different models from the ten-fold cross-validation. For OQMD, we use one OQMD-SC model ten

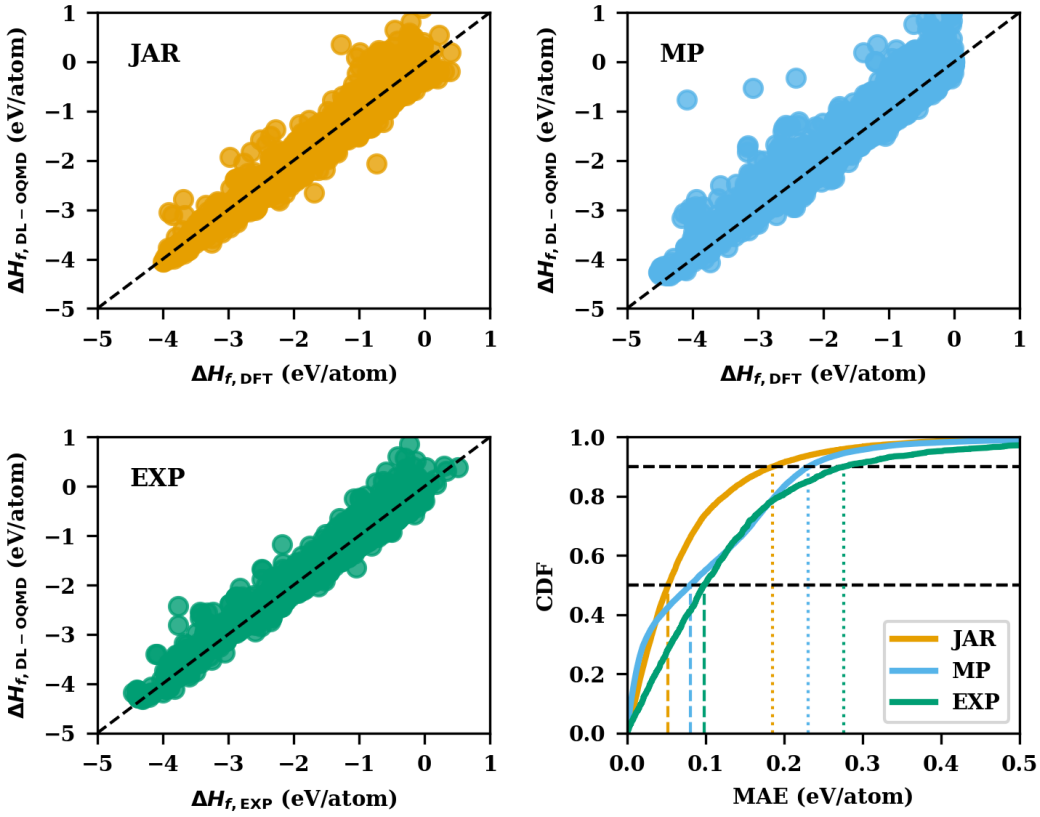


Figure 5.5. Prediction error analysis using OQMD-SC model on the other three datasets. The OQMD-SC model is trained from scratch (with random weight initialization from a uniform distribution) using a 9:1 random split of training and test set from the OQMD. Although OQMD-SC model has low prediction error against the test set from OQMD, the prediction error is high if we compare against other datasets. This is because of the difference in the DFT-computations used in JARIVS and Materials Project, and OQMD. Since DFT-computations from the OQMD has an error of around 0.1 eV/atom against experimental observations, this error is inherent in the OQMD-SC model leading to higher prediction errors.

times since use of Dropout [**86**] results in different predictions for same input. As we can observe from these results, the performance of all the models trained on DFT-computed datasets is significantly worse compared to their performance against unseen test sets from the dataset on which they are trained (Table 6.1). There is a minor impact of the use of

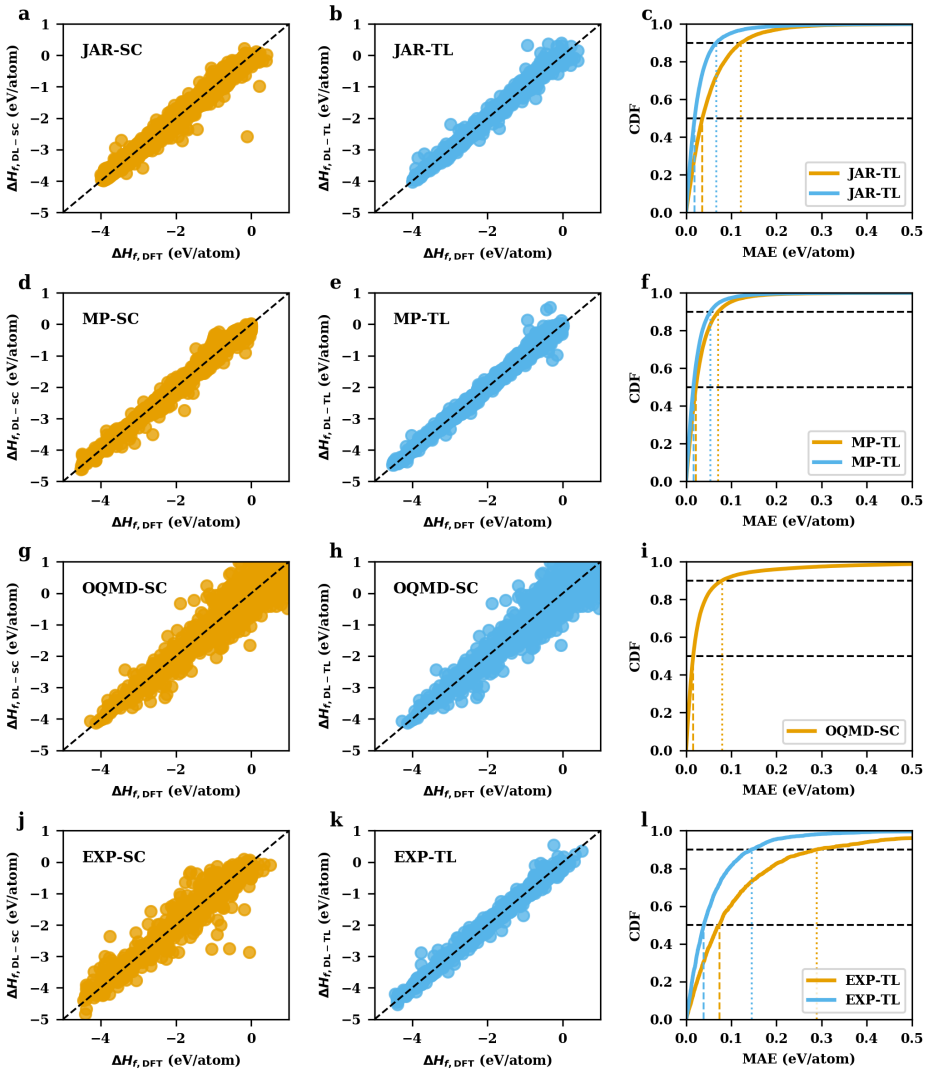


Figure 5.6. Prediction error analysis on the test (validation) sets from the ten-fold cross validation (except for OQMD-SC). For OQMD-SC, ElemNet model is trained from scratch using a 9:1 random split of training and test (validation) set from the OQMD. For other datasets, we aggregate the prediction errors on the test (validation) sets from the ten-fold cross-validation for each model. The four rows represent the four datasets- (a-c) JARVIS (JAR), (d-f) Materials Project (MP), (g-i) OQMD and (j-l) the experimental observations (EXP); first (a, d, g and j) and second (b, e, h and k) columns of each row show the predictions using the model trained on the particular dataset from scratch (SC) and using transfer learning (TL), respectively, the third column (c, f, i and l) shows the respective CDF of the prediction errors using models trained from scratch (SC) and using transfer learning (TL), trained on the particular dataset.

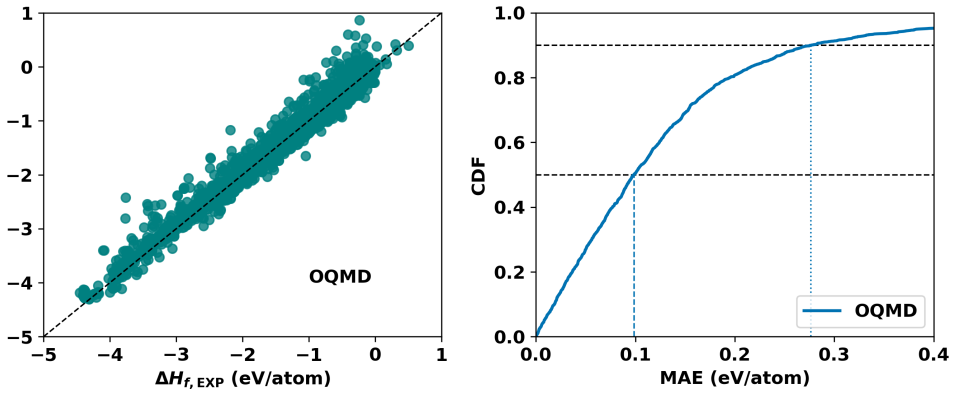


Figure 5.7. Prediction error analysis for the ElemNet architecture trained using OQMD and evaluated on the experimental data containing 1,963 observations. When training from scratch, the weights are initialized randomly from a normal distribution; for transfer learning, the model is first trained on the OQMD dataset and then fine-tuned using the corresponding dataset.

transfer learning for the models trained on the JARVIS and Materials Project dataset. Among all the models trained using DFT-computed datasets, the OQMD-SC model has the lowest discrepancy which is comparable to the prediction error of model trained on experimental dataset from scratch. The performance of OQMD-SC model re-emphasizes the impact of training data size which enables the model to automatically capture the physical and chemical interactions from the input data representation that is essential for making correct predictions. The error in predictions using different models are at least double than that of the model trained on the experimental dataset using transfer learning from the OQMD-SC model. Our observations demonstrate the need to leverage DFT-computed datasets with experimental datasets to build robust prediction models which can make predictions closer to true experimental observations, thereby questioning

and providing an alternative to the current practice of using predictive models built using DFT-computed datasets alone.

Table 5.4. Performance of ElemNet models on the experimental data in MAE (eV/atom).

Training Dataset	**Test Dataset**	**Scratch [SC]**	**Transfer Learning [TL]**
OQMD	Experimental	0.1354 ± 0.0000	–
JARVIS	Experimental	0.1911 ± 0.0042	0.1487 ± 0.0027
Materials Project	Experimental	0.1619 ± 0.0020	0.1613 ± 0.0016
Experimental	Experimental	0.1299 ± 0.0136	0.0642 ± 0.0061

Figure 5.8 illustrates the scatter plot of the predicted values against the true experimental values and CDF of the corresponding errors. If we look at the prediction results using the OQMD-SC model in Figure 5.8, the predictions are less concentrated on the diagonal of the scatter plot; the 50th percentile error is 0.1 eV/atom and the 90th percentile error is 0.28 eV/atom. This is significantly worse than the test error of OQMD-SC model on OQMD itself (MAE of 0.04 eV/atom in Table 6.1) and the discrepancy of the DFT-computations for OQMD against experimental values (0.1 eV/atom [**42**]). This illustrates the high deviation of the OQMD-SC model in the predicted values against the true experimental observations. The improvement due to transfer learning in the prediction error distribution is negligible for the models trained using JARVIS and Materials Project datasets. This again illustrates the inefficacy of using a model trained using DFT-computed datasets alone, since they will have high prediction error against experimental observations due to the inherent discrepancy of the DFT-computation itself against experimental observations.

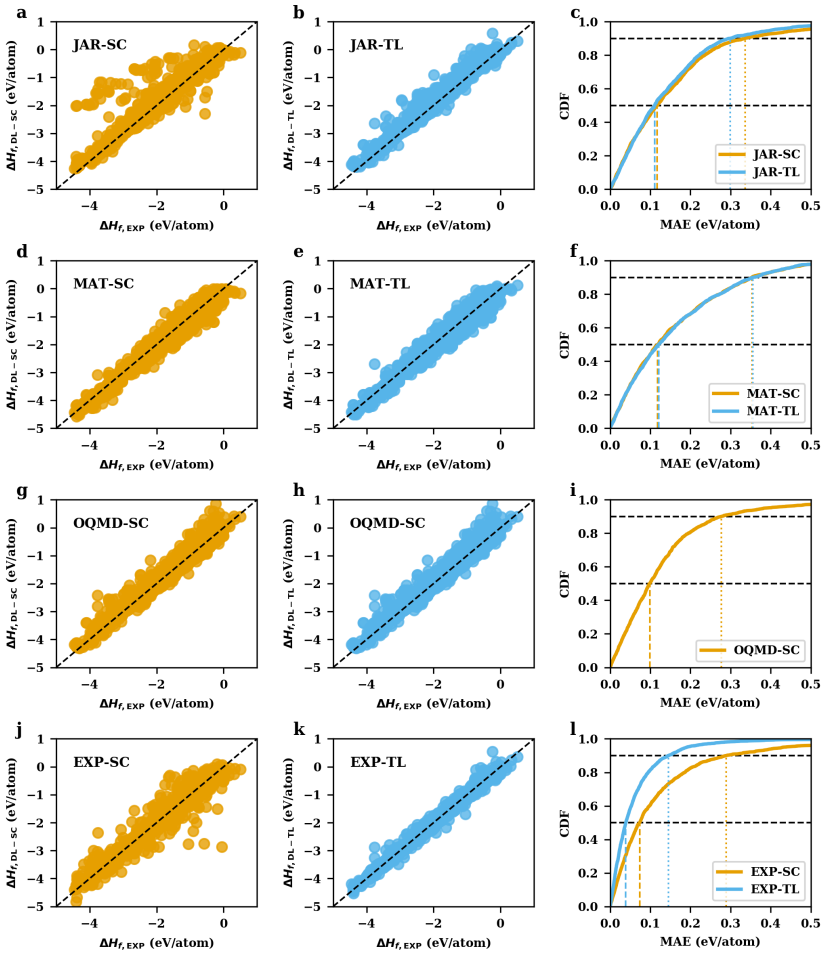


Figure 5.8. Prediction error analysis on the experimental dataset containing 1,963 observations using different models. For the models trained using experimental dataset, the predictions on the test sets are aggregated from validation sets using ten-fold cross-validation. For the models trained using JARVIS and Materials Project, since we have ten models from the ten-fold cross-validation during training, we take the mean of their prediction for each data point in the experimental dataset. For OQMD-SC, we make ten predictions on each point in the experimental dataset and take the mean. The four rows represent the four datasets- (a-c) JARVIS (JAR), (d-f) Materials Project (MP), (g-i) OQMD and (j-l) the experimental observations (EXP); first (a, d, g and j) and second (b, e, h and k) columns of each row show the predictions using the model trained on the particular dataset from scratch (SC) and using transfer learning (TL), respectively, the third column (c, f, i and l) shows the respective CDF of the prediction errors using models trained from scratch (SC) and using transfer learning (TL), trained on the particular dataset.

5.3.8. Activation Analysis

Next, to understand the impact of transfer learning on the performance of models trained using different datasets, we analyzed the activations from different layers of ElemNet architecture to visualize the physical and chemical interactions and similarities captured by the model. We performed two kinds of analysis for two different classification tasks using two different datasets. The first analysis involved taking the activations from each layer of different models and apply principal component analysis (PCA) for dimensionality reduction; since the number of activations varies from 1024 in the first hidden layer to 32 in the penultimate layer, we use PCA to get first two principal components and scale them in the range of [0,1] for ease of visualization using a scatter plot. The second analysis involved taking the activations from each hidden layer without applying PCA and training a Logistic Regression for classification using a random split of training and test set in the ratio of 9:1. We analyze the activations to see how well they can be used to perform three classification tasks- magnetic vs non-magnetic (1 vs 0) from JARVIS, insulator vs metallic (1 vs 0) from JARVIS, and insulator vs metallic (1 vs 0) from Materials Project.

Figure 5.9 demonstrates the scatter plot and ROC (Receiver Operating Characteristics) curves of the Logistic Regression model trained using activations from the first hidden layer of the ElemNet model trained from scratch and using transfer learning on different datasets. Logistic Regression is a statistical model based on using a logistic function to model the binary dependent variable for binary classification problems [**149, 150**]. A ROC curve is generated by plotting the true positive rate (TPR) against the false positive rate (FPR) at a varying threshold, and the area under the curve (AUC) of a ROC curve represents the performance measurement for the binary classification problem [**151**]

Higher the AUC of a ROC, better is the model at distinguishing between the binary classes. The distinction of magnetic vs non-magnetic materials is evident from the visualization using the scatter plot of the first two PCA components of the activations of the same hidden layer in Figure 5.9. In the case of the OQMD-SC model, we find that the distinction between the two classes is more distinguished which agrees with the fact that the ElemNet model trained on OQMD dataset captures the physical and chemical interactions between different elements automatically [**99**]. From the scatter plot of the first two components of the PCA analysis, we find that other than the OQMD-SC model, other models trained from scratch hardly capture the distinction between magnetic and non magnetic class (1 vs 0) from the training dataset due to their relatively small size used for training (first row of Figure 5.9). When using transfer learning, we find that this ability to distinguish between magnetic and non-magnetic is passed to the fine-tuned models, thereby enhancing the prediction performance of the models trained using transfer learning from the ElemNet-QOMD model. Although there is no clear boundary between the magnetic vs non-magnetic materials in the scatter plot, the magnetic materials are concentrated towards the lower part of the scatter plot for the models trained using transfer learning.

This enhancement in the ability to distinguish between magnetic and non-magnetic materials becomes more evident if we look at the ROC curve of the Logistic Regression model trained using the actual activations from the same layer. As shown in Figure 5.9, the Logistic Regression models trained using activations from the model trained using transfer learning from OQMD-SC model exhibit a significant difference in the AUC of the ROC curve - 0.97 compared to that of around 0.93 using the activations from the model

trained from scratch (except the OQMD-SC model). We observe a similar impact on the classification task to distinguish magnetic and non-magnetic materials for activations up to the first six layers. Further, we observed similar results for insulator vs metallic class for different datasets, and the analysis for JARVIS dataset is available in the Supplementary Figure 2 of [**147**]. We also performed this task on activations of different layers for the dataset from the Materials Project, and observed similar results. An interesting observation is that although the activation plots of all the different models trained from scratch look distinct, they look almost similar after the use of transfer learning from the OQMD-SC model. This illustrates that the knowledge of chemical and physical interactions and similarities between different elements transferred from the OQMD-SC model dominates even after the models are fine-tuned using the target datasets; this is because data representation learned from OQMD is very rich compared to the limited representation present in the relatively smaller training datasets from JARVIS, the Materials Project and the experimental observations.

5.4. Discussion

In this work, we demonstrated the benefit of leveraging both DFT-computations and experimental observations to build more robust prediction models whose predictions are closer to the experimental observations compared to the predictive models built using only DFT-computed datasets. Since we already illustrated how ElemNet can automatically capture the underlying chemistry from only elemental fractions using artificial intelligence (deep learning) and perform better than the traditional machine learning approach in our previous work [**99**], here we focused on using the deep neural network architecture

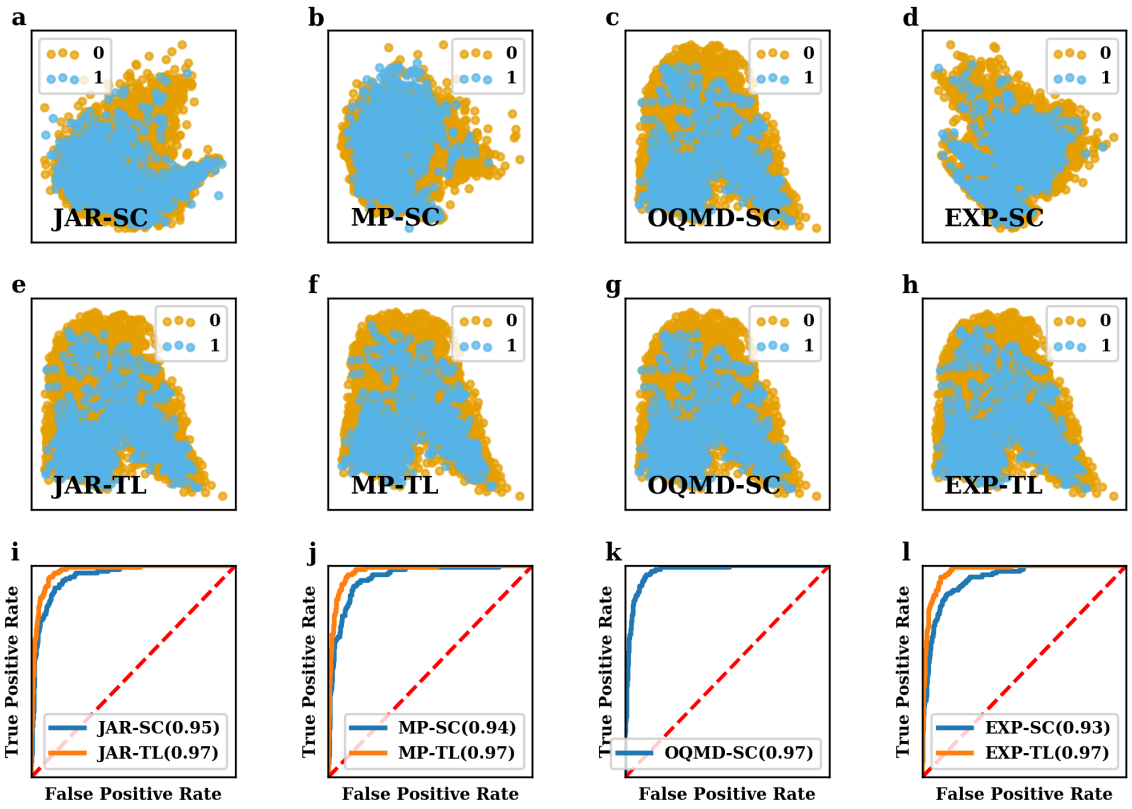


Figure 5.9. Analysis of the activations from the first hidden layer of the ElemNet architecture for the magnetic vs non-magnetic class (1 and 0) from JARVIS dataset. The four columns represent the models trained using four different datasets- (a, e and i) JARVIS (JAR), (b, f and j) Materials Project (MP), (c, g and k) OQMD and (d, h and l) the experimental observations (EXP); the first (a-d) and second (e-h) rows represent the models trained from scratch (SC) and using transfer learning (TL), while the third row (i-l) represents the ROC curves from the Logistic Regression model trained using all activations from the same hidden layer (the corresponding AUC values are shown in brackets) on the respective datasets. The scatter plots demonstrate the first two principal components of the activations using principal component analysis (PCA) technique.

of ElemNet for deep transfer learning of the chemistry learned from large datasets to smaller datasets using DFT or experimental observations; the comparison of ElemNet

against traditional machine learning approaches for all datasets is available in the Supplementary Table 1 of [**147**]. Our analysis of the prediction models based on different DFT-computed and experimental datasets illuminates the fundamental problem of building prediction models using DFT-computed datasets. Prediction models built using only the DFT-computed values exhibit high prediction errors against the experimental values; this results from the inherent discrepancy of DFT-computations against the experimental observations themselves, in addition to the error of the model against the DFT-computed values used for its training. We expect the proposed approach to perform better with the increasing availability of DFT-computations (for source dataset) as well as an increase in the experimental observations for fine-tuning.

We have shown the application of deep transfer learning in predicting formation energy of materials (and hence, the stability of materials) such that they are closer to experimental observations, which in turn, can be used for performing more robust combinatorial screening for hypothetical materials candidates for new materials discovery and design [**13, 99**]. Formation energy is an extremely important material property since it is required to predict compound stability, generate phase diagrams, calculate reaction enthalpies and voltages, and determine many other important properties. Note that while formation energy is so ubiquitous, DFT calculations allow prediction of many other properties (such as band gap energy, volume, energy above the convex hull, elasticity, magnetization moment), which are very expensive to measure experimentally. The presented approach can be leveraged for predicting many other such materials properties where we have large computational datasets (such as using DFT), but small ground truth (experimental observations), a scenario which is very common in materials science; some

examples being predicting bandgap energies of certain classes of crystals [**152, 14, 153**], thermal conductivity, thermal expansion coefficients, Seebeck coefficient of thermal compounds [**154, 155**], mechanical properties of metal alloys [**152, 116**], magnetic properties of materials [**53**], and so on, for various types of applications in materials design. DFT databases are in the order of 10^4, however, the computationally hypothetical materials are in the order of 10^{10}, that is where machine learning models can be extremely valuable for the pre-screening process [**13, 99**]. As long as the source dataset for transfer learning contains a diverse range of chemistry and the target dataset contains compounds having similar chemistry (a subset of elements or features present in the source dataset for transfer learning), we expect the proposed method to work well. The presented approach can also be leveraged for building more robust predictive systems for other scientific domains where the amount of experimental observations and ground truth is not sufficient to train a machine learning model on its own, but there exists a large set of computational/simulation dataset from the same domain for transfer learning.

CHAPTER 6

Extracting Grain Orientation from EBSD Patterns of Polycrystalline Materials using Convolutional Neural Networks

6.1. Introduction

Engineering materials are usually crystalline, most often in polycrystalline form. They consist of multiple "grains" having different crystallographic orientations. While the microscopic properties of these grains are anisotropic due to their crystalline nature, the macroscopic properties of the whole crystal depend on the material's texture - the relative fractions of each of these grain orientations. Texture also provides information about the thermo-mechanical processing history of materials and can be used to reconstruct the conditions leading to the micro-structure, for example, in geological rocks. Thus, texture is paramount in understanding the processing-structure-property relationships.

Since its development in the early 1990s, automated Electron Backscatter Diffraction (EBSD) has become the primary tool to determine the crystal orientation of crystalline materials across a wide variety of material classes ([**156**]). The technique provides quantitative information about the grain size, grain boundary character, grain orientation, texture and phase identity of the sample by measuring the angular distribution of backscattered electrons using a combination of a scintillator screen and a charge coupled device (CCD) camera. The schematic of the EBSD setup is shown in Figure 6.1. The sample sits at a tilt of σ (typically 70°) with the camera tilted at angle θ_c (typically

$0° - 10°$). Electrons travel down from the pole piece and interacts with the specimen at point **O**. The backscattered yield is measured as a function of direction by the scintillator. A physics-based model can be used to predict the backscattered yield based on the

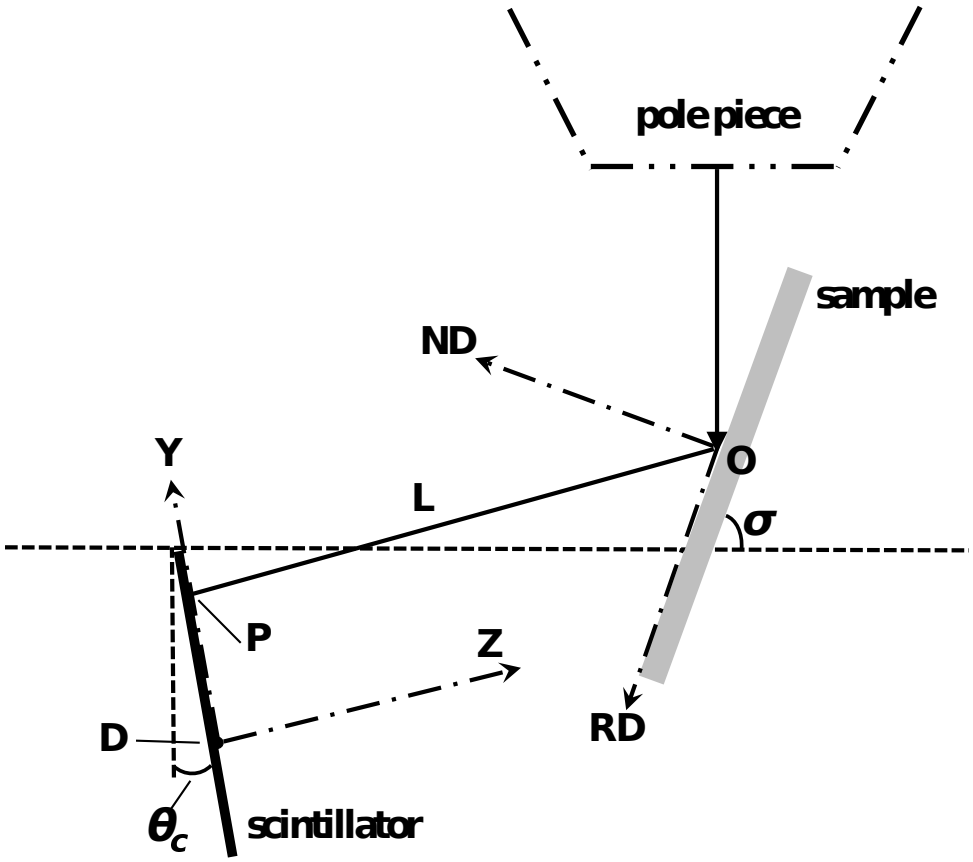


Figure 6.1. Schematic of the EBSD geometry.

principles of quantum mechanics. Assuming the microscope is parametrized by $\mathcal{M}$, the geometry of the setup is denoted by $\mathcal{G}$, and the crystal under investigation is parametrized by $\mathcal{C}$, the forward model is given by

$$\mathcal{F} \equiv \mathcal{F}(\mathcal{M}, \mathcal{G}, \mathcal{C}) . \tag{6.1}$$

Further details of this model can be found in ([**157**]). An example experimental EBSD pattern of Iron with its corresponding physics-based simulation is shown in Figure 6.2(a)-(b) respectively.

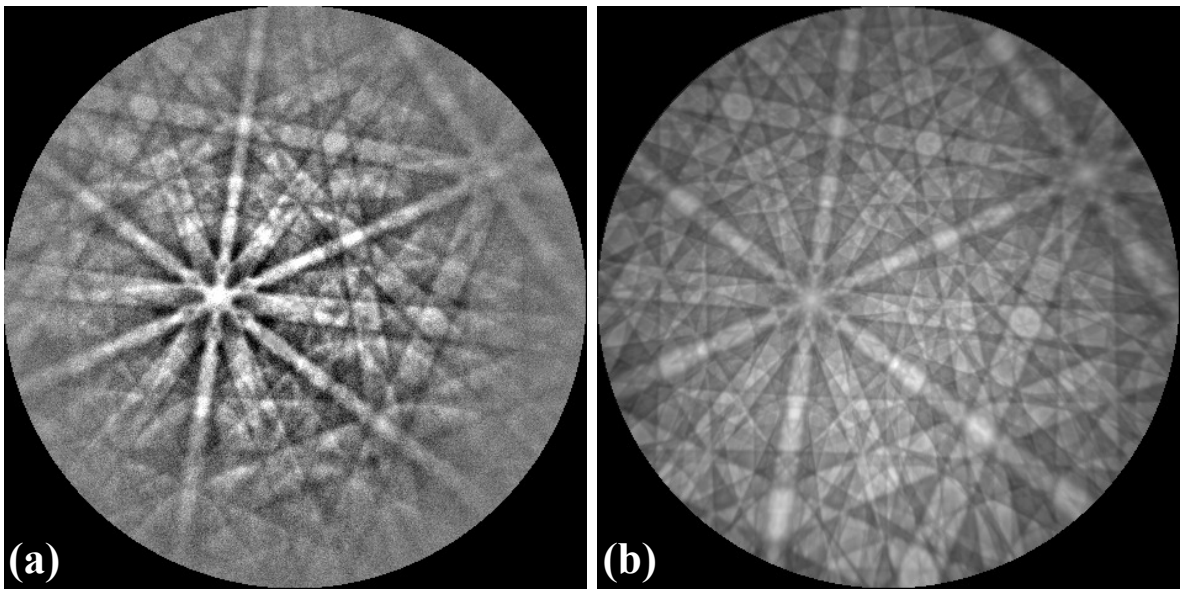


Figure 6.2. EBSD pattern from Iron (a) Experimental and (b) Simulation.

There are two major techniques to indexing EBSD patterns, each with its advantages and drawbacks. These include the commercially available Hough-transform based approach ([**158**]) and the newly developed Dictionary Indexing method ([**159, 160, 161, 162**]). The commercially available solution to the indexing problem uses a feature detection algorithm ([**163**]). A Hough transform of the diffraction is performed to identify linear features (Kikuchi bands). The angles between the extracted linear features are compared to a pre-computed look up table to determine the crystal orientation. This method has been very successful in indexing EBSD patterns and has led to significant

advances in materials characterization. However, the performance of this method quickly deteriorates in the presence of noise ([**160, 164**]).

In essence, dictionary based indexing is a nearest neighbor search approach in which the output angles correspond to the orientation angles of the closest EBSD pattern present in dictionary. The distance function used is a dot product as follows:

$$d(\vec{x_1}, \vec{x_2}) = 1 - \frac{\vec{x_1} \cdot \vec{x_2}}{|\vec{x_1}||\vec{x_2}|}, \tag{6.2}$$

where $\vec{x}$ is a vector representing the pixels in the EBSD patterns. The dot product between the pixel intensities in the test sample and each sample in the simulation-based dictionary set is computed and the nearest training sample is used to make the prediction. This method has been shown to be very robust to noise in the diffraction pattern and outperforms the line feature based Hough transform method ([**164**]) for a wide variety of crystal classes. However, this approach is computationally very expensive, which limits the technique to be an off-line method, and a real time solution to the indexing problem is currently not possible using this approach.

In the present paper, we present a deep learning ([**3**]) based model, trained using a simulated diffraction dataset, to predict the crystal orientations for experimental EBSD patterns, such that they have a minimum "disorientation" with respect to their ground truth. Deep learning leverages deep neural networks composed of multiple processing layers to automatically learn the representations of data with multiple levels of abstraction ([**3**]). They have achieved great success in the field of computer science with state-of-the-art results in computer vision ([**4, 73**]), speech recognition ([**74, 75**]) and text processing ([**76**]),

and are increasingly being used in the relatively nascent field of materials informatics ([**9**]) for deciphering processing-structure-property relationships.

Convolutional Neural Network (CNN) is a type of artificial neural network which is composed of convolution layers ([**165**]) in addition to fully connected layers. Since they require minimal preprocessing, they have gained significant attention in fields like computer vision ([**4, 73**]), recommender system ([**166**]) and natural language processing ([**167**]). Recently, CNNs have been applied for building models from microstructural data and improving characterization methods, ([**77, 78, 79**]) and they have been shown to be useful for predicting properties of crystal structures and molecules ([**80, 81**]), detecting cracks in materials/infrastructure images ([**168**]), and so on. [**169**] used CNNs for the classification of X-ray diffraction (XRD) patterns in terms of crystal system, extinction group and space group using a large dataset of 150, 000 XRD patterns, without any manual feature engineering. [**170**] developed CNNs to automatically analyze position averaged convergent beam electron diffraction patterns to extract pattern size, center, rotation, specimen thickness, and specimen tilt, without any need for pretreating the data. [**171**] applied CNNs to learn crystal orientations from simulated EBSD patterns; they built three separate CNN models to individually predict the three Euler angles, but did not take into account the mean disorientation between the predicted and true crystal orientations. Moreoever, their models were not tested on experimental data. In this study, our goal is to learn to predict the crystal orientation of experimental EBSD patterns such that they have minimum disorientation with the ground truth.

Building a predictive model for the indexing of EBSD patterns poses two significant challenges. First, we need to minimize the disorientation between the predicted and the

ground truth crystal orientations; this requires optimizing for the mean disorientation error which is metric for a highly non-linear orientation space. Furthermore, this cost function is computationally intensive, making it difficult to manually compute and implement its derivatives with respect to the orientation angles. Therefore, we designed a differentiable approximation to the mean disorientation; it is implemented using TensorFlow ([**93**]) and optimized using stochastic gradient descent ([**26**]). The training of the deep learning model was optimized to take advantage of the parallelization available in Graphics Processing Unit (GPU) to process a complete mini-batch.

The second challenge is that the crystal orientation is represented using three Euler angles which requires learning all three angles simultaneously using a single model, which is different from multi-labeling problems that require predicting different objects present in the input image ([**172, 173, 174**]). Most state-of-the-art deep learning architectures are limited to predicting a single output ([**4, 73, 120**]); existing work on multi-output learning using neural networks has been limited to shallow feed-forward networks with a single regression layer having multiple outputs ([**175, 176, 177**]).

We design and implement a novel branched deep convolutional neural network (CNN) optimized for learning multiple outputs; we refer to this model as OMNet. The training and test sets are composed of EBSD patterns of polycrystalline Nickel. The simulation dataset used for training the models is composed of $374,852$ EBSD patterns. The models are evaluated using a set of $1,000$ EBSD patterns from real experiments. The OMNet model outperforms the current dictionary based indexing by 16%, resulting in a mean disorientation of 0.548° compared to 0.652° for the dictionary approach.

6.2. Background

6.2.1. Crystal Orientation and Disorientation

The orientation of a crystal is represented by a passive 3D rotation, $\mathbf{g}$, which maps the specimen's right-handed Cartesian coordinate frame, $\mathbf{e}^s \equiv (\mathbf{e}_1^s, \mathbf{e}_2^s, \mathbf{e}_3^s)$ onto a right-handed Cartesian coordinate system attached to the crystal, $\mathbf{e}^c \equiv (\mathbf{e}_1^c, \mathbf{e}_2^c, \mathbf{e}_3^c)$, such that $\mathbf{e}_i^c = \mathbf{g}_{ij}\mathbf{e}_j^s$; in this representation, the orientation $\mathbf{g}$ corresponds to a 3×3 special orthogonal matrix, i.e., an element of $SO(3)$. There are numerous other representations for orientations, such as the unit quaternion, Rodrigues-Frank vectors, axis-angle pair and cubochoric vector; each with its own distinct properties and advantages. Furthermore, all crystals have certain symmetries associated with them, which leads to degeneracies such that all crystal orientations are not unique. For any crystal, let $\mathcal{O}_c$ represent the set of symmetry operators including the identity operation, with cardinality $\#\mathcal{O}_c = N$. All orientations in the set $\mathcal{O}_c\mathbf{g}$ are equivalent for this crystal symmetry and represent identical orientations.

In the absence of crystal symmetry, the distance metric between two orientations $\mathbf{g}_1$ and $\mathbf{g}_2$ is represented by $\mathcal{D}(\mathbf{g}_1, \mathbf{g}_2)$, and is referred to as the misorientation. This metric represents the angle of rotation about some axis to go from one crystal orientation to the other. In this case, the space has simple analytical expressions for the metric tensor as well as smooth and continuous geodesics. The distance metric is given by (assuming $\mathbf{g}$ is in matrix notation)

$$\mathcal{D}(\mathbf{g}_1, \mathbf{g}_2) = \arccos\left(\left(\mathrm{tr}\left[\mathbf{g}_1^{-1}\mathbf{g}_2\right] - 1\right)/2\right). \tag{6.3}$$

However, in the presence of crystal symmetry, the rotation space becomes degenerate and such an expression is no longer valid. In the presence of crystal symmetry given by the set $\mathcal{O}_c$, the distance metric referred to as disorientation is given by the following expression (assuming $\mathbf{g}$ and $\mathcal{O}_c$ are both in matrix notation):

$$\mathcal{D}(\mathbf{g}_1, \mathbf{g}_2) = \min_{i,j \in [1,N]} \left\{ \begin{array}{l} \arccos\left(\left(\mathrm{tr}\left[\left(\mathcal{O}_c^i \mathbf{g}_1\right)^{-1}\left(\mathcal{O}_c^j \mathbf{g}_2\right)\right] - 1\right)/2\right), \\ \arccos\left(\left(\mathrm{tr}\left[\left(\mathcal{O}_c^i \mathbf{g}_2\right)^{-1}\left(\mathcal{O}_c^j \mathbf{g}_1\right)\right] - 1\right)/2\right) \end{array} \right\}. \tag{6.4}$$

This expression gives the *minimum* angle of rotation, i.e., the disorientation, about some axis between any two symmetrically equivalent variants of the two orientations $\mathbf{g}_1$ and $\mathbf{g}_2$.

Since its development in the early 1990s, automated Electron Backscatter Diffraction (EBSD) has become the primary tool to determine the crystal orientation of crystalline materials across a wide variety of material classes [**156**]. The technique provides quantitative information about the grain size, grain boundary character, grain orientation, texture and phase identity of the sample by measuring the angular distribution of backscattered electrons using a combination of scintillator screen and a charge coupled device (CCD) camera. The schematic of the EBSD setup is shown in 6.1. The sample sits at a tilt of σ (typically 70°) with the camera tilted at angle θ_c (typically $0° - 10°$). A parallel beam of electron travels from the pole piece and interacts with the sample at point **O**. The backscattered yield is measured by the scintillator. The physics-based model predicts the

backscattered yield based on the principles of quantum mechanics. Assuming the microscope is parametrized by $\mathcal{M}$, the geometry of the setup is denoted by $\mathcal{G}$ and the crystal under investigation is parametrized by $\mathcal{C}$, the forward model is given by

$$\mathcal{F} \equiv \mathcal{F}(\mathcal{M}, \mathcal{G}, \mathcal{C}). \tag{6.5}$$

Further details of the model can be found in [**157**]. An example experimental EBSD pattern from Iron with its corresponding physics-based simulation is shown in Fig. 6.2(a)-(b) respectively.

6.3. Design

The ideal data driven approach for building a predictive model would be to train a machine learning model on the EBSD patterns from experiments. However, experiments are generally expensive and yield a relatively small number of diffraction patterns; in our case, we have selected 1000 "experimental" diffraction patterns. Instead, we leverage the ability to simulate realistic EBSD patterns for training, such that the model can predict the crystal orientations for the experimental EBSD patterns with minimum disorientation with respect to the true orientations.

The training and test datasets are composed of EBSD patterns of polycrystalline Nickel. The training dataset contains two simulated EBSD pattern dictionaries, one generated with a cubochoric sampling of $N = 100$ samples along the cubic semi axis, the other with $N = 50$ (see [**178**] for details). The first dataset has $333,227$ patterns, and the second has $41,625$ patterns. Combining them, a total of $374,852$ patterns were used as the training set without any data augmentation. As the pattern pixel values

range from $[0, 255]$, they were rescaled to the range of $[0, 1]$. The performance of the models was evaluated by indexing $1,000$ simulated patterns with known orientations. The disorientation between the predicted and known Euler angles provides the efficacy of the approach. It is important to note that the microscope and diffraction geometry for the training and test set were identical. There exist two main challenges associated with developing such a predictive model. We discuss these challenges along with how we tackle them below.

6.3.1. Optimizing the Mean Disorientation Error

This problem requires optimizing the mean disorientation error between the predicted and the true crystal orientations. This is a challenging task for two reasons: first, the disorientation is the distance metric of a non-euclidean manifold. In the absence of symmetries in orientation, this metric is easily computed using analytical expressions. However, the presence of crystal symmetries introduces degeneracies in the space resulting in discontinuities in the gradient of the disorientation metric with respect to the input orientations. This renders the disorientation function inappropriate for optimization using stochastic gradient descent for any deep learning model.

Second, the original disorientation algorithm is computationally intensive; for 432 cubic rotational symmetry, it takes one pair of predicted and true Euler angles and computes their 24×24 symmetrically equivalent orientations to find the disorientation (Equation 6.4). It would be extremely cumbersome to manually compute and implement its derivatives with respect to the input Euler angles. Hence, it is infeasible to train any predictive model using the disorientation function in feasible time.

The disorientation is computed using Equation 6.4. It consists of $1,152$ evaluations, each step computing the symmetrically equivalent orientation pairs, followed by 2 computations to determine the required angle of rotation between them. The disorientation computation contains the $\arccos(x)$ function which is undefined for values outside its domain of $[-1, 1]$. We approximated it by putting an upper bound of 1 to the magnitude of all the values passed to the $\arccos(x)$ function.

We implemented a differentiable approximation of the mean disorientation by building a computational tensor graph using TensorFlow ([**93**]). We leveraged its auto-differentiation support for computing the gradients of the mean disorientation error with respect to the Euler angles. The mean disorientation error was optimized by training a deep learning model using the stochastic gradient descent algorithm [**26**]. When the minibatch size was 64, the sequential algorithm involved $64 \times 1,152 = 73,728$ computations for the misorientation between the symmetrically equivalent predicted and ground truth crystal orientations. It was very costly both in terms of processing time and memory transfer; it took around 24 hours to train our model for one epoch using a TitanX GPU with 12GB memory. This made it impractical to train a deep learning model using the sequential implementation in feasible time. We optimized it to process one mini-batch so that it could leverage the parallelization available in GPUs.

The models are evaluated using the mean disorientation error and mean symmetrically equivalent orientation absolute error (MSEAE). The MSEAE is computed by considering the periodicity of orientation angles as follows:

$$\text{mseae}_1(\vec{y_s}, \hat{\vec{y}_s}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{3} |\vec{y_{sj}} - \hat{\vec{y}_{sj}}|, \tag{6.6}$$

$$\text{mseae}_2(\vec{y_s}, \hat{\vec{y_s}}) = \text{mseae}_1(\vec{y_s}, \hat{\vec{y_s}}) \mod (2\pi), \tag{6.7}$$

$$\text{MSEAE}(\vec{y_s}, \hat{\vec{y_s}}) = \begin{cases} \text{mseae}_2(\vec{y_s}, \hat{\vec{y_s}}), \text{if mseae}_2(\vec{y_s}, \hat{\vec{y_s}}) \leq \pi \\ = 2\pi - \text{mseae}_2(\vec{y_s}, \hat{\vec{y_s}}), \text{else} \end{cases} \tag{6.8}$$

where $\vec{y_s}$ and $\hat{\vec{y_s}}$ are Euler angle triplets of the symmetrically equivalent true and predicted orientations with minimum disorientation.

6.3.2. CNN Architectures for Learning Multiple outputs

Optimizing for the mean disorientation requires learning the crystal orientation angles using a single model such that they can be used to optimize the mean disorientation error. The conventional approach to learning multiple outputs would be to train individual models for learning each output. Since the three Euler angles are correlated with the crystal orientation, we have to learn all the three orientation angles simultaneously using a single model such that they can be leveraged for optimizing for mean disorientation. Existing multi-output learning using neural networks has been limited to shallow feed-forward networks having a regression layer with multiple outputs ([**175, 176, 177**]).

We explored several design approaches for learning multiple outputs. Figure 6.3 demonstrates a novel CNN model architecture – a branched model with individual and independent model components for each output. The first four model components are composed of multiple convolution layers and max pooling. Convolution layers capture the locally correlated features present in the input EBSD patterns; they learn the high level abstract features from the inputs. As the three outputs require similar learning

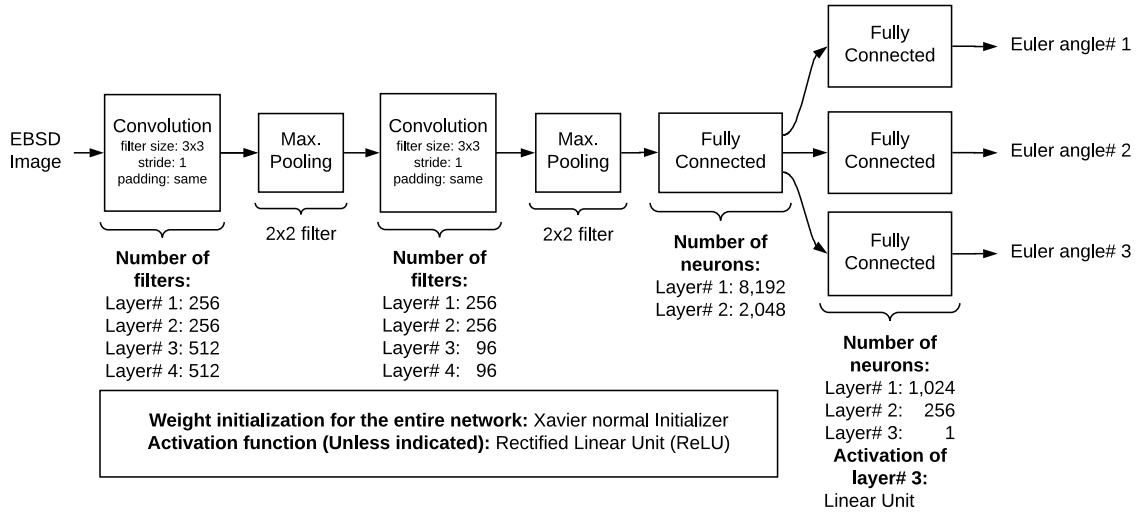


Figure 6.3. OMNet: CNN architectures for learning multiple outputs.

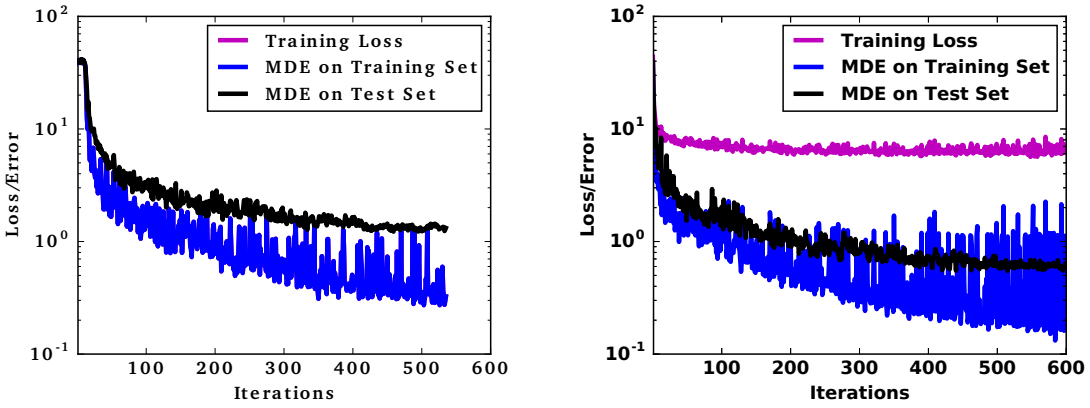
capability, the branched model is composed of three classifier branches containing equal numbers of layers and parameters.

Each branch leverages around 2 million parameters independent of each other which can be optimized to learn the individual outputs. As the outputs are correlated with each other, they all share the same convolution outputs and the first fully connected component. The convolutional layers are the computationally expensive components; they extract high level features from the inputs that are required for learning all outputs. Sharing the convolutional layers keeps the computational cost comparable to the model having a single regression layer with multiple outputs. The branching technique is currently used in the inception model architectures but for a different reason [**179**]. The point where the model starts branching can have a significant impact on the model performance. For the training, we explored branching at different layers but it was limited by the available

GPU memory. Since the model architecture is designed to optimize for learning multiple outputs, we refer to this architecture as OMNet.

All models were implemented using Python and TensorFlow [**93**]. They are trained using Titan X GPUs with 12GB memory. An extensive search was carried out to tune the hyperparameters, such as learning rate, optimization algorithm, momentum and learning rate decay. We used a batch size of 64 and trained using Adam [**130**] for 100 epochs with a patience of 10. We searched through several CNN model architectures and loss functions – conventional loss functions, followed by mean disorientation error, and finally their hybrids.

6.4. Experimental Results



(a) MDE

(b) MAE+MDE

Figure 6.4. Loss and mean disorientation error using different loss functions. (a) shows the training loss and mean disorientation error (MDE) on training set and test set for MDE as the loss function. (b) shows the loss and MDE for the hybrid loss function of the sum of mean absolute error (MAE) and MDE.

Table 6.1. Mean Disorientation Error (MDE) and mean symmetrically equivalent orientation absolute error (MSEAE) using different models and loss functions.

Model	**Loss Function**	**Simulation Data**	**Experimental Data**	
		MDE	**MDE**	**MSEAE**
Dictionary-based Indexing	-	-	0.652°	[0.6592°, 0.3534°, 0.6484°]
Deep Learning (OMNet)	MAE	0.064°	0.596°	-
Deep Learning (OMNet)	MSE	0.292°	1.285° -	-
Deep Learning (OMNet)	MDE	0.272°	1.224°	-
Deep Learning (OMNet)	MAE+MDE	0.132°	0.548°	[0.7155°, 0.2194°, 0.7066°]
Deep Learning (OMNet)	MSE+MDE	0.171°	0.658°	-

We optimized using the mean disorientation error as the loss function as shown in Figure 6.4(a). Since our goal was to optimize for mean disorientation error, we expected the algorithm to result in an improved mean disorientation error on both sets. However, the mean disorientation error alone as the loss function did not perform well; it achieved a mean disorientation error of 1.224° on the test set (experimental EBSD patterns). We observed a lot of oscillations in the training loss curve compared to while using conventional loss functions. This may be due to the mean disorientation error being computed using the symmetrically equivalent orientations rather than the actual outputs (Equation 6.4). Optimizing for the mean disorientation error increased the model training time by around 30%-40%.

Since optimizing for mean disorientation did not perform well, we designed and experimented with several hybrid loss functions, combining the mean disorientation with the conventional loss functions such as mean absolute error (MAE) and mean squared error (MSE). We assigned different weights to the constituent losses, even some conditional loss functions that optimized for the Euler angles first, followed by optimizing for the mean disorientation or a hybrid loss. The best loss function was the sum of the MAE and the mean disorientation error, as shown in Figure 6.4(b) and Table 6.1. The mean disorientation error decreased steadily with time although the total loss became almost constant. The mean disorientation error on the training set is computed using only the current mini-batch; hence, the mean disorientation error curve for training set has oscillations. The model achieved a mean disorientation error of 0.548° on the experimental EBSD patterns, about 16% better than using dictionary based indexing.

6.5. Conclusion

It is one of the goals of EBSD analysis to obtain grain orientations that are as accurate as possible. Having the ability to determine grain orientations to within a fraction of a degree makes it possible to perform quantitative comparisons between experimental data sets and predictive micro-structure models, in particular for cases in which the material contains large amounts of plastic deformation. Current commercially available EBSD indexing solutions do not perform well when the material is heavily deformed. The recent dictionary indexing approach performs significantly better but suffers from a high computational cost which makes the approach unfeasible as a real-time indexing solution. The deep learning approach described in this work has the potential to impact the field of materials science by providing an indexing approach that is both rapid and accurate, once the training process has been completed.

If deep learning based indexing in real time becomes possible, then this would have a significant impact on the field of materials characterization by providing a faster and more accurate indexing approach than is currently available commercially. While this work establishes the efficacy of neural networks in learning orientations from EBSD patterns for pristine simulated patterns, the characteristics of such a network in the presence of noise still need to be established. The current model was trained with orientations as the only variable and assuming that the microscope geometry is precisely known; this is seldom the case for real experiments. New strategies leveraging techniques in incremental learning might prove to be useful for training the model for different detector geometry parameters. Finally, the current model can also be extended to other electron diffraction modalities such as the Scanning Electron Microscope (SEM) based Electron Channeling

Patterns (ECP) and Transmission Kikuchi Diffraction (TKD) modality as well as the Transmission Electron Microscope (TEM) based Precession Electron Diffraction (PED) patterns.

CHAPTER 7

Peak Area Detection Network for Directly Learning Phase Regions from Raw X-ray Diffraction Patterns

7.1. Introduction

In materials science and crystallography, X-ray diffraction (XRD) is a widely used experimental technique to probe materials at the atomic level. XRD analysis is used by scientists and engineers to understand atomic-scale crystal structures and predict their properties [**180, 181, 182, 183, 184**]. XRD patterns not only provide the geometrical information about the crystal structure, they are also used to determine the possible flaws in materials [**185**]. High throughput experimental techniques developed over the last few decades have accelerated the exploration of material properties. Combinatorial methods allow experimentalists to synthesize hundreds or thousands of materials at a time, with each sample varying by synthesis and processing parameters [**186**]. Composition spreads are one example, where a wafer is generated containing hundreds of samples, each varying in composition. Once such a wafer is generated, the properties of each sample can be rapidly measured using scanning microscopy techniques [**187**]. As a result, over the course of hours, XRD data can be collected for hundreds or thousands of materials.

Currently, human experts analyze the XRD patterns using domain knowledge such as peak shape and location; they are correlated with the sample composition and known phases to identify the phases in the measured sample. The current approach for the

analysis of XRD patterns is a multi-stage process composed of multiple computationally intensive steps. The first step is to convert the raw 2D XRD pattern to an intensity-2θ (1D) pattern by mapping the raw XRD image to the χ vs 2θ space and then integrating the intensity peaks along the 2θ axis [**188**]. XRD patterns are often noisy due to a collection of issues including background radiation, detector noise, and low count of incident X-rays. In addition, other background issues may be introduced by the sample-detector configuration, resulting in a significantly varying measurement background from sample to sample [**189**]. The presence of a highly irregular background makes the peak searching procedure complicated. Hence, the background signal is removed from the 1D patterns by fitting background curve [**190, 191, 192**]. This is followed by indexing the peaks against an existing database of reference peaks and correlating with the sample composition to identify the phases in the measured sample using available software which often requires verification by a domain expert.

Over the last decade, machine learning has been used to accelerate the process of indexing using 1D XRD patterns [**193, 194, 195**]. Clustering has been used to sort samples into groups of materials that share the same constituent phases - thus reducing the number of samples required to index for unique phases [**196, 197**]. When plotted against the synthesis and processing parameters used to generate the samples, these clusters describe geometric "phase regions" - regions of the generative space where materials are expected to share the same constituent phases. Additionally, once a subset of samples has been sorted into phase regions, classification has been used to extrapolate these phase region labels to the rest of the samples [**195**]. Recently, Park et al. [**169**] used a convolutional neural network (CNN) to classify 1D XRD patterns into space group,

extinction-group and crystal-system classification. They used 150 000 powder XRD patterns calculated from the structure solutions of entries in the Inorganic Crystal Structure Database (ICSD) using Density Functional Theory (DFT).

Time is a limiting factor when collecting and analyzing X-ray diffraction data. For typical laboratory systems, low beam intensity means measuring each sample can take tens of minutes to hours. Additionally, when performing X-ray diffraction at the beamline, measurements only take tens of seconds, but the total time is often limited to hours or days. Accelerating the computational time required in data analysis can impact the measurement experiment since it directly impacts the decisions. Hence, our goal is to build a predictive model for eliminating the current computationally intensive process of multi-stage XRD analysis process. We focus on this issue by directly learning the phase regions from the raw (2D) XRD patterns through classification using deep learning to make the overall process automatic and faster.

In this paper, we introduce a Peak Area Detection Network (PADNet) to directly learn to predict the phase regions from the raw XRD patterns without any need for preprocessing or background noise removal. PADNet is a specially designed CNN that contains large symmetrical convolutional filters with filter size of 50×50 in the first layer. These filters are initialized using either 1 or -1 across different symmetries and they compute the difference in intensity counts across different symmetries which enable them to capture the peak areas and automatically remove the background noise. To evaluate the proposed approach, we experiment using two sets of XRD patterns from the Stanford Linear Accelerator Center (SLAC) [**187**] and the National Institute of Standards and

Technology (NIST) [**198**]; each of them contains 177 XRD patterns from a Sn-Ti-Zn-O thin-film, composition-spread, combinatorial library sample with eight phase regions as the labels. The XRD patterns from SLAC contain significant irregular background which varies by sample, while the ones from NIST contain comparably low background which, as a function of 2θ, does not significantly vary from sample to sample. To our knowledge there does not exist any algorithm for removing background noise for the raw 2D XRD image; hence, we also explore some novel background removal techniques based on minimum and mean convolutional filters.

We evaluate the performance of PADNet using 10-fold cross validation. PADNet achieves an overall classification accuracy of 84% for the multi-class labeling task, with the SLAC model performing slightly better than the NIST model. Our results demonstrates that PADNet can successfully predict the phase regions from the raw 2D XRD patterns independent of presence of background noise. We also compared our approach against the recent approach of phase region classification using 1D XRD patterns from Bunn et al. [**195**]; PADNet significantly outperformed the AdaBoost classifier for both datasets.

7.2. Background

X-ray diffraction is an atomic scale probing technique for determining the crystal structure of materials [**181, 182, 183, 184**]. The crystal structure causes the beam of incident X-rays to diffract into many specific directions; a 3D image representing the density of electrons in the crystal can be constructed by measuring the angles and intensities of the diffracted intensity patterns. An X-ray diffraction image is a plot of the intensity of X-rays scattered at different angles by a materials sample, as measured by a 2D detector,

with each pixel measuring the number of incident X-rays. The atomic-scale structures of materials can be determined using the XRD technique [**180**].

The XRD pattern from a material composed of periodic atomic structures is composed of multiple sharp spots known as Bragg diffraction peaks; the positions and intensities of these peaks determine the phase of the materials - the specific chemistry and atomic arrangement. For instance, quartz, cristobalite and glass are all different phases of SiO_2; they are chemically identical but the atoms are arranged differently, the XRD pattern is distinct for each phase. A phase map represents the physical conditions at which thermodynamically distinct phases occur and coexist. The constituent phases in the phase map represent the different crystal lattice structures for varying material composition. Scott [**188**] provides more details about X-ray powder diffraction.

7.2.1. Motivation

Our current work is motivated by the success of CNNs for image classification [**4, 73**]. In our previous work, we have shown the efficiency of deep neural networks in learning crystal orientations directly from electron diffraction patterns [**116**]. Recently, Park et al. applied CNN for classification of crystal structure using 1D XRD patterns. Since deep neural networks are supposed to require large training datasets and the experimentally measured XRD patterns are limited, they used 150,000 1D XRD patterns calculated from the structure solutions of every entry from the Inorganic Crystal Structure Database [**96**]. However, our previous work has shown that deep neural networks can be leveraged even with relatively smaller datasets and perform better than traditional machine learning techniques [**99**].

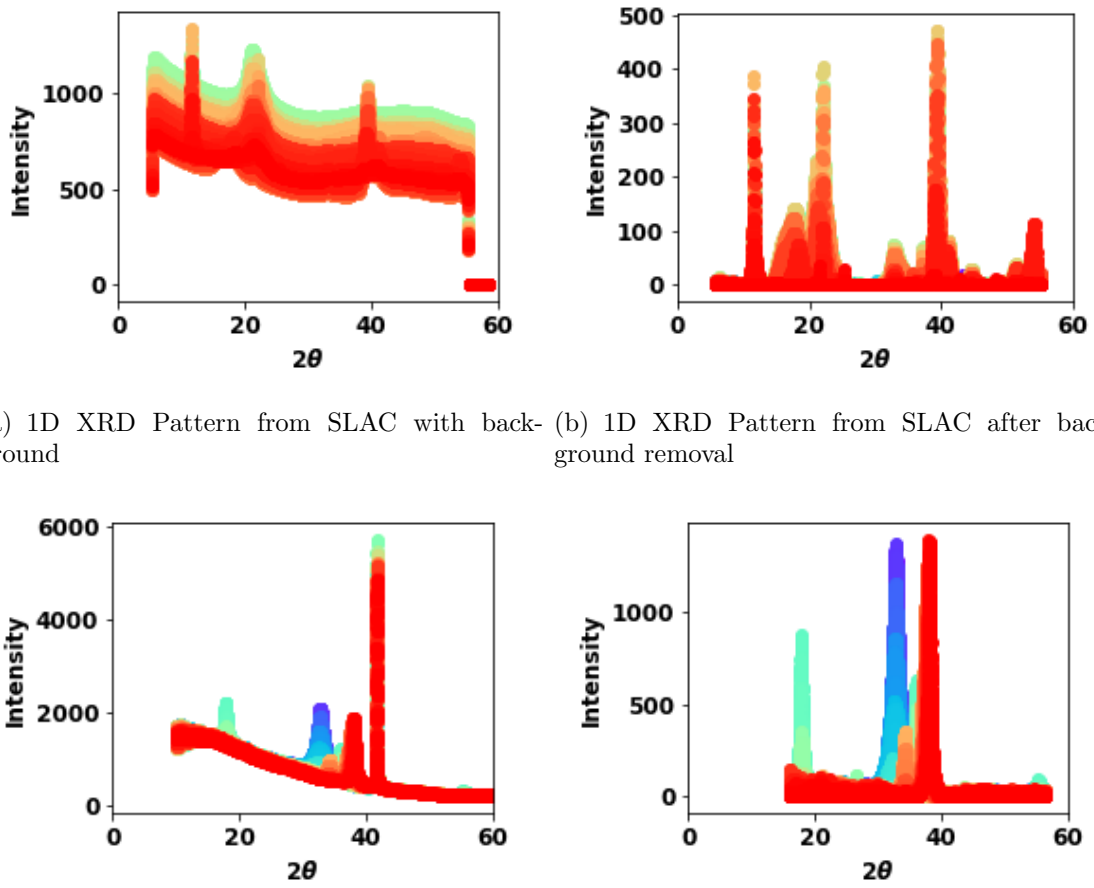
Directly using the raw 2D XRD patterns is also beneficial from the perspective of information content. The conversion from 2D raw patterns to 1D intensity-2θ patterns results in loss of important information due to their limited representation. The peak characteristics such as peak height, peak width, presence of secondary peaks (peak doublets), are very critical to correctly understand the materials structure. For instance, the peak broadening can be used to quantify the average crystalline size of nanoparticles, lines on the 2D raw pattern represent polycrystalline structure, and points on the 2D raw pattern represent very well ordered crystalline structure. Such fine grained differentiation is very critical to understand the true structure of materials. However, these facts are ignored because during the conversion to 1D, such information is lost.

7.3. Design

7.3.1. Challenges

The primary challenge of XRD data analysis is the presence of background noise which can be highly irregular such as in the case of SLAC as shown in Figure 7.1. During the experiment, several factors can impact the XRD pattern captured, some of which are beyond human control. It depends on multiple effects: machine setup, air around the wafer, etc. The presence of background makes it difficult to detect the intensity peaks which are important for obtaining the crystal information. For beamline, the resulting background is not a simple bias to subtract. Hence, background removal is a primary concern for XRD pattern analysis.

Although several techniques exist for removing background in the 1D XRD samples [**190, 191, 192**], and parsed 2D XRD patterns [**199**], to our knowledge, there exists



(a) 1D XRD Pattern from SLAC with background

(b) 1D XRD Pattern from SLAC after background removal

(c) 1D XRD Pattern from NIST with background

(d) 1D XRD Pattern from NIST after background removal

Figure 7.1. 1D XRD Patterns from SLAC and NIST. The XRD patterns from SLAC contains highly irregular noise while the noise in the case of XRD patterns from NIST is a constant function of 2θ.

no technique for removal of background noise from the raw XRD patterns coming directly from the experiments. The raw 2D XRD patterns are convoluted rather than being a 2D rectangle; the background removal methods for parsed 2D XRD patterns do not work for the raw XRD patterns. For example, we implemented the Cache-efficient 2D Bruckner

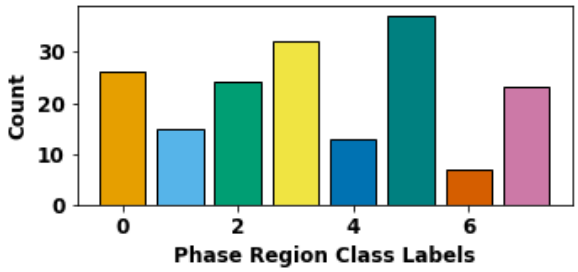


Figure 7.2. Distribution of class labels for the two XRD datasets. XRD Patterns are collected for the same composition space of Sn-Ti-Zn-O from both NIST and SLAC; hence, they refer to same samples.

Filter from Baur et al. [**200**] which is designed for parsed 2D XRD patterns, but does not work for the case of raw 2D XRD patterns.

Another challenge associated with this task is the limited dataset size. Our dataset contains only 177 XRD samples (and we have eight classes to learn). Since deep neural networks are supposed to require large training datasets and the experimentally measured XRD patterns are limited, Parker et al. [**169**] used 150,000 1D XRD patterns calculated from the structure solutions of every entry from the Inorganic Crystal Structure Database [**96**]. However, here our goal is to directly learn from the raw 2D XRD images coming from experiments.

7.3.2. Datasets

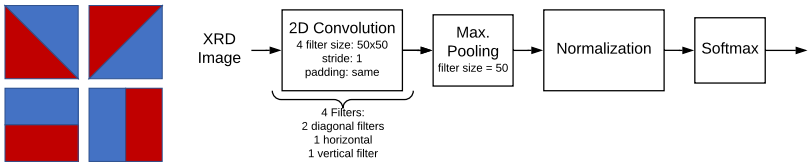
We leverage the XRD patterns collected at SLAC [**187**] and NIST. SLAC has a high throughput system for XRD experiments [**187**]; it outputs a single XRD pattern for a specific range of 2θ; the configuration used gives a 2θ range of 5.365 to 58.566 for our experiments. The NIST system was used to collect two diffraction frames centered at the 2θ values of 25 and 45, the range for the two frames were [10, 40] and [30, 60]. XRD

patterns from SLAC contain more features due to the high energy of the beam, we are able to resolve XRD with greater signal to noise ratio at greatly reduced exposure times; however, the instrument is less available.

Each dataset is composed of 177 XRD patterns for the material alloy system with different compositions of Tin, Titanium, and Zinc (Oxygen is also present but not controlled) from experiments. Each XRD pattern is of size 2048×2048 containing the intensity values; hence, they are not like regular images with three channels (red, green, blue) used for image classification such as in ImageNet [**4**]. In addition to the XRD patterns, the composition information for each sample is also available in the dataset. The samples were labeled by converting to 1D, clustered, then followed by human expert validation. There are eight phase region classes, some represent pure constituent phases while others represent mixed phases. As shown in Figure 7.2, the distribution of the dataset is not balanced. The largest class has 37 samples while the smallest class has only 7 samples. We used random split during our ten-fold cross validation; for each training set from the random split, the smallest class was present in all of the training set during ten-fold cross validation. One option could be to remove the class with data count below a certain threshold, but our dataset is already limited and some of the phase regions are mixed (combination) of other phase regions (classes); hence, learning one phase region can help in predicting the other phase region. Therefore, we decided not to drop any phase region class from our dataset.

(a) PADNet for XRD patterns from SLAC (b) PADNet for XRD patterns from NIST

Figure 7.3. PADNet model architectures for the XRD patterns from SLAC and NIST. Since both datasets refer to the same composition space of Sn-Ti-Zn-O and have same samples, we constrained both models to have same number of model parameters and same architecture. PADNet for NIST is composed of two convolutional graphs to handle the two XRD patterns compared to the PADNet for SLAC having one convolutional graph since SLAC outputs one XRD image.



(a) Slope Filters (b) Peak Area Detection Component

Figure 7.4. Peak Area Detection Component with Slope Filters: This component contains slope filters which help in peak detection by measuring the difference in slope across different symmetries. The slope filters has two regions: blue representing -1 and red representing 1. Since they are symmetric, they can effectively detect the high slope areas containing peaks.

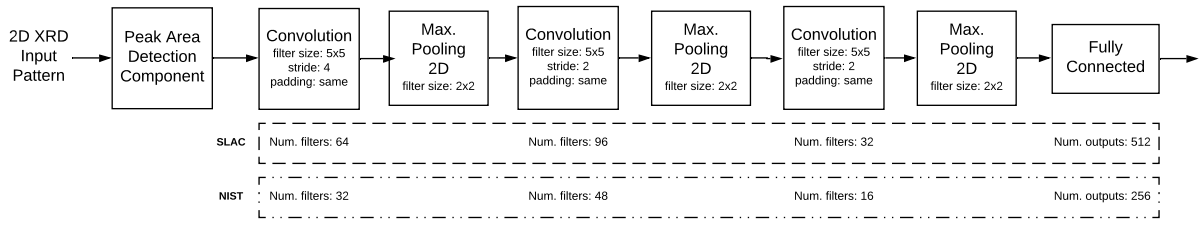


Figure 7.5. Convolutional Graph: The component of the CNN network for the raw 2D XRD pattern in the input.

7.3.3. Peak Area Detection Network

PADNet is a deep convolutional neural network for directly learning the phase regions from the 2D raw XRD patterns. PADNet is composed of three components: a convolutional graph with a peak area detection component for each input XRD pattern, a dense network for vector composition in the input and a final classifier network containing dense layers for classification. The PADNet architecture for the XRD patterns from both datasets are shown in Figure 7.3.

The first layer of the convolutional graph is composed of four large convolutional filters with filter size $f = 50$. These filters are initialized in a special symmetrical manner as follows:

$$F1_{i,j} = \begin{cases} -1, & \textit{if } i < j \\ 1, & \textit{otherwise} \end{cases} \tag{7.1}$$

$$F2_{i,j} = \begin{cases} -1, & \textit{if } i + j > f \\ 1, & \textit{otherwise} \end{cases} \tag{7.2}$$

$$VF_{i,j} = \begin{cases} -1, & \textit{if } i < f/2 \\ 1, & \textit{otherwise} \end{cases} \tag{7.3}$$

$$HF_{i,j} = \begin{cases} -1, & if\ j < f/2 \\ 1, & otherwise \end{cases} \tag{7.4}$$

$F1$ and $F2$ are two diagonal filters symmetric about the diagonals. HF is a filter symmetric about the horizontal and VF is symmetric about the vertical. These filters are illustrated visually in Figure 7.4. Due to their symmetries with opposite signs on the two sides, these filters F measure the difference in intensity counts and the background is automatically implicitly subtracted at each point in Equation 7.5, where I is the input XRD pattern and we refer to these filters as slope filters.

$$I_{i,j} = \sum (I_{i+x,j+y}) \cdot F_{x,y} \qquad \text{for} - f/2 \leq x, y \leq f/2 \tag{7.5}$$

The symmetry with opposite sign also means that the value computed on opposite symmetries around the peak will have opposite signs. We are interested in the peak area. Hence, we take their absolute values as follows:

$$I_{i,j} = |I_{i,j}| \tag{7.6}$$

The value of slope measured at the actual peak should be zero since the intensity counts across a peak should be symmetric. Hence, to detect the area around a peak including the peak itself, we apply a maximum filter with a filter size $f = 50$ as follows:

$$I_{i,j} = \max(I_{i+x,j+y}) \qquad \text{for } -f/2 \leq x, y \leq f/2 \tag{7.7}$$

Next, we normalize the outputs from each filter using batch normalization to make the mean zero and variance 1 for proper learning in the next convolutional layers. After the batch normalization, we apply a softmax activation function so that the network puts more emphasis on the points with high slopes and hence, high intensity counts. The softmax function is defined as follows:

$$softmax(I_{i,j}) = \frac{e^{I_{i,j}}}{\sum e^{I_{i,j}}} \tag{7.8}$$

Figure 7.4 illustrates the specially designed network component for the peak area detection. The output from the peak area detection component is fed into the next convolutional layer of the convolution graph component. Figure 7.5 illustrates the convolutional graph of the CNN network used for the two datasets. Since NIST samples contain two XRD patterns, the NIST model contains two convolutional graphs for each input pattern, but with half number of filters compared to SLAC model. In this way, both SLAC and NIST models have equal number of trainable parameters, and we can fairly compare their performance with each other using our domain intuition.

The dense network for composition input is composed of two fully connected layers with 256 outputs in each layer. The output from the convolutional graph is concatenated with output from the dense network for composition input and fed into a final classification

network that learns to predict the crystal phase label. The final classification network is composed of two layers with 256 outputs in the penultimate and 8 outputs in the last layer. ReLU [**88**] is used as the activation function. Batch normalization [**126**] is used after each layer for the faster convergence. Since there are eight phase labels, the last fully connected layer in the classification network has softmax activation with eight outputs.

7.3.4. 2D Background Removal from Raw XRD Pattern

One of the domain constraints before performing any analysis is how to remove the background from the XRD patterns so that the peaks can be easily detected. Hence, we explored some of the commonly used techniques used for background removal and smoothing for background removal from the raw 2D XRD patterns. The raw 2D XRD patterns are either in GFRM or TIF format, we will refer them as I. These XRD patterns contain intensity values for different values of χ vs 2θ and have a size of 2048×2048. Generally, resizing is done to reduce the computation required; but, we do not perform any resizing as that can lead to information loss.

First, we apply a minimum 2D filter of size $f \times f$ to the raw input image I as follows:

$$MF_{i,j} = \min(I_{i+x,j+y}) \text{ for } -f/2 \leq x, y \leq f/2 \tag{7.9}$$

MF represents the background obtained by fitting a minimum filter. This can be subtracted from the raw pattern I to obtain the pattern with background removal IM.

After applying the minimum filter, we found that the output pattern IM contains some edges and corners. Next, following the smoothing techniques for 1D XRD patterns

such as in [**190**], we apply a convolutional filter of size $f \times f$ to smooth the background as follows:

$$CS_{i,j} = \sum (IM_{i+x,j+y} \cdot F_{x,y}) \quad \text{for } -f/2 \leq x, y \leq f/2 \tag{7.10}$$

where F is a constant mean filter containing the same value at each position that sum up to one. Since the XRD pattern is large in size 2048×2048, fitting a polynomial using least square can be very expensive without scaling down the size which will impact the quality leading to loss of information. Hence, we applied the convolution mean filter. The smooth background CS can be subtracted from the image I to obtain the XRD pattern without background: IC. We will used both IM and IC as the input XRD pattern evaluate our model performance.

Time Complexity: The time complexity for the background removal for 2D raw XRD patterns is $\mathcal{O}(h \cdot w \cdot f^2)$ where h is the height and w is the width of the input XRD pattern I and f is the filter size.

7.4. Experimental Results

7.4.1. Experimental Settings

We have used Python and the TensorFlow [**93**] deep learning framework to implement the deep neural network models. For machine learning algorithms, we used their implementations from Scikit-learn [**94**]. The models were trained using NVIDIA Titan X GPUs. We learn to predict the phase label for understanding the crystal structure. As the dataset

is small, we performed a ten-fold cross-validation and aggregated the results. Generally each fold had 160 and 18 samples in the training and test set respectively. The data splitting used for cross-validation is the same across all experiments. We experimented with different types of preprocessing such as normalizing and image whitening, but none of them worked well. Hence, we do not use any kind of preprocessing or feature engineering other than the background removal as stated. We performed a detailed hyperparameter search and architecture search for the PADNet model for both cases, but limited the two PADNet models for both datasets to same architectures for a fair comparison between the two sources since both datasets are for the same composition space. Since the dataset is limited, we used early stopping with patience of 30 and also used L2 regularization with regularization coefficient of 0.0001 to avoid overfitting. For training our models, we used a learning rate of 0.001 and Adam as the optimizer. Since we are dealing with a multi-class classification problem, we used the softmax cross entropy as the loss function and the evaluation metric is prediction accuracy which represents the total number of samples correctly classified by the model across all class in the dataset. The evaluation is carried out by training and testing the models on the raw XRD pattern I and the XRD patterns after removing background using the two methods: MF and CS. The results represent the mean and the standard deviation of the prediction accuracy from the ten-fold cross validation.

7.4.2. Background Removal from 2D Raw XRD Patterns

Figure 7.6 presents the results from background removal using the two techniques. In the case of SLAC, the background is very high and varies within a sample. The background

removal using minimum filter method demonstrates that the background obtained using this technique contain some edges and patches. The mean convolution filter removes them by smoothening using the mean of a window of size $f \times f$ where we used $f = 200$. There exist a trade-off between the size of filter and loss of peaks. If the filter size is small, it leads to loss of peaks. If the filter size is large, the background signal is still present in the output. Also, large value for filter size makes it computationally expensive since the computations required are directly proportional to the square of filter size f. The convolutional operation to compute CS background with $f = 200$ took around 7 minutes for each image on a single core of a 2.3 GHz CPU. We implemented the convolution mean filter operation using TensorFlow [**93**] to run on Tesla Titan X GPU, this reduced the operation time by $7\times$. We experimented with several values of f, $f = 200$ worked best for our experimentation here.

7.4.3. Performance using PADNet

Figure 7.7 illustrates the efficiency of using the PADNet for learning phase regions from raw 2D XRD patterns. For a thorough evaluation, we trained different models on the original raw 2D XRD images with and without background removal using the two methods MF and CS; hence, there are three types of training datasets: I, $I-MF$ and $I-CS$. To evaluate the efficiency of PADNet trained on different input types, we evaluate them using the three types of inputs for each model: I, $I-MF$ and $I-CS$. Next, the peak detection component in the convolutional graph of each model can be either held constant or trained using backpropagation so that the network can learn the slope filter parameters itself. Therefore, Figure 7.7 demonstrates the performance of models using different possible

combinations of type of training data, type of test data and the configuration of slope filters.

For SLAC, we observe a consistent performance across all input types used during evaluation for both types of configuration of slope filters and for all types of training data. The predictions made for input raw XRD pattern is completely independent of background removal which illustrates that the model can be directly used to predict the phase region labels from the raw input XRD pattern measured from experiment. The performance of the model slightly improves if the background is removed from the data. We observe similar performance for NIST model; the performance is almost consistent for all types of test XRD patterns when the peak area detection component is held constant. If the peak area detection component is allowed to be trained, the performance is slightly lower for raw XRD patterns if the model is trained using XRD patterns with background removed.

The results illustrates that PADNet can be used for directly making prediction of phase region labels from the raw XRD patterns without any need for background removal. The performance is specially interesting in the case of highly irregular background present in the case of SLAC (Figure 7.1) where the performance is completely independent of the background removal process.

7.4.4. Comparison with Current Approach

We compared the performance of PADNet against the current approach of phase region classification using 1D XRD patterns. For dataset from NIST, since there are two XRD patterns for different range of 2θ, we combined them. Next, we subtracted the background

from the 1D XRD patterns for both datasets using the envelope function in MATLAB as shown in Figure 7.1. Finally, we applied traditional machine learning approaches to both datasets.

We followed the recent approach of training an AdaBoost classifier from Bunn et al. [**195**]. We performed an elaborate grid search for hyperparameter tuning of AdaBoost. For AdaBoost classifier, we used Decision Tree Classifiers with varying depth from 2 to 10 as the estimator. For learning rate, we used the values in [1, 0.1, 0.001] and for number of estimators, we used [5, 10]. For SLAC, we obtained $(34.78 \pm 14.04)\%$ accuracy and for NIST we obtained $(83.80 \pm 14.50)\%$ accuracy using a 10-fold cross validation. While using composition, we achieved an accuracy of $(70.80 \pm 15.80)\%$ on XRD patterns from SLAC and an accuracy of $(84.03 \pm 22.33)\%$ on XRD patterns from NIST.

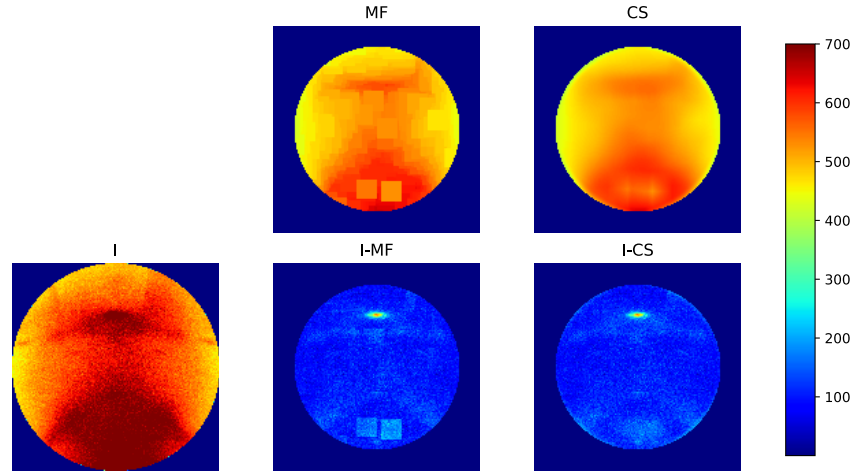
We also analyzed the performance of other types of classifiers such as Logistic Regression, Naive Bayes, DecisionTree Classifier and SGD Classifier on the two datasets with and without composition information. We performed extensive grid search for hyperparameter tuning for all of them. In general, the use of composition results into slight improvement of performance. All these models had slightly worse performance compared to AdaBoost on the two datasets, performing significantly poorly on dataset from SLAC compared to the dataset from NIST. One reason behind this might be that, for SLAC, the 1D XRD input contains only 931 intensity values compared to XRD patterns from NIST having 2501 intensity values. The more information present in the input, the better the models perform on the dataset. Another conjecture is that these models perform bad on SLAC because of high noise present in the raw XRD pattern which can lead to loss of information while converting them to 1D.

Our PADNet performs significantly better than the current approach of indexing using 1D XRD patterns on SLAC and slightly better than AdaBoost and other classifier on 1D XRD patterns from NIST. PADNet also exhibits lower deviation in the performance which shows that it can make more robust predictions. The current approach analysis of 1D XRD patterns is a computationally intensive process which involves multiple steps-converting to 1D by integrating along 2θ axis, background removal and comparison to reference database or applying a machine learning based predictive model. PADNet can provide a fast approach as the prediction takes less than one second (on a Tesla Titan X GPU) for directly predicting phase region from 2D XRD pattern.

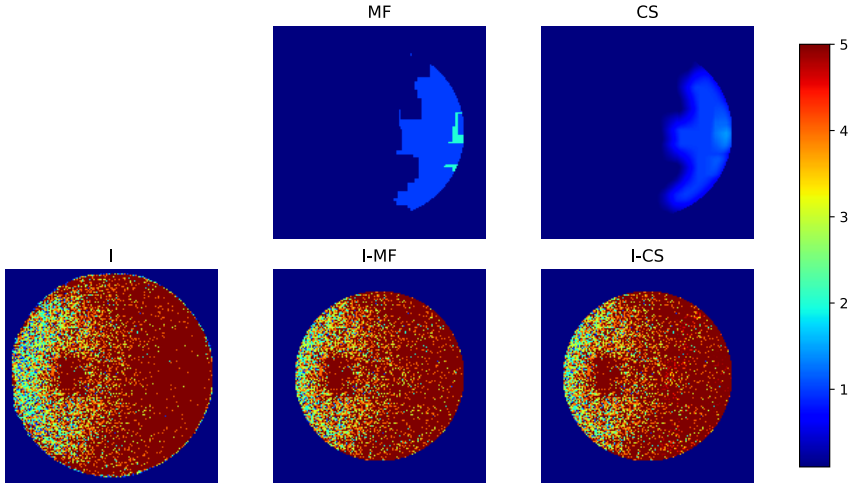
7.5. Conclusion

We designed a peak area detection network for predicting phase regions directly from raw 2D XRD patterns from real experiments. The classification results using the peak area detection network demonstrated their invariance to the presence of background in the input XRD pattern during evaluation. This illustrates that PADNet can be directly used to predict the phase regions from the raw 2D XRD patterns without any background removal almost without any impact in prediction performance. This is the first application of deep learning on the raw 2D XRD patterns from real experiments. Since deep learning works better with big training data, our approach should provide better performance if applied on larger datasets. It will hopefully pave the way for future works tapping the potential of deep learning for this and related problems. We hope this will foster the adoption of deep learning techniques for rapid and automated analysis of X-ray diffraction images, and more broadly in the field of materials science and imaging. There exists

significant potential for future research to understand the efficacy of the proposed methods such as automating the filter size selection, evaluating these on other XRD patterns, and incorporating them in a real time system for XRD analysis.

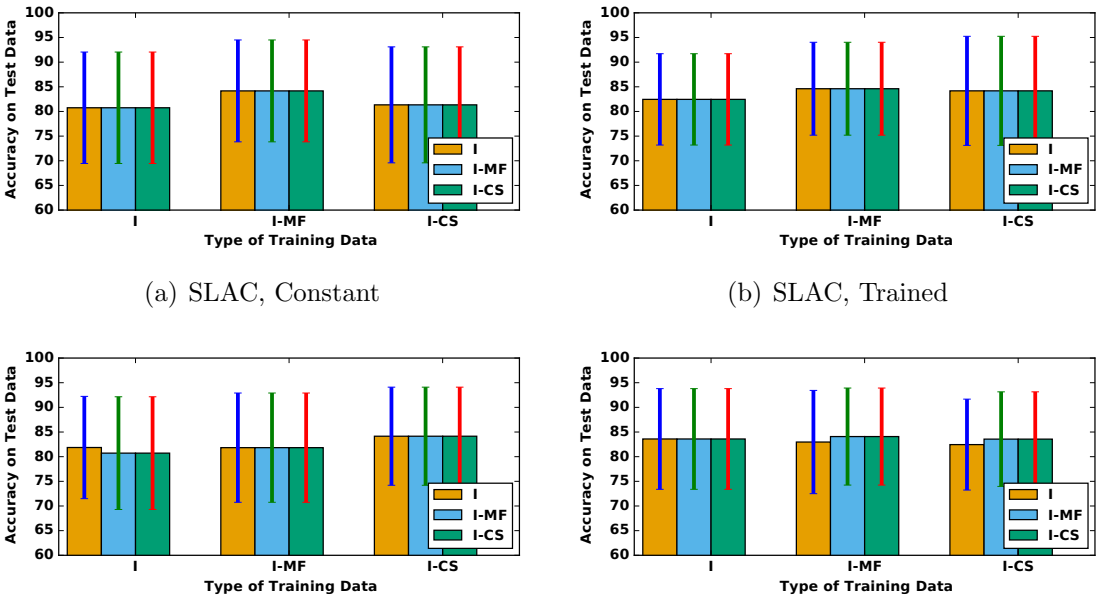


(a) SLAC



(b) NIST

Figure 7.6. Background and processed XRD images: I is the original XRD pattern on the left of (a) and (b). The top row of subfigures represent the background (MF, CS) and the bottom row of subfigures shows the XRD patterns after background removal using the two techniques (I-MF and I-CS). We used a filter size of 200 for both the minimum filter and the convolutional mean filter for both cases. For SLAC, the raw XRD pattern is similar to the background images using the two techniques; this illustrates that SLAC image contains high background noise. For NIST, the pattern after background removal look similar to the raw pattern since the background is very small. This concurs with the domain expertise, thereby suggesting that the proposed background detection module is working as expected.



(a) SLAC, Constant

(b) SLAC, Trained

(c) NIST, Constant

(d) NIST, Trained

Figure 7.7. Performance of PADNet using a ten-fold cross validation (mean and standard deviation). The uniform performance across all test pattern types exhibits the efficacy of PADNet for phase region classification directly from raw 2D XRD pattern.

CHAPTER 8

Conclusion and Future Works

8.1. Conclusion

This dissertation has presented several methodologies for designing deep neural network architectures that can handle different challenges associated with building machine learning based predictive models using scientific datasets. All the works involved developing state-of-the-art predictive models for different scientific applications for the advancement of scientific knowledge discovery.

Chapter 3 presented direct application of deep learning to learn chemistry from large computational materials dataset to build state-of-the-art predictive model for materials property prediction. In Chapter 4, a deep residual regression framework is presented for vector inputs which can be applied to different scientific applications and also modified and used for classification tasks. Chapter 5 demonstrated how we can leverage deep transfer learning on limited experimental observations from pretrained model on large computational datasets for building robust and highly accurate predictive models for scientific applications. Chapter 6 presents methodology for building a multi-output regression deep convolutional network that can optimize for domain specific loss function. Chapter 7 presents how we can design a deep residual network that can automatically handle background noise in X-ray diffraction patterns using deep learning such that the prediction is independent of presence of background noise.

8.2. Future Works

8.2.1. Activations as Materials Representations for Predictive Modeling

In Chapter 5, Logistic Regression was used for metallic vs insulator and magnetic vs non-magnetic classification by using activations from ElemNet as the model inputs and the results demonstrated superior performance compared to using raw elemental composition as the model inputs. Hence, leveraging materials representation learned by ElemNet for property predictions would an interesting future direction to explore.

8.2.2. IRNet for Classification

Chapter 4 presented a deep residual regression framework for materials property prediction. An important future direction would be to modify the IRNet architecture for classification by adding as many outputs at output layer as the number of classes and using Softmax function as the activation. With this simple modification, IRNet can be used for classification tasks in scientific applications where the input is a vector.

8.2.3. Predicting Crystal Structure from Composition

Since properties of materials depends on their crystal structure, knowing the crystal structure is critical for accurately predicting materials properties; DFT computations need materials crystal structure as the input to compute their properties. An important future line of work would be to learn to predict crystal structure given composition. This can be addressed as a multi-labeling problem since a compound can exist in multiple crystal structure given the same composition.

8.2.4. Multi-Property Prediction

All the works in Chapter 3, 4 and 5 involved designing deep neural network architectures for single property prediction. The presented architectures can be modified to predict multiple property at the same time; the advantage of this approach would be lower training and prediction time and memory consumption. This can be good future direction to follow.

8.2.5. ElemNet for Classification

Chapter 3 presented how to directly learn to predict materials property from elemental composition and developed the state-of-the-art model for materials property given composition type of tasks. The ElemNet architecture can be modified for classification tasks by using a Softmax activation at the output layer; this will have many scientific applications such as classification of magnetic vs non-magnetic, metallic vs insulator.

8.2.6. Transfer Learning on Experimental EBSD

In Chapter 6, a deep convolutional neural network was designed and trained on simulation datasets. Simulation datasets are large in size and does not contain any noise; however, experimental datasets are smaller in size and contain real-world noise coming from different environmental and experimental settings during experiments. Chapter 5 presented how one can leverage together computational and experimental datasets together to build more robust predictive models. Similar approach of transfer learning from simulation EBSD patterns to experimental EBSD patterns can be leveraged together to build more accurate predictive models that be used with experimental EBSD patterns.

8.2.7. Peak Area Detection Network on Larger XRD Datasets

In Chapter 7, PADNet was presented with its peak area detection component; the model was trained and tested using an unbalanced limited dataset of 177 samples. The PADNet for both SLAC and NIST datasets could predict the phase regions from the original 2D XRD pattern independent of the removal of background noise. The background noise in SLAC is highly irregular while the background noise in the case of NIST was small and almost regular for all XRD samples for the given system. The background noise pattern would be different for other alloy systems. Also, we know that deep neural networks perform better with increase in training data size. Therefore, it would be interesting in the future to train and evaluate the effectiveness of PADNet for other materials systems, collected using other type of diffractometers. As more labelled data becomes available, exploring PADNet would be interesting since it can directly predict the phase region labels without any need for background removal and preprocessing to 1D patterns.

References

[1] Logan Ward, Alexander Dunn, Alireza Faghaninia, Nils ER Zimmermann, Saurabh Bajaj, Qi Wang, Joseph Montoya, Jiming Chen, Kyle Bystrom, Maxwell Dylla, et al. Matminer: An open source toolkit for materials data mining. *Computational Materials Science*, 152:60–69, 2018.

[2] Deep learning, October 2017.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[5] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[6] Quoc V Le. Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8595–8598. IEEE, 2013.

[7] Adam Coates, Andrej Karpathy, and Andrew Y Ng. Emergence of object-selective features in unsupervised feature learning. In *Advances in Neural Information Processing Systems*, pages 2681–2689, 2012.

[8] Gregory B Olson. Computational design of hierarchically structured materials. *Science*, 277(5330):1237–1242, 1997.

[9] Ankit Agrawal and Alok Choudhary. Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science. *APL Materials*, 4(5):053208, 2016.

[10] Ian M Watt. *The principles and practice of electron microscopy*. Cambridge University Press, 1997.

[11] Bernard Borie. X-ray diffraction in crystals, imperfect crystals, and amorphous bodies. *Journal of the American Chemical Society*, 87(1):140–141, 1965.

[12] Derek Steele. Infrared spectroscopy: theory. *Handbook of vibrational spectroscopy*, 2006.

[13] Bryce Meredig, Ankit Agrawal, Scott Kirklin, James E Saal, JW Doak, A Thompson, Kunpeng Zhang, Alok Choudhary, and Christopher Wolverton. Combinatorial screening for new materials in unconstrained composition space with machine learning. *Physical Review B*, 89(9):094104, 2014.

[14] Logan Ward, Ankit Agrawal, Alok Choudhary, and Christopher Wolverton. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Computational Materials*, 2:16028, 2016.

[15] Dezhen Xue, Prasanna V Balachandran, John Hogden, James Theiler, Deqing Xue, and Turab Lookman. Accelerated search for materials with targeted properties by adaptive design. *Nature communications*, 7, 2016.

[16] Venkatesh Botu and Rampi Ramprasad. Adaptive machine learning framework to accelerate ab initio molecular dynamics. *International Journal of Quantum Chemistry*, 115(16):1074–1083, 2015.

[17] Ankit Agrawal, Parijat D Deshpande, Ahmet Cecen, Gautham P Basavarsu, Alok N Choudhary, and Surya R Kalidindi. Exploration of data science techniques to predict fatigue strength of steel from composition and processing parameters. *Integrating Materials and Manufacturing Innovation*, 3(1):1–19, 2014.

[18] Ruoqian Liu, Abhishek Kumar, Zhengzhang Chen, Ankit Agrawal, Veera Sundararaghavan, and Alok Choudhary. A predictive machine learning approach for microstructure optimization and materials design. *Scientific reports*, 5, 2015.

[19] HKDH Bhadeshia, RC Dimitriu, S Forsik, JH Pak, and JH Ryu. Performance of neural networks in materials science. *Materials Science and Technology*, 25(4):504–510, 2009.

[20] Edward O Pyzer-Knapp, Kewei Li, and Alan Aspuru-Guzik. Learning from the harvard clean energy project: The use of neural networks to accelerate materials discovery. *Advanced Functional Materials*, 25(41):6495–6502, 2015.

[21] Edward O Pyzer-Knapp, Gregor N Simm, and Alán Aspuru Guzik. A bayesian approach to calibrating high-throughput virtual screening results and application to organic photovoltaic materials. *Materials Horizons*, 3(3):226–233, 2016.

[22] Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole von Lilienfeld. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics*, 15(9):095003, 2013.

[23] Bobby G Sumpter and Donald W Noid. On the design, analysis, and characterization of materials using computational neural networks. *Annual Review of Materials Science*, 26(1):223–277, 1996.

[24] Y Sun, WD Zeng, YQ Zhao, YL Qi, X Ma, and YF Han. Development of constitutive relationship model of ti600 alloy using artificial neural network. *Computational Materials Science*, 48(3):686–691, 2010.

[25] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[26] Léon Bottou. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8), 1991.

[27] Juan José Rodriguez, Ludmila I Kuncheva, and Carlos J Alonso. Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1619–1630, 2006.

[28] Ankit Agrawal, Bryce Meredig, Chris Wolverton, and Alok Choudhary. A formation energy predictor for crystalline materials using ensemble data mining. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW) Demo*. IEEE, 2016.

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[31] Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

[32] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657, 2015.

[33] Oswald Kubaschewski and Warwick Slough. Recent progress in metallurgical thermochemistry. *Progress in Materials Science*, 14(1):3–54, 1969.

[34] Oswald Kubaschewski, Charles B Alcock, and PJ Spencer. *Materials Thermochemistry. Revised*. 1993.

[35] H Bracht, NA Stolwijk, and H Mehrer. Properties of intrinsic point defects in silicon determined by zinc diffusion experiments under nonequilibrium conditions. *Physical Review B*, 52(23):16542, 1995.

[36] Stephen R Turns. Understanding nox formation in nonpremixed flames: experiments and modeling. *Progress in Energy and Combustion Science*, 21(5):361–385, 1995.

[37] Blas P Uberuaga, Michael Leskovar, Arthur P Smith, Hannes Jónsson, and Marjorie Olmstead. Diffusion of ge below the si (100) surface: Theory and experiment. *Physical review letters*, 84(11):2441, 2000.

[38] JA Van Vechten and CD Thurmond. Comparison of theory with quenching experiments for the entropy and enthalpy of vacancy formation in si and ge. *Physical Review B*, 14(8):3551, 1976.

[39] Walter Kohn. Nobel lecture: Electronic structure of matterwave functions and density functionals. *Reviews of Modern Physics*, 71(5):1253, 1999.

[40] Jürgen Hafner, Christopher Wolverton, and Gerbrand Ceder. Toward computational materials design: the impact of density functional theory on materials research. *MRS bulletin*, 31(9):659–668, 2006.

[41] James E Saal, Scott Kirklin, Muratahan Aykol, Bryce Meredig, and Christopher Wolverton. Materials design and discovery with high-throughput density functional theory: the open quantum materials database (oqmd). *Jom*, 65(11):1501–1509, 2013.

[42] Scott Kirklin, James E Saal, Bryce Meredig, Alex Thompson, Jeff W Doak, Muratahan Aykol, Stephan Rühl, and Chris Wolverton. The open quantum materials database (oqmd): assessing the accuracy of dft formation energies. *npj Computational Materials*, 1:15010, 2015.

[43] Stefano Curtarolo, Wahyu Setyawan, Shidong Wang, Junkai Xue, Kesong Yang, Richard H. Taylor, Lance J. Nelson, Gus L.W. Hart, Stefano Sanvito, Marco Buongiorno-Nardelli, Natalio Mingo, and Ohad Levy. AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations. *Computational Materials Science*, 58:227–235, jun 2012.

[44] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *Apl Materials*, 1(1):011002, 2013.

[45] NoMaD. http://nomad-repository.eu/cms/.

[46] Tony Hey, Stewart Tansley, Kristin M Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.

[47] Krishna Rajan. Materials informatics: The materials "gene" and big data. *Annual Review of Materials Research*, 45:153–169, 2015.

[48] Joanne Hill, Gregory Mulholland, Kristin Persson, Ram Seshadri, Chris Wolverton, and Bryce Meredig. Materials science with large-scale data and informatics: unlocking new opportunities. *Mrs Bulletin*, 41(5):399–409, 2016.

[49] Logan Ward and Chris Wolverton. Atomistic calculations and materials informatics: A review. *Current Opinion in Solid State and Materials Science*, 21(3):167–176, 2017.

[50] Rampi Ramprasad, Rohit Batra, Ghanshyam Pilania, Arun Mannodi-Kanakkithodi, and Chiho Kim. Machine learning in materials informatics: recent applications and prospects. *npj Computational Materials*, 3(1):54, 2017.

[51] Zachary D Pozun, Katja Hansen, Daniel Sheppard, Matthias Rupp, Klaus-Robert Müller, and Graeme Henkelman. Optimizing transition states via kernel-based machine learning. *The Journal of chemical physics*, 136(17):174101, 2012.

[52] Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics, Focus Issue, Novel Materials Discovery*, 2013. to appear.

[53] Aaron Gilad Kusne, Tieren Gao, Apurva Mehta, Liqin Ke, Manh Cuong Nguyen, Kai-Ming Ho, Vladimir Antropov, Cai-Zhuang Wang, Matthew J Kramer, Christian

Long, et al. On-the-fly machine-learning for high-throughput experiments: search for rare-earth-free permanent magnets. *Scientific reports*, 4:6367, 2014.

[54] Michael Fernandez, Peter G Boyd, Thomas D Daff, Mohammad Zein Aghaji, and Tom K Woo. Rapid and accurate machine learning recognition of high performing metal organic frameworks for co2 capture. *The journal of physical chemistry letters*, 5(17):3056–3060, 2014.

[55] Chiho Kim, Ghanshyam Pilania, and Ramamurthy Ramprasad. From organized high-throughput data to phenomenological theory using machine learning: the example of dielectric breakdown. *Chemistry of Materials*, 28(5):1304–1311, 2016.

[56] Felix A Faber, Alexander Lindmaa, O Anatole Von Lilienfeld, and Rickard Armiento. Machine learning energies of 2 million elpasolite (a b c 2 d 6) crystals. *Physical review letters*, 117(13):135502, 2016.

[57] Anton O Oliynyk, Erin Antono, Taylor D Sparks, Leila Ghadbeigi, Michael W Gaultois, Bryce Meredig, and Arthur Mar. High-throughput machine-learning-driven synthesis of full-heusler compounds. *Chemistry of Materials*, 28(20):7324–7331, 2016.

[58] Paul Raccuglia, Katherine C Elbert, Philip DF Adler, Casey Falk, Malia B Wenny, Aurelio Mollo, Matthias Zeller, Sorelle A Friedler, Joshua Schrier, and Alexander J Norquist. Machine-learning-assisted materials discovery using failed experiments. *Nature*, 533(7601):73–76, 2016.

[59] Logan Ward, Ruoqian Liu, Amar Krishna, Vinay I Hegde, Ankit Agrawal, Alok Choudhary, and Chris Wolverton. Including crystal structure attributes in machine learning models of formation energies via voronoi tessellations. *Physical Review B*, 96(2):024104, 2017.

[60] Olexandr Isayev, Corey Oses, Cormac Toher, Eric Gossett, Stefano Curtarolo, and Alexander Tropsha. Universal fragment descriptors for predicting properties of inorganic crystals. *Nature communications*, 8:15679, 2017.

[61] Fleur Legrain, Jesús Carrete, Ambroise van Roekeghem, Stefano Curtarolo, and Natalio Mingo. How chemical composition alone can predict vibrational free energies and entropies of solids. *Chemistry of Materials*, 29(15):6220–6227, 2017.

[62] Valentin Stanev, Corey Oses, A Gilad Kusne, Efrain Rodriguez, Johnpierre Paglione, Stefano Curtarolo, and Ichiro Takeuchi. Machine learning modeling of superconducting critical temperature. *npj Computational Materials*, 4(1):29, 2018.

[63] Atsuto Seko, Hiroyuki Hayashi, Keita Nakayama, Akira Takahashi, and Isao Tanaka. Representation of compounds for machine-learning prediction of physical properties. *Physical Review B*, 95(14):144110, 2017.

[64] Maarten De Jong, Wei Chen, Randy Notestine, Kristin Persson, Gerbrand Ceder, Anubhav Jain, Mark Asta, and Anthony Gamst. A statistical learning framework for materials science: application to elastic moduli of k-nary inorganic polycrystalline compounds. *Scientific reports*, 6:34256, 2016.

[65] Eric W Bucholz, Chang Sun Kong, Kellon R Marchman, W Gregory Sawyer, Simon R Phillpot, Susan B Sinnott, and Krishna Rajan. Data-driven model for estimation of friction coefficient via informatics methods. *Tribology Letters*, 47(2):211–221, 2012.

[66] KT Schütt, H Glawe, F Brockherde, A Sanna, KR Müller, and EKU Gross. How to represent crystal structures for machine learning: Towards fast prediction of electronic properties. *Physical Review B*, 89(20):205118, 2014.

[67] Felix Faber, Alexander Lindmaa, O Anatole von Lilienfeld, and Rickard Armiento. Crystal structure representations for machine learning models of formation energies. *International Journal of Quantum Chemistry*, 115(16):1094–1101, 2015.

[68] Luca M Ghiringhelli, Jan Vybiral, Sergey V Levchenko, Claudia Draxl, and Matthias Scheffler. Big data of materials science: critical role of the descriptor. *Physical review letters*, 114(10):105503, 2015.

[69] Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547, 2018.

[70] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[71] Simon AJ Winder and Matthew Brown. Learning local image descriptors. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[72] Pierre Moreels and Pietro Perona. Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, 73(3):263–284, 2007.

[73] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[74] Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, et al. Recent advances in deep learning for speech research at microsoft. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8604–8608. IEEE, 2013.

[75] Tomáš Mikolov, Anoop Deoras, Daniel Povey, Lukáš Burget, and Jan Černockỳ. Strategies for training large scale neural network language models. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pages 196–201. IEEE, 2011.

[76] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[77] Ahmet Cecen, Hanjun Dai, Yuksel C Yabansu, Surya R Kalidindi, and Le Song. Material structure-property linkages using three-dimensional convolutional neural networks. *Acta Materialia*, 146:76–84, 2018.

[78] Ruho Kondo, Shunsuke Yamakawa, Yumi Masuoka, Shin Tajima, and Ryoji Asahi. Microstructure recognition using convolutional neural networks for prediction of ionic conductivity in ceramics. *Acta Materialia*, 141:29–38, 2017.

[79] Julia Ling, Maxwell Hutchinson, Erin Antono, and Brian Decost. Building Data-driven Models with Microstructural Images : Generalization and Interpretability. pages 1–22.

[80] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[81] Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

[82] Kristof T Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus R Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature communications*, 8:13890, 2017.

[83] Jonathan Schmidt, Jingming Shi, Pedro Borlido, Liming Chen, Silvana Botti, and Miguel AL Marques. Predicting the thermodynamic stability of solids combining density functional theory and machine learning. *Chemistry of Materials*, 29(12):5090–5103, 2017.

[84] Ann M Deml, Ryan OHayre, Chris Wolverton, and Vladan Stevanović. Predicting density functional theory total energies and enthalpies of formation of metal-nonmetal compounds by linear regression. *Physical Review B*, 93(8):085142, 2016.

[85] Atsuto Seko, Hiroyuki Hayashi, Hisashi Kashima, and Isao Tanaka. Matrix- and tensor-based recommender systems for the discovery of currently unknown inorganic compounds. *Physical Review Materials*, 2(1):013805, jan 2018.

[86] Vincent Tinto. Dropout from higher education: A theoretical synthesis of recent research. *Review of educational research*, 45(1):89–125, 1975.

[87] Douglas M Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12, 2004.

[88] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[89] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer, 2012.

[90] Ilya Sutskever, James Martens, George E Dahl, and Geoffrey E Hinton. On the importance of initialization and momentum in deep learning. *ICML (3)*, 28:1139–1147, 2013.

[91] Robert A Jacobs. Increased rates of convergence through learning rate adaptation. *Neural networks*, 1(4):295–307, 1988.

[92] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, pages 1–7, 2010.

[93] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[94] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[95] Open quantum materials database. `http://oqmd.org/`.

[96] Guenter Bergerhoff, R Hundt, R Sievers, and ID Brown. The inorganic crystal structure data base. *Journal of chemical information and computer sciences*, 23(2):66–69, 1983.

[97] Sten Andersson, Bengt Collén, Ulf Kuylenstierna, and Arne Magnéli. Phase analysis studies on the titanium-oxygen system. *Acta chem. scand*, 11(10):1641–1652, 1957.

[98] FC Walsh and RGA Wills. The continuing development of magnéli phase titanium sub-oxides and ebonex® electrodes. *Electrochimica Acta*, 55(22):6342–6351, 2010.

[99] Dipendra Jha, Logan Ward, Arindam Paul, Wei-keng Liao, Alok Choudhary, Chris Wolverton, and Ankit Agrawal. ElemNet: Deep learning the chemistry of materials from only elemental composition. *Scientific reports*, 2018.

[100] P P Fedorov. Systems of Alkali and Rare-Earth Metal Fluorides. *Russ. J. Inorg. Chem.*, 44(April):1703–1727, 1999.

[101] EV Peresypkina and VA Blatov. Structure-forming components in crystals of ternary and quaternary 3d-metal complex fluorides. *Acta Crystallographica Section B*, 59(3):361–377, 2003.

[102] P.J.M. Isherwood. Copper zinc oxide: Investigation into a p-type mixed metal oxide system. *Vacuum*, 139:173 – 177, 2017.

[103] S. Benmokhtar, A. El Jazouli, J.P. Chaminade, P. Gravereau, F. Guillen, and D. de Waal. Synthesis, crystal structure and optical properties of BiMgVO5. *Journal of Solid State Chemistry*, 177(11):4175–4182, nov 2004.

[104] Etude par rayons X et neutrons de la serie isomorphe ATiTO5 (A = Cr, Mn, Fe, T = Terres Rares). *Journal of Physics and Chemistry of Solids*, 31(5):1171–1183, 1970.

[105] N M Nusran, K R Joshi, K Cho, M A Tanatar, W R Meier, S L Bud'ko, P C Canfield, Y Liu, T A Lograsso, and R Prozorov. Spatially-resolved study of the meissner effect in superconductors using nv-centers-in-diamond optical magnetometry. *New Journal of Physics*, 20(4):043010, 2018.

[106] Zhao Qin, Gang Seob Jung, Min Jeong Kang, and Markus J Buehler. The mechanics and design of a lightweight three-dimensional graphene assembly. *Science advances*, 3(1):e1601536, 2017.

[107] Surya R Kalidindi. Data science and cyberinfrastructure: critical enablers for accelerated development of hierarchical materials. *International Materials Reviews*, 60(3):150–168, 2015.

[108] Stefano Curtarolo, Gus LW Hart, Marco Buongiorno Nardelli, Natalio Mingo, Stefano Sanvito, and Ohad Levy. The high-throughput highway to computational materials design. *Nature materials*, 12(3):191–201, 2013.

[109] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin a. Persson. The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, 2013.

[110] B Blaiszik, K Chard, J Pruyne, R Ananthakrishnan, S Tuecke, and I Foster. The Materials Data Facility: Data services to advance materials science research. *JOM*, 68(8):2045–2052, 2016.

[111] Alden Dima, Sunil Bhaskarla, Chandler Becker, Mary Brady, Carelyn Campbell, Philippe Dessauw, Robert Hanisch, Ursula Kattner, Kenneth Kroenlein, Marcus Newrock, et al. Informatics infrastructure for the Materials Genome Initiative. *JOM*, 68(8):2053–2064, 2016.

[112] Logan Ward and Chris Wolverton. Atomistic calculations and materials informatics: A review. *Current Opinion in Solid State and Materials Science*, 2016.

[113] Materials genome initiative, July 2016.

[114] Arindam Paul, Dipendra Jha, Reda Al-Bahrani, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. CheMixNet: Mixed DNN architectures for predicting chemical properties using multiple molecular representations. In *Proceedings of the Workshop on Molecules and Materials at the 32nd Conference on Neural Information Processing Systems*, 2018.

[115] Quan Zhou, Peizhe Tang, Shenxiu Liu, Jinbo Pan, Qimin Yan, and Shou-Cheng Zhang. Learning atoms for materials discovery. *Proceedings of the National Academy of Sciences*, 115(28):E6411–E6417, 2018.

[116] Dipendra Jha, Saransh Singh, Reda Al-Bahrani, Wei-keng Liao, Alok Choudhary, Marc De Graef, and Ankit Agrawal. Extracting grain orientations from ebsd patterns of polycrystalline materials using convolutional neural networks. *Microscopy and Microanalysis*, 24(5):497–502, 2018.

[117] Garrett B Goh, Nathan O Hodas, Charles Siegel, and Abhinav Vishnu. SMILES2Vec: An interpretable general-purpose deep neural network for predicting chemical properties. *arXiv preprint arXiv:1712.02034*, 2017.

[118] Wouter Boomsma and Jes Frellsen. Spherical convolutions and their application in molecular modelling. In *Advances in Neural Information Processing Systems*, pages 3433–3443, 2017.

[119] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385, 2015.

[120] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[121] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[122] Yiren Wang and Fei Tian. Recurrent residual learning for sequence classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 938–943, 2016.

[123] Lu Huang, Ji Xu, Jiasong Sun, and Yi Yang. An improved residual lstm architecture for acoustic modeling. In *Computer and Communication Systems (ICCCS), 2017 2nd International Conference on*, pages 101–105. IEEE, 2017.

[124] David W Oxtoby, H Pat Gillis, and Laurie J Butler. *Principles of modern chemistry*. Cengage Learning, 2015.

[125] Katja Hansen, Franziska Biegler, Raghunathan Ramakrishnan, Wiktor Pronobis, O. Anatole Von Lilienfeld, Klaus-Robert Robert Müller, and Alexandre Tkatchenko. Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *The Journal of Physical Chemistry Letters*, 6(12):2326–2331, jun 2015.

[126] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[127] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[128] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[129] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE, 2017.

[130] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[131] Ryan Chard, Zhuozhao Li, Kyle Chard, Logan T. Ward, Yadu N. Babuji, Anna Woodard, Steven Tuecke, Ben Blaiszik, Michael J. Franklin, and Ian T. Foster. DLHub: Model and data serving for science. In *33rd IEEE International Parallel and Distributed Processing Symposium*, 2019.

[132] Anubhav Jain, Geoffroy Hautier, Shyue Ping Ong, Charles J. Moore, Christopher C. Fischer, Kristin A. Persson, and Gerbrand Ceder. Formation enthalpies by mixing gga and gga + u calculations. *Phys. Rev. B*, 84:045115, Jul 2011.

[133] Anubhav Jain, Geoffroy Hautier, Charles J Moore, Shyue Ping Ong, Christopher C Fischer, Tim Mueller, Kristin A Persson, and Gerbrand Ceder. A high-throughput infrastructure for density functional theory calculations. *Computational Materials Science*, 50(8):2295–2310, 2011.

[134] Kamal Choudhary, Gowoon Cheon, Evan Reed, and Francesca Tavazza. Elastic properties of bulk and low-dimensional materials using van der waals density functional. *Phys. Rev. B*, 98:014107, Jul 2018.

[135] Kamal Choudhary, Qin Zhang, Andrew CE Reid, Sugata Chowdhury, Nhan Van Nguyen, Zachary Trautt, Marcus W Newrock, Faical Yannick Congo, and Francesca Tavazza. Computational screening of high-performance optoelectronic materials using optb88vdw and tb-mbj formalisms. *Scientific data*, 5:180082, 2018.

[136] Kamal Choudhary, Irina Kalish, Ryan Beams, and Francesca Tavazza. High-throughput identification and characterization of two-dimensional materials using density functional theory. *Scientific Reports*, 7(1):5179, 2017.

[137] Kamal Choudhary, Brian DeCost, and Francesca Tavazza. Machine learning with force-field-inspired descriptors for materials: Fast screening and mapping energy landscape. *Phys. Rev. Materials*, 2:083801, Aug 2018.

[138] Scientific Group Thermodata Europe (SGTE) et al. Thermodynamic properties of inorganic materials. *Landolt-Boernstein New Series, Group IV*, 1999.

[139] Ankit Agrawal and Alok Choudhary. Perspective: Materials informatics and big data: Realization of the fourth paradigm of science in materials science. *APL Materials*, 4(053208):1–10, 2016.

[140] Ankit Agrawal and Alok Choudhary. Deep materials informatics: Applications of deep learning in materials science. *MRS Communications*, pages 1–14, 2019.

[141] Dipendra Jha, Logan Ward, Zijiang Yang, Christopher Wolverton, Ian Foster, Weikeng Liao, Alok Choudhary, and Ankit Agrawal. IRNet: A general purpose deep residual regression framework for materials discovery. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2385–2393. ACM, 2019.

[142] George Kim, SV Meschel, Philip Nash, and Wei Chen. Experimental formation enthalpies for intermetallic phases and other inorganic compounds. *Scientific data*, 4:170162, 2017.

[143] Anubhav Jain, Geoffroy Hautier, Shyue Ping Ong, Charles J Moore, Christopher C Fischer, Kristin A Persson, and Gerbrand Ceder. Formation enthalpies by mixing gga and gga+ u calculations. *Physical Review B*, 84(4):045115, 2011.

[144] David A Young. *Phase diagrams of the elements*. Univ of California Press, 1991.

[145] Sinno Jialin Pan, Qiang Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[146] Shin Hoo-Chang, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285, 2016.

[147] Dipendra Jha, Kamal Choudhary, Francesca Tavazza, Wei-keng Liao, Alok Choudhary, Carelyn Campbell, and Ankit Agrawal. Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning. *Nature communications*, 10(1):1–12, 2019.

[148] I Hurtado and D Neuschutz. Thermodynamic properties of inorganic materials, compiled by sgte, vol. 19, 1999.

[149] Alan Agresti. Introduction: distributions and interference for categorical data. categorical data analysis, 2002.

[150] Amemiya Takeshi. Qualitative response models. *Advanced Econometrics. Oxford: Basil Blackwell. ISBN 0-631-13345-3*, 1985.

[151] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[152] Ghanshyam Pilania, Chenchen Wang, Xun Jiang, Sanguthevar Rajasekaran, and Ramamurthy Ramprasad. Accelerating materials property predictions using machine learning. *Scientific reports*, 3:2810, 2013.

[153] Joohwi Lee, Atsuto Seko, Kazuki Shitara, Keita Nakayama, and Isao Tanaka. Prediction model of band gap for inorganic compounds by combination of density functional theory calculations and machine learning techniques. *Physical Review B*, 93(11):115104, 2016.

[154] Tianzhuo Zhan, Lei Fang, and Yibin Xu. Prediction of thermal boundary resistance by the machine learning method. *Scientific reports*, 7(1):7109, 2017.

[155] Ying Zhang and Chen Ling. A strategy to apply machine learning to small datasets in materials science. *Npj Computational Materials*, 4(1):25, 2018.

[156] Brent L. Adams, Stuart I. Wright, and Karsten Kunze. Orientation imaging: The emergence of a new microscopy. *Metallurgical Transactions A*, 24(4):819–831, 1993.

[157] Patrick G. Callahan and Marc De Graef. Dynamical electron backscatter diffraction patterns. part i: Pattern simulations. *Microscopy and Microanalysis*, 19(5):12551265, 2013.

[158] A.J. Schwartz, M. Kumar, B.L. Adams, and D.P. Field, editors. *Electron Backscatter Diffraction in Materials Science*. Springer, 2nd. edition, 2009.

[159] Yu H Chen, Se Un Park, Dennis Wei, Greg Newstadt, Michael A Jackson, Jeff P Simmons, Marc De Graef, and Alfred O Hero. A dictionary approach to electron backscatter diffraction indexing. *Microscopy and Microanalysis*, 21(3):739–752, 2015.

[160] Stuart I. Wright, Matthew M. Nowell, Scott P. Lindeman, Patrick P. Camus, Marc De Graef, and Michael A. Jackson. Introduction and comparison of new ebsd post-processing methodologies. *Ultramicroscopy*, 159:81 – 94, 2015.

[161] Saransh Singh and Marc De Graef. Dictionary indexing of electron channeling patterns. *Microscopy and Microanalysis*, 23(1):110, 2017.

[162] Katharina Marquardt, Marc De Graef, Saransh Singh, Hauke Marquardt, Anja Rosenthal, and Sanae Koizuimi. Quantitative electron backscatter diffraction (ebsd) data analyses using the dictionary indexing (di) approach: Overcoming indexing difficulties on geological materials. *American Mineralogist*, 102(9):1843, 2017.

[163] N.C. Krieger Lassen. Automatic crystal orientation determination from ebsps. *Micron and Microscopica Acta*, 6:191–192, 1992.

[164] Farangis Ram, Stuart Wright, Saransh Singh, and Marc De Graef. Error analysis of the crystal orientations obtained by the dictionary approach to ebsd indexing. *Ultramicroscopy*, 181:17 – 26, 2017.

[165] Yann LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, page 20, 2015.

[166] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in neural information processing systems*, pages 2643–2651, 2013.

[167] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[168] Kasthurirangan Gopalakrishnan, Siddhartha K. Khaitan, Alok Choudhary, and Ankit Agrawal. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, 157:322–330, 2017.

[169] Woon Bae Park, Jiyong Chung, Jaeyoung Jung, Keemin Sohn, Satendra Pal Singh, Myoungho Pyo, Namsoo Shin, and K-S Sohn. Classification of crystal structure using a convolutional neural network. *IUCrJ*, 4(4):486–494, 2017.

[170] Weizong Xu and James M LeBeau. A deep convolutional neural network to analyze position averaged convergent beam electron diffraction patterns. *Ultramicroscopy*, 188:59–69, 2018.

[171] Ruoqian Liu, Ankit Agrawal, Wei-keng Liao, Alok Choudhary, and Marc De Graef. Materials discovery: Understanding polycrystals from large-scale electron patterns. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 2261–2269. IEEE, dec 2016.

[172] Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. Miml: a framework for learning with ambiguous objects. *CORR abs/0808.3231*, 112, 2008.

[173] Anh Pham, Raviv Raich, Xiaoli Fern, and Jesús Pérez Arriaga. Multi-instance multi-label learning in the presence of novel class instances. In *International Conference on Machine Learning*, pages 2427–2435, 2015.

[174] Forrest Briggs, Xiaoli Z Fern, and Raviv Raich. Context-aware miml instance annotation. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 41–50. IEEE, 2013.

[175] Kukjin Kang, Jong-Hoon Oh, Chulan Kwon, and Youngah Park. Generalization in a two-layer neural network with multiple outputs. *Physical Review E*, 54(2):1811, 1996.

[176] Anton Bezuglov, Brian Blanton, and Reinaldo Santiago. Multi-output artificial neural network for storm surge prediction in north carolina. *arXiv preprint arXiv:1609.07378*, 2016.

[177] Ning An, Weigang Zhao, Jianzhou Wang, Duo Shang, and Erdong Zhao. Using multi-output feedforward neural network with empirical mode decomposition based signal filtering for electricity demand forecasting. *Energy*, 49:279–288, 2013.

[178] S. Singh and M. De Graef. Orientation sampling for dictionary-based diffraction pattern indexing methods. *Modeling and Simulations in Materials Science and Engineering*, 24:085013, 2016.

[179] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[180] Michael M Woolfson and Michael Mark Woolfson. *An introduction to X-ray crystallography*. Cambridge University Press, 1997.

[181] Harold P Klug and Leroy E Alexander. X-ray diffraction procedures: for polycrystalline and amorphous materials. *X-Ray Diffraction Procedures: For Polycrystalline and Amorphous Materials, 2nd Edition, by Harold P. Klug, Leroy E. Alexander, pp. 992. ISBN 0-471-49369-4. Wiley-VCH, May 1974.*, page 992, 1974.

[182] Duane Milton Moore and Robert C Reynolds. *X-ray Diffraction and the Identification and Analysis of Clay Minerals*, volume 322. Oxford university press Oxford, 1989.

[183] David L Bish and Jeffrey Edward Post. *Modern powder diffraction*, volume 20. Mineralogical Society of America Washington, DC, 1989.

[184] BD Cullity. Elements of xrd diffraction, addition-wesley. *Reading, MA*, 1978.

[185] Jin-Seok Chung and Gene E Ice. Automated indexing for texture and strain measurement with broad-bandpass x-ray microbeams. *Journal of applied physics*, 86(9):5249–5255, 1999.

[186] Hideomi Koinuma and Ichiro Takeuchi. Combinatorial solid-state chemistry of inorganic materials. *Nature materials*, 3(7):429, 2004.

[187] JM Gregoire, DG Van Campen, CE Miller, RJR Jones, SK Suram, and A Mehta. High-throughput synchrotron x-ray diffraction for combinatorial phase mapping. *Journal of synchrotron radiation*, 21(6):1262–1268, 2014.

[188] Mit cmse x-ray diffraction facility, 2019.

[189] Yvonne M Mos, Arnold C Vermeulen, Cees JN Buisman, and Jan Weijma. X-ray diffraction of iron containing samples: The importance of a suitable configuration. *Geomicrobiology Journal*, 35(6):511–517, 2018.

[190] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[191] Robert E Dinnebier. Fwhm optimized polynomial smoothing filters: A practical approach. *Powder Diffraction*, 18(3):199–204, 2003.

[192] Sergio Brückner. Estimation of the background in powder diffraction patterns through a robust smoothing procedure. *Journal of Applied Crystallography*, 33(3-2):977–979, 2000.

[193] Melkon Tatlier. Artificial neural network methods for the prediction of framework crystal structures of zeolites from xrd data. *Neural Computing and Applications*, 20(3):365–371, 2011.

[194] Christopher J Gilmore, Gordon Barr, and Jonathan Paisley. High-throughput powder diffraction. i. a new approach to qualitative and quantitative powder diffraction pattern analysis using full pattern profiles. *Journal of applied crystallography*, 37(2):231–242, 2004.

[195] Jonathan Kenneth Bunn, Jianjun Hu, and Jason R Hattrick-Simpers. Semi-supervised approach to phase identification from combinatorial sample diffraction patterns. *JOM*, 68(8):2116–2125, 2016.

[196] Jason R Hattrick-Simpers, John M Gregoire, and A Gilad Kusne. Perspective: Composition–structure–property mapping in high-throughput experiments: Turning data into knowledge. *APL Materials*, 4(5):053211, 2016.

[197] Yuma Iwasaki, A Gilad Kusne, and Ichiro Takeuchi. Comparison of dissimilarity measures for cluster analysis of x-ray diffraction data from combinatorial libraries. *npj Computational Materials*, 3(1):4, 2017.

[198] NIST X-ray diffraction data acquired using a Bruker D8.

[199] Michael A Bauer, Alain Biem, Stewart McIntyre, and Yuzhen Xie. A pipelining implementation for parsing x-ray diffraction source data and removing the background noise. In *Journal of Physics: Conference Series*, volume 256, page 012017. IOP Publishing, 2010.

[200] Michael A Bauer, Alain Biem, Stewart McIntyre, Nobumichi Tamura, and Yuzhen Xie. High-performance parallel and stream processing of x-ray microdiffraction data on multicores. In *Journal of Physics: Conference Series*, volume 341, page 012025. IOP Publishing, 2012.