Assessing Structure–Property Relationships of Crystal Materials using Deep Learning

Zheng Li

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Masters of Science

in

Computer Science & Application

Anuj Karpatne, Chair

Hongliang Xin

Bimal Viswanath

July 2, 2020

Blacksburg, Virginia

Keywords: Density functional theory, Material design, Machine learning

Assessing Structure–Property Relationships of Crystal Materials using Deep Learning

Zheng Li

(ABSTRACT)

In recent years, deep learning technologies have received huge attention and interest in the field of high-performance material design. This is primarily because deep learning algorithms in nature have huge advantages over the conventional machine learning models in processing massive amounts of unstructured data with high performance. Besides, deep learning models are capable of recognizing the hidden patterns among unstructured data in an automatic fashion without relying on excessive human domain knowledge. Nevertheless, constructing a robust deep learning model for assessing materials' structure-property relationships remains a non-trivial task due to highly flexible model architecture and the challenge of selecting appropriate material representation methods. In this regard, we develop advanced deep-learning models and implement them for predicting the quantum-chemical calculated properties (i.e., formation energy) for an enormous number of crystal systems. Chapter 1 briefly introduces some fundamental theory of deep learning models (i.e., CNN, GNN) and advanced analysis methods (i.e., saliency map). In Chapter 2, the convolutional neural network (CNN) model is established to find the correlation between the physically intuitive partial electronic density of state (PDOS) and the formation energies of crystals. Importantly, advanced machine learning analysis methods (i.e., salience mapping analysis) are utilized to shed light on underlying physical factors governing the energy properties. In Chapter 3, we introduce the methodology of implementing the cutting-edge graph neural networks (GNN) models for learning an enormous number of crystal structures for the desired properties.

Assessing Structure–Property Relationships of Crystal Materials using Deep Learning

Zheng Li

(GENERAL AUDIENCE ABSTRACT)

Machine learning technologies, particularly deep learning, have demonstrated remarkable progress in facilitating the high-throughput materials discovery process. In essence, machine learning algorithms have the ability to uncover the hidden patterns of data and make appropriate decisions without being explicitly programmed. Nevertheless, implementing machine learning models in the field of material design remains a challenging task. One of the biggest limitations is our insufficient knowledge about the structure-property relationships for material systems. As the performance of machine learning models is to a large degree determined by the underlying material representation method, which typically requires the experts to have in-depth knowledge of the material systems. Thus, designing effective feature representation methods is always the most crucial aspect for machine learning model development and the process takes a significant amount of manual effort. Even though tremendous efforts have been made in recent years, the research process for robust feature representation methods is still slow. In this regard, we attempt to automate the feature engineering process with the assistance of advanced deep learning algorithms. Unlike the conventional machine learning models, our deep learning models (i.e., convolutional neural networks, graph neural networks) are capable of processing massive amounts of structured data such as spectrum and crystal graphs. Specifically, the deep learning models are explicitly designed to learn the hidden latent variables that are contained in crystal structures in an automatic fashion and provide accurate prediction results. We believe the deep learning models have huge potential to simplify the machine learning modeling process and facilitate the discovery of promising functional materials.

Acknowledgments

First of all, I would like to express my sincere gratitude and appreciation to my advisor Prof. Anuj Karpatne, for his continuous guidance and support of my M.S. research and study. More importantly, thanks to my advisor for leading me to this promising field of machine learning and I believe the valuable knowledge I gained would have a huge impact on my future career. Besides my advisor, I would like to thank the rest of my committee: Prof. Hongliang Xin and Prof. Bimal Viswanath, for their helpful suggestions and recommendations. All the suggestions are great driving forces for me to make continuous improvement. My sincere thanks also go to all my workmates in Anuj's group, who provided me technical assistance and ideas during my research and studies. Also, I would like to thank the Department of Computer Science and Advanced Research Computing (ARC) at Virginia Tech, for providing me with the extraordinary educational and computational resources to complete my study and research. Last but not the least, I would like to express my deepest gratitude to my parents Chengqing Li and Guifeng Ba, for giving birth to me and supporting me both spiritually and physically throughout my life.

Contents

List of Figures

Chapter 1

Machine learning in functional material design

1.1 Introduction

An important objective of modern scientists is to discover novel functional materials with desired characteristics and incorporate the materials into the existing devices to optimize general performance. Advanced functional materials have wide applications and play an essential role in every historical period of human beings. The development of high-performance functional materials has helped expand the limits of human endeavor and achievement. In 21^{st} century, we have seen a growing and rapid demand for advanced material in industries such as aerospace, automotive, electronics, etc. For instance, the composite materials are increasingly being used in cars and wind turbine blades for lightweight and enhanced properties.[1] The emergence of Graphene type materials completely revolutionized the modern industrial design of smartphones. Ceramics are becoming the most in-demand advanced functional materials for manufacturing capacitors in the application of electrical and electronic devices. Nanomaterials and conductive composites polymers are considered as one of the most researched material categories in the domains applications of automotive and aerospace industries. Energy-related materials including semiconductors and metal-organic complexes are crucial components in energy conversion and storage devices, thus ensuring

a sustainable energy economy.[2] Biomolecular materials play an enormous role in a variety of applications in the healthcare industries such as medical diagnostics, drug delivery, and etc.[3]

Due to structural versatility and vast anticipated chemical space, designing novel functional material is always considered as an extremely difficult task. Even though the tremendous effort has been made by the researchers in the last few decades to push the frontiers of material properties to more extreme levels to meet the rapidly growing demand for various applications. Nevertheless, the conventional trial-and-error experimentation relying on human's expert knowledge and experiences is still the mainstream approach at present, which significantly limits the material development speed. Machine learning approaches, since emerged, have revolutionized a range of traditional mathematical modeling strategies for material design, which has been adapted for decades. Compared to the conventional knowledge-based modeling approaches, machine learning algorithms provide the promise to establish accurate structure-property relationships of a wide range of materials without relying on extensive human knowledge. In recent years, machine learning methods, namely the quantitative structure-activity relationships (QSARs) models, are widely used by researchers to facilitate the discovery process of high-performance materials. As traditional trial-and-error experiments or computational modeling approaches, i.e., first-principle calculations and molecular dynamics, often cost tremendous time and resources of researchers. Thus, there is always growing demand for alternative less expensive approaches that can be applied in the search of broader materials space for material structures with promising properties. Even though machine learning methods provide a new means of screening new materials rapidly and accurately, it still requires some degree of intelligence and thoughts based on expertise in fundamental material structures. In another word, developing novel material representation methods is the key to the success of machine learning modeling, however, it remains the

most challenging step. Therefore, coming up with a robust material representation method is always the central theorem of current machine learning study.

Recent years have witnessed much important progress in automating the machine learning modeling process in terms of model selection, hyper-parameter tuning. For instance, an integrated machine learning framework, namely TPOT is developed to search for most appropriate models and the corresponding hyper-parameters in an intelligent and efficient way.[4] Nevertheless, very little effort has been devoted to automating feature engineering, which is the most important aspect of probably every machine learning pipeline. In this regard, we develop a high-level automated feature engineering scheme in the domain application of functional material design using deep learning models. More specifically, we apply a deep learning algorithm to learn latent vectors for representing the material structure in a fully automatic fashion. The promise of automated feature engineering is to surpass the limitations of manual feature engineering by implementing advanced deep learning algorithms, which subsequently facilitating the high-throughput material screening process. In this chapter, we briefly introduce some important techniques that are used in the high-throughput material design and fundamental theory of some advanced deep learning algorithms such as convolutional neural networks (CNNs) and graph neural networks (GNNs). Understanding the basic concepts and knowledge are extremely essential for us to have a clear overview of the concept of automatic feature engineering.

1.2 Quantum chemical simulations

It is noteworthy that the primary research focus of this work is to predict the expensive quantum-chemical calculated properties using deep learning algorithms. Density functional theory (DFT),[5] a quantum mechanical simulation method, has attracted wide attention in

the field of physical science. DFT has gained immense popularity in recent years because it provides an efficient means for assessing the ground-state electronic structures of materials prior to the actual experiments. The revolutionary development of the DFT method is largely attributed to the rapid advancement of high-performance computing (HPC) technology and the theoretical algorithm. Over the last few decades, the DFT simulation approach has revolutionized the theoretical research of surface science, which leverages an atomic-scale understanding of material properties in an extremely efficient manner. In this chapter, we introduce the essential concepts and the corresponding implementation methodology of DFT for practical computation.

The modern DFT theory is primarily based on the Kohn-Sham paradigm.[6] Notably, the Kohn-Sham DFT (KS-DFT) method has made tremendous contributions by representing the multi-electron wave-function with a non-interaction system. Within the non-interaction system, the electrons of atoms are affected by a simultaneous effect from the external potential and the Coulomb interactions between the electrons (i.e., exchange and correlation interactions). Therefore, the Kohn-Sham framework allows us to describe the multi-electron wave-function with a Slater determinant of orbitals, which makes it possible to approximate the kinetic energy functional. Note that the explicit rigorous solution for the energy term of exchange-correlation interactions remains unknown for the KS-DFT method and the Local-Density Approximation (LDA) based on the uniform electron gas model by Thomas-Fermi theory[7] is widely applied to approximate the exchange-correlation energy. To data, the first-principle DFT is becoming a well known computational tool, which provides a trade-off between computational cost and accuracy. According to the Kohn-Sham scheme, the ground state energy term can be written as:

$$E(n(r)) = \int v_{ext}(r)n(r)dr + T_0[n(r)] + \frac{1}{2}\int\int \frac{n(r)n(r')}{|r-r'|}drdr' + \int n(r)\varepsilon_{xc}(n(r))dr, \quad (1.1)$$

where $v_{ext}(r)$ denotes the external potential from the nuclei. In addition, $n(r)$ represents the electronic density function in 3-dimensional spatial space and $\varepsilon_{xc}(n(r))$ denotes the per-electron exchange and correlation energy. Typically, the exchange and correlation energy can be approximated by the Local Density Approximation (LDA)[8] or more advanced Generalized gradient approximations(GGA)[9]. The ground state energy term can be understood as a summation of the energy contributions from the external potential energy, the non-interacting kinetic energy, the classical Coulomb interaction, and the exchange-correlation interaction. Therefore, the central theorem of KS-DFT is about finding an appropriate electron density for the ground state energy. In the practical implementation, DFT solves the ground state electronic properties of a system with an iterative self-consistency loop. Within the self-consistency loop, the electron density $n(r)$ is propagated through the process until the change of electron density is smaller than a specific threshold value. Figure 1.1 shows the schematic display of the DFT self-consistency mechanism.

1.3 Deep learning models

1.3.1 Artificial neural networks

Feedforward neural networks

Artificial neural network (ANN) is a computational algorithm inspired by the mammalian biological neurons.[10] Typically, the neural networks are organized by layers. Each layer is

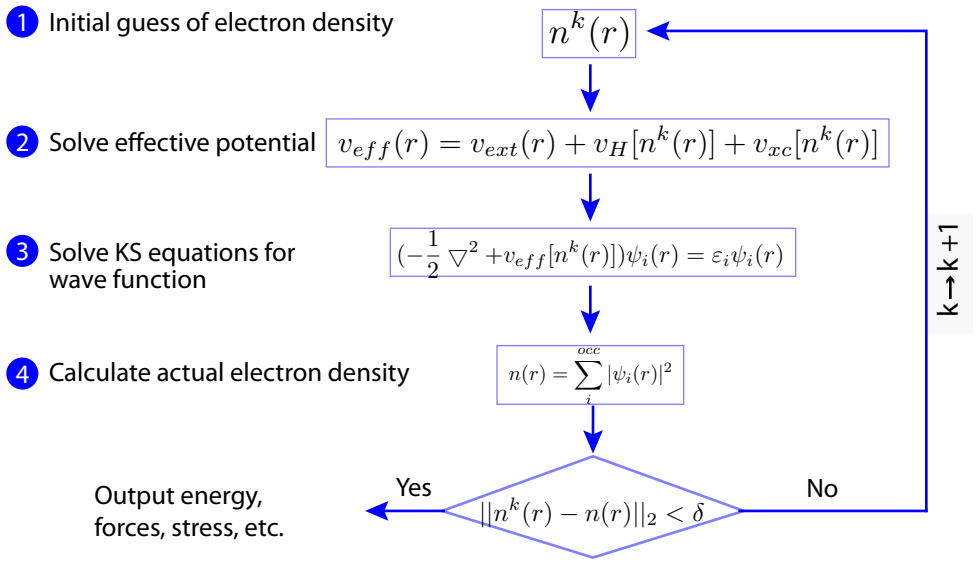

Figure 1.1: Schematic of DFT self-consistency loop.

implemented with a number of interconnected nodes, which are processing units, namely activation functions. The activation functions are a crucial component of an ANN model. The activation functions have a major effect on model training performance such as convergence efficiency and prediction accuracy. Essentially, the activation functions are mathematical equations that calculate the output values corresponding to a series of inputs from the previous layer. The selection of appropriate activation functions is of vital importance for the ANN learning. For instance, the nonlinear sigmoid activation function is designed to normalize the outputs of each neuron to a range between 0 and 1, thus ensuring computationally efficiency for the regression tasks. Besides, the rectified linear unit (ReLU) activation function,[11] known as a ramp function, is introduced to enable better training performance for a range of deep learning models, i.e., ANNs, CNNs. Figure 1.2 below shows an example of simple feed-forward ANN architecture.

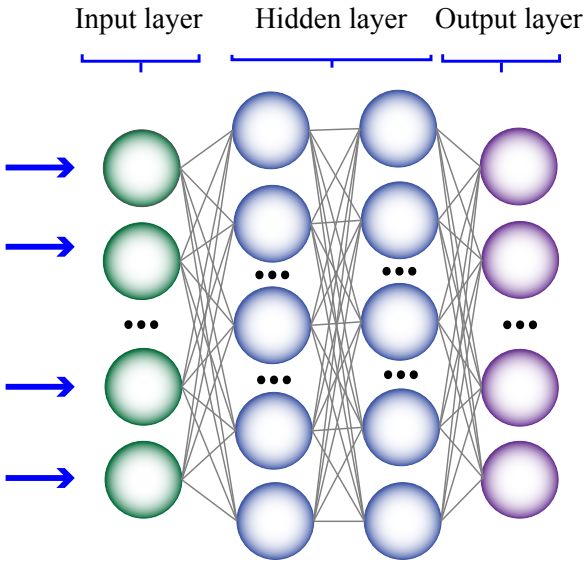

Figure 1.2: Feed-forward artificial neural network structure. The nodes represent the activation function and the edges represent the weight parameters.

Recurrent neural networks

Notably, deep learning refers to a branch of machine learning algorithms using different types of neural network architectures. Other advanced ANN models such as long short term memory (LSTM),[12] a special category of recurrent neural networks (RNNs), have gained a lot of popularity in recent years for the application of time-series forecasting. Unlike the ANN model, one of the appeals of the RNN is that it is explicitly designed to remember the information for long periods of time enabled by the chain of repeating modules. More specifically, the repeating modules in RNNs save the output value of a particular neural and send the output value to another neuron in the same hidden layer during the training process. Because of this special working mechanism, the RNNs show superior predictive performance over the other conventional ANNs algorithms in modeling univariate time series forecasting problems. Figure 1.3 below shows the basic concept of the standard RNN model structure.

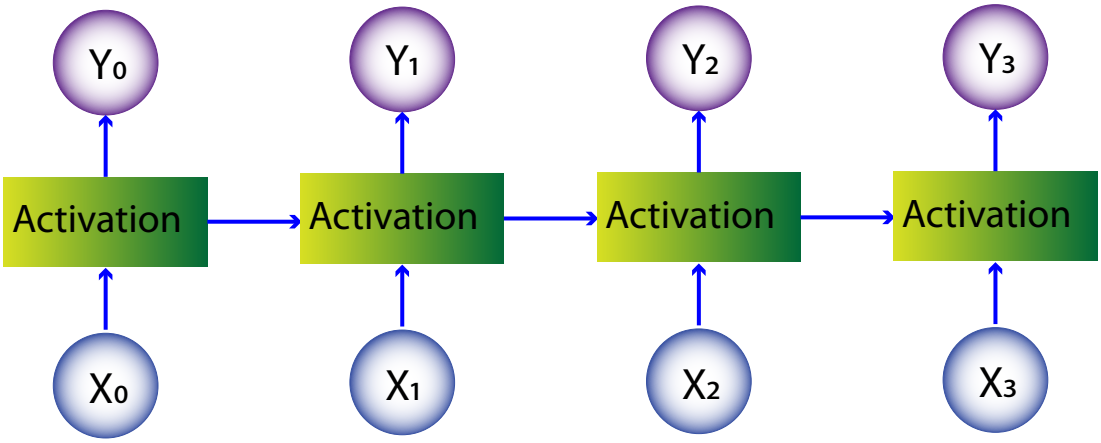

Figure 1.3: Schematic of recurrent neural network structure.

Back-propagation

Back-propagation plays an extremely important role in any ANN models. Essentially, back-propagation is the practice of fine-tuning a large number of weight parameters of ANNs according to the error rate at each epoch of training iteration. Here, the loss value is calculated by taking the difference between the model predictions with the true values. The loss functions are varied according to learning objectives. For instance, the Mean Squared Error (MSE) is widely used for the regression tasks while the Binary Cross-Entropy (BCE) works tremendously well for binary classification problems. During the actual training process, the partial derivative of the error at each epoch is calculated with respect to each individual weight parameter through a series of chain rules and this procedure is repeated until the optimal error value is stabilized under a certain threshold value or reaches a constant value. In brief, the optimal goal of back-propagation is to find a proper set of weight parameters to ensure a lower loss value, thus improving the generalization performance. To illustrate the theory of back-propagation, we derive the mathematical equations for a simple network

structure. Figure 1.4 describes a simple 2-layer multiple output ANN structure. The sigmoid activation functions are used for all the neurons on both hidden layers (i.e,, h_1, h_2) and the output layers (i.e., Out_1, Out_2). Our objective is to calculate the gradient of the total loss (i.e., E_{out1}, E_{out2}) with respect to the weight w_5.

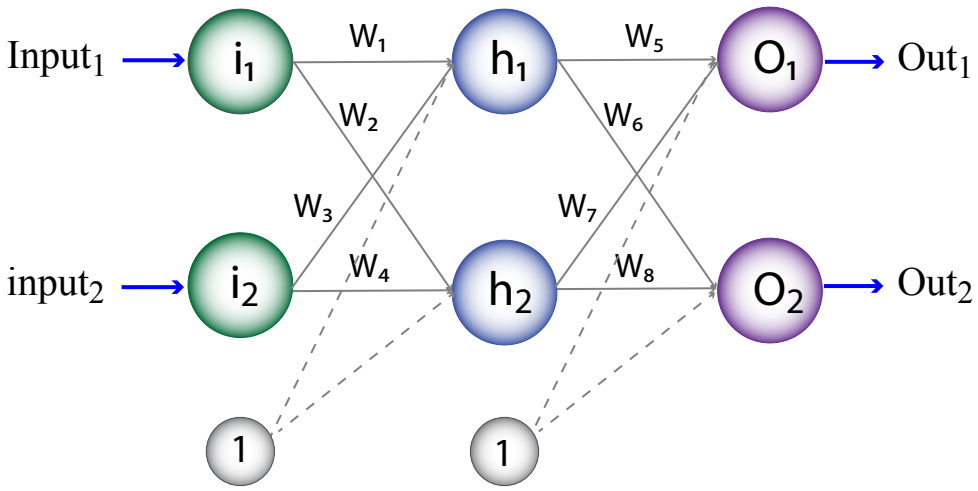

Figure 1.4: Schematic of multiple-output neural network structure.

Equation 1 shows the RMSE (total error) of the model predictions and the true values.

$$E_{total} = E_{out1} + E_{out2} = \frac{1}{2}(Y_{out1} - Y_{target1})^2 + \frac{1}{2}(Y_{out2} - Y_{target2})^2, \tag{1.2}$$

where Y_{out1} and Y_{out2} indicate the model prediction outputs and $Y_{target1}$ and $Y_{target2}$ denote the actual value of the corresponding output values. The sigmoid activation function that is implemented for the output neuron is shown below:

$$Y_{out1} = \frac{1}{1 + exp^{-Y_{net1}}}, \tag{1.3}$$

where Y_{net1} indicates the weighted summation of all the weight parameters in the hidden layer as shown below:

$$Y_{net1} = w_5 Y_{out,h1} + w_6 Y_{out,h2} + b_2. \tag{1.4}$$

The partial derivative of the Y_{net1} corresponding to the weight parameter of w_5 is calculated by $\frac{\partial Y_{net1}}{\partial w_5} = Y_{out,h1}$. The partial derivative of the Y_{out1} corresponding to Y_{net1} is calculated as shown below:

$$\frac{\partial Y_{out1}}{\partial Y_{net1}} = Y_{out1}(1 - Y_{out1}) = \frac{1}{1 + exp^{-Y_{net1}}}(1 - \frac{1}{1 + exp^{-Y_{net1}}}). \tag{1.5}$$

Therefore, the overall partial derivative δ_{w_5} corresponding to weight w_5 is calculated though the chain rule as shown below:

$$\delta_{w_5} = \frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial Y_{out1}} \frac{\partial Y_{out1}}{\partial Y_{net1}} \frac{\partial Y_{net1}}{\partial w_5} = \frac{Y_{out1} Y_{out,h1}}{1 + exp^{-Y_{net1}}}(1 - \frac{1}{1 + exp^{-Y_{net1}}}). \tag{1.6}$$

By knowing the partial derivative δ_{w_5}, we can calculate the overall gradient descent value η_{w_5} corresponding to w_5 by multiplying a constant ratio, namely learning rate α. Then we can update the weight parameter of w_5 by applying the calculated gradient descent value as shown below:

$$w_5^* = w_5 - \eta_{w_5} = w_5 - \delta_{w_5} \alpha. \tag{1.7}$$

Note that the learning rate is a free parameter and it is typically determined through a trial-and-error fashion. Furthermore, all the gradient descent values can be calculated in

similar manners through the chain rules at each epoch to update all the model parameters.

1.3.2 Convolutional neural networks

In this chapter, we introduce the concept of convolutional neural networks (CNNs). Convolutional neural network (CNN) is an integrated machine learning framework by combining the conventional artificial neural network with convolutional layers. A common application of CNN is image recognition and classification.[13, 14, 15] The CNNs are also widely applied for other mainstream tasks such as signal processing[16, 17] and Natural Language Processing (NLP).[18, 19] An intuitive explanation of CNNs is the neural network model with special characteristics of automatic feature extraction. A typical CNN model contains two essential components, which include a convolutional processing layer and the artificial neural network model. The convolutional layer contains a number of filters, known as convolutional kernels, which are responsible for pooling the information that is contained in the image in a systematic way. Essentially, a filter is a matrix of randomly generated numeric matrices, which is responsible for calculating the product value for a subset of image pixels. As we move the kernel matrix across all the image pixels in either one or two dimensional (1D or 2D), the matrix multiplication or linear transformation is consistently operated until the kernel has covered all the pixels in the image. The result of the filtering process is a generalized squared matrix with a smaller dimension depending on the kernel size. Note that the number of kernels is a very important free parameter and typically, the generalized feature map becomes incredibly large as more kernels are applied in the convolutional layer. Another critical component of CNN model is the pooling layer, where the downsampling procedure is applied to summarize the complexity of the feature map with the goal to reduce the over-fitting effect. Two commonly used effective pooling methods are average pooling and max pooling. The second essential component of CNN is the artificial neural network

(ANN), which is constructed to learn the correlation between the latent variables from the convolutional layer and the output values. The weight parameters in the ANN model are trainable and being optimized during the actual learning process. Figure 1.5 shows the architecture of the convolutional neural network (CNN) models.

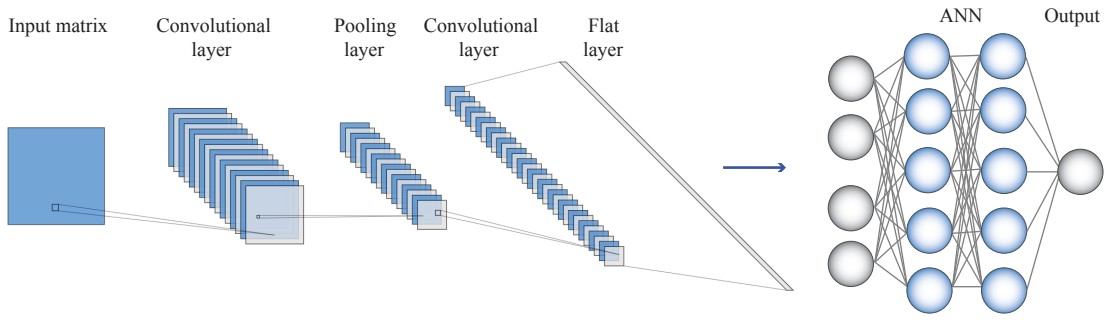

Figure 1.5: Schematic of convolutional neural network model.

1.3.3 Graph neural networks

Graph neural networks (GNNs)[20] are a powerful variant of conventional convolutional neural networks (CNNs) type machine learning algorithms. The GNNs are particularly designed to operate on graph type objectives and have recently received wide attention for many scientific types of research and studies. For instance, the GNNs shows promising prediction performance for semi-supervised node classification tasks in the domain real-world application of social network study,[20, 21, 22]. Besides, the GNNs are also widely used in the regression tasks to predict some interesting properties such as Human-Computer Interaction behaviors,[23] hourly bike-sharing demands,[24], social relationship inference for user mobility.[25] To date, GNNs have become one of the most successful emerging deep learning models, which provide leading-edge solutions for a series of challenging problems. Importantly, the research and study for GNNs is always an ongoing process and a lot of

research efforts are devoted to improving the model robustness. In this chapter, we briefly introduce three typical GNN models and the corresponding fundamental theory.

Graph convolutional network model

The graph convolutional network (GCN) model[20] is the most basic graph neural networks (GNNs) architecture. It was originally developed for semi-supervised node classifications. The GCN model attempts to learn an appropriate node embedding with a message passing/graph convolution operation. More specifically, a learned node embedding is obtained by propagating the graphs' underlying node feature representations and the corresponding topological information. The learned node embedding can be further used for different downstream tasks such as regression and classification. For a graph object $G = (V, E)$, where $V \in \mathbb{R}^{N \times F}$. Here V denotes a set of N nodes with F-dimensional node feature vectors. E represents a set of interconnected edges within the graph G. The objective of GCN convolutional is to learn a novel node embedding X^{i+1} over the graph G for i^{th} convolutional layer. The node embedding X^{i+1} is calculated as shown below:

$$X^{i+1} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}X^{i}\theta, \tag{1.8}$$

where $\tilde{A}$ denotes the graph adjacency matrix and $\tilde{D}$ denotes the graph diagonal degree matrix. $\tilde{A}$ is a binary squared matrix determined by the edges of the graph G. $X^i \in \mathbb{R}^{N \times F}$ is the node feature matrix at i^{th} convolutional layer in the propagation process. θ is the trainable parameters, which are determined by the supervised learning process. Note that the propagation process can be repeated as many times as possible and the newly obtained node embedding matrix X^{i+1} can be considered as new input for another GCN convolutional layer.

Graph attention network model

Graph attention network (GAT)[26] model is an advanced GNN model, which is characterized by an intelligent self-attention propagating mechanism. In essence, the processing unit, namely the graph attention layer, is designed to propagate the node features and the corresponding topological information that is contained in the graph. Unlike the GCN model, GAT model has a more sophisticated form of message passing process. More specifically, GAT convolutional layer is operated in the following three primary steps: 1) A node from the graph G is selected as the central node and the corresponding neighbors of the given node are labeled as neighbor nodes, 2) A linear transformation with weight matrix of $W \in \mathbb{R}^{F^{'} \times F}$ is applied to process all the selected nodes one after another, 3) A non-linear operation $\sigma(.)$ is applied to aggregate all the convoluted nodes features for the new learned node feature of the central node. It is noteworthy that the dimension $F^{'}$ of the trainable weight matrix of W is adjustable in the learning process and the attention coefficients e_{ij} of the central node i with respect to the first-order surrounding neighbors is calculated as shown below:

$$e_{ij} = a(WX_i, WX_j), \tag{1.9}$$

where a denotes a shared attentional mechanism $a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \to \mathbb{R}$. Then the attention coefficient e_{ij} is normalized by taking into account all the surrounding nodes, so that the coefficients are comparable across different nodes. The normalized attention coefficients are calculated by the softmax function as show below:

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in N_i} exp(e_{ik})}, \tag{1.10}$$

where N_i denotes the neighborhood nodes of node i in the graph G. The normalized attention

coefficient e_{ij} indicates the importance of node j with respect to node i. The normalized attention coefficients are used to compute the aggregated features $X \in \mathbb{R}^{F'}$ using a non-linear activation function $\sigma(.)$ as shown below:

$$\sigma(\sum_{j \in N(i)} \alpha_{ij} W X_j). \tag{1.11}$$

Figure 1.6 illustrates the concepts of messaging passing of GAT along with the schematics of some key components such as graph convolution, graph pooling, and graph gathering.

Graph isomorphism network model

The concept of Graph isomorphism network (GIN)[28] model is analogous to the Graph attention network (GAT). Both GAT and GIN contain the same processing components and units such as graph convolution, graph pooling, and graph gathering operations. However, instead of linearly propagating the node features, the GIN transforms the node features $X^i \in \mathbb{R}^F$ to novel node features $X^{i+1} \in \mathbb{R}^{F'}$ with variant dimensions using a much-complicated multi-layer perceptron (MLP) as shown below:

$$X^{i+1} = MLP((1+\epsilon)X^i + \sum_{j \in N(i)} X_j), \tag{1.12}$$

where ϵ denotes a trainable parameter or fixed scalar for taking account of the impact of central node features toward neighbor nodes N_i. Recent study[28] proves that the GIN achieves the maximum discriminative power over the other GNN models in the prediction tasks and is considered as the state-of-the-art GNN model at the moment.

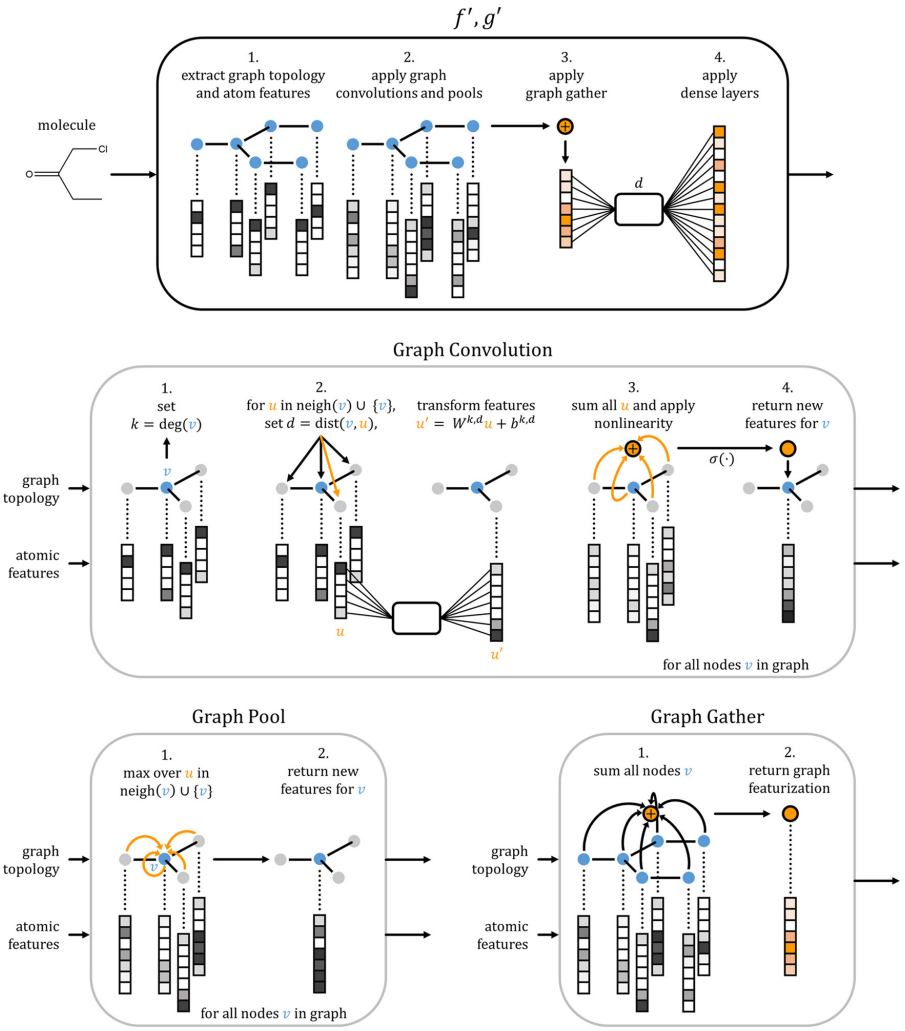

Figure 1.6: Schematics of GNNs message passing process for learning drug molecules. Figure is adapted from Altae-Tran et al. (2017)[27]

1.4 Saliency maps

Saliency map,[29, 30], a popular visualization technique, has attracted huge attention in the field of computer vision. The general goal of the saliency map is to transform the original view of an image to an intuitive way where the high-impact features (i.e, image pixels, resolutions) are distinguished from the low-impact feature with respect to the model prediction outputs. The salience map, as an effective tool, is of particular importance for interpreting complex

machine learning models such as a convolutional neural network (CNN). More precisely, it provides a novel means to understand the model by highlighting the important feature factors, which captures the most neuron activity of interest output and the corresponding variances. Essentially, a salience map can be viewed as an image segmentation, which is the process of segmenting the input image into pixels and quantifying the contribution of individual pixels toward the target variable in the deep learning model using gradient ascent in the input space. Here, the quantified contribution of a specific input feature or pixel refers to the corresponding weight vector w_c^T. Typically, a large weight vector indicates that the feature pixel plays a very important role in the model output variance in terms of either likelihood in the classification model or numeric values in the regression task. To illustrate the concept of saliency map method, we provide a motivating example as shown below:

$$S_c(I) = w_c^T I + b_c, \tag{1.13}$$

where I denotes the input matrix and $S_c(I)$ denotes the model likelihoods corresponding to a specific class c. However, in the case of deep convolutional neural network (CNN), the model output $S_c(I)$ has a highly non-linear relationship with input I, therefore the first-order Taylor expansion is applied to approximate $S_c(I)$ by establishing a linear relationship with the neighbourhood of I as shown below:

$$S_c(I) \approx w_c^T I + b_c. \tag{1.14}$$

The weight vector w_c of each individual input feature I_0 can be approximated by taking the derivative of $S_c(I)$ with respect to the input feature as shown below:

$$w_c = \frac{\partial S_c}{\partial I}|_{I_0}. \tag{1.15}$$

It is noteworthy that another intuitive interpretation of the magnitude of computed weight vector w_c is the sensitivity of a specific feature input toward the model output variance.

Chapter 2

Convolutional neural network for crystal formation energy

2.1 Introduction

Crystal (crystalline solid) is a solid material whose constituent atoms are organized with highly ordered microscopic structure.[31] Typically, the crystal materials are composed of a number of repeating units, which are the smallest unit cells that contain all the necessary information toward the entire geometry such as atom types and coordination. A complete crystal material can be formed by repeating the smallest unit cells in the three dimensions. Crystal material plays an essential role in the formation of industrial functional substances such as semiconductors, fuel cells, leading to a series of new fundamental researches. The assessment of crystal material properties is becoming the primary aspect of modern material scientists and a large number of research efforts are devoted to understanding the materials structure-property relationship.

In the last few decades, the first principle methods, i.e., density functional theory (DFT), have attracted enormous attention. To date, it has become one of the most widely used computational tools in the research and study of physical science. DFT, a quantum mechanical simulation, is a computer-based simulation approach in the field of solid-state physics based on the optimization of materials' many-electron wave-functions. DFT has attracted enor-

mous attention because it offers an accurate and rapid approximation of materials' properties at the atomic level. Even though DFT provides a highly accurate computational means to assess a large number of material properties prior to actual experiments, it still consumes a lot of computational resources. Recently, the data-driven models based on high-speed machine learning algorithms are considered as surrogate models for the DFT method for fascinating the material screening process. Machine learning algorithms in nature have huge advantages over the conventional physical models in processing massive amounts of unstructured data with high prediction accuracy. Essentially, machine learning provides algorithms the ability to automatically discover the hidden patterns of data without being explicitly programmed. Because of this, machine learning models have gained huge successes in applications such as website recommendation systems, online fraud detection, robotic technologies, image recognition, etc. Nevertheless, implementing machine learning techniques in the field of functional material discovery remains difficult due to our insufficient knowledge about the structure-property relationships for diverse material systems. In another word, developing a robust physically intuitive feature method requests in-depth expert knowledge about the underlying physics of the material system and it is always a challenging task.

As machine learning models heavily depend on features or representation methods, developing a robust representation method for material systems is the key to the success of any machine learning tasks. In recent years, many novel representation methods are developed to capture the underlying physics of a variety of materials for the desired properties. For instance, moment-based descriptors according to the knowledge of d-band theory[32] are used in the machine learning models to predict the metallic alloys and metal oxides' adsorption properties.[33, 34, 35, 36, 37] Coulomb matrix[38] and the corresponding extended versions[39, 40] based on the Coulomb repulsion between atoms are developed to capture the formation energies of a diverse set of organic molecules and crystals. Elemental and

structural representations (i,e., coordination number, and radial distribution function) are developed to describe a wide range of crystal structures for predicting the first-principle properties.[41, 42]

Even though the machine learning models have gained huge success in predicting the properties of a wide range of materials, the model prediction performance is primarily dependent on the selected representation method, namely descriptor. Developing a robust descriptor for given material typically requires experts' domain knowledge about the material's underlying physics. Following are the standard procedures for descriptor design, which have been largely adopted by the material science community for many years: (1) generate a candidate set of material feature vectors or descriptors according to domain knowledge, (2) select a subset of descriptors from the candidate set via the trial-and-error method (i.e, cross-validations). Nevertheless, the standard descriptor design procedure is an inefficient process because it requires an extensive amount of human efforts. In this regard, we implement an automated feature engineering scheme using convolutional neural networks (CNNs). The scheme aims to extract the latent vectors that are contained in materials' DFT calculated electronic structure in an automatic fashion. To evaluate the performance of our CNN model, we evaluate the model prediction performance using selected data from Material Project[43] database. We show that the CNN model can capture the DFT calculated formation energies with high accuracy for a wide range of crystal structures. Importantly, we shed light on the important molecular orbitals toward the formation of energy using the saliency map technology.

2.2 Computational method

2.2.1 Data description

The primary research focus of this project is to estimate the DFT-calculated ground state formation energy of metal oxides' bulk structure using the deep learning models. ABO_3 type metal oxide materials have wide industrial applications such as solid fuel cells, water-splitting cells, etc. ABO_3 metal oxide has a versatile structure with a wide variety of element choices for the A and B cations. Usually, the A-site cations are rare-earth metals while the B-site cations are transition metals from the periodic table. According to crystal field theory,[44] the strong interaction forces between the positively charged transition metals and the negatively charged oxygen ligands would give rise to an unstable structure. Subsequently, the metal oxide with the same composition could have multiple coexistence phase structures due to the static electric field within the crystal structure. As formation energy is of vital importance in determining the most stable phase state, an enormous research effort has been made to predict those energy properties with the minimum computational cost. In this work, we adapted the 785 ABO_3-type metal oxides' formation energies and the associating electronic structures from Material Project[43] database. The metal oxide structures have an enormous compositional and configurational degrees of freedom with 65 unique A-site cations, 77 unique B-site cations, and 7 common crystal phases (i.e., cubic, hexagonal, monoclinic, orthorhombic, tetragonal, triclinic, trigonal). Note that the formation energies are normalized by the total number of atoms in the bulk structures so that the energy values are comparable between different structures. Figure 2.1 below exhibits the theoretical computational structure of the $CaTiO_3$ and the corresponding electronic structures.

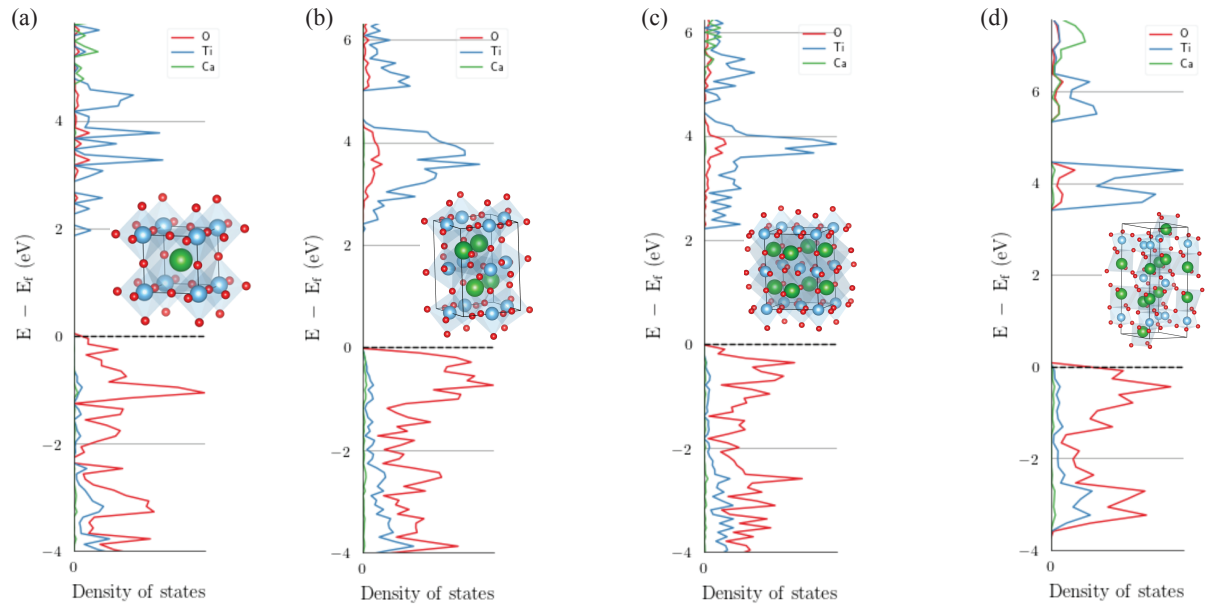

Figure 2.1: $CaTiO_3$ metal oxides and the corresponding electronic structures. The electronic structures for phase states of (a) cubic, (b) orthorhombic, (c) tetragonal (d) trigonal.

2.2.2 Electronic structure

In this work, the DFT-calculated orbital-resolved Partial Density of States (PDOS) feature is used as primary features to capture the formation energy for different metal oxide systems. Here, PDOS can be considered as a degenerated form of the total density of states (DOS). The density of States (DOS), a property from solid-state physics, is essentially developed for describing the number of electronic states at specific energy levels, where the electrons are allowed to occupy. DOS is a very important concept in the context of quantum-mechanically to describe the properties of a macroscopic system. A DOS graph of a given material contains much valuable information about the material's special characteristics. For instance, if a significant energy gap (energy difference between the conduction band and the valence band) is observed from the DOS graph, we know that the material is a semiconductor. Furthermore, according to [32], the electronic structure characteristics (i.e., d-band center, d-band width) are closely related to the system's energy properties such as bond formation energy. The d-

band model has been widely recognized as one of the most successful theoretical frameworks in understanding a variety of surface adsorption properties.

According to Figure 2.1, we know that DOS is very sensitive to the crystal's structure, compositions, and morphology. A slight structure change would give rise to a distinguished DOS graph. Therefore, a robust feature representation that is capable of capturing all the detailed information is essential for machine learning models. Nevertheless, obtaining effective features for a material's electronic structure is a very challenging task in particular for metal oxides. According to Figure 2.1, the electronic states are degenerated into separate main peaks due to the interactions between the transition metals and the ligands within the structure. Even though the conventional moment-based descriptor introduced by Nørskov et al.[45] works extremely well in capture the metallic systems' formation energy, it works poorly for the metal oxide systems because of the nature of DOS degenerated effect. In this regard, a robust and powerful representation method for encoding DOS is highly demanded.

2.3 Results and discussion

2.3.1 Feature Representation

Feature engineering plays an essential role in any machine learning models. The general purpose of feature engineering is to represent the input object with numeric feature vectors that capture the underlying characteristics. For instance, the image objects can be encoded as numeric matrices and the matrix values denote the color-intensive in the application of image recognition. In this study, the orbital-resolved Partial Density of States (PDOS) of crystal material is used as input for the machine learning model. Our primary objective is to develop an automatic approach that can encode all the detailed information from the Partial

Density of States (PDOS) for all the crystal structures. One of the major advancements of our feature representation approach is that it is fully automatic and doesn't require any manual feature engineering process.

The orbital-resolved Partial Density of States (PDOS) as shown in Figure 2.1 can be considered as a probability distribution of electronic density of states over energy levels. However, the distribution doesn't follow the common well-known probability density function (PDF) such as Gaussian distribution, binomial distribution, and etc. To capture the full information about the orbital-resolved Partial Density of States (PDOS), we proposed a novel algorithm that can automatically learn the noisy PDOS distributions by fitting a multiple Gaussian distributions functions. For a given PDOS graph, our algorithm is capable of determining the number of significant peaks and assigns the Gaussian functions with respect to the identified peaks. In brief, we aim to model the noisy PDOS distribution based on the assumption that the PDOS is fully made up of a set of Gaussian distribution functions. Then we can obtain the model-predicted PDOS (PDOS spectrum) over a specific range of energy levels using the fitted mixture Gaussian function. One of the main advantages of our algorithm is that the PDOS spectrum can be constrained at a fixed range of energy levels so that all the feature vectors corresponding to different crystals are comparable. The proposed algorithm for PDOS spectrum contains 5 consecutive steps as shown below,

Algorithm 1: Fit multiple Gaussian function for PDOS

1. Identify the significant peaks $(p_1, p_2, ..., p_n)$ from the PDOS
2. Initialize the Gaussian parameters $[(c_1, w_1, h_1)$, (c_2, w_2, h_2),...,$(c_n, w_n, h_n)]$ corresponding to the identified peaks $(p_1, p_2, ..., p_n)$
3. Construct the multiple Gaussian model $\mathrm{G}(p_1, p_2, ..., p_n) = \sum_{i=1}^{n} h_i exp^{\frac{-(x-c_i)^2}{2(w_i)^2}}$ by adapting all the peak parameters from the previous step
4. Optimize the Gaussian parameters by minimizing the Least-square error of the multiple Gaussian model
5. Predict the PDOS values for energy levels [-15, 15] using the fitted multiple Gaussian function

In practice, we fit 9 multiple Gaussian models for all the molecular orbital density of states (i.e., d_{z^2}, d_{x^2}, d_{xy}, d_{xz}, d_{yz}, p_x, p_y, p_z, s) for a given crystal structure. Here the 5-dimensional d_{z^2}, d_{x^2}, d_{xy}, d_{xz}, d_{yz} and 1-dimensional s orbital correspond to the PDOS spectrum for B-site transition cations. The 3 p_x, p_y, p_z corresponds to the PDOS spectrum for oxygen ligands. The PDOS spectrum for a given crystal has a fixed dimension of 30 by 9, which describes all the electronic density of states over the energy level from -15 to 15 eV. Figure 2.2 below shows a comparison of model-fitted PDOS and the actual PDOS for the crystal structure of cubic $SrNiO_3$.

According to Figure 2.2 (a), we observe that the fitted multiple Gaussian functions work extremely well capturing all the distributions of PDOS with a set of discrete variables. The model fitted variables, namely the PDOS spectrum, are used as input for learning the convolutional neural network (CNN) model for predicting the crystal formation energy.

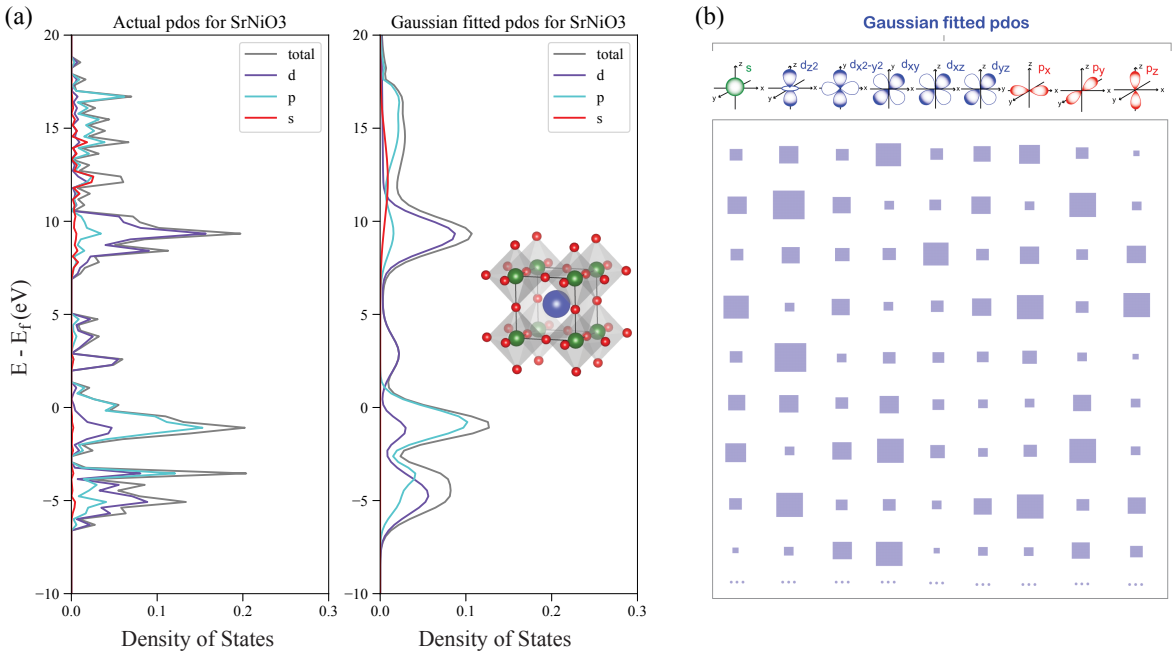

Figure 2.2: (a) Comparison of the model-fitted DOS and actual PDOS for $SrNiO_3$. (b) Schematic representation of the model fitted DOS spectrum.

2.3.2 Convolutional neural network configuration

We set up a convolutional neural network (CNN) framework for learning the crystal formation energy using the open-source Keras library.[46] The CNN framework as shown in Figure 2.3 consists of a regular CNN model and an embedding processing layer. The CNN model is responsible for processing the PDOS matrix through 2 consecutive convolutional layers with 16 2x2 dimensional kernels. It is noteworthy that each convolutional layer is followed by a 2x2 dimensional max-pooling layer, which is responsible for preventing the over-fitting issues by reducing the dimensionality of the convoluted feature matrix. Then, we transformed the pooled feature matrix to a 1-dimensional latent vector as a partial-input for the feedforward artificial neural network (ANN). To capture the crystal's composition information, we added another linear embedding layer to transform the atomic properties to the 1-dimensional latent vector for the ANN. The 10-dimensional atomic properties are calculated purely based on

the crystal's structural composition. Those properties include 2-dimensional A/B cation electronegativity, 2-dimensional A/B cation oxidation state, tolerance factor and octahedral factor, 4-dimensional including ionic radii ratios and the differences in electronegativity of A and B atoms relative to O atom using the literature method.[47] In the final step, the ANN takes the merged latent vectors from the previous processing layers and maps the merged latent vectors to the formation energy through 2-layer neural network architecture. The ANN model contains 2 hidden layers and each hidden layer includes 16 rectified linear units (ReLU) as a processing engine. Note that all the parameters associated with either CNN or ANN models are determined through a trial-and-error fashion for the best model prediction performance.

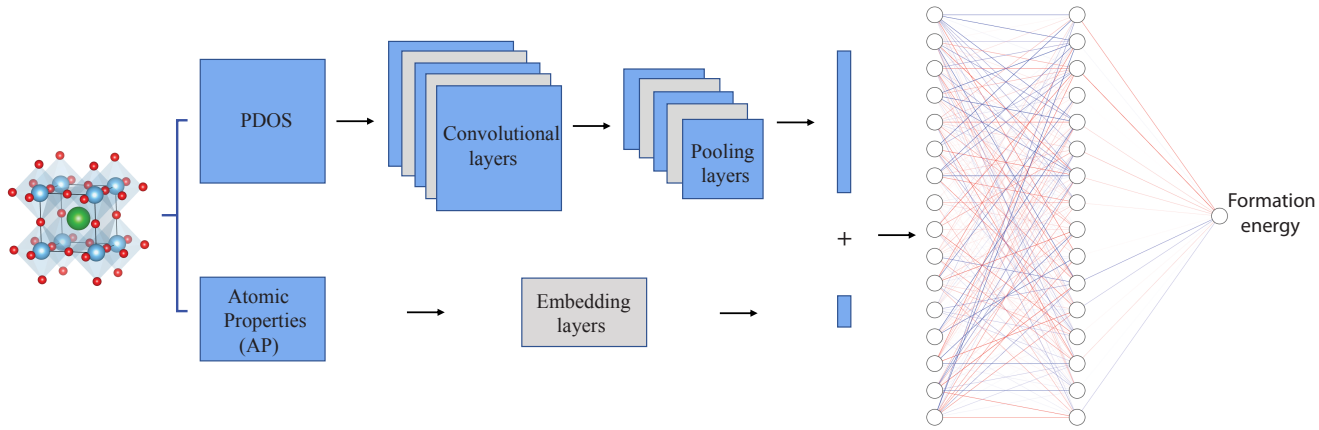

Figure 2.3: Schematic illustration of CNN framework for crystal formation energy.

2.3.3 Model prediction evaluation

We divided the ABO_3 dataset into two subsets with 75% for training the convolutional neural network (CNN) framework and the rest 25% for testing the model performance. To prevent model overfitting caused by the excessive number of epochs, a validation dataset (25% of the training dataset) is created internally for terminating the back-propagation

trainer when the cross-validation root-mean-square error (RMSE) reaches the minimum. Generally, the cross-validation error shows a volcano relationship with respect to the training epochs. Therefore, the purpose of the early stop mechanism of deep learning models is to stop the training process at an appropriate time to prevent overfitting. During the training process of the CNN model, the 10-fold cross-validation is performed to identify the optimal network structure (i.e., number of convolutional layers and number of neurons at each layer, kernel size). To eliminate the sampling bias toward the model prediction performance, we randomly select the training/testing datasets 16 times and evaluate the overall model prediction performance by calculating the averaged validation error. According to Figure 2.4, the CNN model trained with 75% data provides testing error around 0.5 eV for the rest 25 % data while a combined feature set of atomic properties and the PDOS matrix would give rise to an improved model prediction performance (RMSE=0.34 eV). Therefore, the CNN model learned both the atomic features and the PDOS features attained the best results.

Besides the testing error, we also evaluate the model generalization capability corresponding to the training data size. Here, generalization is termed as the model's predictive performance with respect to the unknown data. Theoretically, a robust machine learning model is able to capture the desired properties of unknown data without the need of training a large number of data. To evaluate the model's generalization power, we conduct an experiment by inspecting the model's performances for different training data size. More specifically, we split the entire crystal data with different train/test ratios as shown in Figure 2.5, and evaluate the model prediction error. Note that we randomly sampled 8 times for each train/test ratio and the bar heights as shown in Figure 2.5 indicates the averaged value of the errors and the black bar indicates the standard deviation of the errors. According to Figure 2.5, we observed that both CNN models exhibit good generalization corresponding to the unknown testing data

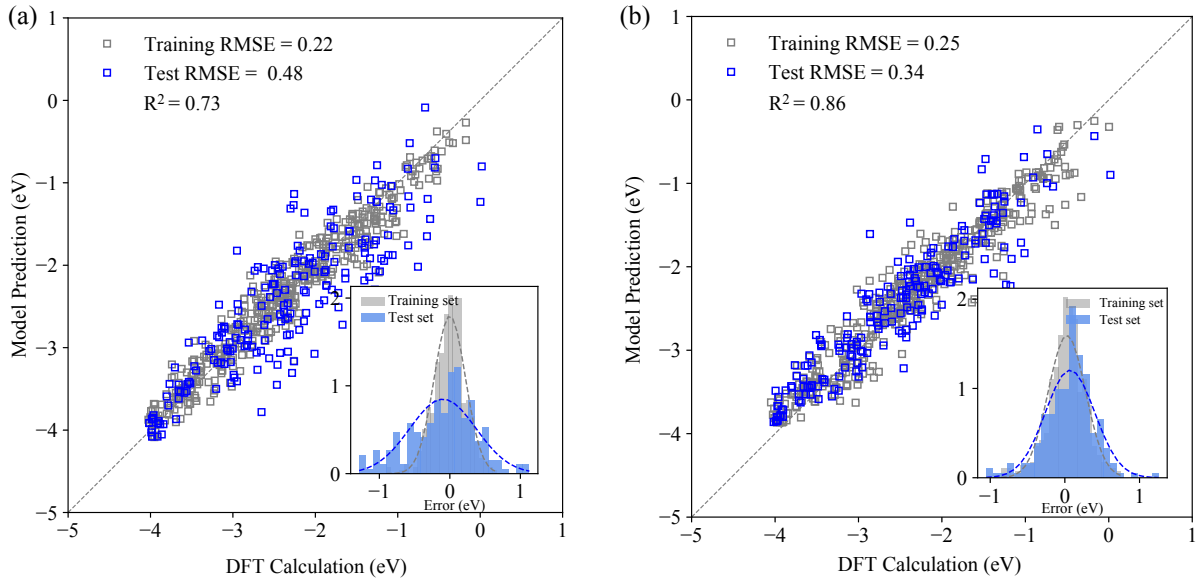

Figure 2.4: CNN model prediction evaluation. DFT-calculated vs. model predicted formation energies for (a) Partial Density of States (PDOS) features and (b) a combined features including Partial Density of States (PDOS) and atomic properties. The insert plot shows the histogram distributions of both training (grey) and testing (blue) errors.

by training on relatively small data (∼50% training data). Thus, the CNN model trained with PDOS and atomic properties can capture well with the formation energies for a large amount of crystal structure with a high degree of freedom of configurations and composition.

2.3.4 Saliency map analysis

It is noteworthy that our CNN framework has a number of pros and cons in terms of usability and interpretability. Following are the benefits (pros) of our model: (1) the latent variables are extracted from the PDOS in an automatic fashion, (2) PDOS is high-level intuitive feature within the context of d-band theory, (3) PDOS can capture the crystal formation energies with good accuracy between 0.3 eV to 0.4 eV. However, the CNN model framework has huge disadvantages in terms of practical usability. For instance, the CNN model is heavily relying on the DFT-calculated PDOS, which is an expensive feature resource compared to

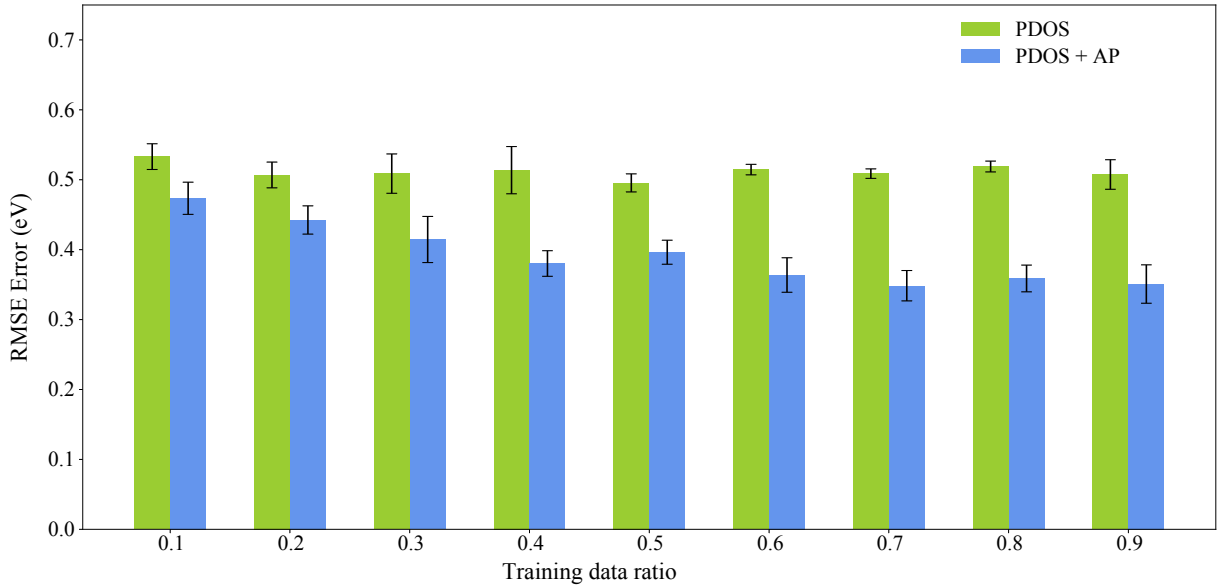

Figure 2.5: CNN model generalization evaluation for different training data ratio. The green/blue bars indicate the model prediction errors by training different features.

other types of structural information such as atomic properties and the crystal structure properties. In another word, DFT calculation for the PDOS is required prior to model prediction, which significantly limits the CNN method's widespread applications for a large material space. Even though the PDOS feature has issues in terms of usability, it allows us to draw valuable molecular orbital insights toward the formation energy using the technique such as saliency maps. The saliency map approach has gained huge success in the field of computer vision to visualize the important pixels in the training image corresponding to the output value in the supervised deep learning model. The saliency score value on the saliency map is calculated by taking the gradient of the model output value with respect to the input matrix. Typically, if an image pixel is identified by the saliency map, we know that the pixel plays a crucial role in the model decision. Figure 2.6 shows the example of saliency heat maps along with the PDOS features of $BaRuO_3$ and $SrIrO_3$.

According to Figure 2.6, we observe that the saliency maps are varied significantly for the

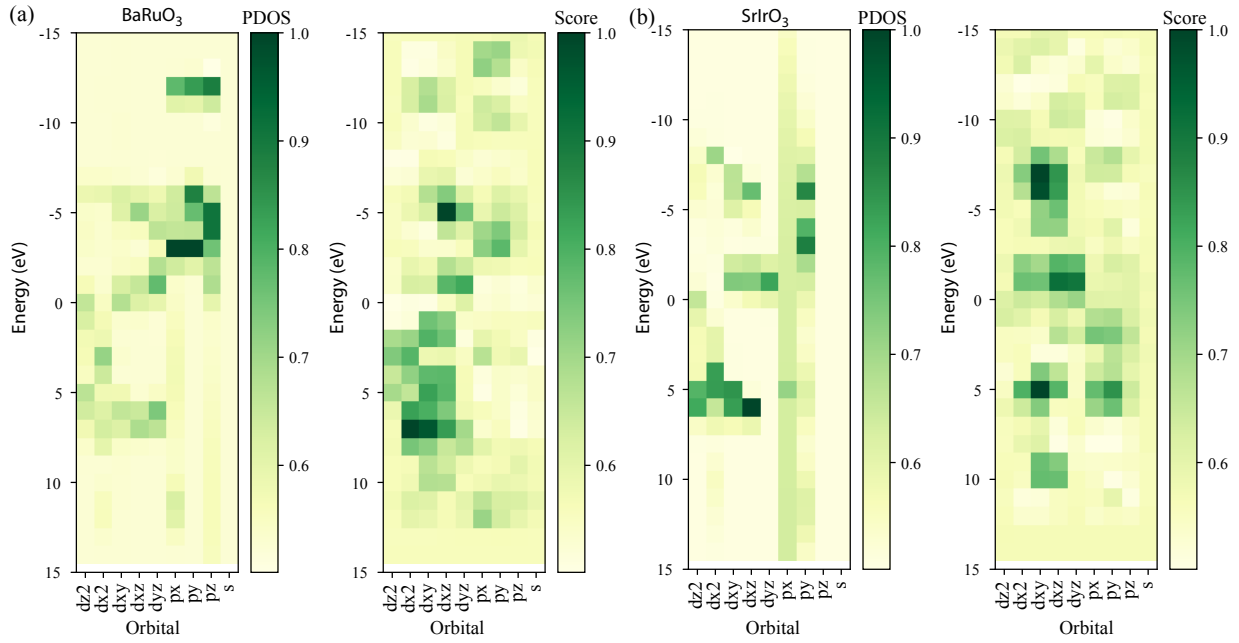

Figure 2.6: Saliency maps for crystals. The crystals with structures of (a) $BaRuO_3$ and (b) $SrIrO_3$.

crystals with distinguished PDOS features. For both $BaRuO_3$ and $SrIrO_3$ crystals, the d-orbitals PDOS (i.e., d_{xy}, d_{xz}, d_{yz}) on energy levels between -5 to 10 are import toward formation energies for relative large saliency score values. Then, we rationalized the important molecular orbitals of La-based and Ba-based metal oxides by inspecting the model calculated saliency scores. The La-based and Ba-based metal oxides are selected because they are important electrode materials in the water splitting cells. For each crystal structure, we calculate the sum of the score values across all the energy levels of each molecular orbital. We assume that the orbital with the largest saliency score is likely to be the key electronic factor toward formation energy. Figure 2.7 exhibits the heat map distributions of the important molecular orbitals for a series of La-based and Ba-based metal oxides. According to Figure 2.7, we know that the B-site metal's d orbital properties play an extremely important role toward crystals' formation energies for both the La-based and Ba-based metal oxides. While the p-based orbitals have a secondary effect on the formation properties for relatively

small sums of saliency score values.

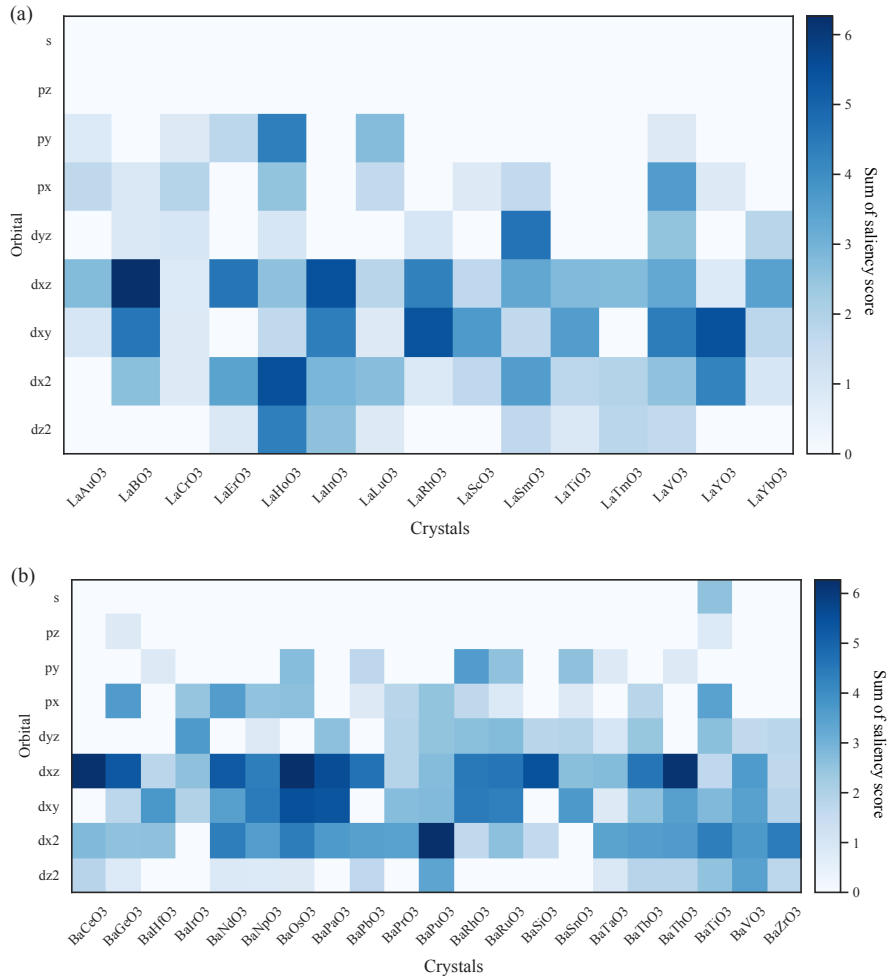

Figure 2.7: Distribution of important molecular orbitals for selected crystal structures. The sum of saliency scores for (a) *La*-based orthorhombic metal oxides and (b) *Ba*-based cubic metal oxides.

2.4 Conclusions

In summary, we develop a novel CNN framework for predicting the crystal formation energy using a combined feature set including the DFT-calculated partial electronic density of state (PDOS), together with the atomic properties of the crystal structure. Compared with the previous conventional machine learning models based on extensive expert domain knowledge, our proposed CNN framework leverages an automated feature engineering process. Thus, a significant amount of manual effort and time is saved because of the automated feature engineering mechanism. The CNN framework gives us a favorable prediction error of $\sim$ 0.3 eV, which is similar to the DFT system error $\sim$ 0.2 eV. More importantly, the PDOS matrix, a high-level intuitive representation method, allows us to draw important molecular orbital insights toward the crystal formation energy by interpreting the CNN model using the saliency map approach.

Chapter 3

Graph convolutional neural network for crystal first-principle property

3.1 Introduction

In this chapter, we investigate the performance of another deep learning model, namely graph neural network (GNN), and apply the model to predict the first-principle properties (i.e., formation energy) of crystal structures. Essentially, GNN is a variant form of a conventional convolutional neural network (CNN). A common characteristic of CNN and GNN models is that they both attempt to learn the latent variables that are contained in the objects such as numeric matrix or graph via a series of convolutional processes. However, the convolutional process of GNN models is significantly different from the CNN models. More specifically, the convolutional layers of GNNs are particularly designed to capture the spatial hidden patterns from the graph objects systematically. In recent years, the GNN model has attracted growing interest in emerging fields of computational functional material design because GNN outperforms the conventional machine learning models by providing state-of-the-art solutions to many challenge machine learning tasks including both regression and classification problems. For instance, the GNN models are applied to learn drug molecules and proteins to predict the drug-target interaction properties.[48] The GNN models are established to predict the organic chemical reactivity by learning the reactant compounds.[49] GNN models are applied

to extract the circular molecular fingerprints for a number of regression tasks.[50]

3.2 Related work

Previously, Xie et al.[51] presented a graph convolutional neural network (GCNN) model for predicting a series of crystal material properties. Even though the reported GCNN exhibits high performance in capturing the material properties, the GCNN model has flexibility issues with a rigid model structure (i.e, feature representation, and graph convolutional process). More specifically, the graph representation method of GCNN was designed for the most basic graph convolutional neural (GCN) model and the method is not generic enough to compatible with other advanced graph neural network models. In this regard, we introduce a robust graph-based deep learning framework for learning the crystal material properties by using the easily accessible crystal's topological structural characteristics. Compare to the previous GCNN model, our graph-based framework has a flexible structure with a generic graph representation method that is compatible with all the newly-developed graph neural network models (i.e., GCN, GAT, GIN, etc). Besides, our graph-based deep learning models achieve high accurate prediction results with respect to the DFT-calculated formation energies, which is attributed to the robust graph representation method. As one of the most important ingredients, the graph representation method is of vital importance to model performance. In this regard, we primarily focus on addressing the methodology of constructing graph representations including the node feature vectors, node connectivity, and etc. Besides, we conducted a comprehensive analysis by evaluating the key parameters toward the model prediction results using the benchmark data involving above 20,000 crystal structures. Compared to the CNN based model as mentioned in Chapter 2, the GNN model shows extraordinary advantages in terms of applicability and accuracy. In particular, the GNN model

relies on crystal graphs, which are accessible features compared to the DFT-calculated partial density of state (PDOS). Therefore, the GNN model exhibits extraordinary applicability and can be easily applied to assess the crystal properties of vast chemical space. Figure 3.1 below illustrates the methodology of the GNN model framework.

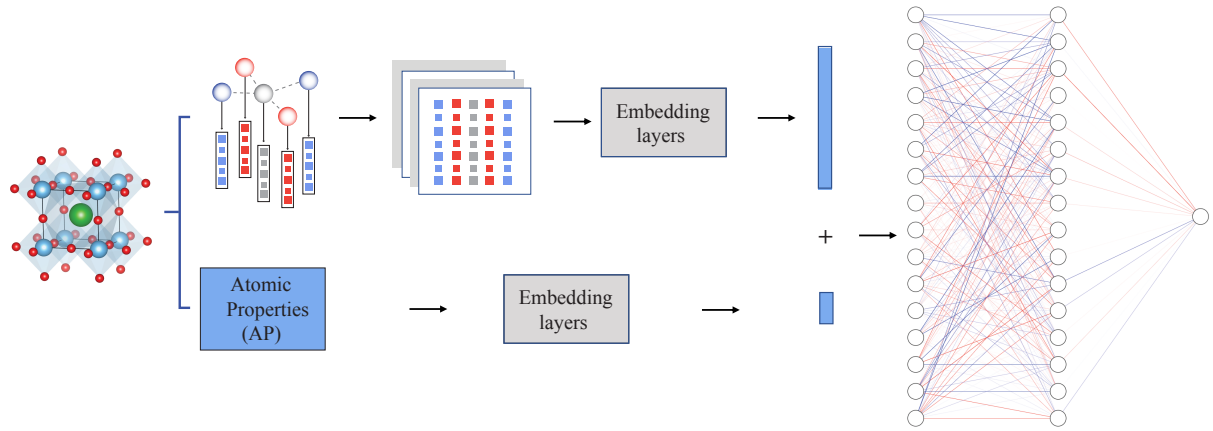

Figure 3.1: Schematic display of graph neural network framework for crystal properties.

3.3 Computational methods

3.3.1 Data description

In this work, we evaluate the performance of graph neural network models for predicting the DFT-calculated ground state formation energy along with the bandgap properties using the benchmark data. We adopted a large number of crystal structures along with their material properties from the Material Project[43] database for training the deep learning model. The training data includes 23,029 ABX_3-type crystal configurations. ABX_3 type inorganic crystals have versatile configurations with a wide variety of element choices for the A, B, and X elements. Specifically, the A/B site cation contains 65/77 unique elements respectively

involving the majority elements from s-block, d-block, and f-block elements in the periodic table. X denotes the ligand substitute including O, S, N, F p-block atoms. Note that the initial computational structures of the crystals are extracted to build the crystal graphs as input features for the deep learning model. Here, the initial structure represents the theoretical model structure for the DFT calculated, which is constructed according to the material's lattice constant and bulk coordination. Besides the crystal structures, atomic properties are also calculated based on the crystal structure and the corresponding constituent elements. The crystal's formation energy and band gap are the target properties for the machine learning model. Note that bandgap, an intrinsic property of solid, indicates the energy difference between the crystal's valence band of electrons and the conduction band. Typically, insulators have relatively large bandgap values while the bandgap values for transition metals are 0. Semiconductor usually has an intermediate-sized but non-zero band gap value, which can be varied by thermal excitation. Therefore, predicting the bandgap property is of vital importance because bandgap is a major factor determining material functionality such as electrical conductivity, electro photocatalytic property and etc.

3.4 Results and discussion

3.4.1 Graph representation method

The selection of appropriate feature representation methods is always the key to the success of the development of machine learning models. In the GNN models, we consider each crystal structure as a graph representation of interacted nodes. Each graph representation is unique and contains all the valuable information about the crystal structure such as the node types, node connectivity, size of edge pairs, and etc. As the convolution performance is closely

related to the graph representation method, developing appropriate graph representation methods is one of our primary objectives. For a given crystal structure, we consider all the atoms as nodes and describe all the node information with a one-dimensional binary vector. Specifically, the binary vector encodes the node information such as element type, node degree, and node degree uniqueness. For the element type, we created a library including 83-dimensional most common elements from the periodic table and used this sparse vector to encode the represent or absence of a specific element. The node degree, an 11-dimensional binary vector, describes the number of surrounding atoms within a cut-off radius parameter. The node degree uniqueness (4-dimensional binary vector) describes the number of unique element types in the surrounding neighborhoods of a central atom. In total, we encode the information of each atom in the crystal structure using a 98-dimensional binary vector and those node feature vectors are propagated during the GNN learning process.

Besides the node feature vectors, the graph representation also includes the information about nodes' connectivity, i.e., edge indexes and edge weight. The node connectivity features are based on the crystal's geometric information. More specifically, we created a set of paired atom indexes by assuming that the chemical bonds are established between the atoms and their closet neighbors. In another word, we calculate the Euclidean distance between every pair of atoms, if the distance values are smaller than the cut-off radius (5 angstroms), we assume that edges are created between those atom pairs and the edge weight parameters are calculated by the corresponding Euclidean distances. Note that the edge weight parameter is used as a normalization factor for the atom's node feature vector for learning the GNN model. Furthermore, the cut-off radius is considered as a crucial free parameter and the value of 5 angstroms is determined in a trial-and-error fashion. Figure 3.2 shows the schematic of the graph representation for encoding node features and the node connectivity of crystal structure.

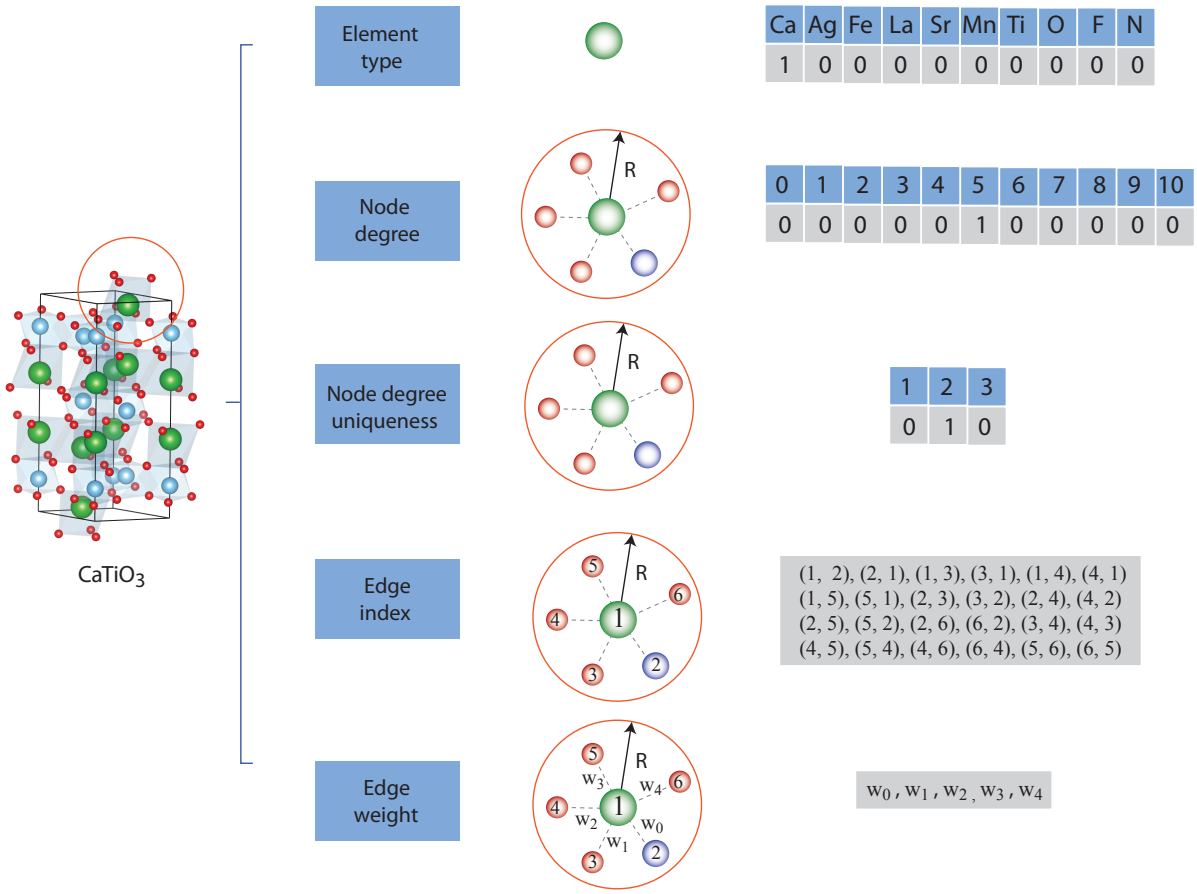

Figure 3.2: Schematic display of graph representation for crystal structure.

3.4.2 Graph neural network configuration

Graph convolutional network model

In this study, we use PyTorch Geometric[52], an open-source deep-learning library, for learning the crystal graphs for the desired DFT-calculated formation energy. PyTorch Geometric builds on top of the well-known PyTorch[53] deep learning library and contains a variety of cutting-edge graph convolutional methods from recent publications. The graph convolutional network (GCN) model is the most basic GNN model. The theory of GCN was first introduced by Schlichtkrull et al.[54] in 2017. This model has been proven effective to han-

dle some standard statistical relation modeling problems such as link prediction and entity classification. See Chapter 1 for a detailed description of the GCN theory. Here, we invested the performance of GCN in the regression tasks for the quantum chemical calculations. For each crystal graph, We build 2 consecutive GCN convolutional layers followed by another 2 linear embedding layers with the purpose of propagating the node information in the crystal structure. The first GCN layer is responsible for mapping the original 98-dimensional node feature vector to a 196-dimensional feature vector by propagating the topological information of the crystal graph. The second GCN convolutional layer projects the 196-dimensional feature vector to the high dimension of 392 in the same way. The linear embedding layers with 196×1024 and 1024×128 dimensional trainable weight parameters are established with the purpose of improving the discriminative performance of machine learning models. Besides GCN convolutional layers, we also include 2 16×16 consecutive linear embedding layers for processing the 10-dimensional composition atomic properties. In the end, all the extracted latent vectors are merged and fed into the feed-forward neural network (ANN) for predicting the target property. The ANN contains 3 linear embedding layers with a dimension of 256×512, 512×256, and 256×1.

Graph attention network model

The graph attention network (GAT) model is an advanced graph neural network model. Unlike GCN, the convolutional process is achieved by an intelligent self-attention propagating mechanism. According to Knyazev et al.,[55] the GAT model provides extraordinary prediction performance than other GNN models under certain conditions such as learning the noisy graphs. See Chapter 1 for a detailed explanation of the GAT model theory. In this work, We implement the GAT model for the learning of crystal structures and evaluate the model performance. The GAT model architecture has an analogy to the GCN model ex-

cept for the graph convolutional layers. For the GAN model, we implement 2 convolutional layers with 98×196 and 196×128 dimensions for processing the graph information. Then, a combined feature set including the latent variables obtained from the graph convolutional process and the atomic properties are used as input for the 3-layer ANN model. The ANN model has the exact same model architecture with the GCN model.

Graph isomorphism network model

The graph isomorphism network (GIN) model[28] is a robust variant version of GNN model frameworks. The GIN model has been widely recognized as one of the state-of-the-art GNN models. It shows great discriminative power in a range of machine learning tasks, which is attributed to a recursively aggregating and transforming mechanism. See Chapter 1.3.3 for detailed information about the GIN theory. In this work, we evaluate the performance of the GIN model for predicting a diverse set of crystal structures. Unlike GCN and GAT models, the GIN model is dependent on a series of internal linear processing units (ReLU activation functions), which are responsible for propagating the information in the neighboring nodes. Here, we apply 5 GIN convolutional layers for learning on the crystal structure. Each GIN convolutional layer is implemented with a 2-layer ANN model with 128×128 processing units.

3.4.3 Model training

We split the complete crystal data (∼23029 samples) into two subsets with 75% for training the graph neural network models and the rest 25 % for testing the model performance. To prevent model overfitting caused by the excessive number of epochs, we calculate the testing error at each epoch, and the optimal model parameters are determined at the bottom of the learning curve as shown in Figure 3.3. Besides the number of training epochs, the free

parameters such as learning rate (0.001), training batch size (100) are determined in a trial-and-error fashion. It is noteworthy that all the learning tasks are accomplished on the GPU servers. The GPU server contains 2 nodes and each node is implemented with 12 processors. The run time cost for training the GNN models (100 epochs) on GPU server is around 15 minutes Figure 3.3 displays the learning curve of the 4 investigated GNN models. In Figure 3.3, we observe that the GCN model outperforms other GNN models with the smallest testing RMSE (0.25 eV). In contract, the GAT model performs the worst among all the GNN models. Note that the GCN_GIN model is a hybrid GNN model, which includes a combined graph convolutional layers of GCN and GIN.

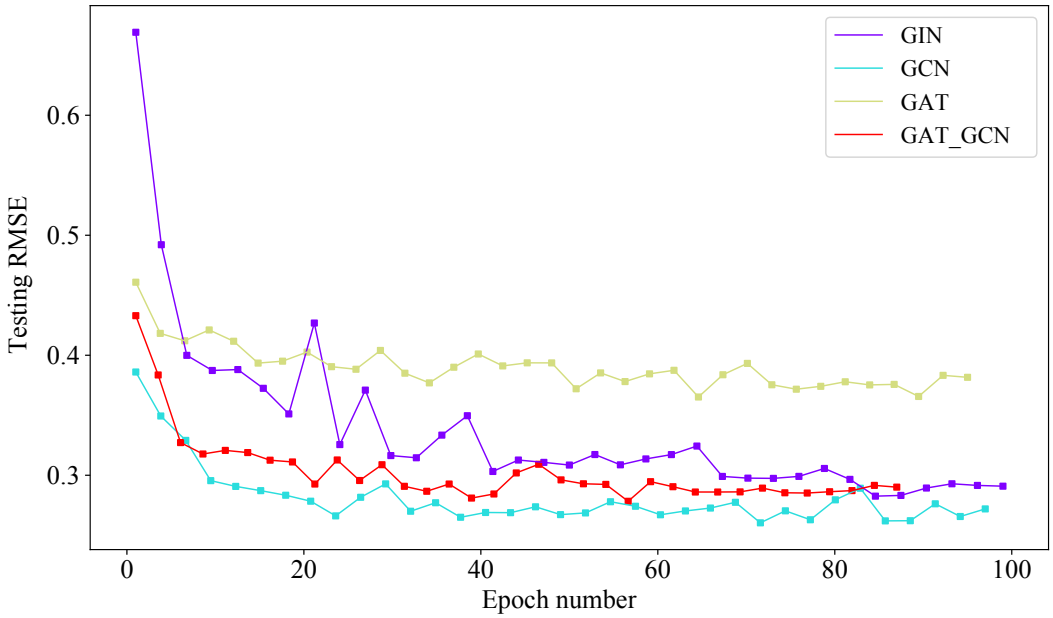

Figure 3.3: Testing RMSE vs. number of training epoch of GNN models.

3.4.4 Model performance evaluation

Accuracy test

Compared to the CNN model in Chapter 2, most GNN models work extremely well for capturing the crystal formation energies with small prediction error (∼0.25 eV) except for GAT. More importantly, the GNN models are capable of processing a large number of materials, which is attributed to the accessible features (crystal structure). Here, the model prediction error is used as the evaluation matrix for the model assessment. Specifically, the GNN models learned by 75% randomly sampled data are applied to make predictions on the rest of 25% data for testing the model performances. According to Figure 3.4(a)(b)(c)(d), we know that the GAT model provides the worst training performance while the other GNN models perform very similarly in terms of root-mean-square error (RMSE) and coefficient of determination (R^2). Besides, we observe many significant outliers in the parity plots. One of the main causes of the outliers is the badly converged DFT-calculated crystals.

To evaluate the model prediction performance for the crystal structure with different ligand groups (i.e., O, N, F, S), we trained the GCN models for the subset data containing different ligand elements. The primary goal of this particular analysis is to assess the model's robustness corresponding to diverse crystal categories. According to Figure 3.5, the training results are varied significantly with respect to different crystal ligand groups. The GNN models trained by ABO_3 and ABS_3 crystal systems perform extremely well with the smallest root-mean-square error (RMSE) while the GNN models for crystal systems of ABN_3 and ABF_3 perform the worst. A general conclusion can be drawn that increasing the training sample size may have a positive impact on model performance.

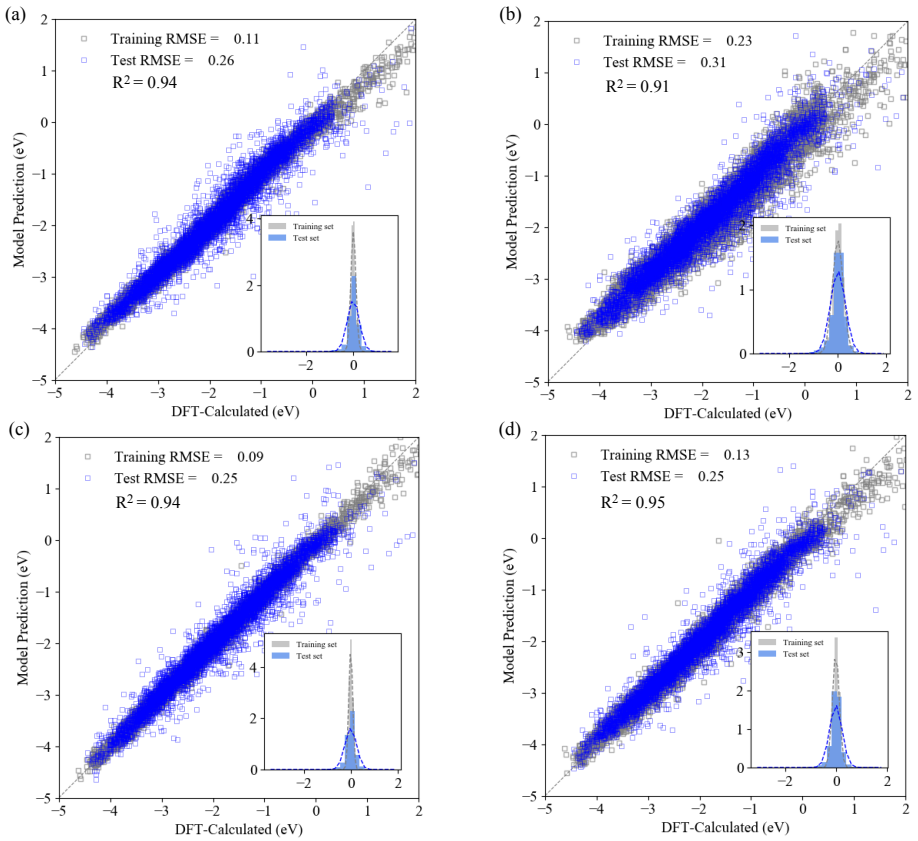

Figure 3.4: GNN model performance evaluation. Parity plots for (a) GCN model trained by 23021 ABX_3 samples, (b) GAT model trained by 23021 ABX_3 samples, (c) GAT_GCN model trained by 23021 ABX_3 samples, (d) GIN model trained by 23021 ABX_3 samples. The inserts show the error distributions.

Training sample size effect

Besides the general accuracy test, we also invest in the GNN model generalization power corresponding to the training data size. Here, generalization power describes the stability of model prediction results with respect to the unknown data. Usually, a robust deep learning model has small prediction variance for the target properties without a large amount of training data. In this work, an empirical experiment test is designed to assess the model's generalization power using the entire crystal data. Specifically, we follow a similar procedure in the Chapter 2.3.3 by inspecting the model prediction performance regarding different

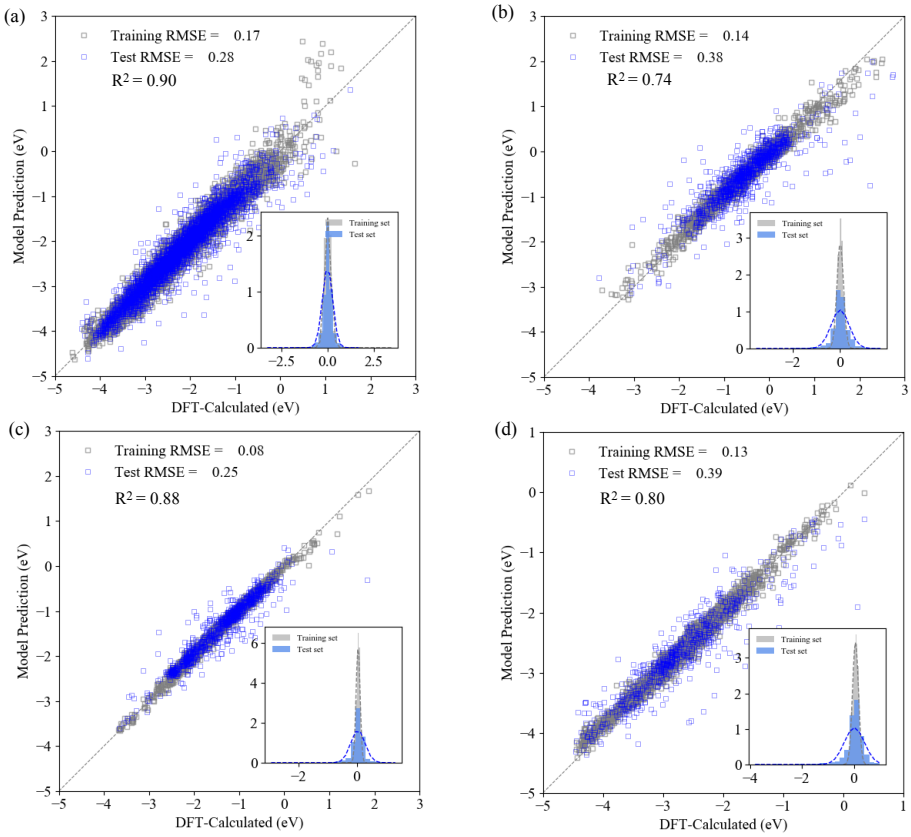

Figure 3.5: GIN model performance evaluation. Parity plots for (a) GIN model trained by 14592 ABO_3 samples, (b) GIN model trained by 3919 ABN_3 samples, (c) GIN model trained by 3553 ABS_3 samples, (d) GIN model trained by 2661 ABF_3 samples. The inserts show the error distributions.

training sample size. Note that in the first evaluation test, we didn't randomly sample the entire crystal data. Thus, the diversity of structural composition has a linear relationship with the training data size. While in the second test, we randomly sampled the training data before training the GNN model, so that both training and testing data contain the same structural compositions. In other words, the randomization effect of training data is diminished in the second evaluation test. According to Figure 3.6, we observed that the prediction error (with randomization effect) of the GCN model is below 0.3 eV after the training data ratio of 0.4, which indicates that the GCN model exhibits good generalization

power corresponding to the unknown testing data. On the other hand, the GCN model shows poor generalization performance for low accuracy values across all the training ratios (no randomization effect). Therefore, we would strongly recommend increasing the diversity of crystal compositions in the training data for maximum model performance.

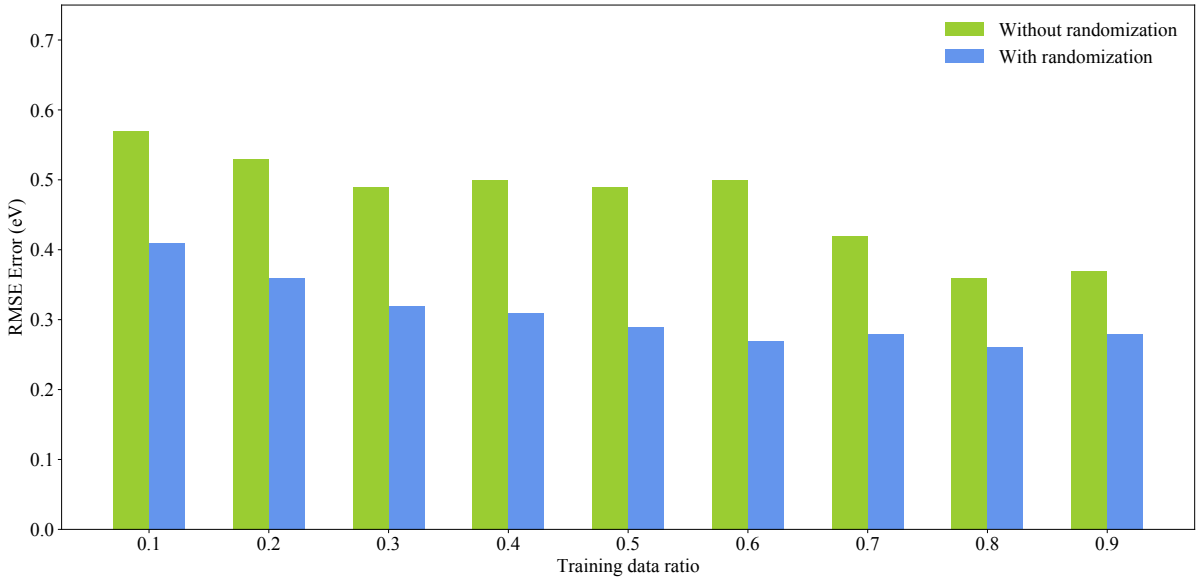

Figure 3.6: GCN model generalization evaluation for different training data ratio. The blue/green bar indicates the model prediction error for the training data w/o randomization.

Cut-off radius analysis

In this section, we evaluate the effect of the cut-off radius, as shown in Figure 3.2, for the model prediction accuracy. The cut-off radius is of vital importance toward crystals' graph configurations. More specifically, the connectivity information of the crystal graphs are predominantly influenced by the magnitude of the cut-off radius, thus subsequently changing the model training process. Typically, a larger cut-off radius parameter indicates that more atoms are considered as neighbors for a given central atom, which increases the complexity of the GNN models. In contrast, a smaller cut-off radius parameter would lead

to under fitted GNN models. In this regard, we conducted an empirical test to determine the appropriate cut-off radius parameter for training the GNN models. Figure 3.7 describes the GIN model training performances along with the distributions of connectivity information (i.e., edge size, number of edges) across all the crystal materials for various sets of cut-off radius parameters.

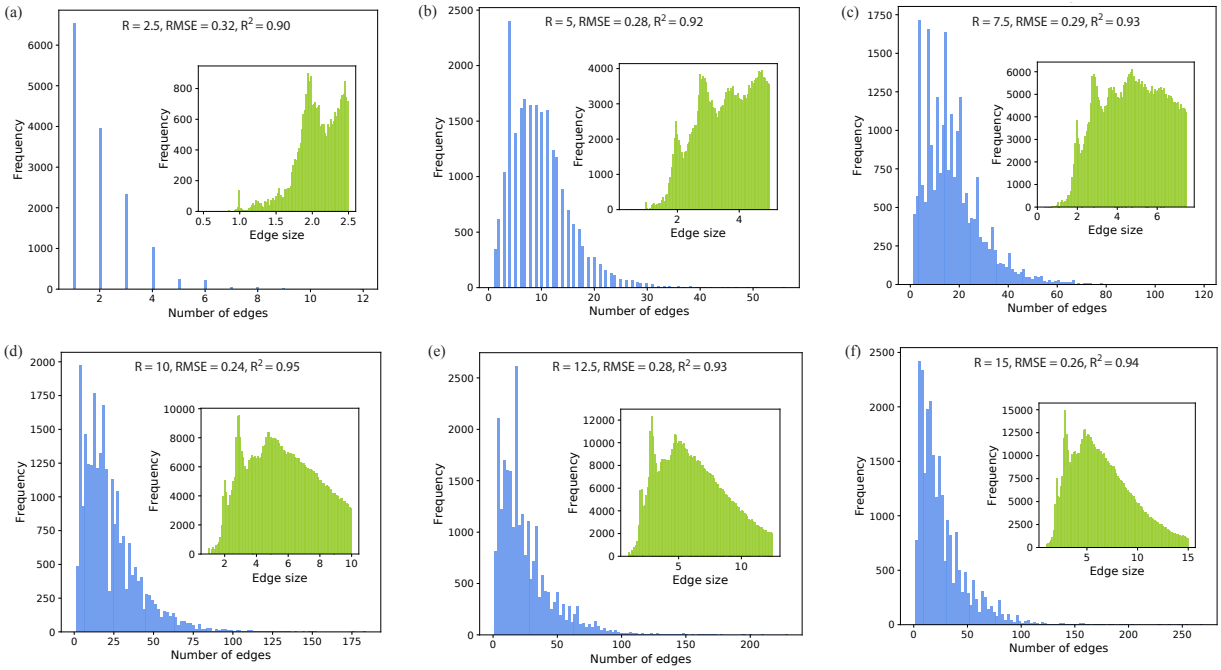

Figure 3.7: Cut-off radius parameter evaluation for GIN model. The distribution of edge size and number of edges for GIN model trained with (a) R = 2.5, (b) R = 5, (c) R = 7.5, (d) R = 10, (e) R = 12.5, (f) R = 15.

According to Figure 3.7, we observe that the magnitude of the cut-off radius parameter plays an important role in GIN model performance. The model RMSE errors show a volcano relationship with respect to the values of cut-off radius parameters. The optimal cut-off radius parameter is 10 for the smallest RMSE and Highest R^2. In addition, the distributions of edges are similar for Figure 3.7(d)(e)(f), which indicates that the graph connectivity information is constant after a specific cut-off radius parameter (R = 10).

Run-time analysis

We conducted the run-time analysis in this work by comparing the training computation time of all the proposed deep learning models including a series of CNN and GNN models trained by different features (i.e., PDOS, AP) for 100 epochs. The available data includes the formation energies and the structural properties of 785 ABO_3-type metal oxides. To consider the data randomization effect, we randomly select 75% of data for training the machine learning model and this procedure was repeated for 4 times. The average value of the 4 measurements indicates the model general computational time. Note that all the learning tasks in this work were accomplished on the GPU server from Advanced Research Computing (ARC)[56] of Virginia Tech. The GPU compute engine contains an Intel Skylake Xeon Gold 3 Ghz-core machine (12-core processors) with an NVIDIA V100 (Volta) GPU for training the deep learning models. Figure 3.8 shows the averaged training computation time of all the deep learning models along with their standard deviations with similar computing resources. According to Figure 3.8, we observe that the deep learning models with diverse model structures and features have similar training computation costs. Among all the deep learning models, the GCN model has the smallest computation time while the CNN model trained by both electronic structure and atomic property features has the highest running time. Therefore, a general conclusion can be drawn that all the deep learning models (i.e., CNNs and GNNs) have efficient computational time for processing a large amount of data. However, the GNN models are scalable for exploring broad material structures compare to CNN models because of the easily accessible feature representations (i.e., crystal graphs), which doesn't rely on any prior DFT calculations like the PDOS in CNN models.

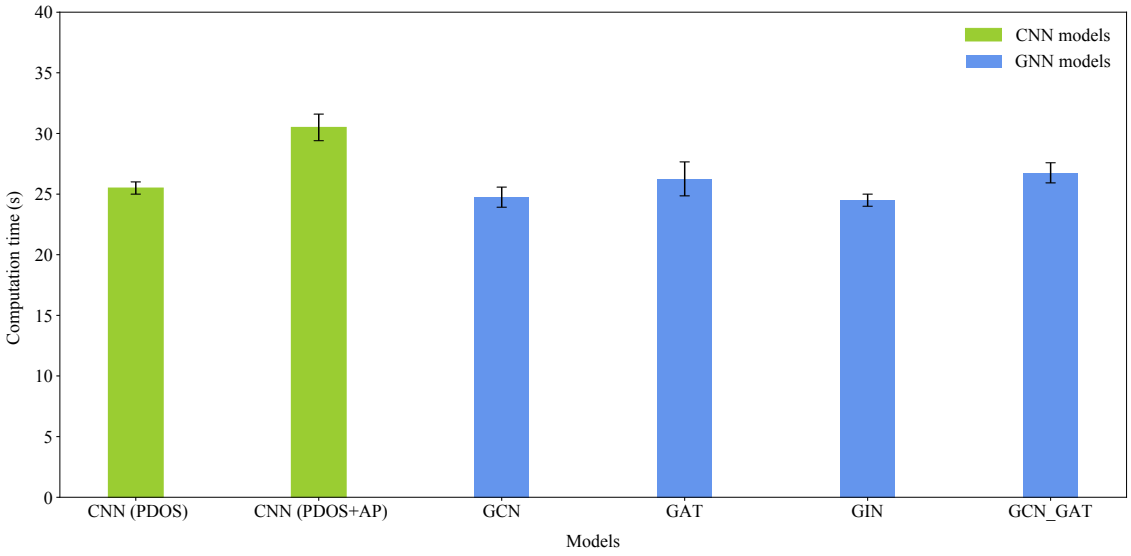

Figure 3.8: Averaged training computation time of CNN and GNN models. The black error bars indicate the standard deviations of the 4 measured training computation time.

3.5 Conclusions

In summary, we develop holistic graph neural network (GNN) frameworks for predicting the expensive quantum-chemical calculated properties (i.e., formation energy) of crystal structures. By learning the easily accessible crystal graphs, the GNN models can rapidly investigate a broad chemical space with enormous compositional and configurational degrees of freedom. The GNN models exhibit high prediction accuracy for the formation of energy ($\sim$0.25 eV), which is comparable to the system error ($\sim$0.2 eV) of density functional theory (DFT). To assess the generalization power of the GNN models, we conduct a series of experiment tests by evaluating the model performance for learning distinguished training samples. We show that the GNN models are able to capture an enormous amount of crystal properties by learning a relatively small number of samples. Compared to CNN models in Chapter 1, the GNN models have the predominant advantages in terms of data accessibility, which is largely attributed to the special graph feature representation method and advanced message

passing process. However, the CNN models have an advantage over the GNN models in the aspect of model interpretability. Future work can focus on the interpretability of GNN models in the domain application of high-throughput material screening. A detailed empirical comparison of our proposed GNN framework with existing GNN literature in material science can also be conducted in a future study.

References

[1] A material revolution: How new materials are changing the manufacturing industry. https://www.sandvik.coromant.com/en-us/aboutus/lookingahead/articles/pages/a-material-revolution.aspx. Accessed: 2020-5-14.

[2] Keith T Butler, Gopalakrishnan Sai Gautam, and Pieremanuele Canepa. Designing interfaces in energy materials applications with first-principles calculations. *npj Computational Materials*, 5(1):19, February 2019.

[3] National Research Council (US) Committee on Biomolecular Materials and Processes. *Advanced Functional Materials*. National Academies Press (US), 2008.

[4] Randal S Olson, Ryan J Urbanowicz, Peter C Andrews, Nicole A Lavender, La Creis Kidd, and Jason H Moore. Automating biomedical data science through Tree-Based pipeline optimization. In *Applications of Evolutionary Computation*, pages 123–137. Springer International Publishing, 2016.

[5] Nathan Argaman and Guy Makov. Density functional theory – an introduction. June 1998.

[6] W Kohn and L J Sham. Self-Consistent equations including exchange and correlation effects. *Phys. Rev.*, 140(4A):A1133–A1138, November 1965.

[7] L H Thomas. The calculation of atomic fields. *Math. Proc. Cambridge Philos. Soc.*, 23(5):542–548, January 1927.

[8] Jean-Louis Calais. Density-functional theory of atoms and molecules. R.G. parr and w.

yang, oxford university press, new york, oxford, 1989. IX + 333 pp. price £45.00: Book review. *Int. J. Quantum Chem.*, 47(1):101–101, July 1993.

[9] J P Perdew, J A Chevary, S H Vosko, K A Jackson, M R Pederson, D J Singh, and C Fiolhais. Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation. *Phys. Rev. B Condens. Matter*, 46(11):6671–6687, September 1992.

[10] Stephen Cole Kleene. Representation of events in nerve nets and finite automata. Technical report, RAND PROJECT AIR FORCE SANTA MONICA CA, 1951.

[11] R H Hahnloser, R Sarpeshkar, M A Mahowald, R J Douglas, and H S Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951, June 2000.

[12] S Hochreiter and J Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In F Pereira, C J C Burges, L Bottou, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*.

[15] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Comput.*, 29(9):2352–2449, September 2017.

[16] Fernando Gama, Antonio G Marques, Geert Leus, and Alejandro Ribeiro. Convolutional neural network architectures for signals supported on graphs. *Trans. Sig. Proc.*, 67(4):1034–1049, February 2019.

[17] Serkan Kiranyaz, Serkan Kiranyaz, Turker Ince2, Osama Abdeljaber3, Onur Avci3, Gabbouj4, and Moncef. 1-D convolutional neural networks for signal processing applications. May 2019.

[18] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of CNN and RNN for natural language processing. February 2017.

[19] Yoon Kim. Convolutional neural networks for sentence classification. August 2014.

[20] Thomas N Kipf and Max Welling. Semi-Supervised classification with graph convolutional networks. September 2016.

[21] Sami Abu-El-Haija, Amol Kapoor, Bryan Perozzi, and Joonseok Lee. N-GCN: Multi-scale graph convolution for semi-supervised node classification. February 2018.

[22] Le Wu, Peijie Sun, Richang Hong, Yanjie Fu, Xiting Wang, and Meng Wang. SocialGCN: An efficient graph convolutional network based model for social recommendation. November 2018.

[23] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3D human pose regression. April 2019.

[24] Tae San Kim, Won Kyung Lee, and So Young Sohn. Graph convolutional network approach applied to predict hourly bike-sharing demands considering spatial, temporal, and global effects. *PLoS One*, 14(9):e0220782, September 2019.

[25] Yongji Wu, Defu Lian, Shuowei Jin, and Enhong Chen. Graph convolutional networks on user mobility heterogeneous graphs for social relationship inference. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019.

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. October 2017.

[27] Han Altae-Tran, Bharath Ramsundar, Aneesh S Pappu, and Vijay Pande. Low data drug discovery with One-Shot learning. *ACS Cent Sci*, 3(4):283–293, April 2017.

[28] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? October 2018.

[29] L Itti, C Koch, and E Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254–1259, November 1998.

[30] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. December 2013.

[31] Neil W Ashcroft and N David Mermin. *Solid State Physics*. Holt, Rinehart and Winston, 1976.

[32] Jens K Nørskov, Frank Abild-Pedersen, Felix Studt, and Thomas Bligaard. Density functional theory in surface chemistry and catalysis. *Proc. Natl. Acad. Sci. U. S. A.*, 108(3):937–943, January 2011.

[33] Zheng Li, Xianfeng Ma, and Hongliang Xin. Feature engineering of machine-learning chemisorption models for catalyst design. *Catal. Today*, 280:232–238, February 2017.

[34] Zheng Li, Siwen Wang, Wei Shan Chin, Luke E Achenie, and Hongliang Xin. High-throughput screening of bimetallic catalysts enabled by machine learning. *J. Mater. Chem. A Mater. Energy Sustain.*, 5(46):24131–24138, November 2017.

[35] Zheng Li, Luke E K Achenie, and Hongliang Xin. An adaptive machine learning strategy for accelerating discovery of perovskite electrocatalysts. *ACS Catal.*, pages 4377–4384, March 2020.

[36] Xianfeng Ma, Zheng Li, Luke E K Achenie, and Hongliang Xin. Machine-Learning-Augmented chemisorption model for CO2 electroreduction catalyst screening. *J. Phys. Chem. Lett.*, 6(18):3528–3533, September 2015.

[37] Zheng Li, Siwen Wang, and Hongliang Xin. Toward artificial intelligence in catalysis. *Nature Catalysis*, 1(9):641–642, September 2018.

[38] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. September 2011.

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

[40] K T Schütt, H Glawe, F Brockherde, A Sanna, K R Müller, and E K U Gross. How to represent crystal structures for machine learning: Towards fast prediction of electronic properties. *Phys. Rev. B Condens. Matter*, 89(20):205118, May 2014.

[41] Atsuto Seko, Atsushi Togo, and Isao Tanaka. Descriptors for machine learning of materials data. In Isao Tanaka, editor, *Nanoinformatics*, pages 3–23. Springer Singapore, Singapore, 2018.

[42] Jonathan Schmidt, Mário R G Marques, Silvana Botti, and Miguel A L Marques. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):83, August 2019.

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

[44] Digvijaysinh K Parmar and H D Nakum. *Crystal Field Theory: Introduction to Crystal Field Splitting*. LAMBERT Academic Publishing, September 2018.

[45] Francesc Illas. Fundamental concepts in heterogeneous catalysis. by jens k. nørskov, felix studt, frank Abild-Pedersen and thomas bligaard. *Angew. Chem. Int. Ed.*, 54(36):10404–10405, September 2015.

[46] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[47] Ghanshyam Pilania, Prasanna V Balachandran, Chiho Kim, and Turab Lookman. Finding new perovskite halides via machine learning. *Frontiers in Materials*, 3:19, 2016.

[48] Thin Nguyen, Hang Le, and Svetha Venkatesh. GraphDTA: prediction of drug–target binding affinity using graph convolutional networks. June 2019.

[49] Connor W Coley, Wengong Jin, Luke Rogers, Timothy F Jamison, Tommi S Jaakkola, William H Green, Regina Barzilay, and Klavs F Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.*, 10(2):370–377, January 2019.

[50] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In C Cortes, N D Lawrence, D D Lee, M Sugiyama, and R Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2224–2232. Curran Associates, Inc., 2015.

[51] Tian Xie and Jeffrey C Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.*, 120(14):145301, April 2018.

[52] Matthias Fey. pytorch_geometric.

[53] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, High-Performance deep learning library. December 2019.

[54] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. March 2017.

[55] Boris Knyazev, Graham W Taylor, and Mohamed R Amer. Understanding attention and generalization in graph neural networks. May 2019.

[56] Nicholas Fearing Polys, Sara Sweeney Ackerson, and Robert E Settlage. Advanced research computing at virginia tech. `https://arc.vt.edu/`. Accessed: 2020-7-19.