Advances in Applied Sciences
2024, Vol. 9, No. 2, pp. 17-30
https://doi.org/10.11648/j.aas.20240902.11

Research Article

Recent Advances and Applications of Graph Convolution Neural Network Methods in Materials Science

Ke-Lin Zhao[1], Qing-Xu Li[1, 2, *]

[1]School of Science, Chongqing University of Posts and Telecommunications, Chongqing, China
[2]Institute for Advanced Sciences, Chongqing University of Posts and Telecommunications, Chongqing, China

Abstract

With the development of artificial intelligence (AI), AI plus science is increasingly valued, presenting new perspectives to scientific research. The research on using machine learning (including deep learning) to discover patterns from data and predict targeted material properties has received widespread attention, which will have a profound impact in material science studies. In recent years, there has been an increased interest in the use of deep learning in materials science, which has led to significant progress in both fundamental and applied research. One of the most notable advancements is the development of graph convolutional neural network models, which combine graph neural networks and convolutional neural networks to achieve outstanding results in materials science and bridge effectively the deep learning models and material properties predictions. The availability of large materials databases due to the rise of big data has further enhanced the relevance of these models in the field. We present, in this article, a comprehensive overview of graph convolutional neural network models, explaining their fundamental principles and highlighting a few examples of their applications in materials science, as well as current trends. The limitations and challenges that these models face, as well as the potential for future research in this dynamic area are also discussed.

Keywords

Materials Science, Deep Learning, Graph Convolutional Neural Network

1. Introduction

The exploration of high-performance materials, along with the comprehension of structure-activity relationships and issues concerning the chemical structure of materials, presents an exciting scientific endeavor with profound societal relevance [1]. It is well-established that computational simulations and experimental measurements are the two principal methodologies employed in materials science. Experimental measurement offers a direct and intuitive method for investigating materials but tends to be less efficient over long durations. This approach requires sophisticated equipment, precise experimental conditions, and a high level of expertise from researchers. On the other hand, computational simulation encompasses various techniques, such as electronic structure calculations via density functional theory (DFT) [2, 3], molecular dynamics [4, 5], Monte Carlo methods [6], and continuum macroscopic approaches. These simulations employ computer algorithms to leverage existing theories for analytical purposes. Thanks to computational simulations, the field of materials design is advancing toward the discovery of

*Corresponding author: liqx@cqupt.edu.cn (Qing-Xu Li)

Received: 6 April 2024; **Accepted:** 22 April 2024; **Published:** 29 April 2024

novel materials, thereby curtailing the time and costs associated with material development. With the aid of computational simulations, the recent surge in computational power, and the accelerated pace of information exchange, we have seen an exponential expansion in the size of public materials datasets [7-9]. Although computational methods offer a faster and more cost-effective alternative to traditional experiments, they are inherently limited by their underlying models and theories, which restrict their applicability. Concurrently, machine learning (ML) techniques are gaining traction in expediting the design of new materials by predicting material properties with an accuracy that rivals *ab initio* calculations, yet at computational speeds that are orders of magnitude faster [10-12]. The convergence of expansive datasets, algorithmic advancements, and enhanced computational power is driving an unparalleled interest in the application of ML in this domain.

Deep learning (DL) [13], which is a subfield of ML, takes inspiration from the cognitive functions of the human brain and biological models [14, 15]. One of the key strengths of DL is its ability to process vast amounts of high-dimensional data, allowing it to uncover complex features and correlations from raw input data. In the field of material design, DL has been shown to complement physics-based methods effectively [16-18]. Moreover, DL approaches can significantly accelerate processes compared to traditional computational simulation methods, achieving a level of accuracy in specific contexts that rivals that of conventional physics-based or computational models. Despite these advantages, DL methodologies do face challenges. A significant concern within the scientific community is the opaque nature of DL methods, which often doesn't allow for the discovery, comprehension, or application of new principles that they might reveal. This "black-box" characteristic [19] can hinder a deeper physical understanding of the phenomena being investigated. Therefore, there is ongoing research to improve the interpretability and explainability of DL models, which is a vibrant field of study.

The Graph Neural Network (GNN) is a cutting-edge DL approach that enables relational reasoning and compositional generalization. Currently, GNNs are one of the most widely used DL techniques that utilize structural information to predict a diverse range of material properties [16, 20-22]. GNNs come in several variants, each defined by its unique propagation rules and aggregation methods. Among these variants, the Graph Convolutional Neural Network (GCNN) is the most extensively used in the field of GNNs. This article aims to explain the fundamental principles of GCNN methodologies, highlight their applications in the field of materials science, and explore the latest trends and advancements in this area.

2. Neural Network

2.1. Forward Propagation Neural Network

The perceptron, also known as an "artificial neuron" or "naive perceptron," was first proposed in 1957 by Frank Rosenblatt [15]. It serves as a crucial component of artificial neural networks (ANNs) and is responsible for forwarding information. During operation, given a set of inputs $[x_1, x_2, x_3, ..., x_n]$, the perceptron assigns corresponding weights $[w_1, w_2, w_3, ..., w_n]$ and includes a bias term b. It then performs a weighted sum of the inputs, which is a vital step in its computation process.

Activation functions play a vital role in ANNs by introducing nonlinearity and enabling the network to approximate complex nonlinear functions. This ability allows ANNs to solve tasks that involve learning intricate patterns and relationships in data. Various types of activation functions are used in ANNs, including sigmoid, hyperbolic tangent (tanh), rectified linear unit (ReLU), leaky ReLU, and Swish. For instance, the sigmoid function is useful for binary classification tasks as it outputs values in the range (0, 1) and is adept at mapping a real number to this interval. It is effective in scenarios where the features exhibit complex interactions or when the distinctions among features are subtle. Similarly, the tanh function scales real values to fall within -1 and 1, offering a centered range compared to sigmoid. Selecting an appropriate activation function is crucial since it significantly influences the network's training dynamics and the model's ultimate performance [23].

In the realm of neural networks, initializing weight matrices is a crucial step that often involves assigning random values or utilizing pre-trained models. These weights play a crucial role in transforming input matrices (or outputs from previous layers) through multiplication to create new representations. Non-linear activation functions are then applied to introduce non-linearity into the model. The loss function is pivotal in evaluating the network's performance, and it involves comparing the network's output with the actual target values. This function, also known as the objective function or empirical risk, quantifies the difference between predicted and true values. To optimize the network, back-propagation is used, which involves calculating the partial derivatives of the loss function concerning each network parameter. This is achieved through the chain rule and reverse automatic differentiation, enabling the iterative minimization of the loss function [24]. Among the gradient descent algorithms utilized for this optimization process are stochastic gradient descent (SGD), Adam, and Adagrad. SGD is a popular choice for iteratively updating network weights to reduce the loss function towards the targeted accuracy. Other algorithms adaptively adjust learning rates. The learning rate is a significant factor in the gradient descent process, influencing the speed of convergence and the stability of network training. Different loss functions, such as binary cross-entropy (BCE), negative log-likelihood (NLLL), or mean absolute error (MAE), are employed based on specific objectives, such as classification or regression, to tailor the training process. Deep learning models, especially neural networks, are sensitive to the scale of input data, requiring careful preprocessing. When features have varying scales, it can lead to the dominance of certain

features during training, potentially overshadowing others, and causing numerical stability issues like gradient explosion or vanishing gradients. To mitigate these issues, inputs are usually scaled to a uniform range. Common scaling techniques include normalization (scaling data to [0,1]) and standardization (scaling data to have a mean of 0 and a standard deviation of 1). Additionally, techniques like batch normalization or layer normalization are applied to inputs of hidden layers to enhance the stability and performance of ANNs.

The multilayer perceptron (MLP), also known as forward neural network, is one of the most elementary architectures in neural networks. It consists of several layers that perform linear transformations interspersed with non-linear activation functions. Each layer in this network processes an input vector $\mathbf{x} \in \mathbb{R}^m$ using the transformation:

$$f(\mathrm{x};\mathrm{w},\mathrm{b}) = \mathrm{w}^T\mathrm{x} + \mathrm{b}, \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^{m \times n}$ is the weight matrix and $\mathbf{b} \in \mathbb{R}^n$ is the bias vector. Both $\mathbf{w}$ and $\mathbf{b}$ are adjustable parameters within the network. By integrating non-linear activation functions like ReLU or tanh on f, the network is able to model complex relationships beyond linear mappings. A forward neural network is essentially a sequential assembly of such layers, depicted as $f^{(k)}\left(f^{(k-1)}\left(\cdots f^{(1)}(\mathbf{x})\cdots\right)\right)$, where k indexes the layers. This stacked structure enables the network to learn hierarchical representations of data. Compared to a general neural network, a forward neural network has simpler input data vectors and more straightforward inter-layer connectivity. It lacks the complex feedback loops or recurrent structures found in more advanced networks. Nonetheless, the core principle underpinning both network types is leveraging a concatenation of simple computational units to craft a model capable of capturing and expressing intricate patterns within the data.

2.2. Graph Neural Network

In mathematical terms, a graph G is represented as a trio of nodes (or vertices) V, edges (or links) E, and node features X, expressed as $G = (V, E, X)$. Edges connect pairs of nodes, encapsulating the relational dynamics between them. Both nodes and edges may possess associated attributes or features that define their characteristics. The adjacency matrix A serves as a square matrix that delineates the connections between nodes, using 1 to denote a connection and 0 for its absence.

Graphs are classified into various categories, such as undirected or directed, weighted or unweighted, homogeneous or heterogeneous, and static or dynamic. Undirected graphs represent bidirectional, symmetrical relationships between nodes, while directed graphs illustrate unidirectional, asymmetrical relationships, where the condition $A_{mn} \neq A_{nm}$ may hold. Weighted graphs assign a scalar value to each edge to denote its weight, contrasting with unweighted graphs, which use binary values. Homogeneous graphs feature nodes and edges of a single type, signifying a uniform class of entities or relationships. In contrast, heterogeneous graphs consist of multiple types of nodes and edges, indicating a diversity of entities and their interrelations.

GNNs are a subclass of deep learning techniques designed to process graph-structured data. They model the dependencies within a graph via a message-passing paradigm, where information is exchanged and processed across nodes and edges. GNN training involves two pivotal processes: (a) aggregating information from neighboring nodes and edges and (b) updating the attributes of nodes and edges, with the aggregation step being designed to be permutation invariant.

Various GNN architectures have been developed, each characterized by its specific rules for propagation and methods for aggregation. Notable examples include the graph convolutional network (GCN) [25], which applies convolutional operations in the graph domain; the graph attention network (GAT) [26], which implements attention mechanisms to weigh the importance of nodes; the relational graph convolutional network (Relational-GCN) [27], tailored for heterogeneous graphs; the graph recurrent network (GRN) [28], which incorporates recurrent neural network principles; the graph isomorphism network (GIN) [29], designed to capture the isomorphic properties of graphs; and the line graph neural network (LGNN) [30], which explores higher-order interactions between graph edges.

2.3. Convolutional Neural Network

The convolutional neural network (CNN), which can be considered an enhanced version of the MLP, introduces a robust inductive bias that is particularly adept at learning translation-invariant features from image data. This characteristic makes CNNs exceptionally suited for image processing and recognition tasks. The architecture of a CNN encompasses four primary elements: (a) learnable convolution filters, (b) nonlinear activation functions, (c) spatial reduction techniques such as pooling or stridden convolutions, and (d) a prediction mechanism, usually consisting of fully connected layers that synthesize the extracted features into a global representation for the final output. In the convolutional layers, CNNs employ trainable filters or kernels, which are small matrix-like structures that traverse the input image in a sliding manner, applying element-wise multiplication to capture localized feature patterns. This process produces feature maps that highlight the presence of specific features at different locations in the input. By using these shared weights across the entire input space, CNNs can detect features regardless of their position in the image, embodying the concept of translation invariance.

Following the convolution operation, nonlinear activation

functions, such as ReLU, are applied to introduce nonlinearity into the model, enabling it to capture complex patterns and dependencies in the data. The spatial reduction component, often realized through pooling operations, serves to downsample the feature maps, reducing their dimensionality while retaining essential information. Pooling can be performed in various ways, including max pooling, average pooling, and sum pooling, each contributing to the model's robustness to slight variations in the input.

The culmination of these processes in a CNN is a series of convolutional and pooling layers that progressively abstract the input's features, culminating in a dense layer that integrates these features into a comprehensive representation. This representation is then used in the prediction module, typically consisting of one or more fully connected layers, to make final decisions or classifications. Through this sophisticated architecture, CNNs have established a benchmark in the field of image analysis, demonstrating remarkable efficacy in tasks like image classification and object recognition [31, 32].

2.4. Graph Convolutional Neural Network

GCN is an extension of CNNs that is specifically designed to model data with a graph structure. In a GCN, the data is represented as a graph $\mathcal{G}=(\mathcal{V},\varepsilon)$, where $\mathcal{V}$ is the set of nodes and ε is the set of edges. The graph has N nodes, denoted by $v_i \in \mathcal{V}$, connected by edges $(v_i, v_j) \in \varepsilon$. The connections between the nodes are described by the adjacency matrix $A \in \mathbb{R}^{N\times N}$, which can be either binary or weighted, and the degree matrix D, which is a diagonal matrix where $D_{ii}=\sum_j A_{ij}$.

In GCNs, the convolution operation is adapted to handle graph data. The propagation rule is given by:

$$H^{l+1}=\sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{l}W^{l}), \tag{2}$$

where $\tilde{A}=A+I_N$ represents the adjacency matrix with added self-connections (via the identity matrix I_N), which enhances the flow of information. $\tilde{D}$ is the degree matrix corresponding to $\tilde{A}$, and W^l denotes the layer-specific trainable weight matrix. The matrix $H^l \in \mathbb{R}^{N\times D}$ contains the node features or activations at layer l, with $H^{(0)}=X$ being the initial node features.

The convolution operation in GCNs can be understood as a signal processing operation, where the convolution of a signal x (node features) with a filter g is defined as:

$$g \star x = UgU^{\top}x, \tag{3}$$

where U is the matrix of eigenvectors of the graph Laplacian. This transformation allows the filter g to be applied in a manner that respects the graph structure, ensuring that the convolution is sensitive to the patterns of connectivity within the graph.

By iteratively applying this convolution process, followed by nonlinear activation functions (and potentially pooling layers), GCNs can effectively capture and model the intricate structures and feature relationships present in graph data. This ability makes GCNs particularly powerful for tasks where data is inherently structured in non-Euclidean domains, such as social networks, molecular structures, and communication networks.

3. Application of Graph Convolution Neural Network

The successful implementation of DL in materials science relies on several crucial factors that ensure the reliability and relevance of the developed models. These factors include:

Acquisition of Large, Balanced, and Diverse Datasets: In order to train DL models effectively, it is necessary to obtain extensive datasets that accurately reflect the diversity of materials and their properties. These datasets should be balanced to avoid biases towards particular material classes or properties, ensuring that the model can generalize well across various materials.

Determination of Appropriate Representations for DL Methods and Input Samples: The way materials data is represented can significantly influence the effectiveness of DL models. In materials science, this might involve choosing the proper descriptors or features that capture the essential attributes of materials. For graph-based models like GCNs, this includes deciding how to represent materials as graphs (e.g., atoms as nodes and bonds as edges) and defining node and edge features that accurately reflect the materials' characteristics.

Selection of Performance Metrics Relevant to the Target Properties: The metrics used to evaluate the performance of DL models should align with the specific goals of the materials science applications. For instance, if the objective is to predict a material's electronic properties, the chosen metrics should reflect the accuracy and reliability of those predictions.

When it comes to the application of GCNs in materials science, these considerations take on specific nuances:

Data Representation: Materials must be represented as graphs for GCNs. The challenge lies in effectively translating the material's atomic or molecular structure into a graph format where nodes and edges correspond to atoms and bonds, respectively. Additional features such as atomic number, charge, or other chemical properties can be included as node or edge attributes.

Graph-Based Data Processing: GCNs' unique advantage is their ability to process data in graph form, capturing the local

and global structural information inherent in materials. This capability allows GCNs to learn from the complex interactions within materials, offering insights that might be inaccessible to other types of DL models.

Domain-Specific Applications: In materials science, GCNs can be applied to a range of tasks, from predicting material properties based on their atomic structures to identifying new materials with desired characteristics. The choice of network architecture, learning algorithms, and training techniques must be tailored to these specific applications to maximize the effectiveness of the GCNs.

In the following section, we will explore the applications of graph convolutional neural networks in materials science in greater detail, showcasing their potential to revolutionize our understanding and discovery of new materials.

3.1. Crystal Graph Convolution Neural Network

As we discussed earlier, the application of graph convolutional neural networks (GCNNs) in the field of materials science requires the development of appropriate representational frameworks. One of the major challenges in this context is to represent periodic crystal systems, which come in varying sizes, as fixed-length vector representations that effectively correlate with their target properties. Traditionally, this problem has been addressed through the creation of fixed-length feature vectors, derived from simple material properties [33-35], or by using symmetry-invariant transformations of atomic coordinates [36, 37]. The former approach requires specific designs for predicting multiple properties, making it somewhat rigid. On the other hand, the latter approach often results in complex transformation models that are difficult to interpret. These limitations have hindered the broader application of machine learning in this field. To overcome these challenges, Xie et al. [20] proposed a novel approach that employs graph construction for feature extraction, which is encapsulated in the crystal graph convolutional neural network (CGCNN) framework, as illustrated in Figure 1.

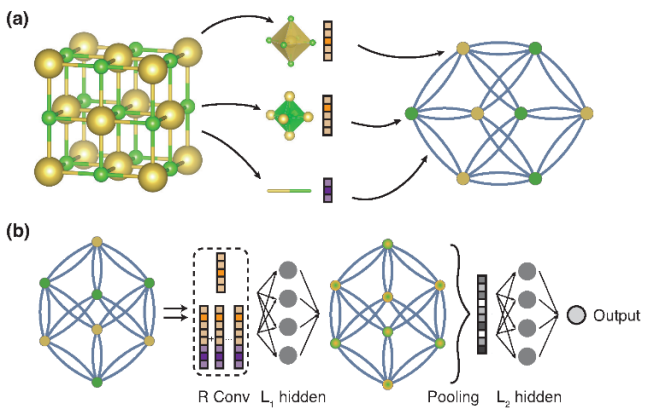

Figure 1. *Illustration of the CGCNN. (a) Construction of the crystal graph. (b) Structure of CNN. From [20].*

The constructed graph represents atomic features as nodes, where each node's characteristics are captured in the feature vector v_i. Instead of using continuous values directly, each continuous-valued attribute is segmented into ten distinct categories to establish a one-hot encoded feature vector. The graph's edges symbolize the bonds formed through interactions between atoms, similar to how material fragment descriptors with property labels are structured. A unique aspect of these crystal graphs is their ability to accommodate multiple edges between two nodes, reflecting the periodicity inherent in crystal structures. Consequently, a one-hot encoded edge feature vector $u_{(i,j)_k}$ signifies the k th bond linking atoms i and j.

Although the crystal graph itself may not be the most effective representation for predicting target properties, its representational efficacy is significantly enhanced through the application of convolutional layers. Each convolutional layer's processing results in the progressive integration of environmental information into the node and edge feature vectors. Specifically, the $(t+1)$th layer of the graph neural network updates the feature vector v_i for node i based on the output from the t th layer, as detailed in the following equation:

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \sum_{j,k} \sigma(z_{(i,j)_k}^{(t)} \mathbf{W}_f^{(t)} + \mathbf{b}_f^{(t)}) \odot g(\mathbf{z}_{(i,j)_k}^{(t)} \mathbf{W}_s^{(t)} + \mathbf{b}_s^{(t)}) \quad (4)$$

Here, $\mathbf{W}_f^{(t)}$, $\mathbf{W}_s^{(t)}$, and $\mathbf{b}_i^{(t)}$ represent the convolutional weight matrix, self-weight matrix, and bias at layer t, respectively. The operation $\odot$ denotes element-wise multiplication, σ is the sigmoid function, and $z_{(i,j)_k}^{(t)}$ embodies the concatenation of neighboring vectors, as defined by:

$$z_{(i,j)_k}^{(t)} = \mathbf{v}_i^{(t)} \oplus \mathbf{v}_j^{(t)} \oplus \mathbf{u}_{(i,j)_k} \quad (5)$$

where $\oplus$ represents vector concatenation. After multiple convolutions, spatial dimensions within the neural network are reduced via pooling layers, which operate on all feature vectors obtained throughout the convolutional stages.

The CGCNN model, trained on DFT computational data from the Materials Project, is proficient at predicting various crystal properties, including formation energy, absolute energy, band gap, Fermi energy, bulk moduli, shear moduli, and Poisson's ratio. Notably, the model's mean absolute error (MAE) for formation energy prediction stands at 0.039 eV/atom, surpassing the accuracy of DFT calculations relative to experimental data. This precision is evident when the model is trained with approximately 10^4 data points, where its MAEs approach or exceed the DFT's accuracy in comparison to experimental results. Ultimately, crystal graph convolutional neural networks offer a robust and adaptable framework for predicting material properties and facilitating knowledge

discovery in materials design.

3.2. Improved Crystal Graph Convolution Neural Network

Expanding upon the foundational principles of the crystal graph convolutional neural network, Cheol *et al.* [38] introduced an advanced iteration of the CGCNN model, aptly named improved crystal graph convolutional neural network (iCGCNN), which is depicted in Figure 2. This enhanced model, iCGCNN, builds on the original framework by incorporating additional features or optimizations to improve its predictive accuracy and efficiency in analyzing material properties.

The iCGCNN model introduces enhancements to the original CGCNN to address its three key limitations. The CGCNN is known for its flexibility in representing various crystal structures and predicting diverse material properties. However, it may not effectively capture the chemical environments of atoms due to its crystal graph design.

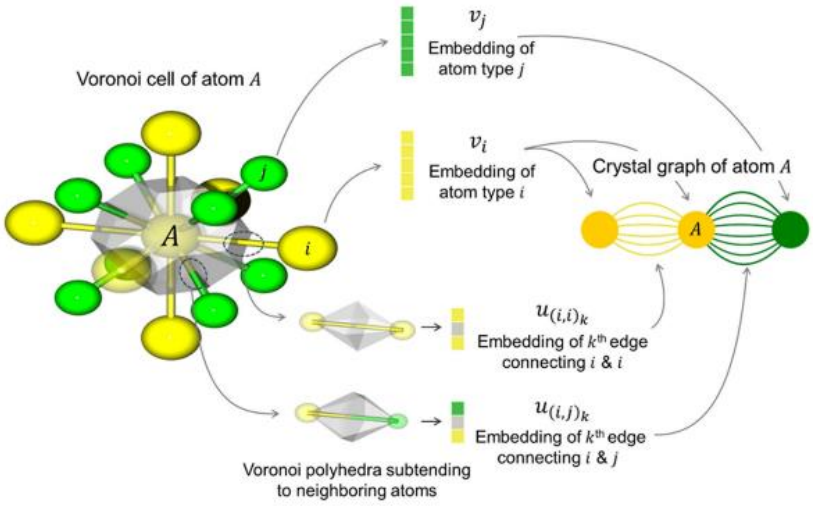

Figure 2. *Illustration of the iCGCNN. From [38].*

The first limitation of CGCNN is that each node within any crystal graph connects to its 12 nearest neighbors, regardless of the crystal's structure. This approach may not accurately reflect an atom's local chemical environment, which is significantly influenced by its closest neighbors, particularly the first and second nearest. The second limitation is that CGCNN encodes only pairwise interactions in the convolutional layers, neglecting higher-order correlations like three-body interactions. Lastly, the model's representation of chemical bonds using edge vectors lacks optimization.

To overcome these limitations, the iCGCNN model enhances the local environment representation by linking each node to its Voronoi neighbors and incorporating the following term into Equation (4):

$$\sum_{j,l,k,k'} \sigma(z'^{(t)}_{(i,j,l)_{k,k'}} W_1'^{(t)} + b_1'^{(t)}) \odot g((z'^{(t)}_{(i,j,l)_{k,k'}} W_2'^{(t)} + b_2'^{(t)})) \tag{6}$$

Here, $z'^{(t)}_{(i,j,l)_{k,k'}} = v_i^{(t)} \oplus v_j^{(t)} \oplus v_l^{(t)} \oplus u^{(t)}_{(i,j)_k} \oplus u^{(t)}_{(i,j)_{k'}}$. Unlike the original CGCNN, where node vectors undergo iterative optimization while edge vectors remain static during training, the iCGCNN updates edge vectors to better represent chemical bonding. The updated convolutional function for edge vectors is illustrated by Equation (7):

$$\begin{aligned} u^{(t+1)}_{(i,j)_k} &= u^{(t)}_{(i,j)_k} + \sigma(z^{(t)}_{(i,j)_k} W_1^{(t)} + b_1^{(t)}) \\ &\odot g(z^{(t)}_{(i,j)_k} W_2^{(t)} + b_2^{(t)}) \\ &+ \sum_{l,k'} \sigma(z'^{(t)}_{(i,j,l)_{k,k'}} W_1'^{(t)} + b_1'^{(t)}) \\ &\odot g((z'^{(t)}_{(i,j,l)_{k,k'}} W_2'^{(t)} + b_2'^{(t)})) \end{aligned} \tag{7}$$

This equation highlights how the chemical attributes of atoms i and j impact their bond, with the summation term indicating the influence of neighboring atoms on this interaction. By incorporating the Voronoi tessellation of crystal structures, explicit three-body correlations among neighboring atoms, and optimized representations of chemical bonds in the crystal graph, the iCGCNN model surpasses the performance of the original CGCNN.

The authors conducted a comparative analysis of the original and improved CGCNN models by utilizing the Open Quantum Materials Database (OQMD) data to assess the accuracy of thermodynamic stability predictions. This analysis involved two methodologies: (1) predicting the formation energy and then determining the hull distance in relation to the convex hull constructed from OQMD data; (2) directly pre-

dicting the hull distance, thereby eliminating the need to calculate convex hull energies. In this context, thermodynamic stability is defined as the disparity between a compound's formation energy and the lowest-energy linear combination of phases for that composition, which is often derived from convex hull constructions. The comparative results revealed notable disparities in predictive accuracy between the two models. Specifically, the improved CGCNN demonstrated a 25% enhancement in predictive accuracy over the original CGCNN for the first method and a 20% improvement for the second method, as measured by the Mean Absolute Error (MAE) across the entire test dataset. Moreover, when applied to the prediction of new compounds with the $ThCr_2Si_2$ structure, the iCGCNN not only identified twice the number of stable compounds compared to the original CGCNN but also exhibited a success rate that was 2.4 times higher. This superior performance in identifying stable compounds underscores iCGCNN's potential to significantly expedite the discovery of new materials, highlighting its value beyond the mere comparative analysis with the original CGCNN model.

3.3. Orbital Graph Convolutional Neural Network

Karamad *et al.*[39] introduced the Orbital Graph Convolutional Neural Network (OGCNN), a novel approach that integrates atomic orbital interactions into the graph convolutional framework, as depicted in Figure 3. This model enhances the representation of atomic interactions by accounting for the specific orbital characteristics of each atom, offering a more nuanced understanding of material properties and behaviors.

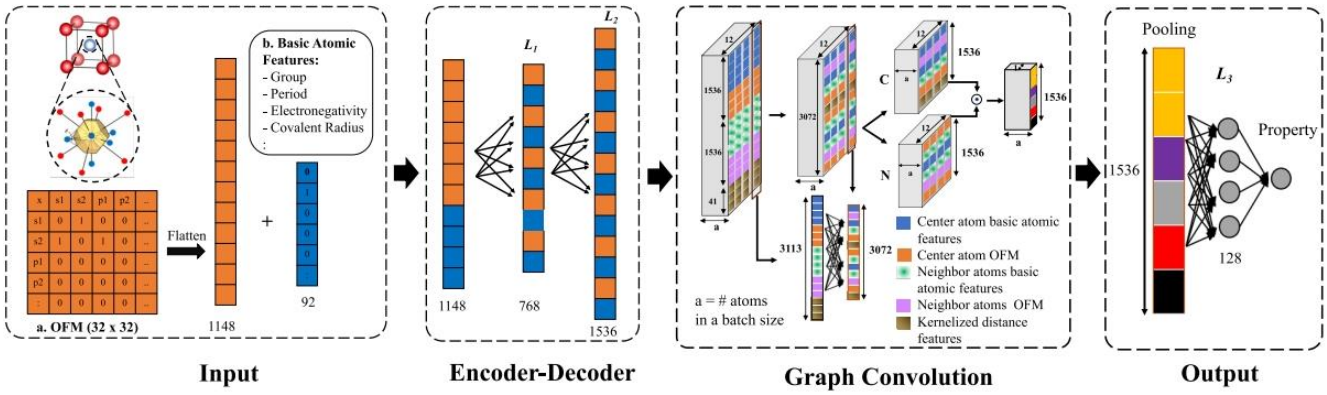

Figure 3. *Schematic diagram of OGCNN framework. From [39].*

The following text discusses the Orbital Graph Convolutional Neural Network (OGCNN) proposed by Karamad *et al.* and its advantages over the traditional CGCNN for encoding local chemical environments of atoms. OGCNN incorporates orbit-orbit interactions leading to more accurate representations of the atomic environment. This is achieved by using an orbital field matrix (OFM), which captures atomic orbital interactions based on the distribution of valence electrons. The OFM approach represents the electronic configuration of an atom as a one-dimensional binary vector, and local atomic structure is represented by a matrix. The matrix is the sum of weighted vector representations of neighboring atoms.

OGCNN has demonstrated its effectiveness in capturing complex interactions through its superior performance on property prediction across five datasets. It consistently outperforms CGCNN, particularly on the lanthanide dataset, where OGCNN achieved a Mean Absolute Error (MAE) of 0.061 eV/atom, significantly lower than CGCNN's MAE of 0.133 eV/atom, indicating a 54% improvement in predictive accuracy. Across other datasets, OGCNN's performance exceeded that of CGCNN by 25% to 50%, underscoring its enhanced ability to predict material properties by effectively integrating atomic orbital interactions.

3.4. Transfer Learning for Materials Informatics Using Crystal Graph Convolutional Neural Network

Lee *et al.* [40] addressed the challenge of data scarcity in materials science machine learning applications by proposing a transfer learning approach using Crystal Graph Convolutional Neural Networks (TL-CGCNN). This methodology leverages the knowledge acquired from one domain and applies it to another, thus mitigating the issue of insufficient training data in the target domain. Transfer learning is particularly beneficial in fields like materials science, where generating large datasets can be expensive or impractical. In their work, as illustrated in Figure 4, they compare the conventional machine learning approach with the transfer learning approach based on convolutional neural networks. Traditional machine learning models require extensive data from the domain of interest to learn effectively and make accurate predictions. However, when such data are not available, these models can underperform due to overfitting or lack of gener-

alizability. In contrast, transfer learning, as applied through TL-CGCNN, allows the model to leverage pre-learned patterns and knowledge from a related domain where abundant data exist. This pre-training process on a large dataset helps the model develop a robust understanding of the underlying patterns and features, which can then be fine-tuned with a smaller dataset from the target domain, enhancing its predictive performance and generalizability despite the data scarcity in the new context.

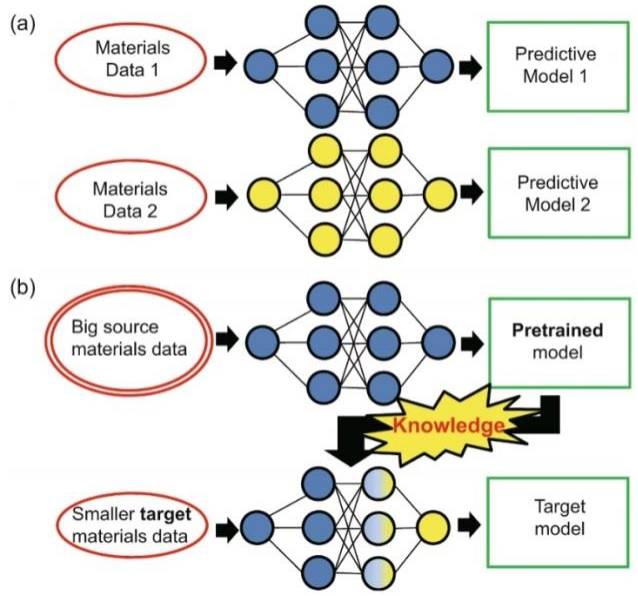

Figure 4. *Comparative schematic diagram of convolutional ML and TL concepts based on convolution neural network. From [40].*

In traditional machine learning methods, creating separate models for each target variable is common practice, which can be limiting when data is scarce for certain variables. However, transfer learning has emerged as a promising solution to this challenge, especially in the context of CNNs. By utilizing a "pre-trained" model that has been developed using a large and diverse dataset, transfer learning enables the extraction and transfer of learned features (weights and biases) in the convolutional layers to a new target model that may have limited data available. The TL-CGCNN approach applies this principle by using a pre-trained model to capture intricate features from a large dataset (source domain), and then transferring this knowledge to a target domain where the model is fine-tuned with a smaller dataset to predict specific material properties. This process is particularly helpful for material science applications where data may be limited or expensive to obtain. Several studies have demonstrated the effectiveness of TL-CGCNN, which consistently achieves lower MAE in predictions compared to conventional CGCNN models. This improvement is attributed to the nuanced feature recognition and generalization capabilities inherited from the pre-trained model. Moreover, it has been quantitatively shown that the size of the dataset used for the pre-trained model directly influences the effectiveness of knowledge transfer; larger pre-trained datasets typically result in better performance on the target task. By addressing the issue of data scarcity, TL-CGCNN enhances the predictive accuracy of models in material science, and widens the scope for applying machine learning to a broader range of material properties, even those with limited data available. This approach holds significant potential for accelerating discoveries and innovations in materials science by making efficient use of available data and learned knowledge across different domains.

3.5. Atomistic Line Graph Neural Network

Graph Neural Networks have proven to be a significant advancement in the field of materials science, particularly in modeling atomic material representations, outperforming traditional descriptor-based machine learning models. The strength of GNNs lies in their ability to capture complex structural and relational information within materials at the atomic level. However, a standard limitation among many GNNs is that they rely solely on atomic distance information. Although distances are essential for understanding material structure, the absence of explicit bond angle information can be a critical gap. Bond angles provide critical geometric information that is necessary to accurately distinguish between different atomic structures, which, in turn, can significantly impact the material's properties.

To address this limitation, Choudhary *et al.*[22] introduced the Atomistic Line Graph Neural Network (ALIGNN). This innovative approach extends beyond the conventional atomic graph by incorporating a line graph representation. In the context of the SiO_4 polyhedron, as illustrated in Figure 5, the crystal graph captures the connectivity between atoms. In contrast, the corresponding line graph further elucidates the interactions between bonds, effectively encoding bond angle information. The line graph is constructed by transforming the bonds (edges in the atomic graph) into nodes in the line graph, allowing the model to capture the angular relationships between atoms by examining the connections between these new nodes. This additional layer of information enriches the model's understanding of the material's atomic structure, enabling more nuanced and accurate predictions of material properties. The ALIGNN model, which integrates both atomic and line graph representations, provides a comprehensive and detailed view of the material's structure, offering significant improvements in predictive performance for a wide range of material properties. This approach not only demonstrates the potential of GNNs in materials science but also highlights the importance of incorporating comprehensive structural information, including bond angles, for a deeper understanding and prediction of material behaviors.

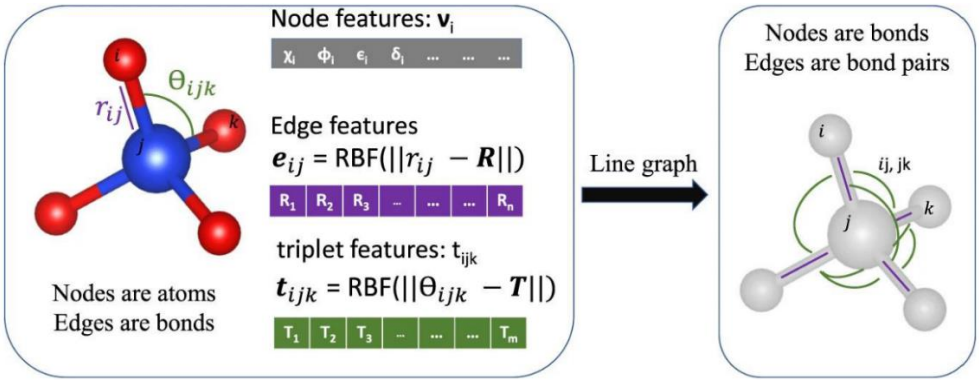

Figure 5. *Illustration of the undirected crystal graph representation and the corresponding line graph construction for a SiO4 polyhedron. From [22].*

In the line graph, each node is representative of an edge from the original atomic graph; both elements symbolize the bonds connecting atoms and maintain a shared foundational representation. Edges within the line graph are associated with atomic triplets or dyads of interatomic bonds.

ALIGNN employs a method called edge-gated graph convolution for the iterative refinement of node and edge features. This method is similar to that used in CGCNN, but with a notable difference: edge features are exclusively integrated into the normalized edge gate. Additionally, pre-aggregated edge messages are used in this convolution process to refine edge representations. The node representation h^I at the I th layer is updated using the following equations:

$$h_i^{I+1} = f\left(h_i^I, \left\{h_j^I\right\} j \in Ni\right) \tag{8}$$

$$h_i^{I+1} = h_i^I + \mathrm{SiLU}\left(\mathrm{Norm}\left(W_{\mathrm{src}}^I h_i^I + \sum_{j \in N_i} \hat{e}_{ij}^I W_{\mathrm{dst}}^I h_j^I\right)\right) \tag{9}$$

$$\hat{e}_{ij}^I = \frac{\sigma(e_{ij}^I)}{\sum k \in N_i + \epsilon} \tag{10}$$

$$e_{ij}^I = e_{ij}^{I-1} + \mathrm{SiLU}\left(\mathrm{Norm}\left(A^I h_i^{I-1} + B^I h_j^{I-1} + C^I e_{ij}^{I-1}\right)\right) \tag{11}$$

The edge messages described in Eq. (11) parallel the gating mechanism in CGCNN's update strategy, with the matrices A, B, and C amalgamated into W_{gate}. The enhanced edge representation is defined as follows:

$$z_{ij} = h_i \oplus h_j \oplus e_{ij} \tag{12}$$

$$e_{ij}^I = e_{ij}^{I-1} + \mathrm{SiLU}\left(\mathrm{Norm}\left(W_{\mathrm{gate}}^I z_{ij}^{I-1}\right)\right) \tag{13}$$

These updates ensure a nuanced enhancement of feature representations within the graph convolution framework.

ALIGNN executes alternating updates between the two graphs, employing Eq. (14) and Eq. (15), as illustrated in Figure 6. This process facilitates the transfer of bond angle information into atomic representations by moving from interatomic bond representations to atom-wise representations and conversely. The integration of bond distances and angles from the line graph into the model captures intricate details of atomic structures, which significantly enhances the model's performance. The detailed incorporation of atomic structure nuances and advanced convolutional update mechanisms has enabled ALIGNN to achieve a leading mean absolute error (MAE) of 0.022 eV/atom for the formation energy of solids at 0 K.

$$m^I, t^I = \text{EDGE Gated Graph Conv}(L(g), e^{I-1}, t^{I-1}) \tag{14}$$

$$h^I, e^I = \text{EDGE Gated Graph Conv}(g, h^{I-1}, m^I) \tag{15}$$

These equations delineate the method by which ALIGNN iteratively refines the features across both the atomic and line graphs, ensuring a robust and detailed representation of the material's structure.

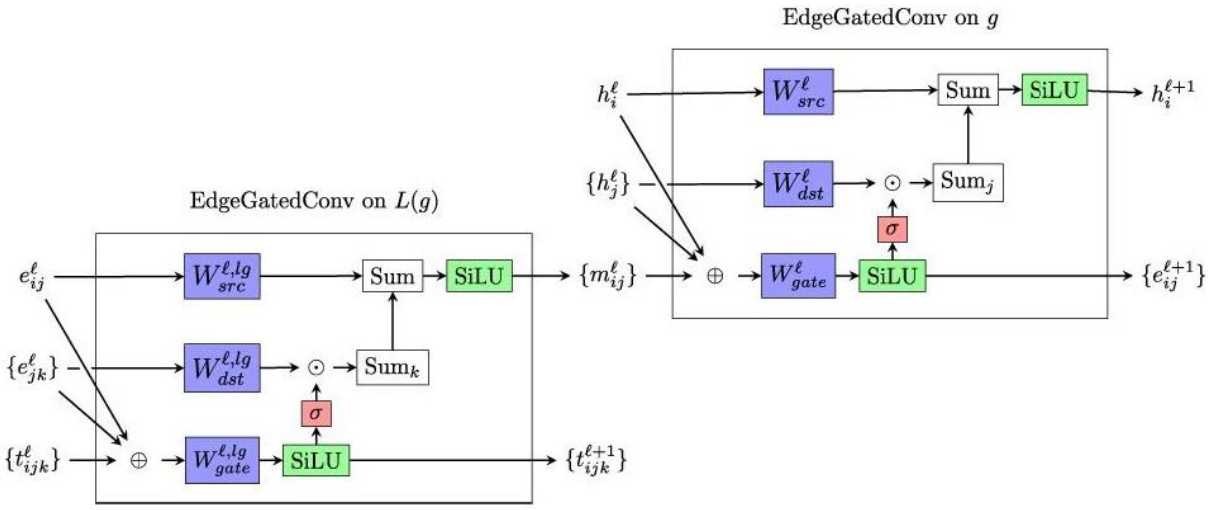

Figure 6. *Schematic of the ALIGNN layer structure. From [22].*

3.6. Self-supervised Learning Graph Neural Network

Supervised learning algorithms such as Graph Neural Networks are known for their effectiveness in predicting material properties. However, training precise machine learning models requires large labeled datasets, which can be expensive in terms of computational resources. Self-Supervised Learning (SSL) methodologies have recently emerged as a solution to this challenge by enabling the training of machine learning models using unlabeled data. This approach has shown remarkable achievements in computer vision and natural language processing, as demonstrated by Chen *et al.* and Lan *et al.* [41, 42]. In this context, Magar *et al.* [43] introduced the Crystal twins (CT) framework, which is an SSL approach designed for predicting the properties of crystal materials using GNN. The CT framework's pre-training phase uses a self-supervised strategy, allowing the model to acquire crystal representations without relying on labeled data. This approach streamlines the model training process, making it more efficient and cost-effective.

Within the CT framework, the CGCNN is responsible for encoding the crystal systems and learning their representations. Through comparison with other leading supervised learning models, the effectiveness of using self-supervised learning for predicting crystal material properties has been convincingly demonstrated. This analysis highlights the potential of self-supervised learning to improve predictive accuracy while reducing the reliance on large labeled datasets.

It's important to note that the CT framework outperforms ALIGNN only in classification tasks. This is because ALIGNN is specifically designed to model three-body interactions, which gives it a high level of accuracy. On the other hand, the CT framework uses CGCNN as its encoder, which focuses primarily on two-body interactions. Because of the explicit modeling of three-body interactions, ALIGNN is more robust and accurate than CGCNN. As a result, when the CT framework uses CGCNN, it is limited in its ability to achieve the same level of performance as ALIGNN in tasks that require three-body interactions for predictive accuracy.

Researchers in the field of self-supervised learning for graph neural networks, such as Kong *et al.* [44,45], have shown that GNNs heavily rely on end-to-end learning from extensive material datasets. This approach often results in a limited understanding of the multiscale material information due to the absence of prior knowledge. Moreover, the data labeling process is labor-intensive, error-prone, and consumes substantial resources, which can limit the accuracy of subsequent predictions. To address these challenges, Kong *et al.* advocate for the adoption of node and edge information from crystal graphs within a self-supervised learning framework to pre-train GNN models. This method fosters the generation of self-supervised atomic representations, which outperform traditional, manually crafted material descriptors in terms of predictive capabilities. These representations allow for adjustable information scopes and offer enhanced insights into material properties. The application of self-supervised atomic representations on a magnetic moment dataset illustrates their potential in pattern and information extraction from magnetic materials. To encapsulate rich physical information within the GNN model, they introduced the Node Embedding Graph Neural Network (NEGNN) framework, depicted in Figure 8, which demonstrated a marked improvement in predictive accuracy.

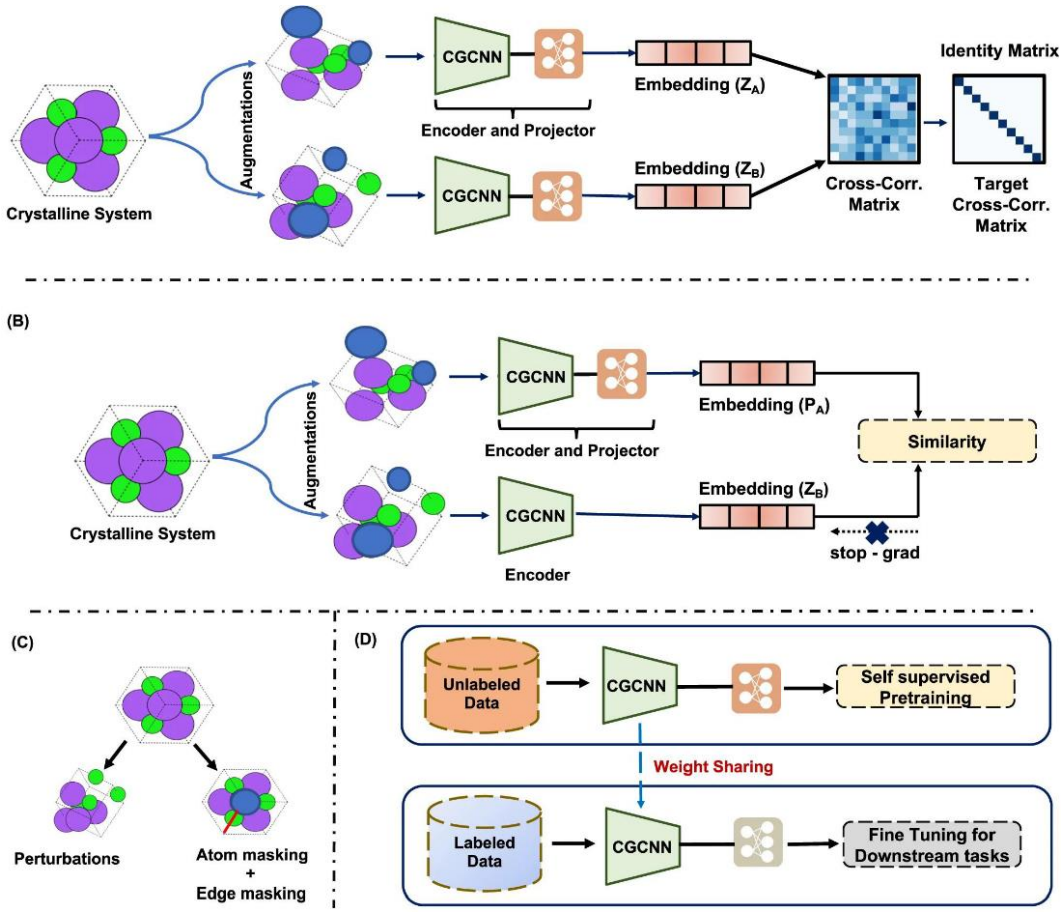

Figure 7. *Overview of the Crystal twins framework. From [43].*

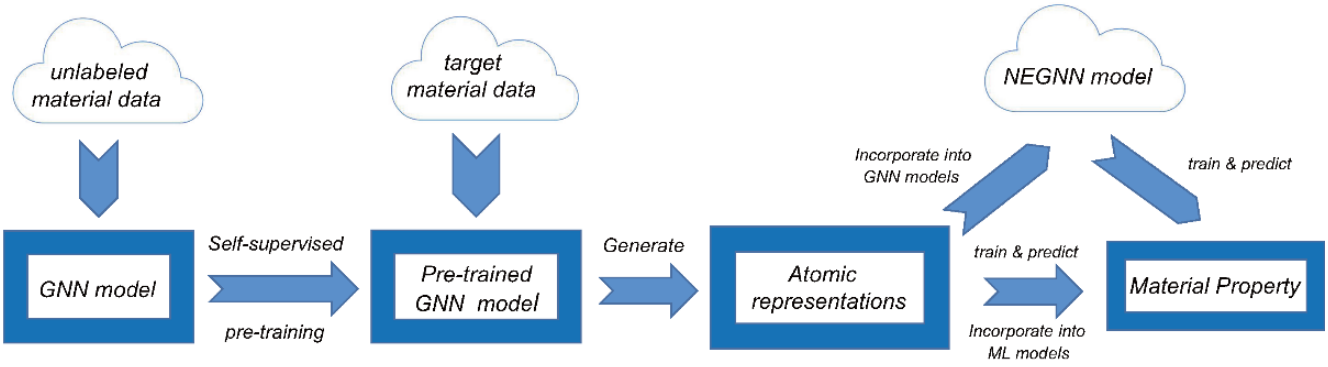

Figure 8. *Illustration of self-supervised training and the NEGNN framework. From [45].*

The combination of the self-supervised material representation and the NEGNN framework allows for a more thorough analysis of material information, making it particularly effective for smaller datasets and improving prediction accuracy.

4. Limitations and Challenges

GCNs, as a subset of DL methodologies, face several prevalent challenges. These include the reliance on annotated data, the lack of ground truth in non-simulated datasets, marked discrepancies between training data and real-world data distributions, and difficulties in benchmarking and interpreting results [46, 47]. Additionally, the "black box" nature of these models [19] complicates the extraction and comprehension of high-level physics or materials information. Research aimed at model interpretability is crucial for mitigating these issues, thereby improving GCN model performance in specific applications.

While predictions based on atomic graphs offer a comprehensive atomic-level description, their applicability has so far been limited to bulk materials, excluding defective sys-

tems. This limitation underscores the necessity for input features to be predictive of output labels and not to omit crucial information. Although GCNs have shown significant accuracy enhancements over previous models, there remains a need to reduce further model errors to achieve "chemical accuracy" akin to what is expected in deep learning. Moreover, the GCN architecture needs to scale beyond a few layers of depth [48].

Looking ahead, the integration of GCNs with the latest model technologies to overcome GCN-specific challenges, characterize material structural information, and boost model performance represents an exciting new research avenue. The evaluation and improvement of DL model interpretability continue to be significant areas of focus in the field.

Abbreviations

DFT: Density Functional Theory
ML: Machine Learning
DL: Deep Learning
GNN: Graph Neural Network
GCNN: Graph Convolutional Neural Network
ANN: Artificial Neural Network
tanh: Hyperbolic Tangent
ReLU: Rectified Linear Unit
SGD: Stochastic Gradient Descent
BCE: Binary Cross-Entropy
NLLL: Negative Log-Likelihood
MAE: Mean Absolute Error
MLP: Multilayer Perceptron
GCN: Graph Convolutional Network
GAT: Graph Attention Network
Relational-GCN: Relational Graph Convolutional Network
GRN: Graph Recurrent Network
GIN: Graph Isomorphism Network
LGNN: Line Graph Neural Network
CNN: Convolutional Neural Network
CGCNN: Crystal Graph Convolutional Neural Network
OQMD: Open Quantum Materials Database
iCGCNN: Improved Crystal Graph Convolutional Neural Network
OGCNN: Orbital Crystal Graph Convolutional Neural Network
OFM: Orbital Field Matrix
TL-CGCNN: Transfer Learning Approach Using Crystal graph Convolutional Neural Networks
ALIGNN: Atomistic Line Graph Neural Network
SSL: Self-Supervised learning
CT: Crystal Twins
NEGNN: Node Embedding Graph Neural Network

Author Contributions

Ke-Lin Zhao: Conceptualization, Investigation, Writing – original draft

Qing-Xu Li: Conceptualization, Methodology, Supervision, Writing – review & editing

Funding

This work has been supported by the research foundation of Institute for Advanced Sciences of CQUPT (Grant No. E011A2022328).

Data Availability Statement

Not applicable.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Nantasenamat, C., Isarankura-Na-Ayudhya, C., Naenna, T., Prachayasittikul, V. A Practical Overview of Quantitative Structure-activity Relationship. *EXCLI Journal*. 2009, 8, 74-88. https://doi.org/10.17877/DE290R-690

[2] Hohenberg, P., Kohn, W. Inhomogeneous Electron Gas. *Physical Review*. 1964, 136(3B): B864. https://doi.org/10.1103/PhysRev.136.B864

[3] Kohn, W., Sham, L. J. Self-consistent Equations Including Exchange and Correlation Effects. *Physical Review*. 1965, 140(4A): A1133. https://doi.org/10.1103/PhysRev.140.A1133

[4] Alder, B. J., Wainwright, T. E. Studies in Molecular Dynamics. I. General Method. *The Journal of Chemical Physics*. 1959, 31(2): 459-466. https://doi.org/10.1063/1.1730376

[5] Rahman, A. Correlations in the Motion of Atoms in Liquid Argon. *Physical Review*. 1964, 136(2A): A405. https://doi.org/10.1103/PhysRev.136.A405

[6] Acioli, P. H. Review of Quantum Monte Carlo Methods and Their Applications. *Journal of Molecular Structure: THEOCHEM*. 1997, 394(2-3): 75-85. https://doi.org/10.1016/S0166-1280(96)04821-X

[7] Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., Cholia, S., Gunter, D., Skinner, D., Ceder, G., Persson, K. A. Commentary: The Materials Project: A materials Genome Approach to Accelerating Materials Innovation. *APL Materials*. 2013, 1(1). https://doi.org/10.1063/1.4812323

[8] Belsky, A., Hellenbrandt, M., Karen, V. L., Luksch, P. New Developments in the Inorganic Crystal Structure Database (ICSD): Accessibility in Support of Materials Research and Design. *Acta Crystallographica Section B: Structural Science*. 2002, 58(3): 364-369. https://doi.org/10.1107/S0108768102006948

[9] Zhang, T., Jiang, Y., Song, Z., Huang, H., He, Y., Fang, Z., Weng, H., Fang, C. Catalogue of Topological Electronic Materials. *Nature*. 2019, 566(7745): 475-479. https://doi.org/10.1038/s41586-019-0944-6

[10] Seko, A., Togo, A., Hayashi, H., Tsuda, K., Chaput, L., Tanaka, I. Prediction of Low-thermal-conductivity Compounds with First-principles Anharmonic Lattice-dynamics Calculations and Bayesian Optimization. *Physical Review Letters*. 2015, 115(20): 205901. https://doi.org/10.1103/PhysRevLett.115.205901

[11] Faber, F. A., Lindmaa, A., Von Lilienfeld, O. A., Armiento, R. Machine Learning Energies of 2 Million Elpasolite (ABC_2D_6) Crystals. *Physical Review Letters*. 2016, 117(13): 135502. https://doi.org/10.1103/PhysRevLett.117.135502

[12] Xue, D., Balachandran, P. V., Hogden, J., Theiler, J., Xue, D., Lookman, T. Accelerated Search for Materials with Targeted Properties by Adaptive Design. *Nature Communications*. 2016, 7(1): 1-9. https://doi.org/10.1038/ncomms11241

[13] LeCun, Y., Bengio, Y., Hinton, G. Deep Learning. *Nature*. 2015, 521(7553): 436-444. https://doi.org/10.1038/nature14539

[14] McCulloch, W. S., Pitts, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics*. 1943, 5: 115-133. https://doi.org/10.1007/BF02478259

[15] Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*. 1958, 65(6): 386. https://doi.org/10.1037/h0042519

[16] Chen, C., Ye, W., Zuo, Y., Zheng, C., Ong, S. P. Graph Networks as A Universal Machine Learning Framework for Molecules and Crystals. *Chemistry of Materials*. 2019, 31(9): 3564-3572. https://doi.org/10.1021/acs.chemmater.9b01294

[17] Choubisa, H., Todorović, P., Pina, J. M., Parmar, D. H., Li, Z., Voznyy, O., Tamblyn, I., Sargent, E. H. Interpretable Discovery of Semiconductors with Machine Learning. *npj Computational Materials*. Year, Volume (Issue), 2023, 9(1): 117. https://doi.org/10.1038/s41524-023-01066-9

[18] Chen, C., Zuo, Y., Ye, W., Li, X., Ong, S. P. Learning Properties of Ordered and Disordered Materials from Multi-fidelity Data. *Nature Computational Science*. 2021, 1(1): 46-53. https://doi.org/10.1038/s43588-020-00002-x

[19] Holm, E. A. In Defense of the Black Box. Science. *Science*. 2019, 364(6435): 26-27. https://doi.org/10.1126/science.aax0162

[20] Xie, T., Grossman, J. C. Crystal Graph Convolutional Neural Networks for An Accurate and Interpretable Prediction of Material Properties. *Physical Review Letters*. 2018, 120(14): 145301. https://doi.org/10.1103/PhysRevLett.120.145301

[21] Schütt, K. T., Kessel, P., Gastegger, M., Nicoli, K. A., Tkatchenko, A., Muller, K. R. SchNetPack: A Deep Learning Toolbox for Atomistic Systems. *Journal of Chemical Theory and Computation*. 2018, 15(1): 448-455. https://doi.org/10.1021/acs.jctc.8b00908

[22] Choudhary, K., DeCost, B. Atomistic Line graph Neural Network for Improved Materials Property Predictions. *npj Computational Materials*. 2021, 7(1): 185. https://doi.org/10.1038/s41524-021-00650-1

[23] Nwankpa, C., Ijomah, W., Gachagan, A., Marshall, S. Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. *Arxiv Preprint*. 2018, arxiv: 1811.03378. https://doi.org/10.48550/arXiv.1811.03378

[24] Rumelhart, D. E., Hinton, G. E., Williams, R. J. Learning Representations by Back-propagating Errors. *Nature*. 1986, 323(6088): 533-536. https://doi.org/10.1038/323533a0

[25] Kipf, T. N., Welling, M. Semi-supervised Classification with Graph Convolutional Networks. *Arxiv Preprint*. 2016, arxiv: 1609.02907. https://doi.org/10.48550/arXiv.1609.02907

[26] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y. Graph Attention Networks. *Arxiv Preprint*. 2017, arxiv: 1710.10903. https://doi.org/10.48550/arXiv.1710.10903

[27] Vashishth, S., Sanyal, S., Nitin, V., Talukdar, P. Composition-based Multi-relational Graph Convolutional Networks. *Arxiv Preprint*. 2019, arXiv: 1911.03082. https://doi.org/10.48550/arXiv.1911.03082

[28] Hajiramezanali, E., Hasanzadeh, A., Narayanan, K., Duffield, N., Zhou, M., Qian, X. Variational Graph Recurrent Neural Networks. *Advances in Neural Information Processing Systems*. 2019, 32.

[29] Xu, K., Hu, W., Leskovec, J., Jegelka, S. How Powerful Are Graph Neural Networks?. *Arxiv Preprint*. 2018, arXiv: 1810.00826. https://doi.org/10.48550/arXiv.1810.00826

[30] Chen, Z., Li, X., Bruna, J. Supervised Community Detection with Line Graph Neural Networks. *Arxiv Preprint*. 2017, arXiv: 1705.08415. https://doi.org/10.48550/arXiv.1705.08415

[31] Krizhevsky, A., Sutskever, I., Hinton, G. E. Imagenet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*. 2012, 25.

[32] Krizhevsky, A., Sutskever, I., Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*. 2017, 60(6): 84-90. https://doi.org/10.1145/3065386

[33] Isayev, O., Oses, C., Toher, C., Gossett, E., Curtarolo, S., Tropsha, A. Universal Fragment Descriptors for Predicting Properties of Inorganic Crystals. *Nature Communications*. 2017, 8(1): 15679. https://doi.org/10.1038/ncomms15679

[34] Ghiringhelli, L. M., Vybiral, J., Levchenko, S. V., Draxl, C., Scheffler, M. Big Data of Materials Science: Critical Role of the Descriptor. *Physical Review Letters*. 2015, 114(10): 105503. https://doi.org/10.1103/PhysRevLett.114.105503

[35] Pham, T. L., Kino, H., Terakura, K., Miyake, T., Tsuda, K., Takigawa, I., Dam, H. C. Machine Learning Reveals Orbital Interaction in Materials. *Science and Technology of Advanced Materials*. 2017, 18(1): 756. https://doi.org/10.1080/14686996.2017.1378060

[36] Schütt, K. T., Glawe, H., Brockherde, F., Sanna, A., Müller, K. R., Gross, E. K. How to Represent Crystal Structures for Machine Learning: Towards Fast Prediction of Electronic Properties. *Physical Review B*. 2014, 89(20): 205118. https://doi.org/10.1103/PhysRevB.89.205118

[37] Seko, A., Hayashi, H., Nakayama, K., Takahashi, A., Tanaka, I. Representation of Compounds for Machine-learning Prediction of Physical Properties. *Physical Review B*. 2017, 95(14): 144110. https://doi.org/10.1103/PhysRevB.95.144110

[38] Park, C. W., Wolverton, C. Developing An Improved Crystal Graph Convolutional Neural Network Framework for Accelerated Materials Discovery. *Physical Review Materials*. 2020, 4(6): 063801. https://doi.org/10.1103/PhysRevMaterials.4.063801

[39] Karamad, M., Magar, R., Shi, Y., Siahrostami, S., Gates, I. D., Farimani, A. B. Orbital Graph Convolutional Neural Network for Material Property Prediction. *Physical Review Materials*. 2020, 4(9): 093801. https://doi.org/10.1103/PhysRevMaterials.4.093801

[40] Lee, J., Asahi, R. Transfer Learning for Materials Informatics Using Crystal Graph Convolutional Neural Network. *Computational Materials Science*. 2021, 190: 110314. https://doi.org/10.1016/j.commatsci.2021.110314

[41] Chen, T., Kornblith, S., Norouzi, M., Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 2020 International Conference on Machine Learning*, Baltimore, USA, 2020; pp. 1597-1607.

[42] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. Albert: A Lite Bert for Self-supervised Learning of Language Representations. *Arxiv Preprint*. 2019, arXiv: 1909.11942. https://doi.org/10.48550/arXiv.1909.11942

[43] Magar, R., Wang, Y., Barati Farimani, A. Crystal Twins: Self-supervised Learning for Crystalline Material Property Prediction. *npj Computational Materials*. 2022, 8(1): 231. https://doi.org/10.1038/s41524-022-00921-5

[44] Kong, J. G., Li, Q. X., Li, J., Liu, Y., Zhu, J. J. Self-supervised graph neural networks for accurate prediction of N{\'e}el temperature. *Chinese Physics Letters*. 2022, 39(6): 067503. https://doi.org/10.1088/0256-307X/39/6/067503

[45] Kong, J. G., Li, J., Li, Q. X., Liu, Y., Zhang, R., Zhu, J. J., Chang, K. Self-supervised Representations and Node Embedding Graph Neural Networks for Accurate and Multi-scale Analysis of Materials. *Arxiv Preprint*. 2022, arXiv: 2211.03563. https://doi.org/10.48550/arXiv.2211.03563

[46] Sapoval, N., Aghazadeh, A., Nute, M. G., Antunes, D. A., Balaji, A., Baraniuk, R.,... Treangen, T. J. Current Progress and Open Challenges for Applying Deep Learning Across the Biosciences. *Nature Communications*. 2022, 13(1): 1728. https://doi.org/10.1038/s41467-022-29268-7

[47] Chen, X. W., Lin, X. Big Data Deep Learning: Challenges and Perspectives. *IEEE Access*. 2014, 2: 514-525. https://doi.org/10.1109/ACCESS.2014.2325029

[48] Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K. I., Jegelka, S. Representation Learning on Graphs with Jumping Knowledge Networks. In *Proceedings of the 2018 International Conference on Machine Learning*, Jinan, China, 2018; pp. 5453-5462.

Biography

Ke-Lin Zhao is a graduate student in physics at Chongqing University of Posts and Telecommunications. He received his B.S. in applied physics from Chongqing University of Posts and Telecommunications in 2021. His main research interests are focused on machine learning study for prediction on material properties.

Qing-Xu Li is a professor in physics at Chongqing University of Posts and Telecommunications. He received his Ph.D. in Chemistry from Institute of Chemistry of Chinese Academy of Sciences in 2008. His recent research interests include AI plus sciences, non-hermitian physics.

Research Field

Ke-Lin Zhao: Machine learning in material science

Qing-Xu Li: Artificial intelligence plus sciences, Non-hermitian physics, Theoretic and computational chemistry, Computational physics, Finite-element method and its applications