Transfer Learning and Representations for Material Properties

Simon Deng **Jerry Park** **Aaron Sy***
University of California, Berkeley
{simond, jjhpark, raaronsy}@berkeley.edu

Abstract

Crystal Graph Convolutional Neural Networks (CGCNN) are used to predict crystal compound properties given only atomic information and crystal geometry. To assess the generalizability of the pre-trained networks, we apply transfer learning on CGCNN onto various crystal properties. In particular, we are able to find a more generalizable network than other previous pre-trained networks by training on a combination of target properties. We also develop a CGCNN architecture with attention mechanism, which uses less parameters and computation.

1 Introduction

Creation of new materials and measurement of their material properties is an active area of research. For a long time now, materials discovery has had been driven by an experimental trial-and-error approach. However, experiments of new materials can be prohibitively costly and time-consuming. Hence, we look for computational models that simulate material properties based on crystal structure with high accuracy. While different models can be created to predict different material properties, it would be useful to have a more generalized model. In particular, such a model must be able to extract fundamental features that can be used to predict a variety of properties from any new crystal structure. Additionally, such a model could be investigated through visualization and attention, among other techniques, to elucidate useful or intriguing fundamental properties of crystal graphs and CGCNNs.

1.1 Problem Statement

An open problem in material informatics is how one should represent a crystal, which is a three-dimensional structure of atoms. Being able to model various interactions between atoms as a set of features will help in material properties prediction tasks. Based on known atomic properties of atoms, and the geometry of a crystal unit cell, we wish to predict properties at the crystal compound level, such as band gap, shear modulus, or formation energy. Atomic properties may include the period, group number, the number of valence electrons, electronegativity, and covalent radius, among others.

In particular, we explore the ability of generalizability of extracted crystal features to properties different from the original training properties. The importance of generalizability is shown in the area of computer vision and natural language processing. These days, many machine learning models use off-the-shelf pretrained CNN weights for images and pretrained word embeddings for text. We aim to develop a pretrained CGCNN that can generalize well to different tasks and labels.

*Alphabetical Order

1.2 Data Source

We use crystals from The Open Quantum Materials Database (OQMD), an online, publicly accessible database which is managed by a group at Northwestern University [1]. This database has many known stable crystalline compounds, which allows us to extract chemically valid compositions.

For atomic data and element featurization, we use the labels used in [3]. We also use the method from [3] to convert crystal unit cells into the graph representation.

2 Related Work

2.1 Graph Convolutional Networks

Graph convolutional networks make use of graph convolution layers. In general, graph convolutional layers transform some vector representation of a vertex $v_i^{(l)} \in \mathbb{R}^{F^{(l)}}$ to a new vector $v_i^{(l+1)} \in \mathbb{R}^{F^{(l+1)}}$ as a function of the surrounding neighbor vertices of v_i and the adjacency matrix A. If we have n vertices, we represent all vertices at layer l as the stacked matrix $H \in \mathbb{R}^{N \times F}$ then we can write

$$H^{(l+1)} = f(A, H^{(l)})$$

Often the function f includes trainable weights as in [3]. Graph convolution layers, however, are often combined with other layers, such as nonlinear activations and fully connected layers.

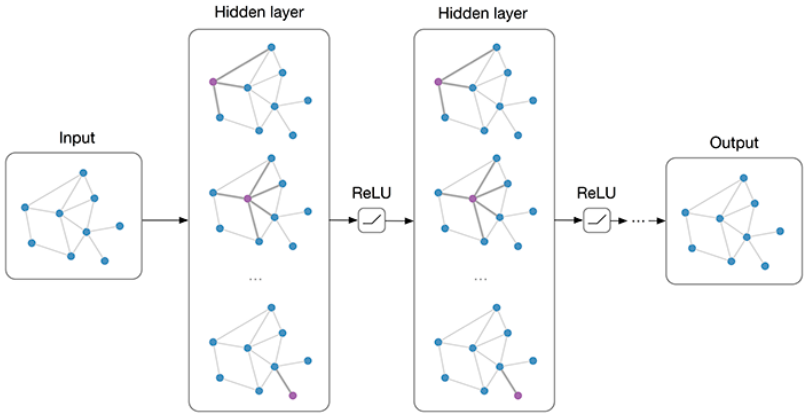

Figure 1: Multi-layer Graph Convolutional Network (GCN) with first-order filters [3]

Note that these new layers are convolutional in the sense that the same function should be applied locally to each vertex v_i and should not be confused with the convolution operations commonly used for images.

2.2 Crystal Graph Convolutional Networks

The architecture we use is that of Xie and Grossman [4]. See figure 1 for an illustration. We use their preprocessing for elemental features (vertices v_i) and bond lengths (edges $u_{(i,j)_k}$) converting crystals into graphs. Note that due to geometry of some crystals, these graphs may have multiple parallel edges hence we refer to the kth edge between vertices i and j.

The graph convolution used is

$$v_i^{(l+1)} = v_i^{(l)} + \sum_{j,k} \sigma(z_{(i,j)_k}^{(l)} W_f^{(l)} + b_f^{(l)}) \odot g(z_{(i,j)_k}^{(l)} W_s^{(l)} + b_s^{(l)})$$

where

$$z_{(i,j)_k}^{(l)} = v_i^{(l)} \oplus v_j^{(j)} \oplus u_{(i,j)_k}$$

is the concatenation of the current vertex i, its neighbor j and their kth edge connection. g is the soft-plus nonlinearity. After R graph convolutions, there are L_1 fully connected layers followed by a pooling layer. In [4], $L_1 = 0$ and the pooling layer simply averages the feature vectors over all vertices to get the crystal feature vector. This is followed by L_2 fully connected layers to output.

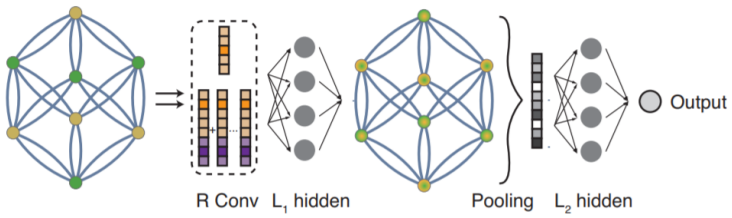

Figure 2: The architecture from [4]. R crystal graph convolution layers are applied, followed by L_1 fully connected layers. Then a pooling layer is followed by L_2 fully connected layers.

3 Data Preprocessing

We use a computational materials library in Python (pymatgen) to extract elemental properties. As some of the compounds within the dataset have elements that are relatively rare, the dataset was trimmed to remove any elements after atomic number 80. The remaining dataset consist of 219635 compounds, which we use for transfer learning.

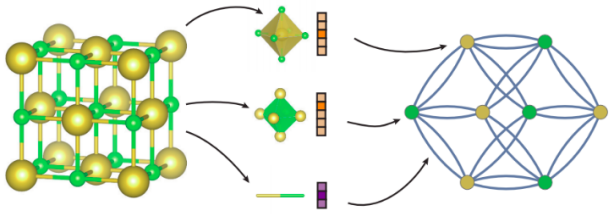

Figure 3: In crystals there are no physical edges connecting the atoms. We only define edges between atoms which have a distance below the set threshold level, because the further the atoms are, the less effect they have on each other.

To obtain the target crystal properties (training labels in our machine learning model), we ran the 219635 compounds through the Materials Project database [2] to find various properties of the crystals. However, the database does not include information for the whole dataset of 219k compounds. After a thorough search and extract process, we had property labels for 12k compounds, though there were even fewer compounds for some properties.

4 Our Approach

Our initial neural architecture to predicting crystal properties was a 3D Convolutional Neural Network by voxelization of the crystal structure. This approach would encode the positions of atoms into a voxel and apply convolutional filters to detect local structures that may affect the target crystal property. However, we were faced with two main problems. First, this particular discretization will lose the precise measurements of the input. Loss of precision can easily lead to a noisy and possibly low-performing final model. Second, in the effort to retain precision of the input data, the size of our 3D CNN would easily blow up. An increase in precision led to a cubic increase in parameters, storage, and computation.

In order to alleviate the problems with voxelization, we turned to Graph Convolutional Networks, which are much more efficient for crystal structures, to find a model that is able to extract fundamental features used for predicting various crystal properties.

4.1 Transfer Learning

We use the PyTorch infrastructure from [4], modifying it for transfer learning. To comply with the CGCNN from [4], the data we use is converted to Crystallographic Information Framework (CIF) format [5], and crystal properties obtained from [2] were converted to CSV.

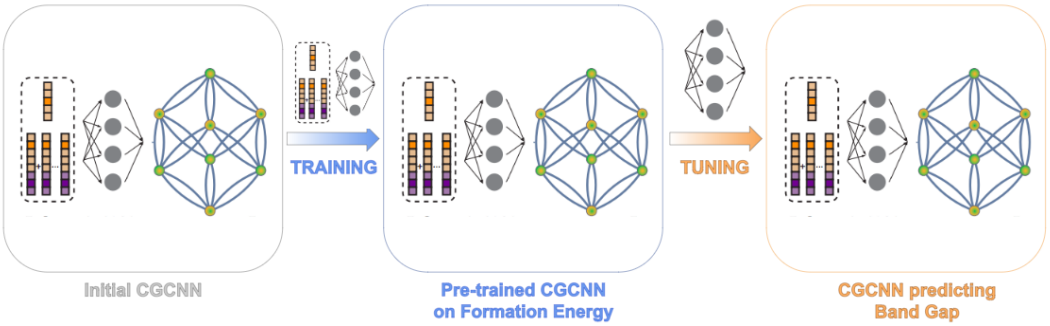

Figure 4: Transfer learning for crystal properties

Our goal is to apply transfer learning to pre-trained CGCNN models, predicting unseen crystal properties. For this work, the property to predict is crystal band gap. As a baseline, we initialize a "fresh" model, which is trained to predict band gap directly. To investigate transferability, we obtain several models pre-trained on different crystal properties. For each pre-trained model, we freeze shallow layers, reinitialize deeper layers, and retrain the models to predict the band gap crystal property. However, we also pre-train a new model to predict multiple properties - elasticity, total energy, and formation energy. One of our hypotheses is that the model trained to predict multiple properties will transfer better than models trained to predict only a single. The loss function across all models is the Mean Squared Error (MSE) loss.

While we keep the overall architecture of each model the same, not every model has the same training parameters. Notably, some models train on more crystal data points than others. Considering this, we perform follow-up experiment within a more controlled setting. Table 1 summarizes the models we use, including training label and training size.

Table 1: All models use 1 embedding layer, 4 convolution layers, 1 pooling layer, and 1 hidden layer.

Pre-train Property	Pre-train data size	Source
Formation Energy	28046	[4]
Bulk Modulus	2041	[4]
Shear Modulus	2041	[4]
Combined	7042	This work

4.2 Graph Convolutions with Attention

For the convolutional layers for CGCNN, [4] states that the $\sigma(\cdot)$ functions as a learned weight matrix to differentiate interactions between neighbors. We notice that the convolutional updates closely resemble that of Recurrent Neural Networks, as each vertex feature vector get updated through each layer. In our approach, we apply an attention mechanism, instead of a sigmoid function, to learn to differentiate interactions between atoms.

Following the same notation as in Section 2.2, graph convolution with attention is as follows.

$$v_i^{(l+1)} = v_i^{(l)} + \sum_{j,k} c_{j,k} \odot g(z^{(l)}_{(i,j)_k} W_s + b_s)$$

where the attention weight applied to each nonlinearity $c_{j,k}$ is

$$c_{j,k} = \frac{\exp\left(\langle v_i, v_j \rangle \cdot \langle u_{(i,j)_k}, u_{(i,j)_k} \rangle\right)}{\sum_{j,k} \exp\left(\langle v_i, v_j \rangle \cdot \langle u_{(i,j)_k}, u_{(i,j)_k} \rangle\right)}$$

The attention mechanism in our network allows us to not only reduce weight parameters and computation time, but also to parallelize multiple networks to create an ensemble of networks that

are combined through fully connected layers at the end. Thus, it allows the network to learn different kinds of attention for various target properties like the Transformer attention model.

5 Experiments and Results

5.1 Experiment Design

We obtain 4 pre-trained GCGNN networks, 3 provided by [3] and pre-trained on individual properties, and 1 pre-trained in this work to predict multiple crystal properties. These properties are crystal elasticity, total energy, and formation energy. All networks have an embedding layer, four graph convolutional layers, one pooling layer and one fully connected layer. To assess generalizability of the network, we measure the effectiveness in transfer learning, so we freeze the embedding layer with either the first 1, 2, or 3 convolutional layers. We then retrain the remaining layers to predict the crystal band gap property. As a baseline, we also train and evaluate a model with the same architecture from scratch for predicting the band gap.

We train each model for 30 epochs on a data set containing 1000 training points, 1000 validation points, and 1000 testing points, chosen randomly from our dataset, with the same partition for all models. Also, we train the attention model with the formation energy label. We use the mean absolute error (MAE) on a held-out test set to assess the quality of a model; this is the same metric as [4]. Our results compare the MAE achieved for the different models over either 1, 2, or 3 transferred convolutional layers. Table 2, Figure 5 and Figure 6 summarize the results.

Table 2: Results for Test MAE. Bolded are results that surpass the baseline.

Model	**Mean Absolute Error**		
Baseline		0.812	
Pre-trained Model	*First Layer*	*First 2 Layers*	*First 3 Layers*
Formation Energy	**0.797**	**0.765**	0.865
Bulk Modulus	**0.776**	0.897	0.956
Shear Modulus	0.846	0.993	0.988
Combined	0.820	0.850	0.870

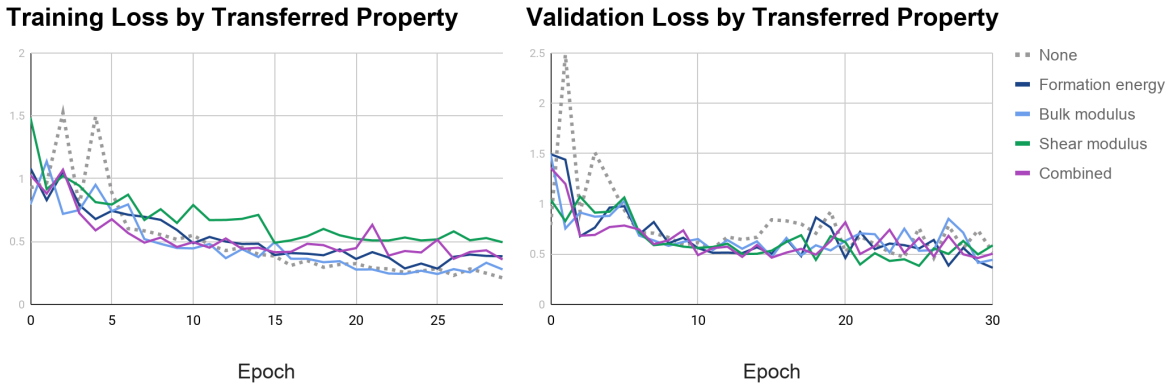

Figure 5: Training and validation loss over time for different models. All models plotted had 2 transferred convolutional layers.

5.2 Discussion

Only two models outperform baseline, Formation Energy and Bulk Modulus, and only when we transfer fewer layers. There is a general trend that, with more layers transferred, the final test MAE increases. This is as expected, since models that transfer fewer layers have more free parameters to

fit the data. This may indicate that models pre-trained to predict those properties transfer the best, though differences in final test MAE may also be attributable to variance. In addition, evidently there is a tradeoff between accuracy and computation time. The more layers we transfer, the quicker the fine-tuning is, though we lose some accuracy. It is up to the user of pre-trained weights to determine which weights to transfer for the given task.

To assess the quality of training, we plot training loss and validation loss over the training period for all four models, specifically for 2 transferred convolution layers. As expected the baseline model, which has the most free parameters, fits the training data the closest overall. When we consider validation loss, however, the variance is very prominent for all models. This may indicate that, compared to the baseline model, the transferred models overfit training less. It also cautions our test MAE results; perhaps the differences between models are more attributable to variance caused by training parameters than by inherent differences between pre-train properties

Comparing Table 1 and Table 2, we also notice that overall the final test MAE roughly tracks the amount of data used during pre-training. This indicates that quality and amount of data used in pre-training likely accounts for differences in the quality of the model. To account for this, we pre-train an additional model to predict formation energy, but using the same 7042 data points as the combined model. We then use it to predict band gap, similarly to the first experiment. The results are summarized in Table 3.

Table 3: Results for Test MAE. Pre-train data size 7042 for both models. Combined model is reproduced from Table 2.

Model	**Mean Absolute Error**		
Pre-trained Model	*First Layer*	*First 2 Layers*	*First 3 Layers*
Combined	0.820	0.850	0.870
Formation Energy	1.022	1.142	1.437

Table 3 shows us positive results for a generalizable CGCNN. When all variability in training size is controlled to be identical, formation energy, which performs better than other labels in transfer learning, performs much worse than the combined labels in transfer learning. Thus, Table 3 shows us that training on multiple labels may learn the fundamental features of the crystal better than training on a specific label, as our intuition tells us.

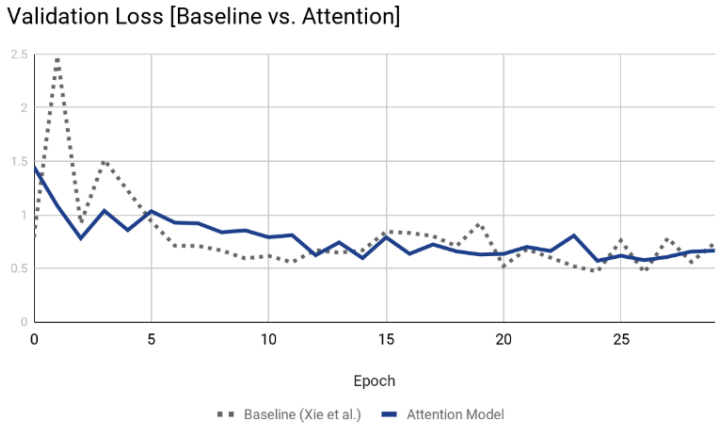

Figure 6: Validation loss over time for attention model compared to state-of-the-art baseline model.

On the other hand, our attention model performs similarly to the baseline model. The attention model and baseline model run similar in computation time, but it is interesting to notice that the validation loss are similar to each other, given that the attention model has half the weights of the baseline model. However, the test MAE for attention scored slightly higher than the baseline, so we did not conduct experiments of transfer learning on the attention model.

6 Conclusion

In this work we investigate the CGCNN model from [4] and transferability and generalizability of different models to different crystal properties than pre-training. We also incorporate a simple attention model to observe potential improvements in accuracy or run time. In initial experiments we find that models pre-trained on formation energy transfer best, but further analysis suggests that this may be due to high variance and the amount of pre-training data. If we control for the amount of pre-training data, models pre-trained to predict multiple crystal properties appear to transfer better than models pre-trained on a single property. Our simple attention model achieves similar run time and validation accuracy while reducing the number of parameters in the model. Further studies and visualizations based on this attention model may help to elucidate salient interactions between atoms in crystals.

For future work, one should investigate how to make graph networks more versatile in crystals. One potential improvement is to standardize what defines and constitutes an edge in crystal graphs. We currently define edges based on a threshold, so for a small but not insignificant amount of crystals, we cannot establish any edge between any atom, meaning CGCNN cannot make any predictions on its property.

7 Team Contribution

Simon Deng (33.33%) Added transfer learning features to CGCNN model. Converted crystal property data to supported CSV format. Ran initial experiments on transfer learning properties and layers. Created figures of the performance results for both the baseline and transfer learning models.

Jerry Park (33.33%) Applied and implemented attention mechanism into the convolutional network. Ran part of the experiments on transfer learning. Ran experiments on attention networks.

Aaron Sy (33.33%) Contributed partially to transfer learning implementation. Converted code and data for prediction on multiple properties. Ran experiments for combined model predicting multiple properties, and formation energy model trained on limited data. Dealt with data preprocessing and investigation.

References

[1] Saal, J. E., Kirklin, S., Aykol, M., Meredig, B., and Wolverton, C. "Materials Design and Discovery with High-Throughput Density Functional Theory: The Open Quantum Materials Database (OQMD)", JOM 65, 1501-1509 (2013). doi:10.1007/s11837-013-0755-4

[2] A. Jain*, S.P. Ong*, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K.A. Persson (*=equal contributions) The Materials Project: A materials genome approach to accelerating materials innovation APL Materials, 2013, 1(1), 011002. doi:10.1063/1.4812323

[3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv:1609.02907, 2016.

[4] T. Xie and J. C. Grossman, "Crystal Graph Convolutional Neural Networks for Accurate and Interpretable Prediction of Material Properties", arXiv:1710.10324, 2017.

[5] International Tables for Crystallography (2006). Volume G, Definition and exchange of crystallographic data. doi:10.1107/97809553602060000107