Convolutional Neural Network for Atomic Structures

Junwoong Yoon
Department of Chemical Engineering
Carnegie Mellon University
junwoony@andrew.cmu.edu

Ni Zhan
Department of Chemical Engineering
Carnegie Mellon University
nzhan@andrew.cmu.edu

Mingjie Liu
Department of Chemical Engineering
Carnegie Mellon University
mingjie1@andrew.cmu.edu

1 Introduction

The demand for energy is increasing globally. In an effort to meet this demand, alternative fuels such as H2 and synthetic hydrocarbons are used. These fuels are generated by electrochemical reduction reactions, which are currently limited by lack of a sufficiently active, selective, and low-cost catalyst [5].

In the reaction process, a reactant molecule (adsorbate) lands or adsorbs onto a catalyst surface site. The energy associated with this adsorption indicates effectiveness of a potential catalyst. To screen catalysts, we can calculate the adsorption energy for any possible catalyst using first principles calculation called density functional theory (DFT). DFT is the most accurate method to obtain energies and forces on atoms computationally, but is computationally expensive.

To calculate the adsorption energy using DFT, we require the exact positions of the adsorbate and surface catalyst at the equilibrium state. The usual way to obtain the equilibrium structure is by guessing a starting configuration and using DFT to predict the next configuration for a small timestep. Repeating this many times is known as a molecular dynamics (MD) simulation and results in a trajectory of configurations, reaching equilibrium. Because DFT calculations along the trajectory are computationally expensive and we are only interested in the final state, we aim to use machine learning (ML) to predict the final equilibrium state given an initial configuration.

We have data from MD trajectories and DFT; each snapshot along the trajectory has configuration of atoms, its energy, and forces on atoms. The final snapshot of the trajectory is the ground truth equilibrium state that will be used to compare with a ML model prediction. Specifically, we predict the distance between the adsorbate and its neighboring atoms.

The model inputs are atomic configurations which are analogous to images or 3D images. We use a convolutional neural network (CNN) to learn representations of each atom's local environment, and then apply a feed forward network to predict distances between two atoms.

Preprint. Work in progress.

2 Background

As a preliminary result, we demonstrate that our convolutional neural network can extract meaningful representations from the input features (atomic structures) to predict properties. We trained the CNN to predict adsorption energies of CO on various pure and intermetallic surfaces. The model was trained with 90% of 20,000 DFT data and then validated and tested with 5% of the data, each. We use the same DFT dataset to train our main-results model to predict distances, described in the following sections. For the preliminary adsorption energy model, the test and validation RMSEs were less than 0.3 eV, and MAEs were less than 0.2 eV.

The preliminary results show that the CNN was successful in predicting adsorption energies from configuration of atoms. In literature, models with high accuracy for adsorption energy reach errors of 0.1 to 0.2 eV for RMSE or MAE [6], while our model reached 0.3 eV error. Therefore, the model should also accurately predict distance between the adsorbate and surface atoms because adsorption energy is related to these distances. In this work, we extend the CNN to predict a novel property, and thereby show that the model structure can be applied to multiple physical properties.

3 Related Work

Advanced deep convolutional networks have enabled breakthroughs in processing raw representation of data such as image, video, and speech, and have recently been used for atomic configurations. Xie et. al. applied a convolutional neural network to a graph representation of crystals to predict various properties of crystals using 10^4 DFT calculations as training data [6]. The graph representation of the crystal structure encoded both atomic and bonding interactions between atoms. Their model, called crystal-graph convolutional neural network (CGCNN), predicted accurate properties with respect to DFT calculations at computational speeds orders of magnitude faster. Other work by Schütt et. al. used continuous filter convolutional layers to represent the location and interaction of atoms and learn energies and forces [4, 2, 3].

Professor Ulissi's group in chemical engineering department, Carnegie Mellon University, used the graph convolutional neural networks to predict adsorption properties of CO on a variety of intermetallic surfaces, and achieved high accuracy to DFT data.

Other researchers are actively using deep learning methods and DFT data to predict chemical properties of atoms or molecules for design of novel materials. However there has been little effort to circumvent extra DFT calculations by directly predicting equilibrium atomic structures as proposed in this work.

4 Methods

The training data are from snapshots of DFT MD trajectories. A data point for model input is the atomic configuration from an initial snapshot of an MD trajectory, with various numbers of atoms in a configuration or "slab". The corresponding model output are the distances between the adsorbate atom and its four nearest atoms. For the ground truth label, we use the final snapshot in the MD trajectory to identify the closest four atoms and their distances to the adsorbate. We use data structures to index and keep track of atoms in a configurational image. The entire configurational image impacts the prediction because physical properties are the result of combined interaction between all atoms with each other. The overall model, or CGCNN, structure is shown in Fig. 1. The first step is to represent the atomic configurations in suitable data format. We describe this process and our input data structure in the "Graph Representation of Data" section. We then describe the details of the convolutions which result in a feature vector describing the local environment around each atom in the configuration. Finally, we concatenate the feature vectors of two atoms as a "bond feature vector", which we feed to a feed forward neural net to predict the bond distance between those two atoms.

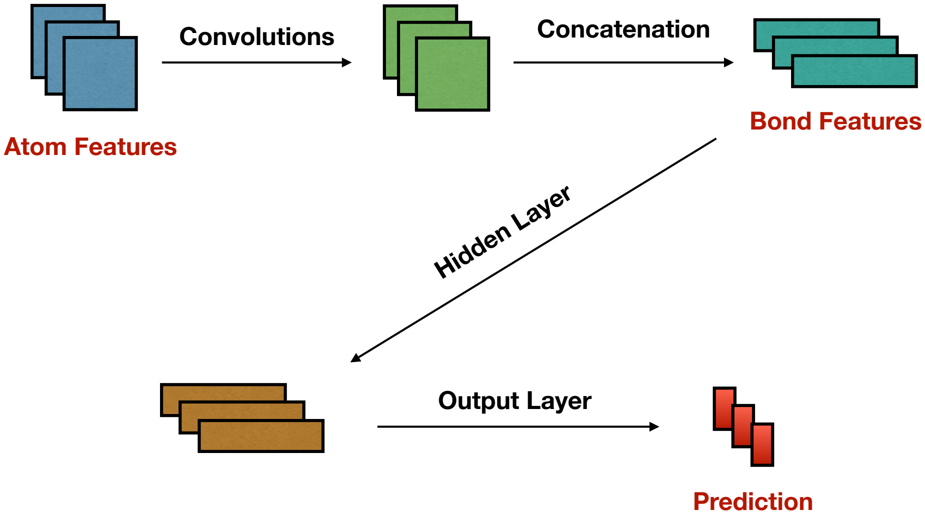


Figure 1: Complete CGCNN architecture

Graph Representation of Data

We collect about 20,000 MD DFT trajectories for CO adsorption on surfaces of pure metals, metal alloys, and intermetallic alloys for this work. This dataset contains indices and positions of individual atoms, distances between neighboring atoms, and adsorption energies of the CO adsorbate to the surface for both the initial and final slab structures. We convert initial atomic slab structures into machine readable graph representations by encoding atomic information and bonding interactations between neighboring atoms. Fig. 2 shows some of the key components of the numerical representation of atomic slab structures, and Fig. 3 shows the data structure for an example slab. Each slab has a matrix of Atomic Feature Vectors $\mathbf{v}$ and a tensor of Neighbor Feature Vectors $\mathbf{u}$, which will be described further.

Each atom in the slab structures is represented by an Atomic Feature Vector $\mathbf{v}_i$ that encodes the useful atomic properties such as atomic group number, period number, etc. With one hot encoding of these atomic properties, the final atomic feature vector has 92 binary values and maps to the unique chemical element the atom corresponds to. This atomic feature representation is identical to the one used in the original CGCNN model [6].

The Neighbor Feature Vectors and Neighbor Feature Index capture connectivity and relative location information between the atoms. We represent the connectivity of atoms with neighboring features based on "Voronoi Connectivity" [1]. Each connection (bond) between atoms is represented with a vector with $N_{periods}$ elements. $N_{periods}$ is the number of repeated cell considered. (Slabs are cells repeated in x and y directions infinitely). In our case, we consider six cells. Therefore, there will be six bonds between atom i and atom j, since atom j in the repeated cells also counts. We consider 12 neighboring atoms to create neighbor feature and index vectors for each atom. Therefore, for each atom in the slabs, it has a neighbor feature tensor of $N_{neighbors} \times N_{periods}$, specifically 12×6. Atomic features and neighbor features are the input to the CGCNN model.

The target is to predict the N closest distances of the adsorbate atom from the other atoms. In our case, N is chosen to be four. As shown in Fig. 3, we store indices of the adsorbate and its closest four atoms, and their corresponding bond distances to the adsorbate based on the final configurations. We use these bond distances as our target values.

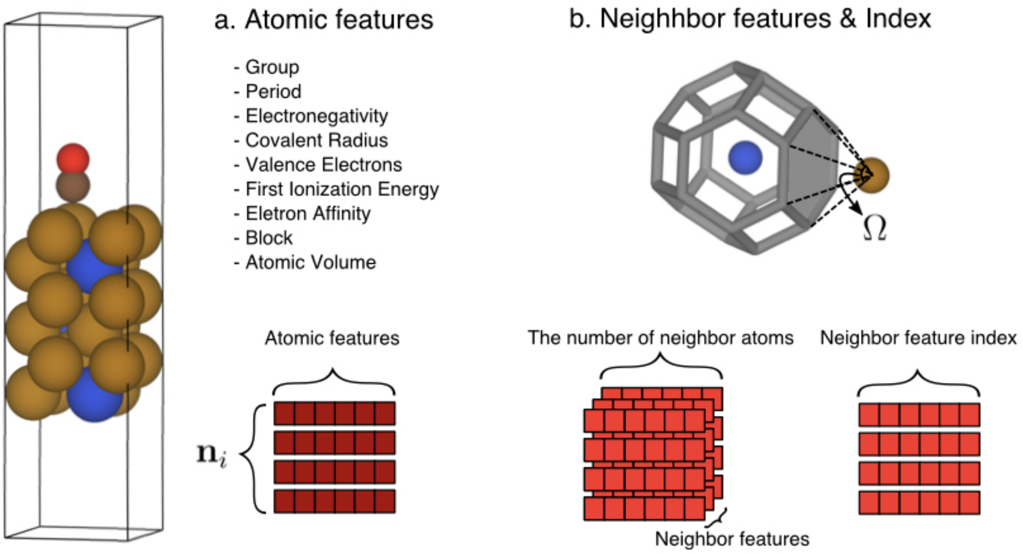
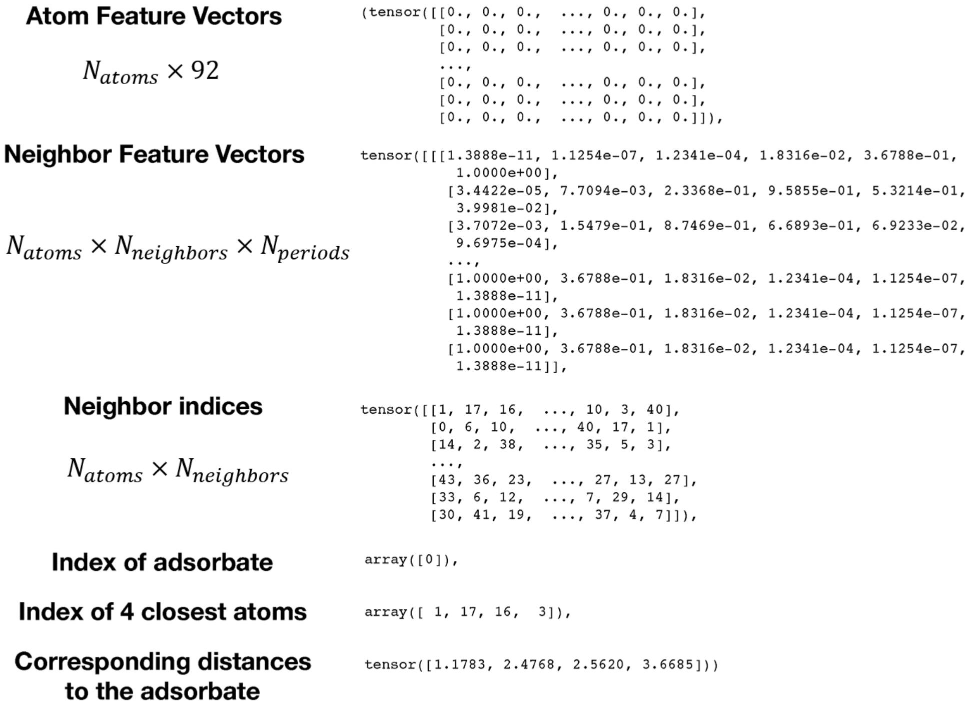


Figure 2: A graphical representation of converting an atomic structure into numerical inputs for the convolutional neural network. (a) Nine atomic properties used to represent each atom in a slab (b) The solid angle (Ω) used to encode neighbor information to create neighbor atom feature vectors and corresponding indices of the neighboring atoms



Atom Feature Vectors

$N_{atoms} \times 92$

```
(tensor([[0., 0., 0.,  ..., 0., 0., 0.],
         [0., 0., 0.,  ..., 0., 0., 0.],
         [0., 0., 0.,  ..., 0., 0., 0.],
         ...,
         [0., 0., 0.,  ..., 0., 0., 0.],
         [0., 0., 0.,  ..., 0., 0., 0.],
         [0., 0., 0.,  ..., 0., 0., 0.]]),
```

Neighbor Feature Vectors

$N_{atoms} \times N_{neighbors} \times N_{periods}$

```
tensor([[[1.3888e-11, 1.1254e-07, 1.2341e-04, 1.8316e-02, 3.6788e-01,
          1.0000e+00],
         [3.4422e-05, 7.7094e-03, 2.3368e-01, 9.5855e-01, 5.3214e-01,
          3.9981e-02],
         [3.7072e-03, 1.5479e-01, 8.7469e-01, 6.6893e-01, 6.9233e-02,
          9.6975e-04],
         ...,
         [1.0000e+00, 3.6788e-01, 1.8316e-02, 1.2341e-04, 1.1254e-07,
          1.3888e-11],
         [1.0000e+00, 3.6788e-01, 1.8316e-02, 1.2341e-04, 1.1254e-07,
          1.3888e-11],
         [1.0000e+00, 3.6788e-01, 1.8316e-02, 1.2341e-04, 1.1254e-07,
          1.3888e-11]],
```

Neighbor indices

$N_{atoms} \times N_{neighbors}$

```
tensor([[1, 17, 16,  ..., 10, 3, 40],
        [0, 6, 10,  ..., 40, 17, 1],
        [14, 2, 38,  ..., 35, 5, 3],
        ...,
        [43, 36, 23,  ..., 27, 13, 27],
        [33, 6, 12,  ..., 7, 29, 14],
        [30, 41, 19,  ..., 37, 4, 7]]),
```

Index of adsorbate

```
array([0]),
```

Index of 4 closest atoms

```
array([ 1, 17, 16,  3]),
```

Corresponding distances to the adsorbate

```
tensor([1.1783, 2.4768, 2.5620, 3.6685]))
```

Figure 3: Example of the numerical representation of an atomic slab structure

Model

After we create graph representations of atoms, we concatenate atomic feature vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ of atoms i and j and neighbor feature vectors $\mathbf{u}_{(i,j)}$ between atoms i and j, to a new vector $\mathbf{z}^t_{(i,j)} = \mathbf{v}^t_i \oplus \mathbf{v}^t_j \oplus \mathbf{u}_{(i,j)}$. We then build convolutional neural network on top of each atom to extract features of atoms that are optimum for predicting target bond distances. We iterate several convolutions to update atom feature vectors $\mathbf{v}_i$ by convolution with surrounding atoms and bonds features.

$$\mathbf{v}^t_i = \mathbf{v}^{t-1}_i + \sum_j \sigma\left(\mathbf{z}^{t-1}_{(i,j)}\mathbf{W}^{t-1}_f + \mathbf{b}^{t-1}_f\right) \odot g\left(\mathbf{z}^{t-1}_{(i,j)}\mathbf{W}^{t-1}_s + \mathbf{b}^{t-1}_s\right) \tag{1}$$

where $\odot$ denotes an element-wise multiplication, σ denotes a sigmoid function, g denotes any non-linear activation function, and $\mathbf{W}$ and $\mathbf{b}$ are weights and biases of the neural networks, respectively. After $\mathbf{R}$ convolutions, we create bond feature vectors of the adsorbate and each of its closest four atoms by concatenating the atom feature vectors of the adsorbate and each of those four atoms, i.e., $\mathbf{v}_{bond} = \mathbf{v}^{\mathbf{R}}_{adsorbate} \oplus \mathbf{v}^{\mathbf{R}}_{closestatom}$. Then we create multiple hidden layers to capture the complex mapping between the extracted bond features and the target bond distances followed by an ouput layer that predict bond distances.

Experiment Details

For each atom in the slabs, the atom feature vector for each atom i, $\mathbf{v}_i \in \mathbb{R}^{92}$. The neighbor vector for each bond ij, $\mathbf{u}_{ij} \in \mathbb{R}^6$, since six repeated cells are considered. In the CGCNN model, eight convolution layers and two fully connected layers are used. The atom feature vector after the convolutions is in $\mathbb{R}^{46}$. After concatenation, the feature vector for bond distance prediction fed to the fully connected layer is in $\mathbb{R}^{92}$. There are 92 nodes in each of the two fully connected layers. Table 4 shows the hyperparameters and architecture of the CGCNN. We used 12,000 data points for train, 4,000 for validation, and 4,000 for test sets. We chose the best model weights based on lowest validation loss.

batch size	learning rate	max epochs	convolution layers	fully connected layers
214	0.0056	200	8	2

Table 1: Hyperparameters and Architecture of CGCNN

5 Results

In the experiment, we predict the bond distances of carbon atom from the four closest neighbor atoms. Fig. 4 shows a parity plot that compares the prediction from CGCNN and the target values generated from DFT. The training, validation, and test MAE are 0.09 Å, 0.16 Å, and 0.15 Å, respectively. The training and test MAE are in the same range, which indicates that the model is not overfitting. We expect a satisfactory model MAE of 0.05 Å, and the model should be further improved to reach this level of accuracy. Another metric used to compare our model is percent error, or residual of predicted and actual bond length over actual bond length. An estimate of our percent error for test set is 5%. Typically the bond distance difference between the final and initial structures is around 0.2 Å (or 10% error) so using our predictive model improves on randomly guessing an initial distance for the adsorbate above the surface.

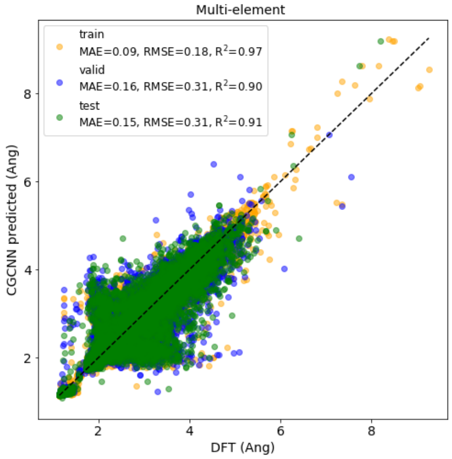


Figure 4: Parity plot of predicted and actual bond distance

6 Discussion and Analysis

To better understand our model performance and limitations, we perform further analysis, including separating the four bond distance predictions within each slab. Fig. 5 shows the parity plots for each bond distance of a slab configuration, from closest bond to furthest bond. The closest bond has the lowest MAE of 0.01, and the MAE successively increases for the farther atoms to 0.09, 0.15, and 0.17.

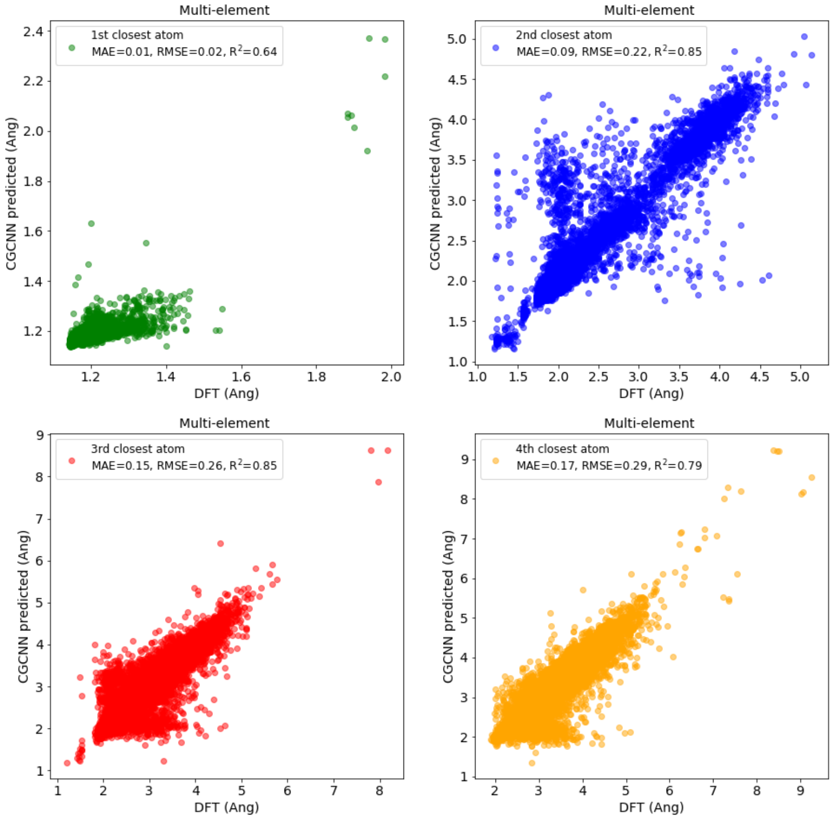


Figure 5: Parity plot of predicted and actual bond distance separated by bond distance

The average error increases when examining further bond distances within a slab. This could be because there is better connectivity or feature respresentation for the close atoms within a slab configuration. The CGCNN is likely learning a relatively local representation around each atom. To improve the CGCNN's ability to predict further bond distances and capture information about further atoms, the connectivity information could be modified to change relative connectivity strength between atoms based on their distance.

The closest bond distance is probably between C and O, and it is a distance within the adsorbate molecule and therefore likely easier to predict. From Fig. 5, the model appears to underpredict the CO bond distance, however the reason for this underprediction is unclear.

From Fig. 5, the model has higher error when predicting bond lengths around 2 to 3 Å. This is probably the most relevant bond, most affected by the combinations of atom interactions, and possibly the bond between the adsorbate and the surface. Therefore, this bond has the most variance across slabs.

One source of error could be that the adsorbate moved significantly from the initial to final configurations. It is possible that the adsorbate moves across the slab surface and becomes closer to different atoms than in the initial configuration. In these cases, the predicted bond lengths could have much higher error. To investigate this, we could find the indices of the closest four atoms of the initial and final configurations and compare them.

To check for overfitting, we compared the training and validation errors trends. Training error continued to decrease, while validation error reached a minimum, which indicates possible overfitting. However, the test error is not satisfactorily low enough.

To decrease overfitting, we could decrease the dimension of the bond feature. As examples, we could try different bond feature vectors such as averaging two atomic feature vectors or element multiplying them, and both would decrease the bond feature dimension to $\mathbb{R}^{46}$. Another way to decrease overfitting is decreasing the size of feed forward net, for example decreasing the number of nodes in each layer. Using a smaller feed forward net would force the convolutional layers to learn more useful representations of atom environments.

In our training data, we have 7000 different types of slabs. Many slabs appear only once in the MD DFT trajectories. The CGCNN model should generalize to new slabs, because of the atom feature vectors for different atom types and convolutions. However, it is possible that the model does not generalize well because there was not enough data used for training. We can include more configurations along the trajectory, including the final configuration, in the training set to improve model performance.

References

[1] Olexandr Isayev et al. “Universal Fragment Descriptors for Predicting Properties of Inorganic Crystals”. In: *Nature Communications* 8.nil (2017), p. 15679. DOI: 10.1038/ncomms15679. URL: https://doi.org/10.1038/ncomms15679.

[2] K. T. Schütt et al. “Schnet - a Deep Learning Architecture for Molecules and Materials”. In: *The Journal of Chemical Physics* 148.24 (2018), p. 241722. DOI: 10.1063/1.5019779. URL: https://doi.org/10.1063/1.5019779.

[3] K. T. Schütt et al. “Schnetpack: a Deep Learning Toolbox for Atomistic Systems”. In: *Journal of Chemical Theory and Computation* 15.1 (2018), pp. 448–455. DOI: 10.1021/acs.jctc.8b00908. URL: https://doi.org/10.1021/acs.jctc.8b00908.

[4] Kristof Schütt et al. “SchNet: A continuous-filter convolutional neural network for modeling quantum interactions”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 991–1001. URL: http://papers.nips.cc/paper/6700-schnet-a-continuous-filter-convolutional-neural-network-for-modeling-quantum-interactions.pdf.

[5] Kevin Tran and Zachary W. Ulissi. “Active Learning Across Intermetallics To Guide Discovery of Electrocatalysts for Co2 Reduction and H2 Evolution”. In: *Nature Catalysis* 1.9 (2018), pp. 696–703. DOI: 10.1038/s41929-018-0142-1. URL: https://doi.org/10.1038/s41929-018-0142-1.

[6] Tian Xie and Jeffrey C. Grossman. “Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties”. In: *Physical Review Letters* 120.14 (2018), p. 145301. DOI: 10.1103/physrevlett.120.145301. URL: https://doi.org/10.1103/physrevlett.120.145301.