GRAPH NEURAL NETWORK ANALYSIS OF LAYERED MATERIAL PHASES

Kuang Liu, Ken-ichi Nomura,
Rajiv K. Kalia, Aiichiro Nakano,
Priya Vashishta

Collaboratory for Advanced Computing and Simulations
University of Southern California
Los Angeles, CA, USA
{liukuang, knomura, rkalia, anakano, priyav}@usc.edu

Pankaj Rajak

Argonne Leadership Computing Facility,
Argonne National Laboratory

Argonne, IL, USA
prajak@anl.gov

ABSTRACT

We apply graph neural network (GNN)-based analysis to automatically classify different crystalline phases inside computationally-synthesized molybdenum disulfide monolayer by reactive molecular dynamics (RMD) simulations on parallel computers. We have found that addition of edge-based features like distance increases the model accuracy up to 0.9391. Network analysis by visualizing the feature space of our GNN model clearly separates 2H and 1T crystalline phases inside the network. This work demonstrates the power of the GNN model to identify structures inside multimillion-atom RMD simulation data of complex materials.

Keywords: Graph, Neural network, Layered material.

1 INTRODUCTION

Machine learning (ML) has proved ubiquitous applicability in image recognition, natural language processing and many other areas involving real-world data. Such success has inspired chemists to apply ML, especially deep learning, to better understand chemical processes (Hansen *et al.* 2015; Montavon *et al.* 2012). With the tremendous amount of data generated by computer simulations, deep neural networks in particular have shown significant power in many chemistry tasks, including the prediction of properties (Wei *et al.* 2012), drug discovery (Altae-Tran *et al.* 2017) and understanding of quantum molecular dynamics (Mendoza *et al.* 2018). Most of the existing works deal with small organic molecules, where the datasets are formed by a collection of molecular graphs with adjacency matrix describing the connectivity of atoms. Such graph-based datasets require a unique deep-learning architecture to capture the additional structured information. Graph neural networks (GNN) (Scarselli *et al.* 2009), a variation of the widely used convolutional neural networks (CNN) (Krizhevsky *et al.* 2012), process graph data as nodes and edges in a learnable fashion. Duvenaud *et al*. (2015) generalized molecular feature extraction methods by a series of differentiable functions to produce low-dimensional molecular fingerprints. Kearnes *et al*. (2016) integrated edge representations into the learning functions to capture more structural features from the graph. Li *et al*. (2016) applied gated recurrent units to update the hidden state during the learning phase, so that sequence-based methods (*e.g*., LSTM) can also be injected in graph models.

More recently, these GNN applications to isolated molecules have been extended by materials scientists to handle infinitely-repeated crystal structures (Xie and Grossman 2018). However, this crystal-graph CNN has been limited to periodically-repeated, small crystalline unit cells each involving a few atoms. In this paper, we extend the applicability of GNN to general material graphs composed of millions of nodes. This

is a challenge since material structure often is a mixture of various crystalline phases and defects, which are interconnected to each other *via* bonds, resulting in a highly-complex, massive graph. Here, we propose a new variant of GNN to identify different phases inside an atomically-thin molybdenum disulfide (MoS_2) monolayer that is computationally synthesized by reactive molecular dynamics simulation (RMD) simulation (Hong *et al.* 2017), mimicking experimental chemical vapor deposition (CVD). MoS_2 is an archetype of atomically-thin layered materials (Geim and Grigorieva 2013), for which ML has extensively been applied (Bassman et al. 2018). Here, our model analyzes the local graph topology around each atom, and classifies it into crystalline 2H or 1T structures as is shown in Fig. 1.

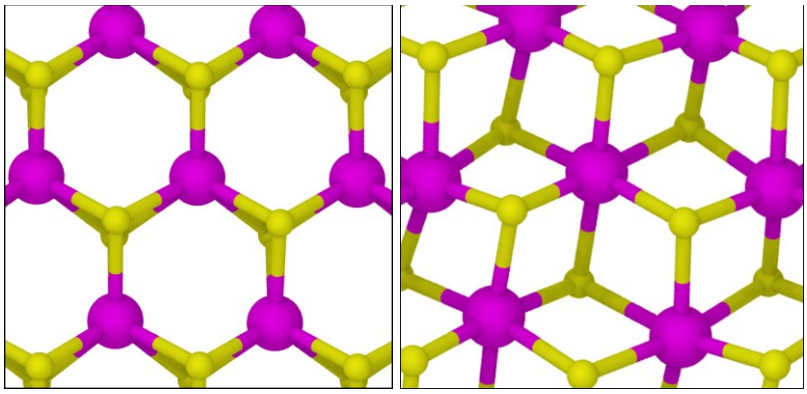

Figure 1: Top views of 2H (left) and 1T (right) crystalline phases of MoS2 monolayer. Magenta and yellow spheres represent Mo and S atoms, respectively, whereas Mo-S bonds are represented by cylinders.

2 METHOD

Figure 2 presents a high-level schematic of our classification task between 2H and 1T crystalline phases using GNN. The following subsections explain key components of this learning architecture, including dataset generation and GNN.

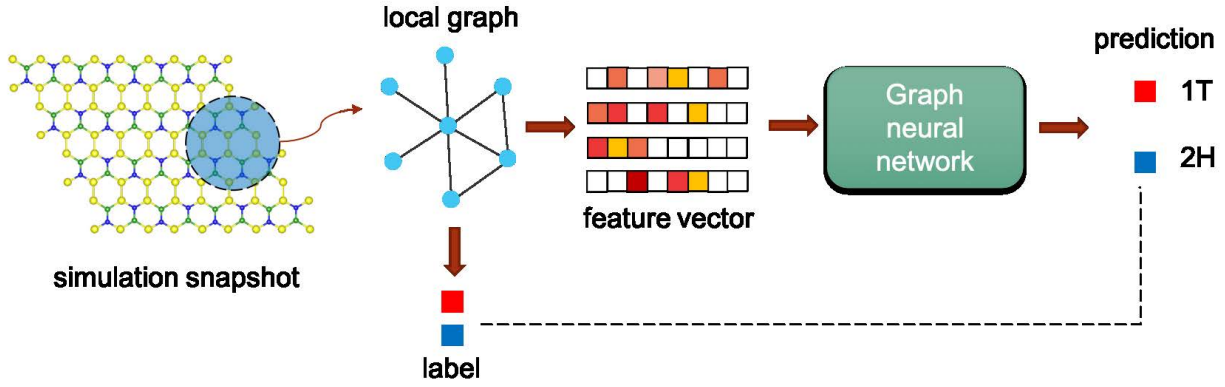

Figure 2: Schematic of 2H vs. 1T phase classification by graph neural network.

2.1 Dataset Generation

We have performed RMD simulation to synthesize MoS_2 monolayer by sulfidation of molybdenum trioxide (MoO_3) precursor (Hong *et al.* 2017), followed by thermal annealing. RMD simulation follows the trajectory of all atoms while computing interatomic interaction using first principles-informed reactive bond-order and charge-equilibration concepts (Senftle *et al.* 2016). We have designed scalable algorithms to perform large RMD simulations on massively parallel computers (Nomura *et al.* 2008; Nomura *et al.* 2015). Our RMD produces polycrystalline MoS_2 monolayer, where different regions of the monolayer belong to either 2H or 1T phase of MoS_2 crystal. Here, each atom inside the synthesized polycrystal is

connected to its nearest-neighbor atoms by forming bonds with them, which in turn are connected to their nearest neighbors and so on. Such bond formation between atoms makes the entire MoS_2 monolayer a massive graph consisting of multimillion nodes (atoms) and edges (bonds). Hence, identification of 2H and 1T phases using a conventional graph-based ML model is not feasible for the entire graph. To circumvent this problem, we have randomly sampled 66,896 atoms from the total MoS_2 monolayer, and for each of these atoms, created a local graph using all their neighbors within a cutoff radius of 0.8 nm. The rationale behind this choice is that, for each atom, graph structure generated by its first and second nearest neighbors is able to distinguish its phase (2H or 1T). These 66,896 local graph structures serve as the training data for our neural-network model, and it consisted of 18,650 2H, 32,446 1T and 16,000 disordered structures.

2.2 Graph Neural Networks

Graph-based data in general can be represented as $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ is the set of nodes and $\boldsymbol{E}$ is the set of edges. Each edge $e_{uv} \in \boldsymbol{E}$ is a connection between nodes u and v. If $\boldsymbol{G}$ is directed, we have $e_{uv} \not\equiv e_{vu}$; if $\boldsymbol{G}$ is undirected, instead $e_{uv} \equiv e_{vu}$. Unless specified, the remaining of this paper will deal with undirected graphs, but we will show that it is trivial to modify our model to process directed graph data. It should be pointed out that, in molecular graphs, the nodes are actually atoms and the edges are atomic bonds, thus, the two pairs of terms are used interchangeably in this paper.

The goal of GNN is to learn low-dimensional representation of graphs from the connectivity structure and input features of nodes and edges. The forward pass of GNN has two steps, *i.e.*, message passing and node-state updating. The architecture is summarized by the following recurrence relations, where *t* denotes the iteration count:

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}) \,, \tag{1}$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}), \tag{2}$$

where $N(v)$ denotes the neighbors of v in graph $\boldsymbol{G}$. The message function M_t takes node state h_v^t and edge state e_{vw} as inputs and produces message m_v^{t+1}, which can be considered as a collection of feature information from the neighbors of v. The node states are then updated by function U_t based on the previous state and the message. The initial states h_v^0 are set to be the input features of atoms, which we will discuss in the next section. Here, we use normalized adjacency matrix $\tilde{A}$ (Chen *et al.* 2018) of the graph coupled with some other features (which will be discussed below) as the edge state. As shown in Fig. 3, these two steps are repeated for a total of *T* times in order to gather information from distant neighbors, and the node states are updated accordingly. GNN can be regarded as a layer-wise model that propagates messages over the edges and update the states of nodes in the previous layer. Thus, *T* can be considered to be the number of layers in this model.

The exact form of message function is

$$m_v^{t+1} = A_v \boldsymbol{W^t} [h_1^t \ \dots \ h_v^t] + \boldsymbol{b}, \tag{3}$$

where $\boldsymbol{W^t}$ are weights of GNN and $\boldsymbol{b}$ denotes bias. We use gated recurrent units (Cho *et al.* 2014) as the update function:

$$z_v^t = \sigma(\boldsymbol{W^z} m_v^t + \boldsymbol{U^z} h_v^{t-1}) \quad , \tag{4}$$

$$r_v^t = \sigma(\boldsymbol{W^r} m_v^t + \boldsymbol{U^r} h_v^{t-1}) \quad , \tag{5}$$

$$\widetilde{h_v^t} = \tanh(\boldsymbol{W} m_v^t + \boldsymbol{U}(r_v^t \odot h_v^{t-1})) \,, \tag{6}$$

$$h_v^t = (1 - z_v^t) \odot h_v^{t-1} + z_v^t \odot \widetilde{h_v^t}, \tag{7}$$

where $\odot$ denotes element-wise matrix multiplication and $\sigma(\cdot)$ is sigmoid function for nonlinear activation.

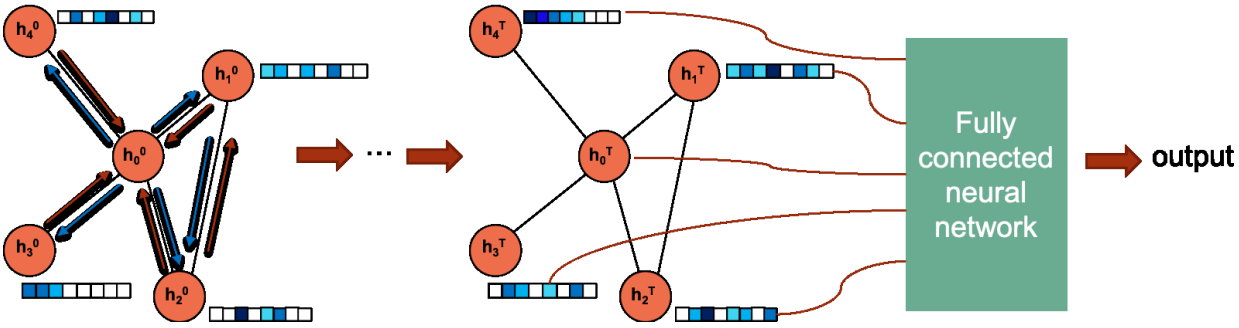

Figure 3: Repeating message function and update function T times to learn atom representations.

Once GNN learns low-dimensional atom representations, we feed them to a generic ML model, *e.g.* fully-connect network, to predict 2H *vs*. 1T phases as a complete classification task. We argue that the learned atom representations for the molecular graphs can better interpret the structural uniqueness of 2H and 1T phases, so that the model can achieve higher predictive performance. In the next section, we show the experiment results to support this assumption.

3 EXPERIMENT AND ANALYSIS

We have studied the structure of CVD-grown MoS_2 monolayer using RMD simulation (Fig. 4). The initial system for RMD simulation consists of MoO_3 slab surrounded from top by S_2 gas. The dimension of the RMD simulation is 211.0 × 196.3 × 14.5 (nm^3) in the x-, y- and z-directions, respectively, and it consist of a total of 4,305,600 atoms. The entire system is subjected to an annealing schedule, where the system temperature is first increased to 3,000 K, and subsequently its temperature is quenched to 1,000 K, where it is held for 1 nanosecond (ns). This is followed by two annealing cycles consisting of a heating step from 1,000 K to 1,600 K for 0.4 ns followed by a thermalization step at 1,600 K for 1.5 ns and a cooling step from 1,600 K to 1,000 K for 0.4 ns. This annealing schedule facilitates the reaction of S_2 with MoO_3 slab, which results in the formation of polycrystalline MoS_2 monolayer where different regions of the synthesized MoS_2 monolayer belongs to either 2H or 1T phase.

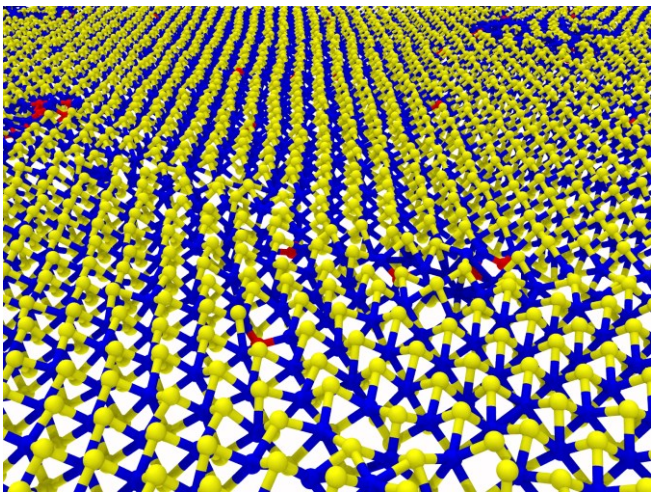

Figure 4: Snapshot of computationally-synthesized MoS2 monolayer. Ball-and-stick representation is used to show the positions of atoms and covalent bonds with neighboring atoms. Atoms are color-coded as yellow for sulfur (S), blue for molybdenum (Mo) and red for oxygen (O), respectively.

3.1 Input Features and Training Settings

The small molecular graphs carved out from the simulation results carry a number of properties on atoms as well as bonds. See Table 1 for details. These properties are transformed into vector format and embedded with nodes and edges as the initial states of the GNN model. Due to the fact that the numbers of atoms in different atom-centered graphs are different, all input features are zero-padded up to a larger dimension, $d = 40$, to conform with Tensorflow's dataset interface (Abadi *et al.* 2015).

Table1: Input node and edge features.

	Feature	Description	Datatype
Node	atom type	Mo or S, a one-hot vector	2 integers
	charge	Atomic charge	1 float
Edge	distance	Distance between atoms in nm	1 float
	bond order	Dimensionless chemical bond order	1 float

We randomly shuffle and split the dataset as follows: 50,000 graphs in training set; 5,000 graphs in validation set; and another 5,000 in test set. The remaining 6,896 graphs are not used in our experiments. The number of layers of our GNN model is $T = 2$, batch size set to 20, and we train the model for a maximum of 100 epochs using Adam (Kingma & Ba 2015) with a learning rate of 0.01.

3.2 Predictive performance

To investigate how the input features affect the predictive performance, we perform a series of experiments with different selections of edge features, while node features are kept fixed. Results are shown in Table 2. In the first trail, we only use node features as inputs, revealing no spatial information but just adjacency matrix for training. The F1 score of 2H in test set is only 0.5424, meaning that nearly half of the 2H phases are misclassified. Next, we add edge features to the model, then observe improvements in both 1T and 2H classes. It turns out that distance and bond order have almost equivalent effect to the GNN. The reason for not seeing increase in performance of the model when distance and bond order are added together as edge feature is because both of them make an estimate of bonding between two atoms and hence are highly correlated. High bond-order values mean strong bonding between atoms and small values mean weak bonding, whereas distance cutoff based feature makes an absolute decision whether atoms are bonded or not. Since the bond length between atoms is an important feature to distinguish these phases, the node+distance based model gives the higher performance.

Table 2: F1 scores for different input features. Higher value sinifies better classification accuracy.

input features	1T	2H
node only	0.7821	0.5424
node + edge	0.9321	0.8642
node + distance	**0.9391**	**0.8855**
node + bond order	0.9305	0.8761

Figure 5 plots the ROC (receiver operating characteristic) curve obtained in one of our experiments using both node and edge features. The curves of 1T and 2H are close to the upper-left corner, indicating that the model achieves high accuracy on both classes. In addition, we calculate a more quantitative measure, ROC-AUC (area under the ROC curve) score, as is shown in Table 3, to verify the performance of the model. We observe similar results as the F1 scores, which confirms the robustness of our conclusion drawn above.

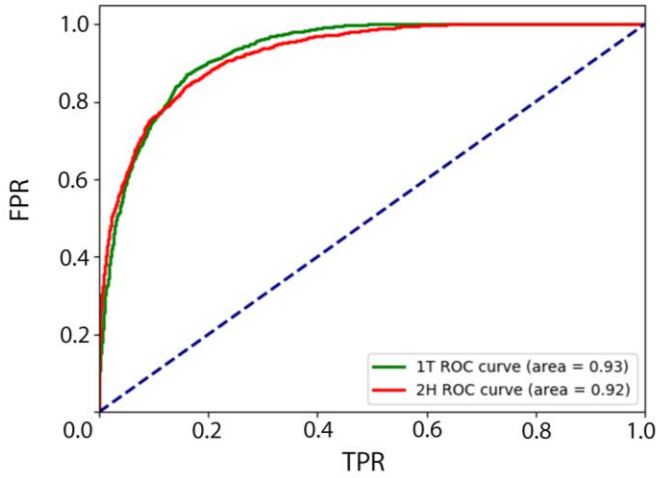

Figure 5: ROC curves for 1T (green) and 2H (red) phases using node and edge features as input, which correspond to the second row in Table 3. The x-axis is the true positive rate (TPR) and y-axis is the false positive rate (FPR).

Table 3: ROC-AUC for different input features. Value is in the range of (0, 1). Higher value signifies better classification accuracy.

input features	1T	2H
node only	0.88	0.87
node + edge	0.93	0.92
node + distance	**0.95**	**0.95**
node + bond order	0.93	0.93

3.3 Visualization of Hidden Node States

It is still an open and active research area to interpret the learning process of ML models. In this work, we try to provide some insight on how GNN learns from graphs by visualizing the evolution of the hidden states during the training phase. As is shown in Fig. 6, this molecular graph has 20 atoms with a dimension of 3 (see Table 1 for details) for the initial atom features. There are several patterns shared by the atoms, for example, the first 7 atoms have very similar feature encodings. In the second and third layers, GNN expands the dimension to 25, which is a hyperparameter of the model, while the feature of each atom gradually becomes divergent compared to the initially state. Such divergence would make it easier to find the hyperplane in the feature space, so that the model can achieve high classification accuracy.

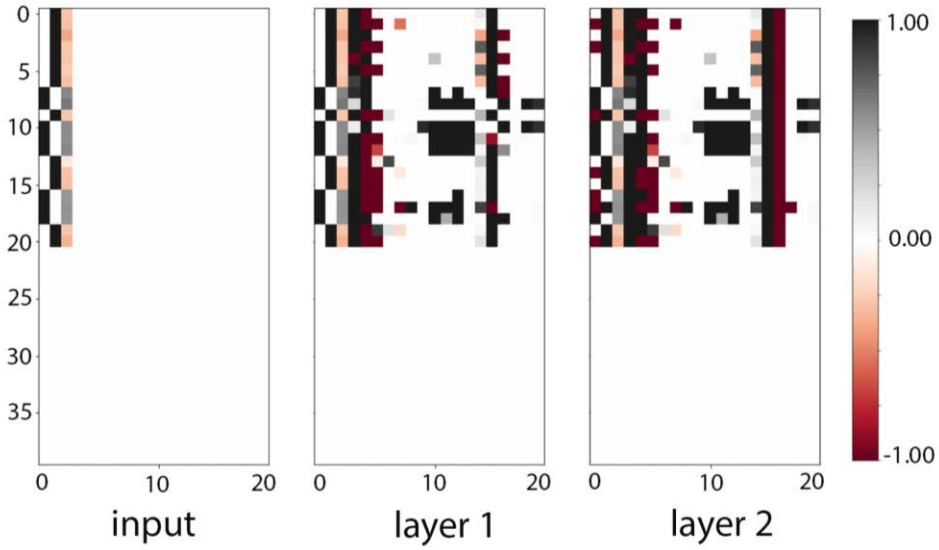

Figure 6: Layer-wise evolution of node states. The x-axis is the dimensions of features (padded to 20) and y-axis is the number of atoms (zero-padded to 40) in the graph. Each row represents a feature vector of an atom, and the color of each pixel indicates the value on the unit.

4 CONCLUSION

In summary, we have shown that graph neural network-based analysis can automatically classify different phases present in RMD simulation of MoS_2 synthesis. Furthermore, we found that addition of edge based features (especially bond distance) increases the model accuracy significantly. Network analysis by visualizing the feature space of our GNN model shows clear separation of the 2H and 1T graph structures inside the network, which helps identify and better understand these structures.

This is contrary to conventional techniques for structural analysis (*e.g.*, common neighborhood analysis and centro-symmetry parameter calculation). While they work for mono-atomic FCC, BCC and HCP crystals, these conventional methods do not distinguish 2H and 1T crystalline phases in transition-metal dichalcogenide layers considered here. Due to the lack of a readily available order parameter that can identify each structure type, our GNN model serves as an indispensable analysis tool for general materials.

ACKNOWLEDGMENTS

This work was supported as part of the Computational Materials Sciences Program funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, under Award Number DE-SC0014607. Computations were performed at the Argonne Leadership Computing Facility (ALCF) under the DOE INCITE and Aurora Early Science programs, as well as at the Center for High Performance Computing of the University of Southern California. ALCF is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., "Tensorflow: a system for large-scale machine learning." *OSDI* 16 (2016): 265-83.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Neural Information Processing Systems 25 (2012): 1106–14.

Altae-Tran, Han, Bharath Ramsundar, Aneesh S. Pappu, and Vijay Pande. "Low data drug discovery with one-shot learning." ACS Central Science 3 (2017): 283-93.

Chen, J., Ma, T. and Xiao, C., 2018. "FastGCN: fast learning with graph convolutional networks via importance sampling." arXiv preprint arXiv:1801.10247.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., 2014. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078.

David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gomez-Bombarelli, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P. Adams. "Convolutional networks on graphs for learning molecular fingerprints." Advances in Neural Information Processing Systems 28 (2015): 2224–32

Grégoire Montavon, Katja Hansen, Siamac Fazli, Matthias Rupp, Franziska Biegler, Andreas Ziehe, Alexandre Tkatchenko, O. Anatole von Lilienfeld, and Klaus-Robert Müller. "Learning invariant representations of molecules for atomization energy prediction." Advances in Neural Information Processing Systems 25 (2012): 449-57.

Hansen, Katja, Franziska Biegler, Raghunathan Ramakrishnan, Wiktor Pronobis, O. Anatole von Lilienfeld, Klaus-Robert Müller, and Alexandre Tkatchenko. "Machine learning predictions of molecular properties: accurate many-body potentials and nonlocality in chemical space." The Journal of Physical Chemistry Letters 6 (2015): 2326-31.

Geim, A. K., and I. V. Grigorieva. 2013. "Van der Waals heterostructures." Nature 499 (July 24 2013):419-425. doi: 10.1038/nature12385.

Bassman, L., P. Rajak, R. K. Kalia, A. Nakano, F. Sha, J. Sun, D. J. Singh, M. Aykol, P. Huck, K. Persson, and P. Vashishta. "Active learning for accelerated design of layered materials." npj Computational Materials 4 (2018):74.

Hong, S., A. Krishnamoorthy, P. Rajak, S. Tiwari, M. Misawa, F. Shimojo, R. K. Kalia, A. Nakano, and P. Vashishta. "Computational synthesis of MoS2 layers by reactive molecular dynamics simulations: initial sulfidation of MoO3 surfaces." Nano Letters 17 (2017): 4866-72.

Kearnes, Steven, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. "Molecular graph convolutions: moving beyond fingerprints." Journal of Computer-Aided Molecular Design 30, (2016): 595-608.

Kingma, D.P. and Ba, J., 2014. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980.

Nomura, K., R.K. Kalia, A. Nakano, and P. Vashishta. "A scalable parallel algorithm for large-scale reactive force-field molecular dynamics simulations." Computer Physics Communications 178 (2008): 73-87.

Nomura, K., P. E. Small, R. K. Kalia, A. Nakano, and P. Vashishta. "An extended-lagrangian scheme for charge equilibration in reactive molecular dynamics simulations." Computer Physics Communications 192 (2015): 91-96.

Scarselli, F., M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. "The graph neural network model." IEEE Transactions on Neural Networks 20 (2009): 61-80.

Senftle, T. P., S. Hong, M. M. Islam, S. B. Kylasa, Y. Zheng, Y. K. Shin, C. Junkermeier, et al. "The Reaxff reactive force-field: development, applications and future directions." npj Computational Materials 2 (2016): 15011.

Tamayo-Mendoza, Teresa, Christoph Kreisbeck, Roland Lindh, and Alán Aspuru-Guzik. "Automatic differentiation in quantum chemistry with applications to fully variational Hartree–Fock." ACS Central Science 4 (2018): 559-66.

Wei, Jennifer N., David Duvenaud, and Alan Aspuru-Guzik. "Neural networks for the prediction of organic chemistry reactions." ACS Central Science 2 (2016): 725-32.

Xie, T., and J. C. Grossman. "Crystal Graph Convolutional Neural networks for an accurate and interpretable prediction of material properties." Physical Review Letters 120 (2018): 145301.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. "Gated graph sequence neural networks." International Conference on Learning Representations (2016).

AUTHOR BIOGRAPHIES

KUANG LIU is a PhD student in Collaboratory for Advanced Computing and Simulations at University of Southern California. His email address is liukuang@usc.edu.

KEN-ICHI NOMURA is an associate professor in Collaboratory for Advanced Computing and Simulations at University of Southern California. His email address is knomura@usc.edu.

PANKAJ RAJAK is a Postdoc in Argonne Leadership Computing Facility at Argonne National Laboratory. His email address is prajak@anl.gov.

RAJIV K. KALIA is a professor in Collaboratory for Advanced Computing and Simulations at University of Southern California. His email address is rkalia@usc.edu.

AIICHIRO NAKANO is a professor in Collaboratory for Advanced Computing and Simulations at University of Southern California. His email address is anakano@usc.edu.

PRIYA VASHISHTA is a professor in Collaboratory for Advanced Computing and Simulations at University of Southern California. His email address is priyav@usc.edu.