CS230

Deep Neural Network Driven Molecular Dynamics Simulation of Graphene

Jingchao Zhang
NVIDIA
Santa Clara, CA 95051, USA
jingchao@stanford.edu

1. Introduction

Simulation is an important approach for scientific discovery aside from experimental and theoretical methods. In the molecular simulation community, researchers have been facing the efficiency-vs-accuracy dilemma for decades.[1] Classical method such as molecular dynamics (MD) simulation is driven by empirical interatomic potentials. The atom motions are governed by Newton's law which is very fast to compute. The MD method can calculate systems with millions of atoms but lacks the accuracy to replicate real-world properties. On the other hand, the density functional theory (DFT) describes atomic systems much accurately by solving the Schrödinger equation but is very expensive to compute. Thus the DFT calculations are restricted to a few hundred atoms.

This work aims to train a deep neural network model to describe molecular systems with DFT accuracy and MD efficiency. We will exemplify this idea using the popular two-dimensional material graphene in this work.[2] A third-party package DeePMD-kit (DP), is used.[1] DeePMD-kit is a deep learning package for many-body potential energy representation and MD simulation. As shown in **Fig. 1**, *ab initio* molecular dynamics (AIMD) simulation is performed to generate training data with DFT accuracy. The feature extraction and model training is then performed using DeePMD-kit. The classical MD simulation is carried out on the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS).[3]

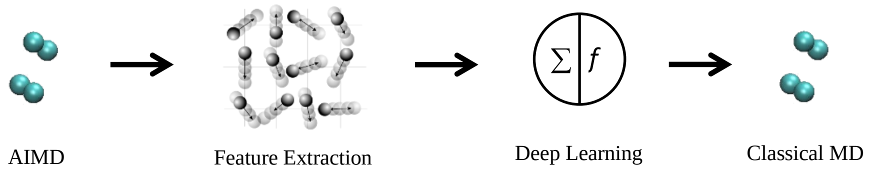


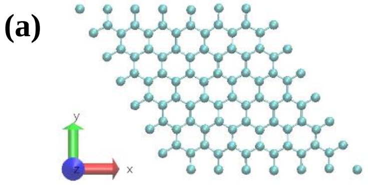
Figure 1. Schematic of the AIMD-DNN-MD workflow

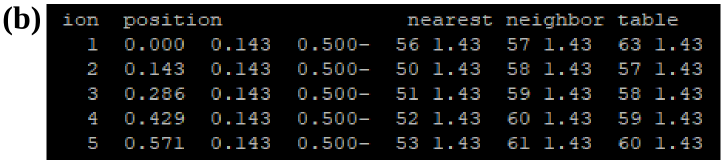
2. Related Work

Several frameworks have been developed in recent years to help train neural network models to describe molecular systems. To name a few, DeePMD-kit (DP),[1] Neuroevolutionary Potential (NEP),[4] Gaussian Approximation Potentials(GAP),[5] Moment Tensor Potential(MTP),[6] and QUantum mechanics and Interatomic Potentials(QUIP)[7]. In this work, the DP framework is selected given to its growing user community and open-source codebase. In addition, the DP code is GPU optimized, which makes it significantly faster in model training and inference.

3. Dataset and Features

To generate the AIMD training dataset, a graphene system with 98-atoms is created, as shown in **Fig. 2(a)**. The Vienna Ab initio Simulation Package (VASP)[8] is used to perform the AIMD simulation. For simulation setup, the time step is 0.5 fs (1 fs = 10^{-15} s), and the temperature is 300 K. The canonical ensemble (NVT) is used where the system temperature is controlled by a thermal bath. The projector augmented wave pseudopotentials are employed to describe the interactions between the valence electrons and frozen cores. The generalized gradient approximation exchange−correlation functional of Perdew−Burke−Ernzerhof. Only the *Γ*-point was used for *k*-space sampling. A total of 1000 steps and 0.5 ps (1 ps = 10^{-12} s) simulation is performed to extract the atomic trajectory, force and energy. The atomic trajectories are used as feature values and the force and energy are the target values.





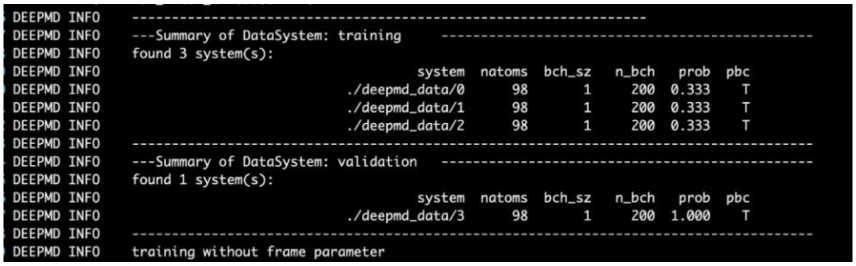
```
ion  position                      nearest neighbor table
  1  0.000   0.143   0.500-   56 1.43   57 1.43   63 1.43
  2  0.143   0.143   0.500-   50 1.43   58 1.43   57 1.43
  3  0.286   0.143   0.500-   51 1.43   59 1.43   58 1.43
  4  0.429   0.143   0.500-   52 1.43   60 1.43   59 1.43
  5  0.571   0.143   0.500-   53 1.43   61 1.43   60 1.43
```

Figure 2. (a) A graphene system with 98 atoms is used. (b) A snippet of the training data.

The generated data from the AIMD simulation need to be pre-processed by the DP plugin *dpdata*[1] before feeding into the deep neural network. The following information is needed for the model training: atom type, box boundary, coordinate, force and system energy. A snapshot, or a frame of the system, contains all these data points for all atoms at one time step. In total 1000 snapshot is taken from the AIMD simulation. For each snapshot, information such as atom trajectory, nearest neighbor list and system energy is provided for the training. A snippet of the training data is shown in **Fig. 2(b).**

4. Methods

The DP framework has three major components. The first part is a library that computes the feature vectors, forces and viral. The second part is a TensorFlow API which handles the model training and testing. The last part is an API to classical MD programs such as LAMMPS. The deep neural network is implemented using standard tensor operations. The force and viral terms can be calculated based on the system energy with respect to atomic positions and box tensors.[1] Adam optimization is used in training. Root mean square error (RMSE) is selected as the loss function. The training, validation and test dataset have proportions of 60%, 20% and 20%, respectively. As suggested by the DP document, a three-layer neural network with 25, 50 and 100 neurons in each layer is used. The start and end learning rate and decay rate are 1×10^{-3}, 3.5×10^{-8} and 5000, respectively. The training dataset is composed of three data systems, while the validation data set is composed of one data system. The number of atoms, batch size, the number of batches in the system, and the probability of using the system are all shown on the screen as shown below. The last column presents if the periodic boundary condition is assumed for the system.



```
DEEPMD INFO    ------------------------------------------------------------------
DEEPMD INFO    ---Summary of DataSystem: training      ------------------------------------------------
DEEPMD INFO    found 3 system(s):
DEEPMD INFO                                         system  natoms  bch_sz   n_bch   prob  pbc
DEEPMD INFO                                ./deepmd_data/0      98       1     200  0.333    T
DEEPMD INFO                                ./deepmd_data/1      98       1     200  0.333    T
DEEPMD INFO                                ./deepmd_data/2      98       1     200  0.333    T
DEEPMD INFO    ---------------------------------------------------------------------------------------
DEEPMD INFO    ---Summary of DataSystem: validation    ------------------------------------------------
DEEPMD INFO    found 1 system(s):
DEEPMD INFO                                         system  natoms  bch_sz   n_bch   prob  pbc
DEEPMD INFO                                ./deepmd_data/3      98       1     200  1.000    T
DEEPMD INFO    ---------------------------------------------------------------------------------------
DEEPMD INFO    training without frame parameter
```

Once the model is trained, the neural network parameters are frozen and the model is compressed. The compression step brings negligible loss to the inference accuracy but greatly speedup the inference performance in MD simulation. The speedup can be more than 10 times on both CPU and GPU devices. In addition, model compression can significantly reduce memory consumption, which can be 20 times less memory usage on the same hardware.

5. Experiments/Results/Discussion

The model is trained for 1,000,000 steps and the training results are shown in **Fig. 3(a)**. From top to bottom, **Fig. 3(a)** displays the RMSE for the overall error, decomposed error for energy, decomposed error for force and the learning rate decay schedule, respectively. It can be observed that the RMSEs decrease monotonically during training for both training and validation datasets.

The trained model is tested on the reserved test dataset and the predicted force and energy values are shown in **Fig. 3(b)**. The *x*-axes are ground truth generated from AIMD simulation and the *y*-axes are the predicted values. The inference results are distributed in the diagonal direction, which indicates high prediction performance.

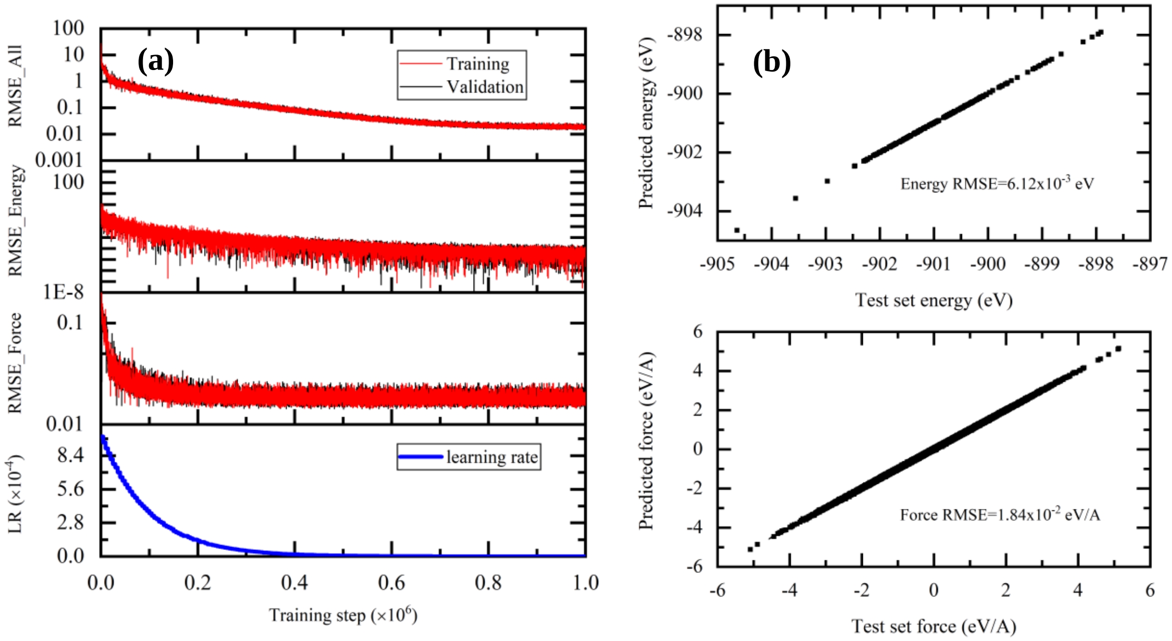


Figure 3. (a) Training loss with steps. (b) Model inference on system force and energy.

Next, the trained DP model is deployed in LAMMPS to drive the classical MD simulation using LAMMPS. First, the trained model is used to test the structural stability of the system. The relaxed graphene structures using the DP model are shown in **Fig. 4**. The displayed graphene structure is not entirely flat, which is expected due to the ripples in the cross-plane *z*-direction. Next, the DP model is used to test the thermal stability of the system. The system temperature under NVT and NVE ensembles is shown in **Fig. 5(a)**. After 10 ps NVT calculation, the system is fully relaxed under NVE condition for another 10 ps without temperature or energy controls. It can be observed that the system temperature is very stable under NVE, which indicates good thermal stability. Lastly, the radial distribution functions (RDF) are calculated to check the carbon atom distances described by DFT, DP and Tersoff empirical potential[9] respectively as shown in **Fig. 5(b)**. The DP model can accurately replicate the RDF profile of those generated from the DFT calculation.

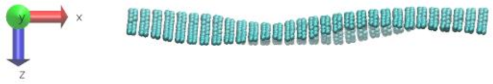


Figure 4. Atomic configuration of the graphene system after relaxation with deep potential.

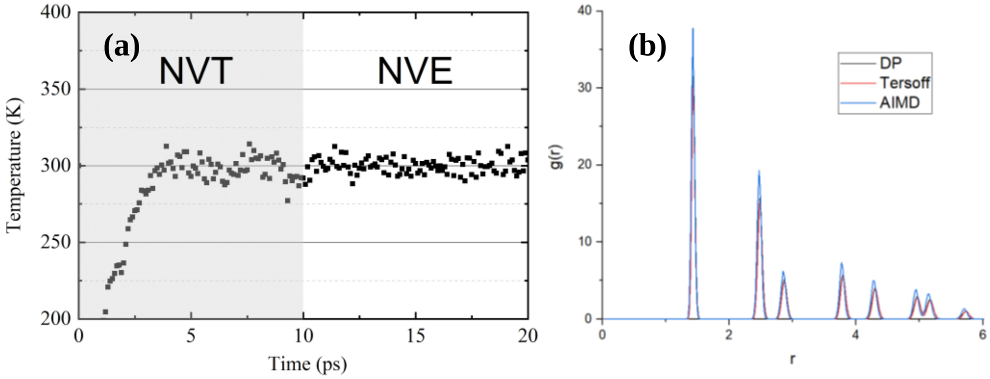


Figure 5. (a) System temperature under NVT and NVE ensembles. (b) Radial distribution function calculated by AIMD, DP and Tersoff potential, respectively.

6. Conclusion/Future work

In this work, a neural network based atomic potential is developed to describe the 2D graphene system with DFT accuracy and classical MD efficiency. The trained model can accurately predict the force and energy of the graphene system at a given temperature. Thermal stability and radial distribution function are calculated to verify the validity of the generated model. In the future, more training data can be generated using AIMD to train a more comprehensive model that can drive the MD simulation under different conditions such as temperature, strain, and surface defects.

7. Contributions

J. Z. is the sole contributor to this work. The author acknowledges the valuable discussions with CS230 TA Ruta Joshi, Yifan Li from Princeton University, Dan Han, Xinyu Wang from Shandong University, and Linfeng Zhang from DeepModelling Community.

References

[1] Wang H, Zhang L, Han J and E W 2018 DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics *Computer Physics Communications* **228** 178-84

[2] Zhang J, Huang X, Yue Y, Wang J and Wang X 2011 Dynamic response of graphene to thermal impulse *Physical Review B* **84** 235416

[3] Plimpton S 1995 Fast Parallel Algorithms for Short-Range Molecular Dynamics *Journal of Computational Physics* **117** 1-19

[4] Unke O T and Meuwly M 2019 PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges *Journal of Chemical Theory and Computation* **15** 3678-93

[5] Martinez J A, Yilmaz D E, Liang T, Sinnott S B and Phillpot S R 2013 Fitting empirical potentials: Challenges and methodologies *Curr Opin Solid St M* **17** 263-70

[6] Shchygol G, Yakovlev A, Trnka T, van Duin A C T and Verstraelen T 2019 ReaxFF Parameter Optimization with Monte-Carlo and Evolutionary Algorithms: Guidelines and Insights *J Chem Theory Comput* **15** 6799-812

[7] Behler J and Parrinello M 2007 Generalized neural-network representation of high-dimensional potential-energy surfaces *Phys Rev Lett* **98** 146401

[8] Kresse G and Hafner J 1993 Ab initio molecular dynamics for liquid metals *Physical Review B* **47** 558-61

[9] Tersoff J 1988 New empirical approach for the structure and energy of covalent systems *Physical Review B* **37** 6991-7000