Artificial Neural Network Potentials for Mechanics and Fracture Dynamics of Materials

Gang Seob Jung (jungg@ornl.gov)
Oak Ridge National Laboratory https://orcid.org/0000-0002-8047-6505
Hoon Joo Myung
Korea Institute of Science and Technology Information
Stephan Irle
Oak Ridge National Laboratory

Article

Keywords: Neural Network Potential, Graphene, Mechanics, Fracture, Crack Propagation

Posted Date: January 12th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1178290/v1

Abstract

Atomistic understanding of mechanics and failure of materials is the key for engineering and applications. Modeling accurately brittle failure with crack propagation in covalent crystals requires a quantum mechanics-based description of individual bond-breaking events for large system sizes. Machine Learned (ML) potentials have emerged to overcome the traditional, physics-based modeling tradeoff between accuracy and accessible time and length scales. Previous studies have shown successful applications of ML potentials for describing the structure and dynamics of molecular systems and amorphous or liquid phases of materials. However, their application to deformation and failure processes in materials is yet uncommon. In this study, we discuss apparent limitations of ML potentials to describe deformation and fracture under loadings and propose a way to generate and select training data for their employment in simulations of deformation and fracture of crystals. We applied the proposed approach to 2D crystal graphene, utilizing the density-functional tight-binding (DFTB) method for more efficient and extensive data generation in place of density functional theory (DFT). Then, we explore how the data selection affects the accuracy of the developed artificial neural network potential (NNP), indicating that only the errors in total energies and atomic forces are insufficient to judge the NNP's reliability. Therefore, we evaluate and select NNPs based on their performance in describing physical properties, e.g., stress-strain curves and geometric deformation. In sharp contrast to popular reactive bond order potentials, our optimized NNP predicts straight crack propagation in graphene along both armchair and zigzag lattice directions, as well as higher fracture toughness of zigzag edge direction. Our study provides significant insight into crack propagation mechanisms at atomic scales and highlights strategies for NNP developments of broader materials.

Introduction

Understanding fracture mechanics and crack propagation is key to predicting and controlling mechanical behaviors for materials processing and subsequent materials applications. In many materials, crack propagation under loading is a governing failure mechanism and, therefore, one of the critical problems in materials science. Accurate computational modeling of crack propagation, thus, becomes an essential tool for understanding the underlying mechanisms[1–3] and will open pathways for the computational design of advanced materials processing.[4]

Crack propagation in covalent crystals is a multiscale phenomenon requiring high atomic level accuracy but large-length scales for computational modeling. However, physics-based molecular simulations performed on associated atomic-scale potential energy surfaces are often limited by the fundamental tradeoff between the simulation accuracy on one hand and accessible time and length scales on the other. For example, highly accurate quantum mechanics-based electronic structure approaches such as density functional theory (DFT) are computationally demanding and unsuitable for long-timescale simulation of complex and large systems. On the other hand, classical molecular mechanics-based forcefields, including reactive forcefields, can cover larger system sizes but have difficulty for applications due to their inferior accuracy. Recently, machine learning techniques have revolutionized problem-solving

approaches in a wide range of fields of science and engineering.[5–9] Significantly, machine learned (ML) potentials have emerged as a possible solution to resolve the conflicts arising from such tradeoffs.[10]

A large number of ML-based potentials have already been proposed. Almost all of them rely on mapping or encoding the local chemical environment of atoms in a molecular or bulk system to a unique “input feature.” One of the most common approaches is the neural network potential (NNP) based on human-designed symmetry functions proposed by Behler and Parinnello (BP).[11] BP symmetry functions use translational and rotational invariant representations of atomic geometry and can be readily expanded to multiple chemical element types[12,13]. More recently, graph convolutional neural network potentials have been proposed[14] which show overall better accuracy than BP-NNPs but require a higher computational cost.[15] Another class of ML potentials, namely kernel-based approaches, offers good performance with the small number of data set, but the computational time increases with the number of reference data.[16]

While many studies have used ML potentials for various applications[10], mechanical properties related to the deformation and fracture of materials were less commonly explored. One of the critical issues to address this problem is the insufficient data set. Configurations driven by mechanical forces, *e.g.*, tensile or shear loadings, cannot be simply obtained from equilibrium molecular dynamics simulations or normal mode analysis.[12] Instead, one should perform deformation simulations until failure occurs. Another issue comes from the fact that the moment of failure or fracture is a rare event in the entire loading history, so that naturally, a data imbalance problem emerges.

A schematic of the current study is shown in Figure 1a. We developed NNP for one of the most famous two-dimensional (2D) materials, namely graphene, to describe its crack propagation and compared the results with previous experimental observations. 2D materials are ideal for validating computational models of materials failure under the effects of vacancies, bilayer, crack directions, and folding by comparing the fracture patterns observed through advanced transmission electron microscopy (TEM) techniques.[17–20] We generated data for possible fracture scenarios of graphene under various loadings by mixing two tensile and one shear deformation. The data are reduced and selected based on deformation and energy differences to improve the generalization during the training. Then, we systematically explored how the data selection affects the accuracy of the trained NNPs and the reliability evaluated based on physical properties such as deformation and stress-strain curves. In the end, we performed simulations for the crack propagation of graphene with a sharp crack using the trained NNP. We compared it with a popular empirical bond order potential for carbon systems, AIREBO.[21,22] Our results show improved agreement with previous experiments regarding the resulting edge structures and the frequency of the type of torn edges.

Results And Discussions

One of the most critical parts of ML-based potential development is the training data set. The data for training should cover the essential features of the problem-specific configurations. Previous NNPs have

been trained from the data usually generated from first principles DFT-based molecular dynamics (MD) simulations and conformal searches based on the normal mode analysis.[11,12,23] The initial training set is generally not sufficient for the desired accuracy. Therefore, adaptive and active learning approaches have been proposed and applied.[24,25] The basic principle of such methods is to detect new data not contained in the initial training data set by analyzing the data using configuration fingerprints or comparing values from multiple models of NNPs, an ensemble, or a so-called committee. Iteratively searching the new data, training, and sampling provide better data sets in the end. However, these approaches are not sufficiently good if the initial data set is not accurate enough to describe the overall dynamics during the failure.

We firstly tested previously trained models of ANI-1x[12], ANI-1cxx[26], and ANI-2x[27] provided in the TorchANI library,[28] where the data sets include the deformed geometries of small organic molecules from normal mode sampling. We test the reliability with a small graphene system composed of 24 atoms under three different loadings as shown in Figure 1b by evaluating stress-strain curves and the deformation of bond length (l_1, l_2, and l_3) and angles (θ_1, θ_2, and θ_3) described in Figure 1c. A newly developed PyTorch interface in LAMMPS was utilized for communicating energy, forces, and stress with the TorchANI python library (See Methods). Figures S1-S3 show the results of stress-strain curves and the deformation. We note that the data sets do not explicitly include graphene information, but interestingly, the NNPs from ANIs can well describe graphene's behaviors under small deformation. As expected, however, they clearly fail for fracture behaviors and large deformation.

Therefore, we designed data generation by mixing three loading directions along the *x*, *y*, and *xy* directions and generated more than 700,000 data points (See Methods). Then, we trained NNPs using the TorchANI[28] tool (See Methods). The structure of NNP is shown in Figure 2. From the actual coordinates (q), atomic environmental vectors (AEV, also known as symmetry functions, G) are utilized as input to be invariant under translation and rotational transformation and the permutation of the same atom types[11]. There are two parts in the AEV: radial and angular terms from two atoms (*i* and *j* with distance R_{ij}) and three atoms (*i*, *j*, and *k* with two distances R_{ij}, R_{ik}, and one θ_{ijk}), respectively[28]:

$$G_m^R = \sum_{j \neq i}^{allatoms} e^{-\eta \left(R_{ij} - R_s^{(k)} \right)^2} f_C\left(R_{ij}\right),$$

$$G_m^A = 2^{1-\zeta} \sum_{j,\,k \neq i}^{allatoms} \left(1 + \cos(\theta_{ijk} - \theta_s^{(q)})\right)^{\zeta} \times e^{-\eta \left(0.5\left[R_{ij} + R_{ik} \right] - R_s^{(p)} \right)^2} f_C\left(R_{ij}\right) f_C\left(R_{ik}\right),$$

where η controls the width of Gaussian function with multiple R_s for probing specified radial environments (*m* is an index for R_s); ζ controls the width of probing as η; θ_s decides the specific region in the angular environments as R_s. f_C is a cutoff function to change values to zero at R_C smoothly, defined as $f_C(R) = 0.5[\cos\frac{\pi R}{R_C} + 1]$ for $R \leq R_C$ and 0 for $R > R_C$. AEV from each atom is an input for the neural network to estimate the single atomic energy. We followed the detailed parameters of AEV, and the

structures of nodes and layers are also from ANI-2x: the radial term has 16 different radii, and the angular term has 4 angles and 8 radii[27] shown in Figure 2b. The neural network has three hidden layers with Gaussian error linear unit activation function[29] to add non-linearity.

We checked the performance of current neural network structures and training processes from the data generated from graphene's MD simulation at equilibrium states at 400 Kelvin (See method). A root mean square error (RMSE) in the relative energy lower than 0.8 kcal/mol is achieved around 200 epochs, which we consider a very high accuracy from the perspective of computational chemistry, where a threshold 1 kcal/mol in accuracy is commonly considered a "gold standard". However, our result clearly shows that an agreement only in terms of the relative energy RMSE does not guarantee correct physical properties, as shown in Figure S4. Here, we utilized the same conditions but trained NNPs with 300 epochs, and most trainings were saturated around 200 epochs.

Figure 3a shows a naïve way to save data under loading, recording data based on constant deformation (δr) or at a constant time interval. There are two obvious problems with this approach. First, it is likely to miss essential data during the failure process where the configurations drastically change in a very short time. For the second, the many similar data are close to each other near the non-deformed structures, which probably hinders the training due to the data imbalance[30]. Therefore, we utilized a constant energy difference (δE) to reduce the recorded data, as shown in Figure 3b. It would have better chances to capture the key data during the failure process even with the same number of data points. Figure 3c shows the schematic to represent the data distributions with the two main directions of the loadings: *x* and *xy*. From the total data, we built neighbor lists of each data points with a defined cutoff in the deformation space, δr. Then, the data in the neighbor list is sequentially removed if the energy difference is not bigger than the predefined criterion, δE. Table 1 lists the parameters: δr and δE with the reduced data numbers.

Table 1
The numbers of data selected from the initial 722,000 data points

δE (kcal/mol) / δr (Å)	1	2	5	10	20
0.01	450k	270k	130k	88k	80k
0.02	430k	260k	120k	66k	40k
0.05	360k	200k	100k	57k	29k
0.1	250k	150k	78k	37k	20k
0.2	120k	70k	33k	18k	9k

Table 2
Elastic constants and Poisson's ratio (ν) obtained from DFTB, reactive FFs, and trained neural network potential from the different data set selected by δr with the same δE = 1kcal/mol.

	E (GPa)	$(C_{11}-C_{12})/2$ (GPa)	ν
DFT[49]	1010		0.203
AIREBO	828.8	304.1	0.362
ReaxFF	787.2	256.2	0.536
DFTB (reference)	1108.2	451.9	0.226
δr = 0.01 Å	1168.3	471.0	0.240
δr = 0.02 Å	1173.0	461.6	0.270
δr = 0.05 Å	1192.5	474.4	0.257
δr = 0.1 Å	1141.1	452.6	0.261
δr = 0.2 Å	1132.0	453.1	0.249

Table 3
Elastic constants obtained from neural network potential from the different data set selected by δE with the same δr = 0.1Å.

	E (GPa)	$(C_{11}-C_{12})/2$ (GPa)	ν
δE = 1 kcal/mol	1141.1	452.6	0.261
δE = 2 kcal/mol	1093.2	485.6	0.126
δE = 5 kcal/mol	1112.6	445.9	0.248
δE = 10 kcal/mol	1143.6	467.0	0.224

We checked the stress-strain curves from all data without any reduction, as shown in Figure S5. As expected, it shows much better behaviors than ANIs or the model trained from NVT ensemble trajectories because the current data set explicitly includes various fracture scenarios. However, it fails to describe the fracture patterns and stress-strain after fracture along the x direction loading in Figure S5a. The energy minimization during the quasi-static loading should result in complete bond breaking between broken edges. Figures 4a and b show that the RMSE of energy and forces from training, validation, and total data from each data set selected from the above-mentioned approach as varying δr with fixed δE=1kcal/mol. We note that the data from larger δr is selected from the data set of smaller δr, which means the smaller number of data sets always belong to the larger number of data sets. Therefore, the

difference of RMSE, *e.g.*, better accuracy of δr = 0.1 Å than that of δr = 0.05Å, does not come from data themselves but from better generalization with reducing the overfitting (See SI discussion). We assume the original data set has too much similar data to prevent generalization. So, we re-evaluate the accuracy of all trained models based on the first selected data set from δr = 0.01 Å and δE = 1 kcal/mol as shown in Figures 4c and d. The RMSE of relative energy from the model (δr = 0.01 Å, δE = 1 kcal/mol) shows the lowest value, but RMSE of forces from the model (δr = 0.1 Å, δE = 1 kcal/mol) show the lowest value. Also, we investigated the stress-strain curves and deformation for the reliability of the trained NNPs. The models train from δr = 0.01 Å to δr =0.05 Å have problems near the fracture point as indicated with arrows in Figures S6 - S11. In terms of the stress-strain curves and deformation, the trained model from δr = 0.2 Å looks better than that from δr = 0.1 Å, but the model from δr = 0.1 Å was selected because the RMSE of forces exhibits the lowest value. This is a reasonable choice because the RMSE of atomic force is a good indicator for overfitting (See SI discussion). We also tested how the choice of δE affects the accuracy of models with δr lower than 0.1, as shown in Figure S12. Since the number of data decreases as the value of δE increases, it is expected to lose accuracy. However, this also does not monotonically decrease, and data with δr =0.1 Å and δE = 5 kcal/mol is reasonably optimal with the number of data points, 78,000 (=78k). Considering the fact that the reference data (~450k) is six times larger, the loss of accuracy of energy and forces are only 0.5 kcal/mol and 0.23 kcal/mol/Å, respectively. Also, the NNP describes the stress-strain and deformation under three loadings very well, as shown in Figure S13. This kind of data reduction without losing the essential data is important for active learning. However, such data augmentation is out of current scope, and we selected the model from the data (δr = 0.1 Å, δE = 1 kcal/mol) for further simulations.

A previous microscopy study reported both straight armchair (AC) and zigzag (ZZ) torn graphene edges through the high-resolution transmission electron microscope (TEM).[31] Also, torn lines along the AC edge are twice more frequently observed than the ZZ edge.[32] In previous theoretical studies, ReaxFF and AIREBO have been widely utilized to describe the mechanics and crack propagation from atomistic modeling. However, ReaxFF has some limitations in describing mechanics and stress-strain curves near failure compared to DFT calculations.[33] Especially, brittle crack propagation is hindered by the stiffening effect near the point of failure. Instead, AIREBO is more preferred because the stress-strain curves of pristine graphene are well-matched with DFT calculations once its bond order switching function is controlled.[34] MD simulations of pristine graphene with AIREBO consistently show that the fracture toughness along the zigzag edge is lower than the armchair edge[35–37]. Also, once the crack propagates along the AC direction, the fracture pattern from the AIREBO shows the zigzag torn edge preference. Although nanopores through the electron beam prefer to form the zigzag edge[38], the configuration does not come from the mechanics or crack preference but from the kinetic stability during the reconstructions.[39] This shows that empirical forcefields are limited for predicting pristine graphene's torn edge configuration and the dynamics of crack propagation.

Finally, we perform MD simulations using both selected NNP and AIREBO with a rectangular system with 10 nm x 20 nm with a crack of 2 nm, as shown in Figure 5a. We performed the crack propagation

simulations by combining quasi-static loading and dynamic loading to reduce the computational cost (See method). The NNP results in a straight and clean torn edge in both AC and ZZ crack direction in Figure 5b (See Movie 1 and 2). As described above, AIREBO predicts that the straight propagation along the AC direction is less likely to occur. Instead, it shows meandering crack paths in Figure 5c (See Movie 3 and 4) with ZZ crack edges. The obtained stress-strain curves from NNP and AIREBO are shown in Figure 5d. The notable difference is the critical stress, which is proportional to the fracture toughness under the same geometric and boundary conditions. The NNP predicts the fracture toughness along the AC-edge is lower than ZZ-edge. While AIREBO predicts very similar critical stress with both edges, and critical strain along the AC-Edge is longer than ZZ-Edge. Considering critical energy release rate, AIREBO predicts the fracture toughness along the AC-edge is higher than ZZ-edge, which is completely opposite to the results of the NNP.

The frequency of torn edges in the suspended polycrystalline graphene monolayer depends on the fracture toughness of pristine graphene. The prediction from the NNP shows well agreement with those previous observations, while AIREBO predicts opposite behaviors in terms of frequency of torn edge observation and torn AC edge configuration. The limitation of AIREBO comes from the softened angle stiffness under tensile loading, which also has been compared with DFT and DFTB calculations in the previous study.[40] Figure 6 shows the bond lengths and angle of graphene at the crack tip just after the first bond breaking during the crack propagation. The angular deformation of NNP shows a lower angle (~124°) than that of AIREBO (133°), which results in the elongation of the inner side bond length, l_2 (~1.7 Å) than the outer side bond length, l_3 (~1.6 Å). The relative bond lengths between l_2 and l_3 determine the crack path, and AIREBO prefers ZZ crack paths because l_3 (1.72 Å) is longer than l_2 (1.67 Å). In another 2D material, WS_2, these lattice distortions can result in anisotropic crack dynamics even with the same surface energy[19].

We wish to emphasize that the current NNPs are limited as they should only be used for the simulation of stress-induced fracture and failure of pristine graphene. The training data set does not explicitly have failure dynamics of bilayer graphene, diamond, amorphous carbon network, carbon nanotube, graphyne, grain boundary, vacancies, and folding, etc. Therefore, new data should be generated and tested for the new applications. However, the neural network potential is useful in computational speed compared to *first principles*-based electronic structure approaches. Also, the NNP is very flexible to capture non-linear deformation-stress behaviors well, which is challenging with the fixed functional form in classical forcefields. Crack dynamics is one of the exciting applications for NNPs due to its intrinsic multiscale feature. In this study, we only focus on the data generation and selection from the mixed loading and data reduction using DFTB as the reference method. However, the selected data can be utilized for high throughput calculation with more accurate methods. Also, active learning and transfer learning from the selected data would be interesting topics in the future.

Conclusion

In summary, we propose generating and selecting down the training data for deformation and fracture of 2D crystal, graphene, under mixed loading conditions through DFTB calculations. We utilize the previously developed PyTorch library, TorchANI, to train the models of the Behler-Parrinello type's neural network potential. For the molecular dynamics, we developed a PyTorch interface with LAMMPS, which can be expanded to other machine learning potential libraries through PyTorch. The proposed data reduction process improves the generalization of NNP training by mitigating overfitting. We show that the low RMSEs of energy and force do not automatically guarantee the physical properties of the trained models. The selected model considering physical properties can describe the torn edge configuration observed in the previous studies and explains well the higher frequency of torn AC edge occurrence, which is not possible with the reactive FFs. The proposed work frame can be applied to understand fracture dynamics of 2D and bulk crystals using neural network potentials.

Methods

Generation of data

The molecular dynamics simulations with DFTB for data generation were performed *via* the LAMMPS package.[41] DFTB calculations were performed at each time step through the DFTB+ package,[42] utilizing a previously developed interface for LAMMPS.[40] We employed the 3OB[43] C-C parameters with the DFTB3 scheme because the stress-strain behaviors are well-matched with those from DFT calculations with the PBE functional.[40] We generated 10,000 data points through the NVT ensemble at 400K with the time step of 0.5 fs to obtain reference RMSE of the relative energy from the canonical data generation.

Then, we obtained data points for the deformation and fracture under various loading. Static loading or quasi-static loading with full energy minimization has a limitation for the training data set because all atoms are located in the energy minimized positions. We need slightly perturbated coordinates to train the NNP to distinguish the contribution of a single atomic energy. Therefore, we utilized dynamics loading for the data generation. We consider different loading directions by mixing the loadings along *x*, *y* (pure tensile), and *xy* (pure shearing) directions as (v_x, v_y, v_{xy}). We prepared 361 directions with constant velocity 400m/s with 0.5 fs time step. Each direction has 2000 steps, so the total data number is 722,000. Here, the loading speed is much faster than what is desired to provide reliable behaviors, which is under 20m/s[40]. The effect of the loading speed becomes critical when the speed is too fast for the system to have enough time to relax the structures under the deformation. So, after every 20 steps, we included a small step of energy minimization to overcome the delay. The number of steps was tuned to match the stress-strain curve under shear loading to the results from the quasi-static loading. At each step, we deformed the simulation box by about 0.002 Å, resulting in a total deformation for each loading of about 4 Å.

Selection of data

From the 722,000 data points, we built neighbor lists between data points based on the deformation of the simulation box (*dx, dy, dxy*). We consider the distance (δr) between data as an indicator of the deformation similarity. Figure 3c shows a schematic of generated data points in the deformation space (*dx*, *dxy*). Then, we sequentially deleted the data but saved it if the energy difference in the list is larger than δE. We utilized vales of δr (0.01 Å ~ 0.2 Å) and δE (1 kcal/mol ~ 50 kcal/mol) from the original 722,000 data points, and the number of reduced data from (δr, δE) is listed in Table 1.

Training

We utilized TorchANI library and its setting for training the neural networks. For training, 80% of data was used, while 20% of data was utilized for validation with a small mini-batch size of 64. We basically follow the training process suggested in TorchANI.[28] The loss function is defined as

$$Loss = \frac{1}{N_{data}} \sum \frac{\left(E_{NNP} - E_{ref}\right)^2}{\sqrt{N_{atom}}} + \frac{\alpha}{N_{data}} \sum \frac{\left(\vec{F}_{NNP} - \vec{F}_{ref}\right)}{N_{atom}}$$

,

where α is a parameter to determine the contribution of forces, and we used 0.1. The Adam optimizer was utilized with weight decay for the weights[44,45] and stochastic gradient descent[46] for the biases, as suggested in the previous study. The weights were initialized by Kaiming initialization[47] with the normal distribution, and the initial values of biases were zeros.

Molecular dynamics simulations for crack propagation

We prepared the pristine graphene of 10 nm by 20 nm with a 2 nm sharp crack along the y direction with periodic boundary conditions in Figure 5a. To avoid the interaction between image cells, 20 nm space along the y direction and 3.35 nm space along the z direction were inserted. Instead of full dynamic loading, we applied a 0.01 strain at each iteration to stretch to a 0.04 strain along the x direction with structural relaxations. Then, we applied dynamic tensile loading along the x direction at low temperature, 10 Kelvin. The loading speed was 2.0 m/s, and the time step was set to 1 fs.

PyTorch interface with LAMMPS

We utilize the python functions in LAMMPS (v. 29OCT20) to utilize the python code in the python environment. Through the python environment, we can easily call python library. Utilizing TorchANI, we calculate the AEV for neural networks. Forces and stress are calculated with the given coordinates and simulation box through the autograd engine in PyTorch[48] and updated at each time step.

Declarations

ORCID

Gang Seob Jung: 0000-0002-8047-6505

Stephan Irle: 0000-0003-4995-4991

COMPETING INTERESTS

The authors declare no Competing Financial or Non-Financial interests.

DATA AVAILIBILTY

All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supporting Information. Extra data and machine-readable data are available upon reasonable request to the authors.

CODE AVAILIBILITY

Sample inputs and source codes to use PyTorch with LAMMPS are shared in http://github.com/gsjung0419/LammpsTorch.

AUTHOR CONTRIBUTION STATEMENTS

G.S.J. conceived the idea, developed codes, and performed experiments. G.S.J. and H.J.M optimized codes. G.S.J. performed DFTB calculations under supervision of S.I. G.S.J, H.J.M, and S.I involved in writing and editing the manuscript.

ACKNOWLEDGEMENTS

The authors acknowledge helpful comments and discussions with J. Warner and R. Khanal. G.S.J. acknowledges support for method developments by the Laboratory Directed Research and Development (LDRD) Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the US Department of Energy under contract DEAC05-00OR22725 (Eugene P. Wigner Fellowship). S.I. acknowledges partial support from the Artificial Intelligence Initiative as part of the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory. This research used resources of the Compute and Data Environment for Science (CADES) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. G.S.J. and S.I. acknowledge support for parts of data generation and training models by the U.S. Department of

Energy Fossil Energy and Carbon Management Program, Advanced Coal Processing Program, C4WARD project (FWP No. FEAA155).

References

1. Buehler, M. J. & Gao, H. Dynamical fracture instabilities due to local hyperelasticity at crack tips. *Nature* **439**, 307-310, doi:10.1038/nature04408 (2006).
2. Kermode, J. R.*et al.* Low-speed fracture instabilities in a brittle crystal. *Nature* **455**, 1224-1227, doi:10.1038/nature07297 (2008).
3. Livne, A., Bouchbinder, E., Svetlizky, I. & Fineberg, J. The Near-Tip Fields of Fast Cracks. *Science* **327**, 1359, doi:10.1126/science.1180476 (2010).
4. Wang, H.*et al.* Frank-van der Merwe growth in bilayer graphene. *Matter* **4**, 3339-3353, doi:https://doi.org/10.1016/j.matt.2021.08.017 (2021).
5. Jumper, J.*et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583-589, doi:10.1038/s41586-021-03819-2 (2021).
6. Schmidt, J., Marques, M. R. G., Botti, S. & Marques, M. A. L. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials* **5**, 83, doi:10.1038/s41524-019-0221-0 (2019).
7. Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547-555, doi:10.1038/s41586-018-0337-2 (2018).
8. Pfau, D., Spencer, J. S., Matthews, A. G. D. G. & Foulkes, W. M. C. Ab initio solution of the many-electron Schr\"odinger equation with deep neural networks. *Physical Review Research* **2**, 033429, doi:10.1103/PhysRevResearch.2.033429 (2020).
9. Vamathevan, J.*et al.* Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery* **18**, 463-477, doi:10.1038/s41573-019-0024-5 (2019).
10. Friederich, P., Häse, F., Proppe, J. & Aspuru-Guzik, A. Machine-learned potentials for next-generation matter simulations. *Nature Materials* **20**, 750-761, doi:10.1038/s41563-020-0777-6 (2021).
11. Behler, J. & Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters* **98**, 146401, doi:10.1103/PhysRevLett.98.146401 (2007).
12. Smith, J. S., Isayev, O. & Roitberg, A. E. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chemical Science* **8**, 3192-3203, doi:10.1039/C6SC05720A (2017).
13. Behler, J. Constructing high-dimensional neural network potentials: A tutorial review. *International Journal of Quantum Chemistry* **115**, 1032-1050, doi:https://doi.org/10.1002/qua.24890 (2015).
14. Schütt, K. T.*et al.* SchNetPack: A Deep Learning Toolbox For Atomistic Systems. *Journal of Chemical Theory and Computation* **15**, 448–455, doi:10.1021/acs.jctc.8b00908 (2018).
15. Schütt, K. T.*et al.* SchNetPack: A Deep Learning Toolbox For Atomistic Systems. *Journal of Chemical Theory and Computation* **15**, 448-455, doi:10.1021/acs.jctc.8b00908 (2019).

16. Chmiela, S.*et al.* Machine learning of accurate energy-conserving molecular force fields. *Science Advances* **3**, e1603015, doi:10.1126/sciadv.1603015.
17. Wang, S.*et al.* Atomically Sharp Crack Tips in Monolayer MoS2 and Their Enhanced Toughness by Vacancy Defects. *ACS Nano* **10**, 9831-9839, doi:10.1021/acsnano.6b05435 (2016).
18. Jung, G. S.*et al.* Interlocking Friction Governs the Mechanical Fracture of Bilayer MoS2. *ACS Nano* **12**, 3600-3608, doi:10.1021/acsnano.8b00712 (2018).
19. Jung, G. S.*et al.* Anisotropic Fracture Dynamics Due to Local Lattice Distortions. *ACS Nano* **13**, 5693-5702, doi:10.1021/acsnano.9b01071 (2019).
20. Ryu, G. H., Jung, G. S., Park, H., Chang, R.-J. & Warner, J. H. Atomistic Mechanics of Torn Back Folded Edges of Triangular Voids in Monolayer WS2. *Small* **n/a**, 2104238, doi:https://doi.org/10.1002/smll.202104238 (2021).
21. Stuart, S. J., Tutein, A. B. & Harrison, J. A. A reactive potential for hydrocarbons with intermolecular interactions. *The Journal of Chemical Physics* **112**, 6472-6486, doi:10.1063/1.481208 (2000).
22. Brenner, D. W.*et al.* A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons. *Journal of Physics: Condensed Matter* **14**, 783-802, doi:10.1088/0953-8984/14/4/312 (2002).
23. Wang, H., Zhang, L., Han, J. & E, W. DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics. *Computer Physics Communications* **228**, 178-184, doi:https://doi.org/10.1016/j.cpc.2018.03.016 (2018).
24. Smith, J. S., Nebgen, B., Lubbers, N., Isayev, O. & Roitberg, A. E. Less is more: Sampling chemical space with active learning. *The Journal of Chemical Physics* **148**, 241733, doi:10.1063/1.5023802 (2018).
25. Botu, V. & Ramprasad, R. Adaptive machine learning framework to accelerate ab initio molecular dynamics. *International Journal of Quantum Chemistry* **115**, 1074-1083, doi:https://doi.org/10.1002/qua.24836 (2015).
26. Smith, J. S.*et al.* Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nature Communications* **10**, 2903, doi:10.1038/s41467-019-10827-4 (2019).
27. Devereux, C.*et al.* Extending the Applicability of the ANI Deep Learning Molecular Potential to Sulfur and Halogens. *Journal of Chemical Theory and Computation* **16**, 4192-4202, doi:10.1021/acs.jctc.0c00121 (2020).
28. Gao, X., Ramezanghorbani, F., Isayev, O., Smith, J. S. & Roitberg, A. E. TorchANI: A Free and Open Source PyTorch-Based Deep Learning Implementation of the ANI Neural Network Potentials. *Journal of Chemical Information and Modeling* **60**, 3408-3415, doi:10.1021/acs.jcim.0c00451 (2020).
29. Hendrycks, D. & Gimpel, K. Gaussian Error Linear Units (GELUs). *arXiv [cs.LG]* (2020).
30. Fernandez, A. G. a. S. G. M. P. R. C. K. B. H. F. Learning from Imbalanced Data Sets. (2018).

31. Kim, K.*et al.* Atomically perfect torn graphene edges and their reversible reconstruction. *Nature Communications* **4**, 2723, doi:10.1038/ncomms3723 (2013).
32. Kim, K.*et al.* Ripping Graphene: Preferred Directions. *Nano Letters* **12**, 293-297, doi:10.1021/nl203547z (2012).
33. Jensen, B. D., Wise, K. E. & Odegard, G. M. Simulation of the Elastic and Ultimate Tensile Properties of Diamond, Graphene, Carbon Nanotubes, and Amorphous Carbon Using a Revised ReaxFF Parametrization. *The Journal of Physical Chemistry A* **119**, 9710-9721, doi:10.1021/acs.jpca.5b05889 (2015).
34. Yeo, J.*et al.* Multiscale Design of Graphyne-Based Materials for High-Performance Separation Membranes. *Advanced Materials* **31**, 1805665, doi:10.1002/adma.201805665 (2019).
35. Dewapriya, M. A. N. & Rajapakse, R. K. N. D. Molecular Dynamics Simulations and Continuum Modeling of Temperature and Strain Rate Dependent Fracture Strength of Graphene With Vacancy Defects. *Journal of Applied Mechanics* **81**, doi:10.1115/1.4027681 (2014).
36. Jung, G., Qin, Z. & Buehler, M. J. Molecular mechanics of polycrystalline graphene with enhanced fracture toughness. *Extreme Mechanics Letters* **2**, 52-59, doi:https://doi.org/10.1016/j.eml.2015.01.007 (2015).
37. Zhang, T., Li, X. & Gao, H. Fracture of graphene: a review. *International Journal of Fracture* **196**, 1-31, doi:10.1007/s10704-015-0039-9 (2015).
38. Girit, Ç. Ö.*et al.* Graphene at the Edge: Stability and Dynamics. *Science* **323**, 1705-1708, doi:doi:10.1126/science.1166999 (2009).
39. Govind Rajan, A.*et al.* Addressing the isomer cataloguing problem for nanopores in two-dimensional materials. *Nature Materials* **18**, 129-135, doi:10.1038/s41563-018-0258-3 (2019).
40. Jung, G. S. a. I., Stephan and Sumpter, Bobby. Dynamic Aspects of Graphene Deformation and Fracture from Approximate Density Functional Theory. *Available at SSRN:https://ssrn.com/abstract=3933565*doi:http://dx.doi.org/10.2139/ssrn.3933565
41. Plimpton, S. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *Journal of Computational Physics* **117**, 1-19, doi:https://doi.org/10.1006/jcph.1995.1039 (1995).
42. Hourahine, B.*et al.* DFTB+, a software package for efficient approximate density functional theory based atomistic simulations. *The Journal of Chemical Physics* **152**, 124101, doi:10.1063/1.5143190 (2020).
43. Gaus, M., Goez, A. & Elstner, M. Parametrization and Benchmark of DFTB3 for Organic Molecules. *Journal of Chemical Theory and Computation* **9**, 338-354, doi:10.1021/ct300849w (2013).
44. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv [cs.LG]* (2017).
45. Loshchilov, I. & Hutter, F. Decoupled Weight Decay Regularization. *arXiv [cs.LG]* (2019).
46. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609. 04747* (2016).

47. He, K., Zhang, X., Ren, S. & Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv [cs.CV]* (2015).
48. Paszke, A.*et al.* in *NIPS 2017 Workshop on Autodiff* (2017).
49. Lebedeva, I. V., Minkin, A. S., Popov, A. M. & Knizhnik, A. A. Elastic constants of graphene: Comparison of empirical potentials and DFT calculations. *Physica E: Low-dimensional Systems and Nanostructures* **108**, 326-338, doi:https://doi.org/10.1016/j.physe.2018.11.025 (2019).

Figures

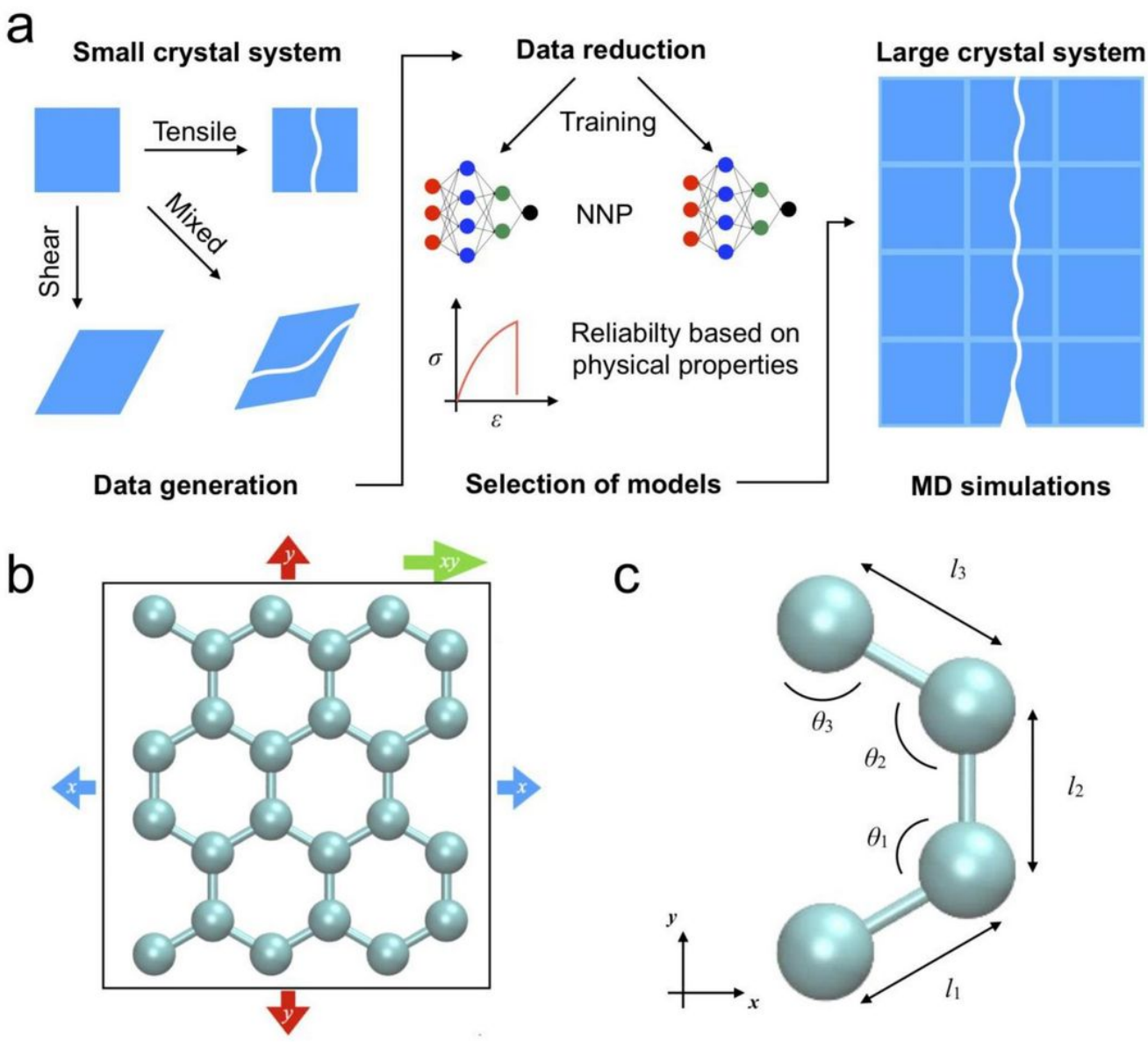


Figure 1

(**a**) Schematic of current study from the data generation to MD simulations. Data are generated for possible fracture scenarios of graphene with a small size. The data are reduced and selected based on physical properties. The selected NNP is utilized for the crack propagation of graphene with a sharp crack. (**b**) Small crystal system of 24 atoms under three main loadings (**c**) Definition of three different bond lengths and angles to evaluate the deformation under loadings.

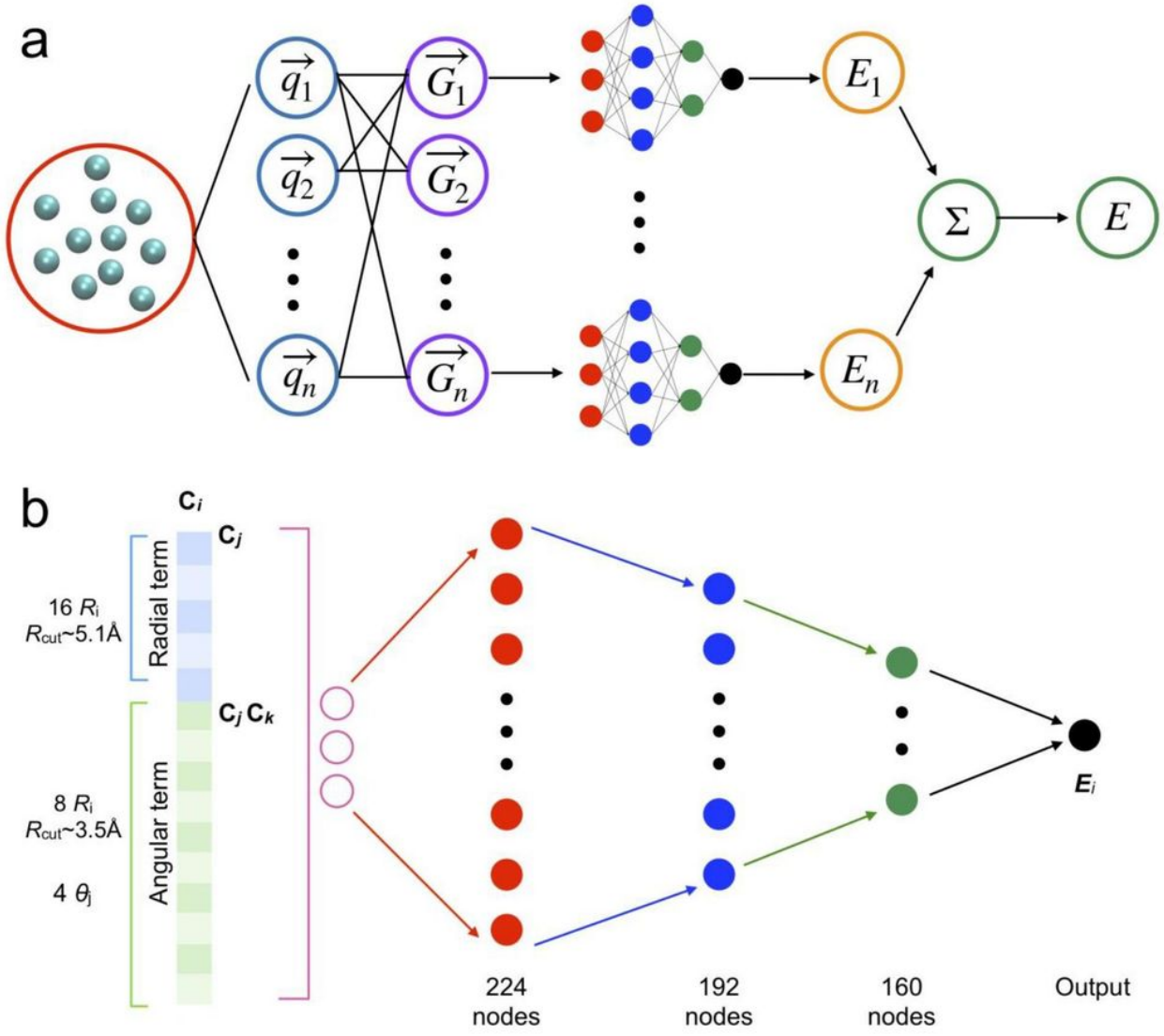


Figure 2

(**a**) Structure of neural network potential based on the symmetry functions or atomic environmental vector (AEV). AEV (***G***) is calculated from the three-dimensional atomic coordinates (***q***) and used as an input for the neural network to obtain atomic energy. (**b**) Dimension of AEV with radial and angular terms from carbon system and details of neural network structure with layers and nodes.

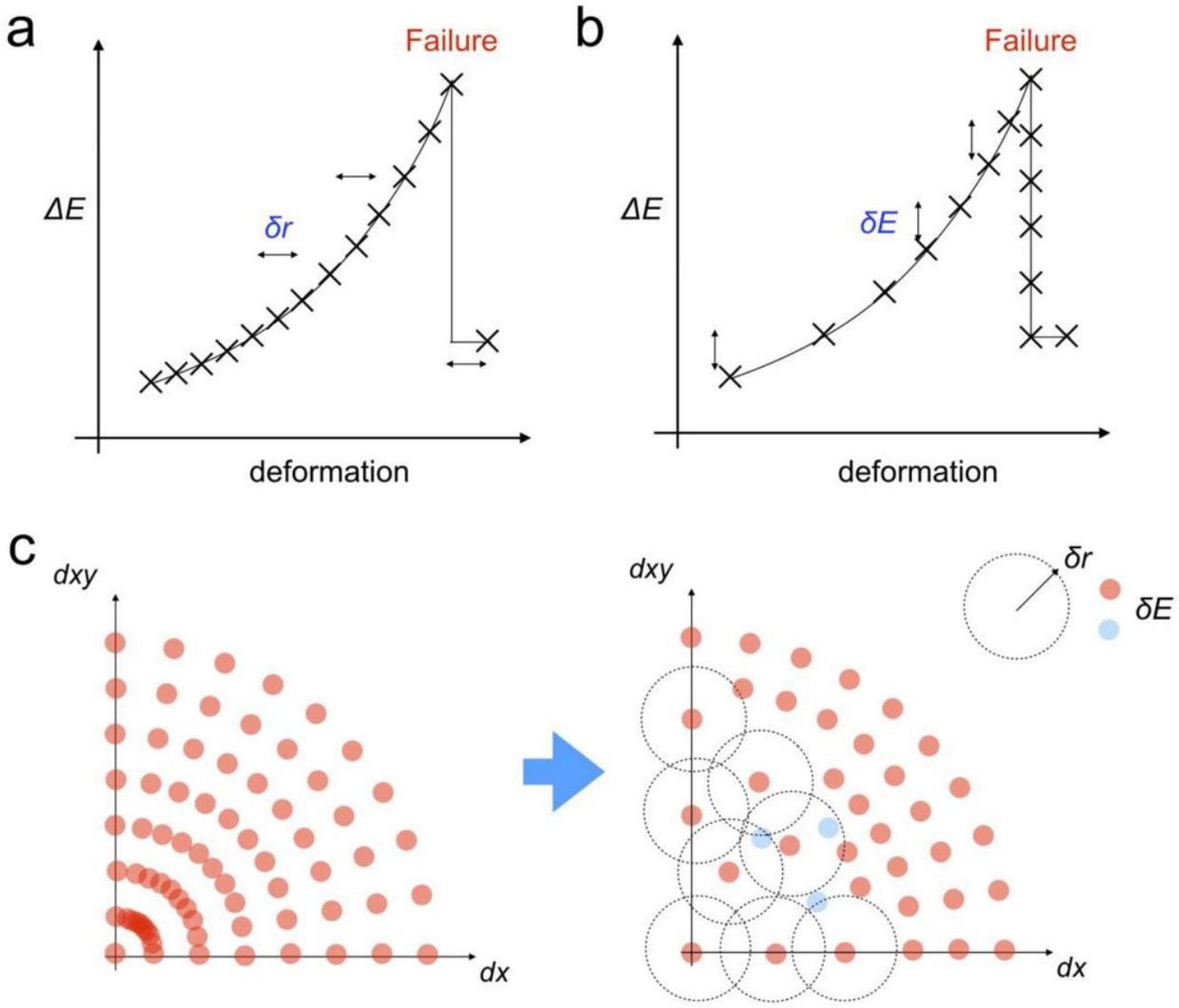


Figure 3

(**a, b**) Schematics of data distribution under loading with *δr* and *δE*. Data with a constant deformation or space (δr), it would lose the important data of the failure. Instead, energy difference (*δE*) based data selection will conserve the key data and reduce data near the equilibrium state. (**c**) Example of data distributions with the two loading directions to show a practical method to achieve data selection based on *δE*. First, neighbor lists are built with a defined cut-off, *δr*. The data of which the energy difference is smaller than *δE* are deleted from the list.

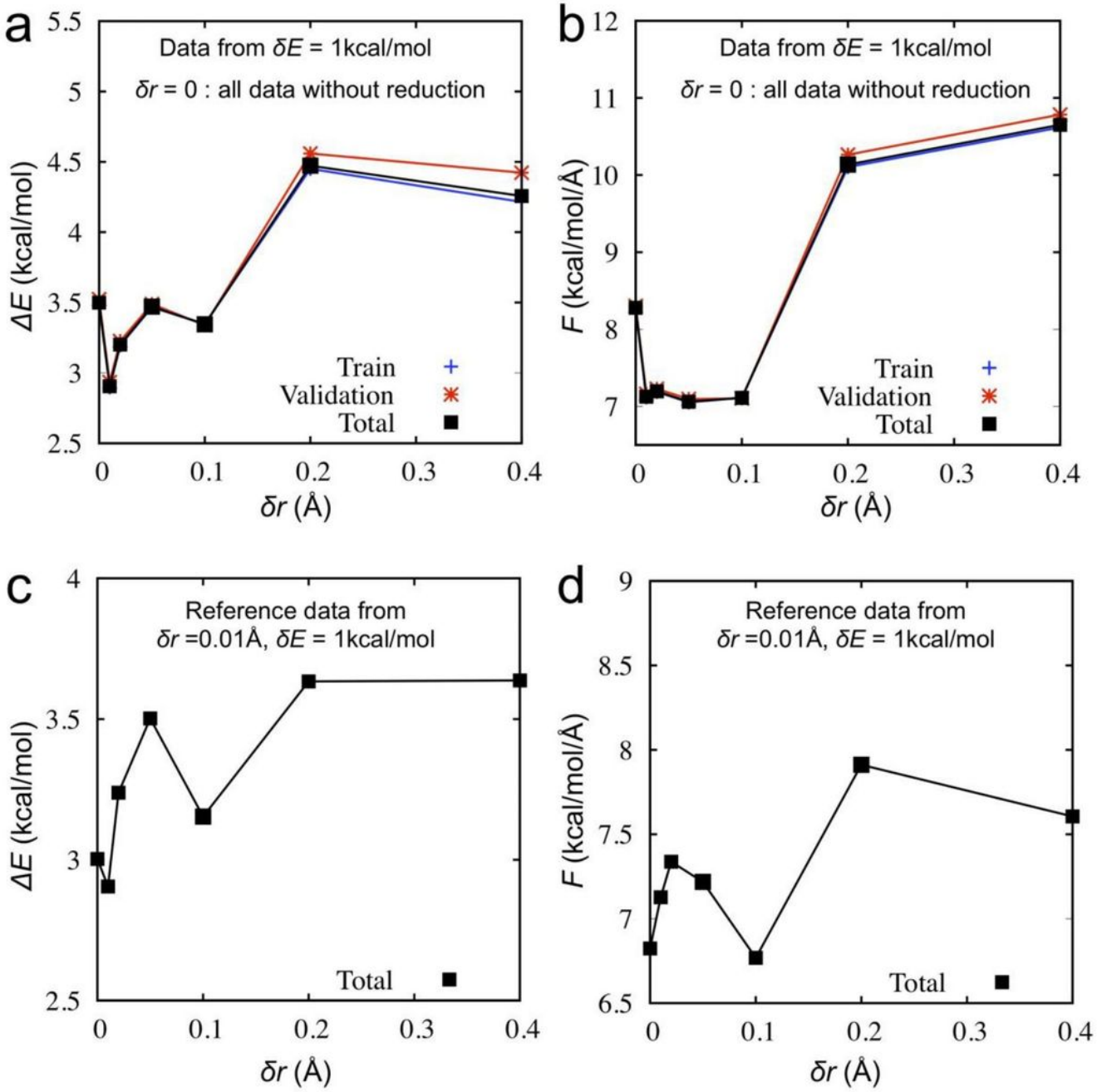


Figure 4

RMSE of relative energy (**a**) and forces (**b**) from training, validation, and total data from each data set selected from *δr* with fixed *δE*=1kcal/mol. δr = 0 indicates the entire data points without reduction. Reevaluation RMSE of relative energy (**c**) and forces (**d**) with the same data set (*δr* = 0.01 Å, *δE* = 1kcal/mol). RMSEs depend on data set, and even smaller numbers of data can have higher accuracy. Also, RMSE does not guarantee the physical properties of the models.

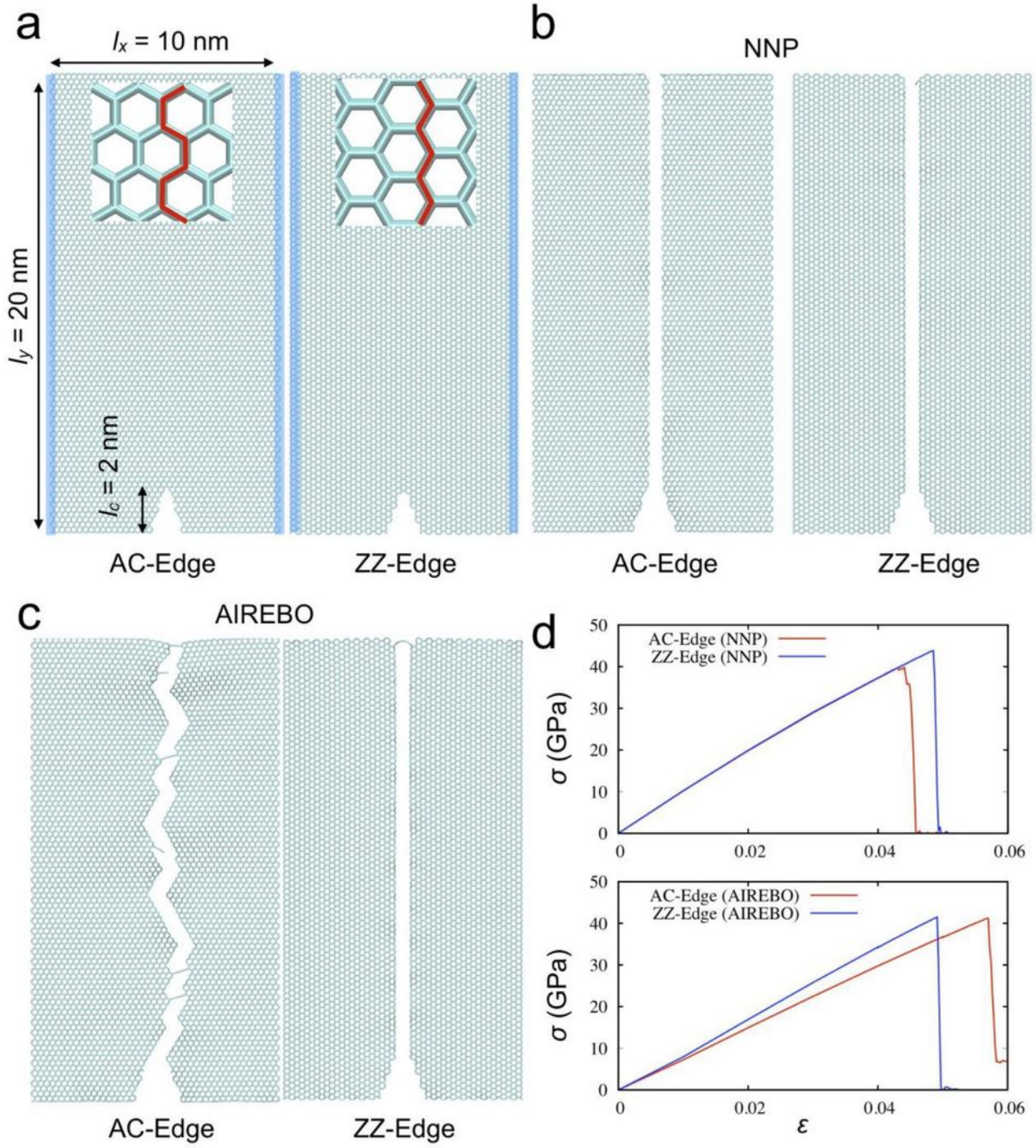
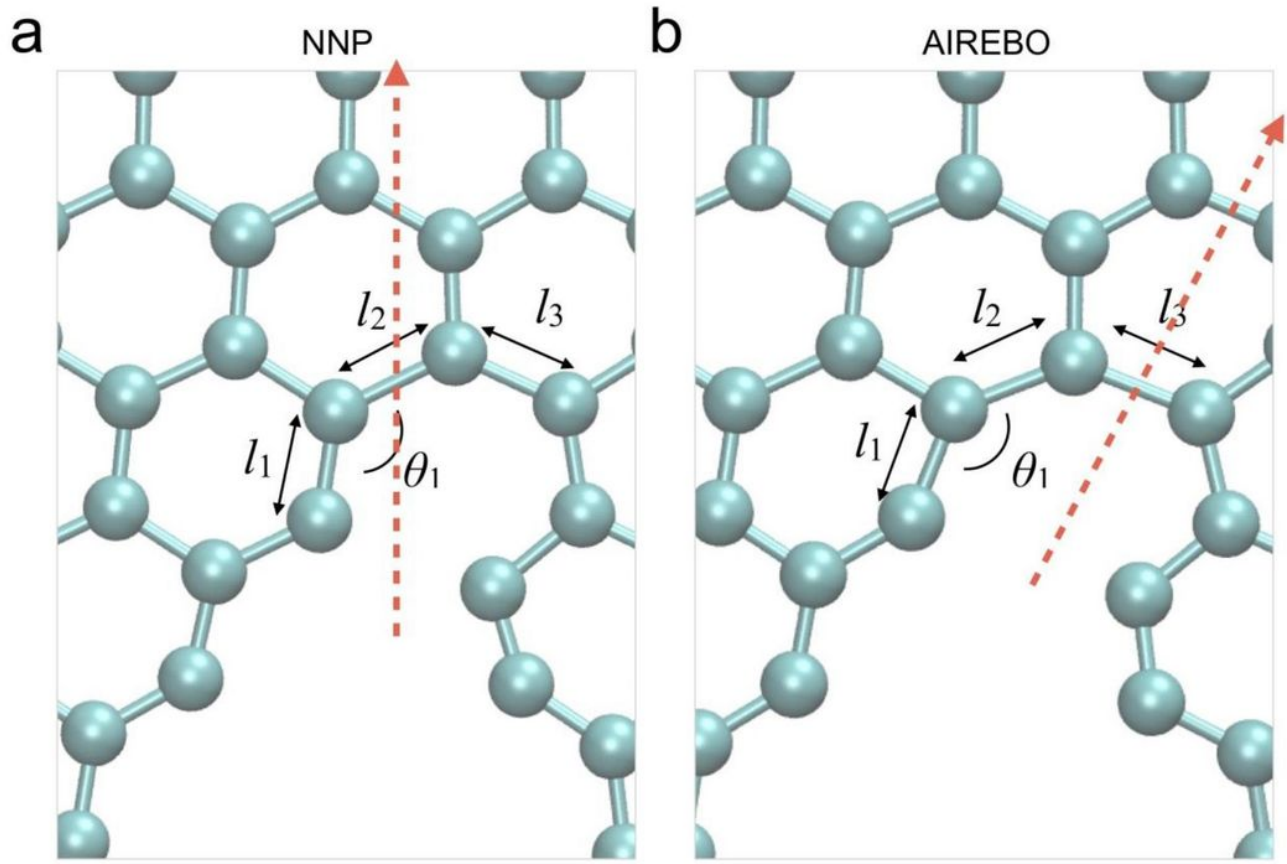


Figure 5

(**a**) Initial geometries of crack inserted graphene along the armchair and zigzag edges. (**b**) Fracture pattern of NNP after crack propagations (See Movie 1 and 2). (**c**) Fracture pattern of NNP after crack propagations (See Movie 3 and 4). (**d**) The obtained stress-strain curves from NNP and AIREBO.



	l_1	l_2	l_3	θ_1
NNP	1.41 Å	1.71 Å	1.58 Å	124.1°
AIREBO	1.47 Å	1.67 Å	1.72 Å	133.0°

Figure 6

Bond and angle at the crack tip after the first bond breaking along the AC edge direction from NNP (**a**) and AIREBO (**b**). Dotted red arrows represent the crack propagation. The angular deformation of trained NNP shows a lower angle than that of AIREBO, which results in the elongation of the inner side bond length, l_2, than the outer side bond length, l_3, from NNP. AIREBO shows propagation along the zigzag edge because l_3 is longer than l_2.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- 4.AIREBOzz.mpg
- 1.NNPac.mpg
- 3.AIREBOac.mpg
- 2.NNPzz.mpg
- SupplementaryInformation.docx