ENHANCED FORCE FIELD CONSTRUCTION FOR GRAPHENE MONOLAYERS VIA NEURAL NETWORK-BASED FITTING OF DENSITY FUNCTIONAL THEORY DATA

XÂY DỰNG TRƯỜNG LỰC NÂNG CAO CHO CÁC LỚP ĐƠN GRAPHENE BẰNG CÁCH ĐIỀU CHỈNH DỮ LIỆU LÝ THUYẾT HÀM MẬT ĐỘ DỰA TRÊN MẠNG NƠ-RON

Tan-Tien Pham[1], Tien B. Tran[2], Viet Q. Bui[1]*

[1]The University of Danang - Advanced Institute of Science and Technology, Vietnam
[2]The University of Danang, Vietnam

*Corresponding author: bqviet@ac.udn.vn

(Received: May 23, 2024; Revised: July 22, 2024; Accepted: September 24, 2024)

Abstract - This study presents a novel neural network (NN) framework for developing force fields specific to graphene monolayers, utilizing data obtained from first-principles calculations. The authors analyze three primary force components, force magnitude and the cosines of two angles across different configurations of surrounding carbon atoms. Initially, the NN applied to the three nearest neighbors, achieving average absolute testing errors of 0.375 eV/Å, 0.092, and 0.085 for the respective components. Then, expanding the input variables to nine surrounding atoms, which significantly enhances the precision of the force field models, reducing the error in force magnitude to approximately 1%. This improvement represents a 33% to 59% increase in accuracy over the initial method. The results demonstrate the potential of NNs to generate highly accurate force fields for graphene.

Keywords - Graphene; force field; neural network $\left|\vec{F}\right|$.

Tóm tắt - Nghiên cứu này giới thiệu một mô hình mạng nơ-ron (NN) mới để phát triển trường lực được thiết kế cho các lớp đơn graphene, sử dụng dữ liệu được tính toán từ nguyên lý thứ nhất. Ba thành phần lực chính được phân tích gồm: độ lớn của lực và cosin của hai góc được đo trên các cấu hình khác nhau của các nguyên tử carbon xung quanh. Ban đầu, áp dụng NN cho ba nguyên tử lân cận gần nhất, với các sai số kiểm tra tuyệt đối trung bình lần lượt là 0,375 eV/Å, 0,092 và 0,085 cho các thành phần tương ứng. Sau đó, mở rộng các biến đầu vào bao gồm chín nguyên tử xung quanh, điều này đã cải thiện đáng kể độ chính xác của mô hình trường lực, giảm sai số độ lớn lực xuống còn khoảng 1%. Sự cải thiện này tương đương với mức tăng độ chính xác từ 33% đến 59% so với phương pháp ban đầu. Kết quả cho thấy, tiềm năng của NN trong việc tạo ra các trường lực có độ chính xác cao cho graphene.

Từ khóa - Graphene; force field; neural network $\left|\vec{F}\right|$.

1. Introduction

In the past decade, significant interest has been devoted to study two-dimensional materials, especially graphene [1]. Such material continues to highly attract attention due to its unique thermodynamic and electronic properties. In this honeycomb structure, carbon atoms are linked together by sp^2-hybridized bonds, which provide remarkable mechanical stability and unusually high thermal conductivity at the nanoscale [2, 3]. Therefore, investigating the thermal properties of graphene is considered an important and challenging task, which has been addressed using various experimental and theoretical approaches. In this study, our objective is to develop a force-field (FF) acting on a C atom infinite graphene monolayer by adopting an efficient numerical fitting strategy.

Density Functional Theory (DFT) has been extensively utilized to calculate electronic structure and properties of materials due to its balance between accuracy and computational efficiency. DFT allows for the determination of total energies, electronic densities, and forces acting on atoms, which are crucial for developing reliable force fields. By employing DFT calculations, the authors can accurately capture the interactions within the graphene monolayer, providing a solid foundation for constructing precise neural network-based force fields.

In reality, graphene is hardly found in its equilibrium ground state, and the force acting on each C atom is different from case to case, which highly depends on the environment of interactions with other surrounding C atoms. In order to clearly understand the heat spreading process in graphene, molecular dynamics simulations should be executed with a reliable FF. However, using ab initio FF in direct Born-Oppenheimer dynamics is too computationally demanding and thus unrealistic. Significant efforts have been made to construct FFs for graphene using empirical approaches, such as using Tersoff's valence force model to describe sp^2 interactions [4], followed by Monte Carlo simulations to estimate important thermodynamic quantities such as Young's modulus and molar heat capacity [5]. Recently, efforts have been made to construct an in-plane FF for graphene based on first-principles calculation data [6]. In this study, the authors present a new approach to interpolate FFs for graphene by employing the neural network (NN) technique [7] for direct force fitting.

As a powerful and robust tool for function fitting with high accuracy, for years, artificial NN has been widely applied to develop Potential Energy Surfaces (PES) by fitting electronic structure data [8, 9, 10]. The most obvious disadvantage of the NN method is the requirement of large dataset for fitting processes. Recently, there have been contributions to reduce the amount of required data by developing a function-gradient simultaneous fitting procedure, which is termed combined-function-derivative approximation [11]. Also, the NN architecture was modified to allow the direct permutation of atoms of similar identity in a molecular system [12]. Depending upon the complexity of molecular systems, the algorithm for NN symmetry adaptation may vary.

The graphene monolayer investigated herein is quite complex, and the development of a global many-body PES would be very computational demanding and may be an unrealistic task. In fact, not the total energy, but the force acting on each C in the graphene network is realized as the major concerning quantities in molecular dynamics simulations. Therefore, a reliable FF of graphene is crucial and should be considered as a priority task in investigating the thermal property of graphene. In this study, the authors employ feed-forward NNs to develop a direct FF for graphene monolayer. Three strategies of fitting with ascending NN size are presented in order to evaluate the influence of C atoms on the FF. Such approach is necessary to understand the role of force acting on each C atom in a graphene monolayer. The present result is useful for quickly and accurately building the PES, which play an important role in graphene studies.

2. FF developing procedure

2.1. Description of geometry representation

In a large-scale graphene monolayer with thousands of C atoms, it is more beneficial to approximate the force acting on each individual C atom directly rather than to develop a full PES for the whole graphene sheet, which only allows indirect extractions of forces. The force acting on each C atom can be approximated by performing NN fitting, in which the internal variables defining relative positions of the surrounding atoms are used as input variables and the force components are used as targeted output.

As illustrated in Figure 1, the currently-considered C atom has direct interactions via sp^2 bonds with three surrounding atoms, which are denoted as C^1, C^2, and C^3. For convenience of NN fitting, C1 is chosen in such a way that it establishes the shortest C-C bond with C^0 among three nearest neighbor atoms, while C^3 gives the longest C-C bond distance with C^0. To describe the relative positions of C^1, C^2, and C^3 with respect to C^0, the authors employ the internal coordinate system (bond distances, bending and dihedral angles) as given in Table 1.

The completeness of long-range interaction may be enhanced by further expanding the surrounding environment and extending the set of FF input parameters. By doing this, the authors consider the long-range interaction effects on the center C atom, and supposingly add more correction terms to FF function. Specifically, this is done by further considering the relative positions of six additional C atoms, which are denoted as C^4,..., C^9 in Figure 1. In our model, C^4 and C^5 have direct interactions with C^1, C^6 and C^7 have direct interactions with C^2, while C^4 and C^5 bond directly to C^3. The internal-coordinate descriptions of those following C atoms are given in Table 1.

Table 1. *Internal variables (bond distances (Å) and angles (°)) that define the relative positions of nine surrounding C atoms*

Input variable	Description	Minimum	Maximum
r_1	C^0-C^1	1.219	1.526
r_2	C^0-C^2	1.271	1.594
r_3	C^0-C^3	1.349	1.737
r_4	C^1-C^4	1.219	1.588
r_5	C^1-C^5	1.289	1.737
r_6	C^2-C^6	1.219	1.594
r_7	C^2-C^7	1.271	1.737
r_8	C^3-C^8	1.219	1.576
r_9	C^3-C^9	1.297	1.737
Θ_1	C^2-C^0-C^1	101.799	139.372
Θ_2	C^3-C^0-C^1	101.818	137.031
Θ_3	C^4-C^1-C^0	101.799	139.168
Θ_4	C^5-C^1-C^0	100.778	141.919
Θ_5	C^6-C^2-C^0	101.799	139.372
Θ_6	C^7-C^2-C^0	101.682	142.127
Θ_7	C^8-C^3-C^0	100.778	142.127
Θ_8	C^9-C^3-C^0	101.818	140.479
Φ_1	C^3-C^0-C^1-C^2	123.828	180.000
Φ_2	C^4-C^1-C^0-C^2	0.000	180.000
Φ_3	C^5-C^1-C^0-C^2	0.000	180.000
Φ_4	C^6-C^2-C^0-C^1	0.000	180.000
Φ_5	C^7-C^2-C^0-C^1	0.000	180.000
Φ_6	C^8-C^3-C^0-C^1	0.000	180.000
Φ_7	C^9-C^3-C^0-C^1	0.000	180.000

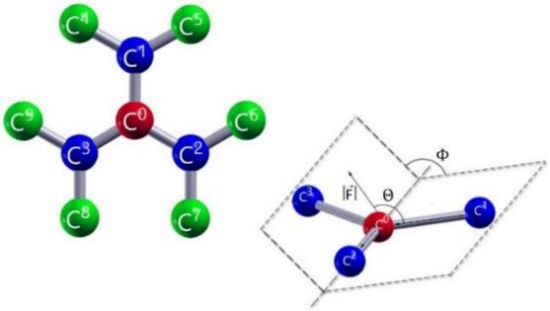

Figure 1. *The geometry configuration for FF construction: one center and nine surrounding C atoms. The force vector $|\vec{F}|$ is decomposed into three major components: force magnitude ($|\vec{F}|$), Θ and Φ. For the uniqueness of a configuration, the order of (C^1, C^2, C^3) is chosen in such a way that C^{0}C^1 is the shortest, while C^{0}C^3 is the longest bond*

In the conventional PES development with NN fitting, the output is a single quantity, which solely represents the total energy. In this study, the NN method is employed to predict the force vector $|\vec{F}|$ acting on a particular C atom instead of energy. The force vector, however, cannot be represented by a single quantity; in fact, it must be fragmented into three components which precisely reveal the magnitude and orientation of $|\vec{F}|$: force magnitude ($|\vec{F}|$), cosine of the bending angle between vectors $|\vec{F}|$ and C^{0}C^1 (cos(Θ)), and cosine of the dihedral angle defined by vectors $|\vec{F}|$, C^{0}C^1, and C^{1}C^2 (denoted as cos(Φ)). A precise pictorial illustration of $|\vec{F}|$, cos(Θ) and cos(Φ) is presented in Figure 1.

In our developing procedure, it is necessary to build a database which describes the orientations of atoms (inputs) and forces (targeted outputs) based on electronic structure calculations. Subsequently, the numerical NN fitting method is employed to fit the database and produce an approximate force field.

2.2. Electronic structure calculations

The electronic structure calculations in this study for force-field sampling are executed by first-principles calculations based on density functional theory (DFT) [13, 14], as implemented in the Quantum Espresso package [15]. In particular, the Perdew-Burke-Ernzerhof (PBE) exchange-correlation functional [16, 17] within the generalized gradient approximation is employed to calculate total energies of the graphene supercell with the Vanderbilt ultrasoft pseudopotential for C atoms [18, 19]. Structural data are obtained by performing sample Born-Oppenheimer molecular dynamics with one restriction that the Brillouin zone is only represented by the Γ point. The kinetic-energy cut-off of 45 Rydberg (612 eV) is chosen for plane-wave expansions.

To sample data points for fitting purposes, the authors perform Born-Oppenheimer molecular dynamics at 1500 K for a graphene supercell consisting of 32 C atoms. Due to periodicity of the graphene sheet, in each step during the MD trajectory, the authors are able to extract 32 configurations by considering each C atom as the center atom. In total, after nearly 5,000 MD steps, the database is constructed with 157,792 configurations.

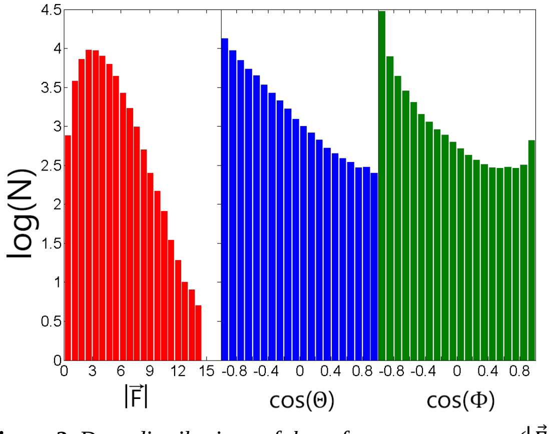

Figure 2. *Data distributions of three force components ($|\vec{F}|$, cos(Θ) and cos(Φ)). N represents the number of configurations in a particular range*

For reducing computational feasibility and attaining higher efficiency in NN training, it is necessary to reduce the database by making a random selection of 55,726 configurations from the initial database. Hence, in most of the NN fits discussed below, the training set is constituted by 55,726 configurations, while an independent set of 2,786 configurations is randomly chosen to construct the testing set for validation purposes. The ranges of three output components in the training set consisting of 55,726 data points are given in Table 2. In Figure 2, the statistical distributions of three force components are shown. It can be seen clearly that forces with low magnitude (near equilibrium region, from 0 to 3 eV/Å) dominates. Since C^0C^1 is chosen as the shortest C-C bond around C^0, $|\vec{F}|$ has a higher tendency to be opposite to the C^0C^1 vector. As a result, more values around the -180° region are obtained for Θ. Overall, the authors believe that a sufficient number of configurations has been sampled to explicitly describe three output components.

Table 2. *Minimum and maximum values of three output components*

Output components	Minimum	Maximum
$|\vec{F}|$ (eV/Å)	0.081	14.508
cos(Θ)	-1.000	1.000
cos(Φ)	-1.000	1.000

2.3. Neural network fitting method

Two-layer feed-forward NNs, which can be consulted from Hagan et al. [7], are employed to approximate the FF data in this study. As mentioned earlier, such an NN architecture is widely utilized in constructing PES for gas-phase molecular as well as condensed-matter systems. Initially, the input and target data are scaled from -1 to 1 to enhance the training effectiveness. For convenience, the authors generally denote input as *p* and targeted output as *t*. The scaling technique is done as following.

$$p = \frac{2(p_i - p_i^{min})}{p_i^{max} - p_i^{min}} - 1 \text{ for i = 1,…,24} \quad (1)$$

$$t_i^{scaled} = \frac{2(t_i - t_i^{min})}{t_i^{max} - t_i^{min}} - 1 \text{ for i = 1, 2, 3} \quad (2)$$

where p_i^{min} and p_i^{max} respectively represent the minimum and maximum values of the i^{th} input parameter, while t_i^{min} and t_i^{max} denote the minimum and maximum values of the i^{th} output parameter. Mathematically, this scaling technique is meaningful because it helps to reduce input and output data ranges for numerical fitting; moreover, it should be noticed that such a technique also makes physical units vanish. As can be seen from Table 1, the ranges of the second and third output parameters (cos(Θ)) and third (cos(Φ)), respectively are almost [-1; 1]. Therefore, the above scaling technique does not have significant impacts on those values; however, for consistency of the overall procedure, scaling is still applied to those two output quantities.

Subsequently to scaling, the *n* scaled inputs are introduced into the first (input) layer and processed by the hidden layer, then finally *m* outputs in the last layer are produced. An illustration for the operating principle of a typical feed-forward NN is introduced in Figure 3. *n* scaled inputs are processed by *M* hidden neurons in the first layer to produce *M* intermediate values using the following equation:

$$x_i = f\left(\sum_{j=1}^{n} w_{1_{i,j}} p_j^{scaled} + b_{1_j}\right) \text{ for } i = 1,\ldots,\mathrm{M}$$

where w_1 is an *M*×*n* matrix identified as the first weight matrix, b_1 is an *M*×*1* column vector representing the bias values of the first layer, and f is the transfer function. In this study, the authors employ the hyperbolic tangent function as the transfer function in the first layer. Hornik et al. [20] showed that the utilization of a sigmoid function in the hidden layer could make a NN an universal approximator for analytic functions.

The final output quantities T are subsequently calculated by employing a linear function to combine all x values as:

$$T_l = \sum_{k=1}^{M} w_{2_{k,l}} x_k + b_{2_l} \text{ for } l = 1,\ldots,m$$

In the above equation, w_2 is an $m\times M$ matrix, b_2 is an $m\times 1$ vector, which represent the weight and bias values of the second layer, respectively. In our case, T consists of three quantities, which represent the scaled force components (targeted outputs).

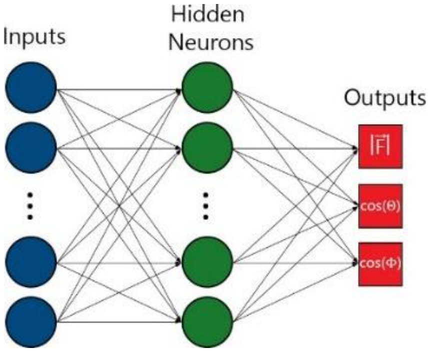

Figure 3. *The feed-forward NN model for FF fitting*

3. Results and discussions

Traditionally, the NN output for PES fitting consists of one single value; however, the description for a force vector requires that at least three parameters get involved as stated in Figure 1. As a result, the fitting process in this study is more complicated. In fact, the authors suggest three strategies to obtain well-fitted FFs using the NN method. In the first strategy, (1) the authors only consider three surrounding C atoms to have impacts on the force acting on C^0; in other words, six input variables (r_1, r_2, r_3, θ_1, θ_2 and ϕ_1), which fully describe the relative positions of (C^1, C^0, C^3), will be taken into account. Hence, a feed-forward NN with six input signals will be employed to predict three components of the force. In the second approach, (2) the authors consider nine surrounding C atoms to have impacts on the magnitude of the force acting on C^0 ($|\vec{F}|$) and cos(Θ), while the prediction of cos(Φ) is attributed by considering three nearest neighbors. Therefore, two different NNs need to be constructed: one NN reading all 24 input signals will be employed to fit $|\vec{F}|$ and cos(Θ), while another NN reading 6 input signals will be employed to predict the last output quantity, cos(Φ). In the last approach, (3) a highly-complex NN operating on 24 input signals will be employed to fit the three outputs simultaneously.

3.1. NN FF with six input signals

As mentioned earlier, three components of a force vector ($|\vec{F}|$) are predicted by considering the influence of only three surrounding carbon atoms. Potentially, there is an advantage when this strategy is used, i.e. the number of involving variables in the FF function will be highly reduced. Compared to the full utilization of nine surrounding C atoms (which results in a total number of 24 input variables), in this case, only six input variables are introduced into the first layer of a feed forward NN. Because of lower numbers of input variables, the size of NN parameters (weight and bias values) in this case would be significantly smaller. As a result, the training process consumes less computational time. In addition, it is also more advantageous to extract the force from the NN function.

The FF is fitted with NNs that have various numbers of hidden neurons (from 10 to 35 neurons). At this point, the root-mean-squared error (*rmse*) and average-absolute error (*aae*) are determined for the training and testing sets for statistical accuracy evaluation. As shown in Table 2, with only 10 neurons in the hidden layer, the testing *rmse* and *aae* for $|\vec{F}|$ of the training set are 0.480 and 0.381 eV/Å, respectively. Compared to the maximum of $|\vec{F}|$ in the database (14.51 eV/Å), the ratio of $aae/|\vec{F}|$ is about 2.63%. For convenience, the *rmse* and *aae* of three outputs given by six different NN FFs are summarized in Table 3. It can be observed that the training *rmse* and *aae* for cos(Θ) and cos(Φ) are relatively large compared to their maximum values (1.00) when the NN is constructed with 10 neurons.

Table 3. *Training (and testing) rmse and aae of the fitted NN FFs which process six input signals*

Number of neurons		10	15	20	25	30	35
rmse	$\|\vec{F}\|$ (eV/Å)	0.480 (0.480)	0.476 (0.471)	0.475 (0.473)	0.474 (0.472)	0.473 (0.477)	0.472 (0.470)
	cos (Θ)	0.161 (0.161)	0.156 (0.152)	0.155 (0.152)	0.153 (0.151)	0.153 (0.153)	0.153 (0.147)
	cos (Φ)	0.179 (0.177)	0.178 (0.161)	0.176 (0.171)	0.176 (0.169)	0.175 (0.184)	0.175 (0.167)
aae	$\|\vec{F}\|$ (eV/Å)	0.381 (0.381)	0.378 (0.375)	0.377 (0.378)	0.376 (0.375)	0.376 (0.380)	0.376 (0.375)
	cos (Θ)	0.104 (0.106)	0.100 (0.100)	0.099 (0.098)	0.097 (0.096)	0.097 (0.099)	0.096 (0.092)
	cos (Φ)	0.093 (0.093)	0.092 (0.088)	0.091 (0.088)	0.090 (0.089)	0.090 (0.093)	0.089 (0.085)

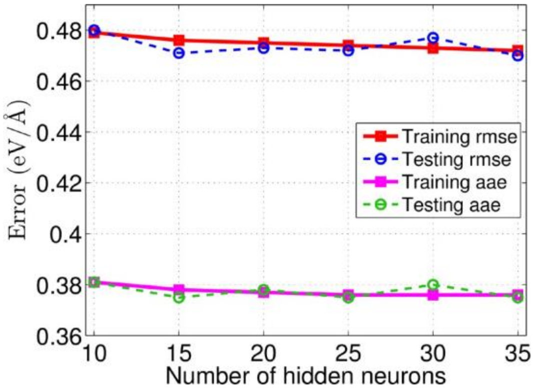

Figure 4. *Training errors (rmse and aae) vs. number of hidden neurons from NN fitting with six input variables*

In Figure 4, the authors show the fitting performance of $|\vec{F}|$, which is revealed by the *rmse* and *aae* of training and testing data, with respect to the number of hidden neurons.

As the number of hidden neurons increases from 10 to 35, it is statistically observed that the fitting quality of both training and testing sets is not significantly improved. For cos(Θ) and cos(Φ) fitting, it is also the case as the utilization of 35 hidden neurons does not result in significant change in the overall accuracy. Therefore, the authors believe that the FF is not well interpolated if only six input variables are considered in NN constructions. In other words, the physical picture of C-C interactions is not well described when the authors only consider the influence of three surrounding C atoms. Hence, NN fitting attains its fitting limit regardless of hidden neuron numbers.

3.2. The combination of two feed-forward NNs to fit the FF

In this approach, the authors combine two feed-forward NNs to represent the FF. All 24 input parameters describing relative positions of nine surrounding C atoms are used as the input layer for the first NN, which is employed to handle two force components ($|\vec{F}|$ and cos(Θ)). The number of hidden neurons for this NN ranges from 50 to 125. In the second NN to fit the last force component (cos(Φ)), the input layer only consists of six variables. For convenience, the statistical fitting errors of the first and second NNs are shown in Table 4.

Table 4. *Training (and testing) rmse and aae of the combination of two feed-forward NNs*

Number of neurons		50	75	100	125
rmse	$|\vec{F}|$ (eV/Å)	0.206 (0.208)	0.200 (0.202)	0.195 (0.196)	0.193 (0.192)
	cos(Θ)	0.087 (0.086)	0.085 (0.085)	0.082 (0.083)	0.082 (0.080)
	cos(Φ)	0.173 (0.174)	0.173 (0.163)	0.173 (0.182)	0.173 (0.171)
aae	$|\vec{F}|$ (eV/Å)	0.162 (0.163)	0.156 (0.160)	0.153 (0.154)	0.151 (0.151)
	cos(Θ)	0.055 (0.055)	0.054 (0.053)	0.053 (0.052)	0.052 (0.052)
Number of neurons		**25**	**30**	**45**	**50**
rmse	cos(Φ)	0.173 (0.174)	0.173 (0.163)	0.173 (0.182)	0.173 (0.171)
aae		0.087 (0.086)	0.087 (0.086)	0.087 (0.090)	0.088 (0.086)

Recall that in the first approach, when the number of hidden neurons increases from 50 to 125, training and testing *rmse* and *aae* drop slowly. Compared to the *rmse* and *aae* in the previous stage, it can be seen that the fitting quality of $|\vec{F}|$ and cos(Θ) is improved. As shown in Table 4, when the FF is fitted with 125 hidden neurons, the authors obtain the best rmse and *aae* for both training and testing sets. The third force component fitting is executed using NNs with 25-50 hidden neuron, and no significant improvements can be observed in the second NN for cos(Φ) prediction. Compared to the results using the first fitting strategy presented above, the current *aae* produced by a 30-hidden-neuron NN is improved by 3%. When the NN size increases to 45-50 neurons, the *aae* does not drop as expected.

By performing such fitting trials, the authors are able to interpret the physical characteristics of the dihedral angle in the system. Even when cos(Φ) is treated separately with a feed-forward NN, the fitting accuracy does not significantly increase. Thus, the six chosen input variables are still not sufficient to describe the true behavior of the third force component. This means that it highly depends on the interaction environment, as will be proved in the third approach to fit the FF shown below.

3.3. NN FF with 24 input signals

In the last approaching strategy, all 24 input parameters which describe the relative positions of nine surrounding C with respect to the center C atom jointly constitute the input layer of the NN FF. At this stage, the number of hidden neurons ranges from 50 to 125. As the authors evaluate the training and testing errors (both *rmse* and *aae*), this approach possesses the most promising fitting ability because of its best accuracy compared to the previous NN fits. With 50 hidden neurons, the *aae* in force magnitude prediction for the testing set is 0.164 eV/Å. Recall that when the NN FF is constructed with six input variables and 35 hidden neurons, the training *aae* only reaches 0.376 eV/Å, which is almost 2.3 times larger than the current error obtained in this case. Also, the *aae* for cos(Θ) and cos(Φ) fitting are 0.055 and 0.061, respectively. Further increasing the number of hidden neurons to 75 or 100 to fit the current training dataset (55,726 configurations), the fitting accuracy for three output quantities is slightly improved (see Table 5). When the 100-hidden-neuron NN is employed, at the termination of training, the distribution of training errors is close to a Gaussian function with a domination of small training errors around 0 as shown in Figure 5.

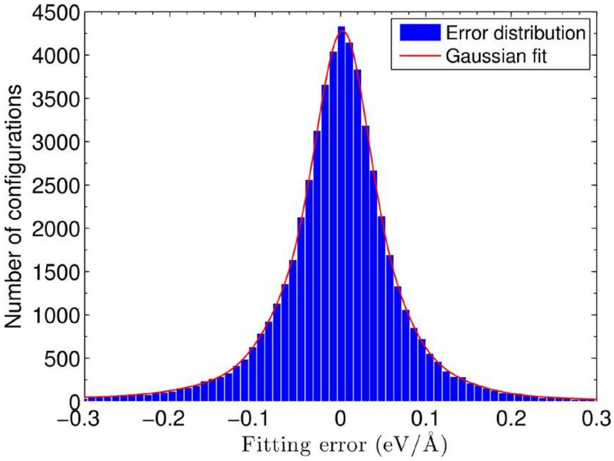

Figure 5. *Distribution of training error when a 100-hidden-neuron NN is employed to fit24 input variables*

The utilization of 125 hidden neurons, however, does not even give any rises to the fitting accuracy. In fact, the testing *aae* for such a fit is almost similar to that of the 100-hidden-neuron fit. It seems that the precision limit has been attained when a 100-hidden-neuron NN is employed to process 24 input parameters. As seen in Figure 6, the outputs predicted by the 100-neuron NN FF and (real) target points of three force components are very close to others, which show that excellent accuracy has been obtained in the NN fit with 100 hidden neurons.

Table 5. *Training (and testing) rmse and aae of the fitted NN FFs which process 24 input signals*

Number of neurons		50	75	100	125
rmse	$\lvert\vec{F}\rvert$ (eV/Å)	0.208 (0.210)	0.201 (0.205)	0.196 (0.194)	0.194 (0.198)
	cos(Θ)	0.085 (0.082)	0.081 (0.081)	0.080 (0.083)	0.080 (0.082)
	cos(Φ)	0.111 (0.112)	0.106 (0.103)	0.104 (0.110)	0.104 (0.110)
aae	$\lvert\vec{F}\rvert$ (eV/Å)	0.163 (0.164)	0.158 (0.160)	0.154 (0.151)	0.152 (0.156)
	cos(Θ)	0.055 (0.055)	0.052 (0.051)	0.052 (0.053)	0.052 (0.052)
	cos(Φ)	0.061 (0.061)	0.058 (0.057)	0.057 (0.058)	0.057 (0.058)

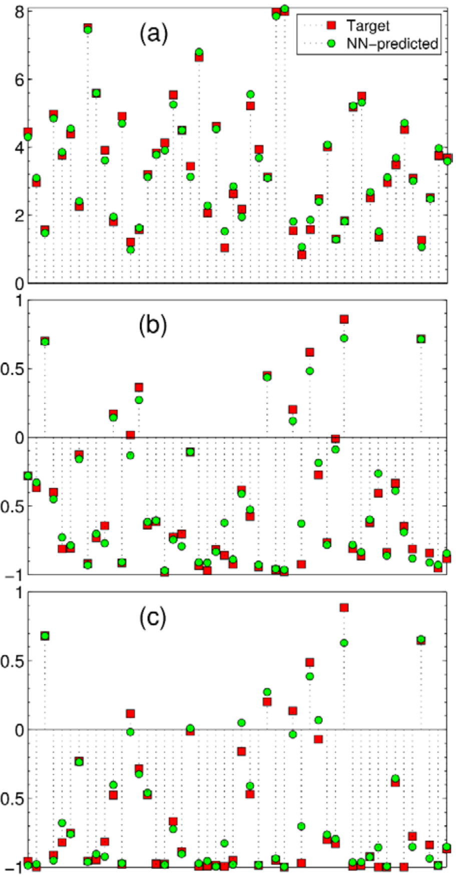

Figure 6. *NN-predicted outputs and actual targets of a random testing set for three force components: (a) $\lvert\vec{F}\rvert$, (b) cos(Θ), and (c) cos(Φ). The fit is produced by employing a 100- hidden-neuron NN*

The authors so far believe that the third strategy is the most promising approach even though it is more computational demanding in numerical fitting. In practice, this NN also costs more computational efforts to extract the force than the first two fitting approaches; however, compared to direct electronic structure calculations using DFT, the utilization of such NN FFs in MD simulations for thousands of C atoms is still far more advantageous.

A large training set (more than 142,000 configurations) is built from the original set of data points, and a 75-hidden-neuron NN is employed to interpolate the new data. Indeed, the authors observe no significant improvements in the fitting accuracy. The *aae* for the testing set (consisting of nearly 8,000 configurations) for $\lvert\vec{F}\rvert$, cos(Θ) and cos(Φ) are reported as 0.156 eV/Å, 0.052, and 0.059, respectively. Such *aae* of $\lvert\vec{F}\rvert$ is close to the corresponding *aae* (0.160 eV/Å) reported when 55,726 data points are fitted using 75 hidden neurons. Therefore, the authors believe that further expansion of database is unnecessary to improve fitting accuracy. More importantly, the expansion of database also results in much higher computational cost.

4. Summary

In this study, the authors have developed FFs for a graphene monolayer using a new NN fitting strategy. The force-field database is obtained by selecting geometry configurations and forces from sample DFT calculations of a periodic graphene sheet. Then, a force acting on a particular C atom is decomposed into three major components: $\lvert\vec{F}\rvert$, cos(Θ) and cos(Φ), which can be defined by the relative positions with the surrounding C atoms.

The NN fitting technique is employed to approximate the force data with three different fitting strategies. In the first approach, the authors only consider the involvement of three nearest-neighbor C atoms in the FF, which consequently results in the participation of six variables in the first (input) NN layer. From fitting accuracy analysis, it is shown that the first strategy is a premature approach and the resultant NNs do not interpolate the FF well. When the authors employ a 35-hidden-neuron NN, the best testing *aae* for $\lvert\vec{F}\rvert$, cos(Θ) and cos(Φ) are obtained as 0.375 eV/ Å, 0.092, and 0.085, respectively. In the second approach, a NN reading 24 input variables is employed to fit $\lvert\vec{F}\rvert$ and cos(Θ), while cos(Φ) is interpolated by a separate six-input-parameter NN. By separating the targeted outputs, the authors expect that the smoothness of NN functions would increase, and thereby improve fitting accuracy significantly. However, the resultant accuracy is slightly improved in comparisons with the previous fitting strategy.

In the last approach, 24 input parameters are simultaneously introduced into the NNs. Noticeably, the fitting accuracy is significantly improved (best *aae* for a testing set are now 0.152 eV/Å, 0.051, and 0.057). Compared to the best *aae* obtained from fitting six input variables, it is observed that the improvement in *aae* ranges from 33% to 59%. The obtained *aae* for $\lvert\vec{F}\rvert$ is only about 1% of the maximum force magnitude. Therefore, the authors conclude that the FF is best approximated when a large interacting environment (with nine surrounding C atoms) is considered. Even though the third fitting strategy causes higher computational expense, the authors still believe that it is much more advantageous for force calculations in large-scale graphene rather than direct executions of first-principles calculations.

The FF developed in this study is considered as a step toward an ultimate goal, in which molecular dynamics investigations of heat spreading process in graphene will be studied. In addition, future development of a FF which can describe structural defects in graphene is recognized as an important task. From fitting accuracy analysis, the authors believe that further expanding the interacting environment (more C atoms are involved) and database would increase the fitting accuracy. However, the authors should imply that the NN employed to deal with 24 input parameters already exceeds the maximum computational resource.

The developed methodologies and insights can significantly impact various applications of graphene-based materials, such as energy storage, thermal management, and catalytic processes, demonstrating the broad potential of advanced computational techniques in material science [21, 22, 23, 24].

Acknowledgements: Viet Q.Bui was funded by the Postdoctoral Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2023.STS.19. The authors also thank Dr. Hung M.Le for his expert support on this project.

REFERENCES

[1] A.K. Geim and K.S. Novoselov, "The Rise of Graphene", *Nature Mater,* vol. 6, no. 3, pp. 183-191, 2007.

[2] R.Prasher, "Graphene Spreads the Heat", *Science,* vol. 328, no. 5975, pp. 185-186, 2010.

[3] V. Q. Bui and H. M. Le, "Naphthalene adsorptions on graphene using Cr/Cr2/Fe/Fe2 linkages: Stability and spin perspectives from first-principles calculations", *Chem. Phys. Lett.,* vol. 614, pp. 238-242, 2014.

[4] V. Perebeinos and J. Tersoff, "Valence Force Model for Phonons in Graphene and Carbon Nanotubes", *Phys. Rev. B,* vol. 79, no. 24, p. 241409, 2009.

[5] A. Lajevardipour, M. Neek-Amal, and F. M. Peeters, "Thermomechanical Properties of Graphene: Valence Force Field Model Approach", *J. Phys.: Condens. Matter,* vol. 24, no. 17, p. 175303, 2012.

[6] G. Kalosakas, N. N. Lathiotakis, C. Galiotis, and K. Papagelis, "In-plane Force Fields and Elastic Properties of Graphene", *J. Appl. Phys.,* vol. 113, no. 13, p. 134307, 2013.

[7] M. T. Hagan, H. B. Demuth and M. Beale, Neural Network Design, Boulder: Colorado Bookstore, 1996.

[8] L. Raff, R. Komanduri, M. Hagan, and S. Bukkapatnam, *Neural Networks in Chemical Reaction Dynamics*, Oxford: Oxford University Press, 2012.

[9] C. M. Handley and P. L. A. Popelier, "Potential Energy Surfaces Fitted by Artificial Neural Networks", *J. Phys. Chem. A,* vol. 114, no. 10, pp. 3371-3383, 2010.

[10] J. Behler, "Neural Network Potential-energy Surfaces in Chemistry: A Tool for Large-scale Simulations", *Phys. Chem. Chem. Phys.,* vol. 13, no. 40, pp. 17930-17955, 2011.

[11] A. Pukrittayakamee, M. Malshe, M. Hagan, L. M. Raff, R. Narulkar, S. Bukkapatnum, and R. Komanduri, "Simultaneous Fitting of a Potential-energy Surface and Its Corresponding Force Fields Using Feedforward Neural Networks", *J. Chem. Phys.,* vol. 130, no. 13, p. 134101, 2009.

[12] H. M. Le, S. Huynh, and L. M. Raff, "Molecular dissociation of hydrogen peroxide (HOOH) on a neural network ab initio potential surface with a new configuration sampling method involving gradient fitting", *J. Chem. Phys.,* vol. 131, no. 1, p. 014107, 2009.

[13] P. Hohenberg and W. Kohn, "Inhomogeneous Electron Gas", *Phys. Rev.,* vol. 136, no. 3B, pp. B864-B871, 1964.

[14] W. Kohn and L. J. Sham, "Self-Consistent Equations Including Exchange and Correlation Effects", *Phys. Rev.,* vol. 140, no. 4A, pp. A1133-A1138, 1965.

[15] P. Giannozzi *et al.,* "QUANTUM ESPRESSO: A Modular and Open-source Software Project for Quantum Simulations of Materials", *J. Phys.: Condens. Matter,* vol. 21, no. 39, p. 395502, 2009.

[16] J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized Gradient Approximation Made Simple", *Phys. Rev. Lett.,* vol. 77, no. 18, pp. 3865-3868, 1996.

[17] J. P. Perdew, A. Ruzsinszky, J. Tao, V. N. Staroverov, G. E. Scuseria, and G. I. Csonka, "Prescription for the Design and Selection of Density Functional Approximations: More Constraint Satisfaction with Fewer Fits", *J. Chem. Phys.,* vol. 123, no. 6, p. 62201, 2005.

[18] D. Vanderbilt, "Soft Self-consistent Pseudopotentials in a Generalized Eigenvalue Formalism", *Phys. Rev. B,* vol. 41, no. 11, pp. 7892-7895, 1990.

[19] A. D. Corso, "Density-functional Perturbation Theory with Ultrasoft Pseudopotentials", *Phys. Rev. B,* vol. 64, no. 23, p. 235118, 2001.

[20] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximators", *Neural Networks,* vol. 2, no. 5, pp. 359-366, 1989.

[21] M. C. Vu *et al.,* "Scalable graphene fluoride sandwiched aramid nanofiber paper with superior high-temperature capacitive energy storage", *Chem. Eng. J.,* vol. 444, p. 136504, 2022.

[22] M. C. Vu, N. A. T. Thieu, J.-H. Lim, W.-K. Choi, J. C. Won, M. A. Islam, and S.-R. Kim, "Ultrathin thermally conductive yet electrically insulating exfoliated graphene fluoride film for high performance heat dissipation", *Carbon,* vol. 157, pp. 741-749, 2020.

[23] V. Q. Bui, H. M. Le, Y. Kawazoe, and D. Nguyen-Manh, "Graphene-Cr-Graphene intercalation nanostructures: stability and magnetic properties from density functional theory investigations", *J. Phys. Chem. C,* vol. 117, no. 7, pp. 3605-3614, 2013.

[24] X. Liu *et al.,* "Advanced Dual-Atom Catalysts on Graphitic Carbon Nitride for Enhanced Hydrogen Evolution via Water Splitting", *Nanoscale,* vol. 16, no. 27, pp. 13148-13160, 2024.