Combining the D3 dispersion correction with the neuroevolution machine-learned potential

OPEN ACCESS
IOP Publishing
Journal of Physics: Condensed Matter
J. Phys.: Condens. Matter **36** (2024) 125901 (7pp)
https://doi.org/10.1088/1361-648X/ad1278

Combining the D3 dispersion correction with the neuroevolution machine-learned potential

Penghua Ying[1,*] **and Zheyong Fan**[2,*]

[1] Department of Physical Chemistry, School of Chemistry, Tel Aviv University, Tel Aviv 6997801, Israel
[2] College of Physical Science and Technology, Bohai University, Jinzhou 121013, People's Republic of China

E-mail: hityingph@163.com and brucenju@gmail.com

Received 8 October 2023, revised 17 November 2023
Accepted for publication 5 December 2023
Published 14 December 2023

Abstract

Machine-learned potentials (MLPs) have become a popular approach of modeling interatomic interactions in atomistic simulations, but to keep the computational cost under control, a relatively short cutoff must be imposed, which put serious restrictions on the capability of the MLPs for modeling relatively long-ranged dispersion interactions. In this paper, we propose to combine the neuroevolution potential (NEP) with the popular D3 correction to achieve a unified NEP-D3 model that can simultaneously model relatively short-ranged bonded interactions and relatively long-ranged dispersion interactions. We show that improved descriptions of the binding and sliding energies in bilayer graphene can be obtained by the NEP-D3 approach compared to the pure NEP approach. We implement the D3 part into the GPUMD package such that it can be used out of the box for many exchange-correlation functionals. As a realistic application, we show that dispersion interactions result in approximately a 10% reduction in thermal conductivity for three typical metal-organic frameworks.

Keywords: machine-learned potentials, neuroevolution potential, D3 dispersion correction, bilayer graphene, GPUMD, metal-organic frameworks, thermal conductivity

1. Introduction

Machine-learned potential (MLP) [1] is an emerging approach for modeling the interatomic interactions in materials. To achieve a linear scaling of the computational cost with respect to the system size, a MLP must be constructed based on local descriptors [2]. The descriptor for an atom is usually constructed based on the positions of its neighbors within a certain cutoff R_c. The average number of neighbors for an atom, hence the computational cost of a MLP, is proportional to R_c^3. Therefore, in practice, R_c is usually chosen to be a few Å. This length is usually sufficient for describing the bond interactions in typical materials, but is not sufficient for describing the London dispersion interactions that can extend to one to a few nm. However, the dispersion interactions are important in describing e.g. the interlayer attractions in the so-called van-der-Waals (vdW) layered materials [3, 4], structural transformation between a narrow-pore and large-pore phase in flexible metal-organic frameworks (MOFs) [5, 6], and host-guest interactions in MOFs [7, 8].

To address this challenge, a few attempts have been made to augment MLPs with dispersion corrections. Wen and Tadmor [9] added an attractive $-C_6/r_{ij}^6$ term multiplied by some switching/damping functions to a MLP, where r_{ij} is the distance between atoms i and j and $C_6 > 0$ is a fitting parameter.

* Authors to whom any correspondence should be addressed.

There are four additional fitting parameters in the switching functions [9]. They have obtained quite good description for the binding and sliding energies in bilayer graphene. By adding a similar $-C_6/r_{ij}^6$ term, Deringer *et al* [10] and Rowe *et al* [11] constructed general-purpose MLPs for phosphorous and carbon systems. Muhli *et al* [12] developed a more sophisticated method that determines the dispersion coefficient and damping function using a local descriptor.

Despite these achievements, it still requires quite a lot efforts to determine the dispersion interactions for a MLP model. The determination of the dispersion coefficient and the damping function is species and system dependent, and the process is thus not very systematical and transferable. Indeed, it is known that the dispersion coefficient is environment dependent, which has been taken into account in many popular dispersion corrections to density functional theory (DFT), such as the D3 approach [13, 14]. To our best knowledge, the D3 approach has not been combined with MLPs to perform large-scale atomitstic simulations. One reason is that there is so far no efficient implementation of D3 for the use with classical potentials. In this work, we make an efficient graphics processing units (GPU) implementation of D3 into the GPUMD package [15] and combine it with the machine-learned neuroevolution potential (NEP) [16–18] to form a NEP-D3 approach. This approach inherits all the merits of D3 and applies to the 94 elements H-Pu for a large number of DFT functionals. Due to the separability of the NEP and D3 parts in our approach, it also allows for isolating the role of dispersion interactions in affecting specific physical properties. We will use two example systems, bilayer graphene and MOFs, to demonstrate the convenience, accuracy, efficiency, generality, and usefulness of the NEP-D3 approach.

NEP is a neural-network-based MLP which got its name due the use of an evolutionary algorithm for training the free parameters [16–18]. In this method, the total energy of a system is taken as the sum of the site energies at each atom, which is taken as the output of a neural-network, as first proposed by Behler and Parrinello [19]. The input layer of the neural-network consists of a number of features called descriptor components. The design of descriptor components is crucial for a MLP and different MLPs presently in use differ from each other mainly by the descriptor. In the very first version of NEP [16], it has been realized to be beneficial to have two different kinds of descriptor components, the radial ones and the angular ones. The radial descriptor components depend only the interatomic distances, which are relatively cheaper to calculate, while the angular descriptor components also depend on bond angles and are relatively more expensive to calculate. Therefore, it has been suggested to use a relatively longer cutoff for the radial descriptor components r_c^R in combination with a relatively shorter cutoff for the angular descriptor components r_c^A when this is beneficial [16]. For example, this strategy has been used for modeling a number of vdW structures [20, 21] that have both strong covalent bonds and weak vdW interactions, where r_c^R was taken to be about 7–8 Å and r_c^A 3–4 Å. While constructing a pure NEP that incorporates dispersion interactions is feasible, the primary objective of this work is to introduce the NEP-D3 method that addresses dispersion interactions in a more elegant manner. We will compare results from the two approaches in terms of both accuracy and computational efficiency.

2. Results and discussion

2.1. Implementation of the D3 dispersion correction into GPUMD

The D3 dispersion correction has contributions from a 'two-body' (the meaning of the quotes will be made clear soon) part and a three-body part. However, it has been recommended not to include the three-body part [13]. We therefore only considered the two-body part, which is also the choice of our reference DFT code Vienna *Ab initio* simulation package (VASP) [22, 23]. We chose to use the Becke–Johnson damping which does not introduce spurious force at short distances [14]. The total D3 energy can be expressed as

$$U^{\mathrm{D3}} = \sum_i U_i^{\mathrm{D3}}; \tag{1}$$

$$U_i^{\mathrm{D3}} = -\frac{1}{2}\sum_{j\neq i}\frac{s_6 C_{6ij}}{r_{ij}^6 + (a_1 R_0 + a_2)^6} - \frac{1}{2}\sum_{j\neq i}\frac{s_8 C_{8ij}}{r_{ij}^8 + (a_1 R_0 + a_2)^8}, \tag{2}$$

where r_{ij} is the distance between atoms i and j, s_6, s_8, a_1, and a_2 are parameters depending on the chosen exchange-correlation functional, C_{6ij} and C_{8ij} are the dispersion coefficients for the ij atom pair, and R_0 is taken as the geometric mean of tabulated parameters for each species. The summation of j is over the neighbors of i within a cutoff R_{pot}.

The two dispersion coefficients are related by $C_{8ij} = C_{6ij}R_0^2$. The dispersion coefficient C_{6ij} is calculated as a function of the coordination numbers n_i and n_j,

$$C_{6ij}(n_i, n_j) = \frac{\sum_a \sum_b C_{6ijab}^{\mathrm{ref}} e^{-4\left[\left(n_i - n_{ia}^{\mathrm{ref}}\right)^2 + \left(n_j - n_{jb}^{\mathrm{ref}}\right)^2\right]}}{\sum_a \sum_b e^{-4\left[\left(n_i - n_{ia}^{\mathrm{ref}}\right)^2 + \left(n_j - n_{jb}^{\mathrm{ref}}\right)^2\right]}}, \tag{3}$$

where the summations of a and b are over the numbers of reference points for atoms i and j, respectively. Here, n_{ia}^{ref} is the ath reference coordination number for atom i, n_{jb}^{ref} is the bth reference coordination number for atom j, and C_{6ijab}^{ref} is the (a, b) reference dispersion coefficient for the atom pair (i, j).

The coordination number for atom i is defined as

$$n_i = \sum_{j\neq i}\frac{1}{1 + e^{-16\left[\left(R_i^{\mathrm{cov}} + R_j^{\mathrm{cov}}\right)/r_{ij} - 1\right]}}, \tag{4}$$

where R_i^{cov} is the effective covalent radius of atom i. The summation of j is over the neighbors of i within a cutoff R_{cn}. Because the coordination number of atom i depends on its neighbors, it is clear to see that the 'two-body' part of D3 is not a truly two-body (pairwise) potential, but a many-body potential. This is the reason for using the quotes. For a many-body

potential, the general formulation for force, virial, and heat current has been established before [24, 25]. The force acting on atom i can be written as

$$\boldsymbol{F}_i^{\mathrm{D3}} = \sum_{j \neq i} \boldsymbol{F}_{ij}^{\mathrm{D3}}, \tag{5}$$

where

$$\boldsymbol{F}_{ij}^{\mathrm{D3}} = \frac{\partial U_i^{\mathrm{D3}}}{\partial \boldsymbol{r}_{ij}} - \frac{\partial U_j^{\mathrm{D3}}}{\partial \boldsymbol{r}_{ji}}, \tag{6}$$

and $\boldsymbol{r}_{ij} \equiv \boldsymbol{r}_j - \boldsymbol{r}_i$. The per-atom virial can be defined as

$$\mathbf{W}_i^{\mathrm{D3}} = \sum_{j \neq i} \boldsymbol{r}_{ij} \otimes \frac{\partial U_j^{\mathrm{D3}}}{\partial \boldsymbol{r}_{ji}}, \tag{7}$$

and the per-atom heat current can be expressed as

$$\boldsymbol{J}_i^{\mathrm{D3}} = \mathbf{W}_i^{\mathrm{D3}} \cdot \boldsymbol{v}_i, \tag{8}$$

where $\boldsymbol{v}_i$ is the velocity of atom i. The efficient GPU implementation of D3 then follows the general algorithms for many-body potentials [15]. From a practical point of view, the use of D3 only requires three inputs: the exchange-correlation functional, the cutoff for the potential R_{pot}, and the cutoff for the coordination number R_{cn}. The combined NEP-D3 potential is simply a sum of D3 and NEP energies,

$$U^{\mathrm{NEP-D3}} = U^{\mathrm{NEP}} + U^{\mathrm{D3}}, \tag{9}$$

where U^{NEP} is the NEP energy as detailed in previous works [16–18]. The NEP model here can also be replaced by the NEP-ZBL model [26] where the Ziegler–Biersack–Littmark (ZBL) potential is used to describe the strong repulsive forces at extremely short distances.

To confirm the correctness of our GPU implementation of D3 in GPUMD and to evaluate the effects of cutoffs, we take three MOF materials (MOF-5, ZIF-8, and HKUST-1) as studied before using NEP models [27] and compare the calculated forces to those from VASP (using the `IVDW = 12` option). The results are shown in figure 1. The VASP code uses $R_{\mathrm{pot}} = 50.2$ Å and $R_{\mathrm{cn}} = 20$ Å as defaults. With $R_{\mathrm{pot}} = 12$ Å and $R_{\mathrm{cn}} = 6$ Å, the forces calculated from GPUMD already agree quite well with those from VASP. With increasing cutoff, the root mean square errors (RMSEs) between GPUMD and VASP implementations, eventually diminish. While D3 is almost free in DFT calculations, it can take a considerable portion of time for MLPs, which is particularly true for the highly efficient NEP approach. Therefore, a trike between accuracy and efficiency must be made in selecting the D3 cutoffs in NEP-D3. We used $R_{\mathrm{pot}} = 12$ Å and $R_{\mathrm{cn}} = 6$ Å for all the subsequent calculations with GPUMD.

2.2. Comparison between NEP-D3 and pure NEP for bilayer graphene

We now compare the NEP-D3 and the pure-NEP approaches, taking bilayer graphene systems as an example. To this end,

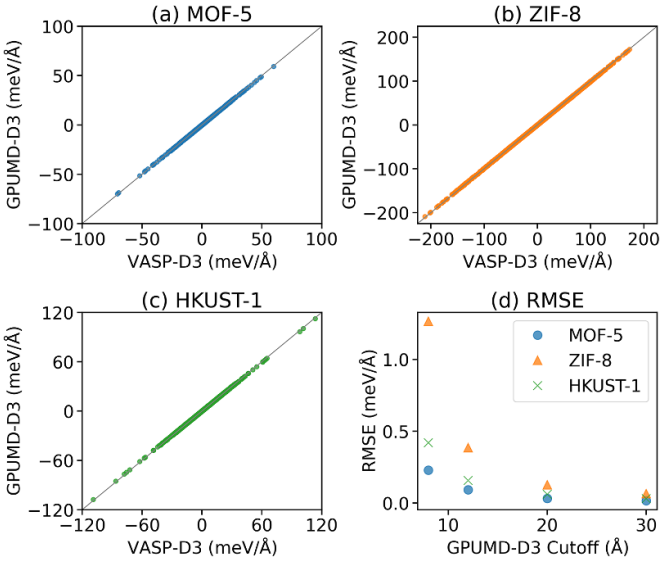

Figure 1. (a)–(c) The D3 forces computed from GPUMD implementation with $R_{\mathrm{pot}} = 12$ Å and $R_{\mathrm{cn}} = 6$ Å as compared to those from VASP implementation with $R_{\mathrm{pot}} = 50.2$ Å and $R_{\mathrm{cn}} = 20$ Å. For each MOF, we generated 10 structures with random cell deformations (from −3% to 3%) and atom displacements (less than 0.1 Å), starting from optimized ones. (d) RMSE between GPUMD and VASP D3 forces as a function of R_{pot} in GPUMD, where R_{cn} is taken as $R_{\mathrm{pot}}/2$.

we generated a training data set consisting of bilayer graphene structures with different interlayer distances (from 2 Å to 10 Å) and relative lateral shifts (including the important AA, AB, and saddle point (SP) stacking patterns and intermediate ones) as well as those from molecular dynamics (MD) simulations (from 300 K to 1500 K) driven by the second-generation reactive empirical bond-order potential [28] in combination with the registry-dependent interlayer potential [29]. For all the structures, we performed single-point DFT calculations as details in appendix A to obtain energy, force, and virial data that are needed for NEP training. Two different kinds of reference data sets were considered: one based on the PBE functional [30] and the other based on PBE combined with D3 [14]. We labeled them as PBE and PBE-D3 data sets, respectively.

To obtain a NEP-D3 model, we first trained a NEP model based on the PBE data set. In this case, there is no need to use a large $r_{\mathrm{c}}^{\mathrm{R}}$ in NEP because the interactions from the PBE functional are essentially short-ranged. We then took $r_{\mathrm{c}}^{\mathrm{R}} = r_{\mathrm{c}}^{\mathrm{A}}$ and tested the convergence of training accuracy with increasing cutoff. All the other hyper-parameters of NEP are listed in appendix B. Figure 2 shows that $r_{\mathrm{c}}^{\mathrm{R}} = r_{\mathrm{c}}^{\mathrm{A}} = 4.5$ Å is quite optimal. By adding up the D3 part, we then obtained a compound potential model named as NEP(4.5 Å, 4.5 Å)-D3. For the pure NEP models, we fixed the angular cutoff to 4.5 Å and considered three radial cutoffs: 6 Å, 8 Å, and 10 Å. This gives rise to three NEP models denoted as NEP(6 Å, 4.5 Å), NEP(8 Å, 4.5 Å), and NEP(10 Å, 4.5 Å), respectively, which were trained against the PBE-D3 data set.

The performances of the four models are compared in table 1. The NEP(4.5 Å, 4.5 Å)-D3 model has the highest

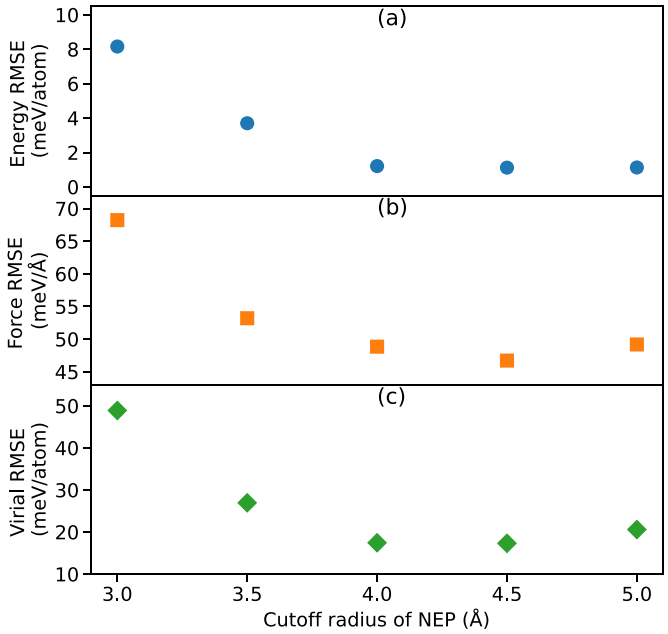

Figure 2. RMSEs of (a) energy, (b) force, and (c) virial for NEPs with different cutoffs (with $r_c^R = r_c^A$) trained against the Perdew–Burke–Ernzerhof (PBE) data set.

Table 1. The RMSEs for energy (meV/atom), force (meV Å^{-1}), and virial (meV/atom) of the NEP models versus the PBE-D3 reference and the computational speeds (atom-step/second) in MD simulations of bilayer graphene with 40 000 atoms using one GeForce RTX 4090 GPU card.

NEP models	Energy	Force	Virial	Speed
NEP (4.5 Å, 4.5 Å)-D3	1.16	46.72	15.38	6.29×10^{-6}
NEP (6.0 Å, 4.5 Å)	1.28	47.45	17.33	2.05×10^{-7}
NEP (8.0 Å, 4.5 Å)	1.25	48.08	16.09	1.53×10^{-7}
NEP (10.0 Å, 4.5 Å)	1.29	49.60	17.44	8.50×10^{-6}

accuracy in terms of energy, force, and virial. However, it has the lowest computational speed. The D3 part takes about 75% of the computational cost in the NEP(4.5 Å, 4.5 Å)-D3 model. This reflects the high computational cost of D3 and also the high computational efficiency of NEP. Indeed, the NEP approach has been shown to be far more computationally efficient than other state-of-the-art MLPs [18]. Because the dispersion interactions are weak forces, the RMSEs are not the best indicators for evaluating the accuracy. To better appreciate the higher accuracy brought about by the NEP-D3 approach compared to the pure NEP approach, we examine the binding and sliding energies in detail below.

Figure 3 shows the binding energy curves from the various calculation methods. The binding energies in bilayer graphene cannot be well captured by the PBE functional, which is essentially zero at an interlayer distance of 4.5 Å, and this is the reason why a cutoff of 4.5 Å is sufficient for NEP to fit the PBE data set (figure 3(a)). By adding D3 to PBE, the binding energies can be nicely captured and the results from D3 is very close to those from the many-body dispersion (MBD) [31] (figure 3(b)), which is one of the most accurate dispersion corrections presently available. By adding D3 to NEP(4.5 Å, 4.5 Å), the resulting model, NEP(4.5 Å, 4.5 Å)-D3 is very close to PBE-D3 (figure 3(c)), and most of the errors come from the NEP part, with small extra errors from the truncation of the D3 part to $R_{pot} = 12$ Å and $R_{cn} = 6$ Å. Without adding up D3, the pure NEP models with a relatively large radial cutoff can also partially capture the binding energies, but the curves are not as smooth as that from NEP-D3 (figure 3(d)–(f)). The best results were obtained with a radial cutoff of 8 Å, which means that increasing the radial cutoff in NEP is not a feasible way to improve the accuracy regarding the vdW interactions. To sum up, the NEP(4.5 Å, 4.5 Å)-D3 model outperform all the pure NEP models for describing the binding energies in bilayer graphene. It is expected that similar conclusions can be drawn for dispersion-dominated binding energies in layered materials.

In contrast to the binding energies, the sliding energies of bilayer graphene with equilibrium interlayer spacing (about 3.4 Å) are not dominated by the D3 part. Therefore, as shown in figure 4(a), results from NEP(4.5 Å, 4.5 Å) are already very close to those from PBE-D3 and adding D3 does not make significant changes. However, the resulting NEP(4.5 Å, 4.5 Å)-D3 model significantly outperforms the pure NEP models with larger radial cutoffs (figure 4(b)). One possible reason is that the long radial cutoff in a pure NEP model introduces extra features that are not needed for describing the sliding energies and thus complicates the training process. Using too large a cutoff than needed, the construction of a MLP becomes more demanding since a large configuration space has to be explored and more descriptor components are needed to distinguish the atom environments [32]. Indeed, among the three pure-NEP models, NEP(6 Å, 4.5 Å) performs the best and NEP(10 Å, 4.5 Å) the worst regarding the sliding energies (figure 4(b)).

2.3. *Applications to heat transport in MOFs*

After confirming the reliability of the combined NEP-D3 approach, we show its usefulness in practical MD simulations. In a previous work [27], some of the present authors have studied heat transport in three typical MOFs (MOF-5, ZIF-8, and HKUST-1) using MD simulations with NEP models. The reference DFT data used for training these NEP models have no dispersion corrections. Because of the porous structures in MOFs, vdW interactions between the organic chains are expected to have noticeable effects in the structural and dynamic properties. Here we study the effects of dispersion interactions on the thermal conductivity in the MOFs.

We used the homogeneous non-equilibrium molecular dynamics (HNEMD) method [33] to calculated the thermal conductivity. The MD simulation details are consistent with the previous work [27] and the relevant inputs are given in appendix C. A driving force with a parameter $F_e = 2 \times 10^{-4}$Å^{-1} was used to drive the system out of equilibrium, and we have confirmed that it is sufficiently small to keep the system within the linear-response regime. To be consistent

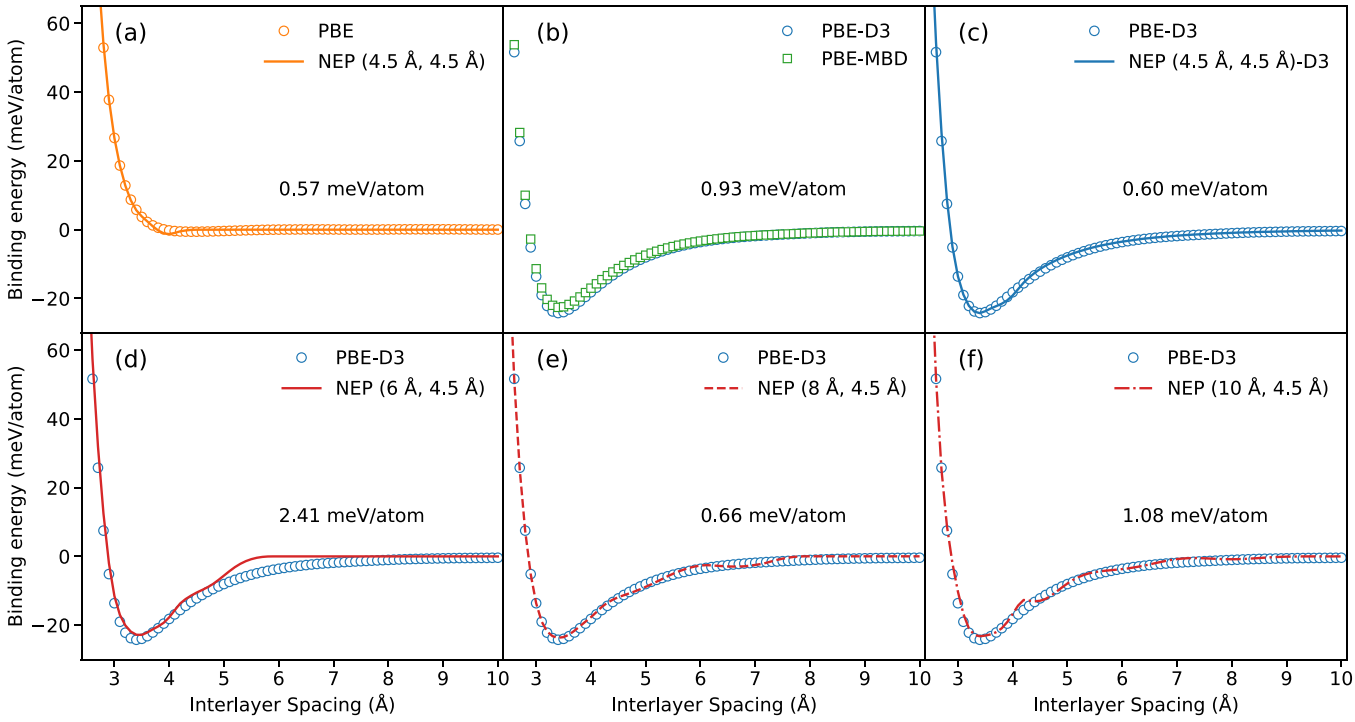

Figure 3. Binding energy of AB-stacked bilayer graphene as a function of the interlayer distance computed from various approaches as indicated in each panel. The RMSE between the two curves in each panel is indicated. See text for details.

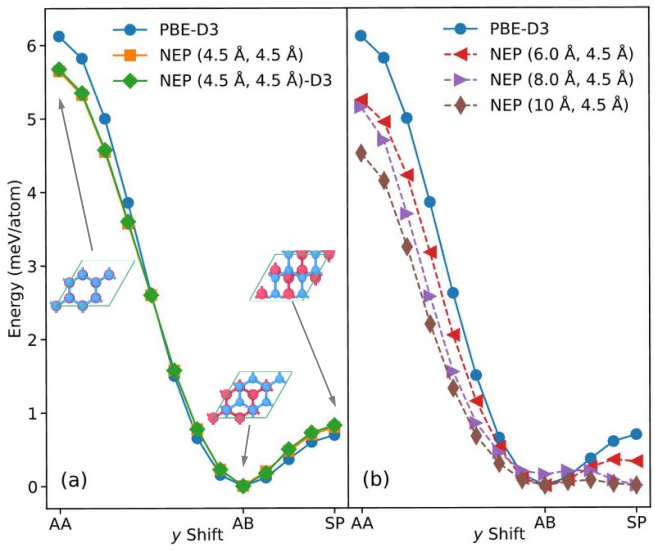

Figure 4. Sliding energies along the AA-AB-SP path of bilayer graphene with a interlayer spacing of 3.4 Å computed from various approaches. Atoms from the top and bottom layers have different colors.

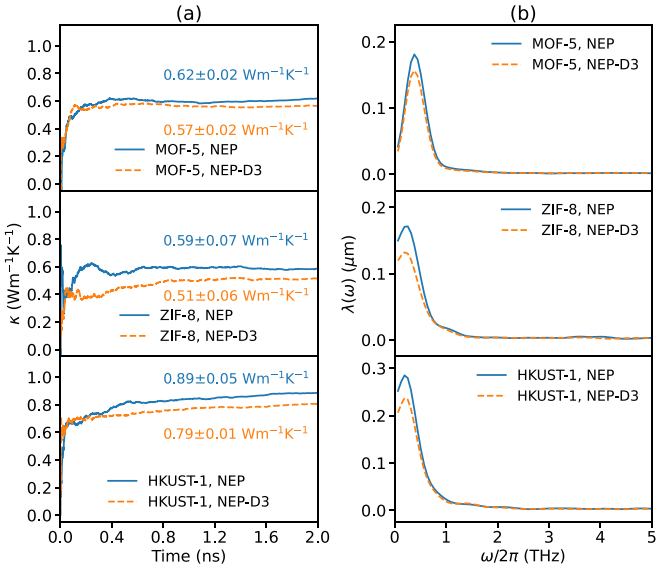

Figure 5. (a) Cumulative average of the thermal conductivity at 300 K as a function of simulation time and (b) spectral phonon mean free path as a function of phonon frequency for MOF-5 (top), ZIF-8 (middle), and HKUST-1 (bottom). The converged thermal conductivity values with error estimates (from five independent runs) obtained from NEP and NEP-D3 are indicated.

with the previous work [27], we have used a supercell with $5 \times 5 \times 5$ conventional cells (53 000 atoms), which was tested to be sufficiently large to eliminate finite-size effects. The system was first equilibrated in the isothermal-isobaric ensemble (300 K and zero GPa) to account for thermal expansion effects.

Figure 5 compares the thermal transport results at 300 K using the previously constructed pure NEP models [27] and the combined NEP-D3 models obtained by adding up D3 ($R_{\text{pot}} = 12$ Å, $R_{\text{cn}} = 6$ Å). For all the MOFs, the introduction of dispersion interactions consistently reduces the thermal

conductivity, and the average amount of reduction being about 10% (figure 5(a)). In the presence of dispersion interactions, the phonon mean free paths of the low-frequency phonons ($\omega/2\pi < 1$ THz) are reduced to some degree, but are still at the sub-micron scale (figure 5(b)). More importantly, the $\sim$10% reduction of the thermal conductivity in MOF-5 still leaves the discrepancy between the calculated ($0.57 \pm 0.02\,\mathrm{Wm^{-1}K^{-1}}$) and measured ($0.32\,\mathrm{Wm^{-1}K^{-1}}$) results [34] unresolved.

3. Summary and conclusions

In summary, we have made a GPU implementation of the D3 dispersion correction into the GPUMD package and enabled its integration with NEP to form a combined NEP-D3 model. We demonstrated the superior accuracy of the NEP-D3 approach than the pure NEP approach by using bilayer graphene systems as an example, for which the dispersion interactions between the two layers play an important role for the binding energies. Although the dispersion interactions are not responsible for the sliding energies between the two layers, the presence of D3 in NEP-D3 allows for using a relatively short cutoff in the NEP part, which indirectly leads to better description of the sliding energies by the NEP part. The D3 dispersion correction we implemented can be readily added to available NEP models that have been trained against reference data without considering dispersion correction. As an example, we showed that adding D3 dispersion correction to the previous NEP models for MOFs [27] leads to about 10% reduced thermal conductivity. The NEP-D3 approach is expected to find broad applications to a diverse range of materials for which dispersion interactions are important, such as molecular crystals, layered materials, polymers, and porous structures, elucidating the effects of dispersion interactions on physical properties such as thermal expansion, phase transition, and gas diffusion and adsorption.

Data availability statement

Complete input and output files for the NEP training of bilayer graphene and MOFs are freely available at https://gitlab.com/brucefan1983/nep-data. The source code and documentation for GPUMD are available at https://github.com/brucefan1983/GPUMD and https://gpumd.org, respectively. The D3 dispersion correction is available starting from GPUMD-v3.9.

Acknowledgments

P Ying was supported by the Israel Academy of Sciences and Humanities & Council for Higher Education Excellence Fellowship Program for International Postdoctoral Researchers. Z Fan was supported by National Natural Science Foundation of China (Project No. 11404033).

Conflict of interest

The authors declare that they have no competing interests.

Appendix A. Details on the DFT calculations

The DFT calculations were performed with the projector-augmented wave method [35] implemented in VASP [22, 23]. We set a threshold of 1×10^{-7} eV for the electronic self-consistent loop, with an energy cutoff of 850 eV for the plane-wave-basis set. We sampled the Brillouin zone using a Γ-centered grid with a k-point spacing of 0.15Å^{-1} and a Gaussian smearing with a width of 0.1 eV. The contents of the `INCAR` input file for VASP are given below.

```
GGA       = PE
LREAL     = Auto
ENCUT     = 850
IVDW      = 12 #remove this for PBE without D3
PREC      = Accurate
KSPACING  = 0.15
KGAMMA    = .TRUE.
ALGO      = Normal
NSW       = 1
IBRION    = -1
ISMEAR    = 0
SIGMA     = 0.1
EDIFF     = 1 × 10^-07
NELM      = 150
```

Appendix B. Inputs for training the NEP models

The NEP models can be trained using the `nep` executable in the GPUMD package. The relevant hyperparameters are specified in the `nep.in` input file. The contents of the `nep.in` input file for training the NEP(4.5 Å, 4.5 Å) model of bilayer graphene are given below. To train the NEP models with different radial cutoffs, one only needs to modify the parameters of the `cutoff` keyword.

```
type          1 C
version       3
cutoff        4.5 4.5
n_max         8 8
basis_size    12 12
l_max         4 2 0
neuron        50
lambda_1      0.05
lambda_2      0.05
lambda_e      1.0
lambda_f      1.0
lambda_v      0.1
batch         10 000
population    50
generation    300000
```

Appendix C. Inputs for thermal conductivity calculations

MD simulations with pure NEP models and NEP-D3 models can be performed by using the `gpumd` executable in the

GPUMD package. The controlling parameters are specified in the `run.in` input file. The contents of the `run.in` input file for calculating the thermal conductivity using a NEP-D3 model are given below. To switch off the D3 contribution, one just needs to remove the line starting with the keyword `dftd3`.

```
potential       nep.txt
dftd3           pbe 12 6
velocity        300

ensemble        npt_ber 300 300 100 0 10 1000
time_step       0.5
run             200 000

ensemble        nvt_nhc 300 300 100
compute_hnemd   1000 0 0 2e-4
compute_shc     10 500 2 500 200
run             4000000
```

ORCID iDs

Penghua Ying https://orcid.org/0000-0002-5758-2369
Zheyong Fan https://orcid.org/0000-0002-2253-8210

References

[1] Behler J 2016 *J. Chem. Phys.* **145** 170901
[2] Musil F, Grisafi A, Bartók A P, Ortner C, Csányi G and Ceriotti M 2021 *Chem. Rev.* **121** 9759–815
[3] Geim A and Grigorieva I 2013 *Nature* **499** 419–25
[4] Mandelli D, Ouyang W, Urbakh M and Hod O 2019 *ACS Nano* **13** 7603–9
[5] Walker A M, Civalleri B, Slater B, Mellot-Draznieks C, Corà F, Zicovich-Wilson C M, RománPérez G, Soler J M and Gale J D 2010 *Angew. Chem., Int. Ed.* **49** 7501–3
[6] Wieme J, Lejaeghere K, Kresse G and Van Speybroeck V 2018 *Nat. Commun.* **9** 4899
[7] Li J, Foster M E and Sohlberg K 2017 *Mol. Simul.* **43** 428–38
[8] Hofer T S, Listyarini R V, Hajdarevic E, Maier L, Purtscher F R S, Gamper J and Hanser F 2023 *J. Phys. Chem. Lett.* **14** 6018–27
[9] Wen M and Tadmor E B 2019 *Phys. Rev.* B **100** 195419
[10] Deringer V L, Caro M A and Csányi G 2020 *Nat. Commun.* **11** 1–11
[11] Rowe P, Deringer V L, Gasparotto P, Csányi G and Michaelides A 2020 *J. Chem. Phys.* **153** 034702
[12] Muhli H, Chen X, Bartók A P, Hernández-León P, Csányi G, Ala-Nissila T and Caro M A 2021 *Phys. Rev.* B **104** 054106
[13] Grimme S, Antony J, Ehrlich S and Krieg H 2010 *J. Chem. Phys.* **132** 154104
[14] Grimme S, Ehrlich S and Goerigk L 2011 *J. Comput. Chem.* **32** 1456–65
[15] Fan Z, Chen W, Vierimaa V and Harju A 2017 *Comput. Phys. Commun.* **218** 10–16
[16] Fan Z, Zeng Z, Zhang C, Wang Y, Song K, Dong H, Chen Y and Ala-Nissila T 2021 *Phys. Rev.* B **104** 104309
[17] Fan Z 2022 *J. Phys.: Condens. Matter.* **34** 125902
[18] Fan Z *et al* 2022 *J. Chem. Phys.* **157** 114801
[19] Behler J and Parrinello M 2007 *Phys. Rev. Lett.* **98** 146401
[20] Dong H, Cao C, Ying P, Fan Z, Qian P and Su Y 2023 *Int. J. Heat Mass Transfer* **206** 123943
[21] Eriksson F, Fransson E, Linderälv C, Fan Z and Erhart P 2023 Tuning the through-plane lattice thermal conductivity in van der Waals structures through rotational (dis)ordering *ACS Nano* (https://doi.org/10.1021/acsnano.3c09717)
[22] Kresse G and Furthmüller J 1996 *Phys. Rev. B* **54** 11169
[23] Kresse G and Joubert D 1999 *Phys. Rev. B* **59** 1758
[24] Fan Z, Pereira L F C, Wang H-Q, Zheng J-C, Donadio D and Harju A 2015 *Phys. Rev.* B **92** 094301
[25] Gabourie A J, Fan Z, Ala-Nissila T and Pop E 2021 *Phys. Rev.* B **103** 205421
[26] Liu J, Byggmästar J, Fan Z, Qian P and Su Y 2023 *Phys. Rev.* B **108** 054312
[27] Ying P, Liang T, Xu K, Zhang J, Xu J, Zhong Z and Fan Z 2023 *ACS Appl. Mater. Interfaces* **15** 36412–22
[28] Brenner D W, Shenderova O A, Harrison J A, Stuart S J, Ni B and Sinnott S B 2002 *J. Phys.: Condens. Matter.* **14** 783
[29] Ouyang W, Mandelli D, Urbakh M and Hod O 2018 *Nano Lett.* **18** 6009–16
[30] Perdew J P, Burke K and Ernzerhof M 1996 *Phys. Rev. Lett.* **77** 3865
[31] Tkatchenko A, DiStasio R A, Car R and Scheffler M 2012 *Phys. Rev. Lett.* **108** 236402
[32] Tokita A M and Behler J 2023 *J. Chem. Phys.* **159** 121501
[33] Fan Z, Dong H, Harju A and Ala-Nissila T 2019 *Phys. Rev.* B **99** 064308
[34] Huang B, McGaughey A and Kaviany M 2007 *Int. J. Heat Mass Transfer* **50** 393–404
[35] Weinan E 1994 *Phys. Rev.* B **50** 17953