ROYAL SOCIETY OF CHEMISTRY

PAPER

Check for updates

Embedding material graphs using the electron-ion potential: application to material fracture†

Sherif Abdulkader Tawfik, ‡*[ab] Tri Minh Nguyen, ‡[a] Salvy P. Russo, [c] Truyen Tran, [a] Sunil Gupta [a] and Svetha Venkatesh [a]

At the heart of the flourishing field of machine learning potentials are graph neural networks, where deep learning is interwoven with physics-informed machine learning (PIML) architectures. Various PIML models, upon training with density functional theory (DFT) material structure–property datasets, have achieved unprecedded prediction accuracy for a range of molecular and material properties. A critical component in the learned graph representation of crystal structures in PIMLs is how the various fragments of the structure's graph are embedded in a neural network. Several of the state-of-art PIML models apply spherical harmonic functions. Such functions are based on the assumption that DFT computes the Coulomb potential of atom–atom interactions. However, DFT does not directly compute such potentials, but integrates the electron–atom potentials. We introduce the direct integration of the external potential (DIEP) methods which more faithfully reflects that actual computational workflow in DFT. DIEP integrates the external (electron–atom) potential and uses these quantities to embed the structure graph into a deep learning model. We demonstrate the enhanced accuracy of the DIEP model in predicting the energies of pristine and defective materials. By training DIEP to predict the potential energy surface, we show the ability of the model in predicting the onset of fracture of pristine and defective carbon nanotubes.

1 Introduction

In the last few years, machine learning (ML) potential models have been amassing an unprecedented number of contributions from interdisciplinary research teams worldwide. The capabilities of these models rapidly expanded into various material science applications, promising a future where highly accurate quantum material science computations can be performed at the cost of classical molecular dynamics, if not at a lower cost. The accuracy and generalisability of the models have been empowered by two key factors: the emergence of graph neutral network (GNN)[1] models that superseded standard ML models in accuracy and complexity,[2] and the abundance of a massive amount of quantum mechanically-computed

material data in online databases. Over the last few years, the online databases, such as Materials Project (MP),[3] JARVIS,[4] AFLOW,[5] Open Quantum Mechanical Database (OQMD),[6] and others, have availed more than 3 million DFT-computed structures, and have been part of a standard benchmarking workflow for new ML models. GNNs have further been improved by the introduction of physical laws within the fabric of the neural network, establishing what is commonly known as physics-informed machine learning (PIML) models, as was demonstrated in DimeNet,[7] its derivative M3GNET[8] and MACE[9] for property prediction and structure discovery. These models have explicitly incorporated physics-based representations for the atomic structure by transforming doublets (atom–atom pairs) using a radial basis function (RBF), and triplets (groups of three atoms within a sphere of a given radius) using spherical harmonics. The utilization of spherical harmonics stemmed from the analytical structure of the wave function, which is the solution of the Schrödinger equation that DFT aims to approximate. Trained on over 133k samples, the M3GNET achieved a mean absolute error (MAE) of 20 meV per atom for predicting the formation energy (E_f) of the test set samples. However, as will be shown in this work, the model struggles with the prediction of the properties of defective crystals, and in predicting the onset of carbon nanotube fracture.

We propose a novel physics-informed ML framework for accurate prediction of both pristine and imperfect crystal

[a] Applied Artificial Intelligence Institute, Deakin University, Geelong, Victoria 3216, Australia. E-mail: s.abbas@deakin.edu.au; tri.nguyen1@deakin.edu.au; svetha.venkatesh@deakin.edu.au; sunil.gupta@deakin.edu.au

[b] ARC Centre of Excellence in Exciton Science, Deakin University, Geelong, Victoria 3216, Australia

[c] ARC Centre of Excellence in Exciton Science, School of Science, RMIT University, Melbourne 3000, Australia
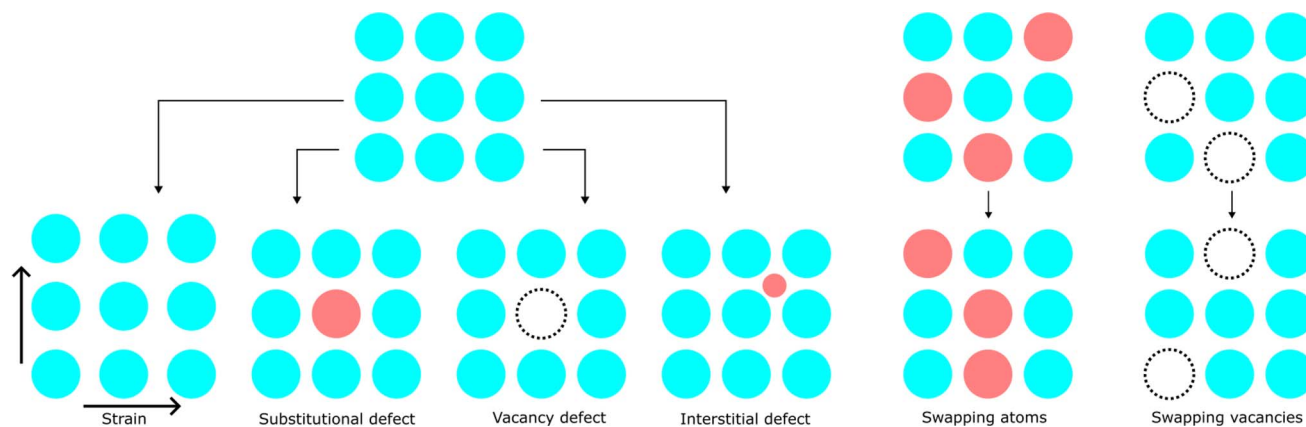
Fig. 1 The types of crystal deformation that are examined in this work: strain deformation, where the crystal lattice size if modified to reflect the application of either a stretching or compressive force on the structure; substitutional defect, where one of the atoms is substituted with another; vacancy defect, where an atom is removed from the crystal leaving a vacancy; interstitial defect, where an atom is introduced into a void in the crystal; swapping of atoms, where atoms within the crystal exchange positions; swapping vacancies, where vacancies, rather than atoms, are exchanged.

materials. Called DIEP, it directly integrates the external potential of a structure, and it was implemented on the code-base of M3GNET. This direct integration is a critical correction to the physics in M3GNET, and we show its value in discerning the impact of material defects. In particular, DIEP is able to more accurately predict the total energy per atom of a defective system, as well as the structural changes that result from the presence of a defect in a material. We demonstrate these two merits of the DIEP model by conducting two learning tasks using datasets of crystalline systems: in task 1, we train the models to predict the total energy per atom and demonstrate the accuracy of DIEP in predicting the total energy per atom for 6 classes of material imperfections, which are displayed in Fig. 1. We further interrogate the models on common defects of diamond, and show that the accuracy of DIEP exceeds that of the trained M3GNET model in most of the test cases. In task 2, we train a potential model that predicts both the energy per atom and the atomic forces, which amount to the prediction of the potential energy surface (PES). We show the ability of DIEP in (1) reproducing the ground state crystal structure of a number of binary materials, and (2) in computing the fracture strain of a large carbon nanotube structure due to the presence of carbon vacancy defects, while the M3GNET model does not predict rupture of the CNT at excessive elongation strains.

2 Improving the physics

The PIML in M3GNET is based on embedding the graph nodes (including the atom–atom pairs, or bonds, and the three-atom structures, or triplets) into the neural network through a layer that calculates DFT-related properties, as was initially proposed in the DimeNET model. The DFT-related properties assume that the atoms in the structure are directly coupled by a Coulomb potential in a simplified Schrödinger equation, and embed the nodes by using the solutions of that equation as functions of the bonds and triplets in the structure. However, the direct atom–atom interaction emulated by the M3GNET is in fact neglected

in typical DFT calculations. This is because, based on the Born–Oppenheimer approximation, the positions of atoms are treated as fixed. In this work, we introduce an alternative PIML approach that is based on the numerical integration of the external potential which is directly relevant to DFT. We embed the graph nodes using an integration layer, in which the atom–electron potential, rather than the atom–atom potential, is computed. Computing an integration for atom–electron inter-action is advantageous because the atom–electron interaction uniquely determines the ground state DFT electron density $\rho(\mathbf{r})$ (a direct implication of the Honenberg–Kohn theorem[10]), where $\rho(\mathbf{r})$ is the function that enables the determination of the ground state total energy. Our method computes a simplified form for this term for the graph's bonds and triplets on a two-dimensional mesh, and passes these atom–electron interac-tion messages into the GNN.

The total DFT energy is calculated by summing the kinetic energy $T[\rho(\mathbf{r})]$, atom–electron energy (or external energy) $E_{\text{ext}}[\rho(\mathbf{r})]$, the Hartree energy $E_{\text{H}}[\rho(\mathbf{r})]$, and the exchange corre-lation energy $E_{\text{XC}}[\rho(\mathbf{r})]$, which are all functionals of the electron density $\rho(\mathbf{r})$:

$$E_{\text{T}}[\rho(\mathbf{r})] = T[\rho(\mathbf{r})] + E_{\text{ext}}[\rho(\mathbf{r})] + E_{\text{H}}[\rho(\mathbf{r})] + E_{\text{XC}}[\rho(\mathbf{r})] \quad (1)$$

where the terms are given by

$$T[\rho(\mathbf{r})] = -\frac{1}{2}\sum_{n\mathbf{k}}^{N}\int d\mathbf{r}\phi^*(\mathbf{r})\nabla^2\phi(\mathbf{r})$$

$$E_{\text{ext}}[\rho(\mathbf{r})] = \int d\mathbf{r}\sum_{N}^{n}\frac{\rho(\mathbf{r})Z_n}{|\mathbf{r}-\mathbf{R}_n|} \quad (2)$$

$$E_{\text{H}}[\rho(\mathbf{r})] = \frac{1}{2}\iint d\mathbf{r}d\mathbf{r}'\frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}$$

The value of E_{T} must be minimized with respect to the given atomic structure, and therefore one must find a density func-tion $\rho(\mathbf{r})$ that will yield such minimum energy. Given the

definition of the electron density in terms of the fictitious electronic orbital functions $\phi_i(\mathbf{r})$

$$\rho(\mathbf{r}) = \sum_i^{N_e} \phi_i^*(\mathbf{r})\phi_i(\mathbf{r})$$

where $\phi_i^*(\mathbf{r})$ is the complex conjugate of $\phi_i(\mathbf{r})$, the minimization problem is solved by applying the variational principle to E_T with respect to the orbital functions $\phi_i(\mathbf{r})$. This yields the Kohn–Sham equations,

$$H_{KS}\phi_i(\mathbf{r}) = E_i\phi_i(\mathbf{r}) \tag{3}$$

where the Kohn–Sham Hamiltonian H_{KS} is given by

$$H_{KS} = -\frac{1}{2}\nabla^2 + V_{ext}(\mathbf{r}) + V_H(\mathbf{r})$$

$$V_{ext}(\mathbf{r}) = \sum_N^n \frac{Z_n}{|\mathbf{r} - \mathbf{R}_n|} \tag{4}$$

$$V_H(\mathbf{r}) = \frac{1}{2}\int d\mathbf{r}' \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}$$

The only term in H_{KS} that is directly dependent on the positions of the atoms is $V_{ext}(\mathbf{r})$. Likewise, the only term in $E_T[\rho(\mathbf{r})]$ that is directly dependent on the positions of the atoms is $E_{ext}[\rho(\mathbf{r})]$. These terms, however, do not use direct atom–atom distances, contrary to the assumption made in the M3GNET and DimeNet models.

In a GNN, structural data are generated for the input structure into a graph object with nodes and edges. For a given structure, nodes can represent information about atoms at the nodes' spacial positions $\mathbf{R}_i$, and edges can represent the bonds between the atoms. The structure of the graph, as well as the count of nodes and edges, is different for each structure. In a GNN, the NN learns the graph connection between each node and the neighboring nodes iteratively. Following the extraction of bonds and triplets from the graphical representation of the input crystal structure, which is routinely performed by common GNN-based packages, DIEP performs the following standardization transformation on bonds and triplets, which is depicted in Fig. 2:

• for a bond between atoms with atomic numbers Z_i and Z_j, positions $\mathbf{R}_i$ and $\mathbf{R}_j$, the line joining the points $\mathbf{R}_i$ and $\mathbf{R}_j$ is transformed into the line joining the two points $-\frac{d_{ij}}{2}$ and $\frac{d_{ij}}{2}$, where d_{ij} is the distance between $\mathbf{R}_i$ and $\mathbf{R}_j$;

• for a triplet involving three atoms with atomic numbers Z_i, Z_j and Z_k, positions $\mathbf{R}_i$, $\mathbf{R}_j$ and $\mathbf{R}_k$, the triangle $\mathbf{R}_i$—$\mathbf{R}_j$—$\mathbf{R}_k$ is transformed to $\mathbf{R}_a$—$\mathbf{R}_b$—$\mathbf{R}_c$ such that $\mathbf{R}_a$—$\mathbf{R}_b$ is the longest side, followed by $\mathbf{R}_b$—$\mathbf{R}_c$ then $\mathbf{R}_a$—$\mathbf{R}_c$.

For a transformed bond, identifying either atoms as Z_i or Z_j is the same, whereas for a triplet, we set the identities of the atoms in the transformed triangle based on the intersection between the identities of the atom pairs in the bonds of the triangle. This transformation ensures that the transformed structure is invariant to permutation of the identities of atoms.
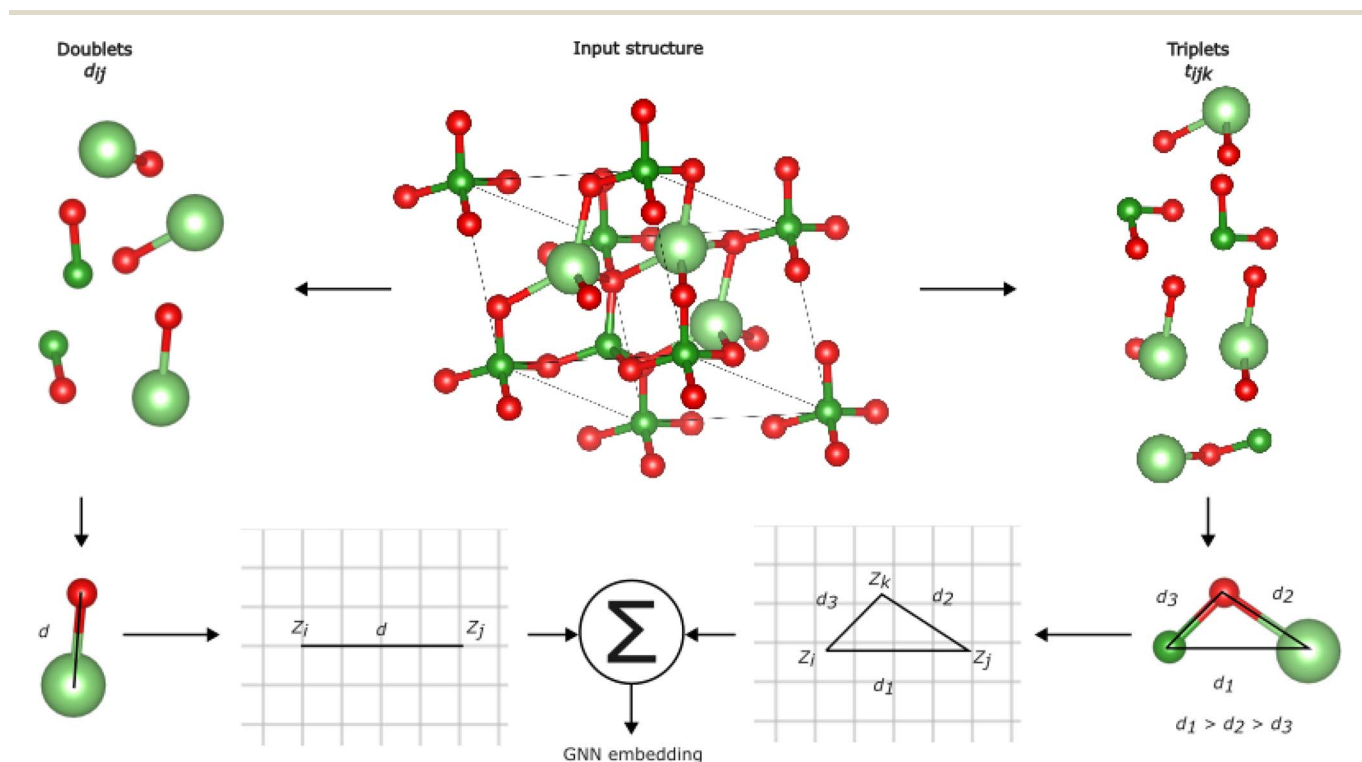

Fig. 2 Outline of the DIEP method and characteristics of the TinyUnitCells dataset. The input crystal structure is decomposed into its constituent bonds d_{ij} and triplets t_{ijk}. Then, each bond is transformed into a line that is centred in a sparse 2D mesh, and each triplet is transformed into a triangle with ordered side lengths, and is also centred in a sparse 2D mesh. The DIEP integration is applied to each d_{ij} and t_{ijk} in their respective 2D meshes, and the value of each integration is used to embed the d_{ij} and t_{ijk} of the input structure.

Both the transformed bond and triangle are then centred within a two-dimensional grid where the x- and y axes range from $-L$ to $+L$ (in units of Å), with step size Δl. The transformation of the triangle $\mathbf{R}_i$—$\mathbf{R}_j$—$\mathbf{R}_k$ ensures that any three atoms at positions $\mathbf{R}_i$, $\mathbf{R}_j$ and $\mathbf{R}_k$ will have a unique value for E_{ijk}.

The 2D grid in DIEP is a simplification for the actual 3D grid in which the external potential is integrated; it is only a 2D "slice" of the 3D box. The points on the mesh represent the coordinates $\mathbf{r}$ of the electronic density function $\rho(\mathbf{r})$. The representation of a triplet as a triangle on a mesh is an explicit incorporation of three-body interactions that combines the properties of the atoms (atomic numbers Z_i) with their positions in a natural way, rather than embedding the triplet using bond angles that are separated from the atomic numbers as in M3GNET.

To embed a transformed bond, we compute the following terms, which are based on the external energy term $E_{\text{ext}}[\rho(\mathbf{r})]$ in eqn (3):

$$E_{ij,mn} = \Delta l^2 \rho(\mathbf{r}_{mn}) \left(\frac{Z_i}{\left|\mathbf{r}_{mn} - \frac{d_{ij}}{2}\right|} + \frac{Z_j}{\left|\mathbf{r}_{mn} + \frac{d_{ij}}{2}\right|} \right), \quad (5)$$

whereas for triplets, the following quantities are computed:

$$E_{ijk,mn} = \Delta l^2 \rho(\mathbf{r}_{mn})$$
$$\times \left(\frac{Z_i}{|\mathbf{r}_{mn} - \mathbf{R}_a|} + \frac{Z_j}{|\mathbf{r}_{mn} - \mathbf{R}_b|} + \frac{Z_k}{|\mathbf{r}_{mn} - \mathbf{R}_c|} \right). \quad (6)$$

where $\rho(r_{mn})$ represents a simple analytical form for the electron density, which is given by

$$\rho(\mathbf{r}_{mn}) = \sum_i^N e^{-\frac{|\mathbf{r}_{mn} - \mathbf{R}_i|^2}{\sigma}} \quad (7)$$

and m and n are indices from the position of a point in the 2D grid, and the grid is of size $L \times L$. The function $\rho(\mathbf{r})$ is the optimization target in DFT, and therefore its relationship with the atomic structure is highly complicated. One can use a simplified or trained[11] analytical form for $\rho(\mathbf{r})$. However, we choose a simplified representation: a summation of Gaussian functions centred at the atomic positions. This representation is based on the trainable expression (eqn (1)–(3)) in ref. 11 in which the electron density is expanded as sum of Gaussian functions with trainable coefficients.

While this charge density representation is not representative for a large number of bonds, particularly in charge transfer situations where one of the atom loses a significant portion of its charge density (such as in the case of transition metals), it is still a reasonable assumption for the broad variety of bonds, as well as being computationally efficient. An improvement to the charge density representation would either use parametrised expressions such as the one in ref. 11 or to express the charge density as a sum of gaussians with learnable parameters. Those learnable parameters would then be tuned as part of the entire training cycles of the DIEP model. Even though these trained 2D fragment charge densities will not be as complete as the densities that are obtained from models trained on actual

densities, such as ref. 12–14 they would be expected to model the densities of their respective fragments. Exploring alternative strategies to represent $\rho(\mathbf{r})$ is currently in progress.

Following the computation of the quantities $E_{ij,mn}$ and $E_{ijk,mn}$, these quantities are then used to embed the bonds ij and triplets ijk into the neural network. This by performing the following numerical integration across the grid:

$$E_{ij} = \sum_{m=1}^{L} \sum_{n=1}^{L} E_{ij,mn},$$
$$E_{ijk} = \sum_{m=1}^{L} \sum_{n=1}^{L} E_{ijk,mn} \quad (8)$$

The quantities E_{ijk} and E_{ij} are then fed into the GNN. These quantities replace the 2D representation quantities $a_{\text{SBF}}^{(kj,ji)}$ and the distance representation quantities $e_{\text{RBF}}^{(ji)}$ in the DimeNet embedding layer (Fig. 4 of ref. 7). The total number of trainable parameters in the DIEP model is 279837, whereas it is 288 157 in the M3GNET model. Hence, an improvement in the DIEP model is not due to introducing additional learnable parameters. The DIEP hyperparameters include: the resolution of the grid (the values of L and Δl), the choice of using the external potential or external energy terms, whether to use the entire grid points or to use the summation of terms, and the value of σ in the exponential formula for $\rho(\mathbf{r})$.

Next, To examine the predictive power of DIEP compared with M3GNET, we perform two training tasks. In the first task, we train DIEP and M3GNET models on a subset of stable materials from MP, and then challenge the generalisability the two models on deformed materials which are described in Fig. 5. In the second task, we train the M3GNET and DIEP models on a dataset of structural optimisation trajectories (the MPF.2021.2.8 dataset used in ref. 8) and compare the accuracy of predicting the total energy per atom and the atomic forces.

3 Results and discussion

3.1 Prediction of the energy of defective materials

3.1.1 Defective materials. We design material deformations with structural features that are significantly different from the materials in TinyUnitCells. We consider deformations that are often encountered in routine material science investigations, as well as those generated in the course of material discovery processes. The classes of deformations we examine are:

• Random strains: for a sample of unit cells with a maximum of 4 atoms, a uniaxial strain is applied, where we randomly pic a strain value from the set of percentage strain values $\pm 1\%$, $\pm 2\%$, $\pm 3\%$, $\pm 4\%$ and $\pm 5\%$ along the a lattice direction. For another sample of unit cells with a maximum of 4 atoms, we applied random triaxial strains, where we randomly pic a strain value from the set of percentage strain values $\pm 1\%$, $\pm 2\%$, $\pm 3\%$, $\pm 4\%$ and $\pm 5\%$ along each of the axial directions.

• Single-site defects: for a sample of unit cells with a maximum of 4 atoms, a point defect is created in $2 \times 2 \times 1$ supercells by either removing an atom, or substituting an atom with another, such that the two atoms are in the same group of

the periodic table. This ensures the same number of valence electrons in the unit cell and avoid the complexity of dealing with potentially charged defects.

• Swapping of atomic positions: this disorder naturally occurs in high entropy alloys, where the distribution of the atoms within the lattice structure is rather stochastic.[15]

• Multi-site defects: a permutation of multiple vacancies is created in a supercell. In some cases, the permutations do not result in significant changes in the total energy. In this case, this class of deformations examines the false positives: whether the ML model will overestimate the energy cost of a small geometric change.

• Interstitial defects: hydrogen atoms are introduced into interstitial sites of metals and perovskites.

To examine the ability of DIEP and M3GNET in predicting the energies of the above defects, we first train both models on a sample of pristine materials from MP with up to 10 atoms in the unit cell, TinyUnitCells (version 1). The composition of this dataset is: 10k materials with up to 4 atoms per unit cell, 1k with 5 or 6 atoms per unit cell, and 1k materials with 7–10 atoms per unit cell. The representation of each of the elements in TinyUnitCells is displayed in Fig. 3a. There are 88 elements represented in the dataset (out of 89 elements that are represented in MP), the most frequently occurring elements are O, Li and Mg. The large representation of O in TinyUnitCells is similar to the case of the larger dataset MPF.2021.2.8.[8]

3.1.2 TinyUnitCells results. We have re-computed the DFT structural optimisation for TinyUnitCells using standardised DFT input parameters (details in the Methods section). The size of unit cells in TinyUnitCells are chosen in order to bias the GNN on learning features of small unit cells, and hence to test the ability of the method to generalise to larger structures, particularly those supercells with a small concentration of defects. The histogram in Fig. 3b illustrates the scale of such difference in terms of the number triplets extracted from each structure in the dataset. Small unit cells (such as those in TinyUnitCells) possess much lower triplets than larger structures, such as the defect structures examined here. The 80–10–10% split is also applied for the TinyUnitCells dataset. DIEP exceeds the accuracy of M3GNET in predicting the total energy per atom of the test set, as displayed in Fig. 4b. Note that the MAE values are higher than those obtained for the JARVIS dataset because the size of TinyUnitCells is nearly one-fifth that of JARVIS. After establishing the accuracy of training the DIEP models for the pristine materials in TinyUnitCells, we demonstrate its accuracy for predicting the defective crystals.

3.1.3 Prediction of the total energy pert atom. The results of running the two models on the 6 classes of deformed materials is displayed in Table 1. DIEP performs better than M3GNET in reproducing the total energy per atom for most of the defect classes considered.

DIEP outperforms M3GNET in predicting the energies of strained unit cells, whether the strain is triaxial, uniaxial optimised or uniaxial unoptimised, as shown in Table 1. We further examine the optimised uniaxial strained structures by observing the errors obtained for the strains ±1%, ±2%, ±3%, ±4%, and ±5%. The MAE for the DIEP model for each group of
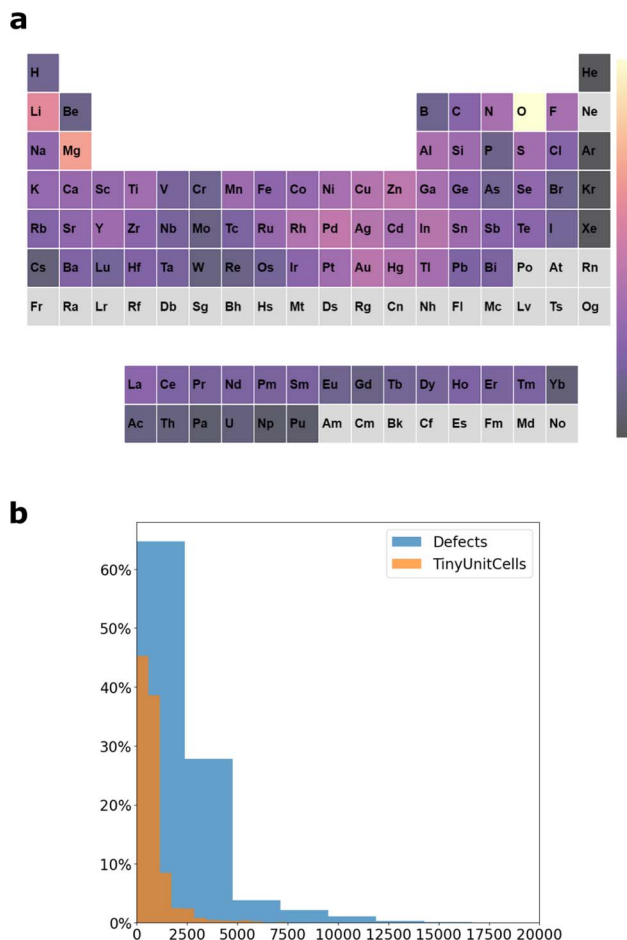
Fig. 3 Dataset characteristics. (a) Periodic table displaying a heatmap of the frequency of occurrence of each element in the periodic table in the TinyUnitCells dataset. (b) Comparison between the number of triplets extracted for materials in the TinyUnitCells dataset (≤10 atoms), and defective materials with > r_{bin} 10 atoms, showing the large difference in the distribution of geometric information between the two sets of materials.

strains gradually increase with strain, as expected: 37 meV per atom, 45 meV per atom, 56 meV per atom, 68 meV per atom and 94 meV per atom. Except for the ±4% strain, these errors are consistently lower than those obtained using the trained M3GNET model: 44 meV per atom, 53 meV per atom, 62 meV per atom, 67 meV per atom and 101 meV per atom.

For single-site defects, DIEP is only 7 meV per atom better than M3GNET. Both models can accurately predict substitutional defects, where an atom is replaced by another atom that is one row above or below it in the periodic table (such as F replaced with Cl). They both struggle to predict the energy of substitutional defects where a small atom is substituted with a larger one that is 2 or more rows lower in the periodic table. For example, the DFT-calculated total energy per atom for BN (mp-1580) with a N atom substituted with Bi is −7.90 eV per atom, but DIEP predicted a value of −9.87 eV per atom and M3GNET predicted −9.74 eV per atom. Vacancy defects are generally accurately predicted.
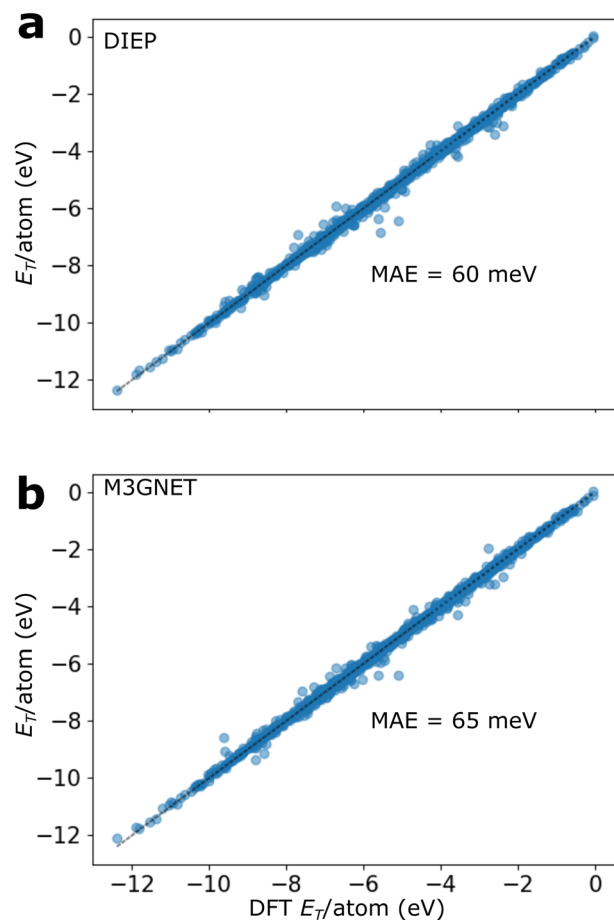
Fig. 4 Correlation plots for the ML prediction of the total energy per atom for the structures in TinyUnitCells using (a) DIEP and (b) M3GNET.

For alloy structures with swapped atomic positions, atomic swapping leads to mild changes in the total energy per atom in the PtPdIrRh, CoCrFeNi and PtIr alloy systems, and significant changes in the FeNi alloy system. The range of energies (difference between the highest and lowest values) in these alloy systems is 44 meV per atom, 73 meV per atom, 21 meV per atom and 354 meV per atom, respectively. Such spread in the data affected the predictive accuracy of both models. We also examine the swapping of multiple vacancy sites in the cathode materials $LiCoO_2$ and $LiFePO_4$, and $LiAlO_2$, a cathode coating material. Lithium diffusion is key to the function of these materials, and it involves the presence of various defect configurations at any instance of time. We swap 3 defect sites in $LiCoO_2$, 4 in $LiFePO_4$ and 3 in $LiAlO_2$. The structures for these defects were obtained by enumerating all the possible symmetrically unique combination of defects using the python library bsym. Swapping of defects in these materials caused very small changes in the total energy per atom: the energies vary within a range of 40 meV per atom. In all three cases, both methods were able to reflect the small variance in energies, with DIEP performing better than M3GNET in 2 of the 3 cases.

Next, we consider H interstitial defects. An extra H is inserted into a crystal void of metallic structures and perovskites using the add_interstitial() function from the OgStructure class in the oganesson python package. The presence of interstitial H in metals is one of the key causes of their embrittlement,[16] and occurs during the diffusion of hydrogen in perovskites en route of water splitting. Predicting the energy of a single interstitial atom in both metals and perovskites was challenging for DIEP and M3GNET. For metals, the highest errors were due to H interstitial defects in crystal with triclinic

Table 1 The mean absolute error (MAE) in meV per atom for predicting the total energy per atom of the deformed materials. The atomic structures of all the deformation classes have been relaxed except for the "unoptimised uniaxial" strains

Deformation		Dataset size	DIEP MAE (meV per atom)	M3GNET MAE (meV per atom)
Strains	Triaxial	404	**90**	94
	Optimised uniaxial	929	**59**	65
	Unoptimised uniaxial	943	**61**	66
Single-site defects	Vacancies	213	**78**	79
	Substitutions	196	**108**	120
Atomic swaps	PtPdIrRh	257	**41**	55
	CoCrFeNi	263	90	**45**
	FeNi	123	328	**287**
	PtIr	268	**41**	43
Multi-vacancy swaps	$LiCoO_2$	53	**53**	112
	$LiAlO_2$	53	96	**11**
	$LiFePO_4$	70	**195**	231
Single interstitial H defect	Metals	246	97	**90**
	Perovskites	141	**91**	101
Multiple interstitial H defects	Zn	18	**48**	103
	Ni	14	**24**	35
	Fe	10	**136**	180
	Pt	10	**56**	144

symmetry (MAE for DIEP is 110 meV per atom, for M3GNET is 160 meV per atom). While DIEP is more accurate than M3GNET in predicting the energies of cubic and triclinic crystals, M3GNET surpassed the accuracy of DIEP for monoclinic (DIEP: 86 meV per atom, M3GNET: 80 meV per atom), orthorhombic (DIEP: 99 meV per atom, M3GNET: 92 meV per atom), trigonal (DIEP: 105 meV per atoms, M3GNET 73 meV per atom) and tetragonal (DIEP: 101 meV per atom, M3GNET: 86 meV per atoms) crystals. Adding more interstitial H atoms in a selection of metals (Fe, Pt, Ni and Zn) improved the accuracy of the DIEP model, as shown in the bottom of Table 1.

3.1.4 Diamond defects. We test the predictions of DIEP and M3GNET for the following neutral diamond defects against the DFT values: nitrogen vacancy (N-V),[17] oxygen vacancy (O-V),[18] phosphorus vacancy (P-V), germanium vacancy (Ge-V),[19] tin vacancy (Sn-V),[20] silicon vacancy (Si-V),[21] sulfur vacancy (S-V),[22] substitutional boron (B_C),[23] substitutional nitrogen (N_C),[23,24] substitutional oxygen (O_C),[18] interstitial hydrogen (H_i),[25] and interstitial carbon (C_i).[26] Fig. 5a shows that DIEP achieves better

predictive accuracy for all of these defects than M3GNET, except for the C_i defect where the two models are close. The most challenging defect for both models is C_i, owing to the presence of bond distances that lie outside of the distribution of bonds in TinyUnitCells. We further examine the impact of combining two defects within the supercell. We generate a permutation of P-V, P-V, S-V, Sn-V, Si-V, N-V, O-V with B_C, N_C and O_C (a total of 18 two-site defects). The accuracy of DIEP in this set surpasses that of M3GNET with an MAE of 22 meV versus 196 meV, respectively.

Further, we examine the impact of dilution on the quality of the predictions. Defect dilution means that a single defect is created in increasingly large supercells. Here we focus on the single carbon vacancy defect. Fig. 5b displays the prediction error for the defect, starting with the defect in a $2 \times 2 \times 2$ diamond supercell, up to $3 \times 3 \times 3$. The MAE of the prediction of both M3GNET and DIEP reduces as the defect becomes more dilute, with DIEP highly surpassing the accuracy of M3GNET at all defect concentrations considered.

3.2 Prediction of the potential energy surface

Training an ML model to predict the potential energy surface (PES)[27] requires the ability of the model to predict the potential energy as well as the atomic force vectors. The equivalence between the negative of the neural network's gradients with respect to the atomic coordinates and the forces on those atoms along those coordinates, is a natural connection between neural learning and the PES that was observed by Behler and Parrinello[28] in their early work on neural network regression of material properties. To train the models, we use the MPF.2021.2.8 dataset which includes the total energy per atom, atomic forces and lattice stresses for nearly 188k structures. We applied the same network architecture as that used for M3GNET training. For the network parameters l_{max} and n_{max}, we test the following combination of values to find the combination (l_{max}, n_{max}) that will yield the lowest validation error for the total energy per atom: (2, 1), (2, 2) and (3, 3). We applied a training-validation-test partitioning of 90%, 5%, 5% and only included structures with atomic forces within the range −10 and 10 eV Å^{-1}, following the procedure in ref. 29. We label the PES trained model DIEP-PES, and compare its performance against the M3GNET-PES model released as part of the `matgl` python library (`M3GNet-MP-2021.2.8-PES`).[8] We train the DIEP-PES model by optimising the total energy per atom and the atomic forces, without optimising the lattice stress. The combination (l_{max}, n_{max} that gave the lowest validation MAE for the total energy per atom was (3, 3), where the validation set error is 41 meV per atom. For the test set, the DIEP-PES models achieves a total energy per atom MAE of 61 meV per atom and a force MAE of 73 meV Å^{-1}. The total energy per atom MAE for DIEP-PES is higher than that of the M3GNET-PES (which is 34 meV per atom) whereas the force MAE of the DIEP-PES is close to that of the M3GNET-PES (70 meV Å^{-1}).

To assess the computation cost for running a PES calculation, we performed a single-point calculation with both DIEP-PES and M3GNET-PES on each structure of the entire MP
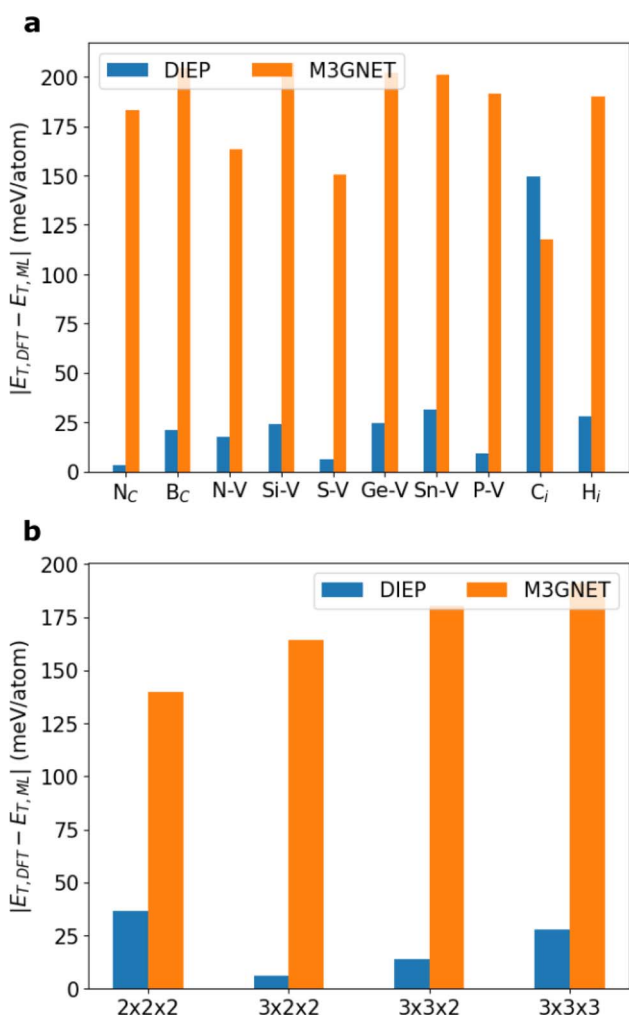

Fig. 5 Prediction results for deformations. (a) The MAE for predicting the total energy per atom for each of the 10 diamond defects. (b) An examination of the influence of defect dilution on the quality of the prediction.

database (155k structures). The average execution time of DIEP-PES on 1 GPU processor is 5.3 milliseconds per atom, while that of M3GNET-PES is 5.5 milliseconds per atom. We further examined the computational cost comparison by performing single-point calculations with DIEP-PES, M3GNET-PES and DFT (using VASP) for a random selection of 96 structures from the MP database where the number of atoms exceeds 30 atoms. The mean number of atoms in the set is 59 atoms. The DIEP-PES on 1 GPU processor consumed 1.4 milliseconds per atom, M3GNET-PES 1.3 milliseconds per atom, while the DFT calculation on 96 CPU cores consumed 18.1 seconds per atom. This shows that the computational complexities of the two methods are very close, and that the computational performance of the ML methods exceeds that of DFT by 3 orders of magnitude.

The computational performance per atom improves with larger number of atoms. To see how the structure size influences the computational performance, we perform a single-point calculation on a supercell of diamond with size $N \times 1 \times 1$ where N takes values from 1 to 100. The number of atoms at $N = 1$ is 8. The result is displayed in Fig. 6, showing that the computational scaling behaviours of both DIEP-PES and M3GNET-PES are very close.

3.2.1 Structure discovery using genetic algorithms. We examine the ability of DIEP-PES to re-discover the stable crystal structure for a number of binary compounds, by employing the genetic optimisation library available in the `oganesson` python package which wraps the genetic library in `ase`. We compare the number of compounds that we successfully re-discovered by DIEP-PES against those re-discovered by M3GNET. We display the results in Table 2.

We examine the ability of DIEP-PES to re-discover the stable crystal structure for a number of binary compounds with an energy above hull of zero eV per atom. We utilise the `oganesson GA` class which is a wrapper for the genetic algorithm code within the `ase` python library. We limit the genetic optimisation process to an initial population of 20 structures, and 5 evolutions where 20 candidate structures are generated every evolution. The genetic algorithm uses the total DFT energy as the fitness function. The following genetic evolution

Table 2 The root mean square distance between the original structure and the structure obtained from genetic optimisation using DIEP and M3GNET models as PES calculators

MP ID	Formula	DIEP-PES	M3GNET-PES
mp-24208	CrH_2		
mp-784631	$CrNi_2$		
mp-182	$SrGa_2$	0.01	
mp-24728	VH_2		
mp-2732	PRh_2		
mp-11237	ScAg	0.00	0.04
mp-1018138	VI_2		
mp-1000	BaTe	0.00	0.00
mp-1169	ScCu	0.00	0.01
mp-2516	YZn	0.00	0.00
mp-1441	CsO_2		
mp-1883	SnTe	0.00	
mp-1008626	VTe_2		
mp-2221	Zr_2Ag		
mp-2697	SrO_2		0.01
mp-2857	ScN	0.02	0.01
mp-23251	KBr	0.00	0.00
mp-987	ZnCu	0.00	0.00
mp-2658	AlFe	0.00	0.00
mp-28013	MnI_2	0.29	
mp-1207380	ZrIn	0.00	0.00

operators are allowed: cut and splice pairing,[30] soft mutation[31] and strain mutation,[32] with probabilities of 0.4, 0.3 and 0.3, respectively. To compare between the optimised structures, we calculate the root mean square distance between two structures using the `StructureMatcher.get_rms_dist()` method in the `pymatgen` python library. The results are displayed in Table 2. The DIEP-PES was able to score more hits than M3GNET-PES: the DIEP-PES optimiser re-discovered the equilibrium structure of 12 out of 22 structures, whereas M3GNET-PES re-discovered 10 structures.

3.2.2 Carbon nanotube fracture. A pristine carbon nanotube (CNT) can withstand an elongation strain up to 20% of its original length, but will rupture at lower strains due to the presence of defects.[33] We investigate the maximum strain that pristine and defective CNTs can withstand by performing quasi-static pulling of the CNT, in which a strain increment of 0.5% is enforced on the structure followed by geometry optimisation until the CNT ruptures. This approach is similar to that in ref. 34–36. The CNT structure we simulate is a 172 Å-long (10,10) zigzag CNT (2798 atoms), and we examine two defects: a single C vacancy defect, in which a single C atom close to the middle of the structure (as indicated in Fig. 7a) is removed, and a double C vacancy defect, in which 2C atoms close to the middle of the structure are removed. Following the removal of the C atoms, the atomic structure of the defect is optimised. For the pristine CNT, the structure ruptures when using DIEP-PIES at 22% (Fig. 7b), which is close to the failure strain value of 20% obtained using DFT for a (10,0) CNT.[36] However, the M3GNET-PES simulation does not lead to rupture. For vacancy defects, rupture is only produced when using the DIEP-PES: the CNT with a single C vacancy defect
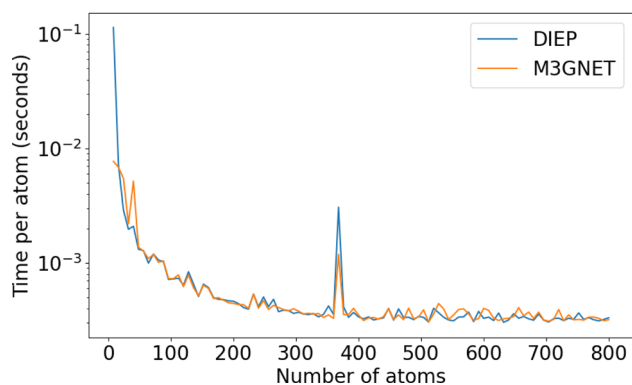

Fig. 6 The scaling behaviour of DIEP-PES and M3GNET-PES: the computational time per atom as the size of the system increases.
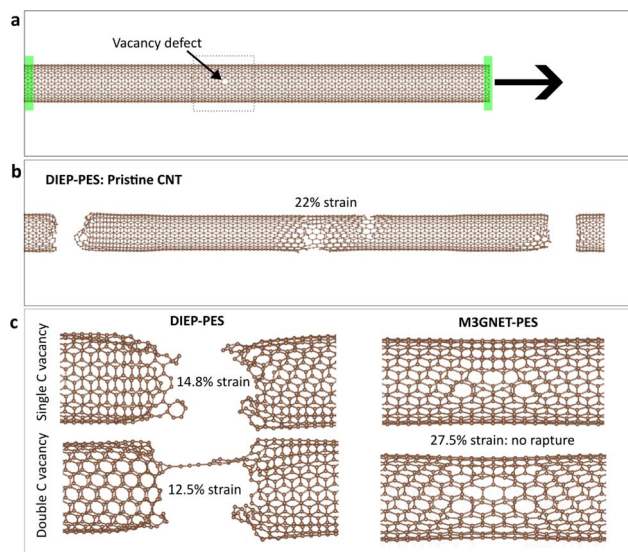
Fig. 7 Simulation of CNT fracture. (a) The simulation cell for the (10,10) zigzag CNT, highlighting the frozen atoms using green rectangles and indicating the carbon vacancy with an arrow. (b) For pristine CNT, it ruptures at a strain of 22% using the DIEP-PES, but never ruptures even when strain reaches 25% using the M3GNET-PES. (c) The defective CNT ruptures at 14.8% strain (single C vacancy) and at 12.5% strain (double C vacancy) when the DIEP-PES is used. However, using the M3GNET-PES, the same defective CNTs never rupture, even when strain reaches 27.5%.

ruptures at a strain of 14.8%, while that with a double C vacancy ruptures at 12.5%, as displayed in Fig. 7c. The drop in failure strain from pristine to double vacancy defective structures qualitatively agree with the results in ref. 33, where a CNT with similar length was examined using classical molecular dynamics. We display the structure of the fractured CNT in Fig. 7. We also examined the influence of the distance between the C vacancies and the rupture strain by applying the strain procedure on two CNT structures where the vacancy–vacancy distances are 7.4 Å and 19.7 Å. We found that the rupture strains are 15.4% and 15.6% respectively, indicating a slight decrease in CNT strength when the distance between the vacancies becomes smaller. The distance between the vacancies in the double vacancy structure in Fig. 7 is the smallest possible, and hence is the weakest CNT structure (12.5% rupture strain). However, for M3GNET-PES, the structure does not rupture at all for an elongation strain of 27.5%. That is, the M3GNET-PES highly overestimates the elasticity of the pristine and defective CNT structure, which is contrary to observation and theoretical calculations.

4 Conclusion

The ability to accurately predict the total energy is critical for the determination of the thermodynamic properties of materials. Our work improved the accuracy of predicting this quantity for materials with and without defects by introducing the direct integration of the external potential (DIEP) method. In

principle, DIEP partially reflects the computations that take place during the electronic structure optimisation step in density functional theory, and therefore more faithfully introduces "physics" in a physics-informed machine learning process. We demonstrated the enhanced accuracy of DIEP in predicting the total energy per atom for several datasets: a 12k dataset of pristine materials, and datasets that represent 6 classes of material imperfections. In addition, we established the ability of DIEP in predicting the potential energy surface for materials (total energies and forces) by performing structure optimisation and molecular dynamics tasks, in particular reproducing the maximum strain of a carbon nanotube structure. Enriched with its unique physical insight, DIEP is therefore suited for high-throughput screening procedures for accelerating material discovery.

5 Methods

5.1 DFT calculations

DFT calculations are performed using VASP 5.4.4.[37] The generalized gradient approximation (GGA) of Perdew, Burke and Ernzerfof (PBE),[38] The energy cut-off for the plane wave basis set is 520 eV, and the energy tolerance is 10^{-6} eV to ensure the accuracy of the calculations. In the structural energy minimization, the internal coordinates are allowed to relax until all of the forces are less than 0.01 eV Å^{-1}. For magnetic structures, the initial magnetic moments are set using the default VASP values.

Code availability

DIEP was implemented on the codebase of Matgl and is available here: **https://github.com/sheriftawfikabbas/diep**. The code that was used to generate the deformed structures is available as part of the `oganesson` package: **https://github.com/sheriftawfikabbas/oganesson**.

Data availability

The VASP POSCAR files for the materials in TinyUnitCells and all of the defects examined in the work, the trained DIEP and M3GNET models are available here: **https://github.com/sheriftawfikabbas/materialsalchemist**.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

References

1 I. I. Baskin, V. A. Palyulin and N. S. Zefirov, *J. Chem. Inf. Comput. Sci.*, 1997, **37**, 715–721.

2 X. Zhang, L. Wang, J. Helwig, Y. Luo, C. Fu, Y. Xie, M. Liu, Y. Lin, Z. Xu, K. Yan, K. Adams, M. Weiler, X. Li, T. Fu, Y. Wang, H. Yu, Y. Xie, X. Fu, A. Strasser, S. Xu, Y. Liu, Y. Du, A. Saxton, H. Ling, H. Lawrence, H. Stärk, S. Gui, C. Edwards, N. Gao, A. Ladera, T. Wu, E. F. Hofgard, A. M. Tehrani, R. Wang, A. Daigavane, J. Kurtin, Q. Huang, T. Phung, M. Xu, C. K. Joshi, S. V. Mathis, K. Azizzadenesheli, A. Fang, A. Aspuru-Guzik, M. Bronstein, M. Zitnik, A. Anandkumar, S. Ermon, P. Liò, R. Yu, S. Günnemann, J. Leskovec, H. Ji, J. Sun, R. Barzilay, T. Jaakkola, C. W. Coley, X. Qian, T. Smidt and S. Ji, Artificial Intelligence for Science in Quantum, Atomistic, and Continuum Systems, *arXiv*, 2023, preprint, arXiv:2307.08423, DOI: 10.48550/arXiv.2307.08423.

3 A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder and K. A. Persson, *APL Mater.*, 2013, **1**, 011002.

4 K. Choudhary, K. F. Garrity, A. C. E. Reid, B. DeCost, A. J. Biacchi, A. R. Hight Walker, Z. Trautt, J. Hattrick-Simpers, A. G. Kusne, A. Centrone, A. Davydov, J. Jiang, R. Pachter, G. Cheon, E. Reed, A. Agrawal, X. Qian, V. Sharma, H. Zhuang, S. V. Kalinin, B. G. Sumpter, G. Pilania, P. Acar, S. Mandal, K. Haule, D. Vanderbilt, K. Rabe and F. Tavazza, *npj Comput. Mater.*, 2020, **6**, 173.

5 S. Curtarolo, W. Setyawan, G. L. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. J. Mehl, H. T. Stokes, D. O. Demchenko and D. Morgan, *Comput. Mater. Sci.*, 2012, **58**, 218–226.

6 S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl and C. Wolverton, *npj Comput. Mater.*, 2015, **1**, 15010.

7 J. Gasteiger, J. Groß and S. Günnemann, Directional Message Passing for Molecular Graphs, *arXiv*, 2022, preprint, arXiv:2003.03123, DOI: 10.48550/arXiv.2003.03123.

8 C. Chen and S. P. Ong, *Nat. Comput. Sci.*, 2022, **2**, 718–728.

9 I. Batatia, D. P. Kovács, G. N. C. Simm, C. Ortner and G. Csányi, MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields, *arXiv*, 2023, preprint, arXiv:2206.07697, DOI: 10.48550/arXiv.2206.07697.

10 P. Hohenberg and W. Kohn, *Phys. Rev.*, 1964, **136**, B864–B871.

11 B. Cuevas-Zuviría and L. F. Pacios, *J. Chem. Inf. Model.*, 2020, **60**, 3831–3842.

12 E. M. Sunshine, M. Shuaibi, Z. W. Ulissi and J. R. Kitchin, *J. Phys. Chem. C*, 2023, **127**, 23459–23466.

13 S. Gong, T. Xie, T. Zhu, S. Wang, E. R. Fadel, Y. Li and J. C. Grossman, *Phys. Rev. B*, 2019, **100**, 184103.

14 P. B. Jørgensen and A. Bhowmik, DeepDFT: Neural Message Passing Network for Accurate Charge Density Prediction, *arXiv*, 2020, preprint, arXiv:2011.03346, DOI: 10.48550/arXiv.2011.03346.

15 T. A. Batchelor, J. K. Pedersen, S. H. Winther, I. E. Castelli, K. W. Jacobsen and J. Rossmeisl, *Joule*, 2019, **3**, 834–845.

16 S. Lynch, in *Stress Corrosion Cracking*, Elsevier, 2011, pp. 90–130.

17 L. Rondin, J.-P. Tetienne, T. Hingant, J.-F. Roch, P. Maletinsky and V. Jacques, *Rep. Prog. Phys.*, 2014, **77**, 056503.

18 G. Thiering and A. Gali, *Phys. Rev. B*, 2016, **94**, 125202.

19 T. Iwasaki, F. Ishibashi, Y. Miyamoto, Y. Doi, S. Kobayashi, T. Miyazaki, K. Tahara, K. D. Jahnke, L. J. Rogers, B. Naydenov, F. Jelezko, S. Yamasaki, S. Nagamachi, T. Inubushi, N. Mizuochi and M. Hatano, *Sci. Rep.*, 2015, **5**, 12882.

20 T. Iwasaki, Y. Miyamoto, T. Taniguchi, P. Siyushev, M. H. Metsch, F. Jelezko and M. Hatano, *Phys. Rev. Lett.*, 2017, **119**, 253601.

21 A. M. Flatae, S. Lagomarsino, F. Sledz, N. Soltani, S. S. Nicley, K. Haenen, R. Rechenberg, M. F. Becker, S. Sciortino, N. Gelli, L. Giuntini, F. Taccetti and M. Agio, *Diamond Relat. Mater.*, 2020, **105**, 107797.

22 J. M. Baker, J. A. Van Wyk, J. P. Goss and P. R. Briddon, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 2008, **78**, 235203.

23 A. Mainwood, *J. Phys. C: Solid State Phys.*, 1979, **12**, 2543–2549.

24 R. Kalish, *Diamond Relat. Mater.*, 2001, **10**, 1749–1755.

25 J. P. Goss, R. Jones, M. I. Heggie, C. P. Ewels, P. R. Briddon and S. Öberg, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 2002, **65**, 115207.

26 C. Weigel, D. Peak, J. W. Corbett, G. D. Watkins and R. P. Messmer, *Phys. Rev. B: Solid State*, 1973, **8**, 2906–2915.

27 J. Simons, P. Joergensen, H. Taylor and J. Ozment, *J. Phys. Chem.*, 1983, **87**, 2745–2753.

28 J. Behler and M. Parrinello, *Phys. Rev. Lett.*, 2007, **98**, 146401.

29 J. Qi, T. W. Ko, B. C. Wood, T. A. Pham and S. P. Ong, *npj Comput. Mater.*, 2024, **10**, 43.

30 L. B. Vilhelmsen and B. Hammer, *Phys. Rev. Lett.*, 2012, **108**, 126101.

31 A. O. Lyakhov, A. R. Oganov and M. Valle, *Comput. Phys. Commun.*, 2010, **181**, 1623–1632.

32 C. W. Glass, A. R. Oganov and N. Hansen, *Comput. Phys. Commun.*, 2006, **175**, 713–720.

33 L. Yang, I. Greenfeld and H. D. Wagner, *Sci. Adv.*, 2016, **2**, e1500969.

34 D. Troya, S. L. Mielke and G. C. Schatz, *Chem. Phys. Lett.*, 2003, **382**, 133–141.

35 Q. Zhao, M. B. Nardelli and J. Bernholc, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 2002, **65**, 144105.

36 S. L. Mielke, D. Troya, S. Zhang, J.-L. Li, S. Xiao, R. Car, R. S. Ruoff, G. C. Schatz and T. Belytschko, *Chem. Phys. Lett.*, 2004, **390**, 413–420.

37 G. Kresse and J. Furthmüller, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1996, **54**, 11169–11186.

38 J. P. Perdew, K. Burke and M. Ernzerhof, *Phys. Rev. Lett.*, 1996, **77**, 3865–3868.