Atomistic Graph Neural Networks for metals: Application to bcc iron

Lorenzo Cian*1, Giuseppe Lancioni*1, Lei Zhang2, Mirco Ianese1, Nicolas Novelli1, Giuseppe Serra1,3, and Francesco Maresca2,3

1 Artificial Intelligence LABoratory (AILAB), University of Udine, Udine, Italy
2 ENgineering and TEchnology institute Groningen (ENTEG), University of Groningen, Groningen, The Netherlands
3 Correspondence to: giuseppe.serra@uniud.it and f.maresca@rug.nl

ABSTRACT

The prediction of the atomistic structure and properties of crystals including defects based on ab-initio accurate simulations is essential for unraveling the nano-scale mechanisms that control the micromechanical and macroscopic behaviour of metals. Density functional theory (DFT) can enable the quantum-accurate prediction of some of these properties, however at high computational costs and thus limited to systems of $\sim 1,000$ atoms. In order to predict with quantum-accuracy the mechanical behaviour of nanoscale structures involving from thousands to several millions of atoms, machine learning interatomic potentials have been recently developed. Here, we explore the performance of a new class of interatomic potentials based on Graph Neural Networks (GNNs), a recent field of research in Deep Learning. Two state-of-the-art GNN models are considered, SchNet and DimeNet, and trained on an extensive DFT database of ferromagnetic bcc iron. We find that the DimeNet GNN Fe potential including three-body terms can reproduce with DFT accuracy the equation of state and the Bain path, as well as defected configurations (vacancy and surfaces). To the best of our knowledge, this is the first demonstration of the capability of GNN of reproducing the energetics of defects in bcc iron. We provide an open-source implementation of DimeNet that can be used to train other metallic systems for further exploration of the GNN capabilities.

Introduction

Recently, Graph Neural Networks (GNNs) have become one of the most active research fields in Artificial Intelligence. GNNs are a class of Deep Learning methods introduced to analyze data which display a graph structure. Graphs represent the topology of a great variety of data structures in which objects (nodes) are connected with each other by some kind of relation (edges). Due to the very general nature of graphs, applications of GNNs are found in very different contexts, such as computer vision, natural language processing, social sciences, and natural sciences including biology, particle physics and astrophysics.

The topology of a graph can also reflect that of atomistic crystal structures: indeed, a graph can be generated by connecting each atom (nodes) with its neighbors (edges), within a specified cutoff radius. The Message Passing Neural Networks framework (MPNN) has been introduced as a common GNN paradigm for atomistic structures in quantum chemistry applications.

Within an atomistic GNN, the atoms and their connections are associated with numerical lists of “features”, also named embedding vectors. Features are updated by the Message Passing framework, which is a two-step process. In the first step, each atom receives a message that is an aggregate of its neighbour’s embeddings. In the second step, an updated embedding of the atom is evaluated, by means of a function that depends on the message and on the current atomic embedding. By iterating this scheme $n$ times, each atom will receive messages from atoms that are distant up to $n$ connections, thus accounting for long-range interactions.

A GNN model for atomistic graphs is therefore determined: (i) by the nature and the size of the embeddings, which convey the informative content of the specific atomistic system and (ii) by the operations it executes on them, i.e. the procedure used to aggregate and update embeddings. Once the aforementioned characteristics are defined, the model can be trained to predict the system potential energy surface (PES).

A number of GNN models have been proposed in the recent years to model atomistic systems. Most of them were first introduced in molecular research and further applied to crystals. Deep Tensor Neural Networks (DTNN) and PhysNet
aggregate the atomic embeddings by means of filters that ensure that the resulting message changes smoothly with respect to small changes of the interatomic distances. The main difference between DTNN and PhysNet lies in how distances are represented and how messages are aggregated. Crystal Graph Convolutional Neural Networks (CGCNN)\textsuperscript{19} were explicitly developed to deal with materials displaying a crystal structure, such as metals. Unlike DTNN and PhysNet, CGCNN considers both atomic (node) and edge embeddings; however, distances are not regularized with continuous functions: the range of the distances is partitioned in ten equally spaced segments, and interatomic distances are encoded within a single vector in which all components are zero but the one associated with the matching segment. Thus, this model lacks the ability to smoothly change the embeddings with respect to small displacements of the atomic positions. SchNet\textsuperscript{20} is based on DTNN and introduces continuous filter convolutions; distances are used as input of a neural layer that generates a continuous mapping to an embedding space. In an updated version\textsuperscript{21}, periodic boundary conditions (PBCs) have been introduced, and the model has been applied to the prediction of formation energies of bulk crystals. Also inspired by SchNet and sharing its overall architecture, the “Neural message passing with edge updates”\textsuperscript{22} uses both node and edge embeddings in the form of a concatenation of the two connected atoms embeddings. This makes edge embeddings directional as they depend on the order of the concatenated elements. MatErials Graph Network (MEGNet)\textsuperscript{23} leverages a similar scheme, by incorporating both directional edge and node updates, while also introducing a global state vector which stores the molecule/crystal-level or state attributes, e.g. the temperature of the system. Updates of atoms, bonds and global state vector are performed in a sequence. All these approaches employ filters that rely only on the distance between pairs of atoms to aggregate and update the atomic embeddings.

It is well-established that classical, empirical interatomic potentials,\textsuperscript{24} that rely on pairwise interactions often fail to reproduce structural changes\textsuperscript{25} and some crucial properties of dislocations in metals\textsuperscript{26}. In the case of phase transitions, the addition of directionality, i.e. angular dependence of the interatomic potential, as well as second nearest-neighbor interactions, has lead to the improved qualitative reproduction of quantum-mechanical PES\textsuperscript{27}.

Within the context of GNNs, there is a remarkable shortage of approaches that rely also on the angle between edges connecting atomic pairs. Embeddings of edges connecting triplets of atoms convey the angular information, and once they are updated via the message passing scheme, they can be used to update the atomic embeddings. With this aim, DimeNet\textsuperscript{27} also leverages the Directional Message (hence the name) by considering the direction of the pairwise connections and by introducing the angle between two edges connected within atomic triplets. DimeNet employs a continuous filter convolution by expanding both distances and angles in a Bessel-Fourier basis. However, to date, DimeNet has been applied merely to isolated molecules and has not been investigated to model crystals such as metals.

Although GNNs have been scarcely explored in the context of interatomic potentials for metals, they introduce a number of advantages with respect to other ML methods\textsuperscript{15}. First, interactions among neighbouring atoms are straightforwardly modeled as pair-wise connections. Previous ML approaches need to introduce specific geometrical descriptors of the environment around atoms (within a cut-off radius), such as atom-centered symmetry functions in the Neural Networks Potentials\textsuperscript{28}, or bispectrum components and then smooth overlap of atomic positions (SOAP)\textsuperscript{29} in Gaussian Approximation Potentials (GAP)\textsuperscript{30}. Second, iterating the process makes the model able to consider the contributions of distant atoms, so as to mimic the influence of long-range interactions beyond the cut-off distance that limits pairwise interactions. This can be easily achieved by stacking message-passing layers in the network. Previous ML approaches either lack these long-range contributions or account for them by adding extra long-range terms to the total energy, e.g. for electrostatic interactions\textsuperscript{31}. Third, the GNN approach guarantees scalability of the system, as the pair-wise nature of the connections means that complex clusters of atoms can be modeled by simply increasing the number of iterations, at a limited computational cost. Finally, since the approach is only dependent on the relative positions of the atoms which determine the connections inside the cut-off radius, it is also invariant with respect to isometric transformations, i.e. reflections, translations, rotations, and combinations of those, and to permutation of atoms.

Here, we use GNNs to explore their ability to reproduce with quantum-accuracy the potential energy surface (PES) of metals, by taking as a reference the challenging and technologically crucial example of ferromagnetic body-centered-cubic (BCC) iron. We consider SchNet as a prototypical GNN framework that is based on the distance of atomic pairs, and we consider DimeNet to assess the performance of a GNN scheme that also includes angular (three-body) interactions. To this purpose, we have implemented periodic boundary conditions (PBCs) and made the new DimeNet implementation that includes PBCs available at https://github.com/AilabUdineGit/GNN_atomistics/. In order to machine-learn the GNN interatomic potential, we use an existing database\textsuperscript{32} that was previously trained to develop a Gaussian Approximation Potential (GAP)\textsuperscript{33}.

The remainder of this paper is organized as follows. Section Results is divided in two main subsections: “Implementation and training” reports a summary of computational details, together with some performance metrics; “Testing the GNN interatomic potentials for bcc Fe” shows a comparative analysis of the networks based on their predictions of the properties of iron. A general summary of the methodology and its achievements, together with suggestions for future improvements, is provided in the “Discussion and Conclusions section”. Finally, section Methods details the approach and the implementation of the networks, and is organized in three subsections: “Graph Neural Networks and Message Passing” contains a formal description of the Message Passing paradigm applied to GNNs for atomistic systems; “Network models” provides details of
both the networks SchNet and DimeNet; “Dataset” reports a summary of the used data.

**Results**

**Implementation and training**

To model bulk crystal structures, the simulated atomic cluster must be embedded in an effectively infinite medium. This is achieved by using periodic boundary conditions (PBCs), which are already implemented in SchNet. Here, PBCs have been implemented also for DimeNet. The training strategy is the same for both SchNet and DimeNet. All data used for the training are from a large, existing, highly-converged DFT database\(^{32}\) of bcc ferromagnetic iron that includes both pristine configurations and configurations with defects such as free surfaces, vacancies and interstitials (see Database section for details). A GAP potential that reproduces accurately DFT vibrational and thermodynamic properties\(^{33}\) is also trained, and employed as a baseline in the comparison of the GNN models.

The training dataset is built as a subset of 80% of the database; samples are randomly shuffled to avoid bias. The remaining 20% of the samples is used to test the trained model; samples are not shuffled in this case. To regularize the distribution of the data and improve training efficiency, the per-atom energies of the whole dataset have been standardized by subtracting the mean value and dividing by the standard deviation. Data samples are then batched with batch size \(N = 6\). A random seed is set to enable reproducibility of the process. The objective function, or loss, to minimize is the mean absolute error (MAE) of the difference between the predicted energy \(\hat{E}_i\) and its target value \(E_i\), averaged over the batch:

\[
L_{MAE} = \frac{1}{N} \sum_{i=1}^{N} |\hat{E}_i - E_i|.
\]

For each batch, the gradient of the loss is evaluated with respect to all the trainable parameters (weights and biases) of the network. Then, the optimization algorithm minimizes the loss by adapting the parameter values. At the end of each epoch (when all the batches are evaluated) the training convergence is assessed by evaluating the MAE over all the test data. In our setting an Adam\(^{34, 35}\) optimizer was adopted. The initial learning rates, \(\alpha = 10^{-4}\) for DimeNet and \(\alpha = 10^{-3}\) for SchNet, have been fixed by performing preliminary tests. A linear scheduler was used to reduce the learning rate if the loss did not decrease significantly; more precisely, for DimeNet (SchNet respectively) the learning factor is reduced by a rate of 1/10 (respectively, 1/2) each time the test loss was detected not to have improved by at least 1% (respectively, 5%) over the last 10 (respectively, 3) epochs. The more strict requirements adopted for SchNet are due to its observed higher computational cost and difficulty for the loss to converge to the minimum. The training is stopped when 100 training epochs have been performed.

Using a Tesla P100 GPU with 16GB RAM, the training time amounts to \(\sim 11\) min/epoch for DimeNet and \(\sim 22\) min/epoch for SchNet, which means a total training time of \(\sim 18\) and \(\sim 37\) hours, respectively. For a rough comparison, we also trained GAP on the same dataset, by using Intel Xeon E7 4860v2 CPU with \(\sim 317\)GB RAM, and the training lasted \(\sim 60\) hours. Final values of the test MAE are in the order of magnitude of tens of meV. Inference latencies have been evaluated for 54 and 128 atoms lattices and are of the order of tens of milliseconds, with the exception of a value of 104 milliseconds for SchNet on the smaller lattice: being a lighter model, SchNet relies less on GPU than DimeNet and uses only \(\sim 1\)% of resources during 54 atoms inference, while DimeNet uses \(\sim 25\)%.

| Metric | Unit | SchNet | DimeNet |
|--------|------|--------|---------|
| Training time | min./epoch | \(\sim 22\) | \(\sim 11\) |
| Test MAE | meV | 54.8 | 23.3 |
| Inference latency (54 atoms cube) | sec. | 0.104 | 0.040 |
| Inference latency (128 atoms cube) | sec. | 0.041 | 0.053 |

**Table 1.** Training time, test MAE and inference latency for SchNet and DimeNet.

In the original papers\(^{21, 27}\) atomic embeddings have size of \(F = 64\) for SchNet and \(F = 128\) for DimeNet. We tested both values on both models, and obtained that while DimeNet improves slightly from 64 to 128 (test MAE from 24.85 to 23.3), SchNet makes a sensible leap forward (test MAE from 76.0 to 54.8). Consequently, an embedding size of 128 was set for both models. We consider this aspect interesting and being worth of future investigation.

The cutoff value is determined as a trade-off between two competing requirements: on one hand, the higher is the value of the cutoff, the higher is the number of connected atoms within an interaction block; on the other hand, the higher is this...
number, the higher is the computational cost during training. For this reason, and considering that DimeNet is a much more complex network in which also triplets of atoms are considered, the cutoff radius is different for the two models: $r_{\text{cut}} = 5.0 \text{ Å}$ for SchNet; 3.5 Å for DimeNet. Using a larger cutoff (up to 4 Å) for DimeNet did not increase the accuracy but did increase the computational time.

The presence of seven interaction blocks in DimeNet with respect to three in SchNet alleviates for the shorter $r_{\text{cut}}$, allowing the network to receive messages from distant atoms and to adequately model long-range interactions.

**Testing the GNN interatomic potentials for bcc Fe**

The SchNet and DimeNet Fe potentials are benchmarked against either published DFT data\(^{33}\) or data computed with Quantum Espresso based on settings (k-mesh and energy convergence) consistent with the training database\(^{33}\). The equation of state is computed with GNNs by varying the lattice constant $a_0 = 2.834$ Å of the primitive unit cell within a range of ±5% volumetric change around the equilibrium volume computed with DFT. As shown in Fig. 1, both GNNs reproduce the DFT data with high accuracy.

![Figure 1. a Equation of state of SchNet and DimeNet compared with DFT data and the EAM\(^{36}\) and MEAM\(^{37}\) empirical potentials. b Equation of state of the DimeNet compared with DFT data and the state-of-the-art GAP iron potential\(^{33}\).](image)

To compare the performance of GNNs with empirical potentials, we compute the equation of state, equilibrium volume and bulk modulus with two broadly used empirical potentials: EAM\(^{36}\), which is based on pairwise interactions; and MEAM\(^{37}\), that includes higher-order interactions (e.g. angular-dependent terms). Both GNN potentials reproduce the DFT results with high accuracy, while both the equilibrium volume and the curvature of the empirical potentials are far from the DFT results (see Fig. 1a). One reason for the discrepancy is that the empirical potentials are fitted to the experimental data of the equilibrium volume $V_0 = 11.7$ Å\(^3\), which is obtained by extrapolation to T=0K\(^{37}\). However, despite being fitted to such value, both EAM and MEAM visibly underpredict the experimental equilibrium volume. In contrast, both GNNs can reproduce closely the dataset they have been trained to and, as shown in Fig. 1b, the level of accuracy is comparable with the state-of-the-art GAP interatomic potential for BCC iron\(^{33}\).

The equilibrium volume and bulk modulus of iron are computed by fitting the Birch-Murnaghan equation of state to the energy-volume curve. The result of the fitting for the GNNs and DFT data is reported in Table 2.

| Property | Unit | DFT | SchNet | $\varepsilon_{\text{SN}}$ | DimeNet | $\varepsilon_{\text{DN}}$ | GAP\(^{33}\) | $\varepsilon_{\text{GAP}}$ |
|----------|------|-----|--------|----------------|---------|----------------|--------|----------------|
| $a_0$    | Å    | 2.834 | 2.834  | 0.0%          | 2.834   | 0.0%           | 2.834  | 0.0%          |
| $B_0$    | GPa  | 199.8 ± 0.1 | 199.0  | -0.4%         | 199.4   | -0.2%          | 198.2  | -0.8%         |

*Table 2. T=0K lattice parameter $a_0$ and bulk modulus $B_0$ for α-iron. GNN results are compared to DFT data. The relative errors of SchNet ($\varepsilon_{\text{SN}}$), DimeNet ($\varepsilon_{\text{DN}}$) and GAP ($\varepsilon_{\text{GAP}}$) with respect to DFT are also shown.*

As indicated by the relative errors $\varepsilon_{\text{SN}}$ and $\varepsilon_{\text{DN}}$, both SchNet and DimeNet reproduce the equilibrium lattice parameter...
and the bulk modulus with an accuracy comparable to GAP. Both the models achieve DFT-accurate results in the equation of state, with a maximum energy difference $< 0.1$ meV in the volume range $[11.0, 12.0] \, \text{Å}^3$. These results thus reveal no apparent difference between the performance of SchNet and DimeNet.

In order to assess the ability of GNNs to reproduce tetragonal lattice distortions, the Bain path is evaluated and compared with DFT data (Figure 2).

![Figure 2. a DFT Bain path compared to the Bain path obtained using the GNN potentials and the empirical potentials. b DFT Bain path compared to the Bain path computed with DimeNet and GAP. In both panels, grey dots represent the cloud of the training data.](image)

In the figure, DFT is used to compute the energy as a function of a distorted primitive cell. The cell is distorted at constant volume, that is by increasing one axis, $c$, while reducing the two other axes, $a$, and keeping the volume constant. Both the Fe SchNet and DimeNet potentials are then used to compute the same path. Volume optimization, i.e. finding the minimum energy configuration at the prescribed $c/a$ by adjusting the volume, has also been performed with the GNNs potential to verify that the path does not deviate strongly from the assumed tetragonal distortion at constant volume, and no strong qualitative changes were found with respect to the result obtained with the constrained Bain path. The plot also shows the $c/a$ distortion of the training database. Fig. 2a shows that SchNet interpolates well within the training set while it extrapolates poorly, with a discontinuous behaviour of the energy vs the $c/a$ ratio. Instead, DimeNet can extrapolate fairly well outside of the training database, also reproducing qualitatively the energy barrier at $c/a \sim 1.4$ as well as the subsequent local energy minimum around $c/a \sim 1.65$. This specific capability of DimeNet sets it aside from SchNet, making it a more promising GNN for atomistic simulations of metals with structural transformations. Moreover, DimeNet outperforms the EAM potential, which shows no metastable minimum for BCC ferromagnetic Fe at $c/a \sim 1.65$. MEAM was fitted on data including both BCC and FCC configurations, and for this reason it deviates strongly from the DFT results, which are based on BCC ferromagnetic configurations only. Fig. 2b shows that DimeNet approaches the transferability of GAP for the Bain path.

Finally, the vacancy formation energy and the surface energies have been predicted for a number of crystal planes. The vacancy formation energy is calculated by using a $3 \times 3 \times 3$ cubic supercell. First, one atom of the supercell is removed and a DFT calculation is performed to relax the atoms around the vacancy. Then, the DFT total energy $E_{\text{def}}$ of the vacancy-containing configuration is computed. The total energy $E_{\text{bulk}}$ of the bulk defect-free supercell is also computed. The vacancy formation energy equals

$$E_v = E_{\text{def}} - \frac{N}{N} E_{\text{bulk}},$$

where $N$ is the number of atoms in the bulk system ($N = 54$ atoms in this case).

The surface energy is evaluated for four crystallographic planes, i.e. $\{100\}$, $\{110\}$, $\{111\}$ and $\{112\}$. The surface is generated by creating a supercell with a vacuum region, the energy of which is indicated as $E_{\text{split}}$. The vacancy formation energy is computed as

$$E_{\text{surf}} = (E_{\text{split}} - E_{\text{bulk}})/2A$$
where $A$ is the newly created surface area. The results are plotted in Fig. 3.

**Figure 3.** Vacancy formation energies and surface energies. The $\{100\}, \{110\}, \{111\}$ and $\{112\}$ surfaces are considered.

Fig. 3a shows that both GNNs can reproduce mostly within 10% accuracy all the considered DFT energies. The GNN potentials largely outperform both EAM and MEAM empirical potentials, that consistently underpredict the energies. DimeNet shows the largest deviation ($\sim 7\%$) for the $\{112\}$ surface energy, and is otherwise approaching the predictive capabilities of the GAP potential (Fig. 3b).

**Discussion and Conclusions**

In this paper we presented a comparative study on the application of two GNN models, SchNet and DimeNet, to the prediction of properties of bcc iron. Since DimeNet was previously tested only on molecules and not on periodic structures/crystals, we implemented a version with PBCs and made it publicly available. Both models predict with DFT accuracy the energy-volume curve and related properties such as the bulk modulus and the equilibrium lattice parameter. This result is consistent with the fact that the energy-volume curve includes datapoints close to those of the training database. The investigated GNN potentials outperform closed-form empirical interatomic potentials (e.g. EAM and MEAM) and approach the accuracy of state-of-the-art interatomic potentials such as GAP. This makes the present GNNs implementation interesting for application to other metallic systems.

A different performance of DimeNet with respect to SchNet is found for configurations including tetragonal distortions (Bain path), point defects and planar defects. DimeNet can predict the energy of these configurations within the MAE, while the predictive capability of SchNet is limited. We attribute this difference to the fact that, in DimeNet, the energy depends explicitly on the angular, three-body contributions that are essential for structural transformations and for local shape distortions, while SchNet only depends on pairwise contributions. It is also remarkable that DimeNet has better transferability, e.g. considering the Bain path.

By showing the capabilities of GNNs and especially the importance of three-body terms, this work supports the further investigation of GNNs and specifically DimeNet. Activity is currently ongoing in the following directions:

- There is a number of potential improvements in terms of efficiency and accuracy of the model, which is related to the hyperparameter optimization. Further investigations will involve finding a tradeoff between choosing larger cutoff radii and/or increasing the number of interaction layers, in order to ensure the efficient description of short- and long-range interactions with high accuracy.

- Another aspect to be investigated is the number of features of both atom and edge embeddings, and their initialization.
These are crucial characteristics in modeling the atomic environment, encoding properties such as the nature of the atom and of the pair interactions, and are expected to impact the model efficiency, e.g. in the convergence of the training.

- The implementation of the developed GNN Fe DimeNet potential within the LAMMPS\textsuperscript{38} open-source package is currently ongoing and will enable the systematic simulation of thermoelastic properties, as well as linear and planar defects such as dislocations and cracks that are relevant for the investigation of the mechanical properties of metals.

- We expect that, in the spirit of Atomic Cluster Expansions (ACE)\textsuperscript{39}, the transferability of GNN potentials will be improved by including more terms in the angular descriptions, by using a different choice of the radial function (e.g. based on Chebyshev polynomials), or by setting different values of the parameters $l,m$ in the angular functions (which, in the current DimeNet implementation, are spherical harmonics with $m = 0$), and/or by introducing higher-body terms, beyond the three-body term currently used in DimeNet (see Methods). This is also the subject of current research.

Methods

Graph Neural Networks and Message Passing

A graph\textsuperscript{40} is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $i \in \mathcal{V}$ are the $N$ nodes and $(i,j) \in \mathcal{E}$ are the edges. The connections among the nodes of a graph can be stored in an adjacency matrix $A \in \mathbb{R}^{N \times N}$ containing the pairs $(i,j) \in \mathcal{E}$. At both nodes and edges, vectors of features (or embeddings) are defined as $x_i \in \mathbb{R}^F$ and $e_{ij} \in \mathbb{R}^D$, respectively, where $F,D$ are model specific parameters. In the message passing with node update, node embeddings are updated iteratively, with each iteration executed in the message passing layers $l$ as follows:

$$
\begin{align*}
    x_i^{(l+1)} &= \gamma(x_i^{(l)}, \sum_{j \in N_i} \mu(e_{ij}^{(l)}, x_j^{(l)})) \\
\end{align*}
$$

(4)

where $N_i$ is the set of the nodes connected to node $i$, $\mu$ is a differentiable function of the nodal and edge embeddings, the sum aggregates the contributions of atoms $j$, and $\gamma$ is a differentiable function which evaluates the update of node embedding.

In the message passing with edge update\textsuperscript{41}, edge embeddings are updated by following a similar scheme:

$$
\begin{align*}
    e_{ij}^{(l+1)} &= \kappa(e_{ij}^{(l)}, \sum_{k \in N_j \setminus \{i\}} \nu(x_j^{(l)}, e_{jk}^{(l)}, x_k^{(l)})) \\
\end{align*}
$$

(5)

with the same conventions as the previous case, $\kappa$ and $\nu$ being differentiable functions of the nodal and edge embeddings, analogously to $\gamma$ and $\mu$. Note that edges connected to $(i,j)$ are the edges $(j,k)$ linking node $j$ and node $k \neq i$, hence the index of the summation.

At the next iteration, the message is evaluated by the layer $l + 1$ by aggregating embeddings $x_i^{(l+1)}$ ($e_{ij}^{(l+1)}$) from the neighbours, which in turn have received a message from their own neighbours: stacking together $L$ layers means that the final update is performed by using messages coming from a distance up to $L$ neighbors away, see fig. 4.

Once iteratively updated via the message passing, embeddings are elaborated by a readout function

$$
\begin{align*}
    y &= f(\{x_i^{(L)} : e_{ij}^{(L)}\}) \\
\end{align*}
$$

(6)

which performs a further aggregation of all the embeddings and outputs the prediction $y \in \mathbb{R}$ of the network.
**Figure 4.** Message Passing with node update. Atomic environment as seen by node \(i\): layer \(l = 1\) aggregates messages from connected neighbours (red area); layer \(l = 2\) acts the same but connected neighbours have been already updated by messages from their neighbours at previous layer, so signals received by atom \(i\) are now from a distance of up to two edges (orange area). A sequence of \(L\) layers means messages coming from nodes at a distance of up to \(L\) edges. A similar scheme works for message passing with edge update.

Within the context of crystalline materials, it is straightforward to consider the nodes of a graph as the atoms and to connect by edges the pairs of atoms that lie within a specified interaction radius. Let \(\mathbf{r}_i \in \mathbb{R}^3\) be the coordinates of the atom \(i\). Then, the graph is defined by connecting all the atoms \(j\) that are inside the cutoff radius \(r_{\text{cut}} > ||\mathbf{r}_i - \mathbf{r}_j||\). Atomic embeddings \(\mathbf{x}_i\) are vectors of learnable numerical features. They are randomly initialized, and atoms with the same set of relevant atomic properties, such as atomic number \(Z\), have the same initial embeddings. Edge embeddings \(\mathbf{e}_{ij}\) are similar, with properties related to pairs of connected atoms, such as the interatomic distance \(d_{ij}\). The message and update functions \(\mu, \gamma\) (4, 5) are neural layers which add learnable weights and define the form of the convolutional filter and of the update procedure. Hence, the prediction of the potential energy \(E \in \mathbb{R}\) of a crystal as a function of the atomic coordinates is a regression task performed on such a graph (6).

**Network models**

In this paper we use two recent models of Graph Neural Networks based on the Message Passing framework: SchNet\(^{21}\) and DimeNet\(^{27}\). There are two main differences between them. The first is related to the embeddings: SchNet relies on atomic embeddings, while DimeNet also uses edge embeddings in the form of pairs of atom embeddings to account for the directionality of the message passing. The second difference is the learned convolutional filters used to aggregate embeddings: while SchNet employs a filter that accounts only for the distance between pairs of atoms, DimeNet considers also the angles formed by pairs of edges, or triplets of atoms. The general scheme of both the models is shown in fig. 5. At an high level of abstraction, they can be described in terms of block diagrams, with each block representing a set of specific neural layers that performs some operations on input data and generates output data. Blocks with the same name in both models perform similar general operations.
Figure 5. Block diagram for SchNet (left) and DimeNet (right). Outputs generated by each block are shown near the arrows. In both models the starting point is the Embedding block that maps atom or edge features in a vector space, generating embeddings. For SchNet, the Interaction blocks are in a sequence, the output of one being the input of the next; the final Output block evaluates the total energy \( \hat{E} \). In DimeNet the output of each block is both passed sequentially to the next and further elaborated by the Output block, and then summed to obtain energy \( E \).

**Filters and physical representation of the atomic environment**

To take into account the physical environment surrounding each atom, the convolutional filter assigns weights to the embeddings received by the neighbours (see below the description of Embedding and Interaction blocks). Filter weights are learned during training and have to change smoothly with respect to small atomic shifts. Therefore, distances and angles are expanded, that is represented as feature vectors whose components are sets of continuous basis functions. In SchNet the filter depends only on the interatomic distance \( d \), expanded by a set of radial, Gaussian (G) basis functions:

\[
\phi^G_k(d) = \exp\left(-\frac{(d - \mu_k)^2}{2\sigma^2}\right)
\]

with \( \mu_k \) equally spaced in the interval \([0, r_{\text{cut}}]\), and \( \sigma \) representing the scale of the distances. Hyperparameters \( k \) and \( \sigma \) define the granularity of the representation, and determine the precision of the filter. The spacing \( r_{\text{cut}}/k \) and the scale \( \sigma \) are both set to 0.1 Å in the original paper\(^{21}\), in order to improve the precision to better compare with DimeNet we set them to 0.04 Å.

DimeNet introduces two different filters: one radial depending only on distances, used to weight the embeddings received by atoms; and one radial-angular which takes into account also the angles to weight the embeddings passed to the edges. Both distances and angles are expanded in a 2D Bessel-Fourier basis which are the solutions of the related time-independent Schrödinger equation, and represent the electron density of the system inside the cutoff radius. For the first, only radial, filter, distances \( d \) are expanded in a feature vector whose components are given by the Radial Basis Functions (RBF):

\[
\phi_k^{\text{RBF}}(d) = \sqrt{\frac{2}{r_{\text{cut}}}} \sin\left(\frac{k\pi}{r_{\text{cut}}} \frac{d}{r_{\text{cut}}}\right).
\]
The second filter depends on the distance $d$ and the angle $\theta$. The components of the bidimensional feature vectors are given in terms of the Spherical Basis Functions (SBF):

$$
\phi_{SBF}^{l,k}(d, \theta) = \sqrt{2} \frac{r_{\text{cut}}}{r_{\text{cut}}^2 j_{i+1}(z_{lk})} j_i \left( \frac{z_{lk}}{r_{\text{cut}} d} \right) Y_0^l(\theta)
$$

where $j_i$ are the spherical Bessel functions of the first kind and $Y_0^l$ are the spherical harmonics; $z_{lk}$ is the $k$-th root of the $l$-order Bessel function. Settings for the non learnable parameters are as per the original paper \cite{27}, namely: for eq. 8 $k \in [1, \ldots, 6]$ while in eq. 9 $k \in [1, \ldots, 6]$, $l \in [0, \ldots, 5]$. To avoid the discontinuity given by the boundary condition $\phi(d) = 0$ for $d > r_{\text{cut}}$, functions 8 and 9 are multiplied by a smoothing polynomial $u(d) \sim O(d^8)$: a step function with a root of multiplicity 3 at $d = r_{\text{cut}}$.

For both SchNet and DimeNet the expanded representations are passed through dense neural layers (see below) which add the learnable weights. The filter is therefore a mapping of the physical representation of angles and distances to a vector space with dimensions matching the ones of the embeddings to weight. The general aspect of the filters and an intuition of how they work are shown in fig. 6.

**Figure 6.** Filters and physical representation of the atomic environment. (a) Starting from the positions of atoms, distances between pairs (top, SchNet and DimeNet) and distances and angles between triplets (bottom, DimeNet) are evaluated. (b) Distances $d$ are expanded in a Gaussian basis of functions (top, SchNet, eq. 7) or in Radial Bessel basis (middle, DimeNet, eq. 8), while distances $d$ and angles $\theta$ for triplets are expanded in a 2D Bessel-Fourier basis (bottom, DimeNet, 9). (c) The convolutional filters: expansions are passed through dense layers whose weights are optimized during training. Learned weights are the convolutional filters. The first three and the last component are extracted and shown for all the cases.

**Dense layers**

Dense layer is the very basic element of a neural network. Given an input $x \in \mathbb{R}^k$ it is defined as

$$
y = \sigma(W \cdot x + b)
$$

where $W \in \mathbb{R}^{m \times k}$, $b \in \mathbb{R}^m$ are the learnable weights and bias, $\cdot$ is the matrix multiplication operator and $\sigma$ is the activation, i.e. a differentiable nonlinear function. Activation is the shifted softplus for SchNet: $\text{ssp}(x) = \ln(0.5 \cdot e^x + 0.5)$, and the self-gated Swish for DimeNet: $\text{sgs}(x) = x \cdot \text{sigmoid}(x)$. In terms of vector algebra a dense layer projects the input vector $x \in \mathbb{R}^k$ into a vector space $\mathbb{R}^m$ with $m \neq k$ in general, and then applies the function $\sigma$ element-wise.
Embedding block
For SchNet, atom embeddings are defined as vectors \( x_i \in \mathbb{R}^F \); initial values \( x_i^{(0)} \) are randomly initialized. For DimeNet, similarly defined atom embeddings are concatenated in pairs to generate an initial edge embedding \( e_{ij}^{(0)} = (x_i^{(0)} | x_j^{(0)}) \phi_RBF(j_{ij}) \). Note that this definition guarantees the directionality, as in general \( e_{ij} \neq e_{ji} \). Once initialized, embeddings are passed through dense layers.

Interaction block
Message passing paradigm is implemented in Interaction blocks. Multiple Interaction blocks are generally stacked together. Each of them performs a convolution by aggregating embeddings from the directly connected entities, and then updating them. The output of one block is passed as the input to the next.

Let \( l \) be the generic Interaction block. In SchNet, the embedding \( x_i^{(l)} \) received by atom \( i \) from neighbour \( j \in N_i \) is first weighted by the gaussian radial filter depending on \( \phi^G(d) \) (eq. 7). Then embeddings are aggregated and the resulting embedding is summed to \( x_i^{(l)} \) and passed through a dense layer to update it to \( x_i^{(l+1)} \) (eq. 4). In DimeNet, the edge \( (j,i) \) receives message embeddings \( e_{ij}^{(l)} \) from edges \( (k,j) \) that are first weighted by means of the radial filter based on \( \phi^{RBF}(d) \) (eq. 8) with \( d = d_{ji} \), and then by the radial-angular filter based on the Bessel-Fourier basis \( \phi^{SBF}(d, \theta) \) (eq. 9), where \( \theta \) is the angle formed by \( (j,i) \) and \( (k,j) \) and \( d = d_{kj} \). Again, exchanged messages are aggregated, summed over \( k \) to the embedding \( e_{ij}^{(l+1)} \) relative to edge \( (j,i) \) and then passed through the dense layers to obtain the updated \( e_{ij}^{(l+1)} \) edge embedding (eq. 5). For an intuition of how the filters are applied see fig. 7. In DimeNet, updated messages are given as input to the next interaction block and to the related output block, see below.

![Figure 7. Schematic of the application of the filters for DimeNet. Features of the edge embedding \( e_{ij} \) are first multiplied element-wise with the value at the point \( d_{ji} \) of the components of the radial filter. Then another element-wise multiplication is performed with the value at the point \( (d_{kj}, \theta) \) of the components of the radial-angular filter. An analogous scheme works for SchNet, limited to the purely radial filter.](image)

Output block
Output blocks are responsible for lowering the dimensions of the atom embeddings, reducing them to scalars.

SchNet has one only Output block at the end of the Interaction blocks stack; it is a sequence of dense layers whose task is to reduce the embedding to a scalar \( x_i^{(L+1)} \rightarrow x_i^{(L+1)} \), to be interpreted as the atom-wise contribution to the total potential energy. The final prediction is evaluated as the sum of atom-wise contributions \( \hat{E} = \sum x_i^{(L+1)} \).

DimeNet performs another convolution here, resulting in the update of the atomic embeddings. Embeddings \( e_{ij}^{(l+1)} \) from the related interaction block \( l \) (and of the embedding block, \( l = 0 \)) are further convoluted by means of a radial filter based on \( \phi^{RBF}(d) \): \( e_{ij}^{(l+1)} \rightarrow e_{ij}^{(l+1)} \). The update of the embedding of atom \( i \) is then evaluated as \( x_i^{(L+1)} = \sum e_{ij}^{(l+1)} \). Multiple dense layers are applied to reduce dimensions to 1: \( x_i^{(L+1)} \rightarrow x_i^{(L+1)} \) to be intended as the per-atom contribution of the level \( l \) blocks to the output of the model. Finally they are summed atom-wise and level-wise to evaluate the final prediction of the network \( \hat{E} = \sum \sum x_i^{(l+1)} \).

Dataset
We use a DFT database\(^3\) of bcc ferromagnetic iron in our study. The database is generated by delicate collinear spin-polarized plane wave DFT computations, which includes the following subsets.
• DB1: Primitive unit cell under arbitrary pressures at 300K
• DB2: $3 \times 3 \times 3$ and $4 \times 4 \times 4$ supercell under a range of pressures and temperatures
• DB3: $3 \times 3 \times 3$ supercell containing a vacanvty under a range of pressures and temperatures the same as DB2
• DB4: $4 \times 4 \times 4$ supercell with divacancies at 800K
• DB5: $4 \times 4 \times 4$ supercell with 3, 4 and 5 vacancies at 800-1000K
• DB6: $4 \times 4 \times 4$ supercell containing self- and di-interstitials at 100-300K
• DB7: $1 \times 1 \times 6$ supercell with (100), (110), (111) and (112) free surfaces
• DB8: $1 \times 1 \times 6$ supercell with $\gamma$ surfaces on (110) and (112) plane

All structures in DBs other than DB1 are bcc lattices; structures in DB1 are primitive unit cells of bcc lattices. More details about the database can be found in the original paper. The DFT database is computed by using the open source codes QUANTUMESPRESSO. An ultrasoft GGA PBE pseudopotential from 0.21 pslibrary is employed. The kinetic energy cutoff for wavefunctions and charge density are set to be 90 and 1080 $\text{Ry}$, respectively. The $k$ spacing is set to be less than 0.03 Å$^{-1}$.

Data and code availability

The data used for training and testing the system is publicly available at the Materials Cloud site: https://archive.materialscloud.org/record/2017.0006/v2. The code generated to obtain the data reported in the paper can be found at the GitHub repository of the project: https://github.com/AilabUdineGit/GNN_atomistics/

References

1. Zhou, J. et al. Graph neural networks: A review of methods and applications. AI Open 1, 57–81, DOI: https://doi.org/10.1016/j.aiopen.2021.01.001 (2020).
2. Wang, X., Ye, Y. & Gupta, A. Zero-shot recognition via semantic embeddings and knowledge graphs. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 6857–6866, DOI: 10.1109/CVPR.2018.00717 (IEEE Computer Society, Los Alamitos, CA, USA, 2018).
3. Quek, A., Wang, Z., Zhang, J. & Feng, D. Structural image classification with graph neural networks. In 2011 International Conference on Digital Image Computing: Techniques and Applications, 416–421, DOI: 10.1109/DICTA.2011.77 (2011).
4. Hu, H., Gu, J., Zhang, Z., Dai, J. & Wei, Y. Relation networks for object detection. CoRR abs/1711.11575 (2017). 1711.11575.
5. Xu, J., Cao, Y., Zhang, Z. & Hu, H. Spatial-temporal relation networks for multi-object tracking. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 3987–3997, DOI: 10.1109/ICCV.2019.00409 (IEEE Computer Society, Los Alamitos, CA, USA, 2019).
6. Yao, L., Mao, C. & Luo, Y. Graph convolutional networks for text classification. Proc. AAAI Conf. on Artif. Intell. 33, 7370–7377, DOI: 10.1609/aaai.v33i01.33017370 (2019).
7. Huang, L., Ma, D., Li, S., Zhang, X. & Wang, H. Text level graph neural network for text classification. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 3444–3450, DOI: 10.18653/v1/D19-1345 (Association for Computational Linguistics, Hong Kong, China, 2019).
8. Bastings, J., Titov, I., Aziz, W., Marcheggiani, D. & Sima’an, K. Graph convolutional encoders for syntax-aware neural machine translation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 1957–1967, DOI: 10.18653/v1/D17-1209 (Association for Computational Linguistics, Copenhagen, Denmark, 2017).
9. Marcheggiani, D., Bastings, J. & Titov, I. Exploiting semantics in neural machine translation with graph convolutional networks. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 486–492, DOI: 10.18653/v1/N18-2078 (Association for Computational Linguistics, New Orleans, Louisiana, 2018).
10. Fan, W. et al. A graph neural network framework for social recommendations. *IEEE Transactions on Knowl. Data Eng.* 1–1, DOI: 10.1109/TKDE.2020.3008732 (2020).

11. Tan, Q., Liu, N. & Hu, X. Deep representation learning for social network analysis. *Front. Big Data* 2, 2, DOI: 10.3389/fdata.2019.00002 (2019).

12. Fotu, A., Byrd, J., Shariat, B. & Ben-Hur, A. Protein interface prediction using graph convolutional networks. In Guyon, I. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 30 (Curran Associates, Inc., 2017).

13. Shlomi, J., Battaglia, P. & Vlimant, J.-R. Graph neural networks in particle physics. *Mach. Learn. Sci. Technol.* 2, 021001, DOI: 10.1088/2632-2153/abbf9a (2021).

14. Choma, N. *et al.* Graph neural networks for icecube signal classification. 386–391, DOI: 10.1109/ICMLA.2018.00064 (2018).

15. Mueller, T., Hernandez, A. & Wang, C. Machine learning for interatomic potential models. *The J. Chem. Phys.* 152, 050902, DOI: 10.1063/1.5126336 (2020). https://doi.org/10.1063/1.5126336.

16. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. In Precup, D. & Teh, Y. W. (eds.) *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, 1263–1272 (PMLR, International Convention Centre, Sydney, Australia, 2017).

17. Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R. & Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commu.* 8, DOI: 10.1038/ncomms13890 (2017).

18. Unke, O. T. & Meuwly, M. Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges. *J. Chem. Theory Comput.* 15, 3678–3693, DOI: 10.1021/acs.jctc.9b00181 (2019).

19. Xie, T. & Grossman, J. C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.* 120, DOI: 10.1103/physrevlett.120.145301 (2018).

20. Schütt, K. T. *et al.* Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, 992–1002 (Curran Associates Inc., Red Hook, NY, USA, 2017).

21. Jørgensen, P. B., Jacobsen, K. W. & Schmidt, M. N. Neural message passing with edge updates for predicting properties of molecules and materials. *The J. Chem. Phys.* 148, 241722, DOI: 10.1063/1.5019779 (2018). https://doi.org/10.1063/1.5019779.

22. Chen, C., Ye, W., Zuo, Y., Zheng, C. & Ong, S. P. Graph networks as a universal machine learning framework for molecules and crystals. *Chem. Mater.* 31, 3564–3572, DOI: 10.1021/acs.chemmater.9b01294 (2019).

23. Daw, M. S. & Baskes, M. I. Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. *Phys. Rev. B* 29, 6443–6453, DOI: 10.1103/PhysRevB.29.6443 (1984).

24. Lee, T., Baskes, M., Valone, S. & Doll, J. Atomic modeling of thermodynamic equilibrium and polymorphism of iron. *J. physics. Condens. matter : an Inst. Phys. journal* 24 22, 225404 (2012).

25. Marques, F., Dragoni, D., Csányi, G., Marzari, N. & Curtin, W. A. Screw dislocation structure and mobility in body centered cubic Fe predicted by a Gaussian Approximation Potential. *npj Comput. Math.* 4, 69, DOI: 10.1038/s41524-018-0125-4 (2018).

26. Klicpera, J., Groß, J. & Günnemann, S. Directional message passing for molecular graphs. In *International Conference on Learning Representations* (2020).

27. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* 98, 146401, DOI: 10.1103/PhysRevLett.98.146401 (2007).

28. Bartók, A. P., Kondor, R. & Csányi, G. On representing chemical environments. *Phys. Rev. B* 87, 184115, DOI: 10.1103/PhysRevB.87.184115 (2013).

29. Bartók, A. P., Payne, M. C., Kondor, R. & Csányi, G. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* 104, 136403, DOI: 10.1103/PhysRevLett.104.136403 (2010).

30. Behler, J. Constructing high-dimensional neural network potentials: A tutorial review. *Int. J. Quantum Chem.* 115, 1032–1050, DOI: https://doi.org/10.1002qua.24890 (2015). https://onlinelibrary.wiley.com/doi/pdf/10.1002qua.24890.
32. Dragoni, D., Daff, T., Csányi, G. & Marzari, N. Gaussian Approximation Potentials for iron from extended first-principles database (Data Download). Mater. Cloud Arch. 2017.0006/v2, DOI: 10.24435/materialscloud:2017.0006/v2 (2017).

33. Dragoni, D., Daff, T. D., Csányi, G. & Marzari, N. Achieving dft accuracy with a machine-learning interatomic potential: Thermomechanics and defects in bcc ferromagnetic iron. Phys. Rev. Mater. 2, 013808, DOI: 10.1103/PhysRevMaterials.2.013808 (2018).

34. Kingma, D. & Ba, J. Adam: A method for stochastic optimization. Int. Conf. on Learn. Represent. (2014).

35. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. In International Conference on Learning Representations (2019).

36. Ramasubramaniam, A., Itakura, M. & Carter, E. A. Interatomic potentials for hydrogen in α–iron based on density functional theory. Phys. Rev. B 79, 174101, DOI: 10.1103/PhysRevB.79.174101 (2009).

37. Liyanage, L. S. I. et al. Structural, elastic, and thermal properties of cementite (fexC) calculated using a modified embedded atom method. Phys. Rev. B 89, 094102, DOI: 10.1103/PhysRevB.89.094102 (2014).

38. Plimpton, S. Fast parallel algorithms for short-range molecular dynamics. J. computational physics 117, 1–19 (1995).

39. Drautz, R. Atomic cluster expansion for accurate and transferable interatomic potentials. Phys. Rev. B 99, 014104, DOI: 10.1103/PhysRevB.99.014104 (2019).

40. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations (ICLR) (2017).

41. Jørgensen, P., Jacobsen, K. & Schmidt, M. Neural message passing with edge updates for predicting properties of molecules and materials (2018). 32nd Conference on Neural Information Processing Systems, NIPS 2018 ; Conference date: 02-12-2018 Through 08-12-2018.

42. Giannozzi, P. et al. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. J. Physics: Condens. Matter 21, 395502, DOI: 10.1088/0953-8984/21/39/395502 (2009).

43. Giannozzi, P. et al. Advanced capabilities for materials modelling with quantum ESPRESSO. J. Physics: Condens. Matter 29, 465901, DOI: 10.1088/1361-648x/aa8f79 (2017).