Multi-task graph neural networks for simultaneous prediction of global and atomic properties in ferromagnetic systems*

IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 025007 https://doi.org/10.1088/2632-2153/ac6a51

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
7 February 2022

REVISED
9 April 2022

ACCEPTED FOR PUBLICATION
25 April 2022

PUBLISHED
5 May 2022

PAPER

Multi-task graph neural networks for simultaneous prediction of global and atomic properties in ferromagnetic systems*

Massimiliano Lupo Pasini[1,**], **Pei Zhang**[1], **Samuel Temple Reeve**[1] and **Jong Youl Choi**[2]

[1] Oak Ridge National Laboratory, Computational Sciences and Engineering Division, Oak Ridge, TN 37831, United States of America
[2] Oak Ridge National Laboratory, Computer Science and Mathematics Division, Oak Ridge, TN 37831, United States of America
** Author to whom any correspondence should be addressed.

E-mail: lupopasinim@ornl.gov

Keywords: multi-task learning, graph neural network, condensed matter, ferromagnetic system, solid solution alloy

Abstract

We introduce a multi-tasking graph convolutional neural network, HydraGNN, to simultaneously predict *both global and atomic* physical properties and demonstrate with ferromagnetic materials. We train HydraGNN on an open-source *ab initio* density functional theory (DFT) dataset for iron-platinum with a fixed body centered tetragonal lattice structure and fixed volume to simultaneously predict the mixing enthalpy (a global feature of the system), the atomic charge transfer, and the atomic magnetic moment across configurations that span the entire compositional range. By taking advantage of underlying physical correlations between material properties, multi-task learning (MTL) with HydraGNN provides effective training even with modest amounts of data. Moreover, this is achieved with just one architecture instead of three, as required by single-task learning (STL). The first convolutional layers of the HydraGNN architecture are shared by all learning tasks and extract features common to all material properties. The following layers discriminate the features of the different properties, the results of which are fed to the separate heads of the final layer to produce predictions. Numerical results show that HydraGNN effectively captures the relation between the configurational entropy and the material properties over the entire compositional range. Overall, the accuracy of simultaneous MTL predictions is comparable to the accuracy of the STL predictions. In addition, the computational cost of training HydraGNN for MTL is much lower than the original DFT calculations and also lower than training separate STL models for each property.

1. Introduction

Material discovery and design of new materials relies heavily on predicting material properties directly from their atomic structure. There are many physics-based computational approaches to model and predict the behavior of materials at the atomic scale from first principles, such as density functional theory (DFT) [1, 2], quantum Monte Carlo (QMC) [3, 4] and *ab initio* molecular dynamics (MD) [5, 6]. While these methods have been instrumental in predictive materials science, they are extremely computationally expensive. The advent of data-driven modeling techniques has provided new methodologies to produce inexpensive and accurate predictions of material properties which helps enable rapid screening of large material search spaces to select potential material candidates with desirable properties [7–10]. Among all data driven models, deep learning (DL) models have the highest potential to accurately represent complex relations between input features and target quantities, but require large volumes of data to attain high accuracy. Data collected in

* This manuscript has been authored in part by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

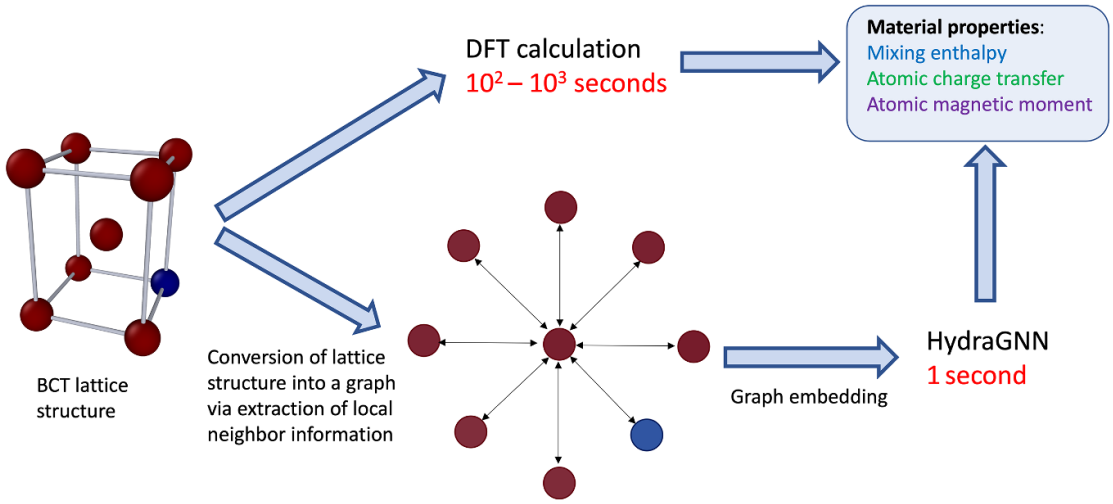

Figure 1. Computational workflow that compares the standard procedure to predict material properties with DFT calculations and the DL-driven methodology that uses the lattice structure as input for HydraGNN to estimate the material properties. Once HydraGNN is trained, it is much faster than DFT.

material science is most often small in volume due to expensive experiments and time-consuming numerical simulations, which challenges, but does not preclude, the use of DL models in the field [11]. Recent efforts have proposed transfer learning [12] and injecting the model with pre-existing physical knowledge [13–15] to overcome this data constraint.

Multi-task learning (MTL) consists of using one DL model to perform several training tasks at the same time [16]. All training tasks mutually influence each other, acting as inductive biases and thus improving each other's predictive performance. When multiple target quantities are correlated with each other, MTL can be used to identify and learn features that are common to all the quantities of interest and transfer knowledge through these common features from one quantity to another. This not only helps to counteract the challenge of data scarcity, but also significantly reduces the computational effort with respect to single-task learning (STL) because only one MTL model is used to predict all properties simultaneously, rather than several distinct STL neural networks. MTL is a specific type of physics informed DL when applied to material science data, because it leverages correlations between multiple material properties dictated by the physics [17–19]. The total number of properties being predicted, the degree to which all properties are correlated, and the difficulty of predicting each property (for a given neural network architecture) all play a role in determining the effectiveness of MTL.

Graph convolutional neural network (GCNN) models extract information from local interactions between nodes of a graph, and transfer the learnt interactions from one local neighborhood to another, to alleviate the computational burden of DL training. Currently, GCNN models are extensively used in material science to predict material properties from atomic information by directly mapping the atomic structure input to graphs, with atoms as graph nodes and chemical bonds as edges [20, 21]. Bond angles [22] and crystallographic information [23] have also been directly included in GCNN models to improve predictive accuracy. GCNNs not only reduce the cumbersome and expensive data pre-processing, but can also naturally transfer the learning across lattices of different structures and sizes.

Recently, MTL has been combined with GCNN models to strengthen the TL property along with the ability to inject physics knowledge [24] in the DL model. In particular, the MT-CGCNN model [25] has been trained on DFT-calculated ordered compounds to simultaneously predict multiple total material properties including mixing enthalpy, Fermi energy, and band gap. Separately, GCNN models have been used to predict per-atom quantities, as with GNNFF which directly predicts atomic forces for MD [26]. However, to the best of our knowledge, no existing work in the literature has used multi-task GCNNs to simultaneously predict both global and atomic material properties. In addition, existing approaches that combine MTL with GCNN focus on one specific graph convolutional layer, without allowing the user to flexibly switch among different aggregation policies to customize the convolutional kernel to the nature of the data.

We present a novel multi-tasking GCNN, HydraGNN, to simultaneously predict multiple physical properties and demonstrate its predictive ability for binary solid solution magnetic alloys. We consider mixing enthalpy, atomic charge transfer, and atomic magnetic moment as target material properties. Once the HydraGNN model is trained, it can simultaneously produce accurate predictions of multiple material properties substantially faster than DFT calculations. As shown in figure 1, HydraGNN inputs each atomic

structure, converts it into a graph, and predicts the same properties as DFT with a GCNN model. We train HydraGNN on open-source DFT data for iron-platinum (FePt) [27] with a fixed body centered tetragonal (BCT) structure and fixed volume, generated with the LSMS-3 code [28]. Numerical results show that HydraGNN learns the dependency of the three material properties with respect to the configurational entropy over the entire compositional range.

2. HydraGNN architecture

HydraGNN directly inputs atomic structure and converts it into a graph, where atoms are interpreted as nodes and interatomic bonds are interpreted as edges, and outputs total (graph-level) and atomic (node-level) physical properties. The architecture of HydraGNN is characterized by two sets of layers: the first set of layers learn features that are common to all the material properties and the last set of layers are separated into multiple heads to learn features that are specific to each material property, shown schematically in figure 2. The shared graph convolutional layers are used to extract common relevant features from pairwise neighbor interactions and, through multiple layers, also represent many-body interactions. The following separate layers are fully connected (FC) and learn mappings between extracted features and the physical properties of interest.

We have implemented HydraGNN using `Pytorch` [29, 30] as both a robust NN library, as well as a performance portability layer for running on multiple hardware architectures. This enables HydraGNN to run on CPUs and GPUs, from laptops to supercomputers, including ORNL's Summit. The `Pytorch Geometric` [31, 32] library built on `Pytorch` is particularly important for our work and enables many GCNN models to be used interchangeably. HydraGNN is openly available on GitHub [33].

2.1. Graph convolutional layers

A graph G is usually represented in mathematical terms as

$$G = (V, \mathcal{E}) \tag{1}$$

where V represents the set of nodes and $\mathcal{E}$ represents the set of edges between these nodes [34]. An edge $(u, v) \in \mathcal{E}$ connects nodes u and v, where $u, v \in V$, $\mathcal{E} \in V \times V$. The topology of a graph can be described through the *adjacency matrix*, A, an $N \times N$ square matrix where N is the number of nodes in the graph, whose entries are associated with edges of the graph according to the following rule:

$$\begin{cases} A[u,v] = 1 & \text{iff} \quad (u,v) \in \mathcal{E} \\ A[u,v] = 0 & \text{otherwise.} \end{cases} \tag{2}$$

The degree of a node $u \in V$ is defined as:

$$d_u = \sum_{v \in V} A[u,v] \tag{3}$$

and represents the number of edges connected to a node. Every node u is represented by a a-dimensional feature vector $\mathbf{x} \in \mathbb{R}^a$ containing the embedded nodal properties and also a label vector $\mathbf{y} \in \mathbb{R}^b$ in tasks related to node-level predictions. In order to take advantage of the topology of the graph, many DL models include both the number of neighbors per node, as well as the length of each edge between nodes.

GCNNs embed the interactions between nodes without increasing the size of the input by representing the local interaction zone as a hyperparameter that cuts-off the interaction of a node with all the other nodes outside a prescribed local neighborhood. This is identical to the approximation made by many atomic simulation methods, including the LSMS-3 code used to generate the DFT training data, which ignore interactions outside a given cutoff range. GCNNs [35, 36] are DL models based on a message-passing framework, a procedure that combines the knowledge from neighboring nodes, which in our applications maps directly to the interactions of an atom with its neighbors.

The typical GCNN architecture is characterized by three different types of hidden layers: graph convolutional layers, graph pooling layers, and FC layers. The convolutional layers represent the central part of the architecture and their functionality is to transfer feature information between adjacent nodes (in this case atoms) iteratively. In the kth convolutional layer ($k = 0, 1, \ldots, K$), message passing is performed in sequential with the following operations:

(a) Aggregate information from neighbors: the node u collects the hidden embedded features of its neighbors $N(u)$ as well as the information on the edges (if available) via an aggregation function:

$$\mathbf{h}_{N(u)}^{k+1} = \text{AGGREGATE}\left(\mathbf{m}_v^k, \forall v \in N(u)\right), \tag{4}$$

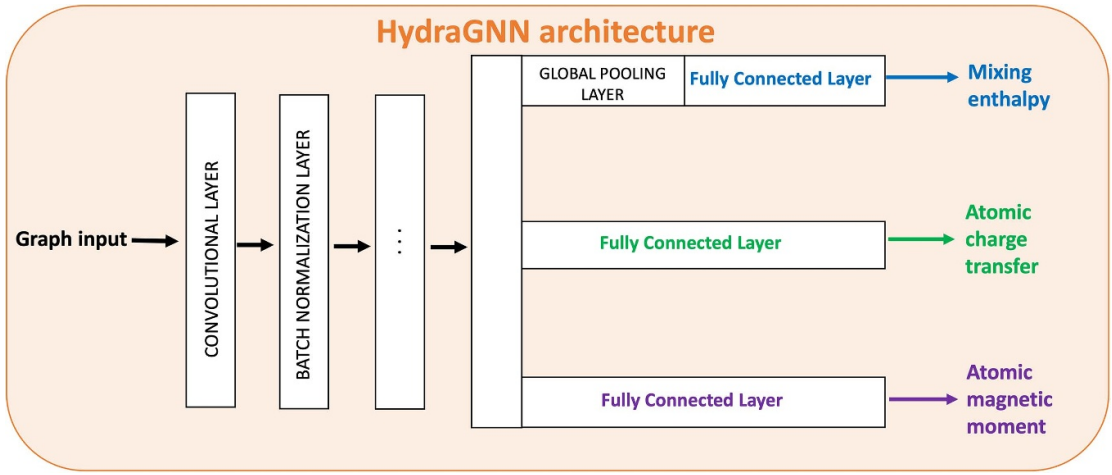

Figure 2. HydraGNN architecture when used as a surrogate model for DFT calculations of mixing enthalpy, atomic charge transfer, and atomic magnetic moment.

where $\mathbf{m}_v^k = \text{MESSAGE}(\mathbf{h}_v^k, \mathbf{h}_{e_{uv}}^k)$ is a message obtained from neighboring node v and the edge e_{uv} that connects them. The vector $\mathbf{h}_v^k$ ($\mathbf{h}_v^k \in \mathbb{R}^{p_k}$) is the embedded hidden feature vector of node v in the kth convolutional layer. When $k = 0$, the hidden feature vector is the input feature vector, $\mathbf{h}_v^0 = \mathbf{x}$.

(b) Update hidden state information: with $\mathbf{h}_{N(u)}^{k+1}$ collected, the nodal feature of node u is updated as in:

$$\mathbf{h}_u^{k+1} = \text{UPDATE}\left(\mathbf{h}_u^k, \mathbf{h}_{N(u)}^{k+1}\right) \tag{5}$$

where UPDATE is a differentiable function which combines aggregated messages $\mathbf{h}_{N(u)}^{k+1}$ from neighbors of node u with its nodal features $\mathbf{h}_u^k$ from the previous layer k.

Through consecutive steps of message passing, the graph nodes gather information from nodes that are further and further away. The type of information passed through a graph structure can be either related to the topology of the graph or features assigned to the nodes. An example of a topological information is the node degree, whereas an example of nodal feature in the context of this work is the proton number of the atom. A variety of GCNNs, e.g. principal neighborhood aggregation (PNA) [37], crystal GCNN [20] and GraphSAGE [38], have been developed, differing in the definitions of functions AGGREGATE, MESSAGE, UPDATE for message passing. One simple example of the function combination is:

$$\begin{aligned} \mathbf{h}_{N(u)}^{k+1} &= \mathbf{W}_{neighborhood}^{(k+1)} \sum_{v \in N(u)} \mathbf{h}_v^k + \mathbf{b}^k, \\ \mathbf{h}_u^{k+1} &= \sigma\left(\mathbf{W}_{self}^{(k+1)} \mathbf{h}_u^k + \mathbf{h}_{N(u)}^{k+1}\right), \end{aligned} \tag{6}$$

where $\mathbf{W}_{self}^{(k+1)}, \mathbf{W}_{neighborhood}^{(k+1)} \in \mathbb{R}^{p_{k+1} \times p_k}$ are the weights of $(k+1)$th layer of GCNN and σ is an activation function (e.g. ReLU) that introduces nonlinearity to the model.

PNA is used in this work and is one of the convolutional layers available in HydraGNN through Pytorch Geometric; PNA combines multiple aggregating techniques to reduce the risk of classifying two different graphs as identical. Batch normalizations are performed between consecutive convolutional layers along with a ReLU activation function. Graph pooling layers are connected to the end of the convolution-batch normalization stack to gather feature information from the entire graph. Global pooling layers aim at collapsing the node feature associated with each atom across a graph into a single feature. This is achieved by summing the local interactions of each atom with its neighbors and use the result to estimate global properties. For atom (node) level features such as the atomic charge transfer and atomic magnetic moment, collapsing the information from all atoms into a total system feature is not needed. FC layers are positioned at the end of the architecture to take the results of pooling, i.e. extracted features, and provide the output prediction.

Larger sizes of the local neighborhood lead to a higher computational cost to train the HydraGNN model, as the number of regression coefficients to train at each hidden convolutional layer increases proportional to the number of neighbors. Further details on the behavior of HydraGNN with different sizes of the local neighborhood have been previously reported [39].

2.2. Multiple heads with FC layers for MTL

MTL utilizes a NN to simultaneously predict multiple quantities [16] when those predicted quantities are mutually correlated and can act as inductive biases for each other. The improvement of an MTL model depends on how strongly the quantities to be predicted are mutually correlated in a particular application. This type of field specific inductive bias has been defined as *knowledge-based*, for which the training of a quantity can benefit from the information contained in the training signal for other quantities. Ultimately, MTL allows a direct and automated incorporation of physics knowledge into the model by extracting correlations between multiple quantities, with manual intervention by a domain expert only needed in determining which quantities to use.

In HydraGNN, each predicted quantity is associated with a separate loss function and the global objective function minimized during NN training is a linear combination of these individual loss functions. Formally, let T be the total number of physical quantities, or tasks, we want to predict. A single task identified by index i focuses on reconstructing a function $f_i : \mathbb{R}^a \to \mathbb{R}^{b_i}$ defined as

$$\mathbf{y}_i = f_i(\mathbf{x}), \quad i = 1, \ldots, T, \tag{7}$$

where $\mathbf{x} \in \mathbb{R}^a$, $\mathbf{y}_i \in \mathbb{R}^{b_i}$. The MTL makes use of the correlation between the quantities $\mathbf{y}_i$, where the functions f_i in (7) are replaced by a single function $\hat{f}: \mathbb{R}^a \to \mathbb{R}^{\sum_i^T b_i}$ that can model all the relations between inputs and outputs as follows:

$$\begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_T \end{bmatrix} = \hat{f}_{\mathbf{W}_{\text{MTL}}}(\mathbf{x}), \tag{8}$$

where $\mathbf{W}_{\text{MTL}}$ represents the weights to be learned.

The global loss function $\ell_{\text{MTL}} : \mathbb{R}^{N_{\text{MTL}}} \to \mathbb{R}^+$ to be minimized in MTL is a linear combination of the loss functions for the single tasks:

$$\ell_{\text{MTL}}(\mathbf{W}_{\text{MTL}}) = \sum_{i=1}^{T} \alpha_i \|\mathbf{y}_{\text{predict},i} - \mathbf{y}_i\|_2^2, \tag{9}$$

where $\mathbf{y}_{\text{predict},i} = \hat{f}_{\mathbf{W}_{\text{MTL}},i}(\mathbf{x})$ is the vector of predictions for the ith quantity of interest and α_i (for $i = 1, \ldots, T$) are the mixing weights for the loss functions associated with each single quantity. The values of the α_i's in equation (9) are hyperparameters of the surrogate model and thus can be tuned. In this work we assigned an equal weight to each property being predicted because we are equally interested in all of them; however, this definition of the loss function enables one to modify the values of the α_i to purposely favor the training toward one property of interest.

As mentioned above, the multiple quantities in MTL can be interpreted as mutual inductive biases because the error of a single quantity acts as a regularizer with respect to the loss functions of other quantities. In order to clarify why MTL can be seen as a regularizer, let us start with the formulation of a regularized training as a constrained optimization problem:

$$\begin{cases} \underset{\mathbf{w}}{\text{argmin}} & \|\mathbf{y}_{\text{predicted}}(\mathbf{w}) - \mathbf{y}\|_2^2 \\ \mathbf{c}(\mathbf{w}) = \mathbf{g} \end{cases}, \tag{10}$$

where $\mathbf{w} \in \mathbb{R}^N$ is the vector of regression coefficients of the model, $\mathbf{y}$ are the target values, $\mathbf{y}_{\text{predict}}(\mathbf{w})$ are the predictions produced by the DL model, $\mathbf{c}(\mathbf{w}) \in \mathbb{R}^c$ and $\mathbf{g} \in \mathbb{R}^c$ are quantities used to express a constraint. The formulation in (10) imposes the constraint $\mathbf{c}(\mathbf{w}) = \mathbf{g}$ in a strong form. The values of $\mathbf{c}(\mathbf{w})$ depend on the model configuration identified by the parameter vector $\mathbf{w}$, whereas the reference values $\mathbf{g}$ are assumed to be given. The constrained optimization problem in (10) can be reformulated so that the constraint is incorporated in the definition of the objective function itself as follows:

$$\underset{\mathbf{w}}{\text{argmin}} \quad \left\{ \|\mathbf{y}_{\text{predicted}}(\mathbf{w}) - \mathbf{y}\|_2^2 + \lambda \|\mathbf{c}(\mathbf{w}) - \mathbf{g}\|_* \right\}, \tag{11}$$

where $\|\cdot\|_*$ may denote the ℓ^1-norm or ℓ^2-norm as explained below. The objective function to be minimized in (11) interprets the constraint as a penalization term with the penalization multiplier λ. This is a weak formulation of the constraint added to the original objective function. Standard ℓ^1 or ℓ^2 regularizations correspond to choosing $\mathbf{c}(\mathbf{w}) = \mathbf{w}$ and $\mathbf{g} = 0$, and they recast the constraint term in (11) from the strong formulation to the weak formulation using the ℓ^1-norm and the ℓ^2-norm, respectively. *In MTL, $\mathbf{c}(\mathbf{w})$ are the predictions of additional target properties whose target values are stored in g, which allows to recast the global objective function used in equation (11) as the global loss function for MTL defined in equation (9)*. For a fair comparison and to determine the benefit of using other tasks as a mutual regularizer, we do not use additional ℓ^1 and ℓ^2 regularizers for the STL training in this work.

Figure 2 shows the topology of an HydraGNN model for multi-tasking learning to model mixing enthalpy, atomic charge transfer, and atomic magnetic moment. The architecture of HydraGNN for MTL is organized so that the first hidden layers are shared between all tasks, while keeping several task-specific output layers. This approach is known in literature as *hard parameter sharing*.

3. Solid solution binary alloy dataset

In this work we focus on a solid solution binary alloy, where two constituent elements are randomly placed on an underlying crystal lattice. We use a dataset for FePt alloys available through the OLCF Constellation [27] which includes the total enthalpy, atomic charge transfer, and atomic magnetic moment. Each atomic sample has a BCT structure with a $2 \times 2 \times 4$ supercell. The dataset was computed with LSMS-3 [28], a locally self-consistent multiple scattering) DFT application [40, 41]. The dataset was created with fixed volume in order to isolate the effects of graph interactions and graph positions for models such as GCNN. This produces non-equilibrium alloy samples, with non-zero pressure and positive mixing enthalpy, shown as a function of composition in figure 3.

The input to HydraGNN for each sample includes the three components of the atom position and the proton number. The predicted values include the mixing enthalpy, a single scalar for each sample (graph), as well as the charge transfer and magnitude of the magnetic moment, both scalars per atom (node). Although the magnetic moment is a vector quantity, we treat it as a scalar because all the atomic magnetic moments in the dataset are co-linear (all magnetic moments point in the same direction).

The dataset consists of 32 000 configurations out of the 2^{32} available, sampled every 3 atomic percent. For this work, if the number of unique configurations for a specific composition is less than 1000 all those configurations are included in the dataset; for all other compositions, configurations are randomly selected up to 1000. This results in a final dataset of 28 033 configurations. In order to ensure each composition is adequately represented in all portions of the dataset, splitting between the training, validation, and test sets is done separately for each composition.

At the ground state, the total enthalpy H of an alloy is

$$H = \sum_{i=1}^{E} c_i H_i + \Delta H_{\text{mix}}, \tag{12}$$

where E is the total number of elements in the system, c_i is the molar fraction of each element i, H_i is the molar enthalpy of each element i, and ΔH_{mix} is the mixing enthalpy. We predict the mixing enthalpy for each sample by subtracting the internal enthalpy from the DFT computed total enthalpy as a value more relevant to materials science (more directly related to the configuration). The chemical disorder makes the task of describing the material properties combinatorially complex; this represents the main difference from open source databases that have very broad elemental and structural coverage, but only include ordered compounds [42–44].

The range of values of the mixing enthalpy expressed in Rydberg is $(0.0, 65.92)$, the range of atomic charge transfer in electron charge is $(-5.31, -0.85)$, and the range of atomic magnetic moment in magnetons is $(-0.05, 3.81)$. Since different physical quantities have different units and different orders of magnitude, the inputs and outputs for each quantity are normalized between 0 and 1 across all data.

Both atomic and global properties arise from complex relations to the lattice structure. However, the relations between lattice structure and atomic properties are not fundamentally different from the relations between lattice structure and global properties for perfect lattice structures as those considered in this work. There are limitations to the amount of configurational entropy present in the small atomic systems considered here. This could be improved by increasing the size of the lattice.

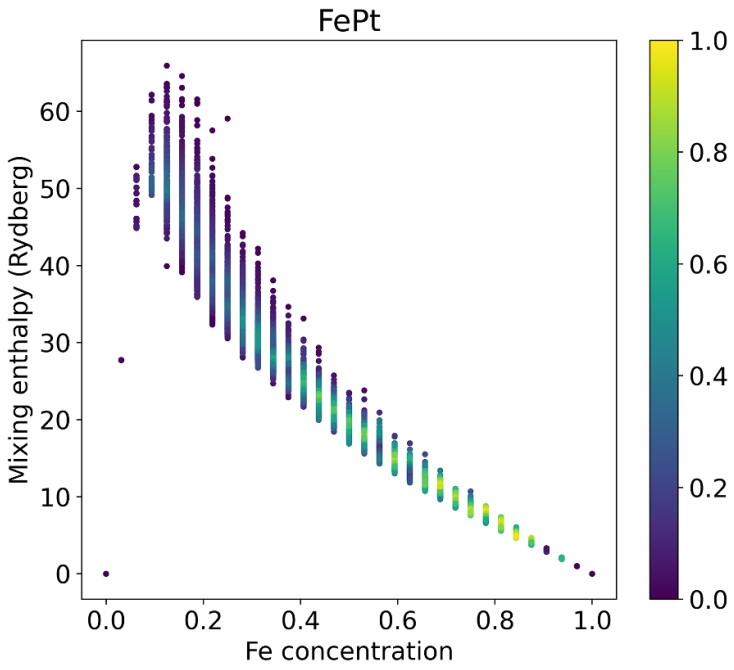

Figure 3. Configurational mixing enthalpy of solid solution binary alloy FePt with BCT structure as a function of Fe concentration. The color map indicates the relative frequency of data.

4. Numerical results

We present numerical results that predict the mixing enthalpy, atomic charge transfer, and atomic magnetic moment for the binary FePt alloy. Specifically, we compare the predictive performance of multiple separate, single-headed HydraGNN models for STL with multi-headed HydraGNN models for MTL. The output of DFT calculations is considered as the exact reference for the DL model to reconstruct.

4.1. Training setup

The architecture of the HydraGNN models has 6 PNA [37] convolutional layers with 20 neurons per layer. A radius cutoff of 7 Å is used to build the local neighborhoods used by the graph convolutional mask. Every learning task is mapped into separate heads where each head is made up of two FC layers, with 50 neurons in the first layer and 25 neurons in the second. This choice for the GCNN architecture is driven by the following considerations. On the one hand, using a small GCNN architecture leads to underfitting because the neural network is not sufficiently complex to learn all the relevant features described in the dataset. On the other hand, an overly complex neural network may trigger overfitting, especially when the dataset is relatively small in volume, as it is for this work. Moreover, a complex MTL neural network would result in similar training to STL because it separates too much the quantities of interest. However, the goal of the MTL is to extract physics correlations, which requires the quantities to overlap sufficiently within the architecture. The DL models were trained using the Adam method [45] with a learning rate equal to 0.001, batch sizes of 64, and a maximum number of epochs set to 200. Early stopping is performed to interrupt the training when the validation loss function does not decrease for several consecutive epochs, as this is a symptom that shows further epochs are very unlikely to reduce the value of the loss function. The training set for each of the NN represents 70% of the total dataset; the validation and test sets each represent half of the remaining data. As discussed in section 3, compositional stratified splitting was performed to ensure that all the compositions were equally represented across training, validation, and testing datasets. Equal loss function weights were used for all properties to define the global loss function for MTL. The training of each DL model was performed on Summit with one model per NVIDIA V100 GPU across two nodes, resulting in ensembles of 12 models per MTL/STL setup discussed in the next section.

4.2. Model accuracy and reliability

We compute and analyze not only the accuracy (root mean-squared error (RMSE)) of each model, but also the standard deviation of RMSE for an ensemble of 12 models. This simple metric enables some quantification of the uncertainty associated with each MTL or STL prediction and an understanding of the reliability of the GCNN models. Note that the RMSE for MTL may be higher compared to STL for two

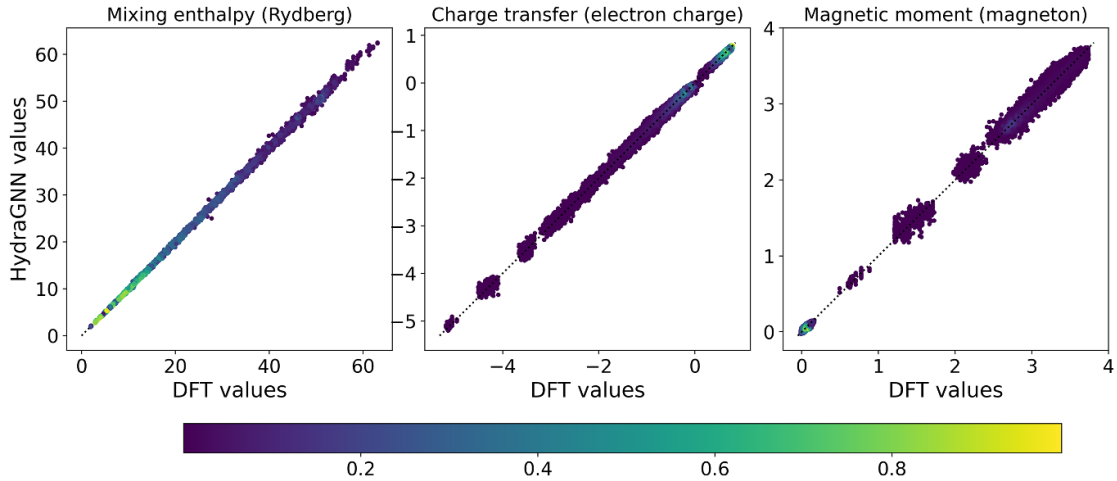

Figure 4. Prediction of FePt test set in MTL, HCM. Scatter plots of mixing enthalpy (left), atomic charge transfer (center), and atomic magnetic moment (right) for multi-task predictions obtained with HydraGNN (y-axis) against DFT calculations (x-axis). The color map indicates the relative frequency of data for each predicted property.

Table 1. Test RMSE of HydraGNN to predict physical properties for an FePt alloy. The training method states whether multi-tasking (MTL) or single-tasking (STL) was used and which quantities were predicted: mixing enthalpy (H), charge transfer (C), and/or magnetic moment (M).

	Test RMSE		
Training method	Mixing enthalpy	Charge transfer	Magnetic moment
MTL, HCM	$7.54 \times 10^{-03} \pm 8.70 \times 10^{-04}$	$6.77 \times 10^{-03} \pm 3.59 \times 10^{-04}$	$1.04 \times 10^{-02} \pm 4.94 \times 10^{-04}$
MTL, HC	$7.33 \times 10^{-03} \pm 4.77 \times 10^{-04}$	$7.36 \times 10^{-03} \pm 3.23 \times 10^{-04}$	–
MTL, HM	$6.64 \times 10^{-03} \pm 5.08 \times 10^{-04}$	–	$1.02e-02 \pm 5.23 \times 10^{-04}$
MTL, CM	–	$5.94e-03 \pm 3.02 \times 10^{-04}$	$9.30 \times 10^{-03} \pm 4.12 \times 10^{-04}$
STL, H	$1.02 \times 10^{-02} \pm 1.16e-03$	–	–
STL, C	–	$5.94 \times 10^{-03} \pm 4.39 \times 10^{-04}$	–
STL, M	–	–	$8.77 \times 10^{-03} \pm 3.18 \times 10^{-04}$

reasons. First, the number of graph convolutional layers and the number of nodes per layer are the same for MTL and STL, but MTL forces these parameters to be shared among the multiple predicted quantities (STL and MTL differ only in the split hidden layers for the heads). Second, MTL introduces an inductive bias in the predictions of a quantity under the influence of other quantities simultaneously predicted.

We first show figure 4 parity plots for the predictions generated by HydraGNN against the DFT data when MTL is used to simultaneously predict mixing enthalpy, charge transfer, and magnetic moment. HydraGNN with MTL predicts all three properties well for most of the samples over the entire dataset, as shown by the alignment of data near the diagonal. The colormap highlights that more sample points for all three properties are tightly clustered near zero, with larger variations as the property magnitudes increase. As expected, this non-uniform concentration of values for the target properties across the data affects the predictive performance of the HydraGNN models, with more accurate predictions in regions with higher concentration of data.

In table 1, we compare the predictive accuracy of HydraGNN used to simultaneously predict mixing enthalpy, atomic charge transfer, and atomic magnetic moment with MTL, MTL with each pair of properties, and STL for each individual property. The table shows the average RMSE and the standard deviation from 12 independent runs with random initialization on the test set. Figure 5 shows the probability distribution functions (PDFs) of the results from table 1 for each combination of properties. The lines mark the average values and the filled area indicates one standard deviation. Models with a lower average RMSE are interpreted as more accurate and lower standard deviation as more reliable, as the model predictions are similar across different trainings due to more relevant features extracted that better characterize the dataset. The results show that adding the magnetic moment as a physical constraint improves the both the predictive accuracy and reliability of MTL models over the STL predicting mixing enthalpy only. Addition or replacement of magnetic momement for charge density somewhat reduces the accuracy or reliability, but much less than using STL.

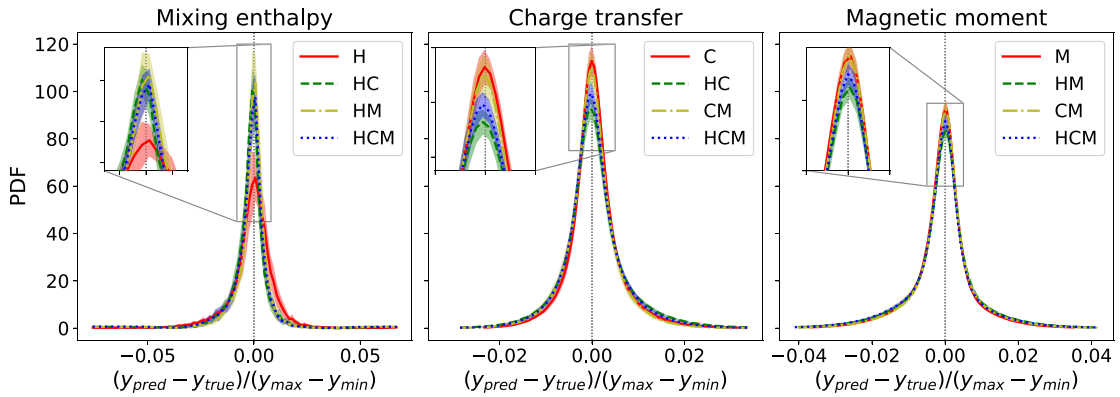

Figure 5. Comparison between prediction error of FePt test set in STL and MTL. PDF profiles of prediction errors for mixing enthalpy (left), atomic charge transfer (center), and atomic magnetic moment (right). The lines are mean values and the filled area indicates the range of standard deviation from 12 independent runs with random initialization. The capital letters are associated with the quantities predicted (H stands for mixing enthalpy, C stands for charge transfer and M stands for magnetic moment).

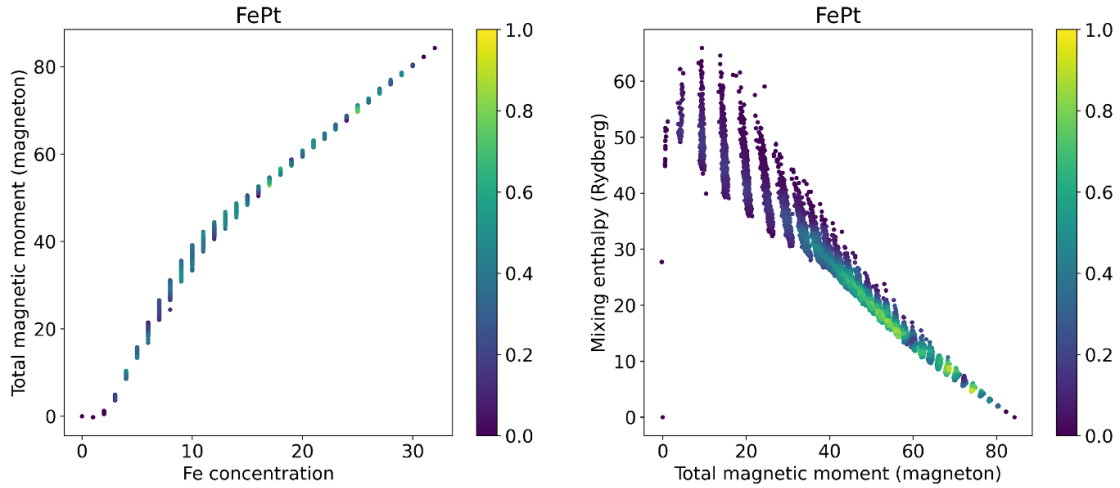

Figure 6. Scatter plot of total magnetic moment against concentration of iron (left) and scatter plot of mixing enthalpy against total magnetic moment (right).

These results suggest that there is a strong correlation between mixing enthalpy and magnetic moment, which has also been documented in the literature for ferromagnetic materials from DFT calculations and experiments [46–48], as well as previous DL studies using multilayer perceptrons for MTL on the same dataset [49]. The total magnetization of the binary alloy FePt is strongly correlated with the concentration of Fe in the alloy, as shown in figure 6 to the left, because only the Fe atoms have a non-negligible magnetic moment. This results in a strong (albeit nonlinear) correlation between the mixing enthalpy and the total magnetization of the alloy, as shown in figure 6 to the right. While this correlation between total properties is useful in determining whether the MTL training should be useful in this case, we note that the correlations between the global and atomic properties of the system are much more complex. The HydraGNN results indicate this correlation indeed helps the MTL predict the enthalpy with the addition of magnetic moment, as well as charge density.

The error distributions for MTL are slightly skewed and not centered at zero because the physical properties operate as mutual inductive biases, but this does not affect the accuracy of MTL. While MTL stabilizes the training and prediction and improves accuracy of the global graph mixing enthalpy, the same is not true of the atomic-level properties. The STL models for charge density and magnetic moment are slightly more accurate (as discussed at the beginning of the section), but also slightly more reliable for magnetic moment. The difference in terms of stabilization and accuracy obtained by MTL on the mixing enthalpy and atomic-level properties is possibly due to the different dimensionality of each property (scalar vs. high-dimensional vector).

We finally note that while our results for the mixing enthalpy are less accurate than the best reported GCNN results for DFT data ($\sim$0.21 eV atom^{-1} in this work versus 0.022–0.067 eV atom^{-1} [22, 23]). This is

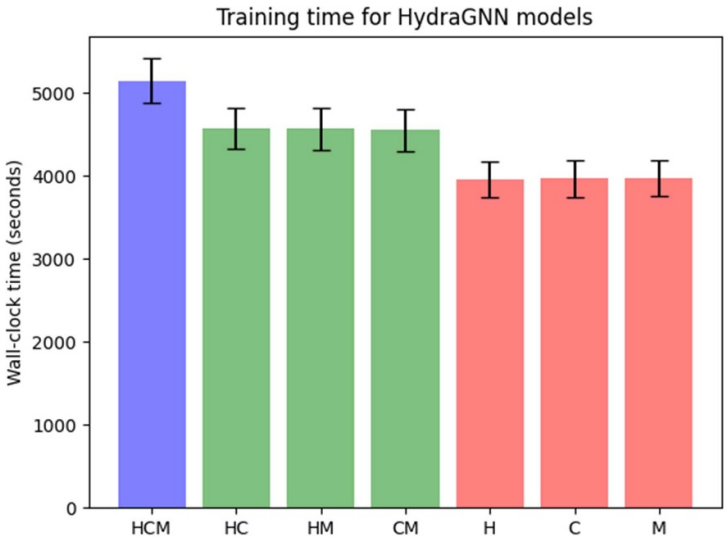

Figure 7. Training time for HydraGNN models in wall-clock seconds to simultaneously predict all three material properties (red), two properties (green) and one property (blue). The bars indicate average training time and the vertical black segments indicate the range of one standard deviation from 12 independent runs with random initialization. Each letter corresponds to one predicted quantity: mixing enthalpy (H), charge transfer (C), and magnetic moment (M).

partially due to the model used, as PNA does not explicitly include bond angle or crystallographic information, as well as the highly non-equilibrium nature of this dataset (although that the literature results include a much more broad range of chemistries should be taken into account). However, the focus of this work is on the MTL capabilities of HydraGNN and comparisons with STL.

4.3. Model training time

In figure 7, we report the average wall-clock time and standard deviation to train all the MTL and STL models for the same 12 averaged runs from section 4.2 for each combination of properties. In general, STL takes approximately 10% less time to train than MTL for two properties, which in turn takes approximately 10% less than MTL to predict all thee material properties. This is due to the fact that each head introduces additional parameters into the neural network, increasing the total computational cost. However, the training for MTL to simultaneously predict all three material properties is still more than 2.2x faster than to train three separate HydraGNN models using STL.

5. Conclusion and future work

We have presented a new DL surrogate model for atomic calculations, HydraGNN, to predict both global and atomic properties, available on GitHub [33]. The multi-tasking GCNN model was jointly trained on mixing enthalpy, charge transfer, and magnetic moment to take advantage of physical correlations in a highly non-equilibrium DFT dataset. Each predicted quantity acts as a physical constraint on the other quantities, allowing the shared convolutional layers to extract features that describe local interatomic interactions and use them across different material properties.

Although the numerical experiments presented here show that MTL can use strong correlations between physical properties to reduce the uncertainty associated with the DL predictions, imposing too many constraints can be counterproductive. Simultaneously predicting too many quantities may lead to a decrease of the predictive power, particularly when the quantities are less correlated. Therefore, the choice of different physical properties to be predicted simultaneously must be done judiciously.

The use of HydraGNN reduces the time needed to predict the material properties by a factor of hundreds compared to the original DFT calculations. Moreover, the computational time in training a multi-headed HydraGNN model is comparable to that of a single-headed GCNN model, resulting in further computational savings (near the order of the number of material properties).

We envisage two primary future applications of HydraGNN. First, the surrogate enables extremely fast estimation of atomic properties once the model has been trained, replacing DFT as long as some reduction in accuracy is acceptable. To further improve the usability for this case, we plan to integrate uncertainty quantification methods with HydraGNN to identify situations where the model requires additional training data. Second, rather than completely replace DFT we intend to integrate physics-based and data-driven

models together in scenarios where the trained HydraGNN surrogate model can act as an initial guess for a further refined DFT calculation. In computational workflows such as QMC, this could substantially reduce the total simulation time without any change in the final accuracy.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://doi.ccs.ornl.gov/ui/doi/135.

Acknowledgments

Massimiliano Lupo Pasini thanks Dr Vladimir Protopopescu and Dr Markus Eisenbach for their valuable feedback in the preparation of this manuscript.

This work was supported in part by the Office of Science of the Department of Energy and by the Laboratory Directed Research and Development (LDRD) Program of Oak Ridge National Laboratory. This research is sponsored by the Artificial Intelligence Initiative as part of the Laboratory Directed Research and Development (LDRD) Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the US Department of Energy under Contract DE-AC05-00OR22725. This work used resources of the Oak Ridge Leadership Computing Facility and of the Edge Computing program at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

ORCID iDs

Massimiliano Lupo Pasini https://orcid.org/0000-0002-4980-6924
Pei Zhang https://orcid.org/0000-0002-8351-0529
Samuel Temple Reeve https://orcid.org/0000-0002-4250-9476
Jong Youl Choi https://orcid.org/0000-0002-6459-6152

References

[1] Hoenber P and Kohn W 1964 Inhomogeneous electron gas *Phys. Rev.* **136** B864–71
[2] Kohn W and Sham L J 1965 Self-consistent equations including exchange and correlation effects *Phys. Rev.* **140** A1133–8
[3] Nightingale M P and Umrigar J C 1999 *Self-Consistent Equations Including Exchange and Correlation Effects* (Berlin: Springer)
[4] Hammond B L, Lester W A and Reynolds P J 1994 *Monte Carlo Methods in Ab Initio Quantum Chemistry* (Singapore: World Scientific)
[5] Car R and Parrinello M 1985 Unified approach for molecular dynamics and density-functional theory *Phys. Rev. Lett.* **55** 2471–4
[6] Marx D and Hutter J 2012 *Ab Initio Molecular Dynamics, Basic Theory and Advanced Methods* (New York: Cambridge University Press)
[7] Gaultois M W, Oliynyk A O, Mar A, Sparks T D, Mulholland G J and Meredig B 2016 Perspective: web-based machine learning models for real-time screening of thermoelectric materials properties *APL Mater.* **4** 2016
[8] Lu S, Zhou Q, Ouyang Y, Guo Y, Li Q and Wang J 2018 Accelerated discovery of stable lead-free hybrid organic-inorganic perovskites via machine learning *Nat. Commun.* **9** 3405
[9] Gómez-Bombarelli R *et al* 2016 Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach *Nat. Mater.* **15** 1120–7
[10] Xue D, Balachandran P V, Hogden J, Theiler J, Xue D and Lookman T 2016 Accelerated search for materials with targeted properties by adaptive design *Nat. Commun.* **7** 11241
[11] Choudhary K *et al* 2022 Recent advances and applications of deep learning methods in materials science *Npj Comput. Mater.* **8** 59
[12] Hutchinson M, Erin Antono B M Gibbons S P, Ling J, and Meredig B 2017 Overcoming data scarcity with transfer learning (arXiv:abs/1711.05099)
[13] Raissi M, Perdikaris P and Karniadakis G E 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations *J. Comput. Phys.* **378** 686–707
[14] Purja G P, Batra R and Mishin Y 2019 Physically informed artificial neural networks for atomistic modeling of materials *Nat. Commun.* **10** 2339
[15] Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S and Yang L 2021 Physics-informed machine learning *Nat. Rev. Phys.* **3** 422–40
[16] Caruana R 1993 Multitask learning: a knowledge-based source of inductive bias *Mach. Learn.* **48** 41–48
[17] Collobert R and Weston J 2008 A unified architecture for natural language processing: deep neural networks with multitask learning *Proc. 25th Int. Conf. on Machine Learning* vol 7 pp 160–7
[18] Ramsundar B, Liu B, Wu Z, Verras A, Tudor M, Sheridan R P and Pande V 2017 Is multitask deep learning practical for pharma? *J. Chem. Inf. Model.* **57** 2068–76
[19] Lupo Pasini M, Li Y W, Yin J, Zhang J, Barros K and Eisenbach M 2020 Fast and stable deep-learning predictions of material properties for solid solution alloys *J. Phys.: Condens. Matter.* **33** 084005
[20] Xie T and Grossman J C 2018 Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties *Phys. Rev. Lett.* **120** 145301

[21] Chen C, Ye W, Zuo Y, Zheng C and Ong S P 2019 Graph networks as a universal machine learning framework for molecules and crystals *Chem. Mater.* **31** 3564–72
[22] Choudhary K and Brian D 2021 Atomistic line graph neural network for improved materials property predictions *npj Comput. Mater.* **7** 1–8
[23] Park C W and Wolverton C 2020 Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery *Phys. Rev. Mater.* **4** 063801
[24] Kim R and Ning Y 2022 Recurrent multi-task graph convolutional networks for COVID-19 knowledge graph link prediction *Driving Scientific and Engineering Discoveries Through the Integration of Experiment, Big Data, and Modeling and Simulation. SMC 2021. Communications in Computer and Information Science* **1512** 411–9
[25] Sanyal S, Balachandran J, Yadati N, Kumar A, Rajagopalan P, Sanyal S and Talukdar P P 2018 MT-CGCNN: integrating crystal graph convolutional neural network with multitask learning for material property prediction (arXiv:abs/1811.05660)
[26] Park C W, Kornbluth M, Vandermause J, Wolverton C, Kozinsky B and Mailoa J P 2020 Accurate and scalable multi-element graph neural network force field and molecular dynamics with direct force architecture (arXiv:2007.14444)
[27] Lupo Pasini M and Eisenbach M 2021 FePt binary alloy with 32 atoms - LSMS-3 data (https://doi.org/10.13139/OLCF/1762742)
[28] Eisenbach M, Li Y W, Odbadrakh O K, Pei Z, Stocks G M and Yin J 2017 LSMS
[29] Paszke A *et al* 2019 Pytorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems* **vol 32**, ed H Wallach, H Larochelle, A Beygelzimer, F dAlché-Buc, E Fox and R Garnett (Red Hook, NY: Curran Associates, Inc.) pp 8024–35
[30] Paszke A *et al* 2019 PyTorch: an imperative style, high-performance deep learning ibrary (https://doi.org/10.48550/arXiv.1912.01703)
[31] Fey M and Lenssen J E 2019 Fast graph representation learning with PyTorch Geometric *ICLR Workshop on Representation Learning on Graphs and Manifolds*
[32] Fey M and Lenssen J E PyTorch Geometric 2019 Fast graph representation learning with pyTorch geometric (https://github.com/pyg-team/pytorch_geometric) (https://doi.org/10.48550/arXiv.1903.02428)
[33] Lupo Pasini M, Reeve S T, Zhang P and Choi J Y 2021 HydraGNN [Computer Software] (https://doi.org/10.11578/dc.20211019.2)
[34] Murty U S R and Bondy J A 1976 Graphs and subgraphs *Graph Theory with Applications* (New York: Elsevier)
[35] Scarselli F, Gori M, Tsoi A C, Hagenbuchner M and Monfardini G 2009 The graph neural network model *IEEE Trans. Neural Netw.* **20** 61–80
[36] Defferrard M, Bresson X and Vandergheynst P 2016 Convolutional neural networks on graphs with fast localized spectral filtering *Advances in Neural Information Processing Systems* **vol 29**, ed D Lee, M Sugiyama, U Luxburg, I Guyon and R Garnett (Red Hook, NY: Curran Associates, Inc.)
[37] Corso G, Cavalleri L, Beaini D, Lió P and Petar Včkovic Principal neighbourhood aggregation for graph nets (arXiv:2004.05718 [cs, stat])
[38] Hamilton W L, Ying R and Leskovec J 2017 Inductive representation learning on large graphs *Proc. 31st Int. Conf. on Neural Information Processing Systems* pp 1025–35
[39] Lupo Pasini M, Burčul M, Reeve S T, Eisenbach M and Perotto S 2021 Fast and accurate predictions of total energy for solid solution alloys with graph convolutional neural networks *Driving Scientific and Engineering Discoveries Through the Integration of Experiment, Big Data, and Modeling and Simulation. SMC 2021. Communications in Computer and Information Science* **1512** 79–98
[40] Eisenbach M, Larkin J, Lutjens J, Rennich S and Rogers J H 2017 GPU acceleration of the locally self-consistent multiple scattering code for first principles calculation of the ground state and statistical physics of materials *Comput. Phys. Commun.* **211** 2–7
[41] Wang Y, Stocks G M, Shelton W A, Nicholson D M C, Szotek Z and Temmerman W M 1995 Order-N multiple scattering approach to electronic structure calculations *Phys. Rev. Lett.* **75** 2867–70
[42] Curtarolo S *et al* 2012 AFLOW: an automatic framework for high-throughput materials discovery *Comput. Mater. Sci.* **58** 218–26
[43] Jain A *et al* 2013 Commentary: the materials project: a materials genome approach to accelerating materials innovation *APL Mater.* **1** 0–11
[44] Saal J E, Kirklin S, Aykol M, Meredig B and Wolverton C 2013 Materials design and discovery with high-throughput density functional theory: the Open Quantum Materials Database (OQMD) *JOM* **65** 1501–9
[45] Kingma D P and Jimmy B 2017 Adam: a method for stochastic optimization (arXiv:1412.6980 [cs])
[46] Staunton J B, Razee S S A, Ling M F, Johnson D D and Pinksi F J 1998 Magnetic alloys, their electronic structure and micromagnetic and microstructural models *J. Phys. D: Appl. Phys* **31** 2355
[47] Hou T P, Liu K M, Peet M J, Hulme-Smith C N, Guo L and Zhang L 2018 Magnetism and high magnetic-field-induced stability of alloy carbides in fe-based materials *Sci. Rep.* **8** 7884
[48] Marshal A, Pradeep K G, Music D, Wang L, Petracic O and Schneider J M 2019 Combinatorial evaluation of phase formation and magnetic properties of femncocral high entropy alloy thin film library *Sci. Rep.* **9**
[49] Lupo Pasini M, Yin J, Li Y W and Eisenbach M 2021 A scalable constructive algorithm for the optimization of neural network architectures *Parallel Comput.* **104–105** 102788