npj | Computational Materials

www.nature.com/npjcompumats

ARTICLE OPEN

Check for updates

AtomSets as a hierarchical transfer learning framework for small and large materials datasets

Chi Chen[1] and Shyue Ping Ong[1]✉

Predicting properties from a material's composition or structure is of great interest for materials design. Deep learning has recently garnered considerable interest in materials predictive tasks with low model errors when dealing with large materials data. However, deep learning models suffer in the small data regime that is common in materials science. Here we develop the AtomSets framework, which utilizes universal compositional and structural descriptors extracted from pre-trained graph network deep learning models with standard multi-layer perceptrons to achieve consistently high model accuracy for both small compositional data (<400) and large structural data (>130,000). The AtomSets models show lower errors than the graph network models at small data limits and other non-deep-learning models at large data limits. They also transfer better in a simulated materials discovery process where the targeted materials have property values out of the training data limits. The models require minimal domain knowledge inputs and are free from feature engineering. The presented AtomSets model framework can potentially accelerate machine learning-assisted materials design and discovery with less data restriction.

npj Computational Materials (2021)7:173 ; https://doi.org/10.1038/s41524-021-00639-w

INTRODUCTION

Machine learning (ML) has garnered substantial interest as an effective method for developing surrogate models for materials property predictions in recent years.[1,2] However, a critical bottleneck is that materials datasets are often small and inhomogeneous, making it challenging to train reliable models. While large density functional theory (DFT) databases such as the Materials Project,[3] Open Quantum Materials Database,[4] and AFLOWLIB[5] have ~O(10^6) relaxed structures and computed energies, data on other computed properties such as band gaps, elastic constants, dielectric constants, etc. tend to be several times or even orders of magnitude fewer.[2] In general, deep learning models based on neural networks tend to require much more data to train, resulting in lower performance in small datasets relative to non-deep learning models. For example, Dunn et al.[6] have found that while graph deep learning models such as the MatErials Graph Networks (MEGNet)[7] and Crystal Graph Convolutional Neural Network (CGCNN)[8] can achieve state-of-the-art performance for datasets with $>$O(10^4) data points, ensembles of non-deep-learning models (using AutoMatminer) outperform these deep learning models when the data set size is $<$O(10^4), and especially when the data set is $<$O(10^3).

Several approaches have been explored to address the data bottleneck. The most popular approach is transfer learning (TL), wherein the weights from models trained on a property with a large data size are transferred to a model on a smaller data size. Most TL studies were performed on the same property.[9–11] For example, Hutchinson et al.[9] developed three TL approaches that reduced the model errors in predicting experimental band gaps by including DFT band gaps. Similarly, Jha et al.[10] trained models on the formation energies in the large OQMD database and demonstrated that transferring the model weights from OQMD can improve the models on the small DFT-computed and even experimental formation energy data. More interestingly, Frey et al.[11] used pre-trained bulk MEGNet models on formation energy, band gap, and Fermi energy and then transferred such models to learn corresponding properties of 2D materials. TL has also been demonstrated between different properties in some cases. For example, the present authors[7] found that transferring the weights from large data-size formation energy MEGNet models to smaller-data-size band gap and elastic moduli models improved convergence rate and accuracy. Another approach uses multi-fidelity models, where datasets of multiple fidelities (e.g., band gaps computed with different functionals or measured experimentally) are used to improve prediction performance on the more valuable, high fidelity properties. For example, two-fidelity co-kriging methods have demonstrated successes in improving the predictions of the Heyd–Scuseria–Ernzerhof (HSE)[12] band gaps of perovskites[13], defect energies in hafnia[14] and DFT bulk moduli.[15] In a recently published work, the present authors also developed multi-fidelity MEGNet models that utilize band gap data from four DFT functionals (Perdew-Burke-Ernzerhof[16] or PBE, Gritsenko–Leeuwen–Lenthe–Baerends with solid correction[17,18] or GLLB-SC, strongly constrained and appropriately normed[19] or SCAN and HSE[12]) and experimental measurements to significant improve the prediction of experimental band gaps.[20]

In this work, we demonstrate that a pre-trained MEGNet formation energy model can be used as encoders to generate universal compositional and structural features for materials. These features can then be used in standard multi-layer perceptron (MLP) models to predict diverse properties with different data sizes accurately. This AtomSets framework unifies compositional and structural features under one umbrella. Using 13 MatBench datasets,[6] we show that AtomSets models can achieve excellent performance even when the inputs are compositional and the data size is small (~300), while retaining MEGNet's state-of-the-art performance on properties with large data sizes. Furthermore, the model construction requires minimal domain knowledge and no feature engineering.

[1]Department of NanoEngineering, University of California San Diego, 9500 Gilman Dr, Mail Code 0448, La Jolla, CA 92093-0448, USA. ✉email: ongsp@eng.ucsd.edu

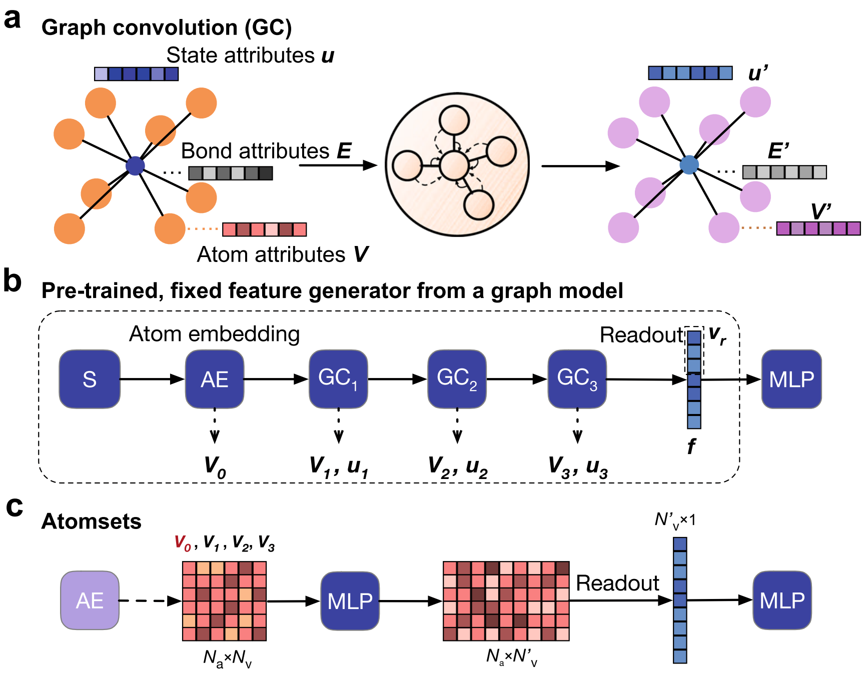

Fig. 1 Graph networks and AtomSets schematics. a The graph convolution (GC) takes an input graph with the labeled atom (**V**), bond (**E**), and state (**u**) attributes and outputs a new computed graph with updated attributes. **b** The graph network model architecture. The input to the model is the structure graph (S) with atomic number as the atom attributes. Then the graph is passed to an atom embedding (AE) layer, followed by three GC layers. After the GC, the graph is read out to a structure-wise vector *f*, and *f* is further passed to multi-layer perceptron (MLP) models. Within the model, each layer output is captured for later use. **c** The AtomSets model takes a site-wise/element-wise feature matrix and passes to MLP layers. After the MLP, a readout function is applied to derive a structure-wise/formula-wise vector, followed by final MLP layers.

RESULTS

Materials graph networks and the AtomSets framework

The MEGNet formalism has been described extensively in previous works[7,20] and interested readers are referred to those publications for details. Briefly, the MEGNet framework featurizes a material into a graph $G = (V, E, \mathbf{u})$, where $\mathbf{v}_i \in V$ are the atom or node features, $\mathbf{e}_k \in E$ are the edges or bonds, and $\mathbf{u}$ are state features. The features matrices/vectors are $\mathbf{V} = [\mathbf{v}_1; \ldots; \mathbf{v}_{N_a}] \in \mathbb{R}^{N_a \times N_f}$, $\mathbf{E} = [\mathbf{e}_1; \ldots; \mathbf{e}_{N_b}] \in \mathbb{R}^{N_b \times N_{bf}}$ and $\mathbf{u} \in \mathbb{R}^{N_u}$, where N_a, N_b, N_f, N_{bf} and N_u are the number of atoms, bonds, atom features, bond features, state features, respectively. For compositional models, N_a is the number of atoms in the formula. For simplicity, the atom and bond features are represented as matrices. However, shuffling the first dimension does not change the results of the models. Hence, the atoms and bonds are essentially sets. A graph convolution (GC) operation uses the connectivity of bonds to transform input graph features $(\mathbf{V}, \mathbf{E}, \mathbf{u})$ to output graph features $(\mathbf{V}', \mathbf{E}', \mathbf{u}')$, as shown in Fig. 1a.

In the initial structural graph (S), the atom attributes are simply the atomic number of the element embedded into a vector space via an atom embedding (AE) layer ($AE : \mathbb{Z} \to \mathbb{R}^{N_f^0}$) to obtain $\mathbf{V_0} \in \mathbb{R}^{N_a \times N_f^0}$, as shown in Fig. 1b, where N_f^0 is the embedded atom dimension. The bonds are constructed by considering atom pairs within a certain cutoff radius R_c. With each GC layer, information is exchanged between atoms, bonds, and state. As more GC layers are stacked (e.g., GC_2 and GC_3 in Fig. 1b), information on each atom can be propagated to further distances.

In this work, a MEGNet model with three GC layers was first trained on the formation energies of more than 130,000 Materials Project crystals as of Jun 1, 2019, henceforth referred to as the parent model. The training procedures and hyperparameter settings of the MEGNet models are similar to the previous work.[7]

The AtomSets framework uses atom-wise features as inputs, as shown in Fig. 1c. In principle, any atom-wise features can be used. If the features are element-based features, e.g., the V_0 elemental features, then the AtomSets models become purely compositional. Likewise, site-wise features that encode local geometric information, such as V_1, V_2, and V_3, can be provided as the AtomSets inputs, and the models can predict structure-relevant properties. In the non-TL AtomSets models, the compositional feature inputs to the models are the atomic numbers, where they undergo a learnable embedding process. In our TL framework, the output atom $\mathbf{V}_i = [\mathbf{v}_1^{(i)}; \ldots; \mathbf{v}_{N_a}^{(i)}] (i = 0, 1, 2, 3)$ features after each embedding or GC layer are extracted from the parent model and transferred as inputs of AtomSets models for other properties, i.e., the MEGNet formation energy model functions purely as an encoder of a material into compositional or structural features. Bond features are not considered in TL since the number of bonds depends on the graph construction settings and parameters, such as cutoff radius. As shown in Fig. 1c, an AtomSets model takes the atom-wise features $\mathbf{V}_i$ matrix of shape $N_a \times N_f$ as inputs, and each row of the feature matrix is passed to an MLP model. These features can either be compositional, e.g., elemental properties, or structural, e.g., local environment descriptor. Afterward, the output feature matrix is read out to a vector, compressing the atom number dimension.

The readout vector can be used to predict properties with the help of MLP or other models, as shown in Fig. 1c. The feature matrix can either be taken as the pre-trained model generated feature matrices $\mathbf{V_i}$ $(i = 0, 1, 2, 3)$ or trained on the fly via a trainable atom embedding layer prepended to the model. When the site-wise/atom-wise features are computed from pre-trained models, information gained from previous model training is retained, and effectively the AtomSets models transfer-learn part of the pre-trained models. A hierarchical TL scheme is achieved by including different GC outputs. The AtomSets models can also be used without transfer learning by training the elemental embedding and hence atom-wise features from the data.

Table 1. Models investigated in this work, categorized by the models types, i.e., compositional (C) or structural (S), and whether they utilize transfer learning (TL).

Model name	Type	TL	Description
AtomSets	C	No	Compositional models directly trained from data
AtomSets-V_0	C	Yes	Compositional models transferring learned V_0 from the parent formation energy model
AtomSets-V_i ($i = 1, 2, 3$)	S	Yes	Structural models transferring learned V_1, V_2 or V_3 features from the parent formation energy model
MLP-V_0-stats	C	Yes	Compositional MLP models using statistics calculated on V_0 from the parent formation energy model as inputs
MLP-u_i ($i = 1, 2, 3$), MLP-f and MLP-v_r	S	Yes	MLP models using learned u_i, f or v_r from the parent formation energy model.
MEGNet	S	No	Graph network models trained directly using each data set without transfer learning

In our definition, S-type models contain compositional information as a superset. It should be noted that the MLP-u_i, MLP-f and MLP-v_r are classified as S-type models because u_i, f or v_r implicitly incorporate structural information due to information passing in the graph convolution layers.

The AtomSets framework is flexible in the choice of input features. For example, if the symmetry functions are provided as inputs, then the AtomSets model becomes the high-dimensional neural network potential.[21] The AtomSets framework also shares similarity with the Deepsets[22] model where the summation of feature vectors are used to get the readout vectors. Since only simple MLP are underlying the AtomSets framework, the model training can be extremely fast. Models investigated in the current work are provided in Table 1. More details about the graph convolutions and readout functions are provided in the Methods section.

Model accuracies

The models were trained on the 13 materials datasets in the MatBench repository[6]. A summary is provided in Supplementary Table 1, where the data sizes range from 312 to 132,752, with both compositional and structural data. The tasks include regression and classification. Detailed summaries are provided in the work by Dunn et al.[6] Before the model metrics evaluations, we performed a rough initial search of model hyperparameter combinations as listed in Supplementary Table 2. The MAE of regression and the AUC of classification for various tasks are shown in Table 2. In addition, hyperparameter optimization was carried out on the AtomSets-V_0 and V_1 models (see Supplementary Table 3), but did not seem to have a significant effect on model performance. Here, we will focus our discussion on the models without further hyperparameter optimization for individual models. To frame our analysis, we will first recapitulate that a key finding of Dunn et al.[6] is that MEGNet models tend to outperform other models when the data size is large (>10,000 data points) but underperform for small data sizes. This can be seen in the last two columns of Table 2, where AutoMatminer models achieve lower MAEs for the small yield strength, exfoliation energies, and refractive index datasets compared to MEGNet.

The AtomSets models do not suffer from the same data size tradeoff observed in the MEGNet models. With a few notable exceptions, the transfer-learned AtomSets models usually achieve close to the best performance (lowest MAE or highest AUC, with the error bar) among all models studied. For the small yield strength and refractive index datasets, AtomSets models perform similarly to AutoMatminer, while for the larger formation energies (Perovskite and MP E_f) and MP band gap (E_g) datasets, AtomSets models perform similarly to MEGNet. The only dataset where the AtomSets and MEGNet models substantially underperform relative to AutoMatminer is the JDFT-2D exfoliation energy data, where the data size is very small. We have also investigated changing the random seeds for the data splitting, and the results are presented in Supplementary Table 4. Overall the results are consistent for AtomSets-V_0 and AtomSets-V_1 but show discrepancies at small datasets such as the JDFT-2D exfoliation energy and the refractive indices. As expected, the results suggest higher model variances when fitted on smaller datasets.

A somewhat surprising observation is that several target properties show minimal dependency on structural information. For example, the average MAEs of the compositional AtomSets-V_0 models and structural AtomSets-V_1 models for the JDFT-2D exfoliation energy, the MP phonon DOS peak, and the refractive index datasets are within the standard deviation. The structural AtomSets-V_1models for the MP elasticity data ($\log K_{\mathrm{VRH}}$ and $\log G_{\mathrm{VRH}}$) only exhibit minor improvements in average MAEs over the compositional AtomSets-V_0 models. To investigate the implications of this observation, we analyzed the polymorphs for each composition in the elasticity data set, see Supplementary Fig. 1. Out of the 10,987 elasticity data, 81% of them do not have polymorphs. For those materials, structural models likely perform similarly to the compositional models. For compositions with more than one polymorph (816 out of 9723 unique compositions), we calculated the range of the target values for polymorphs, as shown in Supplementary Figs. 1b, c. The majority of the polymorphs for the same composition have similar bulk and shear moduli, and the average ranges (max-min for the same compositions) for $\log K_{\mathrm{VRH}}$ and $\log G_{\mathrm{VRH}}$ are 0.134 and 0.158, respectively. If we include compositions with no polymorphs, i.e., range equals zero, the average ranges for $\log K_{\mathrm{VRH}}$ and $\log G_{\mathrm{VRH}}$ are 0.011 and 0.013, respectively. Such small ranges for each composition suggest that composition explains most of the variation in bulk moduli, which is why the accuracy differences between AtomSets-V_0 and AtomSets-V_1 are minimal. This observation also gives a glance at why compositional models have been reasonably successful. It should be noted that there are well-known polymorphs with vastly different mechanical properties, e.g., diamond and graphite carbon, and the AtomSets-V_1 provide far better predictions. For example, the AtomSets-V_1 model predicts the shear moduli of graphite (96 GPa) and diamond (520 GPa) to be 96 GPa and 490 GPa, respectively, while the AtomSets-V_0 model predicts them to be 177 GPa. In contrast, the perovskites and MP formation energy datasets require structural models to achieve accurate results. This observation is consistent with a recent study by Bartel et al.[23] For further verification of the polymorph effects, we have screened out polymorphs with diverse properties from the MP E_f, MP $\log(K_{\mathrm{VRH}})$ and MP $\log(G_{\mathrm{VRH}})$ datasets and evaluated the model errors on those polymorph datasets. The AtomSets-V_1 structural models consistently outperform the compositional AtomSets-V_0 models by a substantial margin in all polymorph datasets, as shown in Supplementary Table 5.

Comparing AtomSets models with various **V**'s, the results show that the features extracted from earlier stage GC layers, e.g., $\mathbf{V_0}$ and $\mathbf{V_1}$, are more generalizable and have higher accuracy

Table 2. Performance of AtomSets models relative to state-of-the-art models using the same five-fold random splitting methods and the random seed of 18012019.

Target, data size	AtomSets	AtomSets-V_0	AtomSets-V_1	AtomSets-V_2	AtomSets-V_3	MLP-f	MEGNet[6]	AutoMatminer[6]
Regression tasks								
Yield strength (MPa), 312[a]	104 ± 15	102 ± 11	–	–	–	–	–	**95**
E_{exfo} (meV atom^{-1}), 636[b]	52 ± 11	52 ± 12	51 ± 8	50 ± 10	57 ± 10	48 ± 8	56	**39**
PhonDOS peak (cm^{-1}), 1265[c]	63 ± 12	53 ± 15	51 ± 6	84 ± 13	78 ± 17	113 ± 8	**37**	51
Expt. E_g (eV), 4604[d]	0.43 ± 0.03	**0.41 ± 0.03**	–	–	–	–	–	**0.42**
n, 4764[e]	0.36 ± 0.07	0.35 ± 0.08	**0.32 ± 0.08**	0.36 ± 0.06	0.35 ± 0.08	0.38 ± 0.08	0.48	**0.30**
log(K_{VRH}) (GPa), 10987[f]	0.08 ± 0.00	0.08 ± 0.00	**0.07 ± 0.00**	0.09 ± 0.00	0.09 ± 0.00	0.11 ± 0.00	**0.07**	**0.07**
log(G_{VRH}) (GPa), 10987[g]	0.11 ± 0.00	0.11 ± 0.00	**0.09 ± 0.00**	0.10 ± 0.00	0.10 ± 0.00	0.12 ± 0.00	**0.09**	**0.09**
Perovskite E_f (meV atom^{-1}), 18928[h]	82 ± 1	82 ± 2	12 ± 0	25 ± 1	24 ± 1	30 ± 0	**8**	39
MP E_g (eV), 106113[i]	0.26 ± 0.01	0.26 ± 0.00	**0.25 ± 0.00**	0.31 ± 0.00	0.31 ± 0.01	0.34 ± 0.01	**0.24**	0.28
MP E_f (meV atom^{-1}), 132752[j]	94 ± 1	95 ± 1	49 ± 1	73 ± 1	61 ± 1	**29 ± 0**[n]	33	173
Classification tasks								
Expt. metallicity, 4921[k]	**0.94 ± 0.01**	**0.95 ± 0.01**	–	–	–	–	–	0.92
Glass forming ability, 5680[l]	**0.91 ± 0.02**	**0.92 ± 0.01**	–	–	–	–	–	0.86
MP metallicity, 106113[m]	0.95 ± 0.00	0.95 ± 0.00	0.96 ± 0.00	0.95 ± 0.00	0.96 ± 0.00	0.96 ± 0.00	**0.98**	0.91

The average and standard deviations of the MAE and AUC are reported for regression and classification tasks, respectively. The properties are sorted by dataset size. Some structural models (e.g., AtomSets-V_1/V_2/V_3 for experimental band gaps) cannot be constructed as the dataset does not contain structural information. The best performing model(s) within the standard deviation are bolded for each target.

[a] Steel yield strength data from Citrine Informatics.[37]

[b] Exfoliation energy of crystals from JARVIS DFT 2D dataset.[38]

[c] Phonon DOS peak frequency from Materials Project.[39]

[d] Experimental composition-band gap dataset from Zhuo et al.[40]

[e] Refractive index from Materials Project.[41]

[f] Log of computed bulk moduli from Materials Project.[42]

[g] Log of computed shear moduli from Materials Project.[42]

[h] Computed perovskite formation energy from Castelli et al.[43]

[i] Computed PBE band gap data from Materials Project[3,44].

[j] Computed PBE formation energy data from Materials Project[3,44].

[k] Experimental metallicity (binary) from Zhuo et al.[40]

[l] Glass forming ability (binary) from Landolt–Bornstein Handbook[45].

[m] Computed PBE metallicity (binary) from Materials Project.[3,44]

[n] Random splitting gives 15 meV atom^{-1}. The 29 meV atom^{-1} result is from splitting the data using the pre-trained model splitting.

in all models compared to those produced by later GC layers. The structure-wise state vectors, $\mathbf{u_i}(i = 1, 2, 3)$, and the readout atom feature vector $\mathbf{v_r}$, are relatively poor features, as shown by the significant errors in all models in Supplementary Table 6. However, the final structure-wise readout vector $\mathbf{f}$, along with MLP models, offers excellent accuracy in MP metallicity and formation energy tasks. We further isolated the effects of structures and studied models with only structural inputs (no element information). We replaced all elements in the benchmark data with H and then generated features based on the replaced structures. As expected, the model errors increase substantially for all datasets, as shown in Supplementary Table 7. Surprisingly, the AtomSets classification models using structure-only information on the MP metallicity dataset show an AUC of 0.92. Common neighbor analysis[24] of these structures reveals that the non-metals tend to have sites that are of low symmetries, which is not the case for metals, as shown in Supplementary Fig. 2. Hence, the geometric information alone can already distinguish metals and non-metals, explaining the high AUC of AtomSets models with only geometry inputs.

Overall, the combined AtomSets-V_0 and AtomSets-V_1 have either similar or better accuracy than AutoMatminer models. The combination also achieves close accuracy in the large structural data limit compared to the MEGNet models but at a much smaller computational cost during model training.

Model convergence

A convergence study of the best models - two compositional models, i.e., AtomSets, AtomSets-V_0, and two structural models, i.e., AtomSets-V_1 and MLP-f - was performed relative to data size. Different data sizes in terms of the fractions of maximum available data are applied. Comparing the two compositional models, the AtomSets-V_0 model achieves relatively higher performance throughout all the tasks and generally converges faster than the non-TL counterpart, i.e., the AtomSets model, as shown in Fig. 2. For the structural datasets in Fig. 2c, d, consistent with previous benchmark results, the structural AtomSets-V_1 and MLP-f models are generally more accurate than the compositional models. The rapid convergence of the MLP-f models in the MP formation energy dataset is expected since the structural features $\mathbf{f}$ were generated by the formation energy MEGNet models in the first place. Model convergences on the rest of the datasets are provided in Supplementary Fig. 3.

The model performance is also probed at tiny datasets. We used several MP datasets in this study to obtain consistent results and then down-sampled the datasets at 100, 200, 400, 600, 1000, and 2000 data points. For comparison, we also include the non-TL MEGNet structural models, as shown in Fig. 3. Similar to the previous convergence study at relatively large data sizes, the TL compositional models AtomSets-V_0 outperform the non-TL compositional AtomSets models at all data sizes. For structural models, the TL AtomSets-V_1 models achieve consistent accuracy at

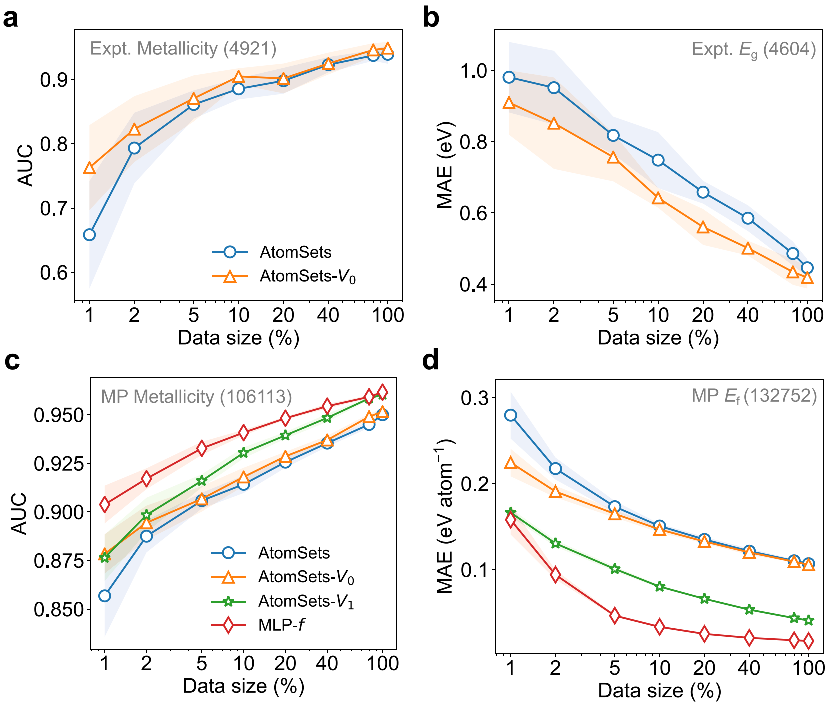

Fig. 2 Model convergence for AtomSets, AtomSets-V_0, AtomSets-V_1 and MLP-*f*. **a** and **b** show the small compositional datasets, and **c** and **d** are for the large structural datasets. **a** and **c** show the area under the curve (AUC) for classification tasks, and **b** and **d** show the mean-absolute error (MAE) for regression tasks. The *x*-axis is plotted on a log scale to provide improved resolution at small data sizes. The shaded areas are the standard deviation across five random data fitting. Additional model results are shown in Supplementary Fig. 3.

small data limits for all four tasks and consistently outperform the non-TL MEGNet models.

Interestingly, the MLP-*f* models specialize in MP metallicity data and MP formation energy data, same as previous benchmark results shown in Table 2. In particular, the MLP-*f* models converge rapidly for the MP metallicity task, with AUC exceeding 85% with only 200 data points and 90% with only 1000 data points. The MLP-*f* models also reach ~0.2 eV atom^{-1} errors on the MP formation energy data when the data size is 600. In both cases, the MLP-*f* models outperform MEGNet models by a considerable margin. However, in terms of generalizability, the AtomSets-V_1 models seem to be a better fit for all generic tasks.

At a data size of 600 (533 train data points), the formation energy and the band gap models errors of AtomSets-V_1 are 0.2 eV atom^{-1} and 0.702 eV, respectively, much lower than the errors achieved by the full MEGNet models with 0.367 eV atom^{-1} and 0.78 eV. The AtomSets-V_1 errors at such small data regimes are on par with the 0.210 eV atom^{-1} and 0.71 eV errors (504 train data points) reported by the MODNet models[25] that specialize in small materials data fitting. Interestingly, the compositional model AtomSets-V_0 also achieved lower errors than full MEGNet, with formation energy model errors of 0.269 eV atom^{-1} and band gap model errors of 0.72 eV.

Model extrapolability

In a typical materials design problem, the target is not finding a material with similar performance as existing materials, but rather materials with extraordinary properties outside the current materials pool. Such extrapolation presents a major challenge for most ML models. Previous works have generally used leave-one-cluster-out cross-validation (LOCO CV)[26] or k-fold forward cross-validation[27] to evaluate the models' extrapolation ability in data regions outside the training data. Here we adopted the concept of forward cross-validation by splitting the data according to their target value ranges and applied the method to elasticity data (MP *log*(K_{VRH}) and MP *log*(G_{VRH})) to simulate the process of finding super-incompressible (high *K*) and superhard (roughly high *G*) materials. The materials with the top 10% highest target values were held out as the test dataset (high-test, extrapolation). Then the remaining data was split into the train, validation, and test (low-test, interpolation) datasets, making two test data regimes in total. We selected AtomSets, AtomSets-V_0, AtomSets-V_1, and the MEGNet models for the comparison. For the bulk modulus *K*, the low-test errors for the compositional models AtomSets and AtomSets-V_0 are identical. However, when the test target value lies outside of the training data range, the errors increase rapidly above the low-test errors. Nevertheless, the TL model AtomSets-V_0 generalizes better in the extrapolation high-test regime compared to the non-TL AtomSets model, as shown by the lower extrapolation errors in Fig. 4b compared to Fig. 4a. For structural models, the low-test errors are again almost the same. Yet, the TL AtomSets-V_1 models have lower errors than the MEGNet counterparts, see Fig. 4c, d. Similar conclusions can be reached using the shear moduli dataset, as shown in Supplementary Fig. 4. If the prediction regime is further away from the training data regime, the accuracy improvements using the transfer-learned AtomSets models are even bigger, as shown in Supplementary Figs. 5 and 6, where only the bottom 50% are used as training data. These results demonstrate that TL approaches can significantly enhance the models' accuracy in extrapolation tasks critical in new materials discovery.

Effect of changing the parent models

The above results utilize the MEGNet formation energy model as the parent model as it is the dataset for which the largest amount of data is available. To investigate the effects of the pre-trained parent models on the quality of transfer learning features, we have developed AtomSets models with features generated from four

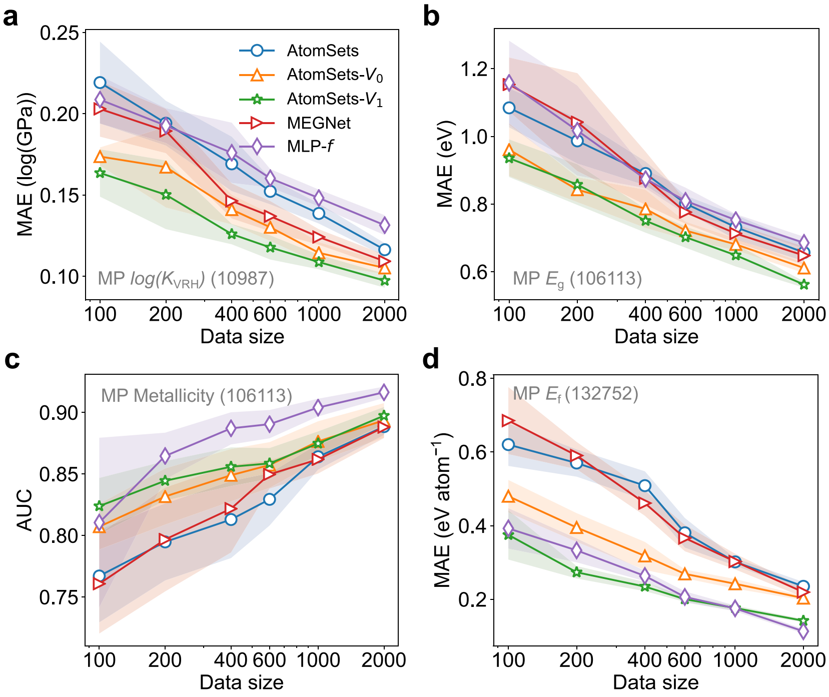

Fig. 3 **Model convergence in the small data limits.** The four datasets are the (**a**) log10 of the bulk moduli, (**b**) band gap, (**c**) binary metallicity, and (**d**) formation energy structural datasets from the Materials Project. The shaded areas are the standard deviation across five random data fitting.

additional MEGNet models fitted using 1,000, 10,000, and 60,000 formation energy data points (MEGNet-E_f-n, where n refers to the number of data points) and 12,179 log (K_{VRH}) data points (MEGNet-log K) from the Materials Project. The results are presented in Supplementary Tables 8–11. The AtomSets-V_0 compositional models using different parent models generally have similar accuracies and are close to the AtomSets compositional models without transfer learning, similar to the conclusions from Table 2.

In contrast, the performance of structural models AtomSets-$V_i(i = 1, 2, 3)$ show a stronger dependence on the parent models. For example, the structural AtomSets-V_1 models utilizing for MEGNet-E_f-1000 and MEGNet-E_f-10,000 encoders have lower accuracies than those using the MEGNet-E_f-60,000 and MEGNet-E_f-133,420 encoders, as shown in Fig. 5. This is especially the case in small datasets such as exfoliation energy (E_{exfo}) and phonon datasets. We also observed consistently better accuracy in the large MP formation energy AtomSets models using E_f parent models with larger data sizes. By changing the parent data from E_f to the log (K_{VRH}), we noticed a substantial accuracy drop for the AtomSets structural models. We believe that this drop is caused by the relatively lower data quality in the log (K_{VRH}) data compared to the formation energy data, and this difference in the data quality translates to the TL feature quality. The elasticity data is computed via the fitting of the stress-strain relationship where the stress calculations in DFT require much higher K-point density and hence challenging to achieve a consistent level of accuracy in a high-throughput fashion. The inaccuracy in stress calculations may amplify errors in the final elasticity results, introducing more significant intrinsic errors in the parent models.

DISCUSSION

The hierarchical MEGNet features provide a cascade of descriptors that capture both short-ranged interactions at early GC (e.g., $\mathbf{V_0}$, $\mathbf{V_1}$) and long-ranged interactions at later GC (e.g., $\mathbf{V_2}$, $\mathbf{V_3}$). The first GC features are better TL features across various tasks, while the latter GC-generated features generally exhibit worse performance. We can explain this part by drawing an analogy to convolutional neural networks (CNN) in facial recognition, where the early feature maps capture generic features such as lines and shapes and the later feature maps form human faces.[28] It is not surprising that if such CNN is transferred to other domains, for example, recognizing general objects beyond faces, the early feature maps may work while the later ones will not. Hence, the AtomSets-V_0 models are best suited for compositional datasets, while the AtomSets-V_1 models are recommended for fitting structure-based datasets.

One surprising result from our studies is the relatively good performance of the compositional models (AtomSets-V_0) on many properties, e.g., the phonon dos and bulk and shear moduli. It would be erroneous to conclude that these properties are not structure-dependent. We believe the main reason for the compositional models' outperformance is that most compositions either do not exhibit polymorphism or have many polymorphs with somewhat similar properties, e.g., the well-known family of SiC polymorphs. These results highlight the importance of generating a diversity of data beyond existing known materials. Existing databases such as the Materials Project typically prioritize computations on known materials, e.g., ICSD crystals. While such a strategy undoubtedly provides the most value to the community for the study of existing materials, the discovery of new materials with extraordinary properties require exploration beyond known materials; additional training data on hypothetical materials is critical for the development of ML models that can extrapolate beyond known materials design spaces. The use of TL, as shown in this work, is nevertheless critical for improving the extrapolability of models.

While the AtomSets framework has shown good model accuracies across the different datasets, the intent of AtomSets was not to be the best performer for all cases. With the current

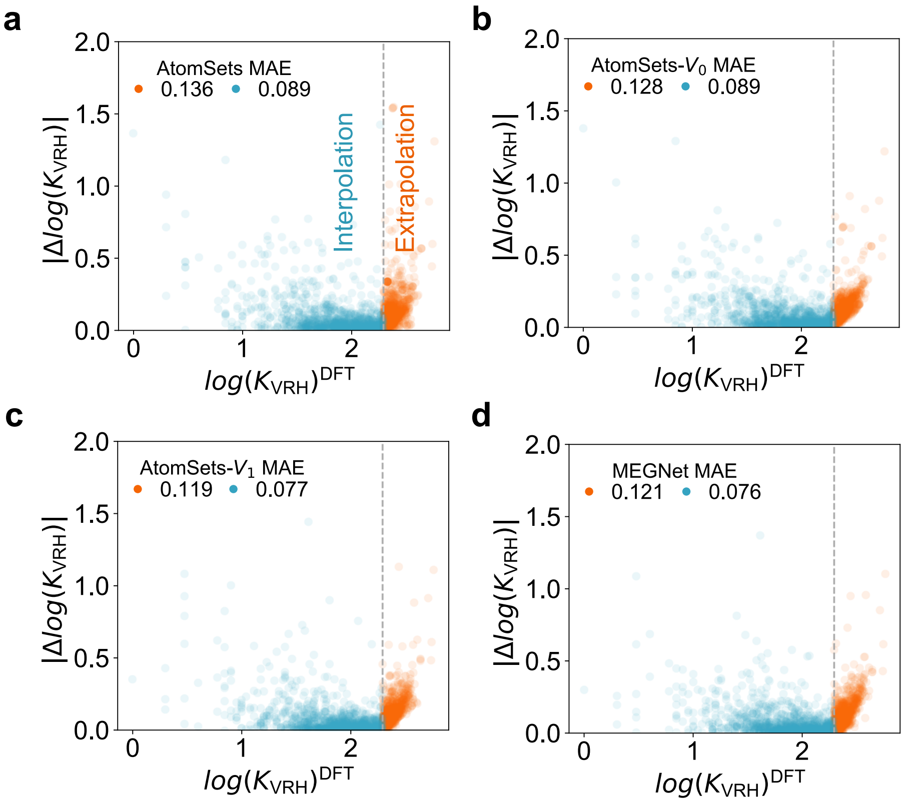

Fig. 4 Absolute differences in predicted and DFT log (K_{VRH}), i.e., |Δlog (K_{VRH})| against the DFT value range for the test data. a–d show the AtomSets, AtomSets-V_0, AtomSets-V_1, and MEGNet model results respectively. The training and validation data are randomly sampled from the 0–90% (vertical dash line) target quantile range. Half of test data comes from the 90–100% quantile (extrapolation) and the other half is from the same target range as the train-validation data (interpolation).

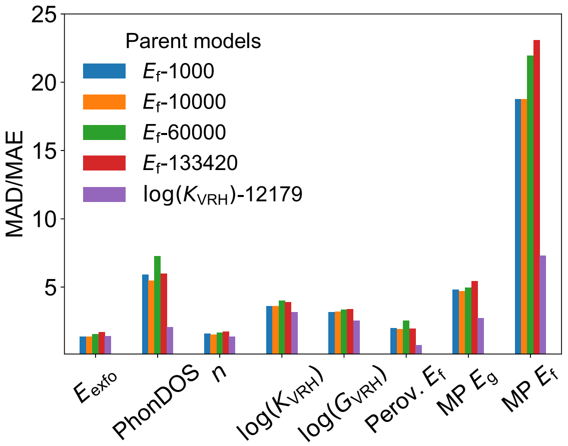

Fig. 5 The effects of changing the parent MEGNet models on the performances of the AtomSets-V_1 models. The *y* axis is the mean-absolute deviation (MAD) divided by the MAE. A higher value corresponds to a better model.

work, we wish to point out that pre-trained graph models can work beyond their original application domains and, for example, as efficient feature generators in other tasks. Even with simple MLP child models, such as the ones in AtomSets models, the final models can already achieve excellent accuracy. For some cases, in particular, in the small data regime, the AutoMatminer model ensembles are on par or slightly better than the AtomSets models. This is somewhat expected given the vast choice of engineered features and the complexity of model ensembles used in the AutoMatminer algorithm. In contrast, the AtomSets rely on a fixed MLP model architecture and deterministic features obtained directly by passing the crystal structure to the pre-trained graph models.

For the comparison with the MEGNet framework, we note that the most time-consuming step in training a graph network model is performing the graph convolutions. Once the graph models are trained, we show that the models can be used as efficient feature generators. Combining with the pre-trained MEGNet model, the AtomSets framework with simple MLPs can already achieve the same accuracy as the expensive MEGNet models and additional applicability in small and compositional datasets. Since there is no parameter update to the MEGNet model, the training speeds are several orders of magnitude faster. For example, it takes about 10 s per epoch and ~360 to ~730 epochs to train the AtomSets-V_1 model on the most extensive MP formation energy data (132,752) using one GTX 1080Ti GPU while training a MEGNet model can take >100 s per epoch and 1582 epochs. The training speed for AtomSets models is also at least one order of magnitude faster than AutoMatminer, as shown in Supplementary Fig. 7.

Recently, several frameworks for ML in materials have been proposed to improve the model prediction accuracies on general materials data. For example, the improved CGCNN model[29] uses Voronoi-tessellation to construct the graph representation and includes explicit three-body interaction, and learnable bond features to the original CGCNN models. Despite its higher accuracy in some structural datasets, it does not solve the small data limitation as seen in MEGNet and CGCNN models. Other frameworks, such as ElemNet[30], IRNet[31], Roost[32] and CrabNet[33] are based purely on compositional information and hence, cannot distinguish polymorphs of materials. While such composition-based models have applications in problems with constrained structural spaces and/or structure-insensitive properties, they do not represent a path to general property predictions across the entire universe of crystal structures and compositions. While a key bottleneck in structure-based models is the requirement for an input structure, these can be mitigated to some extent by employing Bayesian optimization or other similar approaches in combination with a sufficiently accurate energy model, such as the MEGNet formation energy model, to obtain estimated equilibrium crystals.[34]

This work introduces a straightforward deep learning model framework, the AtomSets, as an effective way to learn materials properties at all data sizes and for both compositional and structural data. By combining with TL, the structure-embedded compositional and structural information can be readily incorporated into the model. The simple model architecture makes it possible to train the models with much smaller datasets and lower computational resources than graph models. We show that the AtomSets models can consistently achieve low errors for small data tasks, e.g., steel strength datasets, to extensive data tasks, e.g., MP computational data. The model accuracy further improves with TL. We also show better model convergence for the AtomSets models. The AtomSets framework introduces a facile deep learning framework and helps accelerate the materials discovery process by combining accurate compositional and structural materials models.

METHODS

Graph convolution

During graph convolution, the atom, bond and state features are updated as follows:

$$\mathbf{e}_k^{(i)} = \phi_e\left(\mathbf{e}_k^{(i-1)}, \mathbf{v}_{s_k}^{(i-1)}, \mathbf{v}_{r_k}^{(i-1)}, \mathbf{u}^{(i-1)}\right) \tag{1}$$

$$\mathbf{v}_j^{(i)} = \phi_v\left(\mathbf{v}_j^{(i-1)}, \mathbf{v}_{k\in\mathcal{N}(j)}^{(i-1)}, \mathbf{e}_{l,r_l=j}^{(i)}, \mathbf{u}^{(i-1)}\right) \tag{2}$$

$$\mathbf{u}^{(i)} = \phi_u\left(\frac{1}{N_b}\sum_k \mathbf{e}_k^{(i)}, \frac{1}{N_a}\sum_j \mathbf{v}_j^{(i)}, \mathbf{u}^{(i-1)}\right) \tag{3}$$

where i is an index indicating the layer of the GC, $\mathbf{e}_k^{(i)}$ and $\mathbf{v}_j^{(i)}$ are bond attributes of bond k and atom attributes of atom j at layer i respectively, s_k and r_k are the sending and receiving indices of atoms connecting bond k, ϕs are the update functions approximated using multi-layer perceptrons (MLPs), $\mathcal{N}(j)$ indicates the neighbor atom indices of atom j, and $\mathbf{e}_{l,r_l=j}^{(i)}$ are the bonds connected with atom j, i.e., with receiving atom index r_l as j.

Graph readout function

The readout function aims to reduce the feature matrices with different numbers of atoms to structure-wise vectors subject to permutational invariance. Simple functions to calculate the statistics along the atom number dimension can be used as readout functions. In this work, we tested two types of readout functions. The linear mean readout function averages the feature vectors, as follows

$$\bar{\mathbf{x}} = \frac{\sum_i w_i \mathbf{x}_i}{\sum_i w_i} \tag{4}$$

where $\mathbf{x}_i$ is the feature row vector for atom i and w_i is the corresponding weights. The weights are atom fractions on one site, e.g., $w_{\mathrm{Fe}} = 0.01$ and $w_{\mathrm{Ni}} = 0.99$ in $\mathrm{Fe}_{0.01}\mathrm{Ni}_{0.99}$. We also tested a weight-modified attention-based set2set[35] readout function. We start with memory vectors $\mathbf{m}_i = \mathbf{x}_i\mathbf{W} + \mathbf{b}$, and initialize $\mathbf{q_0}^* = \mathbf{0}$, where $\mathbf{W}$ and $\mathbf{b}$ are learnable weights and biases respectively. At step t, the updates are calculated using long short-term memory (LSTM) and attention mechanisms as follows

$$\mathbf{q_t} = LSTM(\mathbf{q}_{\mathbf{t-1}}^*) \tag{5}$$

$$e_{i,t} = \mathbf{m_i} \cdot \mathbf{q_t} \tag{6}$$

$$a_{i,t} = \frac{w_i \exp(e_{i,t})}{\sum_j w_j \exp(e_{j,t})} \tag{7}$$

$$\mathbf{r_t} = \sum_i a_{i,t}\mathbf{m_i} \tag{8}$$

$$\mathbf{q_{t^*}} = \mathbf{q_t} \oplus \mathbf{r_t} \tag{9}$$

A total of three steps are used in the weighted-set2set readout function. Compared to the simple linear mean readout function, the weighted set2set function has trainable model weights and can describe more flexible and complex relationships between the input and output. The chosen readout functions balance the trade-off between computational speed and model accuracy differently. The linear mean function is faster but potentially less accurate, and the weighed-set2set function is slower but potentially more accurate.

Data and model training

For each model training, we split the data into 80%-10%-10% train-validation-test sets randomly, and the splitting was performed five times with random seeds 0, 1, 2, 3, and 4. The validation set was used to stop the model fitting when the validation metric, i.e., mean-absolute-error (MAE) in regression and area under the curve (AUC) in classification, did not improve for more than 200 consecutive epochs. The model with the lowest validation error was chosen as the best one. Each model was fitted five times using different random splits, and the average and standard deviations of the metric on the test set were reported. In Table 2, to make direct comparisons with existing models, we took the same five-fold shuffle splitting/stratified splitting and the random seed 18012019 from Dunn et al.[6] In this case, each 80% train data is further split into 90%-10% train-validation, and the validation set is used in the same way as in other fittings.

A 5-fold random shuffle split is applied to the data set during the initial hyperparameter screening process, and the parameter set with the lowest average validation error was chosen. The *matbench_expt_gap* (compositional) and *matbench_phonons* (structural) datasets were first used to perform an initial screening for relatively good parameter sets (highlighted in bold in Supplementary Table 2).

DATA AVAILABILITY

The MatBench datasets are available from the AutoMatminer[6] github repository (https://github.com/hackingmaterials/automatminer).

CODE AVAILABILITY

The AtomSets framework and MEGNet featurizations are implemented in the open source materials machine learning (maml) package[36] (https://github.com/materialsvirtuallab/maml).

Received: 3 February 2021; Accepted: 1 October 2021;
Published online: 22 October 2021

REFERENCES

1. Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018).
2. Chen, C. et al. A critical review of machine learning of energy materials. *Adv. Energy Mater.* **10**, 1903242 (2020).
3. Jain, A. et al. Commentary: the materials project: a materials genome approach to accelerating materials innovation. *APL Mater.* **1**, 011002 (2013).
4. Kirklin, S. et al. The Open Quantum Materials Database (OQMD): assessing the accuracy of DFT formation energies. *Npj Comput. Mater.* **1**, 15010 (2015).
5. Curtarolo, S. et al. AFLOWLIB.ORG: a distributed materials properties repository from high-throughput ab initio calculations. *Comput. Mater. Sci.* **58**, 227–235 (2012).
6. Dunn, A., Wang, Q., Ganose, A., Dopp, D. & Jain, A. Benchmarking materials property prediction methods: the Matbench test set and automatminer reference algorithm. *Npj Comput. Mater.* **6**, 1–10 (2020).

7. 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 (2019).
8. Xie, T. & Grossman, J. C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.* **120**, 145301 (2018).
9. Hutchinson, M. L. et al. Overcoming data scarcity with transfer learning. Preprint at https://arxiv.org/abs/1711.05099 (2017).
10. Jha, D. et al. Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning. *Nat. Comm.* **10**, 5316 (2019).
11. Frey, N. C., Akinwande, D., Jariwala, D. & Shenoy, V. B. Machine learning-enabled design of point defects in 2D materials for quantum and neuromorphic information processing. *ACS Nano* **14**, 13406–13417 (2020).
12. Heyd, J., Scuseria, G. E. & Ernzerhof, M. Hybrid functionals based on a screened Coulomb potential. *J. Chem. Phys.* **118**, 8207–8215 (2003).
13. Pilania, G., Gubernatis, J. E. & Lookman, T. Multi-fidelity machine learning models for accurate bandgap predictions of solids. *Comput. Mater. Sci.* **129**, 156–163 (2017).
14. Batra, R., Pilania, G., Uberuaga, B. P. & Ramprasad, R. Multifidelity information fusion with machine learning: a case study of dopant formation energies in Hafnia. *ACS Appl. Mater. Interfaces* **11**, 24906–24918 (2019).
15. Tran, A., Tranchida, J., Wildey, T. & Thompson, A. Multi-fidelity machine-learning with uncertainty quantification and Bayesian optimization for materials design: Application to ternary random alloys. *J. Chem. Phys.* **153**, 074705 (2020).
16. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865–3868 (1996).
17. Gritsenko, O., van Leeuwen, R., van Lenthe, E. & Baerends, E. J. Self-consistent approximation to the Kohn-Sham exchange potential. *Phys. Rev. A* **51**, 1944–1954 (1995).
18. Kuisma, M., Ojanen, J., Enkovaara, J. & Rantala, T. T. Kohn-Sham potential with discontinuity for band gap materials. *Phys. Rev. B* **82**, 115106 (2010).
19. Sun, J., Ruzsinszky, A. & Perdew, J. P. Strongly constrained and appropriately normed semilocal density functional. *Phys. Rev. Lett.* **115**, 036402 (2015).
20. Chen, C., Zuo, Y., Ye, W., Li, X. & Ong, S. P. Learning properties of ordered and disordered materials from multi-fidelity data. *Nat. Comput. Sci.* **1**, 46–53 (2021).
21. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).
22. Zaheer, M. et al. Deep sets. In Guyon, I. et al. (eds.) *Advances in Neural Information Processing Systems* 30, 3391–3401 (Curran Associates, Inc., 2017).
23. Bartel, C. J. et al. A critical examination of compound stability predictions from machine-learned formation energies. *Npj Comput. Mater.* **6**, 97 (2020).
24. Honeycutt, J. D. & Andersen, H. C. Molecular dynamics study of melting and freezing of small Lennard-Jones clusters. *J. Phys. Chem.* **91**, 4950–4963 (1987).
25. De Breuck, P.-P., Hautier, G. & Rignanese, G.-M. Materials property prediction for limited datasets enabled by feature selection and joint learning with MODNet. *Npj Comput. Mater.* **7**, 83 (2021).
26. Meredig, B. et al. Can machine learning identify the next high-temperature superconductor? Examining extrapolation performance for materials discovery. *Mol. Syst. Des. Eng. Eng.* **3**, 819–825 (2018).
27. Xiong, Z. et al. Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation. *Comput. Mater. Sci.* **171**, 109203 (2020).
28. Lee, H., Grosse, R., Ranganath, R. & Ng, A. Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 609–616 (Association for Computing Machinery, New York, NY, USA, 2009).
29. Park, C. W. & Wolverton, C. Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery. *Phys. Rev. Mater.* **4**, 063801 (2020).
30. Jha, D. et al. ElemNet : deep learning the chemistry of materials from only elemental composition. *Sci. Rep.* **8**, 17593 (2018).
31. Jha, D. et al. IRNet: a general purpose deep residual regression framework for materials discovery. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19*, 2385–2393 (Association for Computing Machinery, New York, NY, USA, 2019).
32. Goodall, R. E. A. & Lee, A. A. Predicting materials properties without crystal structure: deep representation learning from stoichiometry. *Nat. Comm.* **11**, 6280 (2020).
33. Wang, A., Kauwe, S., Murdock, R. & Sparks, T. Compositionally-restricted attention-based network for materials property prediction. *Npj Comput. Mater.* **7**, 77 (2021).
34. Zuo, Y. et al. Accelerating materials discovery with Bayesian optimization and graph deep learning. Preprint at https://arxiv.org/abs/2104.10242 (2021).
35. Vinyals, O., Bengio, S. & Kudlur, M. Order matters: sequence to sequence for sets. Preprint at https://arxiv.org/abs/1511.06391 (2016).
36. Chen, C., Zuo, Y., Ye, W., Qi, J., & Ong, S. P. materialsvirtuallab/maml v2021.10.14 https://github.com/materialsvirtuallab/maml (2021).
37. Conduit, G. & Bajaj, S. Citrination. https://citrination.com/datasets/153092 (2017).
38. Choudhary, K., Kalish, I., Beams, R. & Tavazza, F. High-throughput identification and characterization of two-dimensional materials using density functional theory. *Sci. Rep.* **7**, 5179 (2017).
39. Petretto, G. et al. High-throughput density-functional perturbation theory phonons for inorganic materials. *Sci. Data* **5**, 180065 (2018).
40. Zhuo, Y., Mansouri Tehrani, A. & Brgoch, J. Predicting the band gaps of inorganic solids by machine learning. *J. Phys. Chem. Lett.* **9**, 1668–1673 (2018).
41. Petousis, I. et al. High-throughput screening of inorganic compounds for the discovery of novel dielectric and optical materials. *Sci. Data* **4**, 160134 (2017).
42. de Jong, M. et al. Charting the complete elastic properties of inorganic crystalline compounds. *Sci. Data* **2**, 150009 (2015).
43. Castelli, I. E. et al. New cubic perovskites for one- and two-photon water splitting using the computational materials repository. *Energy Environ. Sci.* **5**, 9034–9043 (2012).
44. Ong, S. P. et al. The Materials Application Programming Interface (API): a simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles. *Comput. Mater. Sci.* **97**, 209–215 (2015).
45. Kawazoe, Y., Yu, J.-Z., Tsai, A.-P. & Masumoto, T. (eds.) *Nonequilibrium phase diagrams of ternary amorphous alloys. Condensed Matter* (Springer-Verlag, Berlin Heidelberg, 1997).

ACKNOWLEDGEMENTS

The authors acknowledge the support from the Materials Project, funded by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, Materials Sciences and Engineering Division under contract no. DE-AC02-05-CH11231: Materials Project program, KC23MP. The authors also acknowledge computational resources provided by the Triton Shared Computing Cluster (TSCC) at the University of California, San Diego, and the Extreme Science and Engineering Discovery Environment (XSEDE) supported by the National Science Foundation under grant no. ACI-1053575.

AUTHOR CONTRIBUTIONS

C.C. and S.P.O. conceived the idea. C.C. carried out the model construction, and fitting under the supervision of S.P.O. C.C. and S.P.O. wrote the manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41524-021-00639-w.

Correspondence and requests for materials should be addressed to Shyue Ping Ong.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.