Benchmarking graph neural networks for materials chemistry

Victor Fung,[1]* Jiaxin Zhang,[2] Eric Juarez,[1] Bobby G. Sumpter[1]

[1]Center for Nanophase Materials Sciences, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, United States

[2]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831, United States

**E-mail: fungv@ornl.gov Tel.: +1-951-384-5242*

Abstract

Graph neural networks (GNNs) have received intense interest as a rapidly expanding class of machine learning models remarkably well-suited for materials applications. To date, a number of successful GNNs have been proposed and demonstrated for systems ranging from crystal stability to electronic property prediction and to surface chemistry and heterogeneous catalysis. However, a consistent benchmark of these models remains lacking, hindering the development and consistent evaluation of new models in the materials field. Here, we present a workflow and testing platform, MatDeepLearn, for quickly and reproducibly assessing and comparing GNNs and other machine learning models. We use this platform to optimize and evaluate a selection of top performing GNNs on several representative datasets in computational materials chemistry. From our investigations we note the importance of hyperparameter selection and find roughly similar performances for the top models once optimized. We identify several strengths in GNNs over conventional models in cases with compositionally diverse datasets and in its overall flexibility with respect to inputs, due to learned rather than defined representations. Meanwhile several weaknesses of GNNs are also observed including high data requirements, and suggestions for further improvement for applications in materials chemistry are proposed.

1. Introduction

In the search for materials for various functional applications ranging from catalysis to energy storage to electronics, machine learning (ML) has quickly gained traction as a powerful and flexible approach, especially where a broad exploration of the materials space is needed.[1-6] The adoption of ML for materials discovery is expected to expand even further with the ongoing growth in the availability of high-throughput density functional theory (DFT) datasets and continued advancements in ML algorithms.[7-14] Conventionally, ML models in materials chemistry are descriptor-based, where the key descriptors representing the system must first be defined prior to fitting a suitable ML model for prediction. General examples of these descriptors include stoichiometry, the elemental properties such as group, period, electronegativity and radius, and electronic properties such as partial charges and s, p, d-band positions. A number of structural descriptors have also been proposed satisfying translation and rotational invariance, including but not limited to the Coulomb matrix[15], atom-centered symmetry functions (ACSFs),[16] and smooth overlap of atomic positions (SOAP).[17] In most cases, the descriptors are carefully selected for a particular problem, and require a significant amount of domain expertise and investment of human effort to identify and compile. This can also prove challenging for problems with a large amount of compositionally and structurally diverse materials for which a single set of applicable descriptors is difficult to find.

In recent years, graph neural networks (GNNs)[18-20] have received increasing attention as a method which could potentially overcome the limitations of fixed descriptors by learning the representations on flexible, graph-based inputs. Within this overarching class of ML method, a number of GNN models have been proposed for chemistry related problems, with the earliest adopters focusing on molecular systems.[21-24] Subsequently, GNNs have also been used in materials prediction, with a number of studies tackling systems such as periodic crystals[25-30] and surfaces.[11,31-33] These systems are generally described by their atomic structures, where the atoms can be represented by nodes and the neighbors encoded by the edges. Information regarding the atoms and bonds such as element type and bond distances, respectively, can be further encoded in the node and edge attributes. GNNs operate on these atom-based graphs to create node-level embeddings through convolutions with neighboring nodes and edges. This learnable process is then trained for the prediction of a given property of the material under study.

Given the rapid advances in GNNs for computational materials chemistry currently, a critical evaluation of the current state-of-the-art (SOTA) is warranted. To accomplish this, several criteria should be met: (1) the same datasets should be used across the evaluated models, (2) the datasets used should represent diverse problems in materials chemistry, (3) the same input information and representation should be used, (4) the hyperparameters of the models should be optimized to the same extent, and (5) these should be performed in a reproducible manner. In this work we attempt to address these criteria and provide an open-source framework, MatDeepLearn, which can be used for further studies in this area (Figure 1).[34] Provided with atomic structures and target properties from any dataset of choice, MatDeepLearn handles the processing of structures to graphs, offers a library of GNN models, and provides hyperparameter optimization for the models. Within this framework, improvements and additions to the input representations and model architectures can be easily made, which can greatly reduce development time needed for new ML models. GNNs can also be critically evaluated and actionable information can be quickly obtained when applied to specific problems in chemistry and materials sciences. We then use this framework to benchmark several SOTA models and provide a timely snapshot of current progress and suggestions for future progress.

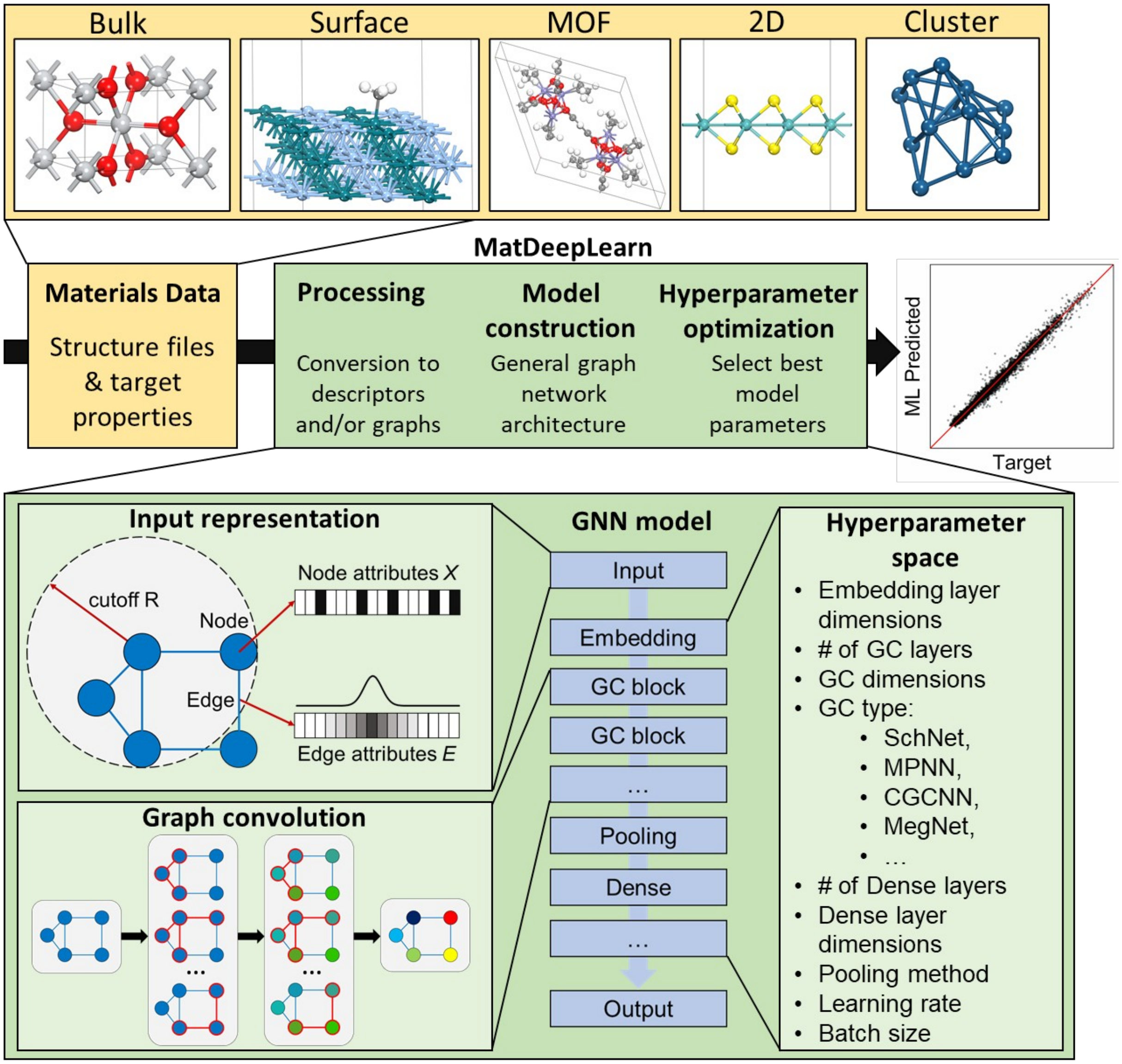

Figure 1. Scheme of GNN testing framework and workflow. A general outline of the approach is presented, starting with materials data (with several examples shown) in the form of structure files and target properties, followed by data processing, model construction and hyperparameter optimization as part of the MatDeepLearn framework. A general graph neural network architecture is constructed, taking in graphs containing nodes, edges, node attributes and edge attributes, inputted into an embedding layer, GC blocks, pooling and dense layers. This allows models to be similarly compared within a shared hyperparameter space.

2. Methods

2.1 Datasets

The datasets we used were each chosen to reasonably represent a variety of different classes of materials ranging from 3D to 0D, each with different target properties, summarized in Table 1. All five datasets were generated through DFT calculations. Four of the datasets were obtained from curated, open computational databases.

Table 1: Dataset information

	Datasets				
	Bulk crystals	Alloy surfaces	MOFs	2D materials	Pt clusters
Target Property	Formation energy (eV/atom)	Adsorption energy (eV)	Band gap (eV)	Work function (eV)	Total energy (eV)
Number of Data	Approx. 37000	Approx. 37000	Approx. 13000	Approx. 4000	Approx. 20000
Material size range (atoms)	1 to 200	13 to 16	17 to 150	2 to 12	10 to 13
Material composition range (elements)	87	42	78	60	1
Calculation method	DFT: PBE+U	DFT: BEEF-vdw	DFT: PBE	DFT: PBE+U	DFT: PBE
Source	Materials project (ref[12])	CatHub (ref[35])	QMOF (ref[36])	C2DB (ref[37])	Literature (ref[38])

2.1.1 Bulk dataset

For bulk materials, we compiled approximately 37,000 structures from the Materials Project,[12] which contains the widest elemental diversity of the datasets in this work, as well as the greatest diversity in structure size from the number of atoms. Bulk crystals have been used as a de facto benchmark system in the computational materials literature, with many successful examples of regression and classification using for both GNN models.[25-30] The property selected for this dataset is formation energy in units of eV/atoms. We note this database is constantly expanding, and here we used a snapshot of a subset of the available data.

2.1.2 Surface dataset

For surfaces, we employed a dataset of another roughly 37,000 structures from the Catalysis-Hub database, containing eleven adsorbates on approximately 2,000 unique metal alloy surfaces.[35] For this dataset, we used the relaxed surfaces containing the adsorbate as the input structures and adsorption energy in eV as the target property. Unrelaxed surfaces are not used in this study due to the ambiguities involved in the placement of the adsorbate which will not be explored in detail here. With this dataset we intend to evaluate the general ability of the model for predicting surface chemistry properties, which are relevant for catalysis.

2.1.3 MOF dataset

For porous materials, we used a metal-organic framework (MOF) dataset from the QMOF database containing 13,000 processed MOF structures and containing band gaps in eV as the target property.[36] Here, the structures used are unrelaxed rather than relaxed, while the band gaps are computed from the relaxed structures. This task involves the prediction of an electronic property as well as an unrelaxed structure to relaxed property prediction, which can be considerably harder.

2.1.4 2D dataset

For 2D materials, we used a dataset from C2DB, containing around 4,000 structures, with the target property being work function in eV, another important electronic property.[37] This dataset is smaller than the other examples in this work while still being compositionally diverse with 60 elements included, and may prove challenging for GNNs with their generally high data requirements.

2.1.5 Cluster dataset

The last dataset is compositionally narrow with only one element, Pt, but is structurally diverse with around 2,0000 different nanoclusters ranging from 10 to 14 atoms, and with total energies in eV as the target. These clusters were obtained from basin-hopping global optimization in a previous study.[38] This dataset was included to evaluate the ability of GNNs to capture structural sensitivity for potential use in machine-learned potentials, a task for which descriptor-based methods are currently most commonly used.

2.2 Structure/Graph Representations

The data are organized as a set of structures and a set of target properties associated with the structures. The structures are described by a set of atomic positions in Cartesian coordinates and an accompanying lattice vector describing the dimensions of the periodic cell. Non-periodic structures can also be represented by simply placing the molecule/cluster in a sufficiently large empty cell and ensuring the distance between images are larger than distance cutoffs for edges. Each atom is represented by a node in the graph, and the edges are usually determined by interatomic distances within a certain radius cutoff. The node attributes in this work contain elemental properties of the atoms which are one-hot encoded as described by Xie et al.[27] Edge attributes encode the interatomic distances, described later in the section. Global graph-wide attributes are not used in this work and left empty.

2.3 Machine learning

The general architecture of overall GNN models used so far in materials chemistry contains several shared characteristics, which we unify into a general architecture here, illustrated in Figure 1. First, an embedding or preprocessing layer is present which transforms the node attributes from the input to a specified dimension. This is followed by N number of graph convolutional (GC) blocks, which perform the convolution and aggregation of the nodes. We used the same graph convolutional operators used in the original GNN models. This is followed by a graph-wide readout/pooling layer which provides an overall graph representation by aggregating the node attributes; in this work we choose from max, average, sum and set2set pooling. This is followed by M dense layers and the scalar output. The dimensions of the embedding, GC, and dense blocks, the type of pooling layer used, layer counts N and M, batch size and learning rate are hyperparameters which are selected through optimization for each model and each dataset. For each model and dataset, hyperparameter optimization was performed for 160 trials using the HyperOpt optimizer and the model with the lowest test error was selected.[39] We use the Ray library which provides distributed hyperparameter optimization on multiple nodes.[40] The reported performance is then obtained from a five-fold cross validation on the selected model. The code was written in Python 3.7, and uses PyTorch v1.6 and PyTorch-Geometric[41] v1.6 libraries for the ML models.[34]

We tested five different GC operators here:

1. SchNet:[22]

$$x_i^{'}=\sum_{j\in N(i)} x_j \odot h_\Theta\left(\exp\left(-\gamma\left(d_{i,j}-\mu\right)^2\right)\right)$$

Here, $d_{i,j}$ is defined as the interatomic distance between atoms i and j, and h_Θ is a neural network containing dense layers which generates filters from interatomic distances. Before being fed to the neural network, the distances are expanded by a Gaussian basis function, which provides a continuous, non-sparse representation. This was also later successfully applied for use in other GNNs such as CGCNN and MEGNet. In the rest of this work we define this term as $e_{i,j}=\exp\left(-\gamma\left(d_{i,j}-\mu\right)^2\right)$ and use these as the edge attributes.

Additionally, update functions are applied to the node attributes in the form of dense layers. In the original SchNet model, atom features are transformed to scalar atom-wise values before being summed over the whole graph. This is common practice for machine-learned potentials which first calculate the atom-wise energies before sum pooling to obtain the total energy.[42,43] Instead, in this work we pool first to obtain the graph-level features before predicting the scalar property using dense layers; this is consistent with other approaches for materials property predictions such as MEGNet and CGCNN.

2. MPNN:[21]

$$x_i^{'}=\Theta x_i+\sum_{j\in N(i)} x_j \odot h_\Theta(e¿¿i,j)¿$$

h_Θ is a neural network containing dense layers, and update functions are also applied to the nodes, this time in the form of a gated recurrent unit.

3. CGCNN:[27]

$$x_i^{'}=x_i+\sum_{j\in N(i)} \sigma\left(z_{i,j}W_f+b_f\right)\odot g\left(z_{i,j}W_s+b_s\right)$$

Here, $z_{i,j} = x_i \oplus x_j \oplus e_{i,j}$, and σ and g are sigmoid and softplus functions respectively.

4. MEGNet:[26]

$$e'_{i,j} = h_{\Theta e}\left(x_i \oplus x_j \oplus e_{i,j}\right)$$

$$x'_i = h_{\Theta v}\left(\left(\frac{1}{N(i)} \sum_{j \in N(i)} e_{i,j}\right) \oplus x_i\right)$$

Two dense layers are added in the beginning of each MEGNet GC block to preprocess the inputs. Here $h_{\Theta e}$ and $h_{\Theta v}$ are edge and node update functions, which are also dense layers. The updates follow the order of edges, nodes and global attributes. A skip connection adds the unprocessed input attributes of each block with the output attributes. In the original work the global attributes were left blank for the inorganic crystals dataset and are similarly unused here.

5. GCN:[44]

$$x'_i = \Theta \sum_j \frac{1}{\sqrt{\hat{d}_i \hat{d}_j}} x_j$$

We include this as a baseline GC with a much simpler construction and not specifically developed for materials chemistry applications. In contrast to the earlier models, a purely linear update function Θ is used here for the node attributes, and edge attributes are not included. Instead, the edge weights are used, containing the inverse normalized atomic distances.

In addition, we tested two non-GNN models, using the Sine matrix[45] (SM) and the Smooth Overlap of Atomic Positions[17] (SOAP) descriptors, for comparison. For the SM descriptor, the size of the matrix is padded with zeros to the maximum atomic size of the dataset, and only the eigenvalues are used, sorted in descending order of their absolute values. For the SOAP descriptor, Gaussian type orbital basis functions are used and an “inner” average is obtained for each element present in the dataset, whereby the average is taken over the sites before summing up the magnetic quantum numbers. The SOAP parameters, the distance cutoff, the number of radial basis functions, the degree of spherical harmonics and the standard deviation of the Gaussians in the basis functions are all considered as hyperparameters to be optimized. For both models the descriptors are fed into dense layers with variable layer count and size as determined from hyperparameter optimization. The DScribe library was used to obtain SM and SOAP descriptors.[46] Finally, an overall baseline is provided for comparison in the form of a dummy regressor which only returns the mean of the training dataset.

3. Results

Table 2: Benchmarking results - models

	MAE error

	Datasets				
ML models	Bulk crystals (eV/Atom)	Alloy surfaces (eV)	MOFs (eV)	2D materials (eV)	Pt clusters (eV)
SchNet	0.050	0.063	0.228	0.214	0.151
MPNN	0.046	0.058	0.245	0.204	0.182
CGCNN	0.049	0.060	0.233	0.208	0.205
MEGNet	0.048	0.069	0.253	0.224	0.180
GCN	0.067	0.175	0.355	0.304	0.577
SOAP	0.047	0.118	0.318	0.203	0.143
SM	0.394	0.621	0.608	0.607	0.460
Baseline	0.978	1.480	0.984	0.773	4.984

We evaluated a total of seven ML models for regression tasks on the five datasets, summarized in Table 2, with the specific hyperparameters for each top model listed in Table S1, with GNN entries shaded in grey. We find all four SOTA models (SchNet, MPNN, CGCNN, MEGNet) performed very well for all tested datasets, once again demonstrating the capability of GNNs for accurate predictions when ample training data is available. Unexpectedly, we find the SOTA models all performed equally well in most cases, which shows hyperparameter optimization can be just as important as the choice of graph convolutional operator and overall machine learning architecture. The SOTA models also offer advantages over simpler GNNs which were not designed for materials chemistry, such as the GCN model, which performed markedly worse for nearly all datasets except the bulk crystals. The poor GCN performance, especially for Pt clusters, suggests convolutions involving edge attributes containing interatomic distances in the Gaussian basis, which the SOTA models share, are much more effective for capturing the spatial information of atomic structures.

Meanwhile, for non-GNN models, SM performed poorly across the datasets, but SOAP performed surprisingly well for bulk crystals and 2D materials and had a performance similar to or better than the SOTA models for Pt clusters. On one hand, this suggests the SOAP descriptors, like other similar descriptors such as ACSFs, remain excellent choices for capturing spatial information in atomic structures, particularly for use in machine-learned potentials. On the other hand, this nevertheless highlights another strength in GNN-based models in that a similar level of performance can still be achieved without any pre-defined descriptors.

Relative to the Baseline, the SOTA GNNs had the highest relative errors in predicting work functions for the 2D dataset, and this is likely due to the much smaller size of the dataset compared to the other datasets. The GNN models performed second worst for the MOFs; while this dataset is larger than the 2D dataset, the prediction task is also more challenging as only the unrelaxed structures are provided to predict the properties of the relaxed structures. Next, the GNNs performed similarly well for bulk and surfaces, with both containing ample amounts of training data for training. Finally, the GNNs performed best for the clusters relative to the Baseline; however, these errors, at approximately 0.015 eV/atom, remain relatively high for use as machine learned potentials. A more thorough evaluation would be needed in the future for

GNN performance in ML potentials, which may require more significant changes in the model architecture and training data and the inclusion of forces.

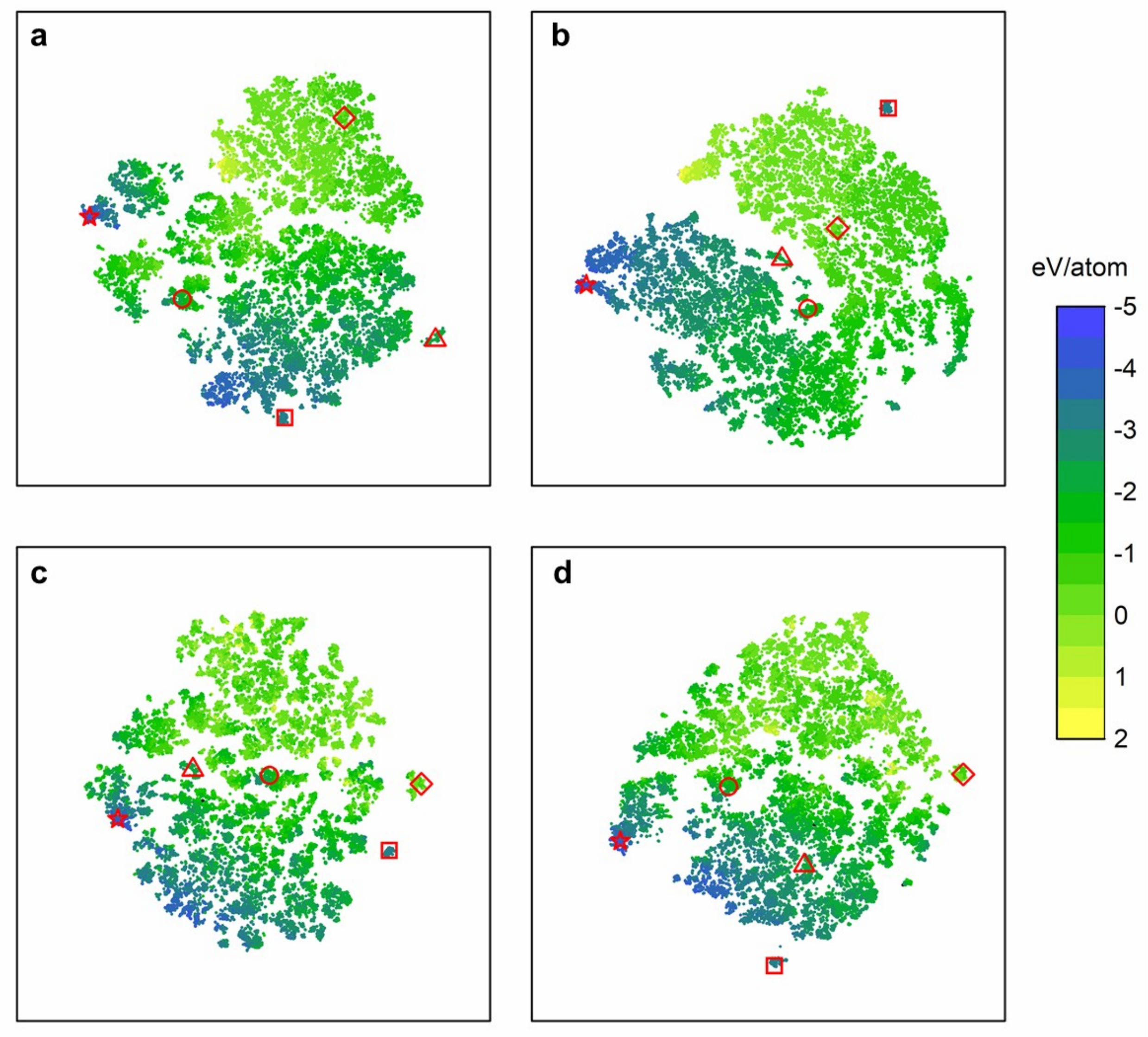

Figure 2. Visualization of graph-wide feature space. t-SNE plot of the graph-level embedding from the readout/pooling layer for (a) CGCNN, (b) MPNN, (c) SchNet, (d) MEGNet trained on the bulk dataset, with each point representing an individual crystal. Colors for each point are mapped to formation energies. Selected structures are marked by red shapes, with the same shape corresponding to the same crystal in each plot.

We then compare the ability of the different GNNs in learning the structure and composition representations by obtaining the graph-level embedding from the output of the readout/pooling layer and visualizing with t-distributed stochastic neighbor embedding (t-SNE) in Figure 2, using the Bulk dataset as an example. The plots in Figure 2 represent a combined structure-composition latent space for the trained materials where points within a grouping can be expected to share similarities in both their atomic structures and elemental compositions. Each of the four SOTA GNNs are able to generate viable representations leading to groupings of crystals delineated by similar formation energies. In some cases, each GNN obtained similar grouping in the latent space, such as for structures labeled with the star or square symbols. In other cases, the GNNs obtained dissimilar grouping, for example with the ones denoted by the diamond, and to a lesser extent the triangle and circle symbols. Ultimately, for purposes of regression each of the plotted latent spaces are equally valid and provide similar prediction errors; though additional investigation may be needed in the future to reveal additional differences for applications such as generative machine learning.

Next, we investigated the training size dependence of the models, another avenue of comparison between models and datasets, and estimate the maximum performance with respect to data size in Figure 3. The model performance is obtained from five parallel runs for each training size using a different train/test split and averaging the errors. We find, in almost all situations, the training size dependence to be very similar between the GNN models for a particular dataset, approximately irrespective of the number of parameters in the model. Unsurprisingly, SOAP has a better training size scaling for the cluster dataset, likely thanks to the effectiveness of the pre-defined descriptors. However, when applied to compositionally diverse systems, SOAP loses this advantage over GNNs and has a similar training size scaling with the other models. We also estimate performance on data sizes by extrapolating training size dependence curves using a power law function, $\varepsilon(m)= \alpha m^{\beta}$ where ε is the error as a function of samples, m.[47] We extrapolate to five times the current data set size for each case. For the bulk datasets, fitting the power law function gives exponents of ~-0.3 for the GNNs and extrapolating predicts MAEs of ~0.02-0.03 for 5x data at ~150,000 training data. For surfaces, power law fitting suggests a better scaling than bulk with exponents of ~-0.5 and provides an MAE estimate of ~0.03-0.04 eV. Meanwhile, a recent study reached an opposite conclusion, finding worse scaling for surface adsorption compared to the bulk.[11] This is likely due to unrelaxed structures being used as inputs, which significantly increases the dimensionality and difficulty of the task. Indeed, when fitting the power law curve for the MOF task which likewise uses unrelaxed structures, the exponents are much lower at -0.2, consistent with our hypothesis. There, MAE estimates for 5x data remains high at around ~-0.17 eV. The differing training size scaling from using the relaxed and unrelaxed structures has important implications for reducing data requirements for training and presents a promising avenue of improvement in the future.

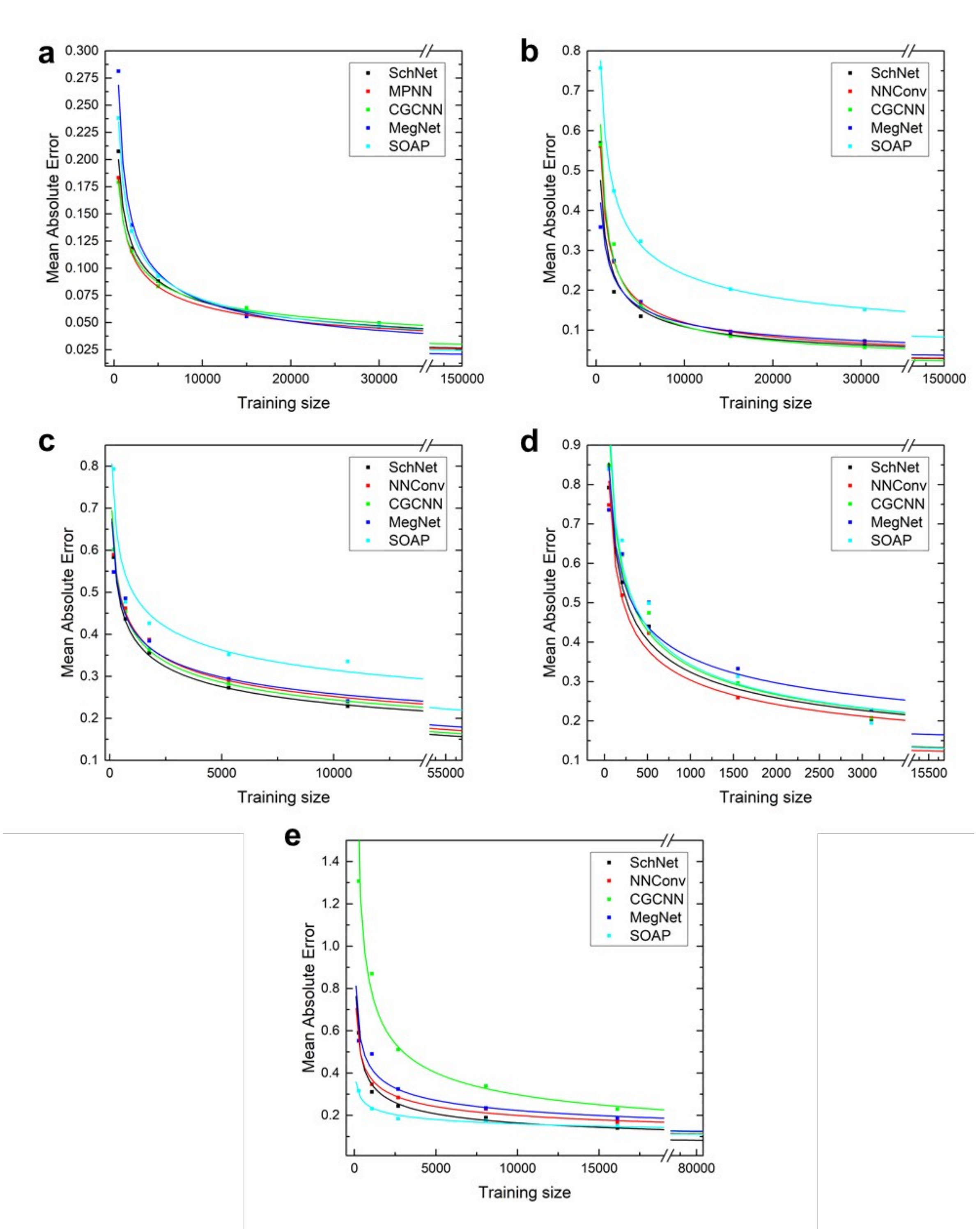

Figure 3. Training size dependence and extrapolation. The training size dependence for (a) bulk, (b) surface, (c) MOF, (d) 2D, (e) and cluster datasets are plotted. Each point represents the average of five separate runs using different train/test splits. Solid lines are obtained from fitting on the power law.

So far, we have focused primarily on model performance and maintained a consistent representation for all models and datasets. We examine the soundness of this methodology by testing several different representations and observing their impact on the performance (Table 3). Here we use the optimal hyperparameters in Table S1, and test using the CGCNN model. Possible avenues for modifying the representations include changing the edge cutoffs and selection criteria for generating the graphs and changing the properties of the node and edge attributes. First, we find reducing the number of considered neighbors from 12 to 4 did not significantly change prediction accuracy for most cases besides the cluster dataset where the performance decreases significantly. Meanwhile, a fully connected graph did not improve prediction accuracy beyond the default, while greatly increasing the computational cost. Moving to node attributes, we found using node features based simply on element identity with one-hot encoding was also sufficient, and including additional elemental properties such as electronegativity and radius were generally unnecessary for sufficiently large data sets. We also tested using blank input node attributes (with all zeroes), thereby removing any information regarding the elemental composition; this significantly reduced the performance for all cases except the cluster dataset (which only has one element in the system). Finally, a test was performed where the length of edge attributes was reduced from 50 to 10, thereby greatly reducing the structural resolution encoded by the Gaussian basis. This had little to no impact on 4 out of 5 datasets, but greatly reduced the prediction ability for clusters as the information needed to distinguish small changes in Pt-Pt bond distances is lost. Thus, we find the default representation used in this work to be satisfactory within the scope of including only structure and atomic information as inputs. Variations in the representation, up to an extent, did not appreciably affect performance likely due to the property of GNNs in producing learned representations via training.

Table 3: Benchmarking results - representations

	MAE error				
	Datasets				
Input representation	Bulk crystals (eV/Atom)	Alloy surfaces (eV)	MOFs (eV)	2D materials (eV)	Pt clusters (eV)
Default	0.049	0.060	0.233	0.208	0.205
Neighbors: 4	0.054	0.082	0.238	0.195	0.410
Neighbors: full	0.047	0.060	0.261	0.198	0.203
Node features: Element One-hot	0.067	0.058	0.232	0.217	0.184
Node features: Blank	0.319	0.570	0.410	0.562	0.187
Edge length: 10	0.050	0.059	0.257	0.218	1.328

4. Discussion and conclusion

With the rapid expansion of readily accessible high-throughput DFT datasets and new GNN models which can train on them, there is a strong demand for a general benchmarking tool to assess which ML models are best for a given target application in materials chemistry. We have developed such a framework, MatDeepLearn and used it to gauge the performance of several GNNs on different materials data sets. We use the Pytorch and Pytorch-Geometric libraries which allows for fast ML calculations which are optimized for GPU-based computing resources, and the Ray library which provides distributed hyperparameter optimization on multiple nodes. In this work, training time ranges from ~10 minutes to ~1-2 hours on an NVIDIA Tesla V100 GPU for the smallest and largest datasets, respectively. Hyperparameter training with 160 trials can be finished in roughly 2 days on a single GPU node containing 8 V100 GPUs. In this work, five GNN models were tested, but this list can be rapidly expanded with the provided message passing network class in Pytorch-Geometric, and with approximately 40 existing methods already implemented for use. Consequently, the development time needed for GNNs can be shortened significantly, along with the rapid testing and benchmarking of these developed methods. In general, we find MatDeepLearn can provide a highly competitive baseline performance with little to no human interaction or effort required; only a suitable dataset containing atomic structures are needed as the inputs.

Our current study also provides some general observations regarding GNNs for materials applications. For the prediction tasks in this work, a GNN which contains nonlinear update functions for nodes (usually neural networks), and a sufficiently descriptive representation of bond distances generally perform quite well, and differences between the current tested GC operators become small through hyperparameter optimization. Within the scope of our study, including edge updates did not appear to improve the performance perceptibly. In moving forward, a more exhaustive screening of the GNN “design space” may be fruitful, such as those proposed in a recent study by You et al.[48]

Training the GNNs can be data-intensive, and we find a minimum of 10^3-10^4 data points are needed to achieve an adequate accuracy in the models. This general sentiment is echoed in a similar study finding descriptor-based models performing better than GNNs with small data sizes, but with GNNs performing better when ample data is available.[25] For applications with very small datasets, pre-defined descriptors still work best, but will quickly fail when moving outside of its domain of applicability. We also found a large difference in training size scaling between using relaxed and unrelaxed structures as inputs; methods which can improve the quality of the initial guess structures could help reduce the training costs significantly.

Approaches to effectively incorporate domain knowledge with GNN models which complements their existing flexibility and ability for learnable representations would also likely improve performance. For example, additional information about the system can be incorporated into the node or edge attributes, such as the bond types, aromaticity and chirality in the case of molecular systems, which is not inherently known from just the atomic structure. Along these lines, recent approaches incorporating some form of features relating to atomic orbitals and their interactions have also yielded promising results.[30,49] Alternatively, the inclusion of physical constraints in the loss functions or through other means which have been used for ML in general could also be considered for GNNs.

Additionally, effectively incorporating knowledge from pre-existing datasets through transfer learning is also another promising avenue for improvement by leveraging current high-throughput computational databases. Besides the datasets used in this study, many other computational databases with 10^4 or more data entries are also available which would be well-suited for training with GNNs.[7-11] GNN methods which can train on multifidelity data have been demonstrated recently which would improve integration with multiple datasets.[50]

Acknowledgements

This work was supported by the Center for Understanding and Control of Acid Gas-Induced Evolution of Materials for Energy (UNCAGE-ME), an Energy Frontier Research Center funded by U.S. Department of Energy, Office of Science, Basic Energy Sciences. Work was performed at the Center for Nanophase Materials Sciences, which is a US Department of Energy Office of Science User Facility. VF was also supported by a Eugene P. Wigner Fellowship at Oak Ridge National Laboratory. JZ was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program; and by the Artificial Intelligence Initiative at the Oak Ridge National Laboratory (ORNL). ORNL is operated by UT-Battelle, LLC., for the U.S. Department of Energy under Contract DEAC05-00OR22725. This research used resources of the National Energy Research Scientific Computing Center, supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Competing Interests

The authors declare no competing interests.

Additional Information

Supplementary information is available.

Data Availability

The machine learning code, MatDeepLearn, used in this work is available at https://github.com/vxfung/MatDeepLearn. The datasets used for testing are also provided in their original form.

References

(1) Schmidt, J.; Marques, M. R. G.; Botti, S.; Marques, M. A. L. Recent advances and applications of machine learning in solid-state materials science. *Npj Comput. Mater.* **2019**, *5*, 83.

(2) Alberi, K.; Nardelli, M. B.; Zakutayev, A.; Mitas, L.; Curtarolo, S.; Jain, A.; Fornari, M.; Marzari, N.; Takeuchi, I.; Green, M. L.; Kanatzidis, M.; Toney, M. F.; Butenko, S.; Meredig, B.; Lany, S.; Kattner, U.; Davydov, A.; Toberer, E. S.; Stevanovic, V.; Walsh, A.; Park, N.-G.; Aspuru-Guzik, A.; Tabor, D. P.; Nelson, J.; Murphy, J.; Setlur, A.; Gregoire, J.; Li, H.; Xiao, R.; Ludwig, A.; Martin, L. W.; Rappe, A. M.; Wei, S.-H.; Perkins, J. The 2019 materials by design roadmap. *J. Phys. D: Appl. Phys.* **2018**, *52*, 013001.

(3) Schlexer Lamoureux, P.; Winther, K. T.; Garrido Torres, J. A.; Streibel, V.; Zhao, M.; Bajdich, M.; Abild-Pedersen, F.; Bligaard, T. Machine Learning for Computational Heterogeneous Catalysis. *ChemCatChem* **2019**, *0*.

(4) Chen, C.; Zuo, Y.; Ye, W.; Li, X.; Deng, Z.; Ong, S. P. A Critical Review of Machine Learning of Energy Materials. *Advanced Energy Materials* **2020**, *10*, 1903242.

(5) Batra, R.; Song, L.; Ramprasad, R. Emerging materials intelligence ecosystems propelled by machine learning. *Nature Reviews Materials* **2020**.

(6) Schleder, G. R.; Padilha, A. C. M.; Acosta, C. M.; Costa, M.; Fazzio, A. From DFT to Machine Learning: Recent Approaches to Materials Science–a Review. *Journal of Physics: Materials* **2019**, *2*, 032001.

(7) Saal, J. E.; Kirklin, S.; Aykol, M.; Meredig, B.; Wolverton, C. Materials Design and Discovery with High-Throughput Density Functional Theory: The Open Quantum Materials Database (OQMD). *JOM* **2013**, *65*, 1501-1509.

(8) Choudhary, K.; Garrity, K. F.; Reid, A. C. E.; DeCost, B.; Biacchi, A. J.; Hight Walker, A. R.; Trautt, Z.; Hattrick-Simpers, J.; Kusne, A. G.; Centrone, A.; Davydov, A.; Jiang, J.; Pachter, R.; Cheon, G.; Reed, E.; Agrawal, A.; Qian, X.; Sharma, V.; Zhuang, H.; Kalinin, S. V.; Sumpter, B. G.; Pilania, G.; Acar, P.; Mandal, S.; Haule, K.; Vanderbilt, D.; Rabe, K.; Tavazza, F. The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design. *Npj Comput. Mater.* **2020**, *6*, 173.

(9) Curtarolo, S.; Setyawan, W.; Wang, S.; Xue, J.; Yang, K.; Taylor, R. H.; Nelson, L. J.; Hart, G. L. W.; Sanvito, S.; Buongiorno-Nardelli, M.; Mingo, N.; Levy, O. AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations. *Comput. Mater. Sci.* **2012**, *58*, 227-235.

(10) Draxl, C.; Scheffler, M. The NOMAD laboratory: from data sharing to artificial intelligence. *Journal of Physics: Materials* **2019**, *2*, 036001.

(11) Chanussot, L.; Das, A.; Goyal, S.; Lavril, T.; Shuaibi, M.; Riviere, M.; Tran, K.; Heras-Domingo, J.; Ho, C.; Hu, W.; Palizhati, A.; Sriram, A.; Wood, B.; Yoon, J.; Parikh, D.; Zitnick, C. L.; Ulissi, Z. The Open Catalyst 2020 (OC20) Dataset and Community Challenges. *arXiv e-prints* **2020**, arXiv:2010.09990.

(12) Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; Persson, K. A. Commentary: The Materials Project: A Materials Genome Approach to Accelerating Materials Innovation. *APL Materials* **2013**, *1*, 011002.

(13) Clement, C. L.; Kauwe, S. K.; Sparks, T. D. Benchmark AFLOW Data Sets for Machine Learning. *Integrating Materials and Manufacturing Innovation* **2020**, *9*, 153-156.

(14) Wang, A. Y.-T.; Murdock, R. J.; Kauwe, S. K.; Oliynyk, A. O.; Gurlo, A.; Brgoch, J.; Persson, K. A.; Sparks, T. D. Machine Learning for Materials Scientists: An Introductory Guide toward Best Practices. *Chem. Mater.* **2020**, *32*, 4954-4965.

(15) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.* **2012**, *108*, 058301.

(16) Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **2011**, *134*, 074106.

(17) De, S.; Bartók, A. P.; Csányi, G.; Ceriotti, M. Comparing molecules and solids across structural and alchemical space. *Phys. Chem. Chem. Phys.* **2016**, *18*, 13754-13769.

(18) Zhou, J.; Cui, G.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. J. a. p. a. Graph neural networks: A review of methods and applications. **2018**.

(19) Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S. Y. J. I. T. o. N. N.; Systems, L. A comprehensive survey on graph neural networks. **2020**.

(20) Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261* **2018**.

(21) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212* **2017**.

(22) Schütt, K.; Kindermans, P.-J.; Sauceda Felix, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *NeurIPS* **2017**, *30*, 991-1001.

(23) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. In *Tilte*2015.

(24) Wu, Z.; Ramsundar, B.; Feinberg, Evan N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science* **2018**, *9*, 513-530.

(25) 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.* **2020**, *6*, 138.

(26) 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.* **2019**, *31*, 3564-3572.

(27) Xie, T.; Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.* **2018**, *120*, 145301.

(28) Louis, S.-Y.; Zhao, Y.; Nasiri, A.; Wang, X.; Song, Y.; Liu, F.; Hu, J. Graph convolutional neural networks with global attention for improved materials property prediction. *Phys. Chem. Chem. Phys.* **2020**, *22*, 18141-18148.

(29) Park, C. W.; Wolverton, C. Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery. *Phys. Rev. Mater.* **2020**, *4*, 063801.

(30) Karamad, M.; Magar, R.; Shi, Y.; Siahrostami, S.; Gates, I. D.; Barati Farimani, A. Orbital graph convolutional neural network for material property prediction. *Phys. Rev. Mater.* **2020**, *4*, 093801.

(31) Back, S.; Yoon, J.; Tian, N.; Zhong, W.; Tran, K.; Ulissi, Z. W. Convolutional Neural Network of Atomic Surface Structures To Predict Binding Energies for High-Throughput Screening of Catalysts. *J. Phys. Chem. Lett.* **2019**, *10*, 4401-4408.

(32) Palizhati, A.; Zhong, W.; Tran, K.; Back, S.; Ulissi, Z. W. Toward Predicting Intermetallics Surface Properties with High-Throughput DFT and Convolutional Neural Networks. *J. Chem. Inf. Model.* **2019**.

(33) Gu, G. H.; Noh, J.; Kim, S.; Back, S.; Ulissi, Z.; Jung, Y. Practical Deep-Learning Representation for Fast Heterogeneous Catalyst Screening. *J. Phys. Chem. Lett.* **2020**, *11*, 3185-3191.

(34) MatDeepLearn. https://github.com/vxfung/MatDeepLearn (accessed 1/4/2021.

(35) Mamun, O.; Winther, K. T.; Boes, J. R.; Bligaard, T. High-throughput calculations of catalytic properties of bimetallic alloy surfaces. *Scientific Data* **2019**, *6*, 76.

(36) Andrew, R.; Shaelyn, I.; Debmalya, R.; Zhenpeng, Y.; Alan, A.-G.; Laura, G.; Justin, N.; Randall Q., S.: *Machine Learning the Quantum-Chemical Properties of Metal–Organic Frameworks for Accelerated Materials Discovery with a New Electronic Structure Database*, 2020.

(37) Haastrup, S.; Strange, M.; Pandey, M.; Deilmann, T.; Schmidt, P. S.; Hinsche, N. F.; Gjerding, M. N.; Torelli, D.; Larsen, P. M.; Riis-Jensen, A. C.; Gath, J.; Jacobsen, K. W.; Jørgen Mortensen, J.; Olsen, T.; Thygesen, K. S. The Computational 2D Materials Database: high-throughput modeling and discovery of atomically thin crystals. *2D Materials* **2018**, *5*, 042002.

(38) Fung, V.; Jiang, D.-e. Exploring Structural Diversity and Fluxionality of Ptn (n = 10–13) Clusters from First-Principles. *J. Phys. Chem. C* **2017**, *121*, 10796-10802.

(39) Bergstra, J.; Yamins, D.; Cox, D. In *Tilte*2013; PMLR.

(40) Liaw, R.; Liang, E.; Nishihara, R.; Moritz, P.; Gonzalez, J. E.; Stoica, I. J. a. p. a. Tune: A research platform for distributed model selection and training. **2018**.

(41) Fey, M.; Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428* **2019**.

(42) Unke, O. T.; Chmiela, S.; Sauceda, H. E.; Gastegger, M.; Poltavsky, I.; Schütt, K. T.; Tkatchenko, A.; Müller, K.-R. Machine Learning Force Fields. *arXiv preprint arXiv:2010.07067* **2020**.

(43) Behler, J. Constructing high-dimensional neural network potentials: A tutorial review. *Int. J. Quantum Chem* **2015**, *115*, 1032-1050.

(44) Kipf, T. N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* **2016**.

(45) Faber, F.; Lindmaa, A.; von Lilienfeld, O. A.; Armiento, R. Crystal structure representations for machine learning models of formation energies. *Int. J. Quantum Chem* **2015**, *115*, 1094-1101.

(46) Himanen, L.; Jäger, M. O. J.; Morooka, E. V.; Federici Canova, F.; Ranawat, Y. S.; Gao, D. Z.; Rinke, P.; Foster, A. S. DScribe: Library of descriptors for machine learning in materials science. *Comput. Phys. Commun.* **2020**, *247*, 106949.

(47) Hestness, J.; Narang, S.; Ardalani, N.; Diamos, G.; Jun, H.; Kianinejad, H.; Patwary, M.; Ali, M.; Yang, Y.; Zhou, Y. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409* **2017**.

(48) You, J.; Ying, R.; Leskovec, J. Design Space for Graph Neural Networks[Online early access]2020. https://ui.adsabs.harvard.edu/abs/2020arXiv201108843Y (accessed November 01, 2020).

(49) Qiao, Z.; Welborn, M.; Anandkumar, A.; Manby, F. R.; MillerIII, T. F. OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *J. Chem. Phys.* **2020**, *153*, 124111.

(50) Chen, C.; Zuo, Y.; Ye, W.; Li, X.; Ong, S. P. Multi-fidelity Graph Networks for Machine Learning the Experimental Properties of Ordered and Disordered Materials. *arXiv preprint arXiv:2005.04338* **2020**.