

ARTICLE OPEN



Atomistic Line Graph Neural Network for improved materials property predictions

Kamal Choudhary^{1,2,3,4} and Brian DeCost^{1,4}

Graph neural networks (GNN) have been shown to provide substantial performance improvements for atomistic material representation and modeling compared with descriptor-based machine learning models. While most existing GNN models for atomistic predictions are based on atomic distance information, they do not explicitly incorporate bond angles, which are critical for distinguishing many atomic structures. Furthermore, many material properties are known to be sensitive to slight changes in bond angles. We present an Atomistic Line Graph Neural Network (ALIGNN), a GNN architecture that performs message passing on both the interatomic bond graph and its line graph corresponding to bond angles. We demonstrate that angle information can be explicitly and efficiently included, leading to improved performance on multiple atomistic prediction tasks. We ALIGNN models for predicting 52 solid-state and molecular properties available in the JARVIS-DFT, Materials project, and QM9 databases. ALIGNN can outperform some previously reported GNN models on atomistic prediction tasks with better or comparable model training speed.

npj Computational Materials (2021)7:185; <https://doi.org/10.1038/s41524-021-00650-1>

INTRODUCTION

Graphs are a powerful non-Euclidean data structure method for establishing relationships between features (nodes) and their relationships (edges)^{1,2}. Graph neural networks (GNN)^{3,4} have immense potential for modeling complex phenomena. Common applications of GNNs include community detection and link prediction in social networks^{5,6}, functional time series on brain structures⁷, gene DNA on regulatory networks⁸, information flow through telecommunications networks⁹, and property prediction for molecular and solid materials¹⁰. From a quantum chemistry point of view, GNNs provide a unique opportunity to predict properties of solids, molecules, and proteins in a much faster way rather than by solving the computationally expensive Schrodinger equation^{11–14}.

There has been rapid progress in the development of GNN architectures for predicting material properties such as SchNet¹⁰, Crystal Graph Convolutional Neural Networks (CGCNN)¹⁵, MatErials Graph Network (MEGNet)¹⁶, improved Crystal Graph Convolutional Neural Networks (iCGCNN)¹⁷, OrbNet¹⁸, and similar variants^{19–31}. This family of models represents a molecule or crystalline material as a graph with one node for each constituent atom and edges corresponding to interatomic bonds. A common theme is the use of elemental properties as node features and interatomic distances and/or bond valences as edge features. Through multiple layers of graph convolution updating node features based on their local chemical environment, these models can implicitly represent many-body interactions. However, many important material properties (especially electronic properties such as band gaps) are highly sensitive to structural features such as bond angles and local geometric distortions. It is possible that these models are not able to efficiently learn the importance of such many-body interactions. Explicit inclusion of angle-based information has already been shown to improve models with hand-crafted

features such as classical force-field inspired descriptors (CFID)³². Recently, there has been growing interest in the explicit incorporation of bond angles and other many-body features^{17,19,20}.

In this work, we use line graph neural networks inspired by those proposed in ref. ⁶ to develop an alternative way to include angular information to provide high accuracy models. Briefly, the line graph $L(g)$ is a graph derived from another graph g that describes the connectivity of the edges in g . While the nodes of an atomistic graph correspond to atoms and its edges correspond to bonds, the nodes of an atomistic line graph correspond to interatomic bonds and its edges correspond to bond angles. Our model alternates between graph convolution on these two graphs, propagating bond angle information through interatomic bond representations to the atom-wise representations and vice versa. We use both the bond distances and angles in the line graph to incorporate finer details of atomic structure which leads to higher model performance. Our Atomistic Line Graph Neural Network (ALIGNN) models are implemented using the deep graph library (DGL)³³ which allows efficient construction and neural message passing for different types of graphs. ALIGNN is a part of the Joint Automated Repository for Various Integrated Simulations (JARVIS) infrastructure³⁴. We train ALIGNN models for several crystalline material properties from JARVIS-density functional theory (DFT)^{34–44} and Materials project⁴⁵ (MP) datasets as well as molecular properties from QM9⁴⁶ database.

RESULTS AND DISCUSSION

Atomistic graph representation

ALIGNN performs Edge-gated graph convolution⁴ message passing updates on both the atomistic bond graph (atoms are nodes, bonds are edges) and its line graph (bonds are nodes,

¹Materials Measurement Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA. ²Theiss Research La Jolla, California 92037, USA. ³DeepMaterials LLC, Silver Spring, MD 20906, USA. ⁴These authors contributed equally: Kamal Choudhary, Brian DeCost. [✉]email: kamal.choudhary@nist.gov

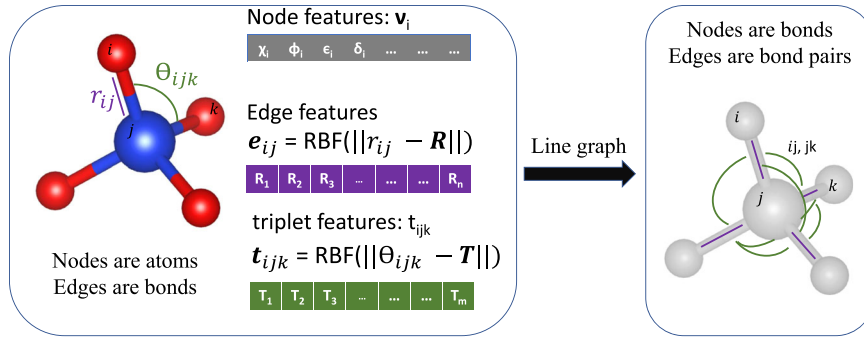


Fig. 1 Schematic showing undirected crystal graph representation and corresponding line graph construction for a SiO_4 polyhedron. For simplicity, only Si–O bonds are illustrated. The ALIGNN convolution layer alternates between message passing on the bond graph (left) and its line graph (or bond adjacency graph, right).

bond pairs with one common atom are edges). The Edge-gated graph convolution variant has the distinct advantage of updating both node and edge features. Because each edge in the bond graph directly corresponds to a node in the line graph, ALIGNN can aggregate features from bond pairs to efficiently update atom and bond representations by alternating between message passing updates on the bond graph and its line graph.

For crystals, we use a periodic 12-nearest-neighbor graph construction. We expand this nearest-neighbor graph to include edges to all atoms in the neighbor shell of the 12th-nearest neighbor. Each node in the atomistic graph is assigned 9 input node features based on its atomic species: electronegativity, group number, covalent radius, valence electrons, first ionization energy, electron affinity, block, and atomic volume. This feature set is inspired by the CGCNN¹⁵ model. The initial edge features are interatomic bond distances. We use a radial basis function (RBF) expansion with support between 0 and 8 Å for crystals and up to 5 Å for molecules. This undirected graph then can be represented as $G = (v, \epsilon)$ where v are nodes and ϵ are edges i.e., a collection of (u_i, u_j) linking vertices from u_i to u_j . G has an associated node feature set $H = \{h_1, \dots, h_N\}$, where h_i is the feature vector associated with node u_i .

Atomistic line graph representation

The atomistic line graph is derived from the atomistic graph. Each node in the line graph corresponds to an edge in the original atomistic graph; both entities represent interatomic bonds, and in our work, they share latent representations. Edges in the line graph correspond to triplets of atoms or pairs of interatomic bonds. The initial line graph edge features are an RBF expansion of the bond angle cosines: $\theta = \arccos\left(\frac{r_{ij} \cdot r_{jk}}{|r_{ij}| |r_{jk}|}\right)$, where r_{ij} and r_{jk} are atomic displacement vectors between atoms i , j , and k . A schematic of an atomistic graph and corresponding atomistic line graph is shown in Fig. 1. To avoid ambiguity between the node and edge features of the atomistic graph and its line graph, we write atom, bond, and triplet representations as h , e , and t .

Edge gated graph convolution

ALIGNN uses Edge-gated graph convolution⁴ convolution for updating both node and edge features. This convolution is similar to the CGCNN update, except that edge features are only incorporated into normalized edge gates. Furthermore, edge gated graph convolution uses the pre-aggregated edge messages to update the edge representations.

Edge gated graph convolution updates node representations h^l from layer l according to the formula:

$$h_i^{l+1} = f\left(h_i^l \left\{ h_j^l \right\}_{j \in N_i}\right) \quad (1)$$

$$h_i^{l+1} = h_i^l + \text{SiLU}\left(\text{Norm}\left(W_{src}^l h_i^l + \sum_{j \in N_i} \hat{e}_{ij}^l W_{dst}^l h_j^l\right)\right) \quad (2)$$

$$\hat{e}_{ij}^l = \frac{\sigma(e_{ij}^l)}{\sum_{k \in N_i} \sigma(e_{ik}^l) + \epsilon} \quad (3)$$

$$e_{ij}^l = e_{ij}^{l-1} + \text{SiLU}\left(\text{Norm}\left(A^l h_i^{l-1} + B^l h_j^{l-1} + C^l e_{ij}^{l-1}\right)\right) \quad (4)$$

The edge messages in this Eq. (4) are equivalent to the gating term in the CGCNN update¹⁵, which coalesces the weight matrices A , B , and C into W_{gate} , and the augmented edge representation

$$z_{ij} = h_i \oplus h_j \oplus e_{ij} \quad (5)$$

$$e_{ij}^l = e_{ij}^{l-1} + \text{SiLU}\left(\text{Norm}\left(W_{\text{gate}}^l z_{ij}^{l-1}\right)\right) \quad (6)$$

ALIGNN update

One ALIGNN layer composes an edge-gated graph convolution on the bond graph (g) with an edge-gated graph convolution on the line graph ($L(g)$), as illustrated in Fig. 2. To avoid ambiguity between the node and edge features of the atomistic graph and its line graph, we write atom, bond, and triplet representations as h , e , and t . The line graph convolution produces bond messages m that are propagated to the atomistic graph, which further updates the bond features in combination with atom features h .

$$m^l, t^l = \text{Edge Gated Graph Conv}(L(g), e^{l-1}, t^{l-1}) \quad (7)$$

$$h^l, e^l = \text{Edge Gated Graph Conv}(g, h^{l-1}, m^l) \quad (8)$$

Overall model architecture and training

We use N layers of ALIGNN updates followed by M layers of edge-gated graph convolution (GCN) updates on the bond graph. We use Sigmoid Linear Unit (SiLU, also known as Swish) activations instead of rectified linear unit (ReLU) or Softplus because it is twice differentiable like Softplus but can result in a better empirical performance like ReLU on many tasks. After $N+M$ graph convolution layers, our networks perform global average pooling over nodes and finally predict the target properties with a single fully connected regression or classification layers. Table 1 presents the default hyperparameters of the ALIGNN model used to train the models reported in “Model performance” section. These hyperparameters were selected through a combination of

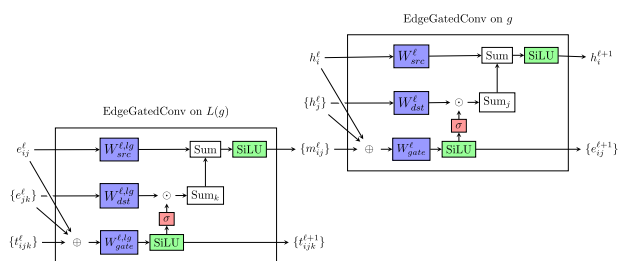


Fig. 2 Schematic of the ALIGNN layer structure. The ALIGNN layer first performs edge-gated graph convolution on the line graph to update pair and triplet features. The newly updated pair features are propagated to the edges of the direct graph and further updated with the atom features in a second edge-gated graph convolution applied to the direct graph.

Table 1. ALIGNN model configuration used for both solid-state and molecular machine learning models.

Parameter	Value
ALIGNN layers	4
GCN layers	4
Edge input features	80
Triplet input features	40
Embedding features	64
Hidden features	256
Normalization	Batch normalization
Batch size	64
Learning rate	0.001

hypothesis-driven experiments and random hyperparameter search, as discussed in detail in the “Methods” section. “Model analysis” section provides a detailed analysis of the sensitivity of model performance and computational cost.

Model performance

Model performance can vary substantially depending on the dataset and task. To evaluate the performance of ALIGNN, we currently use two different solid-state property datasets (Materials Project and JARVIS-DFT) as well as molecular property dataset QM9. Because the solid-state datasets are continuously updated, we use time-versioned snapshots of them, specifically selecting the MP version used by previous works to facilitate a direct comparison of model performance with the literature. It is likely that as these dataset sizes increase in the future the performance of the model can be further improved. We select the MP 2018.6.1 version which consists of 69,239 materials with properties such as Perdew Burke-Ernzerhof functional (PBE)⁴⁷ bandgaps and formation energies. Similarly, we use 2021.8.18 version of JARVIS-DFT dataset, which consists of 55,722 materials with several properties such as van der Waals correction with optimized Becke88 functional (OptB88vdW)⁴⁸ bandgaps, formation energies, dielectric constants, Tran-Blaha modified Becke Johnson potential (MBJ)⁴⁹ bandgaps and dielectric constants, bulk, shear modulus, magnetic moment, density functional perturbation theory (DFPT) based maximum piezoelectric coefficients, Boltztrap⁵⁰ based Seebeck coefficient, power factor, maximum absolute value of electric field gradient and two-dimensional materials exfoliation energies. All of these properties are critical for functional materials design. For the MP dataset we use a train-validation-test split of 60,000–5000–4239 as used by SchNet¹⁰ and MEGNet¹⁶. For the JARVIS-DFT dataset and its properties, we use 80 %:10 %:10 % splits. For QM9 dataset we

Table 2. Test set performance on the Materials Project dataset.

Prop	Unit	MAD	CFID	CGCNN	MEGNet	SchNet	ALIGNN	MAD:MAE
E_f	eV/atom	0.93	0.104	0.039	0.028	0.035	0.022	42.27
E_g	eV	1.35	0.434	0.388	0.33	—	0.218	6.19

Predictions on test set are shown in parity plots in Supplementary Figs. 1, 2.

use a train-validation-test split of 110,000–10,000–10,829 as used by SchNet¹⁰, DimeNet++²⁰, and MEGNet¹⁶.

Performance of ALIGNN models on MP is shown in Table 2, which shows the regression model performance in terms of mean absolute error metric (MAE). The best MAEs for formation energy (E_f) and band gap (E_g) with ALIGNN are 0.022 eV(atom)⁻¹ and 0.218 eV, respectively. In terms of E_f , ALIGNN outperforms reported values of CGCNN, MEGNet, and SchNet models by 43.6%, 21.4%, and 37.1%, respectively. For E_g , ALIGNN outperforms CGCNN and MEGNet by 43.8% and 33.9%, respectively. Good performance on well-known and well-characterized datasets ensures high prediction accuracy of ALIGNN models. Because each property has different units and in general a different variance, we also report the mean absolute deviation (MAD) for each property to facilitate an unbiased comparison of the model performance between different properties. The MAD values represent the performance of a random guessing model with average value prediction for each data point. We also report the CFID based predictions for comparison. Clearly, all the neural networks, especially ALIGNN, perform much better than the corresponding MAD of the dataset as well as CFID performance. Analyzing the MAD:MAE (ALIGNN) ratio, we observe that the ratio could be as high as 42.27 model. Generally, a model with high MAD:MAE ratio (such as 5 and above) is considered a good predictive model⁵¹.

Similarly, we train ALIGNN models on the JARVIS-DFT^{34–44} dataset which consists of data for 55,722 materials. In addition to properties such as formation energies, and bandgaps it also consists several unique quantities such as solar-cell efficiency (spectroscopic limited maximum efficiency, SLME), topological spin-orbit spillage, dielectric constant with (ϵ_x (DFPT)), and without ionic contributions (ϵ_x (OPT, MBJ)), exfoliation energies for two-dimensional (2D), electric field gradients (EFG), Voigt bulk (Kv) and shear modulus (Gv), energy above convex hull (ehull), maximum piezoelectric stress (e_{ij}) and strain (d_{ij}) tensors, n-type and p-type Seebeck coefficient and power factors (PF), crystallographic averages of electron (m_e) and hole (m_h) effective masses. As we converge plane wave-cutoff (ENCUT) and k-points used in Brillouin zone integration (Kpoint-length), we attempt to make machine learning predictions on these unique quantities as well. Such a large variety of properties allow a thorough testing of our ALIGNN models. More details for individual properties, its precision with respect to experimental measurements, applicability, and limitations can be found in respective works. However, it is important to mention that many important issues such as tackling systematic underestimation of bandgaps by DFT methods, the inclusion of van der Waals bonding, and the inclusion of spin-orbit coupling interactions, all critically important for materials-design perspective have been key areas of improvements for the JARVIS-DFT dataset. For instance, meta-GGA (generalized gradient approximation) based Tran-Blaha modified Becke Johnson potential (TbMBJ) band gaps are more reliable and comparable to experimental data than Perdew Burke-Ernzerhof functional (PBE) or van der Waals correction with optimized Becke88 functional (OptB88vdW) bandgaps, but their calculations are computationally expensive and hence they are underrepresented in the dataset. In addition to the ALIGNN performance, we also include hand-crafted Classical force-field inspired descriptors (CFID) descriptor and

Table 3. Regression model performances on JARVIS-DFT dataset for 29 properties using CFID, CGCNN and ALIGNN models on 55,722 materials.

Property	Units	MAD	CFID	CGCNN	ALIGNN	MAD: MAE
Formation energy	eV(atom) ⁻¹	0.86	0.14	0.063	0.033	26.06
Bandgap (OPT)	eV	0.99	0.30	0.20	0.14	7.07
Total energy	eV(atom) ⁻¹	1.78	0.24	0.078	0.037	48.11
Ehull	eV	1.14	0.22	0.17	0.076	15.00
Bandgap (MBJ)	eV	1.79	0.53	0.41	0.31	5.77
Kv	GPa	52.80	14.12	14.47	10.40	5.08
Gv	GPa	27.16	11.98	11.75	9.48	2.86
Mag. mom	μB	1.27	0.45	0.37	0.26	4.88
SLME (%)	No unit	10.93	6.22	5.66	4.52	2.42
Spillage	No unit	0.52	0.39	0.40	0.35	1.49
Kpoint-length	Å	17.88	9.68	10.60	9.51	1.88
Plane-wave cutoff	eV	260.4	139.4	151.0	133.8	1.95
ε _x (OPT)	No unit	57.40	24.83	27.17	20.40	2.81
ε _y (OPT)	No unit	57.54	25.03	26.62	19.99	2.88
ε _z (OPT)	No unit	56.03	24.77	25.69	19.57	2.86
ε _x (MBJ)	No unit	64.43	30.96	29.82	24.05	2.68
ε _y (MBJ)	No unit	64.55	29.89	30.11	23.65	2.73
ε _z (MBJ)	No unit	60.88	29.18	30.53	23.73	2.57
ε (DFPT:elec+ionic)	No unit	45.81	43.71	38.78	28.15	1.63
Max. piezoelectric strain coeff (d _{ij})	CN ⁻¹	24.57	36.41	34.71	20.57	1.19
Max. piezo. stress coeff (e _{ij})	Cm ⁻²	0.26	0.23	0.19	0.147	1.77
Exfoliation energy	meV(atom) ⁻¹	62.63	63.31	50.0	51.42	1.22
Max. EFG	10 ²¹ Vm ⁻²	43.90	24.54	24.7	19.12	2.30
avg. m _e	electron mass unit	0.22	0.14	0.12	0.085	2.59
avg. m _h	electron mass unit	0.41	0.20	0.17	0.124	3.31
n-Seebeck	μVK ⁻¹	113.0	56.38	49.32	40.92	2.76
n-PF	μW(mK ²) ⁻¹	697.80	521.54	552.6	442.30	1.58
p-Seebeck	μVK ⁻¹	166.33	62.74	52.68	42.42	3.92
p-PF	μW(mK ²) ⁻¹	691.67	505.45	560.8	440.26	1.57

Predictions on test set are shown in Supplementary Figs. 3–31.

CGCNN MAE performances for these properties using identical data-splits.

In Table 3 we show the performance on regression models for different properties in the JARVIS-DFT database. We observe that ALIGNN models outperform CFID descriptors by up to 4 times, suggesting GNNs can be a very powerful method for multiple material property predictions. Also, ALIGNN outperforms CGCNN by more than 2 times (such as for OptB88vdW total energy). Cross-dataset comparison of corresponding property entries in Tables 2, 3 shows that generally models generally obtain better performance on the MP dataset, which we attribute primarily to the larger size of MP. For example, the MAE for the formation energy target on MP dataset is 50% lower than for JARVIS-DFT. However, for some targets, the differences in the DFT method and settings, as well as potential differences in the material-space distribution, might significantly contribute to the difficulty of a prediction task. For example, the MAE on high throughput band gaps is lower (by 35.7%) for the JARVIS-DFT dataset, which is interesting in light of MP's dataset size advantage over JARVIS-DFT. One potential source of this discrepancy is the differing computational methodologies used, such as different functionals (PBE vs OptB88vdW), use of the DFT+U method, and settings for various DFT hyperparameters like smearing and k-point settings, all of which can influence the values of computed bandgaps as discussed in ref. 37. Another potential contributing factor could be differing levels of dataset bias in the MP and JARVIS-DFT

datasets stemming from differing distributions in material space. Clarifying this situation is beyond the scope of the present work, though it is of great importance for the atomistic modeling community to resolve.

Nevertheless, application of ALIGNN models on different datasets shows improvements for materials-property predictions. Both CFID, CGCNN and ALIGNN models' MAEs are lower than the corresponding MADs. The MAD:MAE ratios can vary for energy related quantities from a high value of 48.11 (total energy), and 26.06 (formation energy model) to low values such as for DFPT based piezoelectric strain coefficients (1.19) and dielectric constant with ionic contributions (1.63). The results indicate that there is still much room for improvement for the GNN models, especially for electronic properties.

As we notice above, the regression tasks for some of the electronic properties do not show very high MAD: MAE. We train classification models for some of them. Classification tasks predict labels such as high value/low value (based on a selected threshold) as 1 and 0 instead of predicting actual data in regression tasks. Such models can be useful for fast screening purposes³⁸ for computationally expensive methods. We evaluate the performance of these classifiers using the receiver operating characteristic curve area under the curve (ROC AUC). A random guessing model has a ROC AUC of 0.5, while a perfect model would be a ROC AUC of 1.0. Interestingly, we notice most of our classification models (as shown in Table 4) have high ROC AUCs,

Table 4. Classification task ROC AUC performance on JARVIS-DFT dataset for ALIGNNN models.

Model	Threshold	ALIGNNN
Metal/non-metal classifier (OPT)	0.01 eV	0.92
Metal/non-metal classifier (MBJ)	0.01 eV	0.92
Magnetic/non-magnetic classifier	0.05 μ B	0.91
High/low SLME	10 %	0.83
High/low spillage	0.1	0.80
Stable/unstable (ehull)	0.1 eV	0.94
High/low-n-Seebeck	-100μ VK ⁻¹	0.88
High/low-p-Seebeck	100μ VK ⁻¹	0.92
High/low-n-powerfactor	1000μ W(mK ²) ⁻¹	0.74
High/low-p-powerfactor	1000μ W(mK ²) ⁻¹	0.74

The ROC curve plots for these models are provided in Supplementary Figs. 32–41.

Table 5. Regression model performances on QM9 dataset for 11 properties using ALIGNNN.

Target	Units	SchNet	MEGNet	DimeNet++	ALIGNNN
HOMO	eV	0.041	0.043	0.0246	0.0214
LUMO	eV	0.034	0.044	0.0195	0.0195
Gap	eV	0.063	0.066	0.0326	0.0381
ZPVE	eV	0.0017	0.00143	0.00121	0.0031
μ	Debye	0.033	0.05	0.0297	0.0146
α	Bohr ³	0.235	0.081	0.0435	0.0561
R ²	Bohr ²	0.073	0.302	0.331	0.5432
U0	eV	0.014	0.012	0.00632	0.0153
U	eV	0.019	0.013	0.00628	0.0144
H	eV	0.014	0.012	0.00653	0.0147
G	eV	0.014	0.012	0.00756	0.0144

These models were trained with same parameters as solid-state databases but for 1000 epochs. Predictions on test set are shown in Supplementary Figs. 42–52.

ranging up to a maximum value of 0.94 (for convex hull stability) showing their usefulness for material classification-based applications. All results are based on the performance of 10 % test data which is never used during the training or model selection procedures.

Next, we evaluate the ALIGNNN model on QM9 molecular property dataset (130,829 molecules) and compare it with other well-known models such as SchNet¹⁰, MatErials Graph Network (MEGNet)¹⁶, and DimeNet++²⁰ as shown in Table 5. The results from models other than ALIGNNN are reported as given in corresponding papers, not necessarily reproduced by us. QM9 provides DFT calculated molecular properties such as highest occupied molecular orbital (HOMO), lowest unoccupied molecular orbital (LUMO), energy gap, zero-point vibrational energy (ZPVE), dipole moment, isotropic polarizability, electronic spatial extent, internal energy at 0 K, internal energy at 298 K, enthalpy at 298 K, and Gibbs free energy at 298 K. ALIGNNN outperforms competing methods for HOMO and dipole moment tasks while other accuracies are similar to the SchNet model. Most importantly, all ALIGNNN results reported here use the same set of hyperparameters obtained by tuning to validation performance on the JARVIS-DFT bandgap target, suggesting that

Table 6. Effect of changing ALIGNNN and GCN layers on machine learning models for JARVIS-DFT OptB88vdW formation energy database in ALIGNNN models.

Layers	GCN-0	GCN-1	GCN-2	GCN-3	GCN-4
ALIGNNN-0	0.445	0.065	0.050	0.045	0.044
ALIGNNN-1	0.064	0.041	0.037	0.036	0.037
ALIGNNN-2	0.039	0.036	0.034	0.034	0.034
ALIGNNN-3	0.036	0.034	0.033	0.034	0.034
ALIGNNN-4	0.034	0.034	0.034	0.034	0.033

The bold values represent the best performing models.

ALIGNNN provides robust performance with respect to different datasets and material types.

Model analysis

We ablate individual components of the ALIGNNN model to evaluate their contribution to the overall architecture. Keeping other parameters intact in the ALIGNNN model (as specified in Table 1), we vary the number of ALIGNNN and GCN layers as shown in Table 6 and Supplementary Table 1 for JARVIS-DFT OptB88vdW formation energies and bandgaps respectively. We find that without any graph convolution layers the MAE for the formation energy and bandgap are 1248.5% and 453.6% higher than the default model. Adding even a single ALIGNNN or GCN layer can reduce the MAE by 102.9% illustrating the importance of these layers. However, further increase in ALIGNNN/GCN layers doesn't scale well and performance quickly saturates at a depth of 4. Excluding GCN layers and increasing ALIGNNN layers and vice versa show the individual importance of these layers. Performance of GCN-only models saturates at 4 layers with 44 meV/atom MAE on the JARVIS-DFT formation energy task, while ALIGNNN-only models saturate at $34 \text{ meV(atom)}^{-1}$ —a relative reduction of 29.14%. Each of these models, along with the other highlighted configurations in Table 6, performs four atom feature updates via graph convolution modules. At least two ALIGNNN updates are needed to obtain peak performance. Additional atom feature updates provide little marginal increase in performance. This is consistent with the widely reported difficulty of GCN architectures scaling in depth beyond a few layers⁵².

Figure 3 shows in detail the tradeoff between the performance benefit of including ALIGNNN layers and their computational overhead relative to GCN layers. Per-epoch timing for each configuration is reported in Supplementary Table 2. All GCN-only configurations (annotated with the number of GCN layers) are on the low-computation portion of the pareto frontier, but the high-accuracy portion of the pareto frontier is dominated by ALIGNNN/GCN combinations with at least two ALIGNNN updates. The ALIGNNN-2/GCN-2 configuration obtains peak performance (again, relative reduction of MAE by 29.14 %) with a computational overhead of roughly 2 \times relative to the GCN-4 configuration. Supplementary Table 1 and Supplementary Fig. 53 present layer ablation study yielding similar conclusions on the JARVIS-DFT OptB88vdW band gap target.

This layer ablation study clearly demonstrates that inclusion of bond angle information and propagation of bond and pair features through the node updates improves the generalization ability of atomistic GCN models. This is satisfying from a materials science perspective, as interatomic bonding theory clearly motivates the notion that inclusion of bond angles should improve accuracy of the model.

Similarly, we vary the number of hidden features (i.e., the width of the graph convolution layers), edge input features, and embedding input features to evaluate the MAE performance for

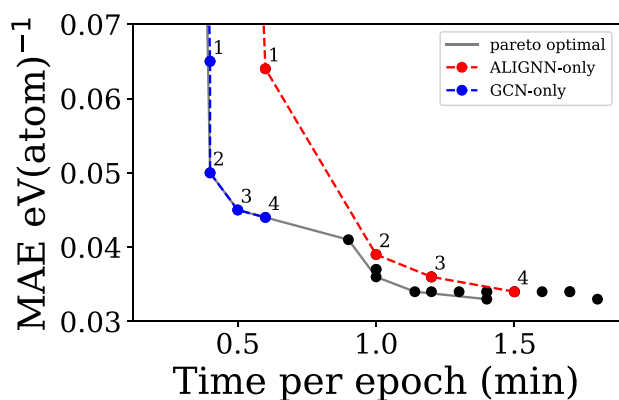


Fig. 3 ALIGNN accuracy-cost ablation study on JARVIS-DFT formation energy target. The red and blue markers represent the number of layers in GCN-only and ALIGNN-only models.

JARVIS-DFT formation energy and bandgap model in comparison with the default model in Table 1. In Supplementary Table 3, we observe that the marginal performance from increasing the hidden features saturates at 256 for both properties. Supplementary Table 4 shows that the number of edge input features is optimal at 80 for formation energy model, while for the bandgap model performance saturates at 40. Similarly, embedding features are optimized at 64 for formation energy while 32 for bandgap model (Supplementary Table 5). Additionally, we tried three different node feature attributes such as 1) CFID chemical features (total 438), only atomic number (total 1), and default CGCNN type attributes (total 92) and compared them for formation energy model in Supplementary Table 6. We observe that the default node attributes have the lowest MAE.

Next, we study time taken per epoch of several models for QM9 and JARVIS-DFT formation energy dataset in Supplementary Table 7. To help facilitate fair comparison, we train all models with the same computational resources using the reference implementations and configurations reported in the literature. We note that the timing code for the reference implementations of different methods may include differing amounts of overhead. For example, the ALIGNN timings reported in Supplementary Table 7 amortize the overhead of initial atomistic graph construction across 300 epochs, and each epoch includes the overhead of evaluating the model on the full training and validation sets for performance tracking. Additionally, the computational cost of deep learning models, in general, is not independent of certain hyperparameters; in particular, larger batch sizes can better leverage modern accelerator hardware by exposing more parallelism. We find ALIGNN requires less training time per epoch time compared to other models except DimeNet++ and MEGNet. However, it is important to note that DimeNet++ and other models usually take around 1000 epochs or more to reach desired accuracy, while ALIGNN can converge in about 300 epochs, resulting in lower overall training cost for similar or better accuracy.

While we report timing comparisons using our standard hyperparameter configuration used to train models reported in “Model performance” section, through subsequent model analysis we have identified several strategies that substantially reduce computational workload without incurring a large performance penalty. We observe in Supplementary Fig. 54 that model performance converges after 300 epochs; shorter training budgets incur a modest performance reduction and slightly increased variance with respect to the training split. The performance tradeoff presented in Table 6 and Fig. 3 indicates that switching from the default configuration of 4 layers each of ALIGNN and GCN updates to 2 layers each could offer a speedup of $\sim 1.5\times$ with negligible reduction in accuracy. Finally, we performed a drop-in replacement study comparing batch normalization and layer

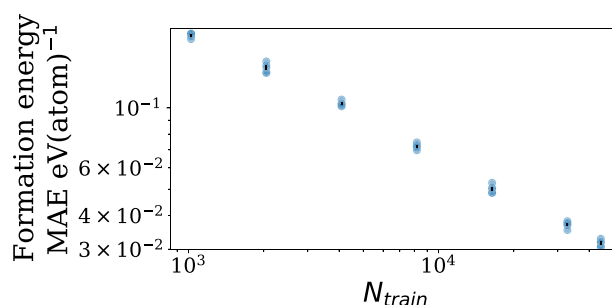


Fig. 4 Learning curve for JARVIS-DFT formation energy regression target. Blue markers indicate validation set MAE scores for individual cross-validation iterates. Error bars indicate the mean cross-validation MAE \pm one standard error of the mean.

normalization in Supplementary Table 8, finding that switching to layer normalization provides an additional $\sim 1.7\times$ speedup with a slight degradation in validation loss and negligible degradation in validation MAE. Because the cost of retraining models for all targets reported is still high, and because some of these strategies equally apply to competing models, we defer a more comprehensive performance-cost study to future work.

Finally, we simultaneously investigate the effects of dataset size and different train-validation-test splits by performing a learning curve study in cross-validation for the JARVIS-DFT formation energy (Fig. 4 and Supplementary Table 9) and bandgap (Supplementary Fig. 55 and Supplementary Table 9) targets. We perform the cross-validation splitting procedure by merging the standard JARVIS-DFT train and validation sets and randomly sampling without replacement N_{train} training samples and 5000 validation samples. The learning curve study shows no sign of diminishing marginal returns for additional data up to the full size of the JARVIS-DFT dataset. On the full training set size (44,577) we obtain an average validation MAE of 0.0316 ± 0.0004 eV/at (uncertainty corresponds to the standard error of the mean over five cross-validation (CV) iterates). The standard deviation over CV iterates is 0.0009 eV/at, indicating that model performance is relatively insensitive to the dataset split.

In summary, we have developed an ALIGNN model which uses the line graph neural network that improves the performance of GNN predictions for solids and molecules. We have demonstrated that explicit inclusion of angle-based networks in GNNs can significantly improve model performance. A key contribution of this work is the inclusion and development of both the undirected atomistic graph and its line graph counterpart for solid-state and molecular materials. We develop regression and classification ALIGNN models for some of the well-known pre-existing databases and it can be easily applied for other datasets as well. Our model significantly improved accuracies over prior GNN models. We believe the ALIGNN model will rapidly improve the machine learning prediction for several material properties and classes.

METHODS

JARVIS-DFT dataset

The JARVIS-DFT^{34–44} dataset is developed using Vienna Ab-initio simulation package (VASP)⁵³ software (please note commercial software is identified to specify procedures. Such identification does not imply recommendation by National Institute of Standards and Technology (NIST)). Most of the properties are calculated using the OptB88vdW functional⁴⁸. For a subset of the data we use TBmBJ⁴⁹ for getting better band gaps. We use density functional perturbation theory (DFPT)⁵⁴ for predicting piezoelectric and dielectric constants with both electronic and ionic contributions. The linear response theory-based⁵⁵ frequency based dielectric function was calculated using both OptB88vdW and TBmBJ and the zero-energy values are trained for the machine learning model. Note that the linear response

Received: 3 June 2021; Accepted: 2 October 2021;
Published online: 15 November 2021

based dielectric constants lack ionic contributions. The TBmBJ frequency dependent dielectric functions are used to calculate the spectroscopic limited maximum efficiency (SLME)³⁸. The magnetic moments are calculated using spin-polarized calculations considering only ferromagnetic initial configurations and neglecting any density functional theory (DFT)+U effects. The thermoelectric coefficients such as Seebeck coefficients and power factors are calculated using BoltzTrap⁵⁰ software using constant relaxation time approximation. Exfoliation energy for the van der Waals bonded two-dimensional materials are calculated as the energy per atom differences between the bulk and corresponding monolayer counterparts. The spin-orbit spillage⁴⁰ is calculated using the difference in wavefunctions of a material with and without inclusion of spin-orbit coupling effects. All the JARVIS-DFT data and Classical force-field inspired descriptors (CFID)³² are generated using the JARVIS-Tools package. The CFID baseline models are trained using the LightGBM package (please note commercial software is identified to specify procedures. Such identification does not imply recommendation by National Institute of Standards and Technology (NIST)).⁵⁶ using the models developed in ref. ³².

ALIGNN model implementation and training

The ALIGNN model is implemented in PyTorch⁵⁷ and deep graph library (DGL)³³; the training code heavily relies on PyTorch-ignite⁵⁸. For regression targets we minimize the mean squared error (MSE) loss, and for classification targets we minimize the standard negative log likelihood loss. We train all models for 300 epochs using the AdamW⁵⁹ optimizer with normalized weight decay of 10^{-5} and a batch size of 64. The learning rate is scheduled according to the one-cycle policy⁶⁰ with a maximum learning rate of 0.001. We use the same model configuration for each regression and classification target. We use the initial atom representations from the CGCNN paper, 80 initial bond radial basis function (RBF) features, and 40 initial bond angle RBF features. The atom, bond, and bond angle feature embedding layers produce 64-dimensional inputs to the graph convolution layers. The main body of the network consists of 4 ALIGNN and 4 graph convolution (GCN) layers, each with hidden dimension 256. The final atom representations are reduced by atom-wise average pooling and mapped to regression or classification outputs by a single linear layer. These hyperparameters are selected to optimize validation MAE on the JARVIS-DFT band gap task through a combination of manual hypothesis-driven experiments and random hyperparameter search facilitated and scheduled through Ray Tune⁶¹; hyperparameter ranges are given in Supplementary Table 10. The random search results indicate that model performance is most highly sensitive to the learning rate, weight decay, and convolution layer width, and beyond a relatively low threshold is insensitive to the sizes of the initial feature embedding layers.

We used NIST's Nisaba cluster to train all ALIGNN models, and we reproduce results from the literature using the reference implementations for each competing method on the same hardware. Each model is trained on a single Tesla V100 SXM2 32 gigabyte Graphics processing unit (GPU), with 8 Intel Xeon E5-2698 v4 CPU cores for concurrently fetching and preprocessing batches of data during training (please note commercial software is identified to specify procedures. Such identification does not imply recommendation by National Institute of Standards and Technology (NIST)). For the MP dataset we use a train-validation-test split of 60,000–5000–4239. For the JARVIS-DFT dataset, we use 80%:10%:10% splits. The 10% test data is never used during training procedures. For QM9 dataset we use a train-validation-test split of 110,000–10,000–10,829.

DATA AVAILABILITY

All data used in this work is available at Figshare link https://figshare.com/collections/ALIGNN_data/5429274. During the training these datasets are accessed using JARVIS-Tools's figshare module.

CODE AVAILABILITY

The code and full model and training configurations used in this work are available on GitHub at <https://github.com/usnistgov/alignn>, along with general tooling at <https://github.com/usnistgov/jarvis>. An interactive web-app for using ALIGNN models is also made available at <https://jarvis.nist.gov/jalignn>.

REFERENCES

1. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
2. Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. & Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **20**, 61–80 (2008).
3. Wu, Z. et al. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **32**, 4 (2020).
4. Dwivedi, V. P., Joshi, C. K., Laurent, T., Bengio, Y. & Bresson, X. Benchmarking graph neural networks. *arXiv* 2003, 00982. Preprint at <https://arxiv.org/abs/2003.00982> (2020).
5. Guo, Z. & Wang, H. A deep graph neural network-based mechanism for social recommendations. *IEEE Trans. Ind. Inform.* **17**, 2776 (2020).
6. Chen, Z., Li, X. & Bruna, J. Supervised community detection with line graph neural networks. *arXiv* 1705, 08415. Preprint at <https://arxiv.org/abs/1705.08415> (2017).
7. Li, X. et al. Braingnn: Interpretable brain graph neural network for fmri analysis. *Med. Image Anal.* **74**, 102233 (2021).
8. Baumbach, J. CoryneRegNet 4.0—A reference database for corynebacterial gene regulatory networks. *BMC Bioinforma.* **8**, 1–11 (2007).
9. Wu, K., Chen, Z. & Li, W. A novel intrusion detection model for a massive network using convolutional neural networks. *IEEE Access* **6**, 50850 (2018).
10. Schütt, K. T. et al. Schnet: a continuous-filter convolutional neural network for modeling quantum interactions. *arXiv* 1706, 08566. Preprint at <https://arxiv.org/abs/1706.08566> (2017).
11. Duvenaud, D. et al. Convolutional networks on graphs for learning molecular fingerprints. *arXiv* 1509, 09292. Preprint at <https://arxiv.org/abs/1509.09292> (2015).
12. Kearnes, S., McCloskey, K., Berndl, M., Pande, V. & Riley, P. Molecular graph convolutions: moving beyond fingerprints. *J. Comput. Aided* **30**, 595–608 (2016).
13. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. *PMLR* **70**, 1263 (2017).
14. Faber, F. A. et al. Prediction errors of molecular machine learning models lower than hybrid DFT error. *J. Chem. Theory Comput.* **13**, 5255–5264 (2017).
15. 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).
16. 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).
17. Park, C. W. & Wolverton, C. Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery. *Phys. Rev. Mater.* **4**, 063801 (2020).
18. Qiao, Z., Welborn, M., Anandkumar, A., Manby, F. R. & Miller, T. F. III OrbNet: deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *J. Chem. Phys.* **153**, 124111 (2020).
19. Klicpera, J., Groß, J. & Günnemann, S. Directional message passing for molecular graphs. *arXiv* 2003, 03123. Preprint at <https://arxiv.org/abs/2003.03123> (2020).
20. Klicpera, J., Giri, S., Margraf, J. T. & Günnemann, S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv* 2011, 14115. Preprint at <https://arxiv.org/abs/2011.14115> (2020).
21. Unke, O. T. & Meuwly, M. PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges. *J. Chem. Theory Comput.* **15**, 3678–3693 (2019).
22. Shui, Z. & George, K. "Heterogeneous molecular graph neural networks for predicting molecule properties". 2020 *IEEE International Conference on Data Mining (ICDM)*, 492 (2020).
23. Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R. & Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **8**, 1–8 (2017).
24. Anderson, B., Hy, T.-S. & Kondor, R. Cormorant: covariant molecular neural networks. *arXiv* 1906, 04015. Preprint at <https://arxiv.org/abs/1906.04015> (2019).
25. Zhang, S., Liu, Y. & Xie, L. Molecular mechanics-driven graph neural network with multiplex graph for molecular structures. *arXiv* 2011, 07457. Preprint at <https://arxiv.org/abs/2011.07457> (2020).
26. Lubbers, N., Smith, J. S. & Barros, K. Hierarchical modeling of molecular energies using a deep neural network. *J. Chem. Phys.* **148**, 241715 (2018).
27. Schutt, K. et al. SchNetPack: A deep learning toolbox for atomistic systems. *J. Chem. Theory Comput.* **15**, 448 (2018).
28. Jha, D. et al. Elemnet: Deep learning the chemistry of materials from only elemental composition. *Sci. Rep.* **8**, 1–13 (2018).
29. Westermayr, J., Gastegger, M. & Marquetand, P. Combining SchNet and SHARC: The SchNarc machine learning approach for excited-state dynamics. *J. Phys. Chem. Lett.* **11**, 3828 (2020).

30. Wen, M., Blau, S. M., Spotte-Smith, E. W. C., Dwaraknath, S. & Persson, K. A. BondNet: a graph neural network for the prediction of bond dissociation energies for charged molecules. *Chem* **12**, 1858 (2020).
31. Isayev, O. et al. Universal fragment descriptors for predicting properties of inorganic crystals. *Nat. Commun.* **8**, 1 (2017).
32. Choudhary, K., DeCost, B. & Tavazza, F. Machine learning with force-field-inspired descriptors for materials: Fast screening and mapping energy landscape. *Phys. Rev. Mater.* **2**, 083801 (2018).
33. Wang, M. et al. Deep graph library: a graph-centric, highly-performant package for graph neural networks. *arXiv* 1909, 01315. Preprint at <https://arxiv.org/abs/1909.01315> (2019).
34. Choudhary, K. et al. The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design. *Npj Comput. Mater.* **6**, 1–13 (2020).
35. Choudhary, K., Cheon, G., Reed, E. & Tavazza, F. Elastic properties of bulk and low-dimensional materials using van der Waals density functional. *Phys. Rev. B* **98**, 014107 (2018).
36. Choudhary, K., Kalish, I., Beams, R. & Tavazza, F. High-throughput identification and characterization of two-dimensional materials using density functional theory. *Sci. Rep.* **7**, 1–16 (2017).
37. Choudhary, K. et al. Computational screening of high-performance optoelectronic materials using OptB88vdW and TB-mBJ formalisms. *Sci. Data* **5**, 1–12 (2018).
38. Choudhary, K. et al. Accelerated discovery of efficient solar cell materials using quantum and machine-learning methods. *Chem. Mater.* **31**, 5900 (2019).
39. Choudhary, K., Garrity, K. F. & Tavazza, F. High-throughput discovery of topologically non-trivial materials using spin-orbit spillage. *Sci. Rep.* **9**, 1–8 (2019).
40. Choudhary, K., Garrity, K. F., Ghimire, N. J., Anand, N. & Tavazza, F. High-throughput search for magnetic topological materials using spin-orbit spillage, machine learning, and experiments. *Phys. Rev. B* **103**, 155131 (2021).
41. Choudhary, K., Ansari, J. N., Mazin, I. I. & Sauer, K. L. Density functional theory-based electric field gradient database. *Sci. Data* **7**, 1–10 (2020).
42. Choudhary, K., Garrity, K. F. & Tavazza, F. Data-driven discovery of 3D and 2D thermoelectric materials. *J. Condens. Matter Phys.* **32**, 475501 (2020).
43. Choudhary, K. et al. High-throughput density functional perturbation theory and machine learning predictions of infrared, piezoelectric, and dielectric responses. *Npj Comput. Mater.* **6**, 1–13 (2020).
44. Choudhary, K. & Tavazza, F. Convergence and machine learning predictions of Monkhorst-Pack k-points and plane-wave cut-off in high-throughput DFT calculations. *Comput. Mater. Sci.* **161**, 300–308 (2019).
45. Jain, A. et al. Commentary: The Materials Project: a materials genome approach to accelerating materials innovation. *APL Mater.* **1**, 011002 (2013).
46. Ramakrishnan, R., Dral, P. O., Rupp, M. & Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **1**, 1 (2014).
47. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865 (1996).
48. Klimeš, J., Bowler, D. R. & Michaelides, A. Chemical accuracy for the van der Waals density functional. *J. Condens. Matter Phys.* **22**, 022201 (2009).
49. Tran, F. & Blaha, P. Accurate band gaps of semiconductors and insulators with a semilocal exchange-correlation potential. *Phys. Rev. Lett.* **102**, 226401 (2009).
50. Madsen, G. K. & Singh, D. J. BoltzTraP. A code for calculating band-structure dependent quantities. *Comput. Phys. Commun.* **175**, 67–71 (2006).
51. Ward, L., Agrawal, A., Choudhary, A. & Wolverton, C. A general-purpose machine learning framework for predicting properties of inorganic materials. *Npj Comput. Mater.* **2**, 1 (2016).
52. Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K. I. & Jegelka, S. Representation learning on graphs with jumping knowledge networks. *PMLR* **80**, 5453 (2018).
53. Kresse, G. & Furthmüller, G. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Comput. Mater. Sci.* **6**, 15 (1996).
54. Baroni, S. & Resta, R. Ab initio calculation of the macroscopic dielectric constant in silicon. *Phys. Rev. B* **33**, 7017 (1986).
55. Gajdoš, M., Hummer, K., Kresse, G., Furthmüller, J. & Bechstedt, F. Linear optical properties in the projector-augmented wave methodology. *Phys. Rev. B* **73**, 045112 (2006).
56. Ke, G. et al. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **30**, 3146 (2017).
57. Paszke, A. et al. Pytorch: an imperative style, high-performance deep learning library. *arXiv* 1912, 01703. Preprint at <https://arxiv.org/abs/1912.01703> (2019).
58. PyTorch-ignite documentation. <https://pytorch.org/ignite/> (2020).
59. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. *arXiv* 1711, 05101. Preprint at <https://arxiv.org/abs/1711.05101> (2017).
60. Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1-learning rate, batch size, momentum, and weight decay. *arXiv* 1803, 09820. Preprint at <https://arxiv.org/abs/1803.09820> (2018).
61. Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E. & Stoica, I. Tune: a research platform for distributed model selection and training. *arXiv* 1807, 05118. Preprint at <https://arxiv.org/abs/1807.05118> (2018).

ACKNOWLEDGEMENTS

K.C. and B.D. thank the National Institute of Standards and Technology for funding, computational, and data management resources. Contributions from K.C. were supported by the financial assistance award 70NANB19H117 from the U.S. Department of Commerce, National Institute of Standards and Technology. This work was also supported by the Frontera supercomputer, National Science Foundation OAC-1818253, at the Texas Advanced Computing Center (TACC) at The University of Texas at Austin.

AUTHOR CONTRIBUTIONS

Both K.C. and B.D. equally contributed to developing the model and writing 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/s41524-021-00650-1>.

Correspondence and requests for materials should be addressed to Kamal Choudhary.

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.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

This is a U.S. government work and not under copyright protection in the U.S.; foreign copyright protection may apply 2021, corrected publication 2022