IRNet: A General Purpose Deep Residual Regression Framework for Materials Discovery

Dipendra Jha
Northwestern University
Evanston, Illinois, USA
dipendra.jha@eecs.northwestern.edu

Logan Ward
U. Chicago and Argonne National Lab
Chicago, Illinois
loganw@uchicago.edu

Zijiang Yang
Northwestern University
Evanston, Illinois, USA
zyz293@eecs.northwestern.edu

Christopher Wolverton
Northwestern University
Evanston, Illinois, USA
c-wolverton@northwestern.edu

Ian Foster
U. Chicago and Argonne National Lab
Chicago, Illinois, USA
foster@uchicago.edu

Wei-keng Liao
Northwestern University
Evanston, Illinois, USA
wkliao@eecs.northwestern.edu

Alok Choudhary
Northwestern University
Evanston, Illinois, USA
choudhar@eecs.northwestern.edu

Ankit Agrawal
Northwestern University
Evanston, Illinois, USA
ankitag@eecs.northwestern.edu

ABSTRACT

Materials discovery is crucial for making scientific advances in many domains. Collections of data from experiments and first-principle computations have spurred interest in applying machine learning methods to create predictive models capable of mapping from composition and crystal structures to materials properties. Generally, these are regression problems with the input being a 1D vector composed of numerical attributes representing the material composition and/or crystal structure. While neural networks consisting of fully connected layers have been applied to such problems, their performance often suffers from the vanishing gradient problem when network depth is increased. Hence, predictive modeling for such tasks has been mainly limited to traditional machine learning techniques such as Random Forest. In this paper, we study and propose design principles for building deep regression networks composed of fully connected layers with numerical vectors as input. We introduce a novel deep regression network with individual residual learning, IRNet, that places shortcut connections after each layer so that each layer learns the residual mapping between its output and input. We use the problem of learning properties of inorganic materials from numerical attributes derived from material composition and/or crystal structure to compare IRNet's performance against that of other machine learning techniques. Using multiple datasets from the Open Quantum Materials Database (OQMD) and Materials Project for training and evaluation, we show that IRNet provides significantly better prediction performance than the state-of-the-art machine learning approaches currently used by domain scientists. We also show that IRNet's use of individual residual learning leads to better convergence during the training phase than when shortcut connections are between multi-layer stacks while maintaining the same number of parameters.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Supervised learning by regression**; **Neural networks**.

KEYWORDS

deep learning, deep neural networks, deep residual networks, deep regression, materials discovery, predictive modeling

ACM Reference Format:
Dipendra Jha, Logan Ward, Zijiang Yang, Christopher Wolverton, Ian Foster, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. 2019. IRNet: A General Purpose Deep Residual Regression Framework for Materials Discovery . In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330703

1 INTRODUCTION

Materials discovery plays an important role in many domains of science and engineering [32, 37]. The slow pace of development and deployment of new/improved materials is a major bottleneck in the innovation cycles of emerging technologies [25]. Collection of large scale datasets through experiments and first-principle computations such as high throughput density functional theory (DFT) calculations [10, 22, 27] and the emergence of integrated data collections and registries [6, 11] have spurred the interest of materials scientists in applying machine learning (ML) models to understand materials and predict their properties [8, 13, 28, 30, 36, 39, 42, 47, 51], leading to the novel paradigm of materials informatics [3, 38, 39, 49]. Such interests have been supported by government initiatives such as the Materials Genome Initiative (MGI) [1].

Predictive modeling tasks in materials science are generally regression problems where we need to predict materials properties from an input vector composed of numerical features derived from

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330703

their composition and/or crystal structures by incorporating domain knowledge [8, 13, 24, 39, 42, 47, 51]. Since the model input contains vector of independent features, the neural network models used for such tasks are composed of fully connected layers. Vanishing gradient and performance degradation issues that arise when using deeper architectures have caused the neural network architectures used for such prediction modeling to be limited in their depth [24, 30, 34, 36, 52]. For instance, Montavon et al. [30] trained a four-layer network on a database of around 7000 organic compounds to predict multiple electronic ground-state and excited-state properties. In the Harvard Energy Clean Project, Pyzer-Knapp et al. [36] used a three-layer network for predicting power conversion efficiency of organic photo-voltaic materials. Zhou et al. [52] used a fully connected network with single hidden layer to predict formation energy from high-dimensional vectors learned using Atom2Vec. ElemNet [24] used a 17-layered architecture to learn formation energy from elemental composition, but experienced performance degradation beyond that depth. Hence, domain scientists have mainly used traditional ML techniques such as Random Forest, Kernel Ridge Regression, Lasso, and Support Vector Machines for materials prediction tasks [12, 14, 29, 47].

Recently, several projects have used domain knowledge-based model engineering within a deep learning context for predictive modeling in materials science [16, 23, 41]. Deep learning was used for directly predicting the crystal orientations of polycrystalline materials from their electron back-scatter diffraction patterns [23]. SchNet [41] used continuous filter convolutional layers to model quantum interactions in molecules for the total energy and interatomic forces that follows fundamental quantum chemical principles. Boomsma and Frellsen [7] introduced the idea of spherical convolution in the context of molecular modelling, by considering structural environments within proteins. Smiles2Vec [16] and CheMixNet [34] have applied deep learning methods to learn molecular properties from the molecular structures of organic materials.

Our goal here is to design a general purpose deep regression network for predicting the properties of inorganic materials from their compositions and/or crystal structures, without using any domain knowledge-based model engineering. We introduce the idea of residual learning to deep regression networks composed of fully connected layers. In a fully connected network, the number of parameters is directly proportional to the product of the number of inputs and the number of output units. Several works have dealt with the performance degradation issue due to vanishing or exploding gradients for other types of data mining problems [18, 19, 43]. Srivastava et al. [43] introduced an LSTM-inspired adaptive gating mechanism that allowed information to flow across layers without attenuation; the gating mechanism required more model parameters. They designed highway networks composed of up to 100 layers that could be optimized. A highway network [43] uses gated connections, which double the number of parameters in a fully connected network. In a DenseNet [19], all previous inputs are combined before being fed into the current layer. For a fully connected network, this approach results in a tremendous increase in the number of model parameters, a particular problem when working with limited GPU memory. He et al. [18] introduced the idea of residual learning, in which a stack of layers learns the residual mapping between the output and input; they built deep CNN models composed of 152 layers for image classification problem. Since the input is added to the residual output, the number of required parameters for residual learning was lower than that in Srivastava et al. [43]. This technique has been used in several CNN and LSTM architectures, with shortcut connections being placed after a stack of multiple CNN or LSTM layers to build deeper networks for better performance [20, 44, 46]. For a fully connected network, an elegant approach is to use the residual mapping approach used in ResNet [18]. However, although residual learning has been widely used in classification networks, no previous work leverages residual learning for building deep regression networks composed of fully connected layers for numerical vector inputs.

In this paper, we study and propose design principles for building deep residual regression networks composed of fully connected layers for data mining problems with numerical vectors as inputs. We introduce a novel deep regression network architecture with individual residual learning (IRNet), in which shortcut connections are placed after each layer such that each layer learns only the residual mapping between its output and input vectors. We compare IRNet against two baseline deep regression networks: and a stacked residual network (SRNet) with shortcut connections after stack of multiple layers. We focus on the *design problem* of learning the formation enthalpy of inorganic materials from an input vector composed of 126 features representing their crystal structure, and another 145 composition-based physical attributes from the OQMD-SC dataset. OQMD-SC contains 435 582 materials with their composition and crystal structure from the Open Quantum Materials Database (OQMD) [27].

Our proposed 48-layered IRNet achieves significantly better performance than does the best state-of-the-art ML approach, Random Forest: a mean absolute error (MAE) of 0.038 eV/atom compared to 0.072 eV/atom on the OQMD-SC dataset. IRNet also performed significantly better than both the plain network and SRNet. The use of individual residual learning (IRNet) led to faster convergence compared to the existing approach of residual learning in SRNet, while maintaining the same number of parameters. We also evaluated IRNet performance for learning materials properties with 145 composition-based physical attributes in two other datasets: OQMD-C (341 443 data points) and MP-C (83 989) [22]. IRNet significantly outperformed the plain network and the traditional ML approach on the new prediction tasks; the deeper models performing better in case of larger dataset (OQMD-C). We performed a combinatorial search for materials discovery using the proposed models. The models were trained on 32 111 entries in OQMD-SC-ICSD dataset. The evaluation was performed by searching for stable materials with specific crystal structures. The proposed model provided significantly more accurate predictions compared to the traditional ML approach (Random Forest).

2 BACKGROUND

2.1 Property Prediction

The prediction of chemical properties from material crystal structure and composition is strongly related to the discovery of new materials. One important material property is formation enthalpy:

the change in energy when one mole of a substance in the standard state (1 atm of pressure and 298.15 K) is formed from its pure elements under the same conditions [33]. In other words, it is the energy released when forming a material (chemical compound) from the constituent elements. By knowing the formation enthalpy, one can know whether the material is stable and thus feasible to experimentally synthesize in laboratory. The more negative the formation enthalpy, the more stable the compound. Materials properties also contain various other properties [22, 27].

2.2 Materials Representation

Most ML approaches require manual feature engineering and a representation that incorporates domain knowledge into model inputs. They thus take composition-based physical attributes and/or crystal structure as the input. Recently, Ward et al. [47] presented a ML framework for formation energy prediction that used an input vector with 145 features computed from composition; stoichiometric attributes, elemental property statistics, electronic structure attributes, and ionic compound attributes. We leverage this approach to compute the 145 physical attributes used in our datasets.

The crystal structure of a material is defined by the shape of the unit cell and associated atom positions, which together define the repeat pattern of the atomic structures that form the material. It is possible to represent the unit cell shape and atom positions as a vector of $3 + 3N$ features (where N is the number of atoms), but this representation is not suitable for ML. The atomic coordinates are not unique—rotating or translating the coordinate system does not change the material—and they do not readily reflect important features of the material (e.g., bond lengths). Many crystal structure representations, such as "bag of bonds" [17] and histograms of bond distances [40], have been developed to address this problem. We use the representation developed by Ward et al. [48], which uses 126 features derived from the Voronoi tessellation of a material. The Voronoi tessellation of a crystal structure provides a clear description of the local environment of each atom, which is used to compute features such as the difference in elemental properties (e.g., molar mass) between an atom and its neighbor [48].

3 DESIGN

We next describe how we build deep residual regression models, composed of multiple fully connected layers, for data mining problems with numerical vectors as inputs. We first introduce a plain network without any residual learning. Next, we build a stacked residual network by introducing shortcut connections for residual learning after each of a number of stacks, each composed of one or more layers with the same configuration. Finally, we introduce our novel individual residual learning approach, in which shortcut connections are used after every layer. We use the plain network and stacked networks later as baseline models for comparison against the individual residual network.

3.1 Plain Network

The model architecture is formed by putting together a series of stacks, each composed of one or more sequences of three basic components with the same configuration. Since the input is a numerical vector, the model uses a fully connected layer as the initial

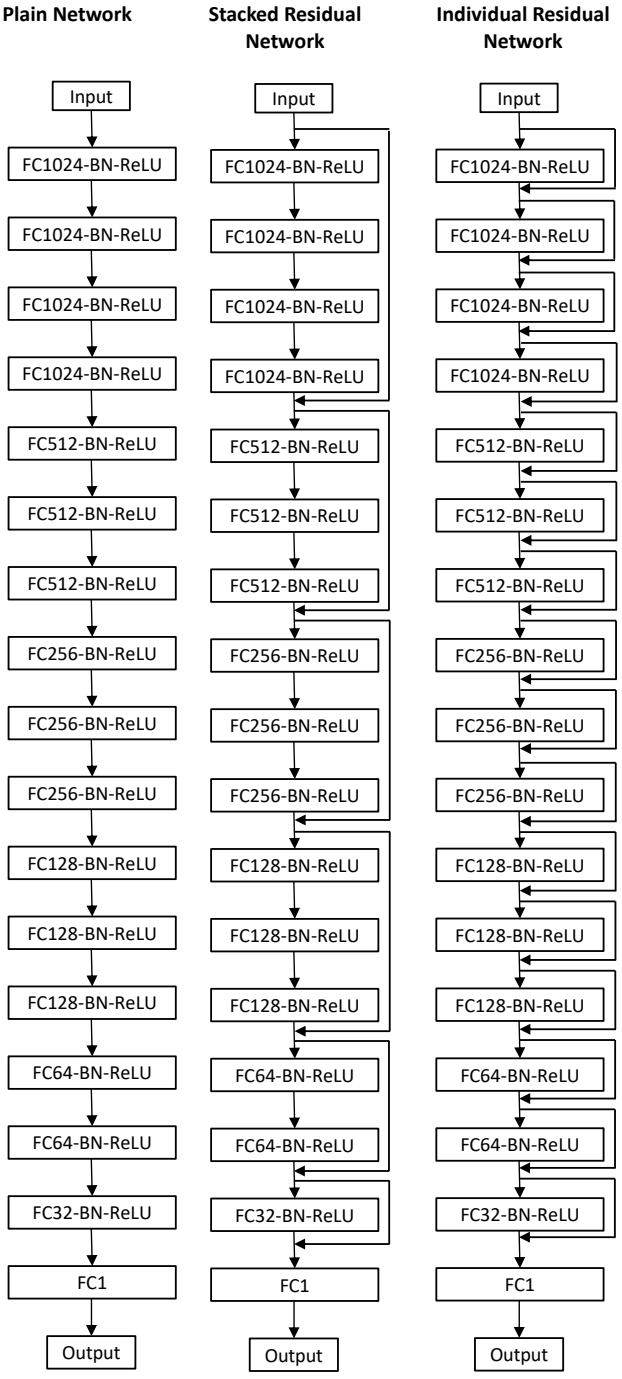


Figure 1: Three types of 17-layer networks. Each "layer" is a fully connected neural network layer with size as described in Table 1; all but the last are followed by batch normalization and ReLU. A *plain network* simply connects the output of each layer to the input of the next. A *stacked residual network* (SRNet) places a shortcut connection after groups of layers called stacks. An *individual residual network* (IRNet) places a shortcut connection after every layer.

Table 1: Detailed configurations for different depths of network architecture. The notation [...] represents a stack of model components, comprising a single (FC: fully connected layer, BN: batch normalization, Re: ReLU activation function) sequence in the case of IRNet and multiple such sequences in the case of SRNet. Each such stack is followed by a shortcut connection.

Output	17-layer SRNet	17-layer IRNet	24-layer SRNet	24-layer IRNet	48-layer SRNet	48-layer IRNet
1024	[FC1024-BN-Re x 4]	[FC1024-BN-Re] x 4	[FC1024-BN-Re x 4]	[FC1024-BN-Re] x 4	[FC1024-BN-Re x 4] x 2	[FC1024-BN-Re] x 8
512	[FC512-BN-Re x 3]	[FC512-BN-Re] x 3	[FC512-BN-Re x 4]	[FC512-BN-Re] x 4	[FC512-BN-Re x 4] x 2	[FC512-BN-Re] x 8
256	[FC256-BN-Re x 3]	[FC256-BN-Re] x 3	[FC256-BN-Re x 4]	[FC256-BN-Re] x 4	[FC256-BN-Re x 4] x 2	[FC1024-BN-Re] x 8
128	[FC128-BN-Re x 3]	[FC128-BN-Re] x 3	[FC128-BN-Re x 4]	[FC128-BN-Re] x 4	[FC128-BN-Re x 4] x 2	[FC128-BN-Re] x 8
64	[FC64-BN-Re x 2]	[FC64-BN-Re] x 2	[FC64-BN-Re x 3]	[FC64-BN-Re] x 3	[FC64-BN-Re x 4] x 2	[FC64-BN-Re] x 8
32	[FC32-BN-Re]	[FC32-BN-Re]	[FC32-BN-Re x 2]	[FC32-BN-Re] x 2	[FC32-BN-Re x 4]	[FC32-BN-Re] x 4
16			[FC16-BN-Re x 2]	[FC16-BN-Re] x 2	[FC16-BN-Re x 3]	[FC16-BN-Re] x 3
1	FC1					

layer in each sequence. Next, to reduce the internal covariance drift for proper gradient flow during back propagation for faster convergence, a batch normalization layer is placed after the fully connected layer [21]. Finally, ReLU [31] is used as the activation function after the batch normalization.

The simplest instantiation of this architecture adds no shortcut connections and thus learns simply the approximate mapping from input to output. We refer to this network as a *plain network*.

3.2 Stacked Residual Learning

Deep neural networks suffer from the vanishing or exploding gradient problem [4, 15], which hampers convergence, and also from the degradation problem: as network depth increases, accuracy becomes saturated and then degrades rapidly. One approach to dealing with these issues is to use shortcut connections for residual learning [18, 19, 43].

Here, we introduce the idea of residual learning to deep regression networks composed of fully connected layers. In a fully connected network the number of parameters is directly proportional to the product of the number of inputs and the number of output units. The gated connection approach from the highway network and the use of all previous inputs from DenseNet [19] would result in a huge increase in model parameters that would not fit in GPU memory. Hence, for a fully connected deep neural network, the residual learning from He et al. [18] is the most elegant approach.

We use stacks of consecutive layers with the same configuration, with the first stack composed of four sequence of layers and the final stack of two sequences. Instead of directly fitting the underlying mapping, the stacked layers explicitly learn the residual mapping. If the underlying mapping is denoted by $H(\mathbf{x})$, the stacked layers fit the residual mapping of $F(\mathbf{x}) = H(\mathbf{x}) - \mathbf{x}$. If the input and output of a stack have the same dimensions, they can be added by using a shortcut connection for residual learning. If the output of a layer, $F(\mathbf{x})$, has a different dimension than the input $\mathbf{x}$, we perform a linear projection W_s to match the dimensions before adding:

$$\mathbf{y} = F(\mathbf{x}) + W_s\mathbf{x}, \tag{1}$$

where $\mathbf{x}$ and $F(\mathbf{x})$ are the input and output to the stack of layers, respectively. W_s acts as a dimension reduction agent. We refer to such a network with shortcut connections across each stack as a *stacked residual network* (SRNet).

3.3 Individual Residual Learning

He et al. [18] introduced the idea of using shortcut connections after a stack composed of multiple convolutional layers. The latest Inception-ResNet [44] architecture for image classification follows a similar approach, with shortcut connections used between stack of Inception-ResNet blocks, where each block is composed of multiple convolutional layers followed by 1×1 convolutional filters for dimension matching. In our case, the stacks are composed of up to four sequences, with each sequence containing a fully connected layer, a batch normalization, and ReLU. Our stacks are comparably more complex and highly non linear when compared to those used in CNN models for image classification. Also, learning the residual regression mapping from input to output vector is comparatively harder than the residual learning for classification task; the activations and gradients can vanish within the stacks.

To solve this issue, we introduce a novel technique of individual residual learning for sequences containing a fully connected layer with batch normalization and non linear activation. We place a shortcut connection after every sequence, so that each sequence needs only to learn the residual mapping between its input and output. This innovation has the effect of making the regression learning task easy. As each "stack" now comprises a single sequence, shortcut connections across each sequence provide a smooth flow of gradients between layers. We refer to such a deep regression network with individual residual learning capability as an *individual residual network* (IRNet).

The detailed architectures for networks with different depths are illustrated in Figure 1 and Table 1. There are several deep network design techniques based on advanced branching techniques such as Inception [44, 45] and ResNext [50], but here our goal is to design a general purpose deep regression network framework rather than optimizing for a specific prediction task. We will explore branching techniques in future work.

4 EMPIRICAL EVALUATION

We now present a detailed analysis of the design and evaluation of our deep regression networks with residual learning. We proceed in three stages. First, we present our evaluation of the proposed deep regression model (IRNet) for the *design problem* and compare its performance with the plain network, SRNet, and traditional ML approaches when applied to the OQMD-SC dataset. Next, we evaluate

the proposed model architecture by learning materials properties from physical attributes for compounds in the OQMD-C and MP-C datasets. Finally, we perform a combinatorial search for materials discovery by training on the OQMD-SC-ICSD dataset. Before presenting our evaluation, we discuss the experimental settings and datasets that we use in this work.

Experimental Settings. We implement the deep learning models with Python and TensorFlow [2]. We performed extensive architecture search and hyperparameter tuning for all deep learning and other ML models used in this study. For deep learning models, we experimented with different activation functions: sigmoid, tanh, and ReLU, both for the intermediate layers and for the final regression layer. We explored learning rates in [1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6]; StochasticGradientDescent, MomentumOptimizer, Adam, and RMSProp optimizers; and mini-batch sizes in [32, 64, 128]. Since we are dealing with regression output, we experimented with mean squared error and mean absolute error as the loss functions. We found the best hyperparameters to be are Adam [26] as the optimizer with a mini batch size of 64, learning rate of 0.0001, mean absolute error as loss function, and ReLU as activation function, with the final regression layer having no activation function. Rather than training the model for a specific number of epochs, we used early stopping with a patience of 200, meaning that we stopped training when the performance did not improve in 200 epochs. For traditional ML models, we used Scikit-learn [35] implementations and employed mean absolute error (MAE) as loss function and error metric.

Datasets. We used four datasets to evaluate our models: OQMD-SC, OQMD-C, MP-C, and OQMD-SC-ICSD. OQMD-SC is composed of 435 582 unique compounds (unique combination of composition and crystal structure) with their DFT-computed formation enthalpy from the Open Quantum Database (OQMD) [27]; this is used for the *design problem*. It is composed of 271 attributes: 125 derived to represent crystal structure using Voronoi tesselations and another 145 physical attributes derived from composition using domain knowledge, as in Ward et al. [47]. OQMD-C is composed of 341 443 compounds with the materials properties from OQMD as of May 2018. MP-C is composed of 83 989 inorganic compounds from the Materials Project database [22] with a set of materials properties as of September 2018. OQMD-C and MP-C contain composition only (no structure information); we compute 145 physical attributes from the composition using Ward et al.'s methods [47]. OQMD-SC-ICSD is composed of entries from the Inorganic Crystal Structure Database (ICSD) [5] present in OQMD-SC. The datasets are randomly split into training and test sets in the ratio of 9:1.

4.1 Design Problem

First, we analyze the impact of different design choices by evaluating the proposed models on the design problem. The design problem involves learning to predict formation enthalpy from input vector composed of 126 attributes to represent crystal structure and 145 physical attributes in OQMD-SC dataset. An extensive architecture search and hyperparameter tuning is performed to search for the best deep regression model for the design problem.

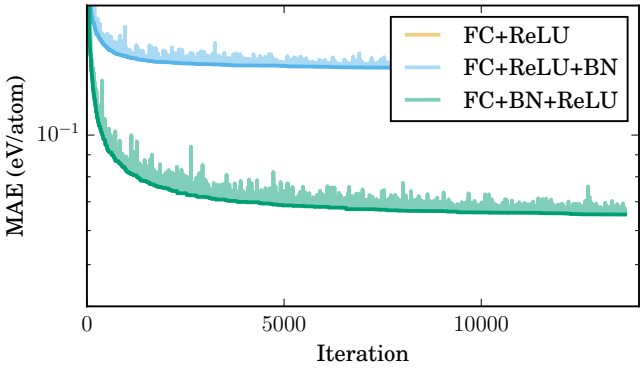


Figure 2: Test error curve for various plain networks for the *design problem*. Batch normalization before activation function (FC+BN+ReLU) improves performance significantly.

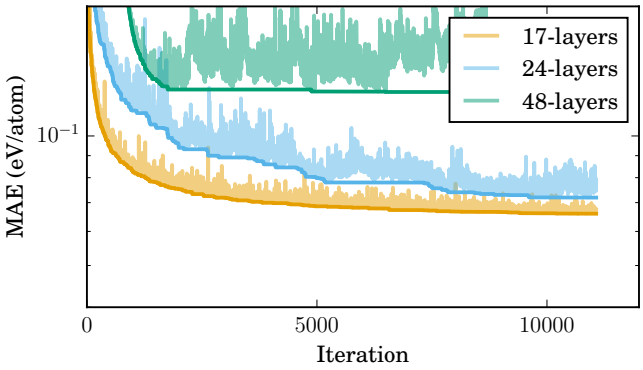
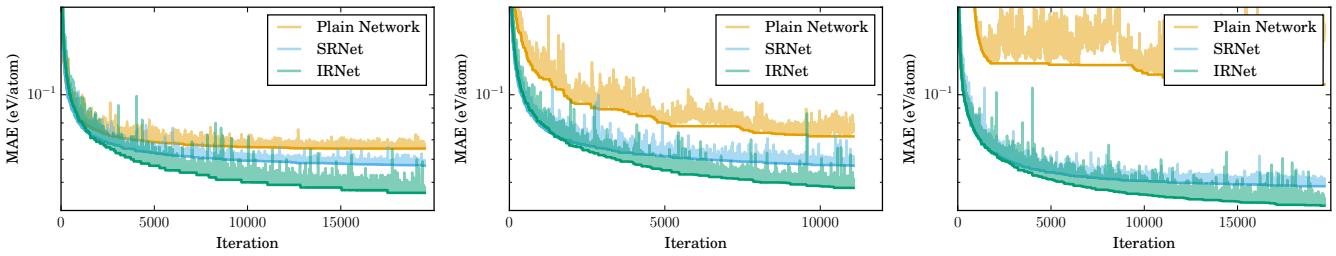


Figure 3: Test error curve for deeper plain networks for the *design problem*. Performance degrades with network depth, even in the presence of batch normalization.

4.1.1 Basic Components. We experimented with different patterns of use of our basic components—fully connected layer, batch normalization, activation function, and dropout—within the plain network. Use of batch normalization resulted in significant reduction in errors, as seen in Figure 2. Batch normalization can be used either before (FC+BN+ReLU) or after the activation function (FC+ReLU+BN). For our regression problem, using batch normalization before ReLU (FC+BN+ReLU) worked better; the original work also used it before the activation function for image classification problem [21]. Since ReLU truncates all negative activations to zero, applying batch normalization on ReLU outputs leads to changes in the activation distribution; since the regression output is dependent on all activations, batch normalization after ReLU leads to higher oscillations and poor convergence.

We also experimented with using dropouts after the first four stacks for better generalization; however, dropouts resulted in slight degradation in the performance. The best plain network architecture for our design problem is composed of 17 sequences containing a fully connected layer, a batch normalization and a ReLU; we refer to this as the *17-layer plain network*. as shown in Figure 1.



(a) 17-layers (b) 24-layers (c) 48-layers

Figure 4: Impact on residual learning for the *design problem*. Both residual networks outperform the plain network, and the individual network outperforms the stacked network for all depths of network. We observe similar trends even in the case of training error curves for all types of networks of all depths; the IRNet converges faster than the SRNet and Plain Network for all depths.

Table 2: Performance of deeper residual networks for the *design problem*. Test errors are MAE in eV/atom. Increased depth of residual network architectures leads to improved performance for both stacked and individual residual networks. The individual residual network (IRNet) clearly outperforms the stacked residual network (SRNet), achieving significantly lower MAE.

Model Type	Plain Network	SRNet	IRNet
17-layer	0.0653	0.0551	**0.0411**
24-layer	0.0719	0.0546	**0.0403**
48-layer	0.1085	0.0471	**0.0382**

4.1.2 Residual Learning. Figure 3 shows how performance can degrade with increased depth for plain networks. This happens mainly because of the vanishing gradient problem. To solve this issue, we introduced residual learning to create SRNet and IRNet, as discussed earlier. We see in Table 2 and Figure 4 that the introduction of shortcut connections to enable residual learning significantly improved model performance, presumably by helping with the smooth flow of gradients from output to input. We compared the individual residual learning in IRNet with the existing approach of use of shortcut connections after stacks of multiple layers in SRNet. The stacks are formed by putting the consecutive layers with equal number of output units in a stack.

We observe a significant benefit from the novel approach of using shortcut connections for individual residual learning in IRNet; the mean absolute error significantly decreased compared to SRNet as seen in Figure 4 and Table 2. Both the training and test error curves in the case of IRNet exhibits better convergence than both SRNet and plain network during the training.We conjecture that learning the residual between the output and the input vector of the sequence is better compared to learning the more complex residual mapping in the case of stacked residual network in SRNet. Also, if the identity mapping using shortcut connections are optimal, the residuals would be pushed to zero and hence, better suited for batch normalization to learn our regression output. This illustrates the advantage of using individual residual learning for deep regression networks composed of fully connected layers for vector inputs.

Table 3: Performance of Traditional ML Approaches for the *design problem*. We performed extensive grid search for hyperparameter tuning for all the listed ML models. Test errors are MAE in eV/atom.

ML Approach	Test Error
AdaBoost	0.479
ElasticNet	0.384
LinearRegression	0.261
Ridge	0.261
SVR	0.243
KNeighbors	0.154
DecisionTree	0.104
Bagging	0.078
RandomForest	**0.072**

4.1.3 Deeper Architectures. Next, we experimented with deeper architectures composed of 24 and 48 sequences of layers for all types of deep regression networks: plain network, SRNet, and IRNet. From Figure 3, we can clearly observe the performance degradation issue in plain networks that do not leverage any shortcut connections for residual network. Figure 4 illustrates the trend in error curves. Although both types of residual networks exhibit reduced test error with increased depth, the rate of reduction for IRNet is significantly better than that for SRNet. To prevent overfitting of such deep models with large numbers of parameters to the training dataset, we used early stopping with a patience of 200. Table 2 shows the final MAE for all types of networks with different depths. Our results illustrates the efficiency of using individual residual learning with deeper architectures.

4.1.4 Comparison with Other ML Approaches. Next, we compared the performance of the proposed deep learning model with traditional ML models: see Table 3. We performed an extensive hyperparameter search to find the best hyperparameters for all ML models. For instance, for Random Forest model, we used a minimum sample split from [5, 10, 15, 20], number of estimators from [100,150,200], maximum features from [0.25, 0.33] and maximum depth from [10,25]. Similarly, extensive grid search for optimization

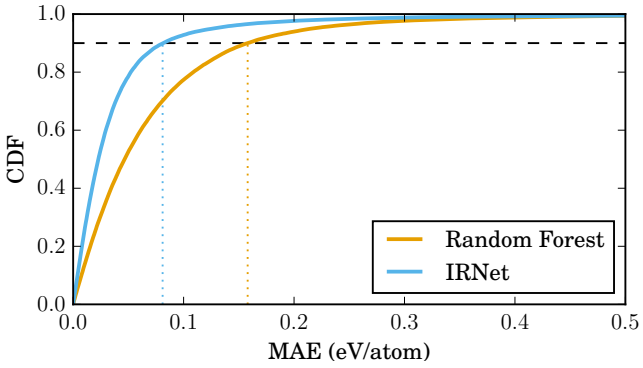


Figure 5: Cumulative distribution function (CDF) of the prediction errors for the *design problem*. Deep learning (IRNet) performs significantly better than the traditional ML approach, Random Forest, achieving a 90th percentile MAE of 0.081 eV/atom vs. 0.158 eV/atom for Random Forest.

of hyperparameters for other ML models are used. Among all of the traditional ML approaches considered, Random Forest achieved the best MAE of 0.072 eV/atom. By comparison, the 48-layer IRNet achieved an MAE of 0.038 eV/atom, significantly outperforming Random Forest for the design problem. Figure 5 illustrates the comparison of the prediction errors for the test set. Deep learning provides a more accurate and robust prediction model than does the state-of-the-art ML approach, Random Forest, predicting the formation enthalpy of 90% of the compounds in the test set with half the error of Random Forest. These results demonstrate that deep learning in general, and IRNet in particular, can help construct a robust model for predicting formation enthalpy from materials crystal structure and composition.

4.1.5 Summary of design insights. We draw the following lessons from our experiments with building deep regression networks for learning regression output from numerical vector inputs.

(1) **Batch Normalization** Batch normalization works better in deep regression networks if used before ReLU. Otherwise, ReLU truncates all negative values to zero, which makes learning the regression output hard. Dropout with batch normalization slightly worsens performance.
(2) **Residual Learning** Residual learning in deep regression always performs better compared to directly learning to fit the underlying mapping from input vector to the regression output.
(3) **Individual Residual Learning** Putting a shortcut connection after each sequence of layers (IRNet) works significantly better than the conventional way of putting the shortcut connection after each stack of multiple layers (SRNet).

The presented architecture can be applied to other data mining problems with vector inputs in scientific domains; they can provide more robust and accurate predictive modeling than the existing ones based on traditional ML approach. The same architecture can be also applied to classification problem by adding a $softmax$ activation at the last layer and using *cross entropy* as the loss function.

4.2 Other Datasets

We evaluated the proposed deep regression architecture on learning materials properties present in two other datasets, OQMD-C and MP-C. OQMD-C is composed of 341 443 samples while MP-C has 83 989 samples; they contain the materials properties with their composition. For comparison, we used the 17-layered plain network and ten other traditional ML approaches. We did not perform hyperparameter tuning and architecture search for deep learning models for these tasks, to illustrate the general purpose use of the proposed deep regression model. The deep regression networks designed for the *design problem* were trained on an input vector containing 145 physical attributes derived from composition; they were trained from scratch using random weights initialization. For the traditional ML models, we performed an extensive grid search for hyperparameter optimization as in the previous case for the design problem.

We can observe three things from the results in Table 4. First, the deep learning network almost always outperforms the traditional ML approaches. Second, the proposed network with individual residual learning performs better than the plain network in all cases. Third, deeper networks worked better in case of OQMD-C while they did not help in case of MP-C, suggesting that deeper networks work better when the dataset size is larger (OQMD-C vs MP-C). This agrees with the fact that deep neural networks perform better with big data. The results demonstrate that although the proposed model was originally designed for a different *design problem*, they almost always outperform the plain network and the traditional ML approaches used by domain scientists. We also experimented with SRNet from design problem for these prediction problems, SRNet performed better than the plain network but worse than the IRNet, similar to the results for the design problem. This illustrate that IRNet can serve as a general purpose deep learning model for different predictive modeling tasks where we need to learn the regression output from an input vector composed of materials composition and/or crystal structures.

4.3 Application for Materials Discovery

Since the proposed model achieved a significant reduction in prediction error for formation enthalpy compared to state-of-the-art approach, it can be applied for high throughput materials discovery. To test the ability of the proposed method to identify new materials, we emulated a common approach in computational materials science, namely combinatorial search . A combinatorial search involves first enumerating all possible combinations of different elements on a specific crystal structure prototype, and then evaluating the stability of each resultant structure with DFT to find which are stable. We performed a combinatorial search using the evaluation settings based on the combinatorial search analysis from [48]. OQMD-SC-ICSD, used as a training set by Ward et al. [48], comprises 32 111 entries in OQMD-SC that correspond to known, experimentally-synthesized materials in ICSD [5]. The proposed IRNet is trained using the OQMD-SC-ICSD dataset and evaluated by predicting the formation enthalpy (stability) of materials with crystal structures from three different, commonly occurring crystal structure types: B2, $L1_0$, and orthorhombically-distorted perovskite. These three structure types were chosen to sample structures with different

Table 4: Performance on OQMD-C and MP-C datasets of our DNN models vs. 10 traditional ML approaches for regression problems: Linear Regression, Lasso, Ridge, Decision Tree, Adaboost, KNeighbors, ElasticNet, SGD Regression, Random Forest and Support Vector, with extensive grid search used to tune hyperparameters for each. Test errors are MAE in eV/atom.

Dataset	Property	Best of 10 ML	17-layer Plain Network	17-layer IRNet	48-layer IRNet
OQMD-C	Formation Enthalpy	0.077	0.072	0.054	**0.048**
	Bandgap	**0.047**	0.052	0.051	**0.047**
	Energy_per_atom	**0.1139**	0.0939	**0.0696**	-
	Volume_pa	0.473	0.0.483	0.415	**0.394**
MP-C	Bandgap	0.4788	0.396	**0.363**	0.364
	Density	0.5052	0.401	**0.348**	0.386
	Energy_above_hull	0.1184	0.098	**0.091**	0.0944
	Energy_per_atom	0.2999	0.175	**0.143**	-
	Total_magnetization	3.232	3.0897	**3.005**	-
	Volume	225.671	219.439	**215.037**	-

Table 5: Performance from combinatorial search. Our 17-layer IRNet, when trained on OQMD-SC-ICSD, predicts formation enthalpy (stability) more accurately than Random Forest for all three types of crystal structures considered.

Crystal Structure	Random Forest MAE (eV/atom)	17-layers IRNet MAE (eV/atom)
B2	0.5114	**0.4780**
$L1_0$	0.4793	**0.4419**
Perovskite	0.6166	**0.3693**

kinds of bonding environments and that are stable with different types of chemistry (e.g., metals vs. oxides).

We show in Table 5 the deep learning model's prediction error for each type of crystal structures. To compare the performance of our deep learning model, we also trained a Random Forest model (the best traditional ML approach from previous analysis) on OQMD-SC-ICSD, with extensive hyperparameter search. Our results demonstrate that our models perform better on the evaluation candidates than does the Random Forest model. Although we do not repeat the entire combinatorial search workflow here with the proposed models, more accurate predictions on the discoveries from Ward et al. [48] suggest that the proposed IRNet model can improve the quality and robustness of the combinatorial search workflow. Despite a small training data size, the IRNet model provides a more robust method for performing combinatorial search for high-throughput materials discovery.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we studied and proposed the design principles for building deep regression networks composed of fully connected layers for data mining problems with numerical vector input. We introduced the use of residual learning in deep regression network; we proposed a deep regression network (IRNet) that leveraged individual residual learning in each layer. The proposed IRNet outperformed the plain network (without residual learning) and traditional machine learning approaches in learning different materials properties from different size of datasets and input vector. For the *design problem* of predicting formation enthalpy from crystal structures and composition, the proposed IRNet significantly reduced the MAE from 0.072 eV/atom to 0.038 eV/atom. We were able to converge the deep regression networks with up to 48 layers, performance increasing with greater depth. Since IRNet kept improving performance with increased depth, we plan to explore deeper IRNet architectures to study their impact on model performance and convergence, and to apply the resulting networks to data mining problems from other scientific domains. It will also be interesting to see how this model performs on experimental datasets using transfer learning from larger simulation datasets. The proposed deep learning model and design insights gained from this work can be used in building predictive models for other applications with vector inputs. The code repository is available at https://github.com/dipendra009/IRNet; we also plan to make the models described in this work available via DLHub [9].

ACKNOWLEDGMENTS

This work was performed under the following financial assistance award 70NANB19H005 from U.S. Department of Commerce, National Institute of Standards and Technology as part of the Center for Hierarchical Materials Design (CHiMaD). Partial support is also acknowledged from DOE awards DE-SC0014330, DE-SC0019358.

REFERENCES

[1] 2016. Materials Genome Initiative. https://www.whitehouse.gov/mgi
[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).
[3] Ankit Agrawal and Alok Choudhary. 2016. Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science. *APL Materials* 4, 5 (2016), 053208.
[4] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–166.
[5] G. Bergerhoff, R. Hundt, R. Sievers, and I. D. Brown. 1983. The inorganic crystal structure data base. *Journal of Chemical Information and Computer Sciences* 23, 2 (1983), 66–69. https://doi.org/10.1021/ci00038a003 arXiv:http://dx.doi.org/10.1021/ci00038a003
[6] B Blaiszik, K Chard, J Pruyne, R Ananthakrishnan, S Tuecke, and I Foster. 2016. The Materials Data Facility: Data services to advance materials science research. *JOM* 68, 8 (2016), 2045–2052.
[7] Wouter Boomsma and Jes Frellsen. 2017. Spherical convolutions and their application in molecular modelling. In *Advances in Neural Information Processing*

Systems. 3433–3443.

[8] Venkatesh Botu and Rampi Ramprasad. 2015. Adaptive machine learning framework to accelerate ab initio molecular dynamics. *International Journal of Quantum Chemistry* 115, 16 (2015), 1074–1083.

[9] Ryan Chard, Zhuozhao Li, Kyle Chard, Logan T. Ward, Yadu N. Babuji, Anna Woodard, Steven Tuecke, Ben Blaiszik, Michael J. Franklin, and Ian T. Foster. 2019. DLHub: Model and data serving for science. In *33rd IEEE International Parallel and Distributed Processing Symposium*.

[10] Stefano Curtarolo, Gus LW Hart, Marco Buongiorno Nardelli, Natalio Mingo, Stefano Sanvito, and Ohad Levy. 2013. The high-throughput highway to computational materials design. *Nature materials* 12, 3 (2013), 191.

[11] Alden Dima, Sunil Bhaskarla, Chandler Becker, Mary Brady, Carelyn Campbell, Philippe Dessauw, Robert Hanisch, Ursula Kattner, Kenneth Kroenlein, Marcus Newrock, et al. 2016. Informatics infrastructure for the Materials Genome Initiative. *JOM* 68, 8 (2016), 2053–2064.

[12] Felix Faber, Alexander Lindmaa, O Anatole von Lilienfeld, and Rickard Armiento. 2015. Crystal structure representations for machine learning models of formation energies. *International Journal of Quantum Chemistry* 115, 16 (2015), 1094–1101.

[13] Felix A Faber, Alexander Lindmaa, O Anatole Von Lilienfeld, and Rickard Armiento. 2016. Machine Learning Energies of 2 Million Elpasolite (A B C 2 D 6) Crystals. *Physical review letters* 117, 13 (2016), 135502.

[14] Luca M Ghiringhelli, Jan Vybiral, Sergey V Levchenko, Claudia Draxl, and Matthias Scheffler. 2015. Big data of materials science: Critical role of the descriptor. *Physical review letters* 114, 10 (2015), 105503.

[15] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 249–256.

[16] Garrett B Goh, Nathan O Hodas, Charles Siegel, and Abhinav Vishnu. 2017. SMILES2Vec: An Interpretable General-Purpose Deep Neural Network for Predicting Chemical Properties. *arXiv preprint arXiv:1712.02034* (2017).

[17] Katja Hansen, Franziska Biegler, Raghunathan Ramakrishnan, Wiktor Pronobis, O. Anatole Von Lilienfeld, Klaus-Robert Robert Müller, and Alexandre Tkatchenko. 2015. Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *The Journal of Physical Chemistry Letters* 6, 12 (jun 2015), 2326–2331. https://doi.org/10.1021/acs.jpclett.5b00831 arXiv:1109.2618

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[19] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[20] Lu Huang, Ji Xu, Jiasong Sun, and Yi Yang. 2017. An improved residual LSTM architecture for acoustic modeling. In *Computer and Communication Systems (ICCCS), 2017 2nd International Conference on*. IEEE, 101–105.

[21] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[22] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin a. Persson. 2013. The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials* 1, 1 (2013), 011002. https://doi.org/10.1063/1.4812323

[23] Dipendra Jha, Saransh Singh, Reda Al-Bahrani, Wei-keng Liao, Alok Choudhary, Marc De Graef, and Ankit Agrawal. 2018. Extracting grain orientations from ebsd patterns of polycrystalline materials using convolutional neural networks. *Microscopy and Microanalysis* 24, 5 (2018), 497–502.

[24] Dipendra Jha, Logan Ward, Arindam Paul, Wei-keng Liao, Alok Choudhary, Chris Wolverton, and Ankit Agrawal. 2018. ElemNet: Deep Learning the Chemistry of Materials From Only Elemental Composition. *Scientific reports* 8, 1 (2018), 17593.

[25] Surya R Kalidindi. 2015. Data science and cyberinfrastructure: critical enablers for accelerated development of hierarchical materials. *International Materials Reviews* 60, 3 (2015), 150–168.

[26] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[27] Scott Kirklin, James E Saal, Bryce Meredig, Alex Thompson, Jeff W Doak, Muratahan Aykol, Stephan Rühl, and Chris Wolverton. 2015. The Open Quantum Materials Database (OQMD): assessing the accuracy of DFT formation energies. *npj Computational Materials* 1 (2015), 15010.

[28] Ruoqian Liu, Abhishek Kumar, Zhengzhang Chen, Ankit Agrawal, Veera Sundararaghavan, and Alok Choudhary. 2015. A predictive machine learning approach for microstructure optimization and materials design. *Scientific reports* 5 (2015).

[29] Bryce Meredig, Ankit Agrawal, Scott Kirklin, James E Saal, JW Doak, A Thompson, Kunpeng Zhang, Alok Choudhary, and Christopher Wolverton. 2014. Combinatorial screening for new materials in unconstrained composition space with machine learning. *Physical Review B* 89, 9 (2014), 094104.

[30] Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole von Lilienfeld. 2013. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics* 15, 9 (2013), 095003.

[31] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 807–814.

[32] N M Nusran, K R Joshi, K Cho, M A Tanatar, W R Meier, S L Bud'ko, P C Canfield, Y Liu, T A Lograsso, and R Prozorov. 2018. Spatially-resolved study of the Meissner effect in superconductors using NV-centers-in-diamond optical magnetometry. *New Journal of Physics* 20, 4 (2018), 043010. http://stacks.iop.org/1367-2630/20/i=4/a=043010

[33] David W Oxtoby, H Pat Gillis, and Laurie J Butler. 2015. *Principles of modern chemistry*. Cengage Learning.

[34] Arindam Paul, Dipendra Jha, Reda Al-Bahrani, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. 2018. CheMixNet: Mixed DNN Architectures for Predicting Chemical Properties using Multiple Molecular Representations. In *Proceedings of the Workshop on Molecules and Materials at the 32nd Conference on Neural Information Processing Systems*.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[36] Edward O Pyzer-Knapp, Kewei Li, and Alan Aspuru-Guzik. 2015. Learning from the harvard clean energy project: The use of neural networks to accelerate materials discovery. *Advanced Functional Materials* 25, 41 (2015), 6495–6502.

[37] Zhao Qin, Gang Seob Jung, Min Jeong Kang, and Markus J Buehler. 2017. The mechanics and design of a lightweight three-dimensional graphene assembly. *Science advances* 3, 1 (2017), e1601536.

[38] Krishna Rajan. 2015. Materials informatics: The materials "gene" and big data. *Annual Review of Materials Research* 45 (2015), 153–169.

[39] Rampi Ramprasad, Rohit Batra, Ghanshyam Pilania, Arun Mannodi-Kanakkithodi, and Chiho Kim. 2017. Machine learning in materials informatics: recent applications and prospects. *npj Computational Materials* 3, 1 (dec 2017), 54. https://doi.org/10.1038/s41524-017-0056-5

[40] KT Schütt, H Glawe, F Brockherde, A Sanna, KR Müller, and EKU Gross. 2014. How to represent crystal structures for machine learning: Towards fast prediction of electronic properties. *Physical Review B* 89, 20 (2014), 205118.

[41] Kristof T. Schütt, Huziel E. Sauceda, Pieter-Jan Kindermans, Alexandre Tkatchenko, and Klaus-Robert Müller. 2017. SchNet - a deep learning architecture for molecules and materials. (2017), 1–10. arXiv:1712.06113 http://arxiv.org/abs/1712.06113

[42] Atsuto Seko, Hiroyuki Hayashi, Keita Nakayama, Akira Takahashi, and Isao Tanaka. 2017. Representation of compounds for machine-learning prediction of physical properties. *Physical Review B* 95, 14 (2017), 144110.

[43] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In *Advances in neural information processing systems*. 2377–2385.

[44] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning.. In *AAAI*, Vol. 4. 12.

[45] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1–9.

[46] Yiren Wang and Fei Tian. 2016. Recurrent residual learning for sequence classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 938–943.

[47] Logan Ward, Ankit Agrawal, Alok Choudhary, and Christopher Wolverton. 2016. A General-Purpose Machine Learning Framework for Predicting Properties of Inorganic Materials. *npj Computational Materials* 2, August (2016), 16028. https://doi.org/10.1038/npjcompumats.2016.28 arXiv:1606.09551

[48] Logan Ward, Ruoqian Liu, Amar Krishna, Vinay I Hegde, Ankit Agrawal, Alok Choudhary, and Chris Wolverton. 2017. Including crystal structure attributes in machine learning models of formation energies via Voronoi tessellations. *Physical Review B* 96, 2 (2017), 024104.

[49] Logan Ward and Chris Wolverton. 2016. Atomistic calculations and materials informatics: A review. *Current Opinion in Solid State and Materials Science* (2016).

[50] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 5987–5995.

[51] Dezhen Xue, Prasanna V Balachandran, John Hogden, James Theiler, Deqing Xue, and Turab Lookman. 2016. Accelerated search for materials with targeted properties by adaptive design. *Nature communications* 7 (2016).

[52] Quan Zhou, Peizhe Tang, Shenxiu Liu, Jinbo Pan, Qimin Yan, and Shou-Cheng Zhang. 2018. Learning atoms for materials discovery. *Proceedings of the National Academy of Sciences* 115, 28 (2018), E6411–E6417.