# BRNet: Branched Residual Network for Fast and Accurate Predictive Modeling of Materials Properties

Vishu Gupta* Wei-keng Liao* Alok Choudhary* Ankit Agrawal*

**Abstract**

Machine Learning (ML) and Deep Learning (DL) have become increasingly popular in the field of materials science for building property prediction models owing to their ability to efficiently extract and understand data-driven relationships between materials composition, structure, and properties. In general, materials property prediction are regression problems with a vector-based input material representation. While fully connected layers have been widely used in deep neural networks to predict materials properties, simply adding more and more layers to create a deep model often degrades their performance due to the vanishing gradient problem, thereby limiting usage. In this paper, we study and propose architectural principles for building deep regression neural networks comprising fully connected layers with numerical vectors that bypass manual feature engineering. We introduce a novel deep regression neural network with branched residual learning, BRNet, consisting of branching of layers to maximize variation of features learned from the input or previous layer and places skip connections after each layer to minimize the information loss due to vanishing gradient. We perform BRNet model training for inorganic material properties using numerical vectors representing the elemental fractions of the compositions of the respective materials and compare its performance against other traditional ML and DL techniques, including ElemNet and IRNet. Using multiple datasets (such as OQMD, MP, JARVIS) for training and testing, we show that BRNet models are significantly more accurate than the state-of-the-art ML methods and DL models for all data sizes by using only raw elemental fractions as input. We also show that BRNet's branched residual learning requires fewer parameters and leads to better convergence during the training phase than other neural networks, thus resulting in faster model training.

## 1 Introduction

Artificial Intelligence (AI) and Machine Learning (ML) has become increasingly popular in the field of materials science, which has greatly enhanced property prediction and materials discovery [22,25]. The availability of large scale datasets through high throughput density functional theory (DFT) calculations [4, 5, 13, 17] and the ease to access and analyze them by using various data infrastructure and mining tools [11, 37] has led to the novel paradigm of materials informatics [1, 2, 26, 36] and has helped materials scientists better understand materials and predict their properties by using advanced data-driven ML techniques [7, 26, 30, 34, 40]. We generally perform regression-based predicting modeling of the properties in materials science by using various numerical features derived from domain knowledge [7, 15, 26, 27, 30, 34, 40], such as composition-based and structure-based features, as input for the predictive model. Since the input representation of materials is usually a 1D numerical vector, neural networks based deep learning (DL) models composed of fully connected layers are widely used to perform the regression, in addition to traditional ML algorithms such as Random Forest and Support Vector Machines [6, 21, 26, 30, 34, 40]. However, we observe performance degradation due to the vanishing gradient problem when deeper neural network architectures are used for predictive modeling, thereby limiting the depth, i.e., the number of layers that can be used to create the model [15, 24, 41]. Some research performed domain knowledge-based model engineering within a deep learning context in materials science for predictive modeling [14, 24, 28]. There also have been several efforts to learn either the atomic interaction or the material embeddings using graph-based networks from the crystal structure [23, 29, 39].

Our goal for this work is to design a deep regression network that can maximize the predictive ability of the model by using the available resources and predict materials properties by using only elemental fraction as input for the model, which can be calculated without the need for any domain knowledge. We introduce the idea of deep residual learning with branched architecture for deep regression networks composed of fully connected layers. Several works have tried dealing with the vanishing/exploding gradient-based performance degradation issue [10, 12, 32]. Highway network introduced in [32] has an LSTM-inspired adaptive gated connection mechanism that allows information to flow across layers, pre-

*ECE Department, Northwestern University, Evanston, Illinois, USA, vishugupta2020@u.northwestern.edu, {wkliao, choudhar, ankitag}@eecs.northwestern.edu

venting the loss of information. This network consists of up to 100 layers that can be optimized but double the number of parameters in the fully connected network due to the gating connection mechanism. He et al. [10] introduced stacked residual learning to learn the mapping between the output and input and built deep CNN models composed of 152 layers for the image classification problem. This idea reduced the number of parameters compared to Srivastava et al. [32] as it adds input to the residual output. An elegant approach for a fully connected network is to use the residual mapping approach as used in ResNet [10]. Jha et al. introduced individual residual network (IRNet) [16] using deep regression networks composed of fully connected layers and batch normalization using numerical vector inputs to predict materials properties. However, the batch normalization layer increases the number of parameters and training time, making it difficult to use with limited GPU memory. There have been several works where a branched structure was used to design the neural network's architecture [19,38]. However, although residual learning and branched network architecture have been widely used in classification networks in different works, no previous work leverages both residual learning and branched structure network architecture together by considering both time and cost effectiveness for building deep regression neural networks composed of fully connected layers for numerical vector inputs.

In this paper, we analyze and propose design principles for building time and parameter efficient deep regression networks composed of fully connected layers when using numerical vectors as inputs. We propose a novel deep regression network architecture that consists of branched residual learning (BRNet), where we combine the branched structure with skip connections after each layer to learn the mapping between output and input by minimizing the loss. We compare BRNet against multiple baseline deep regression networks: ElemNet with dropout at variable intervals of fully connected layers, branch network (BNet) with branching at the initial layers of the network, and individual residual network (IRNet) with shortcut connections and batch normalization after each layer. We focus on the design problem for predicting the formation enthalpy of inorganic materials from an input vector composed of 86 features representing composition-based elemental fraction from the OQMD. OQMD contains 345,220 unique materials from the Open Quantum Materials Database (OQMD) [17] as of January 2020. We also perform model training for predicting the formation enthalpy on MP and JARVIS datasets to analyze the effect of different data sizes. MP contains 89,181 unique materials from the Materials Project Database [13] as of January 2021, and JARVIS consists of 20,072 unique compounds from the Joint Automated Repository for Various Integrated Simulations Database [4] as of July 2020.

Our proposed BRNet architecture achieves significantly better accuracy than the best state-of-the-art ML approach using AutoML: a mean absolute error (MAE) of 0.041 eV/atom compared to 0.149 eV/atom on the OQMD dataset. BRNet and BNet also performed better than ElemNet and IRNet for the design problem. The use of branching and skip connections in branched residual learning (BRNet) led to faster convergence than the existing approach of residual learning in IRNet, while reducing the number of model parameters significantly. We also evaluated BRNet performance learning for other materials properties in OQMD, MP, and JARVIS dataset and found that BRNet consistently outperforms the networks with and without skip connections and the traditional ML approach on the prediction tasks. We also performed an in-depth architectural search on BRNet to extract design principles for optimal branching conditions based on location and distribution for a fixed number of network parameters by evaluating all possible combinations of branching with simple branching configuration. Finally, we emulated combinatorial search without using crystal attributes for materials discovery using the proposed models. The model training was performed using the OQMD-ICSD dataset with 32,111 entries and evaluated by searching for stable materials with specific crystal structures. Overall, the proposed BRNet model provided more accurate predictions than the traditional ML and DL approaches on datasets of different sizes and is expected to be widely useful for fast and accurate predictive modeling of materials properties and to accelerate materials discovery.

## 2 Background

**2.1 Materials Property** The prediction of various materials properties from composition-based elemental fractions can be strongly related to discovering new materials as it can automatically capture the chemical interactions and similarities between different elements effectively [15]. One of the important material properties is formation enthalpy (also known as formation energy) which is defined as the change in energy when one mole of a chemical compound in the standard state (1 atm of pressure and 298.15 K) is formed from its pure elements under the same conditions i.e. the energy released when forming a material from the constituent elements. The knowledge of the formation enthalpy can help materials scientists determine the stability of the material if it were to be experimentally synthesized in the laboratory. When more energy is produced in bond formation than needed for bond breaking then we have

negative formation enthalpy and more negative formation enthalpy leads to more stable material. Materials databases typically also record various other properties such as volume, density, band gap, etc. [4, 13, 17].

**2.2 Materials Representation** Most traditional ML approaches require manually engineered materials representation incorporating domain knowledge as inputs to the model [9, 20, 31, 35]. However, these studies emphasize more on the importance of the manual or domain knowledge-based feature engineering for training models for a specific materials property, and less on the generalizability of the solution. Hence, in this work, we use raw elemental fractions (EF) as input, as they are simple to calculate and have shown remarkable promise using powerful DL techniques that have consistently shown to excel on raw inputs [15]. EF representation used in this work is composed of 86 composition based-attributes where each attribute represents an element in the periodic table. For example, NaCl is represented as a 1D vector of 86 numbers, with the Na and Cl columns containing 0.5 and everything else as 0. Leveraging elemental fractions as an input for the model not only simplifies the input for the different properties in this work, thereby demonstrating good generalization, but is also expected to facilitate the development of such models on other datasets and properties in the future.

## 3 Design

We next describe the deep residual regression models used in this work and how we build them using different types of neural network components. Here, we introduce our novel branch residual network (BRNet) made from branch network (BNet), which consists of skip connections added after every layer to fully connected layers stacked sequentially with a branched structure for the initial layers. We use the ElemNet [15], IRNet [16] and different architectural combinations as baseline models for comparison against BNet and BRNet. ElemNet [15], a network with fully connected layers with dropout at variable intervals and, IRNet [16], a network with fully connected layers and batch normalization with skip connections after each layer are pre-existing deep neural networks commonly used for materials property prediction.

**3.1 Branch Residual Learning** He et al. [10] introduced the idea of using skip connections after a stack composed of multiple convolutional layers. Work in [19, 33, 38] uses branched architecture for image classification. In our case, the stacks are composed of fully connected layers, Leaky ReLU and skip connections, which is highly non-linear compared to image classification problem with CNN models. Also, learning the mapping from input to output in regression tasks is comparatively more challenging than the residual learning for classification tasks.

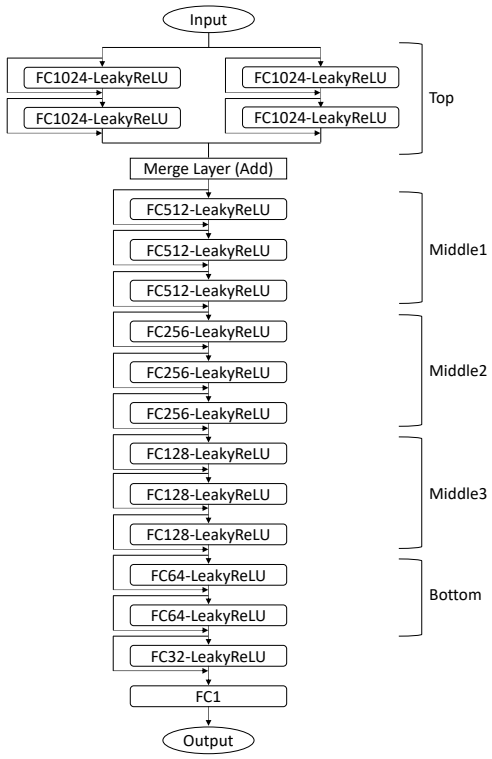



Figure 1: Proposed Branch Residual Network (BRNet). Each "layer" is a fully-connected neural network layer with size depicted in each of the blocks followed by Leaky-ReLU. BRNet has a branched structure in the beginning to learn diverse features, with each branch consisting of layers with the same number of neurons, and places a shortcut connection after every layer.

To address this challenge we first introduce a novel neural network architecture which comprises of a series of stacks, each composed of a fully connected layer and Leaky ReLU with a branched structure in the initial layers to build a neural network which we refer to as branched network (BNet). Next, we use BNet as our base network and add skip connections after each stack to introduce a neural network which we refer to as branched residual network (BRNet). The detailed architecture for the network is illustrated in Figure 1. As our goal is to design a general purpose network framework for regression task rather than optimizing the network for a specific task for prediction as in [33, 38], we will explore more sophisticated branching in future work.

## 4 Empirical Evaluation

We present a detailed analysis of the design and evaluation of the proposed branched deep regression network with skip connections. We will proceed in several steps. First, we perform our evaluation of the proposed branched deep regression model BNet and BRNet for the design problem and compare its performance against ElemNet, IRNet and traditional ML approaches when applied to the OQMD dataset. Next, we evaluate the proposed model architecture by performing model training for different materials properties for compounds in the OQMD, MP and JARVIS datasets using elemental fractions as model input. Finally, we perform time and computational parameter evaluation for different models by training on different datasets to predict formation enthalpy. Before presenting the results, we discuss the experimental settings and datasets used in this work.

**4.1 Experimental Settings** We implement deep learning (DL) models with Python using Keras framework. We performed extensive hyperparameter tuning and architecture search for all DL and other ML models used in this study. For traditional ML models, we used Scikit-learn implementations and used mean absolute error (MAE) as loss function and the error metric. We also used an AutoML library called as hyperopt sklearn [18] to find the best performing ML model. For DL models, we performed the experiments with different activation functions for all the intermediate regression layers. We explored learning rates, mini-batch sizes and different optimizers. We also experimented with different loss functions for our regression problem.

**4.2 Datasets** In this work, we used four datasets to evaluate our models: OQMD [17], MP [13], JARVIS [4] and OQMD-ICSD [3, 17]. Detailed description of datasets can be found in supplementary information. Each property value corresponds to the lowest formation enthalpy among all compounds with the same composition in each of the datasets, representing its most stable crystal structure. The datasets are randomly split with stratification based on the number of elements in a compound (to make the model train, validate, and test on the same proportion of compound with variable no. of elements) and a fixed random seed of 1234567 into training, validation, and test sets in the ratio of 81:9:10.

**4.3 Design Problem** First, we analyze the impact of different architectural choices by evaluating the proposed models on the design problem. We perform formation enthalpy prediction from an input vector composed of 86 composition-based elemental fractions using the OQMD, MP and JARVIS datasets for the design problem. An extensive architecture search and hyperparameter tuning is performed to search for the best deep regression model for the design problem.

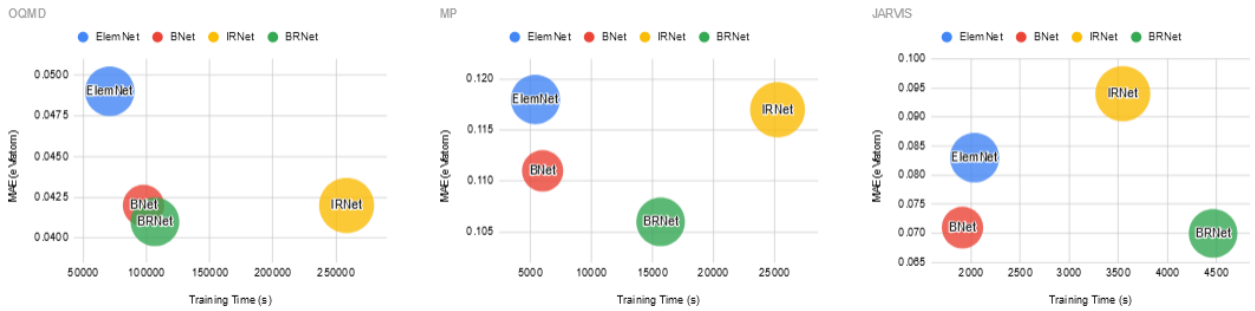
Table 1: Prediction performance benchmarking for the prediction task of "Optimal Architecture Search" for the design problem. We show the comparison in performance between the proposed models and existing models.

| Dataset (Size) | #Model | Test MAE | Training Time (s) | #Model Parameters |
|---|---|---|---|---|
| OQMD (345,134) | ElemNet | 0.049 | 70,851 | 4,631,361 |
| | BNet | 0.042 | 97,587 | 3,670,849 |
| | IRNet | 0.042 | 258,367 | 5,461,473 |
| | BRNet | **0.041** | 106,658 | 4,548,385 |
| MP (89,181) | ElemNet | 0.118 | 5,428 | 4,631,361 |
| | BNet | 0.112 | 8,367 | 3,670,849 |
| | IRNet | 0.117 | 25,249 | 5,461,473 |
| | BRNet | **0.106** | 15,677 | 4,548,385 |
| JARVIS (19,994) | ElemNet | 0.083 | 2,036 | 4,631,361 |
| | BNet | 0.071 | 1,913 | 3,670,849 |
| | IRNet | 0.094 | 3,547 | 5,461,473 |
| | BRNet | **0.070** | 4,467 | 4,548,385 |

**4.3.1 Basic Components** We experimented with different components of the neural network architecture to narrow down the vast space of possible architectures. Here, we use the base architecture consisting of 17 stacks of fully connected layers, which is same as what was used in ElemNet [15] and IRNet [16], in order to facilitate comparison. Detailed description of model components used for the analysis can be found in supplementary information. After initial exploration, we narrowed our hyperparameters for the neural network to be Adam as the optimizer, ReLU and LeakyReLU as activation function (and final regression layer with linear or no activation function), mini-batch size as 32, learning rate as 0.0001, mean absolute error as loss function.

**4.3.2 Optimal Architecture Search** Next, we experimented with different combinations of architectural components selected in the previous analysis for neural network to find the optimal architecture for the neural network which increases the accuracy and decreases the training time of the model. We explored different combinations of five architectural components: Activation Function (ReLU or LeakyReLU), Drop Out, Batch Normalization, Skip Connection, and Branching, where the neural network can have one or more of these architectural components. For Drop Out and Batch Normalization we use the same values and layer arrangement as ElemNet and IRNet respectively. We did not perform experiments using both Drop Out and Batch Normalization within the same neural network as it does not perform well as shown in [16].

Table 1 and Figure 2 illustrate that the neural network consisting of Branching, Skip Connection and




(a) (b) (c)

Figure 2: The bubble charts indicate the performance of the DL models based on the training time on the x-axis, MAE (eV/atom) on the y-axis and model parameters as the bubble size for (a) OQMD (b) MP and (c) JARVIS.

Leaky ReLU as the activation function performs the best. The detailed results on the analysis can be found in Table S1. We see that the novel approach of skip connections with branching significantly improved model performance by creating a smooth flow of gradients from output to input and decreasing the model parameters and convergence time. The use of dropout does not contribute much towards improving the model's accuracy. As for batch normalization, its use in the neural network resulted in a significant increase in the training time and the model parameters. Although earlier work uses batch normalization for image classification problems to produce better results, we find here that it is better not to use it in the neural network for regression problems, as was also observed in [8]. Hence, our design problem's best architecture consists of 17 stacks containing fully connected layers, branching, skip connections and Leaky ReLU as activation for the initial layers. We refer to this as the 17-layer branched residual network, as shown in Figure 1.

**4.3.3 Optimal Branching Search** Till now, we have presented the results of neural networks that consist of branching at the initial layers with an equally distributed two-branched structure. Here, we perform two separate analysis with variation in the branching by changing the branching location and number of branches by keeping the number of parameters constant to find the optimal configuration for branching.

Table 2 shows the performance of BRNet with variation in the branching based on location and distribution of the layers for branching. Here, we perform the distribution of the branching by fixing a location as depicted in Figure 1. Here we limit our analysis to a single occurrence of branching, which can be configured at multiple locations in various distributions. More sophisticated branching with simultaneous multiple branching at different locations is not analyzed in this work. From Table 2, we can observe that changing the configuration of the branching does not significantly vary the performance of the model for large datasets. For small datasets, the variation in model accuracy is somewhat higher, which is not surprising. We also observe that placing the branching structure at the later layers of the network or for the layers with small number of neurons and increasing the number of branches by keeping the number of model parameters constant increases the number of epochs it takes to train the model for most of the cases. Overall, we do not observe a significant

Table 2: Prediction performance benchmarking for the prediction task "Optimal Branching Search" for the design problem. We analyze the effect of branching by changing the location and distribution of branching.

| Dataset (Size) | Branching Location | Branching Distribution | Test MAE | #Epochs |
|---|---|---|---|---|
| OQMD (345,134) | Top | [2, 2] | 0.041 | 954 |
| | | [3, 1] | 0.043 | 714 |
| | | [2, 1, 1] | 0.041 | 960 |
| | | [1, 1, 1, 1] | 0.041 | 721 |
| | Middle 1 | [2, 1] | 0.041 | 1057 |
| | | [1, 1, 1] | 0.041 | 770 |
| | Middle 2 | [2, 1] | 0.040 | 1442 |
| | | [1, 1, 1] | 0.042 | 682 |
| | Middle 3 | [2, 1] | 0.041 | 1501 |
| | | [1, 1, 1] | 0.041 | 787 |
| | Bottom | [1, 1] | 0.041 | 1233 |
| MP (89,181) | Top | [2, 2] | 0.106 | 514 |
| | | [3, 1] | 0.113 | 458 |
| | | [2, 1, 1] | 0.107 | 501 |
| | | [1, 1, 1, 1] | 0.108 | 659 |
| | Middle 1 | [2, 1] | 0.113 | 824 |
| | | [1, 1, 1] | 0.111 | 581 |
| | Middle 2 | [2, 1] | 0.111 | 1557 |
| | | [1, 1, 1] | 0.113 | 493 |
| | Middle 3 | [2, 1] | 0.108 | 1007 |
| | | [1, 1, 1] | 0.114 | 316 |
| | Bottom | [1, 1] | 0.112 | 672 |
| JARVIS (19,994) | Top | [2, 2] | 0.070 | 471 |
| | | [3, 1] | 0.069 | 710 |
| | | [2, 1, 1] | 0.068 | 478 |
| | | [1, 1, 1, 1] | 0.069 | 556 |
| | Middle 1 | [2, 1] | 0.068 | 1488 |
| | | [1, 1, 1] | 0.070 | 564 |
| | Middle 2 | [2, 1] | 0.069 | 1102 |
| | | [1, 1, 1] | 0.071 | 320 |
| | Middle 3 | [2, 1] | 0.069 | 854 |
| | | [1, 1, 1] | 0.070 | 466 |
| | Bottom | [1, 1] | 0.067 | 780 |

variation in model accuracy by changing the location and distribution of branching. Since evenly distributed branching at the initial layers consistently gives the best or near-best performing models both in terms of accuracy and time, we use the same as the default configuration for branching in rest of this paper.

#### 4.3.4 Comparison with Other ML Approaches

Next, we compare the performance of the proposed deep learning models with traditional ML models, which are shown in Table 3.

Table 3: Prediction performance benchmarking for the prediction task of "Comparison with Other ML Approaches". We benchmark the proposed BRNet architecture against all the best traditional ML models with extensive grid search for hyperparameter tuning and AutoML. The existing DL models (ElemNet and IRNet) are also included in the table for reference.

| Dataset (Size) | Model | Test MAE |
|---|---|---|
| OQMD (345,134) | Trad. ML | 0.166 |
| | AutoML | 0.149 |
| | ElemNet | 0.049 |
| | IRnet | 0.042 |
| | BRNet | **0.041** |
| MP (89,181) | Trad. ML | 0.178 |
| | AutoML | 0.167 |
| | ElemNet | 0.118 |
| | IRnet | 0.117 |
| | BRNet | **0.106** |
| JARVIS (19,994) | Trad. ML | 0.135 |
| | AutoML | 0.129 |
| | ElemNet | 0.083 |
| | IRnet | 0.094 |
| | BRNet | **0.070** |

To find the best MAE for the traditional ML models, we performed an extensive hyperparameter search to optimize hyperparameters for all the models as given in Table S2 as well as used AutoML. Among all of the traditional ML approaches we performed to create the model, AutoML achieved the best MAE of 0.149eV/atom (OQMD), 0.167eV/atom (MP) and 0.129eV/atom (JARVIS). On the other hand, BRNet achieved an MAE of 0.041eV/atom (OQMD), 0.106eV/atom (MP) and 0.07eV/atom (JARVIS), thereby significantly outperforming AutoML for the design problem. Our proposed deep learning models provide a more robust and accurate prediction model than the state-of-the-art ML approaches while predicting the formation enthalpy, reducing the test error by more than 70% for the largest dataset (OQMD) for the design problem. These results demonstrate that deep learning, along with our proposed BRNet model, can help construct a robust model for predicting formation enthalpy from materials composition alone.

**4.4 Summary of design insights** We derive the following insights from our experiments with different deep learning architectures for performing deep regression from numerical vector inputs for materials properties.

- **Basic Components** We observe that not using batch normalization and dropout performs the best for performing regression for predicting materials properties. It is also advisable to use Adam as the optimization function and LeakyReLU as the activation function as the basic components for the neural network.
- **Residual Learning** Residual learning using skip connections to perform deep regression for predicting materials properties always helps improve the predictive ability of the model compared to performing model training without skip connections, as it helps the model to fit the underlying mapping from the input vector to the regression output without suffering from the vanishing gradient problem.
- **Branching Structure** Branching the architecture of the neural network helps improve the regression model by increasing the accuracy of the model and decreasing the training time and number of parameters of the model as compared to not using branching. Branching when combined with skip connections can help build a more robust model.

We believe that the proposed BRnet model can help build more accurate and robust predictive models than traditional ML/DL approaches for other data mining problems as well that use a numerical vector as input. BRNet can also be easily adapted to perform classification by modifying the architecture, i.e., using softmax as the activation of the last layer and cross-entropy as the loss function.

**4.5 Other Materials Properties** We evaluated the proposed deep neural network architecture for learning and predicting other materials properties present in OQMD, MP and JARVIS datasets. We use the ElemNet, IRNet, and AutoML for comparison. Architecture search and hyperparameter tuning were not performed for BRNet here to evaluate and illustrate its effectiveness and generalization.

From Table 4, we can observe the following things. First, the deep learning framework outperforms AutoML for almost all the materials properties. Second, the proposed branched network performs better than all the other deep neural networks used for comparison. Third, as we can see that for the same materials property across different datasets, the accuracy of the

Table 4: Prediction performance benchmarking for the prediction task of "Other Materials Properties". We benchmark the proposed BRNet model against AutoML for ML model and ElemNet, IRNet, BNet model for DL model.

| Dataset | Property (Size) | Model | Test MAE |
|---|---|---|---|
| OQMD | Volume (345,134) | AutoML | 21.02 |
| | | ElemNet | 19.56 |
| | | IRNet | 20.09 |
| | | BNet | 17.92 |
| | | BRNet | **16.91** |
| | Band Gap (345,134) | AutoML | 0.075 |
| | | ElemNet | 0.143 |
| | | IRNet | 0.054 |
| | | BNet | 0.050 |
| | | BRNet | **0.048** |
| | Stability (345,134) | AutoML | 0.113 |
| | | ElemNet | 0.051 |
| | | IRNet | 0.047 |
| | | BNet | 0.045 |
| | | BRNet | **0.043** |
| MP | Volume (89181) | AutoML | **205.9** |
| | | ElemNet | 248.8 |
| | | IRNet | 238.6 |
| | | BNet | 233.8 |
| | | BRNet | 227.4 |
| | Density (89181) | AutoML | 0.446 |
| | | ElemNet | 0.373 |
| | | IRNet | 0.373 |
| | | BNet | 0.349 |
| | | BRNet | **0.344** |
| | Band Gap (89181) | AutoML | 0.435 |
| | | ElemNet | 0.342 |
| | | IRNet | 0.316 |
| | | BNet | 0.317 |
| | | BRNet | **0.315** |
| JARVIS | Band Gap Optb (17,924) | AutoML | 0.345 |
| | | ElemNet | 0.306 |
| | | IRNet | 0.301 |
| | | BNet | 0.267 |
| | | BRNet | **0.257** |
| | Bulk Modulus (8,199) | AutoML | 13.46 |
| | | ElemNet | 11.40 |
| | | IRNet | 11.37 |
| | | BNet | 11.10 |
| | | BRNet | **10.45** |
| | Shear Modulus (8,199) | AutoML | 10.75 |
| | | ElemNet | 10.51 |
| | | IRNet | 10.87 |
| | | BNet | 11.01 |
| | | BRNet | **9.88** |
| | Band Gap Mbj (5,287) | AutoML | 0.630 |
| | | ElemNet | 0.544 |
| | | IRNet | 0.551 |
| | | BNet | 0.462 |
| | | BRNet | **0.453** |

deep learning framework decreases with the decrease in the size of the dataset, which is consistent with the general observation that deep learning performs better with bigger datasets. The results demonstrate that our proposed BRNet architecture which was originally modeled for the design problem, performs better than other DL and ML models used by domain experts for multiple materials properties across different materials databases. These results illustrate that BRNet can serve as a general-purpose deep learning architecture for different predictive modeling problems in materials science, and possibly other domains as well.

**4.6 Application to Materials Discovery** Since the proposed model significantly improved the predictive performance for most materials properties across datasets of various sizes compared to state-of-the-art ML approaches and well-known DL models, we can apply it for high throughput materials discovery. To test the ability of the proposed model architecture to identify new materials, we emulated combinatorial search without using crystal attributes as done in [16, 35]. We use the OQMD-ICSD dataset as a training set and evaluate the models by predicting the formation enthalpy (stability) of materials from three different datasets of commonly occurring crystal structure types: orthorhombically-distorted perovskite, $L1_0$ and B2.

Table 5: Prediction performance benchmarking for the prediction task of "Application for Materials Discovery". We benchmark the proposed BRNet model against AutoML for ML model and ElemNet, IRNet model for DL model.

| Dataset (Size) | Model | Test MAE |
|---|---|---|
| B2 (2784) | AutoML | 0.578 |
| | ElemNet | 0.568 |
| | IRNet | 0.530 |
| | BRNet | **0.500** |
| $L1_0$ (3467) | AutoML | 0.542 |
| | ElemNet | 0.577 |
| | IRNet | 0.538 |
| | BRNet | **0.509** |
| Perovskite (1302) | AutoML | 1.728 |
| | ElemNet | 0.482 |
| | IRNet | 0.478 |
| | BRNet | **0.459** |

Table 5 shows the comparison of the performance of proposed BRNet model against AutoML, ElemNet and IRnet for dataset of each type of crystal structures. Although we do not repeat the entire combinatorial search workflow here with the proposed models, the results demonstrates that our proposed BRNet model performs better on the evaluation candidates than other models. This suggests that BRNet model can improve the quality and robustness of the combinatorial search workflow. It is quite encouraging that despite a small training data size, the BRNet model can improve the performance of the combinatorial search for high-throughput materials discovery.

## 5 Conclusion and Future Work

In this paper, we analyzed and proposed design principles for building time and parameter effective deep regression networks composed of fully connected layers when there are numerical vectors as inputs. We introduced the use of branching of the network layers and the residual learning using skip connections in a deep regression network by proposing BRNet, which leverages the concept of branching and skip connection in each layer. The proposed BRNet not only outperforms ML/DL existing approaches for predictive modeling of several materials properties across multiple materials databases of different size but also shows time and computational efficiency during the model training phase. For the design problem,

the proposed BRNet significantly reduced the MAE from 0.149eV/atom (OQMD), 0.167eV/atom (MP) and 0.129eV/atom (JARVIS) to 0.041eV/atom (OQMD), 0.106eV/atom (MP) and 0.07eV/atom (JARVIS). The proposed deep learning architecture and design insights obtained from this work can significantly help build predictive models for other applications with numerical vector inputs. The code, data, and models developed in this work are publicly available at `https://github.com/GuptaVishu2002/BRNet` to the community to facilitate reproducibility and further building upon this work.

In the future, we plan to explore the effect of increasing the branching both horizontally and vertically on model accuracy and convergence and applying the architecture to other data mining problems. It will also be interesting to study the effect of more sophisticated multiple branching configurations. We also plan to study the model's performance on experimental datasets by using transfer learning techniques.

## Acknowledgments

This work is supported in part by the following grants: NIST award 70NANB19H005; DOE awards DE-SC0019358, DE-SC0021399; NSF award CMMI-2053929.

## References

[1] A. Agrawal and A. Choudhary, *Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science*, APL Materials, 4 (2016), p. 053208.

[2] A. Agrawal and A. Choudhary, *Deep materials informatics: Applications of deep learning in materials science*, MRS Communications, 9 (2019), pp. 779–792.

[3] G. Bergerhoff, R. Hundt, R. Sievers, and I. D. Brown, *The inorganic crystal structure data base*, Journal of Chemical Information and Computer Sciences, 23 (1983), pp. 66–69.

[4] K. Choudhary, K. F. Garrity, A. C. E. Reid, B. DeCost, A. J. Biacchi, A. R. H. Walker, Z. Trautt, J. Hattrick-Simpers, A. G. Kusne, A. Centrone, A. Davydov, J. Jiang, R. Pachter, G. Cheon, E. Reed, A. Agrawal, X. Qian, V. Sharma, H. Zhuang, S. V. Kalinin, B. G. Sumpter, G. Pilania, P. Acar, S. Mandal, K. Haule, D. Vanderbilt, K. Rabe, and F. Tavazza, *JARVIS: An integrated infrastructure for data-driven materials design*, 2020, `https://arxiv.org/abs/2007.01831`.

[5] S. Curtarolo, G. L. Hart, M. B. Nardelli, N. Mingo, S. Sanvito, and O. Levy, *The high-throughput highway to computational materials design*, Nature Materials, 12 (2013), p. 191.

[6] F. A. Faber, A. Lindmaa, O. A. Von Lilienfeld, and R. Armiento, *Machine learning energies of 2 million elpasolite (a b c 2 d 6) crystals*, Physical Review Letters, 117 (2016), p. 135502.

[7] F. A. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armiento, *Machine Learning Energies of 2 Million Elpasolite ABC2D6 Crystals*, Physical Review Letters, 117 (2016), p. 135502.

[8] C. Garbin, X. Zhu, and O. Marques, *Dropout vs. batch normalization: an empirical study of their impact to deep learning*, Multimedia Tools and Applications, (2020), pp. 1–39.

[9] L. M. Ghiringhelli, J. Vybiral, S. V. Levchenko, C. Draxl, and M. Scheffler, *Big data of materials science: Critical role of the descriptor*, Physical Review Letters, 114 (2015), p. 105503.

[10] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[11] L. Himanen, M. O. Jäger, E. V. Morooka, F. F. Canova, Y. S. Ranawat, D. Z. Gao, P. Rinke, and A. S. Foster, *Dscribe: Library of descriptors for machine learning in materials science*, Computer Physics Communications, 247 (2020), p. 106949.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, *Densely connected convolutional networks*, in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.

[13] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. a. Persson, *The Materials Project: A materials genome approach to accelerating materials innovation*, APL Materials, 1 (2013), p. 011002.

[14] D. Jha, S. Singh, R. Al-Bahrani, W.-k. Liao, A. Choudhary, M. De Graef, and A. Agrawal, *Extracting grain orientations from ebsd patterns of polycrystalline materials using convolutional neural networks*, Microscopy and Microanalysis, 24 (2018), pp. 497–502.

[15] D. Jha, L. Ward, A. Paul, W.-k. Liao, A. Choudhary, C. Wolverton, and A. Agrawal, *ElemNet: Deep learning the chemistry of materials from only elemental composition*, Scientific reports, 8 (2018), p. 17593.

[16] D. Jha, L. Ward, Z. Yang, C. Wolverton, I. Foster, W.-k. Liao, A. Choudhary, and A. Agrawal, *IRNet: A general purpose deep residual regression framework for materials discovery*, in 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019, pp. 2385–2393.

[17] S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl, and C. Wolverton, *The open quantum materials database (oqmd): assessing the accuracy of dft formation energies*, npj Computational Materials, 1 (2015), p. 15010.

[18] B. Komer, J. Bergstra, and C. Eliasmith,

*Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn*, in ICML workshop on AutoML, vol. 9, Citeseer, 2014, p. 50.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[20] J. Lee, A. Seko, K. Shitara, K. Nakayama, and I. Tanaka, *Prediction model of band gap for inorganic compounds by combination of density functional theory calculations and machine learning techniques*, Physical Review B, 93 (2016), p. 115104.

[21] B. Meredig, A. Agrawal, S. Kirklin, J. E. Saal, J. Doak, A. Thompson, K. Zhang, A. Choudhary, and C. Wolverton, *Combinatorial screening for new materials in unconstrained composition space with machine learning*, Physical Review B, 89 (2014), p. 094104.

[22] N. Nusran, K. R. Joshi, K. Cho, M. A. Tanatar, W. R. Meier, S. Bud'ko, P. C. Canfield, Y. Liu, T. A. Lograsso, and R. Prozorov, *Spatially-resolved study of the meissner effect in superconductors using nv-centers-in-diamond optical magnetometry*, New Journal of Physics, 20 (2018), p. 043010.

[23] C. W. Park and C. Wolverton, *Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery*, Phys. Rev. Materials, 4 (2020), p. 063801.

[24] A. Paul, D. Jha, R. Al-Bahrani, W.-k. Liao, A. Choudhary, and A. Agrawal, *CheMixNet: Mixed DNN architectures for predicting chemical properties using multiple molecular representations*, in Workshop on Molecules and Materials at the 32nd Conference on Neural Information Processing Systems, 2018.

[25] Z. Qin, G. S. Jung, M. J. Kang, and M. J. Buehler, *The mechanics and design of a lightweight three-dimensional graphene assembly*, Science advances, 3 (2017), p. e1601536.

[26] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, and C. Kim, *Machine learning in materials informatics: recent applications and prospects*, npj Computational Materials, 3 (2017), p. 54.

[27] S. Sanyal, J. Balachandran, N. Yadati, A. Kumar, P. Rajagopalan, S. Sanyal, and P. Talukdar, *Mt-cgcnn: Integrating crystal graph convolutional neural network with multitask learning for material property prediction*, arXiv preprint arXiv:1811.05660, (2018).

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

[29] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, *Schnet–a deep learning architecture for molecules and materials*, The Journal of Chemical Physics, 148 (2018), p. 241722.

[30] A. Seko, H. Hayashi, K. Nakayama, A. Takahashi, and I. Tanaka, *Representation of compounds for machine-learning prediction of physical properties*, Physical Review B, 95 (2017), p. 144110.

[31] A. D. Sendek, Q. Yang, E. D. Cubuk, K.-A. N. Duerloo, Y. Cui, and E. J. Reed, *Holistic computational structure screening of more than 12000 candidates for solid lithium-ion conductor materials*, Energy & Environmental Science, 10 (2017), pp. 306–320.

[32] R. K. Srivastava, K. Greff, and J. Schmidhuber, *Training very deep networks*, in Advances in Neural Information Processing Systems, 2015, pp. 2377–2385.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions*, in IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[34] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, *A General-Purpose Machine Learning Framework for Predicting Properties of Inorganic Materials*, npj Computational Materials, 2 (2016), p. 16028.

[35] L. Ward, R. Liu, A. Krishna, V. I. Hegde, A. Agrawal, A. Choudhary, and C. Wolverton, *Including crystal structure attributes in machine learning models of formation energies via Voronoi tessellations*, Physical Review B, 96 (2017), p. 024104.

[36] L. Ward and C. Wolverton, *Atomistic calculations and materials informatics: A review*, Current Opinion in Solid State and Materials Science, 21 (2017), pp. 167–176.

[37] L. T. Ward, A. R. Dunn, A. Faghaninia, N. E. R. Zimmermann, S. Bajaj, Q. Wang, J. E. P. Montoya, J. Chen, K. Bystrom, M. T. Dylla, K. Chard, M. Asta, K. A. Persson, G. J. Snyder, I. T. Foster, and A. Jain, *Matminer: An open source toolkit for materials data mining*, Computational Materials Science, 152 (2018), pp. 60–69.

[38] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, *Aggregated residual transformations for deep neural networks*, in Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, IEEE, 2017, pp. 5987–5995.

[39] T. Xie and J. C. Grossman, *Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties*, Phys. Rev. Lett., 120 (2018), p. 145301.

[40] D. Xue, P. V. Balachandran, J. Hogden, J. Theiler, D. Xue, and T. Lookman, *Accelerated search for materials with targeted properties by adaptive design*, Nature communications, 7 (2016).

[41] Q. Zhou, P. Tang, S. Liu, J. Pan, Q. Yan, and S.-C. Zhang, *Learning atoms for materials discovery*, Proceedings of the National Academy of Sciences, 115 (2018), pp. E6411–E6417.