Rediscovering Scientific Knowledge Using Deep Learning

Shashank Srikanth[1], Utkarsh Azad[1]

Abstract— **Application of conventional machine learning approaches requires one to make extensive use of the prior domain knowledge to manually construct feature vectors. However, the recent advancements in deep learning approaches allow us to bypass this stringent requirement and achieve much better results. In this work, we show a procedure to construct a feature vector for a chemical system ($\mathbb{C}$) using its elemental composition ($A^1_{x_1}, A^2_{x_2} \dots A^n_{x_n}$) and an extensive database of known compounds to which it belongs. We use it as a basic input (X) for our convolution neural network (ConvNet), which are then trained to predict material properties (y) such as Superconductivity i.e. critical temperature (T_c), and formation energy (E_f) with significant accuracy. To foster further research in compound representations and ensure reproducibility, we release all our code and data [1].**

I. INTRODUCTION

The tasks such as molecular discovery or gene prediction requires exhausting traversal in an infinite search space, i.e. validating results from the experimentation of all possible combinations of compositions and structures. In general, such a resource-expensive and time-consuming procedure is neither feasible nor preferred by natural scientists. In the mean time, the consistent growth in our computing capabilities have lead the way to develop computational methods that offers a less expensive means for partial traversal in a given search space. Accumulation of the results from these computational methods, along with the reported experimental data till date have offered opportunities for large-scale data collection such as the Open Quantum Materials Database (OQMD) and the Automatic Flow of Materials Discovery Library (AFLOWLIB). The availability of such massive amounts of both experimental and numerical simulation data in the natural sciences has enabled the use of machine learning (ML) and artificial intelligence (AI) for accelerating tasks concerning predictions of new materials and their properties.

However, despite the excitement and promising results, it must be noted that the conventional ML models requires manual construction of feature vectors which makes use of extensive amount of prior domain knowledge. For example, Wolverton et al. used a feature vector based on the composition of a material to predict its properties such as formation enthalpy. Hence, the prediction accuracy of our models are largely limited by the meaningful information we can encode about the system in its feature vector. Over the years, this has largely restricted applicability of ML techniques in many domains of natural science. On the other hand, development of deep-learning techniques have allowed a route to bypass this requirement by reducing the need for designing physically-relevant features. This is possible due to ability of its network architecture to learn from a general abstracted-data descriptors instead of application-specific descriptors and perform a prediction task with sufficient accuracy.

In this work, we aim to discard the use of human-centric visualization of the compounds (such as SMILES) and instead allow the our DL models to directly learn the properties of materials such as superconductivity i.e. critical temperature (T_c), and formation energy (E_f) from the their elemental composition ($\{A^1_{x_1}, A^2_{x_2} \dots A^n_{x_n}\}$) and descriptors prepared directly from the extensive database of known compounds without incorporating any priors or human knowledge. The rest of this work has been divided into the following parts: In Section II we discuss our approach to generate the descriptors for our DL model, in Section III we present a detailed discussion about our datasets and CNN model, in Section IV we introduce the predictions tasks performed and showcase our results for each of them. Finally, in Section V we present our conclusions.

II. APPROACH

For our DL model to perform tasks such as molecular discovery, it needs to be able to learn the chemical interactions and similarities between different elements constituting a chemical systems $\{C_1, C_2, \dots C_N\}$, where N represents the size of our database. To do this, initially we represent any given chemical system C_i with its elemental composition $E_i = (\{A^1_{x_1}, A^2_{x_2} \dots A^n_{x_n}\})$, where A^j_k represents an element with chemical symbol A^j with composition k in C_i and n corresponds to total number of unique elements present in the dataset. Using these E_is one can gather information about all the composition of environments E^q_{env} that an element with chemical label A^q interacts with as follows:

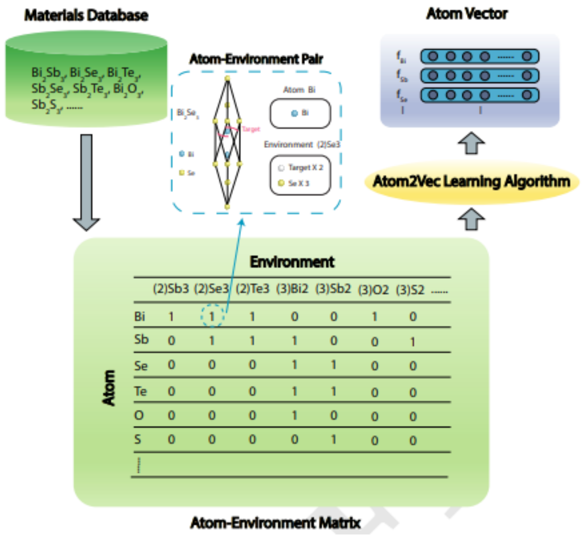
$$E^q_{env} = \{(A^1_{x_1} A^2_{x_2} \dots A^{q-1}_{x_{q-1}} A^{q+1}_{x_{q+1}} \dots A^n_{x_n}) \\ \dots (A^1_{y_1} A^2_{y_2} \dots A^{q-1}_{y_{q-1}} A^{q+1}_{y_{q+1}} \dots A^n_{y_n})\}$$

Given a sufficiently large data set, similarity between any two E^p_{env} and E^q_{env} can be used as a metric to determine the similarity between two elements with chemical label A^p and A^q because similar atoms will tend to appear in similar environments. Using the procedure illustrated by Quan Zhou et al [1] we generated all atom-environment pairs from the given dataset and then record them in an

*This work is the final term paper submission of the ML in Natural Sciences course

[1] H. International Institute of Information Technology, Hyderabad shashank.s at research.iiit.ac.in

[1] Code and data is available at https://rebrand.ly/mlns-unsupervised-code



	(2)Sb3	(2)Se3	(2)Te3	(3)Bi2	(3)Sb2	(3)O2	(3)S2	......
Bi	1	1	1	0	0	1	0	
Sb	0	1	1	1	0	0	1	
Se	0	0	0	1	1	0	0	
Te	0	0	0	1	1	0	0	
O	0	0	0	1	0	0	0	
S	0	0	0	0	1	0	0	
⋮								

Fig. 1. Atom2Vec workflow to learn atom from any dataset

atomenvironment matrix $\mathbf{X}$, with the dimensions $(n \times m)$, where m corresponds to total number of unique environments present in the dataset. Its each entry $\mathbf{X}_{ij}$ gives the count of pairs with the i^{th} atom and the j^{th} environment. Therefore, each row vector gives counts with different environments for one atom, and each column vector yields counts with different atoms for one environment. The sum of counts over the coloumn $\sum_j X_{ij}$ gives the population of the i^{th} atom, which can differ greatly among all symbols. We apply the normalization $X_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{\frac{1}{2}}$ to solve this. Now, the row vectors of the normalized matrix $X = [x^1, x^2, \ldots, x^N]$ provide a primitive representation for atoms. A schematic diagram of generating the atom vectors for a given dataset is given in Figure 1.

In some high dimensional space, the vectors x_i of similar atoms would grouped together. Now, to find the hidden structure in the rows x_i of $\mathbf{X}$, we make use of SVD which extracts entangled and related properties of the data into fewer principal directions with no correlations and the highest variances. This is achieved by decomposing our $n \times m$ matrix $\mathbf{X}$ into r components with the singular value σ_i demonstrating its significance. In matrix representation we can represent this decomposition as $\mathbf{X} = \mathbf{UDV^T}$, where U is the $n \times n$ orthogonal matrix, V is the $m \times m$ orthogonal matrix, and D is the $n \times m$ diagonal matrix with diagonal elements corresponding to singular values. We select d largest singular values from D i.e. a $d \times d$ matrix D', and the corresponding columns from U, namely $n \times d$ matrix U'. We then use the product of these two matrices to get a $n \times d$ matrix F given as:

$$F = [f_1, f_2 \ldots f_n]^{\mathbf{T}}$$

The row vector's f_i of F yield a better and more compact description for atoms. Hence, we are now done with representing every atom A^i using a vector f^i which is an abstract representation of its interaction environment E^i_{env} in the given database.

Next, we to use DL models such as CNN we need a way to represent our chemical system C_i as an image data. Using an approach given by Shuming Zeng et al [2] we represent elemental composition E_i of each C_i as a $g \times g$ pixels image $\mathbf{P}$, where $g = \lceil \sqrt{n} \rceil$. Each pixel P_{ij} then corresponds to an element A^k_l with atomic number $k = i \times g + j$ in the periodic table with value l i.e. the proportion of this element in C_i. Now to complement this representation with the information such as interaction capability of the elements present in E_i of each C_i, we append previously generated abstract representation f^i of every element A^i to its pixel value. This converts the matrix $\mathbf{P}$ in to 3-D matrix with dimensions $(g \times g \times (d+1))$. This representation of any chemical system C_i has been prepared from the data-set itself, without incorporating any priors or human knowledge. It sufficiently encodes essential information about the composition, chemical interactions of its constituent elements. Moreover, for any two chemical systems C_i and C_j, it also encodes the information regarding similarity between the elements in their compositions.

III. Dataset

Our work deals with the prediction of the critical temperature (T_c) and enthalpy of formation E_f. The experimental data for T_c and E_f are extracted from the Supercon database and the Open Quantum Materials Database (OQMD) [3] respectively. The critical temperature dataset has 20,000 compounds with the total number of unique elements in the dataset less than 100. As a single compound can have different T_c value due to different crystallization / experimental techniques, we remove a compound from the dataset if the maximum T_c value is twice the minimum value. In all other cases, we calculate the average T_c value for each compound and fix it as the T_c value for that compound after removing the duplicates. Thus, the cleaned dataset is similar to the dataset used by [2] in their experiments.

We evaluate our trained models on the compounds Hg, MgB_2, FeSe and $YBa_2Cu_3O_7$ to test the generalizability of our approach. We also split our test dataset into four groups similar to [2], to analyze the model in greater detail. The four groups each consist of compounds containing Cu, Fe, both Cu & Fe, and the remaining compounds respectively. The first three groups represent Cu based superconductors, Fe based superconductors, and the common BCS superconductors respectively. We perform ablation study of our models by evaluating their performance on these four separate groups to identify the groups in which our approach performs exceptionally well or badly. We compare the performance of our approach with that of [2] on all these four groups and show superior performance on most of them.

As our approach is quite general in nature, we also test the predictive power of the model by training it on a dataset of E_f energies. We predict the formation energies of elpasolite crystals ABC_2D_6, a type of quaternary minerals with excellent scintillation performance, which are thus very suitable for application in radiation detection [4]. We compare our

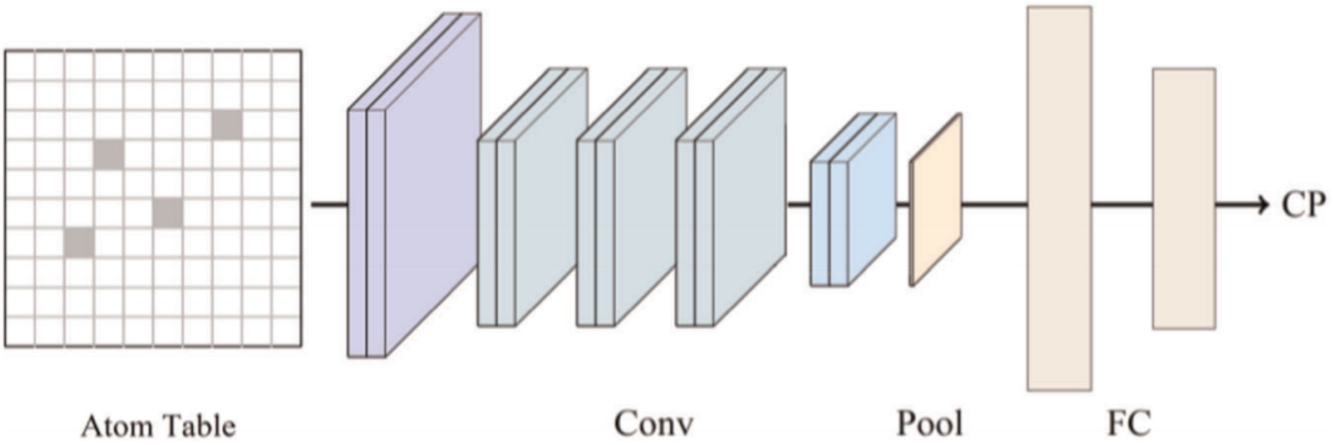


Fig. 2. Schematic diagram of the our approach for T_c prediction

approach with that of [1] on this task and train on over 10,000 elpasolite crystals dataset. As [1] do not make their dataset available, we collect our own dataset from the Materials Project database [5] in a manner similar to [1].

IV. CNN Architecture

A schematic diagram of our approach is given in Figure 2. We utilize a CNN based architecture for predicting properties as they have been shown to perform extremely well in tasks such as Image classification [6] and Object detection [7]. Our approach takes as input an $(g \times g \times (d+1))$ matrix that represents the compound and encodes essential information about the composition, and chemical interactions of its constituent elements. As our dataset has less than 100 unique elements, we set $g = 10$, thus, the input size is $10 \times 10 \times (d+1)$. Through a series of convolution, pooling and non-linearity operations, the encoder reduces the resolution of the input from $10 \times 10 \times (d+1)$ to a flattened column vector of size 200. This flattened column vector is then passed through two fully connected layers with 200 and 100 hidden units respectively, which then outputs the final predicted value. In order to increase the generalizability of the model, we also utilize dropout layers in the network which act as a form of regularization [8]. We also utilize batch normalization [9] to speedup the training process and improve model's predictive power.

The network is trained using the L2 norm between the predicted output and ground truth value and the loss function consists of the above L2 norm and weight regularization. We utilize the AdaDelta [10] optimizer for training with a learning rate of 0.01. The network was trained for a maximum of 100 epochs each with a batch size of 128.

V. Experiments and Results

We evaluate our approach on two prediction tasks: 1) T_c values and 2) E_f values and compare our approach against those of [2] and [1] respectively. All of our models are based on the above CNN architecture and are named as RDNET (Rediscovery Net) subsequently. We test our model in three different experimental settings as mentioned below.

A. Experiment 1

The network is trained on a dataset consisting of only superconductors and the task is to predict the critic temperature T_c. There were a total of 77 unique elements in the dataset with over 50,890 unique environment pairs. We compare our approach with the ATCNN-I model presented in [2]. As mentioned earlier, to test the generalizability of our model, it is tested on a set of four different groups in the test set and also on 4 unique compounds/elements, namely Hg, MgB_2, FeSe and $YBa_2Cu_3O_7$.

We compare the effectiveness of our approach with that of [2] in Table I. The metric used is the MAE (Mean Absolute Error) and lower values indicate better prediction. From the table, we can see that our approach performs better than ATCNN-I for all the groups excepting Cu superconductors. This suggests that our approach is able to represent the compounds better than [2]. We also test the generalizability of our model by predicting the T_c values for the four compounds/elements shown in Table II and comparing it with the experimentally predicted values. Our approach performs better than ATCNN-I in two cases and worse than ATCNN-I in remaining cases. This might be because there were not enough compounds in the dataset that were similar to MgB_2 and $YBa_2Cu_3O_7$.

Compound Type	Dataset Size	Test Results (MAE)	
	No. of elements	ATCNN-I	RDNET
Cu Based	1122	**6.98**	8.31
Fe Based	287	4.90	**3.96**
Cu & Fe Based	69	9.77	**8.59**
Rest	1242	1.69	**1.58**
Total	2720	4.27	**4.16**

TABLE I
Comparison with ATCNN-I

B. Experiment 2

In the second set of experiments, we train our network on a dataset comprising of both superconductors and insulators. The network trained on this dataset can also be used for

Compound Type	Test Results (MAE)		
	ATCNN-I	RDNET	Experimental
Hg	1.4	**2.38**	4.12
MgB_2	**38**	37.51	39
FeSe	9.8	**9.17**	8.0
$YBa_2Cu_3O_7$	**91.5**	88.40	91.0

TABLE II

COMPARISON OF EXPERIMENTALLY MEASURED T_c WITH [2] AND RDNET

classifying between superconductors and insulators as the insulators/non-superconductors have a T_c value of 0. There were a total of 86 unique elements in the dataset with over 83,034 unique environment pairs. In this experiment, we compare our approach with the ATCNN-II model presented in [2].

Similar to the evaluations in experiment 1, we compare the effectiveness of our approach with the ATCNN-II model in Table III and IV respectively. Our model again shows slight gains over the ATCNN-II model proposed in [2].

Compound Type	Dataset Size	Test Results (MAE)	
	No. of elements	ATCNN-II	RDNET
Cu Based	1156	**7.25**	8.76
Fe Based	282	4.62	**3.59**
Cu & Fe Based	67	9.49	**9.14**
Rest	1215	1.71	**1.18**
Total	2720	4.12	**4.01**

TABLE III

COMPARISON WITH ATCNN-II

Compound Type	Test Results (MAE)		
	ATCNN-II	RDNET	Experimental
Hg	2.6	**3.07**	4.12
MgB_2	**38.7**	35.80	39
FeSe	8.1	**7.97**	8.0
$YBa_2Cu_3O_7$	**90.6**	86.09	91.0

TABLE IV

COMPARISON OF EXPERIMENTALLY MEASURED T_c WITH ATCNN-II AND RDNET

C. Experiment 3

We also evaluate our approach on the task of formation energy (E_f) prediction. As our approach is based on [1], we compare our approach with their model on the same task, i.e. predicting E_f energies of Elpasolite crystals. There were a total of 39 unique elements in the dataset with over 35,075 unique environment.

The results for the same are shown in Table V below. From Table V we can see that our model significantly outperforms [1] when trained with the roughly same number of components ($d = 5$ and $d = 10$). In fact RDNET trained with only 5 components outperforms even the atom2vec [1] model trained with $d = 30$ components by a margin of 0.02 eV, suggesting the potent of our method for predictive tasks.

Compound Type	Dataset Size	Test Results (MAE)	
	No. of elements	Atom2Vec	RDNET
Half Heusler	10,000	0.24	0.13

TABLE V

COMPARISON WITH ATOM2VEC [1] ON E_f PREDICTION

D. Representations

In order to understand if the representations are indeed able to capture the elemental properties, we apply PCA on our element's feature vectors and plot it on a 2D plane as shown in Figure 3. From Figure 3, we can see that several similar elements such as F, Cl and O, Se are clustered near each other. Thus, the unsupervised learning method is able to learn the semantics and similarity of elemental composition using only the atom and environment pairs.

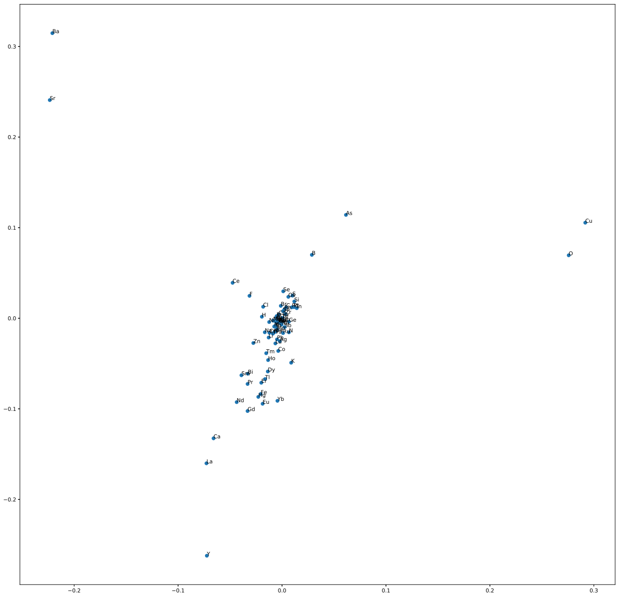


Fig. 3. Clustering of our elemental feature vectors using Principal Component Analysis (PCA)

VI. CONCLUSION

We propose an unsupervised learning method based on [1] to construct a feature vector for a chemical system ($\mathbb{C}$) using its elemental composition $(A^1_{x_1}, A^2_{x_2} \dots A^n_{x_n})$ and an extensive database of known compounds to which it belongs. We combine this feature representation with the approach proposed in [2] and show that a combination of learnt priors and CNN based models can perform extremely well on several prediction tasks. We evaluate our approach on the task of T_c and E_f prediction and show that it performs better than the approach mentioned in [2] and [1] respectively. We also show that our approach is able to learn the elemental properties correctly and that similar elements are clustered together in the feature space. Thus, such unsupervised end to end deep learning based approaches have significant potent to perform well in prediction tasks and need to be studied in detail.

Acknowledgement

We would like to thank all the professors at CCNSB for providing us with the opportunity to work in this project and their valuable inputs and suggestions on the work.

References

[1] Quan Zhou, Peizhe Tang, Shenxiu Liu, Jinbo Pan, Qimin Yan, and Shou-Cheng Zhang. Atom2vec: learning atoms for materials discovery. *arXiv preprint arXiv:1807.05617*, 2018.

[2] Shuming Zeng, Yinchang Zhao, Geng Li, Ruirui Wang, Xinming Wang, and Jun Ni. Atom table convolutional neural networks for an accurate prediction of compounds properties. *npj Computational Materials*, 5(1):1–7, 2019.

[3] James E Saal, Scott Kirklin, Muratahan Aykol, Bryce Meredig, and Christopher Wolverton. Materials design and discovery with high-throughput density functional theory: the open quantum materials database (oqmd). *Jom*, 65(11):1501–1509, 2013.

[4] R Hawrami, E Ariesanti, L Soundara-Pandian, J Glodo, and KS Shah. Tl 2 liycl 6: Ce: A new elpasolite scintillator. *IEEE Transactions on Nuclear Science*, 63(6):2838–2841, 2016.

[5] G Ceder and K Persson. The materials project: A materials genome approach, 2010.

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

[7] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[8] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

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

[10] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.