# Supporting Information: Contrastive Representation Learning of Inorganic Materials to Overcome the Lack of Training Datasets

Gyoung S. Na
Korea Research Institute of Chemical Technology, Republic of Korea[1]
ngs0@krict.re.kr

March 28, 2022

## 1 Materials Descriptors for Machine Learning

Over the past decade, a lot of work has been done to represent materials for more accurate ML [27]. These representations were mainly developed on a solid background in sciences such as condensed matter physics and materials science. These representations of materials are necessary to extract essential information in a computationally efficient manner and to satisfy physical or mathematical requirements such as differentiability and invariance to rotation, translation and permutation [27]. Most representations for crystalline materials utilize structural information with atomic properties such as atom-centered symmetry functions [5] and smooth overlap of atomic positions [4]. Structural fragments with a few atoms are used to represent the local structure of materials [14]. As atoms and their neighborhoods can be described by the graph, GNNs are also adopted to treat the local information in materials and representative examples are CGCNN and MegNet [8; 35]. Moreover, atomic features are taken from the elemental properties of each atom and its surrounding environment. These features can be selected according to the target materials properties [6]. However, to the best of our knowledge, representation methods that universally encode crystal structures by explicitly treating given target materials properties have not been reported publicly.

## 2 Representation Learning in Machine Learning

Representation learning is categorized into unsupervised and supervised methods. In unsupervised representation learning, a new data representation is automatically discovered without label or target data by minimizing density divergence or reconstruction loss [3; 18; 26]. Autoencoder [3] is the most popular algorithm in unsupervised learning owing to its remarkable representation capability. To improve the generalization capability of autoencoders, variational autoencoder [18] was also proposed based on Bayesian inference. However, unsupervised representation learning algorithms have an inherent limitation in that the target values cannot be utilized in representation learning.

By contrast, supervised representation learning generates a new data representation based on the label or target data. Deep metric learning (DML) has been widely studied for supervised representation learning in computer science [12; 34]. The goal of DML is to train an embedding network $f : \mathcal{X} \rightarrow \mathbb{R}^m$ that generates a new $m$-dimensional data representation from the input data in $\mathcal{X}$. However, DML has been mainly studied for image classification tasks of discrete target values [28; 29; 33; 34]. Although a new metric learning loss of DML for continuous target values was recently proposed [17], it is difficult to apply scientific applications owing to its numerical instability. To overcome this numerical instability, smooth log-ratio loss (SLRL) [20] was proposed and showed some improvement in predicting molecular properties. However, no DML methods, including SLRL-based DML, have been applied to materials science to predict materials properties from the crystal structures.

## 3 Graph-Based Machine Learning for Materials Science

A crystal structure is natively represented as a mathematical graph $G = (\mathcal{V}, \mathcal{E}, X, S)$, where $\mathcal{V}$ is a set of atoms in the unit cell; $\mathcal{E}$ is a set of chemical bonds between the atoms; $X$ is a matrix of the atomic features; and $S$ is a matrix of the bond features. In the graph-based machine learning, this graph-shaped crystal structure is entered as it is without being converted to a vector-shaped data. Graph neural networks (GNNs) are used to process the graph-structured data.

GNNs for predicting graph-level properties are composed of aggregation layers, readout, and dense (fully-connected) layers. The purpose of the aggregation layer is to generate latent features of the nodes in the given graph-structured data. The latent node features $H$ are calculated in the $i^{th}$ aggregation layer as:

$$H^{(i)} = \sigma(f_a(A, H^{(i-1)}, S)), \tag{1}$$

where $\sigma$ is a nonlinear activation function (e.g., sigmoid and tanh); $A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ is an adjacency matrix of the input graph $G$; and $H^{(i-1)}$ is a matrix of the latent node features generated in the previous layer. After generating the latent node features, the readout is applied to generate a graph-level embedding $\mathbf{z}$ from the latent node features generated in the layer aggregation layer. The readout is commonly defined by *mean* or *max* operations. For example, a popular CGCNN [35] generates $\mathbf{z}$ using a *mean*-based readout as:

$$\mathbf{z} = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} H_i^{(L)}, \tag{2}$$

where $H_i$ is the $i^{th}$ row vector of $H$, and $L$ is the number of aggregation layers in GNN. Finally, the target graph-level property is predicted by feeding the calculated graph-level embedding $\mathbf{z}$ to the dense layers.

Table 1: Selected elemental features and their brief descriptions.

| Name | Category | Unit |
|---|---|---|
| Atomic volume | Size | $cm^3/mol$ |
| Atomic weight | Size | - |
| Covalent radius by Bragg | Size | pm |
| Fusion heat | Heat | kJ/mol |
| Atomic number | Electronic | - |
| Electron affinity | Electronic | eV |
| First ionization energy | Electronic | eV |
| Pauling's scale of electronegativity | Electronic | - |
| Period in periodic table | Electronic | - |

Table 2: Hyperparameter settings of TGNN for each dataset.

| Dataset | Margin ($=\alpha$) | Initial learning rate | $L_2$ coefficient | Batch size | Neighborhood atom cutoff (Å) |
|---|---|---|---|---|---|
| MPB | 2e-1 | 5e-4 | 5e-6 | 48 | 5 |
| MPS-FE | 2e-1 | 5e-4 | 5e-6 | 48 | 5 |
| MPS-BG | 8e-2 | 5e-4 | 5e-4 | 48 | 4 |
| MPL-FE | 2e-1 | 5e-4 | 5e-4 | 16 | 5 |
| MPL-BG | 8e-2 | 5e-4 | 5e-4 | 16 | 5 |
| HOIP | 2e-1 | 5e-4 | 5e-6 | 32 | 5 |
| NLHM | 2e-1 | 5e-4 | 5e-6 | 32 | 5 |

## 4 Implementation Details

We used PyTorch framework [22] and PyTorch-Geometric library [2] to implement GNNs and EMRL. For the implementation of CGCNN and MEGNet, we used the author's source code of CGCNN [36] and MEGNet [9]. Implementation details and hyperparameter settings of the GNNs and the EMRL-based ML algorithms are summarized in Table 1 and 2.

To convert the crystal structures into the mathematical graphs, we used a well-known pymatgen library [1]. Table 1 shows the elemental features used to convert the crystal structures. However, we followed the elemental features of [35] and [8] in implementing CGCNN and MEGNet, respectively. The bond features were generated by applying the radial basis function (RBF) to the bond length.

We used an architecture of three aggregation and two dense layers for all GNNs used for the evaluations. The architecture of TGNN was also fixed for all datasets. The atom-embedding network $f_v$ and the tuple-embedding network $f_t$ were implemented as one-layer neural networks with 128 neurons. The bond-embedding network $f_e$ was also implemented as a one-layer neural network with 64 neurons. We stacked two dense layers of 128 neurons, after the tuple embedding network and the readout. Table 2 shows the hyperparameter settings of TGNN for each dataset. The hyperparameters expect for the margin, which are common for all GNNs, were applied to the GNNs in the same values as Table 2. The source code of EMRL is publicly available at *Open After the Review Process*.

## 5 Prediction Performances on Benchmark Materials Datasets

To evaluate the effectiveness of EMRL, we conducted experiments to predict materials properties. For our experimental evaluations, we used seven materials datasets containing about 50,000 materials. The main characteristics and sources of each dataset are listed in Table 3. In the experiments, we measured the prediction errors of state-of-the-art GNNs and EMRL-based ML algorithms using mean solute error (MAE) and standard deviation. For all evaluations, the mean and the standard deviation of the prediction performances were measured by repeating the evaluations 10 times on randomly partitioned training and test datasets for each repetition. In the experiments, we generated three EMRL-based ML algorithms by exploiting linear regression, fully-connected neural network (FNN) [24], and XGBoost (GB) [10] as a prediction model $g$. Brief descriptions of the three EMRL-based ML algorithms are as follows.

- **EMRL-LR:** Linear regression is used as a prediction model in EMRL-LR. It directly shows the effectiveness of the materials representation generated via EMRL in predicting materials properties because linear regression is the simplest prediction algorithm in ML.

- **EMRL-NN:** FNN [23] is used as a prediction model in EMRL-NN. It can be used to compare the effectiveness of the graph representations generated via GNNs and EMRL because GNNs employ FNN as their prediction model.

- **EMRL-GB:** GB is the most popular regression algorithm in scientific communities and computer science fields [10; 11; 25; 37]. EMRL-GB uses GB as a prediction model, and it can be regarded as the most advanced EMRL-based ML algorithm in the experiments.

We compared the prediction performances of the EMRL-based ML algorithms with five state-of-the-art GNNs: (1) graph convolutional network (GCN) [19]; (2) graph attention network (GAT) [31]; (3) tuplewise graph neural network (TGNN) [21]; (4) crystal graph convolutional neural network (CGCNN) [35]; and (5) materials graph network (MEGNet) [8]. Table 4 summarizes the evaluation results based on MAE. For a fair comparison, we also compared TGNN and EMRL-based methods as TGNN is adopted as the embedding network for our EMRL implementation. Furthermore, we compared the performance improvement of EMRL and CGCNN because CGCNN is the most popular GNN in ML applications for materials science. As shown in Table 4, EMRL-GB significantly outperformed the state-of-the-art GNNs in predicting materials properties, and the prediction errors were reduced

Table 3: Characteristics and source of the benchmark materials datasets used for experimental evaluations. In all datasets, the unit of band gap and formation energy are eV and eV/atom, respectively.

| Dataset | Type of materials | Target property | # of materials | Range of targets |
|---|---|---|---|---|
| MPB [15] | Binary inorganic materials | Band gap | 6,838 | [0, 1.593] |
| MPS-FE [15; 35] | Inorganic materials | Formation energy | 3,162 | [-4.319, 2.757] |
| MPS-BG [15; 35] | Inorganic materials | Band gap | 3,162 | [0, 8.716] |
| MPL-FE [15; 35] | Inorganic materials | Formation energy | 45,941 | [-4.576, 3.195] |
| MPL-BG [15; 35] | Inorganic materials | Band gap | 45,941 | [0, 16.586] |
| HOIP [16] | Hybrid perovskites | Band gap | 1,345 | [1.025, 5.343] |
| NLHM [7] | Light harvesting materials | Band gap | 2,233 | [0, 9.059] |

Table 4: Prediction errors of four GNNs and three EMRL-based ML algorithms on the benchmark datasets. The errors were measured by mean of MAEs under 10 times repetitions of the evaluations on randomly divided 80% training and 20% test datasets. The standard deviation of the measured errors was presented in the parenthesis. Error reduction (Rdc.) is defined as the relative error reduction of EMRL-GB for CGCNN, as shown in Eq. (3). We compared TGNN and EMRL-based methods in the main text. Two abbreviations FE and BG indicate formation energy and band gap, respectively. The smallest prediction error for each dataset was highlighted in bold.

| Dataset | Graph neural networks | | | | | EMRL-based ML algorithms | | | Rdc. |
|---|---|---|---|---|---|---|---|---|---|
| | GCN | GAT | TGNN | CGCNN | MEGNet | EMRL-LR | EMRL-NN | EMRL-GB | |
| MPB | 0.296<br>(0.007) | 0.293<br>(0.005) | 0.271<br>(0.023) | 0.268<br>(0.035) | N/A | 0.316<br>(0.009) | 0.219<br>(0.016) | **0.189**<br>**(0.015)** | 29.5% |
| MPS-FE | 0.181<br>(0.000) | 0.177<br>(0.013) | 0.105<br>(0.017) | 0.123<br>(0.021) | 0.081<br>(0.002) | 0.077<br>(0.006) | 0.079<br>(0.006) | **0.076**<br>**(0.005)** | 38.2% |
| MPS-BG | 0.339<br>(0.006) | 0.309<br>(0.013) | 0.301<br>(0.021) | 0.298<br>(0.041) | N/A | 0.304<br>(0.025) | 0.267<br>(0.029) | **0.264**<br>**(0.021)** | 11.4% |
| MPL-FE | 0.129<br>(0.002) | 0.122<br>(0.001) | 0.059<br>(0.007) | 0.058<br>(0.001) | N/A | 0.047<br>(0.002) | 0.046<br>(0.001) | **0.026**<br>**(0.001)** | 55.2% |
| MPL-BG | 0.501<br>(0.005) | 0.468<br>(0.004) | 0.389<br>(0.019) | 0.402<br>(0.004) | N/A | 0.395<br>(0.005) | 0.366<br>(0.006) | **0.337**<br>**(0.004)** | 16.2% |
| HOIP | 0.295<br>(0.017) | 0.253<br>(0.014) | 0.325<br>(0.054) | 0.255<br>(0.023) | 0.193<br>(0.071) | 0.142<br>(0.004) | 0.149<br>(0.004) | **0.142**<br>**(0.003)** | 44.3% |
| NLHM | 0.681<br>(0.013) | 0.573<br>(0.021) | 0.529<br>(0.056) | 0.633<br>(0.021) | **0.315**<br>**(0.018)** | 0.468<br>(0.016) | 0.469<br>(0.016) | 0.461<br>(0.015) | 27.2% |

by 11.4%∼55.2% by EMRL-GB over CGCNN. Furthermore, we can notice that EMRL-based methods outperform TGNN, which directly shows the effectiveness of EMRL in predicting materials property prediction.

Although MEGNet outperformed the other GNNs and showed the smallest error on the NLHM dataset, it was not applicable to most materials project datasets. Among the evaluation results, the significant improvement by EMRL on the HOIP dataset is impressive because perovskites have severe non-smooth relations with their band gaps owing to significantly different band gaps for their similar crystal structures [32]. This evaluation result concurs with our motivation that the non-smooth relations between the crystal structures and the target properties make ML-based regression difficult, and these non-smooth relations can be effectively transformed to the smooth relations via supervised representation learning. In addition to EMRL-BG, EMRL-LR and EMRL-NN also exhibited smaller prediction errors compared to the GNNs overall. As summarized in Table S3 of SI, these improvements by EMRL in predicting materials properties were consistent with the evaluation results using the $R^2$ score [13]. In particular, it is important that the simplest linear regression obtained comparable prediction accuracy with CGCNN just by exploiting the materials representations generated by EMRL. This result directly shows that the materials representations generated via EMRL clearly described the target materials properties.

In addition to MAE, we also measured the prediction performance of the machine learning algorithms using $\mathrm{R}^2$ score [13] that is the most widely used criterion to evaluate the regression accuracy. For all datasets and ML algorithms, the $\mathrm{R}^2$ scores were measured by repeating the evaluations 10 times on randomly separated training and test datasets for each repetition. The mean and the standard deviation of the measured $\mathrm{R}^2$ scores were reported. Table 5 shows the measured $\mathrm{R}^2$ scores in predicting materials properties on the benchmark materials datasets.

Although MEGNet showed the best prediction accuracy on the NLHM dataset, it was not applicable in many datasets as we mentioned in the paper. For all other datasets, EMRL-GB achieved the best prediction accuracy. The improvements of EMRL-GB over CGCNN in the $\mathrm{R}^2$ score were 0.6-12.9%. As shown in the results, the accuracy improvements by EMRL-GB on the MPS-FE, MPL-FE, and HOIP datasets were marginal because most algorithms already achieved the $\mathrm{R}^2$ scores of 0.9. The accuracy improvements of EMRL-GB were higher than 11% for the datasets, where the prediction accuracy was relatively low. Furthermore, the EMRL-LR and EMRL-NN also showed better $\mathrm{R}^2$ scores than the state-of-the-art GNNs overall.

## 6 Embedding Distributions

We also interpreted the distributions of the materials in the imaginary materials spaces of EMRL based on chemical domain knowledge. Fig. 1 shows the distributions of the materials in the embedding spaces of EMRL and CGCNN. In the HOIP dataset, as shown in Fig. 1**a** and **c**, EMRL generated more continuous materials representations than CGCNN and established that halogen elements (F, Cl, Br, and I) are more crucial for modulating the band gap than metal elements (Ge, Sn, and Pb). This embedding result is in accordance with previous reports on HOIPs [16].

Table 5: $R^2$ scores of four GNNs and three EMRL-based ML algorithms on the benchmark datasets. The standard deviation of the measured $R^2$ scores was presented in the parenthesis. Accuracy improvement (Imp.) is defined as the relative improvement of EMRL-GB for CGCNN. The abbreviations FE and BG indicate formation energy and band gap, respectively. The highest $R^2$ score for each dataset was highlighted by the bold font.

| Dataset | Graph neural networks | | | | | EMRL-based ML algorithms | | | Imp. |
|---|---|---|---|---|---|---|---|---|---|
| | GCN | GAT | TGNN | CGCNN | MEGNet | EMRL-LR | EMRL-NN | EMRL-GB | |
| MPB | 0.754<br>(0.016) | 0.747<br>(0.011) | 0.751<br>(0.013) | 0.763<br>(0.013) | N/A | 0.776<br>(0.015) | 0.827<br>(0.008) | **0.871**<br>**(0.012)** | 12.4% |
| MPS-FE | 0.887<br>(0.006) | 0.889<br>(0.009) | 0.969<br>(0.012) | 0.967<br>(0.005) | 0.978<br>(0.006) | 0.984<br>(0.006) | 0.982<br>(0.006) | **0.985**<br>**(0.006)** | 1.8% |
| MPS-BG | 0.745<br>(0.011) | 0.755<br>(0.013) | 0.761<br>(0.032) | 0.698<br>(0.009) | N/A | 0.779<br>(0.023) | 0.771<br>(0.041) | **0.788**<br>**(0.032)** | 11.4% |
| MPL-FE | 0.958<br>(0.001) | 0.959<br>(0.001) | 0.989<br>(0.001) | 0.992<br>(0.001) | N/A | 0.995<br>(0.001) | 0.995<br>(0.001) | **0.998**<br>**(0.001)** | 0.6% |
| MPL-BG | 0.702<br>(0.005) | 0.715<br>(0.003) | 0.735<br>(0.010) | 0.733<br>(0.003) | N/A | 0.801<br>(0.004) | 0.811<br>(0.004) | **0.833**<br>**(0.003)** | 12.0% |
| HOIP | 0.895<br>(0.012) | 0.901<br>(0.009) | 0.767<br>(0.153) | 0.908<br>(0.008) | 0.907<br>(0.008) | 0.965<br>(0.005) | 0.964<br>(0.006) | **0.967**<br>**(0.004)** | 6.1% |
| NLHM | 0.633<br>(0.016) | 0.771<br>(0.019) | 0.784<br>(0.058) | 0.721<br>(0.013) | **0.904**<br>**(0.015)** | 0.827<br>(0.018) | 0.826<br>(0.018) | 0.828<br>(0.017) | 12.9% |

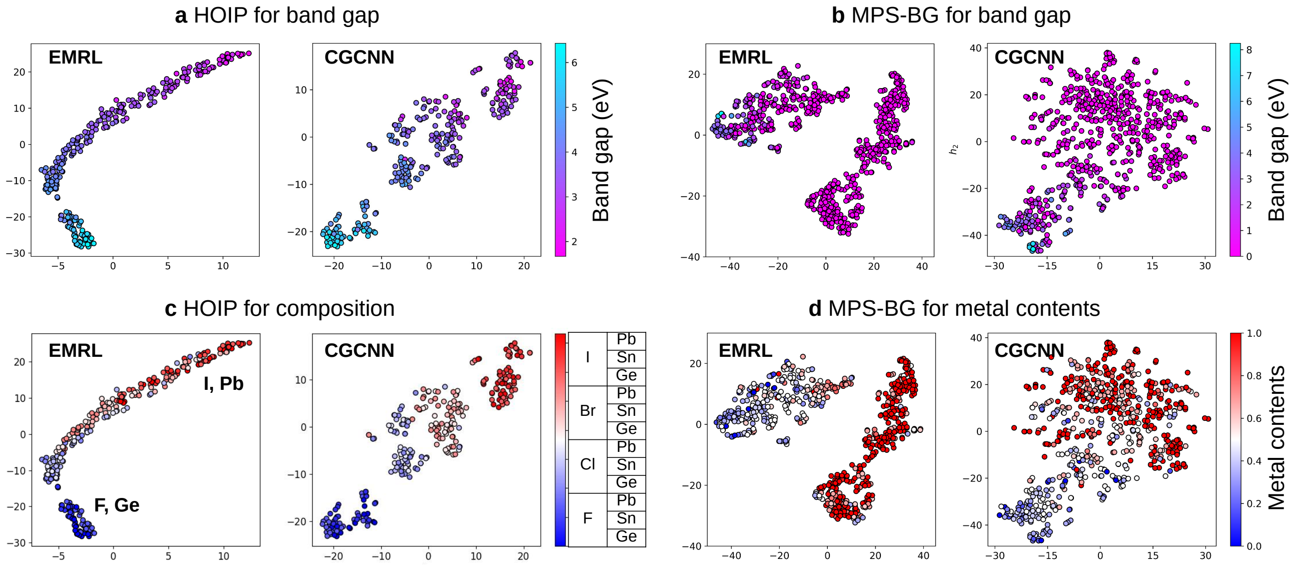

Figure 1: Two-dimensional t-SNE [30] visualization of the distributions of the materials in the embedding spaces of EMRL and CGCNN. The X and Y axes in the embedding distributions are two latent features that were calculated automatically through EMRL or CGCNN. Each point indicates the crystal structure of the material. **a** and **b**: Distributions of the materials in the HOIP and MPS-BG datasets. Each material was colored according to its band gap, where higher and lower band gaps were marked magenta and cyan, respectively. **c**: Distribution of the materials in the HOIP dataset categorized according to their compositions. **d**: Distribution of the materials in the MPS-BG dataset colored according to contents of metal atoms, which were calculated by the ratio of the number of metal atoms to that of all atoms in the unit cell.

In the MPS-BG dataset, both EMRL and CGCNN can distinguish the materials with higher band gaps, which are marked by magenta. However, the materials were roughly separated into two subgroups depending on their band gaps by the materials representation of EMRL as shown in Fig. 1**b**, but these subgroups cannot be easily recognized in CGCNN. This shows that EMRL can handle uncorrelated feature vectors and target values better than CGCNN. To understand the nature of two subgroups from EMRL representations, we interpreted them based on the metal contents in the materials, as the amount of band gap is relevant to the metal non-metal elements and metals are generally known as good conducting materials with no band gap. By displaying the content of metal atoms in the unit cell, we were able to observe that EMRL separated the MPS-BG dataset into two groups of the materials with high and low metal contents. More importantly, the self-organization by EMRL can be helpful in connecting relevant materials properties, which can be explained in the similar fundamental knowledge. In the embedding results on the MPS-BG dataset, we observed that EMRL learned the importance of chemical compositions rather than a direct mapping between the crystal structures and the band gaps. In this viewpoint, we drew the materials distribution in the embedding space of EMRL for their formation energies. As shown in Fig. 1 of SI, the materials were well-separated according to their formation energies even though EMRL was trained to predict band gap. In this sense, we extended the applicability of EMRL into transfer learning and experimentally evaluated the effectiveness of EMRL in the paper.

# References

[1] pymatgen. `https://pymatgen.org/`, 2020 (accessed December 4, 2020).

[2] Pytorch-geometric. `https://pytorch-geometric.readthedocs.io`, 2020 (accessed December 4, 2020).

[3] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. volume 27 of *Proceedings of Machine Learning Research*, pages 37–49, 2012.

[4] Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Phys. Rev. B*, 87 (18):184115, 2013.

[5] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98(14):146401, 2007.

[6] Pierre-Paul De Breuck, Geoffroy Hautier, and Gian-Marco Rignanese. Machine learning materials properties for small datasets. *arXiv:2004.14766*, 2020.

[7] Ivano E. Castelli, Falco Huser, Mohnish Pandey, Hong Li, Kristian S. Thygesen, Brian Seger, Anubhav Jain, Kristin A. Persson, Gerbrand Ceder, and Karsten W. Jacobsen. New light-harvesting materials using accurate and efficient bandgap calculations. *Adv. Energy Mater.*, 5(2):1400915, 2015.

[8] Chi Chen, Weike Ye, Yunxing Zuo, Chen Zheng, and Shyue Ping Ong. Graph networks as a universal machine learning framework for molecules and crystals. *Chem. Mater.*, 31(9):3564–3572, 2019. doi: 10.1021/acs.chemmater.9b01294. URL `https://doi.org/10.1021/acs.chemmater.9b01294`.

[9] Chi Chen, Weike Ye, Yunxing Zuo, Chen Zheng, and Shyue Ping Ong. Github repository of MegNet. `https://github.com/materialsvirtuallab/megnet`, 2020 (accessed December 4, 2020).

[10] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016. doi: 10.1145/2939672.2939785.

[11] Xing Chen, Li Huang, Di Xie, and Qi Zhao. Egbmmda: Extreme gradient boosting machine for mirna-disease association prediction. *Cell Death Dis.*, 9:3, Sep 2017. doi: https://doi.org/10.1038/s41419-017-0003-x. URL `https://www.nature.com/articles/s41419-017-0003-x`.

[12] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. *IEEE Conference on Compute Vision and Pattern Recognition (CVPR)*, pages 539–546, 2005. doi: 10.1109/CVPR.2005.202.

[13] Norman R. Draper and Harry Smith. *Applied Regression Analysis, 3rd Edition.* Wiley-Interscience, 1998. ISBN 978-0-471-17082-2.

[14] Olexandr Isayev, Corey Oses, Cormac Toher, Eric Gossett, Stefano Curtarolo, and Alexander Tropsha. Universal fragment descriptors for predicting properties of inorganic crystals. *Nat. Commun.*, 8(1):1–12, 2017.

[15] 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. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Mater.*, 1(1):011002, 2013. doi: 10.1063/1.4812323.

[16] Chiho Kim, Tran Doan Huan, Sridevi Krishnan, and Rampi Ramprasad. A hybrid organic-inorganic perovskite dataset. *Sci. Data*, 4:170057, May 2017.

[17] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[18] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[19] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*, 2017.

[20] Gyoung S Na, Hyunju Chang, and Hyun Woo Kim. Machine-guided representation for accurate graph-based molecular machine learning. *Phys. Chem. Chem Phys.*, 22(33):18526–18535, 2020.

[21] Gyoung S. Na, Seunghun Jang, Yea-Lee Lee, and Hyunju Chang. Tuplewise material representation based machine learning for accurate band gap prediction. *J. Phys. Chem. A*, 124:10616–10623, 2020. doi: 10.1021/acs.jpca.0c07802.

[22] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*, 2017.

[23] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65(6):386–408, 1958. doi: 10.1037/h0042519.

[24] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, pages 65–386, 1958.

[25] Daphna Rothschild, Omer Weissbrod, Elad Barkan, Alexander Kurilshikov, Tal Korem, David Zeevi, Paul I. Costea, Anastasia Godneva, Iris N. Kalka, Noam Bar, Smadar Shilo, Dar Lador, Arnau Vich Vila, Niv Zmora, Meirav Pevsner-Fischer, David Israeli, Noa Kosower, Gal Malka, Bat Chen Wolf, Tali Avnit-Sagi, Maya Lotan-Pompan, Adina Weinberger, Zamir Halpern, Shai Carmi, Jingyuan Fu, Cisca Wijmenga, Alexandra Zhernakova, Eran Elinav, and Eran Segal. Environment dominates over host genetics in shaping human gut microbiota. *Nature*, 555:210–215, Aug 2018. doi: https://doi.org/10.1038/nature25973. URL `https://www.nature.com/articles/nature25973?_ga=2.144797053.812497913.1541203200-1854234215.1541203200`.

[26] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. volume 5 of *Proceedings of Machine Learning Research*, pages 448–455. PMLR, 2009.

[27] Jonathan Schmidt, Mário RG Marques, Silvana Botti, and Miguel AL Marques. Recent advances and applications of machine learning in solid-state materials science. *Npj Comput. Mater.*, 5(1):83, 2019.

[28] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objectives. *Conference on Neural Information Processing Systems (NIPS)*, 2016.

[29] Hyun Oh Song, YStefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. *IEEE Conference on Compute Vision and Pattern Recognition (CVPR)*, 2017.

[30] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[31] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *International Conference on Learning Representations (ICLR)*, 2018.

[32] Aron Walsh. Principles of chemical bonding and band gap engineering in hybrid organic–inorganic halide perovskites. *J. Phys. Chem. C*, 119(11):5755–5760, 2015.

[33] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. *IEEE Conference on Compute Vision and Pattern Recognition (CVPR)*, 2017.

[34] Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Conference on Neural Information Processing Systems (NIPS)*, 2009.

[35] Tian Xie and Jeffrey C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.*, 120:145301, Apr 2018. doi: 10.1103/PhysRevLett.120.145301. URL `https://link.aps.org/doi/10.1103/PhysRevLett.120.145301`.

[36] Tian Xie and Jeffrey C. Grossman. Github repository of CGCNN. `https://github.com/txie-93/cgcnn`, 2020 (accessed December 4, 2020).

[37] Dahai Zhang, Liyang Qian, Baijin Mao, Can Huang, Bin Huang, and Yulin Si. A data-driven design for fault detection of wind turbines using random forests and xgboost. *IEEE Acess*, 6:21020–21031, Apr 2018. doi: 10.1109/ACCESS.2018.2818678. URL `https://ieeexplore.ieee.org/abstract/document/8329419`.