Noname manuscript No.
(will be inserted by the editor)

Is domain knowledge necessary for machine learning materials properties?

Ryan J. Murdock · Steven K. Kauwe · Anthony Yu-Tung Wang · Taylor D. Sparks

the date of receipt and acceptance should be inserted later

Abstract New featurization schemes for describing materials as composition vectors in order to predict their properties using machine learning are common in the field of Materials Informatics. However, little is known about the comparative efficacy of these methods. This work sets out to make clear which featurization methods should be used across various circumstances. Our findings include, surprisingly, that simple fractional and random-noise representations of elements can be as effective as traditional and new descriptors when using large amounts of data. However, in the absence of large datasets or for data that is not fully representative, we show that the integration of domain knowledge offers advantages in predictive ability.

Keywords Materials Informatics · Machine Learning · Featurization · Descriptors · Neural Networks

Introduction

In Materials Informatics (MI), composition-based Machine Learning (ML) entails the creation of a composition-based feature vector (CBFV) that represents materials based on expertly-curated element properties. Traditionally, descriptive statistics (average, range, sum, and variance) regarding the constituent elements represent the core of a CBFV scheme (see Figure 1). An exemplar of the CBFV method is the `Magpie`[1] descriptor. This domain-derived approach (CBFV) has been successfully employed in materials informatics studies in the literature[2–7]. Not only has this approach been successful, but the information it contains is also human-readable, potentially allowing for physically interpretable results.

Contra to the CBFV are data-driven techniques such as `CGCNN`[8], `mat2vec`[9], `SchNet`[10], `ElemNet`[11], etc. These represent a new philosophy. When featurization is reliant primarily on data, domain knowledge is less important. The representation of chemical systems is no longer relegated to expert opinion. When used within learning frameworks, these data-driven techniques allow for materials insight that may be outside of current scientific understanding. The removal of materials experts stands juxtaposed

R. Murdock, S. Kauwe, & T. Sparks
University of Utah, Materials Science & Engineering Department, Salt Lake City, UT, 84109
E-mail: sparks@eng.utah.edu

A. Wang
Technische Universität Berlin, Fachgebiet Keramische Werkstoffe / Chair of Advanced Ceramic Materials, 10623 Berlin, Germany

to traditional learning that uses hand-engineered materials representations, such as the classic CBFV.

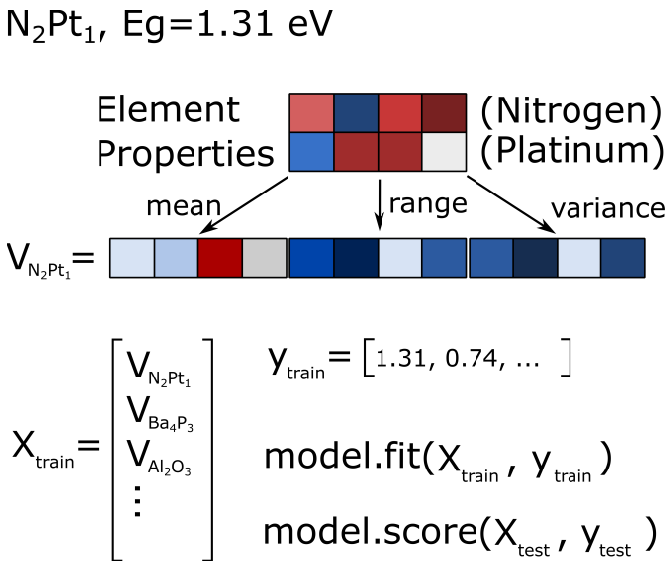

Fig. 1 Construction of a Composition-Based Feature Vector (CBFV).

Although a variety of data-driven approaches can be utilized, works such as `mat2vec` rely heavily on curated materials knowledge. For `mat2vec`, this knowledge comes in the form of materials science abstracts. Natural language processing techniques are applied to these abstracts, reportedly yielding vector encodings containing latent knowledge that is otherwise inaccessible to even the most expert materials scientists. This process results in an encoding of elemental information that is ultimately in a non-human-readable embedding. Although techniques such as vector arithmetic can be applied in an attempt to understand the relationships between various material embeddings, challenges arise in the deciphering of any governing chemistry underlying material properties.

On the other hand, data-driven but domain-agnostic approaches, such as `ElemNet`[11], use a `Fractional` representation (named as such because the initial element vectors, which are dummy encoded, contribute to a representation that identifies what fraction of the composition is made up of each element) to differentiate atoms based solely on their elemental identity. In the case of crystal systems, models such as `CGCNN`[8] use similar dummy element vectors embedded in a non-directed graph to represent crystal structure. These approaches use human-readable inputs and are designed to allow for limited inspection into model workings.

In Materials Informatics, interpretability of a model can be a very attractive quality. If a model's output cannot be understood at some level by humans, it may be more difficult to justify funding the synthesis of whatever materials it may recommend. Further, interpretability may be helpful in the pursuit of new physics and underlying chemical insights. The value of interpretability is often noted within the field of MI[6,12–14].

Regardless of the featurization approaches used it can easily be shown that different approaches to train-test splits, hyperparameter optimization, training time, random seeds etc., can drastically impact model performance. Therefore, drawing conclusions on featurization in the absence of a standardized hyperparameter schema is problematic. Furthermore, due to non-standardized data itself across the studies, we questioned whether or not the published results were even comparable. Therefore, in this brief com-

Table 1 Comparison of current featurization techniques (i.e., model inputs) used in MI

Feature	Data source	Expert knowledge	Hand-engineering	Domain-agnostic
`Magpie`	Element Chemistry	High	High	No
`Jarvis`	Element Chemistry	High	High	No
`Oliynyk`	Element Chemistry	High	High	No
`Atom2Vec`	Extant Compounds	Medium	None	Partial
`mat2vec`	NLP	Medium	None	Partial
`CGCNN`	Structural	Low	None to low	Yes
`SchNet`	Structural	Low	None to low	Yes
`ElemNet`	`Fractional`	Low	None	Yes

munication, we seek to set up a fair comparison. Using published works, we set up the following study using their described (or provided) featurization techniques:

1. Collect featurization schemes: in total, seven schemes comprising `ElemNet`[11] (`Fractional`), `Jarvis`[15], `mat2vec`[9], `Atom2Vec`[16], `Magpie`[1], `Oliynyk`[17]), and `Random`, where a unique random vector is generated and assigned to each element (see Table 1).
2. Establish six benchmark datasets, each representing a material property from the AFLOW database[18].
3. Standardize the train-test dataset split across each property, so that each model is trained and evaluated on the same featurized datasets.
4. Construct two prototypical neural network architectures, representing low-parameter (32×32) and high-parameter situations (512×512).
5. Evaluate performance metrics on each model-property-featurization combination (7 featurization schemes $\times$ 6 properties $\times$ 2 model architectures):
   (a) Benchmark model performance using mean absolute error (MAE), mean squared error (MSE), and r^2 scores (see Equations 1, 2, and 3)
   (b) Compare learning curves
   (c) Test generalizability (withhold elements during training)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

$$r^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{3}$$

Where:
$\bar{y}$ = the mean of all labels
y_i = the ith label
$\hat{y}_i$ = the ith prediction

Results & Discussion

Based on the generated learning curves (such as Figure 2), we find that the top performing featurizers have very similar predictive performance when given sufficient data. In fact, the `Fractional` representation often performs as well as or better than many other featurizers in the limit of "large" data (in agreement with previous work analyzing

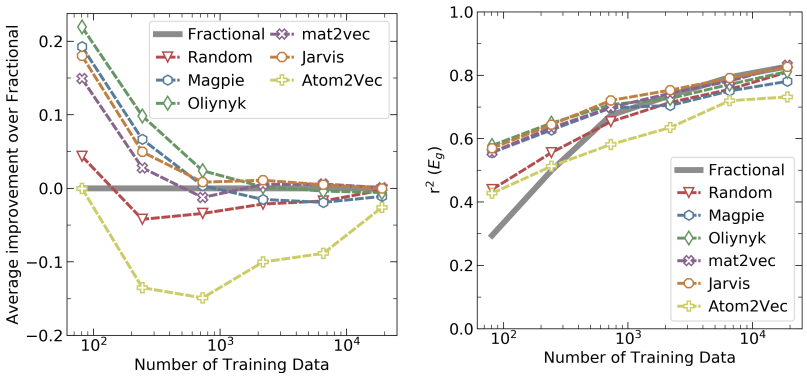

Fig. 2 Left: The difference in r^2 score of each featurization scheme compared to `Fractional` representation averaged over all properties at different amounts of training data. Right: Performance of each featurization scheme at different amounts of training data on band gap.

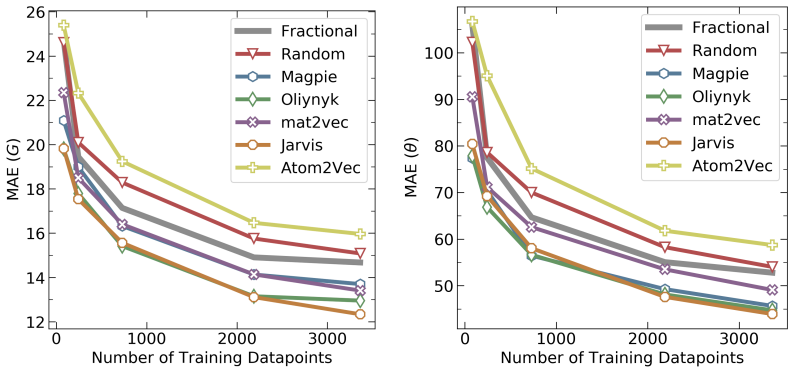

Fig. 3 Learning curves of shear modulus (Left) and Debye temperature (Right).

models that use little-to-no domain knowledge[19]). For instance, band gap and formation energy, which have significantly more data samples in the dataset, show extremely similar performance for `Fractional` featurization. However, when data is scarce, traditional CBFVs tend to outperform other descriptors, with `Jarvis` [15] and `Olyinyk`[17] often producing the best results (see Figure 3; for figures concerning other properties and metrics, see the GitHub[20]). Though it should be noted that `mat2vec`[9] can be seen performing quite well at mid-to-high levels of training data. In addition, CBFVs (and, to a lesser extent, `mat2vec`) generalize more effectively when elements are withheld from the dataset[21] (see Figure 4; the GitHub[20] contains additional figures for comparison). This suggests that projects interested in novel, out-of-dataset chemical systems may find more success with CBFVs.

The `Fractional` representation was treated as a baseline for comparison (Figure 2) because it is plausible as an effective featurization method to be both created and utilized by most data scientists. In contrast, assigning random vectors from a normal distribution to elements was seen as a bare-minimum to beat, although it resulted in better results than `Fractional` on some properties at high training dataset sizes.

Although a portion of the `Random` featurization scheme's success could be attributed to chance, we also speculate that it benefits from reduced sparsity compared to the `Fractional` representations along with reduced redundancy in comparison to CBFVs. While the sparsity of `Fractional` encoding allows for easy differentiation between elements, it may preclude the utilization of a portion of the network's first layer of parameters. The formula H represented sparsely as $[1, 0, ..., 0]$ will have a meaningful repre-

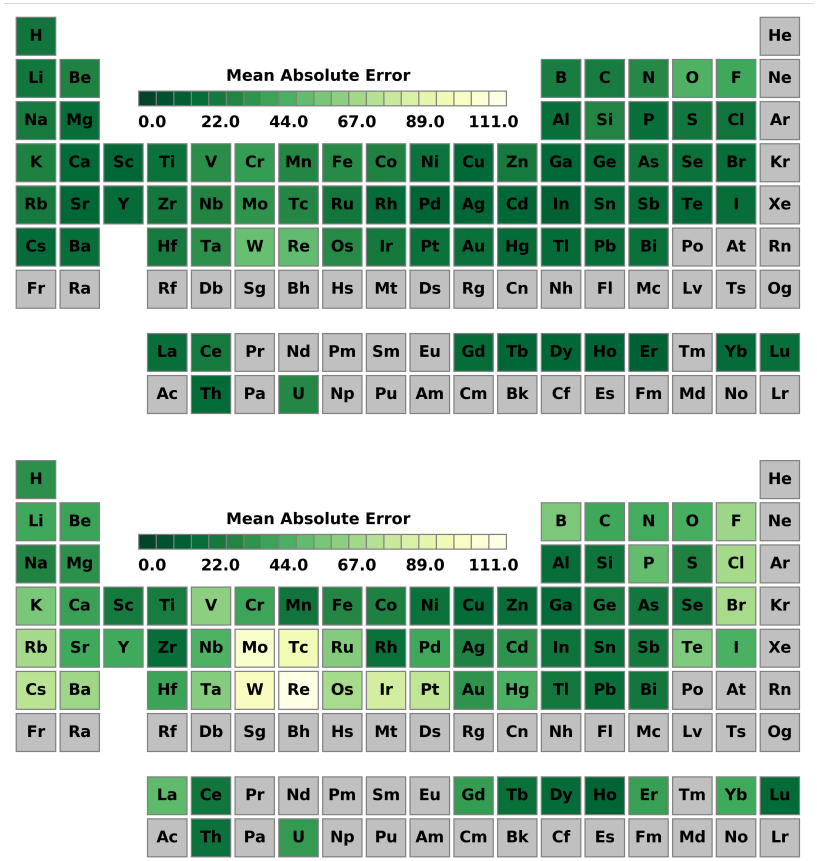

Fig. 4 Mean Absolute Error of bulk modulus predictions using `Oliynyk` (Top) and `Random` (Bottom) when compounds that include the shaded element are heldout during training and then tested on.

sentation of up to the number of units connected to the input, as all others are set to zero (or if scaled/normalized, some other singular value). Meanwhile, H featurized as random noise may allow for more complicated representations to arise early on in the network, utilizing more parameters in the earlier layer. This explanation is supported by the observation that the larger, (512×512) network showed more similar performance between `Random` and `Fractional` representations compared to the (32×32) network. In addition, any given column of a CBFV may have many repeats, and any given row may have many collinear features. Neither of these issues is present in sufficiently-long random noise.

We found that our implementation of `Atom2Vec`[16] was, on multiple properties, worse compared to both the `Fractional` baseline *and* `Random` vector approach. This demonstrates that a CBFV can potentially have adversarial effects on training, making it more difficult to learn some properties.

Conclusion

Our results in aggregate lead to the following recommendations: when using small data or data that may be applied to elements outside of the training set, traditional CBFVs are most likely to provide optimal results. Domain-agnostic approaches to featurization, such as `Fractional`, can be viable with large datasets, but should be compared to

standard CBFVs like `Oliynyk` and `Jarvis`. Although new, data-driven approaches are of interest, those studied here have yet to surpass CBFVs in terms of material property prediction with small data. In the case of projects with large amounts of data, the field may be advanced further through alternative activities to engineering new featurization schemes. These include creating and applying new architectures (such as multi-headed, self-attention networks[22]), collecting more data[23–25], and even investigating avenues outside of traditional property prediction, such as inverse design and semi-supervised learning. In contrast, projects with little data may see significant improvements through carefully selecting features.

Methods

Data Acquisition.

To begin comparing the various featurization methods' efficacy in predicting material properties, we collected various descriptors, first focusing on those that attempt to represent chemical information. The `Jarvis`[15] and `Magpie`[1] chemical featurizers were obtained from matminer[26]. In addition, the `Oliynyk`[17] chemical descriptor vectors were obtained from their author. The `mat2vec`[9] embeddings were obtained from its publication. The `Atom2Vec`[16] encoding was unreleased, and attempts to contact the authors were unsuccesful; consequently, we attempted to recreate it and validated our implementation on a similar dataset of elpasolites, obtaining similar validation mean absolute errors.

In addition, we included the following descriptors: `Fractional` representation, which simply conveys how many of each element is included in a given material's formula, and random noise vectors with a length of 200 sampled from a Gaussian distribution representing each element. Table 1 summarizes key differences between the various featurizers.

Band gap, formation energy, shear modulus, bulk modulus, Debye temperature, thermal expansion, and thermal conductivity data were then collected from the ICSD catalogue of the AFLOW database[18]. Duplicate entries were replaced by a single entry and each material property's formulae and ground-truth values were randomly partitioned into training, validation, and test sets (the dataset is available in the GitHub repository[20]). Note that, for this work, the associated Crystal Information Files (CIF) were discarded.

Table 2 Approximately 2940 models were created and trained to generate our learning curves, given the dimensions below along with six levels of training data per curve.

Properties (x7)
Band Gap
Formation Energy
Shear Modulus
Bulk Modulus
Debye Temperature
Thermal Expansion
Thermal Conductivity

Featurizers (x7)
`Fractional`
`Random`
`Magpie`
`Jarvis`
`mat2vec`
`Oliynyk`
`atom2Vec`

seeds (x5)
1
2
3
4
5

models (x2)
32×32
512×512

Model Training.

In order to understand which descriptors are optimal—and under which circumstances—we used artificial neural networks (ANNs) to map from these descriptors to various material properties. ANNs are models that take in some input, pass them through various summing and learned weighting operations, and produce a prediction. The difference between the predicted values and the known target values determines how these learned weights are adjusted in the process called backpropagation, which iteratively improves the model. The complexity of specific ANNs is determined by how many layers and units are included in the model, i.e., the model's architecture.

To avoid bias in the determination of any one feature as superior and to confirm that our results are not dependent on a single architecture, we chose to use two unoptimized ANN architectures for our models. Both architectures use two fully-connected layers. The first architecture has 32 units in each of its two layers, and in the second architecture, 512 units in each of its two layers. Both ANNs are trained using the Adam optimizer[27] with a learning rate of 1×10^{-3} and a batch size of 16.

Although ideally each model would be separately optimized for each property and feature, an exhaustive search for parameters would require massive amounts of computational time and power. As such, we leave the benchmarking of larger models to future works. We determined that two unoptimized models with varying complexity created the most fair comparison possible with the resources available. Our current learning curve analysis requires 2940 models to be trained (see Table 2).

The networks were trained on varying amounts of data from the training set until an early-stopping mechanism was triggered by a lack of improvement on the validation set. The highest validation metrics were collected for each descriptor on each material property to assess the impact of differing amounts of data. To verify these results, the models were trained on the full datasets and tested on previously unseen data. For more information, please refer to the GitHub repository accompanying this work[20].

Other Methods.

The analysis for held-out elements (Figure 4) was generated using a ridge regressor from `scikit-learn` on bulk modulus data with the formulae including each element within the dataset were withheld. The model was then tested on these withheld formulae. The code for these methods is also available on GitHub[20].

Acknowledgements

The authors gratefully acknowledge support from the NSF CAREER Award DMR 1651668. The authors also thank the Berlin International Graduate School in Model and Simulation based Research as well as the German Academic Exchange Service (program no. 57438025) for their financial support. Special thanks is given to Dr. Aleksander Gurlo for advising Anthony Yu-Tung Wang and encouraging his collaborative stay at the University of Utah.

The authors thank the creators of AFLOW for the creation of the database and for making the material properties available for this study. In addition, the authors express their gratitude to the open-source software community, for developing the excellent tools used in this research, including but not limited to Python, `Pandas`, `NumPy`, `matplotlib`, `scikit-learn`, and `TensorFlow`.

Conflict of interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, "A general-purpose machine learning framework for predicting properties of inorganic materials," *npj Computational Materials*, vol. 2, no. 1, 2016.
2. B. Meredig, E. Antono, C. Church, M. Hutchinson, J. Ling, S. Paradiso, B. Blaiszik, I. Foster, B. Gibbons, J. Hattrick-Simpers, A. Mehta, and L. Ward, "Can machine learning identify the next high-temperature superconductor? examining extrapolation performance for materials discovery," *Mol. Syst. Des. Eng.*, vol. 3, pp. 819–825, 2018.
3. Z. Cao, Y. Dan, Z. Xiong, C. Niu, X. Li, S. Qian, and J. Hu, "Convolutional neural networks for crystal material property prediction using hybrid orbital-field matrix and magpie descriptors," *Crystals*, vol. 9, no. 4, p. 191, 2019.
4. X. Li, Y. Dan, R. Dong, Z. Cao, C. Niu, Y. Song, S. Li, and J. Hu, "Computational screening of new perovskite materials using transfer learning and deep learning," *Applied Sciences*, vol. 9, no. 24, p. 5510, 2019.
5. 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*, vol. 89, no. 9, p. 094104, 2014.
6. R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, and C. Kim, "Machine learning in materials informatics: recent applications and prospects," *npj Computational Materials*, vol. 3, no. 1, pp. 1–13, 2017.
7. M. W. Gaultois, A. O. Oliynyk, A. Mar, T. D. Sparks, G. J. Mulholland, and B. Meredig, "Perspective: Web-based machine learning models for real-time screening of thermoelectric materials properties," *APL Materials*, vol. 4, no. 5, p. 053213, 2016.
8. T. Xie and J. C. Grossman, "Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties," *Phys. Rev. Lett.*, vol. 120, p. 145301, Apr 2018.
9. V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, and A. Jain, "Unsupervised word embeddings capture latent knowledge from materials science literature," *Nature*, vol. 571, pp. 95–98, 07 2019.
10. K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K.-R. Müller, "Schnetpack: A deep learning toolbox for atomistic systems," *Journal of Chemical Theory and Computation*, vol. 15, no. 1, pp. 448–455, 2019.
11. 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*, vol. 8, no. 1, pp. 1–13, 2018.
12. B. Meredig, "Five high-impact research areas in machine learning for materials science," 2019.
13. N. Wagner and J. M. Rondinelli, "Theory-guided machine learning in materials science," *Frontiers in Materials*, vol. 3, p. 28, 2016.
14. L. Ward and C. Wolverton, "Atomistic calculations and materials informatics: A review," *Current Opinion in Solid State and Materials Science*, vol. 21, no. 3, pp. 167–176, 2017.
15. K. Choudhary, B. DeCost, and F. Tavazza, "Machine learning with force-field-inspired descriptors for materials: Fast screening and mapping energy landscape," *Phys. Rev. Materials*, vol. 2, p. 083801, Aug 2018.
16. 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*, vol. 115, no. 28, pp. E6411–E6417, 2018.
17. A. O. Oliynyk, E. Antono, T. D. Sparks, L. Ghadbeigi, M. W. Gaultois, B. Meredig, and A. Mar, "High-throughput machine-learning-driven synthesis of full-heusler compounds," *Chemistry of Materials*, vol. 28, no. 20, pp. 7324–7331, 2016.
18. AFLOW, "AFLOW - automatic-flow for materials discovery," 2018. [Online; accessed 14-July-2019].
19. C. J. Bartel, A. Trewartha, Q. Wang, A. Dunn, A. Jain, and G. Ceder, "A critical examination of compound stability predictions from machine-learned formation energies." arXiv, 2020.
20. R. J. Murdock and S. K. Kauwe, "Online `GitHub` repository for *Is domain knowledge necessary for machine learning material properties*." `https://github.com/rynmurdock/domain_knowledge`, 2020.
21. S. K. Kauwe, J. Graser, R. Murdock, and T. D. Sparks, "Can machine learning find extraordinary materials?," *Computational Materials Science*, vol. 174, p. 109498, 2020.
22. A. Y.-T. Wang, S. K. Kauwe, R. J. Murdock, and T. D. Sparks, "Compositionally-Restricted Attention-Based Network for Materials Property Prediction," 2 2020.

23. F. Belviso, V. E. Claerbout, A. Comas-Vives, N. S. Dalal, F.-R. Fan, A. Filippetti, V. Fiorentini, L. Foppa, C. Franchini, B. Geisler, *et al.*, "Atomic-scale design protocols toward energy, electronic, catalysis, and sensing applications," 2019.
24. C. L. Clement, S. K. Kauwe, and T. D. Sparks, "Benchmark AFLOW Data Sets for Machine Learning," *Integrating Materials and Manufacturing Innovation*, 2020. Accepted, in press. DOI: `https://doi.org/10.1007/s40192-020-00174-4`.
25. A. Dunn, Q. Wang, A. Ganose, D. Dopp, and A. Jain, "Benchmarking Materials Property Prediction Methods: The Matbench Test Set and Automatminer Reference Algorithm." arXiv, 2020. accessed May 5, 2020.
26. L. Ward, A. Dunn, A. Faghaninia, N. Zimmermann, S. Bajaj, Q. Wang, J. Montoya, J. Chen, K. Bystrom, M. Dylla, K. Chard, M. Asta, K. Persson, G. Snyder, I. Foster, and A. Jain, "Matminer: An open source toolkit for materials data mining," *Comput. Mater. Sci*, vol. 152, p. 6069.
27. D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.