



CrabNet for Explainable Deep Learning in Materials Science: Bridging the Gap Between Academia and Industry

Anthony Yu-Tung Wang¹ · Mahamad Salah Mahmoud² · Mathias Czasny¹ · Aleksander Gurlo¹

Received: 22 October 2021 / Accepted: 5 December 2021 / Published online: 17 January 2022
© The Author(s) 2022

Abstract

Despite recent breakthroughs in deep learning for materials informatics, there exists a disparity between their popularity in academic research and their limited adoption in the industry. A significant contributor to this “interpretability-adoption gap” is the prevalence of black-box models and the lack of built-in methods for model interpretation. While established methods for evaluating model performance exist, an intuitive understanding of the modeling and decision-making processes in models is nonetheless desired in many cases. In this work, we demonstrate several ways of incorporating model interpretability to the structure-agnostic Compositionally Restricted Attention-Based network, CrabNet. We show that CrabNet learns meaningful, material property-specific element representations based solely on the data with no additional supervision. These element representations can then be used to explore element identity, similarity, behavior, and interactions within different chemical environments. Chemical compounds can also be uniquely represented and examined to reveal clear structures and trends within the chemical space. Additionally, visualizations of the attention mechanism can be used in conjunction to further understand the modeling process, identify potential modeling or dataset errors, and hint at further chemical insights leading to a better understanding of the phenomena governing material properties. We feel confident that the interpretability methods introduced in this work for CrabNet will be of keen interest to materials informatics researchers as well as industrial practitioners alike.

Keywords Materials informatics · Deep learning · Self-attention · Interpretability · Explainable AI · XAI

Introduction

Machine learning (ML) in materials informatics (MI) has received significant attention in the academic research world and is gaining widespread adoption [1–5]. More specifically, it has recently been extensively studied for its use in the research and design of novel inorganic materials [6–10]. This is enabled by three major developments: (1) the increasing number of material property datasets as well as the improvement in dataset quality and variety, (2) the rapid pace and development of new ML models tailored to

addressing different challenges in materials science (*e.g.*, regression, classification), supplemented by (3) the increase in available computing power and accessibility to ML and deep learning tools. The combination of these developments led to improved capabilities in the exploration and modeling of material properties in the academic world.

Classical ML methods (*e.g.*, linear regression, random forest, support vector machines) have successfully been used for the regression and classification of many material properties [11–17]. These methods usually rely on the featurization of the input chemical formulae into numerical features that are usable by the models. Typically, this is achieved through the use of a composition-based feature vector (CBFV), which uses descriptive statistics of the properties of constituent atoms in each compound to uniquely represent it [18]. Some common CBFV feature sets are Oliynyk, Magpie, Jarvis and mat2vec [11, 12, 19, 20]. Here, a distinction is made between physically derived CBFVs (with features based on measurable element properties) like Oliynyk, Magpie and Jarvis, and

✉ Anthony Yu-Tung Wang
anthony.wang@ceramics.tu-berlin.de

¹ Technische Universität Berlin, Institute of Material Science and Technology, Fachgebiet Keramische Werkstoffe/Chair of Advanced Ceramic Materials, Straße des 17. Juni 135, 10623 Berlin, Germany

² Department of Computer and Data Sciences, Case Western Reserve University, Cleveland, OH 44106, USA

computationally derived CBFVs (with features obtained from computational or deep learning models) like *mat2vec*. For some properties, additional features such as structural information and processing or measurement conditions are included to further improve model performance [2, 16, 21, 22].

In more recent years, deep learning (DL) models have gained widespread popularity in MI due to numerous advantages compared to classical ML methods. Some examples are ElemNet, CGCNN, MEGNet, DimeNet++, and ALIGNN [23–27]. More recently, graph neural network (GNN) models incorporating attention-based mechanisms such as CrabNet, Roost and H-CLMP have gained increasing popularity [28–30]. GNNs have shown improved performance compared to other DL models, particularly in the absence of structural information as model inputs. Another advantage of GNNs is that the inductive biases built into the model and the input data structure are more suited to the learning of material properties, since the interactions between the atoms in the compound can be modeled as weighted interactions between nodes in a graph. In CrabNet, for example, the atom representations are either based on a CBFV feature (*mat2vec* element vectors) or a non-CBFV feature (onehot element vectors) [28]. For the sake of clarity, the remaining text will use the acronym DL to refer to both deep learning (DL) and graph neural network (GNN) models and methods.

Unfortunately, while DL methods show superb performance in modeling material properties, the element features used by these models typically do not represent any measurable physical property of the elements themselves. Instead, the element representations are learned from the data during the model training process. Therefore, they do not directly provide useful information or insights that can be interpreted by humans. This is different from the CBFV representation typically used in classical ML, where the features represent properties of the elements which are known *a priori*, such as the atomic mass, first ionization energy, or number of valence electrons.

Despite the high performance of the DL models, there is a disparity between their extensive study in academic research and their limited adoption in the industry for the exploration of materials. We term this disparity the “interpretability-adoption gap”. One significant hurdle to the widespread adoption of the often “black-box” models is the lack of built-in methods for model interpretation. While there are established methods of evaluating model performance in academia [14, 31–33], those who are less familiar with DL typically require more intuition into how the models function before they can fully trust the results. Particularly in industry, where there is usually a lower risk tolerance compared to academia, findings based on black-box models and vague model evaluation criteria are not enough to justify making high-stakes decisions such as investing in new research [5,

34–38]. Tangible methods of investigating and understanding model decision-making processes are therefore required to facilitate their adoption in an industrial setting [39].

This led to the development of explainable AI (XAI), which aims to introduce methods for deciphering the internal workings of black-box models and thus enabling users to understand the modeling processes and results [39, 40]. Examples of XAI in research fields outside of MI include: visualizing word embeddings in natural language processing [41–43], inspecting decision-making processes in reinforcement learning [44–46], visualizing pixel importances [47, 48], or segmenting in computer vision [49, 50]. To date, however, XAI techniques have—with the exception of a few works employing classical ML—largely been underexplored for DL in the MI field [10, 51, 52].

Two common post-hoc model-agnostic methods for obtaining explainable models in classical ML are SHAP and LIME [39, 53–55]. Both of these methods are built on top of existing black-box models and use local feature perturbation to estimate the contributions from input features towards the predictions. Other models such as random forest, gradient boosting, and lasso regression inherently provide model interpretability via the use of internal feature importance metrics and (in some models) through bootstrap sampling and feature sampling [39, 51, 56]. Nonetheless, these techniques require that the individual features of the input data are meaningful and represent a measurable feature or physical property. This works in the domain of classical ML and when using a physically derived CBFV to featurize compounds; however, this is not the case for DL methods where the features typically do not reflect measurable values. Thus, these traditional ways of model interpretability fall short in use for the DL models.

Therefore, it is the goal of this work to explore how to increase model interpretability in DL models specifically for applications in MI. Here, we demonstrate how parts of the typically black-box modeling process can be communicated visually and in an interpretable way, using our attention-based model, CrabNet [28]. We have extended CrabNet’s architecture to enable intrinsic interpretability using several methods to be discussed below. In this regard, we lay the first bricks in the bridge spanning the interpretability-adoption gap between academia and industry. This will not only aid researchers in further developing complex models with interpretability in focus, but also promote the adoption of these modeling methods in the materials science industry.

Results and Discussion

The results of this study are described in five subsections. We first compare the element embeddings learned by CrabNet against other CBFV feature sets from the literature,

and show how chemical behavior and patterns in element properties can be learned entirely from the training data for each material property. We also show that the learned element representations are comparable to physically derived CBFVs. Secondly, as part of this analysis, we characterize the element prevalence imbalance in the datasets using the Shannon equitability index and relate that to the quality of the learned element embeddings. Third, we further examine how the element representations are successively updated using information about their chemical environment in the compounds, and how they may be used to gain additional insights about element behaviors in different environments. Fourth, we inspect how entire chemical compounds can also be adequately captured using the EDMs and subsequently visualized. We identify interesting trends in the compound representations relating the bond character and number of elements in the compounds to the material property and prediction error, and discuss how such visualizations can lead to additional understanding about the modeling process and the underlying materials chemistry. Lastly, we explore how the self-attention mechanism in CrabNet can be visualized in the form of videos and used to further examine the modeling process, leading to potential new insights about the chemical interactions within a compound. While we use the OQMD_Bandgap dataset to demonstrate the analyses, we note that similar analyses can be also carried out with any of the 28 materials datasets presented in this work.

Learning Meaningful and Per-Property Element Representations

Element representations were obtained as featurized CBFVs, which are fixed-length vectors where each element is uniquely described by the same set of features [12, 18].

For the Oliynyk, Magpie and mat2vec element property feature sets, we use the published vectors to represent the elements [18, 20]. For the CrabNet element representations, we extract the element vectors from the element-derived matrices (EDMs) at the output of the embedding layer (please refer to the CrabNet publication for architecture details [28]). We then examine the similarity between two element vectors x and y by computing the Pearson correlation coefficient r using Equation 1:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \quad (1)$$

where n is the number of features, x_i and y_i are the values of the i th feature, and \bar{x} and \bar{y} are the mean values of x and y , respectively.

The correlation r ranges from -1 to 1; the higher or lower the value of r is, the more correlated or anticorrelated are the features that describe the elements, respectively. A value of zero means that there are no correlations between the features of the elements. We compute the pairwise correlation coefficients between the element vectors for all elements and for all element property representations, and show these as heatmaps in Fig. 1. Note that the plots are cropped to the range of elements of the Oliynyk heatmap to aid comparison; please refer to supplementary Fig. S-1 in the supplementary information (SI) for the full heatmaps. In addition, interactive versions of the plots are provided in the SI.

Here, we can observe that element vectors based on the Oliynyk and Magpie CBFVs contain large regions of similar color in the heatmap. The regions of similar color indicate that the element representations are either highly correlated or highly anticorrelated with each other.

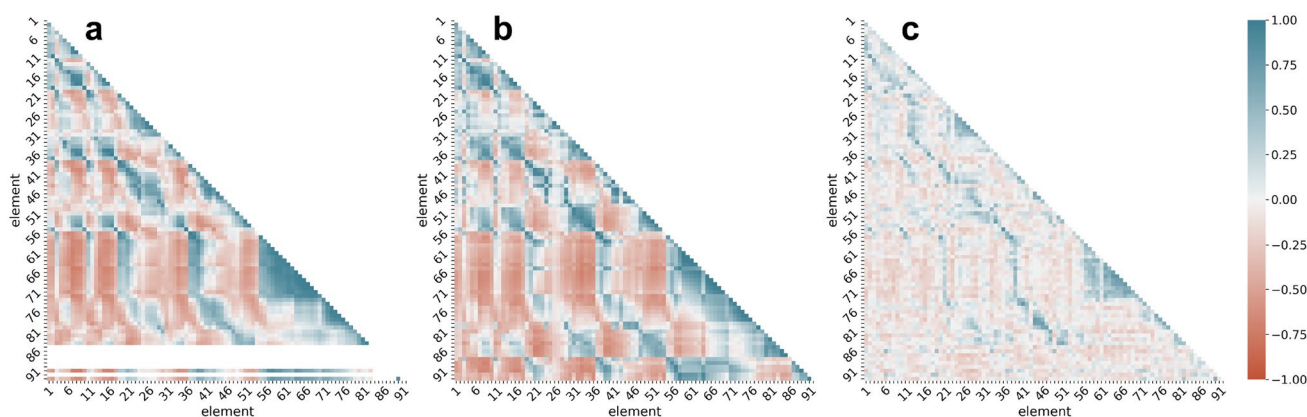


Fig. 1 Heatmaps of Pearson correlation matrices between element vectors featurized using **a** Oliynyk, **b** Magpie, and **c** mat2vec element property feature sets. The x - and y -axes are labeled with the atomic numbers. Each cell at coordinate (x, y) represents the correlation between the corresponding elements with atomic numbers x and y .

Blue represents a high correlation and red represents a high anticorrelation. For the interest of comparison, the heatmaps are truncated to the dimensions of the Oliynyk heatmap. Empty rows indicate that no element vector is available

Furthermore, these regions are very similar between the two CBFVs. This is expected, since the CBFV features are based on physical properties of the elements. Thus, elements with similar physical properties will be more correlated while dissimilar elements will be more anticorrelated. Accordingly, the large colored regions typically correspond to similarities and dissimilarities between elements from families in the periodic table, such as alkali metals, alkaline earth metals, transition metals, metalloids and reactive nonmetals.

On the other hand, the element vectors from a DL model such as `mat2vec` do not exhibit such prominent behavior. Overall, the elements show less correlation with each other, and—with the exception of a few areas (to be discussed in later sections)—do not show large continuous regions of similar color. This is due to the fact that the starting element representations in DL models are randomly initialized and are not based on physical properties of the elements. These vector representations of the elements are only updated by the model throughout the training process using the training data. Thus, the correlation patterns that can be observed in this figure represent distinct patterns that the DL model has learned solely from the provided data.

We also note that a different number of element vectors are recorded in the feature sets. For the Oliynyk and Magpie CBFVs, only the elements up to uranium and berkelium are reported, respectively, while vectors up to the element oganesson are provided by `mat2vec` (please refer to supplementary Fig. S-1 in the SI for the uncropped heatmaps). Particularly for the Oliynyk CBFV, some element vectors are missing, as visible by the empty rows in the heatmap. This disparity in the availability of element vectors between different CBFVs can be caused by reasons such as the instability or rarity of elements, lack of adequate information about the elements, or the inability to measure properties about the elements. The lack of element vectors in some material property feature sets can limit their applicability for certain tasks (such as when studying rare elements) and will be discussed in more detail in later sections.

In addition to learning element representations for a general purpose in materials science, such in the case of `mat2vec`, DL methods can also learn to relate element characteristics on a material property-specific basis. For example, element embeddings were extracted from the CrabNet and HotCrab models which were reproduced using the supplied model weights and the source code [57, 58]. The CrabNet and HotCrab models use `mat2vec` and `onehot`-encoded element features as the starting element representations, respectively. These features are then fine-tuned by the models for each of the 28 reported datasets. We extract one set of element embeddings from each layer of the models. Then, the Pearson correlation between the element vectors are calculated and shown in Fig. 2.

In this work, we use the `OQMD_Bandgap` dataset to demonstrate our findings. Additional example plots for other properties can be found in the SI. The OQMD datasets are widely used by researchers to evaluate model performance. For detailed information about the `OQMD_Bandgap` dataset as well as information and discussion about the calculated values, please see the literature [59–61].

Here, we can observe that both CrabNet and HotCrab are able to learn embeddings for each element of the periodic table, and that the correlations between the elements have a similar pattern, irrespective of the starting element representation (`mat2vec` or `onehot`). The observed correlation patterns are also similar to the `mat2vec` patterns as seen in Fig. 1c. The ability of both CrabNet and HotCrab models to learn similar element embeddings despite having drastically different starting representations is encouraging, and further suggests that domain knowledge is not necessarily required for element featurization if a sufficient quantity and quality of training data is available [18]. This finding is corroborated by the similarly good performance of both models across a wide range of material properties [28]. Interestingly, for deeper layers of the models (Fig. 2b and d), more intense correlation patterns between the elements emerge. This is likely attributed to the self-attention-based learning mechanism of the underlying CrabNet models. At each successive layer within the model, information about additional element-element interactions within the compound (*i.e.*, the chemical environment) are successively taken into account when updating the identity of an element within that compound. As a result, the deeper the layer within the model, the more complex the element interactions—and the element representations—become.

It is also interesting to note the diagonal and horizontal patterns which can be observed in all of the correlation matrices. For example, in Fig. 2d there is a 45-degree diagonal, blue line that can be seen in the correlation matrix starting at the coordinates (13, 31) (corresponding to the element pair (Al, Ga)) and continuing until (40, 58) (corresponding to (Zr, Ce)). This line highlights the well-known periodic law which states that elements with similar chemical properties fall into recurring periodic groups. Please refer to supplementary Fig. S-2 for the enlarged version of the annotated heatmap and for correlation plots for other material properties. Another observation is the triangular region of high correlation between (57, 57) and (71, 71), which indicates that the first-row elements of the *f*-block are highly similar to each other. A similar triangular region can be observed between (23, 23) and (29, 29), indicating similarities between some first-row elements of the *d*-block. Lastly, the vertical blue line starting at the coordinates (39, 57) and continuing to (39, 71) indicate the chemical similarities between yttrium and the first-row elements of the *f*-block. These and other patterns can also be observed in

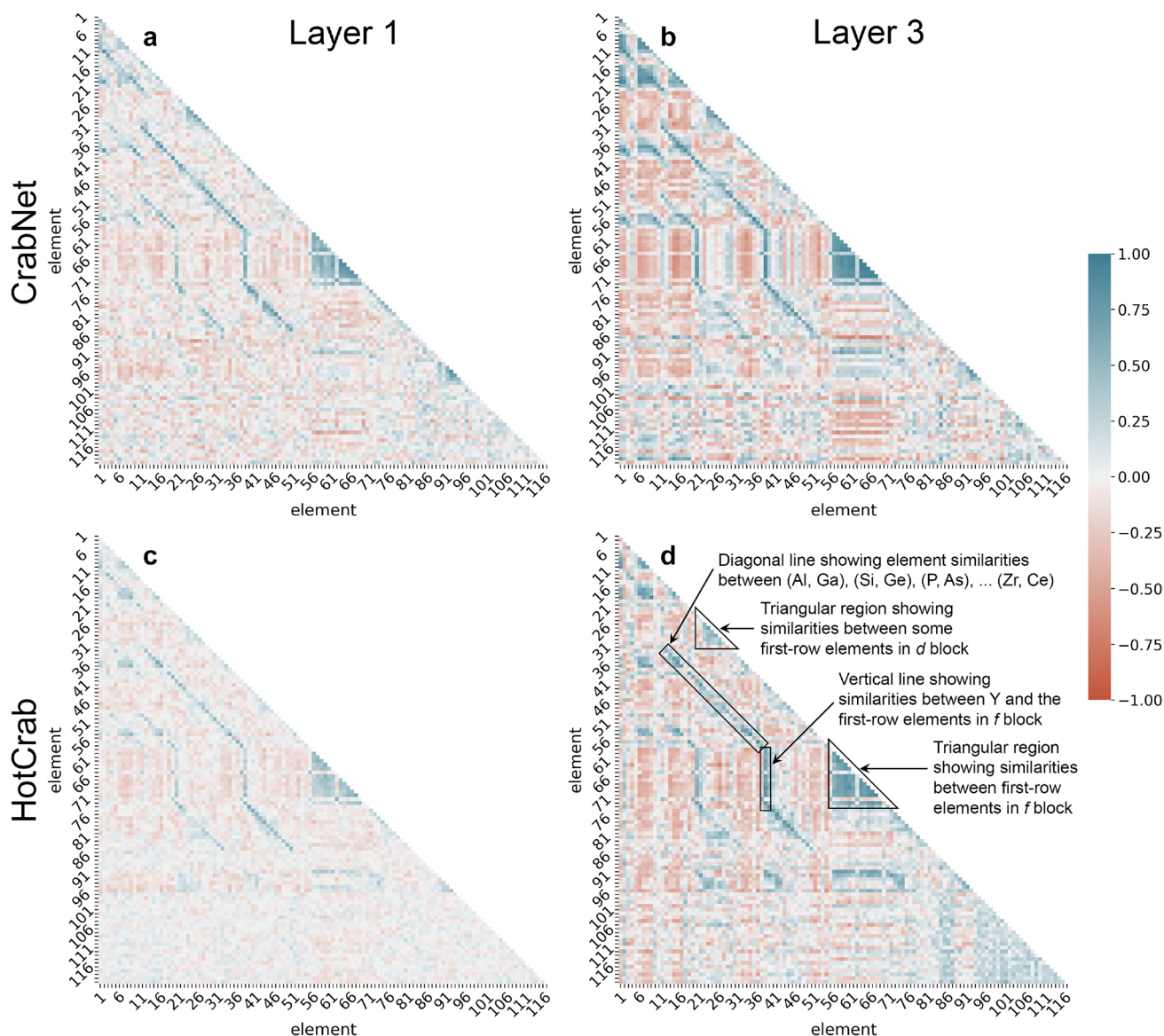


Fig. 2 Heatmaps of Pearson correlation matrices between element vectors extracted from CrabNet and HotCrab. These element representations are learned entirely from data. The x - and y -axes are labeled with the atomic numbers. Each cell at coordinate (x, y) represents the correlation between the corresponding elements with atomic numbers x and y . The top row (**a** and **b**) shows the correlations

between embeddings from CrabNet and the bottom row (**c** and **d**) from HotCrab. The left and right columns represent the embeddings extracted from the first and last layer of the models, respectively. Blue represents a high correlation and red represents a high anticorrelation. In **d**, some regions of interest are annotated

the Oliynyk and Magpie CBFVs in Fig. 1 as well. The ability of the CrabNet and HotCrab models to learn such chemical relationships which are comparable to hand-curated CBFVs based solely on the chemical formulae is exciting, and further reaffirms the finding that hand-engineering of features is not needed when training on big data [18].

Moreover, in Fig. 2c we observe a distinct “border” at the element plutonium (with atomic number 94), where the correlation coefficients between the elements suddenly decrease and the patterns become less pronounced.

Additional analysis of the OQMD_Bandgap dataset showed that it does not contain any compounds with elements past plutonium. Due to the fact that the element representations are learned purely by the model from the dataset, their quality depends heavily on the quality of the dataset. Since the model performance depends on the quality of the element representations, by extension, it also then depends on the dataset quality [32].

We define element prevalence as the number of times a certain element has appeared as part of the compounds

in a given dataset. When examining the OQMD_Bandgap dataset, we note that there is an imbalance in element prevalence, with oxygen and copper appearing almost 1.5 times to twice as often, and fluorine, chlorine, bromine and iodine appearing only less than 0.1 times as often as the majority of the other elements in the dataset, respectively. This imbalance in element prevalence is even stronger for other datasets such as the aflow__Egap, castelli, CritExam, mp_e_form and phonons datasets (see supplementary Fig. S-3 in the SI for some example element prevalence plots).

Quantifying Dataset Imbalance

The degree to which a dataset is imbalanced (otherwise referred to as its “evenness”) can be measured using the Shannon equitability index, which is a function of the Shannon entropy of the dataset [62–64]. Shannon entropy is widely used in information theory and can be used to characterize the degree of imbalance in a dataset [65, 66]. The Shannon entropy H is defined in Equation 2 as:

$$H(X) = - \sum_{i=1}^k P(x_i) \log P(x_i) \quad (2)$$

where X is the set of discrete variables $x_i \in \{x_1, \dots, x_n\}$, i is the class, $P(x_i)$ is the proportional abundance of x_i and k is the total number of classes in the dataset.

For a dataset \mathcal{D} of n data occurrences and k distinct chemical elements (classes), each with counts c_i , $P(x_i) = \frac{c_i}{n}$ and the Shannon entropy can thus also be written as Equation 3:

$$H(\mathcal{D}) = - \sum_{i=1}^k \frac{c_i}{n} \log \left(\frac{c_i}{n} \right) \quad (3)$$

For continuity, we note that when $c_i = 0$, it means that no data sample is related to class i in the dataset, and therefore the multiplicand within the summation is defined to be 0. Mathematically, $\lim_{p \rightarrow 0^+} p \log(p) = 0$. The maximum value of $H(\mathcal{D})$ is $\log(k)$. This value occurs when all element classes in the dataset are observed at the same frequency (*i.e.*, the dataset is completely balanced). Therefore, the Shannon entropy $H(\mathcal{D})$ is scaled by $\log(k)$ to finally obtain the Shannon equitability index $E(\mathcal{D})$, which is defined in Equation 4 as:

$$E(\mathcal{D}) = \frac{H(\mathcal{D})}{\log(k)} \quad (4)$$

$E(\mathcal{D})$ ranges between 0 for a maximally imbalanced dataset and 1 for a maximally balanced dataset. The Shannon equitability indices are calculated for the 28 datasets examined in this work and are presented in Table 1. A plot showing the same information can be found in the SI (supplementary Figure S-4). For more information about the datasets, please refer to the CrabNet publication [28].

As can be seen in the table, the datasets studied in this work are not equally balanced in terms of element diversity. The more imbalanced a dataset is in terms of the element prevalence in the chemical compounds, the less likely the models will be able to adequately learn about the elements and their environments. The element embeddings learned for the infrequent elements will therefore be weaker and will not

Table 1 Shannon equitability indices calculated from the training data splits of the 28 reported datasets. Datasets were taken from [28]

Material property dataset	Equitability	Material property dataset	Equitability
castelli	0.823	aflow__ael_bulk_modulus_vrh	0.948
dielectric	0.864	aflow__ael_debye_temperature	0.948
elasticity_log10(G_VRH)	0.953	aflow__ael_shear_modulus_vrh	0.948
elasticity_log10(K_VRH)	0.953	aflow__agl_thermal_conductivity_300K	0.940
expt_gap	0.931	aflow__agl_thermal_expansion_300K	0.944
expt_is_metal	0.930	aflow__Egap	0.920
glass	0.771	aflow__energy_atom	0.917
jdft2d	0.872	CritExam__Ed	0.914
mp_e_form	0.913	CritExam__Ef	0.914
mp_gap	0.916	mp_bulk_modulus	0.923
mp_is_metal	0.916	mp_elastic_anisotropy	0.921
phonons	0.909	mp_e_hull	0.897
steels_yield	0.959	mp_mu_b	0.897
		mp_shear_modulus	0.921
		OQMD_Bandgap	0.976
		OQMD_Energy_per_atom	0.976
		OQMD_Formation_Enthalpy	0.976
		OQMD_Volume_per_atom	0.976

be able to capture as much information about these elements as compared to more frequently occurring elements. This leads to the observed weak correlation patterns between the less frequently seen elements beyond a certain cutoff atomic number in the datasets, as discussed earlier for Fig. 2.

If the weakly learned elements are then encountered during inference time, the model will not be able to make an adequate prediction using the elements' representations. Additionally, if certain elements or element combinations appear more frequently (majority classes) in the datasets as compared to other elements or combinations (minority classes), the model may be biased to better capture the behavior of majority classes at the expense of sacrificing performance on the minority classes. Such a dataset bias may appear in computational or experimental datasets due to the fact that some elements are more commonly studied for certain material applications. On the other hand, certain elements (*e.g.*, rare or unstable elements) naturally occur less frequently and therefore are also contained in fewer compounds and datasets. Certain elements such as noble gases also rarely form compounds with other elements and are therefore rarely reported in materials datasets.

It is therefore important to implement data processing and modeling techniques to address biases as a result of dataset imbalance. Some example techniques include dataset re-sampling, generating synthetic data for imbalanced classes, implementing weighted loss functions that penalize errors for minority classes more, or using alternative loss functions and metrics to evaluate model performance [64, 67, 68]. Additionally, the model architecture can also be tailored to address dataset bias, and certain types of models (such as those based on self-attention or guided attention architectures) have an increased robustness against dataset bias [69, 70].

Lastly, it is worthy to note that while most DL models learn element representations from structured materials datasets, methods such as word2vec and mat2vec use text mining and other natural language processing (NLP) techniques to learn the element embeddings from academic publications [20, 71, 72]. The data present in publications covers a much longer time period and contains a higher diversity in terms of types of compounds, material properties and applications studied. These data are in unstructured form and therefore cannot be used as training data for DL methods such as CrabNet; however, they can easily be used for word2vec and mat2vec. Therefore, these text mining methods are able to learn from a much larger corpus of materials data and are not restricted by the availability of structured datasets. Accordingly, DL models such as CrabNet can benefit by using the pre-trained mat2vec element embeddings and fine-tuning them to new tasks, thereby minimizing the impact of missing elements in the training dataset.

Capturing the Influence of Chemical Environments on Element Representations

In addition to learning the representations of each element, CrabNet and HotCrab can also capture the behavior of the elements when they are present in different chemical environments. Figure 3 shows the two-dimensional projections of the element vectors corresponding to the silicon atom from 2374 different silicon-containing compounds within the OQMD_Bandgap test dataset. The silicon vectors are extracted from the transformed EDM tensors from HotCrab (a onehot-featurized version of CrabNet) and show the transformation of the silicon representations after they are passed through the three successive self-attention layers. For visualization, the vectors are projected down to two dimensions using the uniform manifold approximation and projection (UMAP) method [73]. The resulting points are plotted and colored by three parameters: (1) the fractional abundance of the element silicon in the compound, (2) the predicted property value of the compound (in this case, band gap), and (3) the oxidation state of silicon as predicted by Pymatgen [74]. For more information, please see the Methods.

As can be seen in the plots from the first layer (first row), there is a large number of distinct point clusters, with one major cluster near the center, two medium clusters above and below the center cluster, and many smaller clusters consisting of a few points. The larger clusters are formed because the initial representations of the silicon atoms are very similar to another (due to the learned element embedding of silicon). The similar silicon vectors are thus projected through UMAP into coordinates that lie close together, even though the silicon atoms are present in different chemical environments.

We can observe as well that the clustering in layer one is mostly attributable to the fractional amount, since each cluster consists primarily of points with the same fractional silicon amount. After the second layer, we observe that the points start to become separated into different and recognizable clusters. The clusters are no longer identifiable entirely based on the fractional amount of silicon, and clusters based on the predicted band gap value of the compound and oxidation state of silicon start to emerge. By the end of the third and last layer, we can observe four clusters that are distinguishable by the fractional amount of silicon, the predicted band gap, and the oxidation state of silicon (the clusters are outlined in Fig. 3, bottom left).

More specifically, we observe that the cluster at the bottom-right side of the plot consists mainly of silicon with a fractional amount of around 0.15 to 0.3 (with a few points reaching 0.5), whereas the cluster near the bottom-left contains almost exclusively of silicon with fractional amounts of 0.5 plus a few points above 0.5. The cluster near the top contains regions of silicon with fractional amounts between

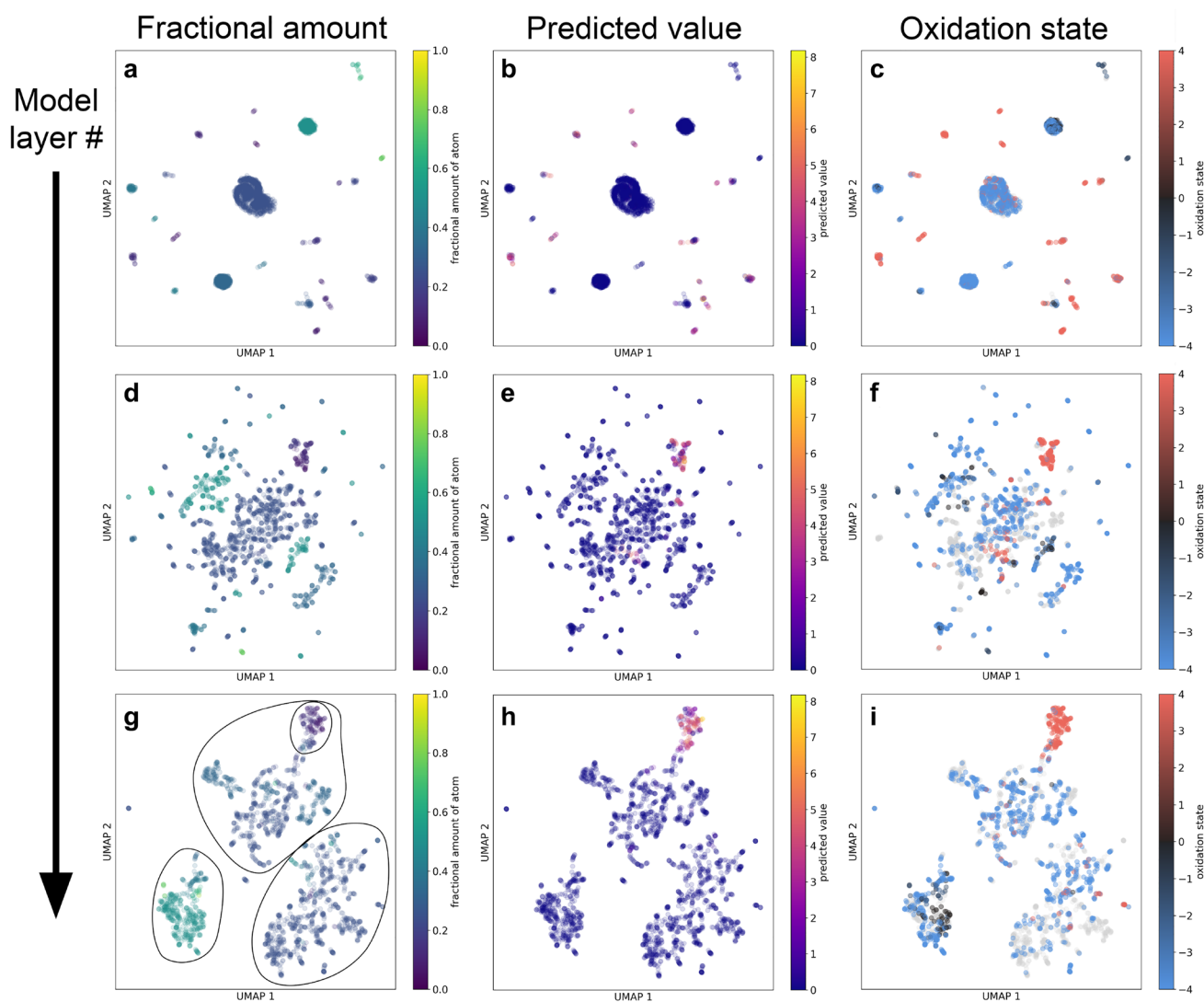


Fig. 3 Vector representations of the silicon element in 2374 different chemical environments and at different layers of the HotCrab model. Each point shows the model-internal representation of the silicon atom, after the information regarding the other atoms in the chemical environment have been introduced via HotCrab through the three attention layers (top row to bottom row). The points are colored by:

(left column) the fractional abundance of silicon, (center column) the predicted value of the compound, and (right column) the predicted oxidation state of silicon, where gray points indicate that the oxidation state was unable to be predicted. Four clusters are outlined in the bottom-left plot

0.3 to 0.4 near the left and right, and around 0.2 to 0.3 in the middle. Near the top of this cluster, a smaller cluster is highlighted which consists mainly of silicon instances with low abundance, between 0.2 and 0. Please note that interactive versions of these plots can be found in the SI together with another example visualization plotted for the element chromium (supplementary Fig. S-5).

In the predicted value plot of the last layer, we observe that only the small cluster near the top contains the silicon element in compounds with a nonzero band gap. Similarly, when examining the oxidation state plot, we note that while most clusters contain a mixture of silicon atoms in several oxidation states, the same cluster near the top consists almost

exclusively of silicon atoms in the +4 oxidation state and very few atoms in other oxidation states. Closer examination reveals that this cluster consists primarily of silicate materials such as Ca_2SiO_4 , CaMgSiO_4 , MgMnSiO_4 , Li_4SiO_4 , $\text{Sr}_3\text{MgSi}_2\text{O}_8$, $\text{Li}_2\text{MgSiO}_4$, and others. Interestingly, while some compounds with silicon in the +4 state are visible in other clusters, these compounds have a zero band gap. This suggests that additional interactions between the elements were captured by HotCrab which lead to these compounds being correctly clustered together with other compounds with zero band gap.

These element behavior plots suggest that for silicon-containing compounds in the OQMD_Bandgap dataset,

the fractional amount and the oxidation state of the silicon atoms are important factors that together determine the band gap of the compounds. By cross-referencing the three plots, we can identify trends between the fractional amount and oxidation state of silicon and relate this information to the predicted band gap of the compounds. On the other hand, the clustering also suggests that there are other interactions between the elements in a compound which are currently not highlighted by the selected properties in Fig. 3. It is our expectation that by examining these interactions, additional insight about the modeling process and element representations can be gained. Moreover, the findings from examining internal representations of elements in this way may suggest additional studies to further improve the understanding of the underlying phenomena governing materials behaviors. Note that while these visualizations were generated using HotCrab, similar results can be obtained using the CrabNet model.

Capturing Globally Unique Representations of Chemical Compounds

In addition to examining the behavior of individual elements in different chemical environments, we can also visualize all of the compounds in a given dataset to uncover additional insights. We extract the internal vector representation of all of the 51242 compounds in the OQMD_Bandgap test dataset from the last self-attention layer of HotCrab, perform dimensionality reduction using UMAP and finally visualize the compounds as shown in Fig. 4. In addition to coloring the plots by the predicted value, prediction error, and number of distinct elements for the compounds, we also highlight the chemical trend between ionic to covalent bonding character within the compounds. This trend is revealed by calculating and visualizing the standard deviation of the Pauling electronegativities of the constituent atoms σ_χ in a given compound [75] according to Equation 5:

$$\sigma_\chi = \sqrt{\frac{\sum_{i=1}^n (\chi_i - \bar{\chi})^2}{n-1}} \quad (5)$$

where χ_i is the Pauling electronegativity of each element i in the compound (totaling n elements), and $\bar{\chi}$ is the average electronegativity of all elements in the compound. A higher σ_χ signifies a more ionic bonding character, and a lower value signifies a more covalent bonding character.

Many clusters with varying sizes are visible in the figure. Some clusters are placed further apart, while some clusters are closer to, or are overlapping other clusters. In particular, the outlined cluster near the right of the figure is of particular interest. This is the only cluster where the compounds with a nonzero band gap are located, as is visible from Fig. 4a. Additionally, it is also within this cluster that

HotCrab makes the largest errors when predicting the band gap value, as seen in Fig. 4b. For the other compounds, the prediction errors of HotCrab are close to zero. Even though a small proportion of model predictions have larger errors, the overall model performance is very good and is comparable with, or better than, other state-of-the-art models [28]. This superior performance of CrabNet and HotCrab models when predicting properties with a defined cutoff (such as the cutoff of 0 eV in this case for band gap) is likely attributed to the prediction of element-logits in the modeling process. These element-logits are used to weight the final model predictions in CrabNet and HotCrab to improve the model accuracy [28].

Notably, we also observe from Fig. 4a, c and d that the band gap only partially depends on the bond nature of the compound and on the number of unique elements in the compound. While most of the compounds in the cluster of interest exhibit more ionic bond characters, there are also other clusters with similar bond character that do not have a nonzero band gap. Similarly, it appears that the compounds with a nonzero band gap mainly contain four or five unique elements; however, there are also other compounds with these numbers of unique elements which have a zero band gap.

Here, we do note that while UMAP can reveal structures and patterns within high-dimensional data, it generally emphasizes local structure at the expense of global structure. Therefore, for the UMAP visualizations shown in this work, it is more appropriate to interpret the local structure (*e.g.*, the elements or compounds present within individual clusters in Fig. 3 and 4) than the global structure. While the number of local neighbors considered can be specified as a hyperparameter in UMAP, a trade-off is made between preserving local versus global structure. Therefore, the distances between elements and compounds within a single cluster are more meaningful than inter-cluster distances in the UMAP visualizations. Lastly, we note that while these visualizations were generated based on the test dataset using HotCrab, similar results can be obtained using CrabNet or the training dataset.

Visualizing the Training Progress

Beyond visualizing the element and compound representations from CrabNet after training, it is also possible to access the self-attention matrices of the CrabNet encoding layers to observe the model learning process during training. The attention matrices (commonly referred to as the attention maps) contain information regarding how each element (rows) is influenced by all other elements in the compound as well as itself (columns). The values in the attention maps are the attention scores and are used in the encoder to update the element representations. An attention score of

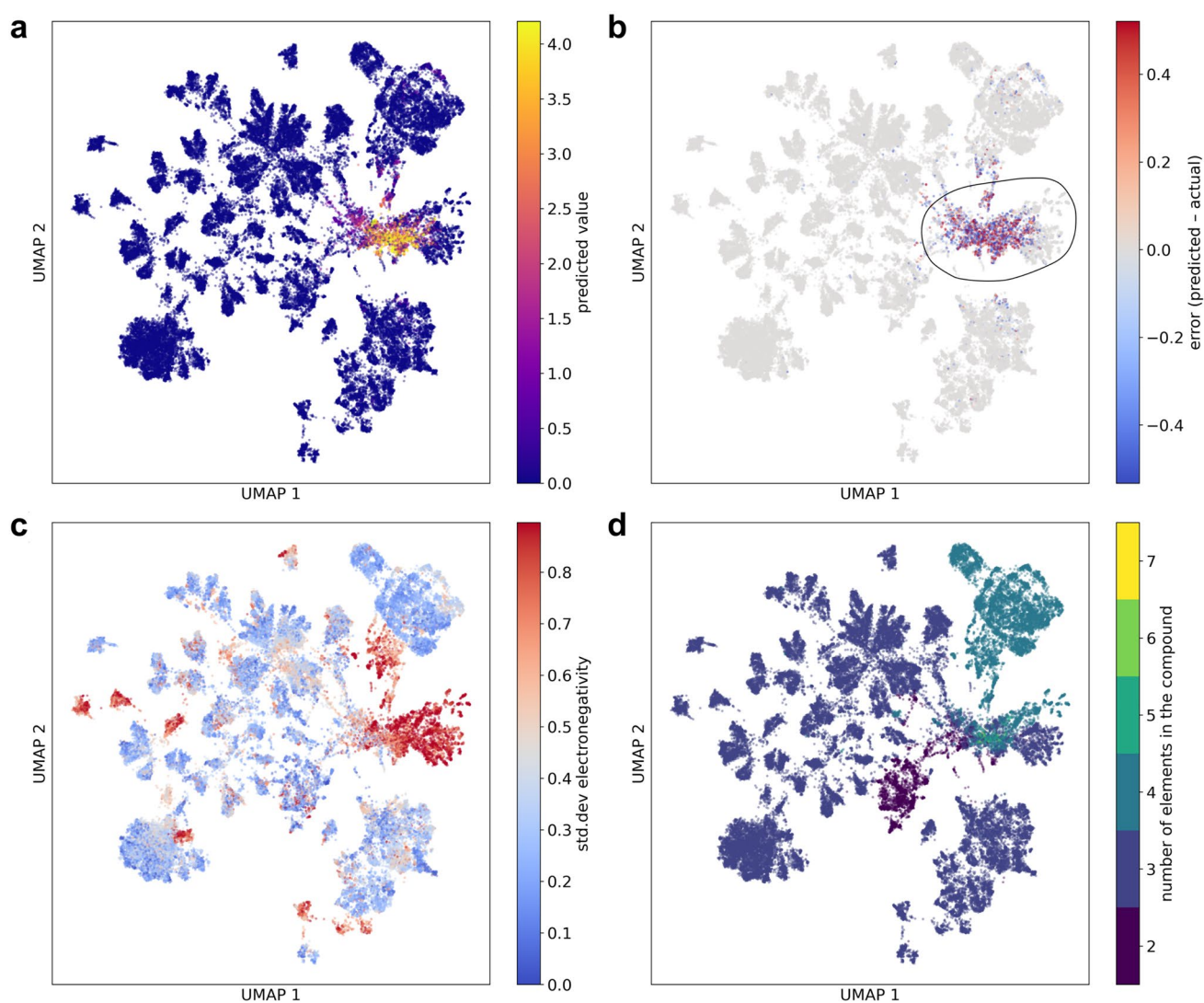


Fig. 4 Global representations of the 51242 compounds in the OQMD_Bandgap test dataset, extracted from layer three of HotCrab, embedded down to two dimensions using UMAP and colored by the parameters: **a** the predicted value of the compound (band gap); **b** the prediction error ($\hat{y} - y$); **c** the bond character of the compounds rang-

ing from more covalent (blue) to more ionic (red) as measured by the standard deviations in the Pauling electronegativities of the constituent elements; and **d** the number of distinct elements in the compound. A cluster of interest is outlined in the plot at the top-right

zero means that the element in the column is completely ignored when updating the element's representation in that row. Conversely, a score of one means that the entire update is based solely on that column's element.

In the CrabNet publication [28], example attention maps were shown for compounds after the model has finished training. Here, we extend this approach by visualizing the CrabNet attention maps during the model training process in the form of attention video clips (see SI files for examples). This is achieved by saving the attention matrices from the model encoder layers after every mini-step in the training process and generating a video to show the learning progress. Fig. 5 shows a snapshot of two example attention

videos obtained at the end of model training. The attention maps from the first encoding layer of CrabNet are plotted as heatmaps in the left column, while the right column shows the predicted values from the model against the target value at every mini-step. This process is performed at every mini-step in the training process, and the resulting plots are merged into a video clip which shows the learning progress of the model throughout training.

From the attention maps, we can observe that some elements are considered less relevant in the determination of the material property, whereas some elements are considered very relevant. Also we can note that individual attention heads pay attention to different element-element interactions

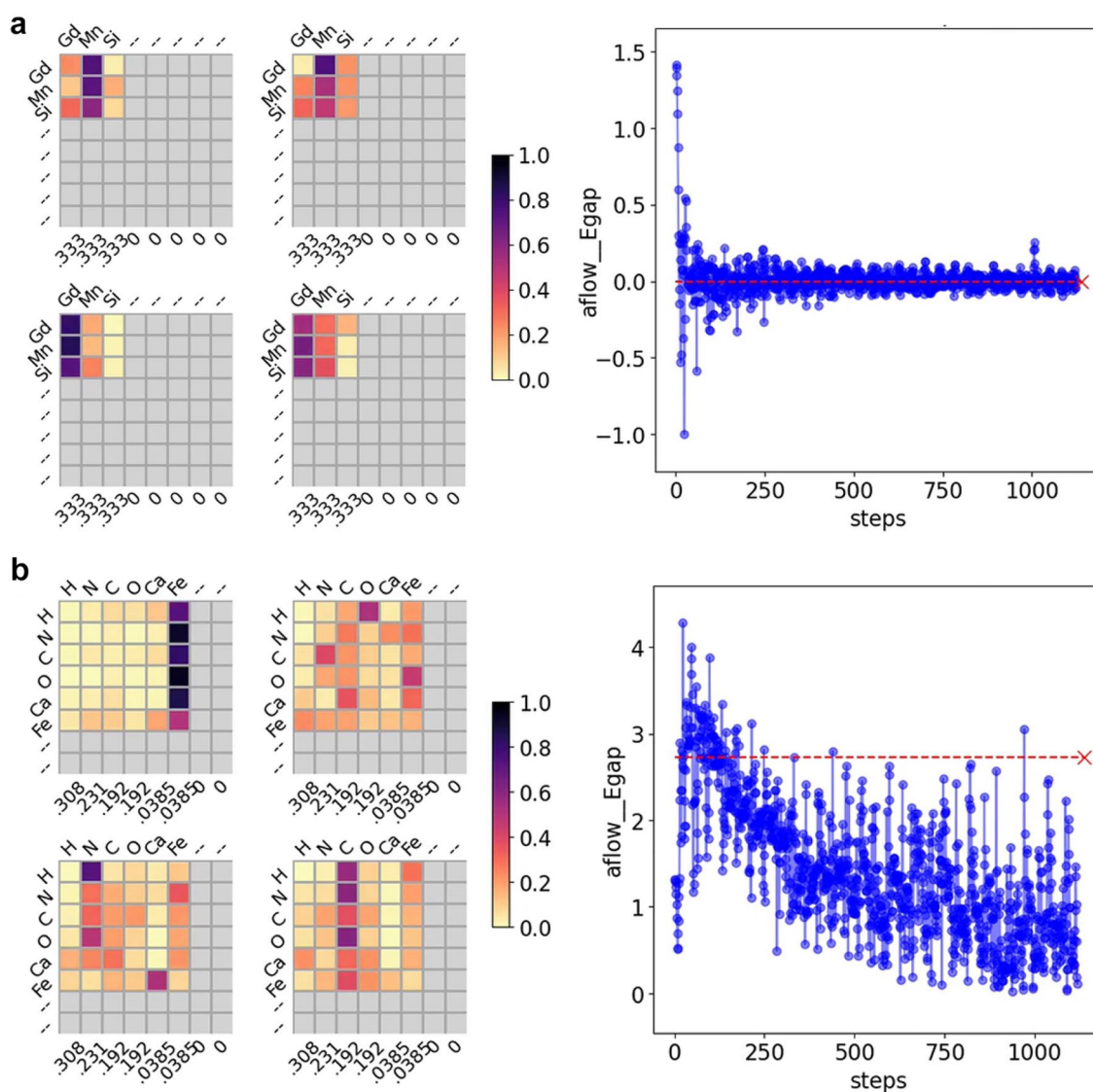


Fig. 5 Snapshots of attention videos for observing the training progress of CrabNet using two example compounds **a** $\text{Gd}_1\text{Mn}_1\text{Si}_1$ and **b** $\text{C}_5\text{Ca}_1\text{Fe}_1\text{H}_8\text{N}_6\text{O}_5$ from the validation data split of the aflow_Egap dataset. The left plots show the attention maps of the four attention heads at the first attention layer, where the x axis of each heatmap is

labeled with the fractional amount of the elements and the other axes are labeled with the element symbol. The right plots show the model predictions (blue) for the compounds, evaluated after each training mini-step throughout the whole training process. The true property value (target) is represented with the red “X” and the dotted line

in the compound, as is visible by the significantly different attention patterns in the plots. Throughout the training process, the attention pattern for each head remains relatively fixed after a few mini-steps, indicating that the model discovers a pattern for recognizing inter-element interactions early on in the training process, which it then continues to refine as more training steps are taken.

For the top compound, we can observe that while the model initially over- and underestimates the property value early on in the training, it learns to correct the error and finally achieves a low prediction error towards the end of

training. Conversely, for the bottom compound, we observe that while the model initially correctly estimates the property value of the compound, the predicted value decreases and the estimation error increases throughout training, with the error finally plateauing towards the end of the training. By examining the attention heatmaps for this compound, we notice that attention head 1 shows a significantly different behavior as compared to the other attention heads. It dedicates almost all of its attention to the element iron, while the other attention heads capture many more inter-element interactions. It may be interesting to investigate further to

find out if CrabNet is misrepresenting the interactions from the iron element with the other elements and thus making the prediction error, or if another phenomenon is contributing to the prediction error on this compound.

By observing the element groups and inter-elemental interactions that CrabNet pays attention to for each material property throughout the training process, we may be able to gain additional insight about which relevant elements and interactions contribute significantly to the material property. Similarly, in the case where the model does not make a good property prediction or fails to learn a specific material property, these attention videos can be informative in showing when, where, and how the model fails. Additionally, since the element representations in a compound are updated according to the attention scores, it would be interesting to train CrabNet on material properties where the property has a high sensitivity to changes in elemental prevalence. An example of this is in the case of dopants, where a small change in the dopant amount can significantly influence a material's electrical [15, 76, 77], mechanical [17, 78–80], and thermal properties [81–84]. Finally, it may be interesting to expand the studied materials to include co-doped materials and use the attention videos to visualize the complex inter-elemental interactions between the co-dopants and the host elements.

Conclusion

In this work, we examined the CrabNet model through the use of several built-in model interpretability methods in order to visualize the data featurization and modeling process. We demonstrated that CrabNet can adequately capture the chemical behavior of compounds in a dataset by using the vector representations of their constituent elements. The element representations can be learned entirely from the training data on a per-property basis, and contain rich information about the elements and their chemical trends. Additionally, we examined dataset imbalance, its relation to the quality of learned representations, and the limitations that imbalanced datasets may ultimately impose on the modeling processes.

The element and compound vectors can be projected using UMAP into distinguishable clusters which can then be visualized and characterized by the element stoichiometry, local chemical environment and oxidation state of the elements, or by the bond behavior of the compounds. Lastly, the examination of the self-attention matrices during model training in the form of attention videos can be used to further understand the modeling process, debug potential model or dataset errors, or gain additional insights about chemical interactions within a given compound.

The model interpretability techniques presented in this work will enable materials science practitioners to not only visualize a specific element's behavior within different chemical environments, but also to obtain a global view of the chemical compounds, behaviors and trends within a larger dataset. The ability of CrabNet to adequately model and express the complex chemical behaviors and interactions of elements and compounds based solely on learning from data is encouraging. With the addition of model interpretability methods to CrabNet, the findings and intuitions presented in this work may lead to further insightful and interesting research. Specifically, we believe that follow-up works may fall into one of these three general directions:

1. Learning and representing elements and compounds.

Our work has shown that it is possible to visualize CrabNet's internal representations of elements and compounds via techniques such as UMAP. However, it would be interesting to further investigate why CrabNet's representations of some of these elements or compounds lead to them being placed into the same cluster or not, despite the fact that these elements and compounds are similar to each other in terms of identity and/or chemical environment. This may also be combined with a more detailed examination of the attention videos and how the attention mechanism in CrabNet leads to the updating of the element representations for each compound.

2. Examination of individual attention head behaviors.

This work used the EDM (element-derived matrix) data from CrabNet to examine the element and compound representations within CrabNet. CrabNet utilizes four self-attention heads to model element-element interactions, the results of which are then concatenated and transformed back to an updated EDM matrix. As such, the EDM is a pooled representation of the compounds. It would be interesting to further examine the per-head modeling of the compounds, as it has been shown that each head can capture different types of inter-element interactions and thus may give additional insight to the modeling process within CrabNet.

3. Discovery of additional inter-element interactions.

From the analyses presented in this study, it is clear that while some changes in the material property (*e.g.*, band gap) can be explained by certain properties of the compounds (such as element stoichiometry, number of unique elements, and/or bond character), there are additional behaviors that govern the material property. These additional interactions are also adequately modeled by CrabNet, since it can predict a wide range of material properties with low errors. Examining the modeling process of these behaviors within CrabNet may lead to

an improved understanding of the complex phenomena underlying material properties.

Further research to answer these and subsequent questions may allow us to gain additional insights about the behaviors and properties of elements and materials, improve our understanding of models such as CrabNet, increase our confidence in the use of data-driven methods, and ultimately, accelerate the adoption of deep learning and machine learning in materials science.

Methods

Adaptation of CrabNet Model

The CrabNet model and material property datasets as originally reported were used as the basis for this study [28]. Fully trained model weights for both CrabNet and HotCrab were obtained from [57]. In order to obtain the EDMs containing the elements and compounds data used in this study, custom function hooks were implemented in PyTorch. These hooks were attached to the CrabNet model architecture to allow access to the model-internal data during training and inference.

The source code as well as the data that were used and generated in this study can be found on the updated CrabNet GitHub repository [58]. In addition, we provide detailed instructions for the use and reproduction of our reported results. Please note that due to the prohibitively large size of the stored attention matrices used in the attention videos, it is not possible to provide these for download. However, instructions and scripts are provided for generating these matrices and videos.

All experiments, unless otherwise noted, were performed on a workstation equipped with an Intel i7-8700K CPU, 32 GB of DDR4 RAM, and one Nvidia RTX 2080 GPU.

Element Embeddings

Element embeddings for pure elements were generated on a per-property basis. To do this, an EDM consisting of all of the elements from hydrogen to oganesson was generated (with each row representing one element). Then, for each material property, the corresponding CrabNet or HotCrab model was loaded and the model hooks attached. The EDM was then passed through the network and the modified EDM at the output of the element embedding layer was obtained and detached from the model graph. This resulting EDM contains the property-specific element embeddings of all of

the elements. Thus, each element was represented by a vector with the shape $(1, d_{\text{model}})$, where d_{model} is the size of the embedding. Element embeddings for Oliynyk, Magpie, and mat2vec were obtained from the original publications [18].

Compound Embeddings

Compound embeddings were obtained in a similar fashion to element embeddings. Instead of generating an EDM from pure elements, the EDMs were generated from the actual chemical formulae from the datasets and collated in batches using the model data loader. Model hooks were then attached to the CrabNet and HotCrab models and enabled during model inference. The transformed EDMs after each of the three self-attention layers of the CrabNet models were then collected.

The obtained compound EDMs have the shape of $(n_{\text{compounds}}, n_{\text{elements}}, d_{\text{model}})$, where $n_{\text{compounds}}$ is the total number of compounds in the dataset, n_{elements} is the maximum number of elements per compound, and d_{model} is the size of the embedding. Thus, each compound in the EDM is represented by one tensor slice with the dimensions $(1, n_{\text{elements}}, d_{\text{model}})$. Due to the fact that different compounds within the same dataset may contain a different number of elements, the extra rows of the EDMs were zero-filled to indicate no elements present. In order to ensure that the compound embeddings are comparable with each other using UMAP, the three-dimensional compound EDMs were collapsed to two dimensions $(n_{\text{compounds}}, 1, d_{\text{model}})$ by calculating summary statistics (such as sum, range, variance) of the EDM columns across the elements dimension.

Dimensionality Reduction

CrabNet uses vectors with a d_{model} dimension of 512 to represent chemical elements and compounds in the input data. It would be infeasible to try to visualize all 512 dimensions. Therefore, dimensionality reduction was applied to the vector representations to transform the vectors into two-dimensional space for visualization.

Three common methods for dimensionality reduction were tested: principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), and uniform manifold approximation and projection (UMAP) [73, 85, 86]. Compared to t-SNE and PCA, UMAP revealed more visually distinct clusters for the data presented in this work. Therefore, UMAP was chosen as the dimensionality reduction method. The random seed was fixed so that each initialization of the UMAP method produces the same results. For element embeddings, the rows of the EDMs with dimensions $(1, d_{\text{model}})$ are transformed using UMAP. For the compound embeddings, the matrices

corresponding to each compound were first collapsed as described above, and the resulting representations with dimensions $(1, d_{\text{model}})$ for each compound were transformed using UMAP.

Oxidation State Estimation

Oxidation states for elements in the compounds were estimated using the Pymatgen package (version 2022.0.8) using the chemical formulae of the compounds. The built-in functions for assigning oxidation states were used, which are based on charge-balancing heuristics and use the most probable oxidation states as determined based on the compounds in the Inorganic Crystal Structure Database [74].

Attention Video Generation

Custom function hooks were programmed and attached to a newly-initialized CrabNet model. During training of CrabNet, the attention matrices of every CrabNet encoder layer was extracted from the model and saved into a compressed Zarr array on disk. The model predictions for the properties were also generated and saved. This procedure is performed after every mini-step during the training process (corresponding to each mini-batch of data). The plots were then generated for each mini-step and merged together using the software FFMPEG to create the attention videos. Due to the large amount of storage and computing power required to store and process the attention matrices, these tasks were performed on a high-performance computing cluster.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s40192-021-00247-y>.

Acknowledgements The authors thank the Berlin International Graduate School in Model and Simulation based Research as well as the German Academic Exchange Service RISE program for their financial support. Special thanks is given to Dr. Steven K. Kauwe, Pay Gießelmann and Joris Weigert for the insightful discussions. Computing resources were graciously provided by the HPC-Cluster at the Institut für Mathematik, Technische Universität Berlin, by the HPC Resource in the Core Facility for Advanced Research Computing at Case Western Reserve University as well as by the Google TPU Research Cloud (TRC) program. 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, PyTorch, Zarr, and FFMPEG.

Funding Open Access funding enabled and organized by Projekt DEAL.

Compliance with Ethical Standards

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

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Ramprasad R, Batra R, Pilia G, Mannodi-Kanakthodi A, Kim C (2017) Machine learning in materials informatics: recent applications and prospects. *npj Comput Mater* 3(1):60
2. Schmidt J, Marques MRG, Botti S, Marques MAL (2019) Recent advances and applications of machine learning in solid-state materials science. *npj Comput Mater* 5(1):83
3. Gomes CP, Selman B, Gregoire JM (2019) Artificial intelligence for materials discovery. *MRS Bull* 44(7):538–544
4. Isayev O, Tropsha A, Curtarolo S (eds) (2019) *Materials informatics: methods, tools, and applications*. Wiley, USA
5. DeCost BL, Hattrick-Simpers JR, Trautt Z, Kusne AG, Campo E, Green ML (2020) Scientific AI in materials science: a path to a sustainable and scalable paradigm. *Mach Learn: Sci Technol* 1(3):033001
6. Stein HS, Gregoire JM (2019) Progress and prospects for accelerating materials science with automated and autonomous workflows. *Chem Sci* 10(42):9640–9649
7. Morgan D, Jacobs R (2020) Opportunities and challenges for machine learning in materials science. *Annu Rev Mater Res* 50(1):71–103
8. Zitnick CL, Chanussot L, Das A, Goyal S, Heras-Domingo J, Ho C, Hu W, Lavril T, Palizhati A, Riviere M, Shuaibi M, Sriram A, Tran K, Wood B, Yoon J, Parikh D, Ulissi Z (2020) An introduction to electrocatalyst design using machine learning for renewable energy storage. <http://arxiv.org/abs/2010.09435> v1
9. Sparks TD, Kauwe SK, Parry ME, Tehrani AM, Brgoch J (2020) Machine learning for structural materials. *Ann Rev Mater Res* 50:27
10. Pilia G (2021) Machine learning in materials science: from explainable predictions to autonomous design. *Comput Mater Sci* 193:110360
11. Oliynyk AO, Antono E, Sparks TD, Ghadbeigi L, Gaultois MW, Meredig B, Mar A (2016) High-throughput machine-learning-driven synthesis of full-heusler compounds. *Chem Mater* 28(20):7324–7331
12. Ward L, Agrawal A, Choudhary A, Wolverton C (2016) A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Comput Mater* 2(1):16028
13. Pilia G, Mannodi-Kanakthodi A, Ueberuaga BP, Ramprasad R, Gubernatis JE, Lookman T (2016) Machine learning band-gaps of double perovskites. *Sci Rep* 6:19375
14. Dunn A, Wang Q, Ganose A, Dopp D, Jain A (2020) Benchmarking materials property prediction methods: the Matbench test set and Automminer reference algorithm. *npj Comput Mater* 6(1):138

15. Kauwe SK, Graser J, Murdock RJ, Sparks TD (2020) Can machine learning find extraordinary materials? *Comput Mater Sci* 174:109498
16. Graser J, Kauwe SK, Sparks TD (2018) Machine learning and energy minimization approaches for crystal structure predictions: a review and new horizons. *Chem Mater* 30(11):3601–3612
17. Tehrani AM, Oliynyk AO, Parry M, Rizvi Z, Couper S, Lin F, Miyagi L, Sparks TD, Brgoch J (2018) Machine learning directed search for ultraincompressible, superhard materials. *J Am Chem Soc* 140(31):9844–9853
18. Murdock RJ, Kauwe SK, Wang AY-T, Sparks TD (2020) Is domain knowledge necessary for machine learning materials properties? *Integr Mater Manuf Innov* 9(3):221–227
19. Choudhary K, DeCost B, Tavazza F (2018) Machine learning with force-field-inspired descriptors for materials: fast screening and mapping energy landscape. *Phys Rev Mater* 2(8):083801
20. Tshitoyan V, Dagdelen J, Weston L, Dunn A, Rong Z, Kononova O, Persson KA, Ceder G, Jain A (2019) Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* 571(7763):95–98
21. Kauwe SK, Graser J, Vazquez A, Sparks TD (2018) Machine learning prediction of heat capacity for solid inorganics. *Integr Mater Manuf Innov* 7(2):43–51
22. Kauwe SK, Welker T, Sparks TD (2020) Extracting knowledge from DFT: experimental band gap predictions through ensemble learning. *Integr Mater Manuf Innov* 9(3):213–220
23. Jha D, Ward L, Paul A, Liao W-K, Choudhary A, Wolverton C, Agrawal A (2018) ElemNet: deep learning the chemistry of materials from only elemental composition. *Sci Rep* 8(1):17593
24. Xie T, Grossman JC (2018) Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys Rev Lett* 120(14):145301
25. Chen C, Ye W, Zuo Y, Zheng C, Ong SP (2019) Graph networks as a universal machine learning framework for molecules and crystals. *Chem Mater* 31(9):3564–3572
26. Klicpera J, Giri S, Margraf JT, Günnemann S (2020) Fast and uncertainty-aware directional message passing for non-equilibrium molecules. <https://arxiv.org/abs/2011.14115>
27. DeCost B, Choudhary K (2021) Atomistic line graph neural network for improved materials property predictions. *npj Comput Mater* 7(1):185
28. Wang AY-T, Kauwe SK, Murdock RJ, Sparks TD (2021) Compositionally restricted attention-based network for materials property predictions. *npj Comput Mater* 7(1):77
29. Goodall REA, Lee AA (2020) Predicting materials properties without crystal structure: deep representation learning from stoichiometry. *Nat Commun* 11(1):6280
30. Kong S, Guevarra D, Gomes CP, Gregoire JM (2021) Materials representation and transfer learning for multi-property prediction. *Appl Phys Rev* 8(2):021409
31. Clement CL, Kauwe SK, Sparks TD (2020) Benchmark AFLOW data sets for machine learning. *Integr Mater Manuf Innov* 9(2):153–156
32. Wang AY-T, Murdock RJ, Kauwe SK, Oliynyk AO, Gurlo A, Brgoch J, Persson KA, Sparks TD (2020) Machine learning for materials scientists: an introductory guide toward best practices. *Chem Mater* 32(12):4954–4965
33. Henderson AN, Kauwe SK, Sparks TD (2021) Benchmark datasets incorporating diverse tasks, sample sizes, material systems, and data heterogeneity for materials informatics. *Data Brief* 37:107262
34. Meredig B (2017) Industrial materials informatics: analyzing large-scale data to solve applied problems in R&D, manufacturing, and supply chain. *Curr Opin Solid State Mater Sci* 21(3):159–166
35. Lipton ZC (2018) The mythos of model interpretability. *Queue* 16(3):31–57
36. Himanen L, Geurts A, Foster AS, Rinke P (2019) Data-driven materials science: status, challenges, and perspectives. *Adv Sci* 6(21):1900808
37. Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat Mach Intell* 1(5):206–215
38. Kolyshkina I, Simoff S (2019) Interpretability of machine learning solutions in industrial decision engineering. In: Data mining (T. D. Le, K.-L. Ong, Y. Zhao, W. H. Jin, S. Wong, L. Liu, and G. Williams, eds.), vol. 1127 of communications in computer and information science, pp. 156–170, Singapore: Springer Singapore
39. Linardatos P, Papastefanopoulos V, Kotsiantis S (2021) Explainable AI: a review of machine learning interpretability methods. *Entropy (Basel, Switzerland)* 23(1):18
40. Gilpin LH, Bau D, Yuan BZ, Bajwa A, Specter M, Kagal L (2018) Explaining explanations: an overview of interpretability of machine learning. In: 2018 IEEE 5th international conference on data science and advanced analytics (DSAA), pp 80–89, IEEE
41. Smilkov D, Thorat N, Nicholson C, Reif E, Viégas FB, Wattenberg M (2016) Embedding projector: interactive visualization and interpretation of embeddings. <http://arxiv.org/abs/1611.05469v1>
42. Liu S, Bremer P-T, Thiagarajan JJ, Srikumar V, Wang B, Livnat Y, Pascucci V (2018) Visual exploration of semantic relationships in neural word embeddings. *IEEE Trans Visual Comput Graphics* 24(1):553–562
43. van Aken B, Winter B, Löser A, Gers FA (2020) VisBERT: Hidden-state visualizations for transformers. In: Companion proceedings of the web conference 2020 (A. E. F. Seghrouchni, G. Sukthankar, T.-Y. Liu, and M. van Steen, eds.), (New York, NY, USA), pp 207–211, ACM
44. Vinyals O, Babuschkin I, Czarnecki WM, Mathieu M, Dudzik A, Chung J, Choi DH, Powell R, Ewalds T, Georgiev P, Oh J, Horgan D, Kroiss M, Danihelka I, Huang A, Sifre L, Cai T, Agapiou JP, Jaderberg M, Vezhnevets AS, Leblond R, Pohlen T, Dalibard V, Budden D, Sulsky Y, Molloy J, Paine TL, Gulcehre C, Wang Z, Pfaff T, Wu Y, Ring R, Yegoroyan D, Wünsch D, McKinney K, Smith O, Schaul T, Lillicrap T, Kavukcuoglu K, Hassabis D, Apps C, Silver D (2019) Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575(7782):350–354
45. Puiutta E, Veith EMS (2020) Explainable Reinforcement Learning: A Survey. In: Machine learning and knowledge extraction (A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl, eds.), vol. 12279 of Lecture Notes in Computer Science, pp 77–95, Cham: Springer International Publishing
46. Heuillet A, Couthouis F, Díaz-Rodríguez N (2021) Explainability in deep reinforcement learning. *Knowl-Based Syst* 214:106685
47. Lapuschkin S (2018) Opening the machine learning black box with Layer-wise Relevance Propagation. PhD thesis, Technische Universität Berlin, Berlin, Germany
48. Chefer H, Gur S, Wolf L (2020) Transformer interpretability beyond attention visualization. <http://arxiv.org/abs/2012.09838v2>
49. Chen J, Lu Y, Yu Q, Luo X, Adeli E, Wang Y, Lu L, Yuille AL, Zhou Y (2021) TransUNet: transformers make strong encoders for medical image segmentation. <http://arxiv.org/abs/2102.04306v1>
50. Khan S, Naseer M, Hayat M, Zamir SW, Khan FS, Shah M (2021) Transformers in vision: a survey. <http://arxiv.org/pdf/2101.01169v3>
51. Kailkhura B, Gallagher B, Kim S, Hiszpanski A, Han TY-J (2019) Reliable and explainable machine-learning methods for accelerated material discovery. *npj Comput Mater* 5(1):221
52. Roscher R, Bohn B, Duarte MF, Garcke J (2020) Explainable machine learning for scientific insights and discoveries. *IEEE Access* 8:42200–42216

53. Ribeiro MT, Singh S, Guestrin C (2016) Why should I trust you?: explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining – KDD '16 (B. Krishnapuram, M. Shah, A. Smola, C. Aggarwal, D. Shen, and R. Rastogi, eds.), (New York, NY, USA), pp. 1135–1144, ACM Press
54. Lundberg S, Lee S-I (2017) A unified approach to interpreting model predictions. <http://arxiv.org/abs/1705.07874v2>
55. Shapley LS (1953) A Value for n-person games. In: contributions to the theory of games (AM-28), Volume II (H. W. Kuhn and A. W. Tucker, eds.), Annals of Mathematics Studies, pp. 307–318, Princeton, NJ: Princeton University Press
56. Hastie T, Tibshirani R, Friedman JH (2009) The elements of statistical learning: Data mining, inference, and prediction. Springer Series in Statistics. Springer, New York
57. Wang AY-T, Kauwe SK, Murdock RJ, Sparks TD (2021) Trained network weights for the paper, Compositionally restricted attention-based network for materials property predictions (CrabNet). <https://doi.org/10.5281/zenodo.4633866>
58. Wang AY-T, Kauwe SK (2020) Online GitHub repository for the paper, compositionally-restricted attention-based network for materials property prediction. <https://github.com/anthony-wang/CrabNet>
59. Saal JE, Kirklin S, Aykol M, Meredig B, Wolverton C (2013) Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD). JOM 65(11):1501–1509
60. Kirklin S, Saal JE, Meredig B, Thompson A, Doak JW, Aykol M, Rühl S, Wolverton C (2015) The open quantum materials database (OQMD): assessing the accuracy of DFT formation energies. npj Computational Materials 1(1):15010
61. Hegde VI, Borg CKH, Rosario Zd, Kim Y, Hutchinson M, Antono E, Ling J, Saxe P, Saal JE, Meredig B (2020) Reproducibility in high-throughput density functional theory: a comparison of AFLOW, Materials Project, and OQMD. <http://arxiv.org/pdf/2007.01988v1>
62. Bonachela JA, Hinrichsen H, Muñoz MA (2008) Entropy estimates of small data sets. J Phys A: Math Theor 41(20):202001
63. Hong C, Ghosh R, Srinivasan S (2016) Dealing with class imbalance using thresholding. <http://arxiv.org/pdf/1607.02705v1>
64. Tahir MAUH, Asghar S, Manzoor A, Noor MA (2019) A classification model for class imbalance dataset using genetic programming. IEEE Access 7:71013–71037
65. Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27(3):379–423
66. Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27(4):623–656
67. Li Y, Vasconcelos N (2019) REPAIR: removing representation bias by dataset resampling. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR) (CVPR Editors, ed.), pp 9564–9573, IEEE
68. Esposito C, Landrum GA, Schneider N, Stiefl N, Riniker S (2021) GHOST: adjusting the decision threshold to handle imbalanced data in machine learning. J Chem Inf Model 61(6):2623–2640
69. Li K, Wu Z, Peng K-C, Ernst J, Fu Y (2018) Tell me where to look: guided attention inference network,” in 2018 IEEE/CVF conference on computer vision and pattern recognition, pp 9215–9223, IEEE
70. Rodriguez AC, D’Aronco S, Schindler K, Wegner JD (2020) Privileged pooling: better sample efficiency through supervised attention. <http://arxiv.org/abs/2003.09168v3>
71. Kim E, Huang K, Tomala A, Matthews S, Strubell E, Saunders A, McCallum A, Olivetti E (2017) Machine-learned and codified synthesis parameters of oxide materials. Sci Data 4:170127
72. Weston L, Tshitoyan V, Dagdelen J, Kononova O, Trewartha A, Persson KA, Ceder G, Jain A (2019) Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. J Chem Inform Model 59:3692
73. McInnes L, Healy J, Saul N, Großberger L (2018) UMAP: uniform manifold approximation and projection. J Open Sour Softw 3(29):861
74. Ong SP, Richards WD, Jain A, Hautier G, Kocher M, Cholia S, Gunter D, Chevrier VL, Persson KA, Ceder G (2013) Python materials genomics (pymatgen): a robust, open-source python library for materials analysis. Comput Mater Sci 68:314–319
75. Hargreaves CJ, Dyer MS, Gaultois MW, Kurlin VA, Rosseinsky MJ (2020) The earth mover’s distance as a metric for the space of inorganic compositions. Chem Mater 32(24):10610–10620
76. Glaudell AM, Cochran JE, Patel SN, Chabinc ML (2015) Impact of the doping method on conductivity and thermopower in semi-conducting polythiophenes. Adv Energy Mater 5(4):1401072
77. Zhang SB (2002) The microscopic origin of the doping limits in semiconductors and wide-gap materials and recent developments in overcoming these limits: a review. J Phys: Condens Matter 14(34):R881–R903
78. Sheng L, Wang L, Xi T, Zheng Y, Ye H (2011) Microstructure, precipitates and compressive properties of various holmium doped NiAl/Cr(Mo, Hf) eutectic alloys. Mater Design 32(10):4810–4817
79. Tehrani AM, Oliynyk AO, Rizvi Z, Lotfi S, Parry M, Sparks TD, Brgoch J (2019) Atomic substitution to balance hardness, ductility, and sustainability in molybdenum tungsten borocarbide. Chem Mater 31(18):7696–7703
80. Mihailovich and Parpia (1992) Low temperature mechanical properties of boron-doped silicon. Phys Rev Lett 68(20):3052–3055
81. Qu Z, Sparks TD, Pan W, Clarke DR (2011) Thermal conductivity of the gadolinium calcium silicate apatites: effect of different point defect types. Acta Mater 59(10):3841–3850
82. Sparks TD, Fuierer PA, Clarke DR (2010) Anisotropic thermal diffusivity and conductivity of La-doped strontium niobate Sr₂Nb₂O₇. J Am Ceram Soc 93(4):1136–1141
83. Grimvall G (1999) Thermophysical Properties of Materials. Amsterdam: North Holland, 1 ed
84. Gaumé R, Viana B, Vivien D, Roger J-P, Fournier D (2003) A simple model for the prediction of thermal conductivity in pure and doped insulating crystals. Appl Phys Lett 83(7):1355–1357
85. Pearson K (1901) On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Phil Magazine J Sci 2(11):559–572
86. van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2605