# Compositionally-Restricted Attention-Based Network for Materials Property Predictions

Anthony Yu-Tung Wang[1,*], Steven K. Kauwe[2,*], Ryan J. Murdock[2], and Taylor D. Sparks[2,3]

[1] *Fachgebiet Keramische Werkstoffe / Chair of Advanced Ceramic Materials, Technische Universität Berlin, 10623 Berlin, Germany*

[2] *Department of Materials Science & Engineering, University of Utah, Salt Lake City, UT, 84112, USA*

[3] Corresponding author, **Email:** sparks@eng.utah.edu

[*] Contributed equally to this work

**Abstract**

In this paper, we demonstrate an application of the Transformer self-attention mechanism in the context of materials science. Our network, the Compositionally-Restricted Attention-Based network (`CrabNet`), explores the area of structure-agnostic materials property predictions when only a chemical formula is provided. Our results show that `CrabNet`'s performance matches or exceeds current best practice methods on nearly all of 28 total benchmark datasets. We also demonstrate how `CrabNet`'s architecture lends itself towards model interpretability by showing different visualization approaches that are made possible by its design. We feel confident that `CrabNet` and its attention-based framework will be of keen interest to future materials informatics researchers.

# Keywords

machine learning, materials informatics, self-attention, transformers, materials discovery, material screening, high-throughput screening, regression, interpretability

# Introduction

Materials scientists constantly strive to achieve better understanding, and therefore better predictions, of materials properties. This began with the collection of empirical evidence through repeated experimentation, resulting in mathematical generalizations, theories, and laws. More recently, computational methods have arisen to solve a large variety of problems that were intractable to analytical approaches alone [1, 2].

As experimental and computational methods have become more efficient, high-quality data has opened up a new avenue to materials understanding. Materials informatics (MI) is the resulting field of research which utilizes statistical and machine learning (ML) approaches in combination with high-throughput computation to analyze the wealth of existing materials information and gain unique insights [2–4]. As this wealth has increased, practitioners of MI have increasingly turned to deep learning techniques to model and represent inorganic chemistry, resulting in approaches such as `ElemNet`, `IRNet`, `CGCNN`, `SchNet` and `Roost` [5–9]. In specific cases including

CGCNN and SchNet, the compounds are represented using their chemical and structural information [7, 8, 10–15].

Modeling approaches based on crystal structure are an excellent tool for MI. Unfortunately, there are many material property datasets that lack suitable structural information. An example of this is the experimental band gap data gathered by Zhou *et al.* [16]. Conversely, many databases such as the Inorganic Crystal Structure Database (ICSD) and Pearson's Crystal Data (PCD) contain an abundance of structural information, but lack the associated material properties of the recorded structures. In both cases, the applicability of structure-based learning approaches are limited. This limitation is particularly evident in the discovery of novel materials, since it is not possible to know the structural information of (currently undiscovered) chemical compounds *a priori*. Therefore, the development of structure-agnostic techniques is well-suited to the discovery of novel materials.

A typical approach to structure-agnostic learning is done by representing chemistry as a composition-based feature vector (CBFV) [17]. This allows for data-driven learning in the absence of structural information. The CBFV is a common way to transform chemical compositions into usable features for ML and is generated from the descriptive statistics of a compound's constituent element properties. Researchers have effectively used CBFV-based ML techniques to generate materials property predictions [17–25].

One potential issue with the CBFV approach lies in the way the element vectors are combined to form the vector describing the chemical compound. Typically, the individual element vectors of the compound are scaled by the element's prevalence (fractional abundance) in the composition, before being used to form the CBFV. This step assumes that the stoichiometric prevalence of constituent elements in a compound dictate their chemical signal, or contribution, to the material's property. However, this is not true in all cases; an extreme example of this is element doping. Dopants can be present in very small amounts in a compound, but can have a significant impact on its electrical [23, 26, 27], mechanical [20, 28–30], and thermal properties [31–34]. In the case of a typical CBFV approach which uses the weighted average of element properties as a feature, the signal from dopant elements would not significantly change the vector representation of a compound. As a result, the trained ML model would fail to capture a portion of the relevant chemical information available in the full composition.

It is apparent that there is no generally-accepted best way to model materials property behaviors. Different ML approaches lend themselves towards different modeling tasks. CGCNN requires access to structural information, ElemNet operates within the realm of large data, and classical models excel when domain knowledge can be exploited to overcome data scarcity [35]. To address the diversity of learning challenges, in Dunn *et al.*, the Automatminer framework uses computationally-expensive searches to optimize classical modeling techniques. They demonstrate effective learning on some data, but show shortcomings when deep-learning is appropriate [36].

In a similar spirit, we seek to overcome general challenges in the area of structure-agnostic learning using an approach we refer to as the Compositionally-Restricted Attention-Based network (`CrabNet`). `CrabNet` introduces the self-attention mechanism to the task of materials property predictions, and dynamically learns and updates individual element representations based on their chemical environment. To enable this, we introduce a featurization scheme that represents and preserves individual element identity while sharing information between elements. Self-attention is a procedure by which a neural network learns representations for each item in a system based on the other items that are present. In this context, we treat the chemical composition as the system and the elements as the items within that system. This representation enables `CrabNet` to learn inter-element interactions within a compound and use these interactions to generate property predictions. To perform self-attention, we use the Transformer architecture, which emerged from natural language processing (NLP) and is based on stacked self-attention layers [37–42]. A typical use of the Transformer architecture in NLP is to encode the meaning of a word given the surrounding words, sentences, and paragraphs. Beyond NLP, other example uses of the Transformer architecture are found in music generation [43], image generation [44], image and video restoration [45–49], game playing agents [50, 51], and drug discovery [52, 53]. In this work, we explore how our attention-based architecture, `CrabNet`, performs in predicting materials properties relative to the common modeling techniques `Roost`, `ElemNet`, and random forest (`RF`) for regression-type problems.

# Results

The results of this study are described in three subsections. First, we describe the collection of materials property data used for benchmarking `CrabNet`. Second, we highlight the performance of `CrabNet` when compared to other current learning approaches which rely solely on composition. Third, we briefly outline how the self-attention mechanism in `CrabNet` enables visualizations and inspectability unique to attention-based modeling.

## Data and Materials Properties Procurement

For this work, we obtained both computational and experimental materials data for benchmarking. Our benchmark data includes materials properties from the Matbench dataset as provided by Dunn *et al.* [36]. In addition, materials properties data from a number of works [6, 54–57] are collected, which are referred to as the Extended dataset. We included 28 benchmark datasets in total: 10 from the Matbench and 18 from the Extended datasets ranging from 312 to 341,788 instances of data.

The Matbench datasets were split using five-fold cross-validation following instructions provided in the original publication [36]. Materials properties in the Extended dataset were split into train, validation, and test datasets using a fixed random seed. For both datasets, several steps were taken to process the original datasets to be compatible with structure-agnostic learning using `CrabNet`. Care was taken to ensure that (1) no duplicate compositions were present in each of the train, validation, and test datasets, and that (2) if a composition exists in the train or validation dataset, all compounds with the same composition are removed from the validation and test datasets. To remain comparable with the Automatminer publication [36], we applied the data processing steps as mentioned above after splitting the data. Please note that since some datasets have more duplicate compositions than others, these processing steps may affect the train/val/test ratios. For duplicate compositions in the OQMD and MP datasets, the target value associated with the lowest formation enthalpy was selected. For other datasets, the mean of the target values was used. Please see the Supplementary Methods for more details.

The full processed benchmark dataset, comprising the Matbench and Extended datasets, were then used with `Roost`, `CrabNet`, `ElemNet`, and `RF` models. The training and validation data were used for training and hyperparameter tuning.

The test data were held-out to provide a fair evaluation of performance metrics across all models. Model performance was only evaluated after all training and hyperparameter tuning was completed. A summary of the datasets is shown in Table 1. All datasets are provided as pre-split csv files to facilitate future comparisons to the metrics reported in this paper. Additional data processing and cleaning details can also be seen in the code on the dataset repository `mse_datasets` [58]. To maintain consistent and simple benchmark comparisons, we selected data suitable for regression tasks and ignored structural information when present.

## Benchmark Comparisons

With the benchmark data described above, we generated materials predictions using the publicly-available code repositories for `Roost` [9], `CrabNet` [59], and `ElemNet` [5]. The performance of these benchmarked models is compared using the mean absolute error (MAE) between true values ($y$) and predicted values ($\hat{y}$) as defined by Equation 1:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} \left| y_i - \hat{y}_i \right| . \qquad (1)$$

This allows for consistent comparison to past works [5–7, 9].

Figure 1 shows the performance metrics from training and testing the models on all the benchmark materials properties outlined above. Here we note that the models for `Roost`, `CrabNet`, and `ElemNet` were all trained using the default model parameters provided with their respective repositories. In contrast to `Roost` and `ElemNet`, the default parameters for `CrabNet` were optimized using validation data from some of the same datasets on which we benchmarked. Although it is possible this offers a small advantage to `CrabNet`'s performance, we do not expect this to be significant due to `CrabNet`'s consistently strong performance on all benchmark tasks.

We tested two versions of `CrabNet`. The default `CrabNet` uses a `mat2vec` embedding when representing elements, similar to `Roost`. The second version of

CrabNet (HotCrab) uses one-hot encodings (in the form of atomic numbers) and fractional amounts to represent each element in a composition. This is similar to ElemNet, as both models start without any chemical information. The random forest (RF) model utilizes a Magpie-featurized CBFV to represent chemistry. This is included as a performance baseline to match similar works [5, 9, 36].

Overall, we see similar performance between Roost and the two versions of CrabNet tested. Given the different architectures and modelling philosophies of Roost and CrabNet, it is promising that both approaches converge towards the same performance metrics. We also see that Roost and both CrabNet versions achieve consistent and significant improvements to MAE compared to ElemNet and RF approaches. Interestingly, Figure 1 shows that the use of mat2vec instead of one-hot with CrabNet improves prediction performance on all materials properties except for AFLOW thermal conductivity, MP elastic anisotropy, and those present in the largest datasets (OQMD).

The Matbench data provided by Dunn *et al.* [36] was benchmarked using the Automatminer tool. These metrics are not included in Figure 1, since all but two (expt_gap, and steels_yield) of Automatminer's models use structural information. Consequently, we focus on these two materials properties when comparing CrabNet's results to those from Automatminer. For these two metrics, CrabNet's structure-agnostic approach outperforms the reported MAE values from Automatminer on the same tasks (expt_gap: 0.416 eV vs. 0.338 eV for CrabNet; steels_yield: 95.2 GPa vs. 91.7 GPa for CrabNet).

The performance of CrabNet on the steels_yield task is particularly interesting. The steels_yield dataset contains compounds with small dopant amounts in large chemical systems (up to 13 elements per composition) and only 312 total data. CrabNet's ability to learn on this data-poor property and outperform all other tested models including the baseline RF model (which is traditionally better in the data-poor regime) is encouraging. We expected the steels_yield task to be difficult for all deep learning approaches. Nevertheless, repeated training and validation of CrabNet consistently produced error metrics better than the best result obtained by Automatminer (95.2 GPa).

## Visualizing Self-Attention

CrabNet's modeling and visualization capabilities are enabled by its attention-based learning framework. In statistical machine learning and many deep learning approaches akin to ElemNet, the chemical composition of a compound is represented as a single CBFV. In contrast, Roost and CrabNet represent a composition as a set of element vectors. Distinct to CrabNet, however, is the Transformer-based self-attention mechanism that learns to update these element vectors using learned attention matrices. In Figure 2, we show example attention matrices for each attention head of a CrabNet model trained on the property mp_bulk_modulus, using $Al_2O_3$ as the example composition. These matrices contain the information regarding how each element (rows) is influenced by all other elements in the system as well as itself (columns). The values in these attention matrices are used in the Transformer encoder to update the element vectors. A value of zero means that the element in the column is completely ignored when updating the element in that row. A value of one means that the entire vector update is based solely on that column's element. Our implementation of CrabNet has three layers, each with four attention heads, with each head using the same data to generate its own independent attention matrix (see Methods for more details).

Shifting our focus to another CrabNet model trained on aflow___Egap data, we show that in addition to visualization of the individual attention heads, we can also generate a global view of attention from the perspective of individual elements. In Figure 3, we use four periodic tables to visualize, for each attention head, the average attention that silicon dedicates to other elements when they are in the same composition. The darker colored elements can be understood as highly influential when updating silicon's vector representation.

Interestingly, each attention head has its own behavior, with some focusing on familiar groups and columns in the periodic table. This behavior lends credibility to CrabNet since there is no inherent reason that data-driven learning should converge to chemical rules that are familiar to materials scientists. Furthermore, the identification of unfamiliar element groupings enabled by the attention-based visualizations may allow us to formulate further research questions to study these inter-elemental interactions.

The preservation of elemental identity within a compound—as a result of the self-attention mechanism—also enables `CrabNet` to generate property predictions in a way that is different to other approaches shown in the literature. Typically, element information of a given compound is collapsed into a single vector first and then used to generate the property prediction. In contrast, `CrabNet` uses each element's vector representation to directly predict the element's contribution to the property prediction. Figure 4a shows the average contributions from each element for a `CrabNet` model trained on AFLOW_bulk_modulus data. The darker colored elements contribute more towards a compound's bulk modulus value. Alternatively, elements can be visualized individually using their specific per-element contributions. In Figure 4b we show distribution plots for lithium and tungsten's contributions to bulk modulus. From these plots, we can see that `CrabNet` expects lithium to contribute little to overall bulk modulus, whereas it expects tungsten to contribute largely. See Supplementary Figure 3 for additional examples of these element contribution plots. The visualizations from Figure 4 match closely—and reinforce—expectations regarding which elements most influence bulk modulus behavior in a compound. Exploration of data in this manner hints at first steps towards model interpretability of `CrabNet`. We expect these types of property visualizations to be useful for exploring and verifying model and element behavior in detail.

Finally, with per-element contributions in mind, we can demonstrate changes to `CrabNet`'s expected material property behavior as a function of chemical composition. To do this, we consider a normalized chemical system consisting of atoms A and B, in the form of $A_xB_{1-x}$. We then generate property predictions for all $x \in \{0.0, 0.02, ..., 1.0\}$. In Figure 5, we visualize `CrabNet`'s behavior when predicting band gap of the $Si_xO_{1-x}$ system using a model trained on the aflow__Egap data.

We first observe that the expected elemental contributions for both oxygen and silicon to band gap are similar throughout the varied stoichiometry range, with the exception of the peak in oxygen contribution at around $x = 0.7$. We also observe that the model indicates a transition of the $Si_xO_{1-x}$ system between conducting and semi-conducting between $x = 0.5$ and $x = 0.7$. We note that the only available training data sample from the $Si_xO_{1-x}$ system in the dataset was from the composition $SiO_2$. Therefore, we can see that the band gap trend predicted here by `CrabNet` is based on the learned chemical representations and inter-elemental interactions from other elements and systems. The visualization of `CrabNet` model predictions within a given chemical space is an alternative way to explore model learning and prediction behavior, and may lead to an improved understanding of inter-elemental interactions within a chemical system.

Furthermore, we note that the ability of CrabNet in predicting material property trends for specific chemical systems without requiring a large amount of training data for that system is of great benefit. For future studies, this ability may be investigated for its application in predicting the behavior of new chemical systems while only requiring a sparse sampling or learning of their chemical information. Furthermore, we believe that transfer learning of trained CrabNet models to other material properties is possible, due to the ability of the self-attention mechanism to accurately capture inter-elemental interactions. We are confident that these ideas of probing and visualizing of CrabNet's modelling process and model predictions will open up further interesting research directions and ultimately lead to more insights in the pursuit of inspectable models.

## Discussion

Unique challenges exists when applying machine learning to materials science. In this paper, we address the limitations of machine learning on chemical composition by introducing CrabNet. The CrabNet architecture uses the self-attention mechanism and the EDM representation scheme to perform context-aware learning on materials properties. Using 28 benchmark datasets, we demonstrate CrabNet's performance compared to Roost, ElemNet, and RF baselines. CrabNet exhibits consistent predictive accuracy across the full range of materials properties tested. Furthermore, we show that the self-attention-based learning technique also provides alternative methods for visualizing model behavior. We demonstrate the use of attention and per-element contribution prediction capabilities for visualizing common trends in our trained models that match chemical expectations.

Given this application of self-attention in the context of materials science, we expect that there can be many informative and impactful follow-up works. Specifically, we believe these will largely fall into three thematic categories:

1. **CrabNet directly contributing to the community's focus towards improved property predictions.**

   CrabNet consistently generates good MAE scores. The performance achieved with the use of self-attention, combined with the innovative use of element and composition featurization techniques, will allow researchers to delve deeper into analyzing and predicting materials properties. As a result, we believe that CrabNet will be relevant in areas where other ML methods fall short (*e.g.*, dopants, small data, and materials extrapolation tasks). We also note that with minimal changes to `CrabNet`, it can also perform classification tasks; we expect `CrabNet` to similarly excel at this.

2. **Attention-based models allow for new ways of thinking about materials-specific problems.**

   In this work, we briefly examined the attention mechanism. Attention highlights important interactions and may be used to understand which element-interactions mediate materials properties. Model explainability has thus far been elusive to the traditional materials informatics paradigms. The inclusion of self-attention in this work has introduced additional methods of model inspectability that may be a step towards this goal.

3. **Augmentation of `CrabNet` using structural and domain-specific knowledge.**

   This work intentionally used a compositionally-restricted `EDM` representation with no structural information. Structure-agnostic learning is an important task in materials informatics and `CrabNet` demonstrates that accurate learning is achievable using the self-attention mechanism. However, the prediction of materials properties using structural information is also an important task. Integration of structural information could be achieved by describing elements in their structural and chemical environments. We expect that the self-attention mechanism of `CrabNet` will be able to utilize this additional information to make more accurate predictions. This application of attention-based learning to crystal systems is an exciting and promising direction. We also expect that materials prediction tasks involving processing steps or other non-compositional features could be used in this approach. Both of these changes could easily be implement as extensions to the `EDM`.

While further research is necessary to fully discern the utility of self-attention in materials problems, we believe that this paper highlights a major new direction in its application in materials informatics and suggests exciting directions for future research.

# Methods

## Self-attention and the `CrabNet` Architecture

Chemical compositions are input using the atomic numbers and fractional amounts of their constituent elements. The atomic numbers are used to retrieve element representations (either `mat2vec` or `one-hot`). The fractional amounts are used to obtain fractional embeddings (described below). An element embedding matrix is generated by applying a fully connected network to the element representations. A fractional embedding matrix is created from the fractional embeddings. These matrices are then added together (element-wise) to generate the element derived matrix (EDM, see Figure 6). Each row of the EDM ($i$-index) represents an element and the columns ($k$-index) contain the element embeddings. We batch each unique chemical composition onto a third dimension (the $i$-index). The resulting three-dimensional tensor contains the input data for the CrabNet architecture.

We use the mat2vec element embeddings [60] as the default source of chemical information for each element, even though there are other choices for element properties available, such as Jarvis [22], Magpie [61], Oliynyk [18] or a simple one-hot encoding. The mat2vec embedding has the advantage of being pre-scaled and normalized, and having no missing elements nor element features. Regardless of the choice of element representation, the representation must be reshaped to fit the the attention input dimensions of ($d_{\text{model}}$). This is done using a learned embedding network; the result is a matrix of size ($n_{\text{elements}}, d_{\text{model}}$). In addition to the default training of CrabNet using the mat2vec embedding, a one-hot embedding of the elements was used to train an additional CrabNet model (HotCrab) to better facilitate comparison with ElemNet.

The stoichiometric information for each element in the EDM is represented by two fractional embeddings. The fractional embeddings are inspired by the positional encoder as described in the seminal work by Vaswani *et al.* [37]. We use *sine* and *cosine* functions of various periods to project the fractional amounts into a high-dimensional space (dimension $d = d_{\text{model}}/2$) where smooth interpolation between fractional values is preserved. The first part of the fractional embedding represents the stoichiometry, using the normalized fractional amounts, on a linear scale with a fractional resolution of 0.01. The second part of the embedding maps stoichiometry using a log scale and spans from $1 \times 10^{-6}$ to $1 \times 10^{-1}$. This logarithmic transformation of the fractional embedding preserves small fractional amounts such as those present in doping. The two parts of the fractional embedding for all elements are concatenated across the embedding dimension to obtain a matrix of size $(n_{\text{elements}}, d_{\text{model}})$. See Supplementary Figure 1 and Supplementary Figure 2 for example visualizations of the `EDM` embedding.

Once the element and fractional embeddings are calculated and added together, we then batch the finished `EDM`s across the first dimension. This gives the final input data of shape $(n_{\text{compounds}}, n_{\text{elements}}, d_{\text{model}})$, where $n_{\text{compounds}}$ is the total number of compounds in a given batch, $n_{\text{elements}}$ is the number of rows in the `EDM` (inferred from the number of elements in the largest composition in a given dataset), and $d_{\text{model}}$ is the size of the embeddings. Here, we also note that the exact ordering of the element rows $(j)$ in a compound in the `EDM` does not influence `CrabNet` due to the permutation-invariant nature of the self-attention mechanism.

CrabNet contains two primary modules with the default hyperparameters as shown in Table 2. The first module is a Transformer encoder with 3 layers and 4 attention heads in each layer. The second module is a residual network that converts element vectors into element contributions.

To understand the Transformer encoder, we first describe the self-attention mechanism. During self attention (Figure 7a), the EDM is operated on by three fully-connected linear networks ($\mathrm{FC_Q}$, $\mathrm{FC_K}$, and $\mathrm{FC_V}$). These networks generate the query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ tensors. These tensors can be conceptualized as a learned high-dimensional space where the model stores chemical behavior from the training data.

The $\mathbf{K}$ and $\mathbf{Q}$ tensors contain information regarding the magnitude to which elements interact. The $\mathbf{V}$ tensor stores the information that is used to map from element to property contribution. The dot product of each $\mathbf{Q}$ and $\mathbf{K}^\mathsf{T}$ tensor pair (where $\mathbf{K}^\mathsf{T}$ denotes the transpose of $\mathbf{K}$) generates the relative element importances in the system (Figure 7b). The importances are scaled using a constant $\sqrt{d_\mathrm{k}}$ and then normalized using a softmax function. This results in the self-attention tensor, commonly referred to as the attention map. We denote this tensor as $\mathbf{A}$. The matrix multiplication of $\mathbf{A}$ with $\mathbf{V}$ updates the element-representations in the compound based on the importance of each element.

Each of the four attention heads independently performs self-attention with their own $\mathbf{Q}_h$, $\mathbf{K}_h$, $\mathbf{V}_h$, and $\mathbf{Z}_h$ tensors, where $h$ denotes the head index for $h = 1, \ldots, H$. As a result, the network generates four different element representations at each layer. The individual $\mathbf{Z}_h$ tensors are concatenated across the last dimension to make the $\mathbf{Z}$ tensor (as seen in Figure 8a). The $\mathbf{Z}$ tensor is then passed into a linear FC network which combines the element representations from each head. The output of this FC network is an updated `EDM′` (for each composition in the batch). This process of converting an `EDM` into an updated `EDM′` is referred to as a self-attention block. `CrabNet` repeats the process of updating the `EDM` via the self-attention block three times (hence, three layers) resulting in the final updated representations, denoted `EDM″`. This concludes the Transformer encoder module.

Once the Transformer encoder has updated the element representations, each `EDM″` passes through a fully-connected residual network hidden with layer dimensions of $\text{node}_{\text{res}}$. The residual network then transforms the `EDM`s into the shape $(n_{\text{elements}}, n_{\text{elements}}, 3)$. We define these final three vectors as the element-proto-contributions $p'$, element-uncertainties $u'$, and element-logits (see Figure 8a). The element scaling factor $s$ is obtained by taking the sigmoid $(\sigma)$ of the element-logits. The element-contributions are then obtained by multiplying the element-proto-contributions $p'$ by their respective scaling factor $s$. This results in element-contributions $y'$. Finally, the mean of the element-contributions is taken and output as the predicted property value for each compound (see Figure 8b). Similarly, the mean of the element-uncertainties is used in the aleatoric uncertainty prediction as described by Roost [9].

## Training CrabNet

After the featurization of compositions into EDMs, the dataset loading and batching is performed with the built-in Datasets and DataLoaders classes from PyTorch. All target values are scaled to zero-centered mean and unit variance for training and inference. The target scaling is then undone for performance evaluation. Batch size during training is dynamically calculated using the training set size for faster training, and limited to be within the range $2^7$ to $2^{12}$. For inference, the batch size was fixed at $2^7$.

Model weights are updated using the look-ahead [62] and Lamb optimizer [63] with a learning rate that is cycled between $1 \times 10^{-4}$ and $6 \times 10^{-3}$ every 4 epochs to achieve consistent model convergence. A robust mean absolute error (MAE) is used as the loss criterion for model performance [9]. The default parameters generalize well when predicting most of the benchmark materials properties. Although we expect that optimization of hyperparameters may improve CrabNet's results for individual materials properties, we believe it is more important that materials scientists be able to use CrabNet with little or no adjustments to the underlying code.

It is a known phenomenon that random weight initialization can impact the performance of the Transformer encoder architecture. Thus, to mitigate variance in the performance metrics between different model runs, we trained `CrabNet` using a fixed random seed of 42 for all training runs across all materials properties. We do note that in the case of random model initialization, the run-to-run variation between different trained models is a feature that could be taken advantage of for determining the epistemic uncertainty. Unfortunately, due to the sheer volume of materials properties investigated in this work and the limited compute resources available, we have not investigated this thus far. Finally, we note that all model training, evaluation and benchmarking (for `CrabNet`, `Roost`, `ElemNet`, and `RF`) was conducted on a single workstation PC equipped with an Intel i9-9900K CPU, 32 GB of DDR4 RAM, and two NVIDIA RTX 2080 Ti GPUs with 11 GB VRAM per GPU. The deep learning models were trained on the GPU, while the `RF` models were trained on the CPU.

## Reference Models

Predictions for all materials properties were generated using code from the `Roost` repository [9]. Minor adaptations were made to the code to allow for automated training and benchmarking. Overall, `Roost` generates consistently impressive results. `Roost` relies on a soft-attention mechanism used over a graph representation of the compound. This is in the same spirit of `CrabNet`, and both seek to generate vector representations for the elements in the system without using structure information. The residual network and robust loss function from `Roost` were helpfully adopted into our architecture [9].

Predictions from `ElemNet` were generated using default parameters using code from the repository [5]. Custom scripts were written to train and evaluate `ElemNet` over all materials properties data. `ElemNet` consistently under-performed compared to `Roost` and `CrabNet`. `ElemNet` failed to converge for multiple properties resulting in `NaN` (not a number) values in the model outputs. Examples of this occurring can be seen in the phonon peak and steels yield datasets. Here, we would like to note that `IRNet` [6] could also be benchmarked and compared in this study. However, due to the prohibitively large computational requirements, we chose not to train and evaluate `IRNet`.

We do however note that the OQMD performance reported in the `IRNet` manuscript [6] is consistently lower than both `Roost` and `CrabNet` for the same properties. These following values show the reported performance of `IRNet` vs. `HotCrab`, respectively, for formation enthalpy (0.048 eV vs. 0.031 eV), band gap (0.047 eV vs. 0.048 eV), energy per atom (0.070 eV vs. 0.033 eV), and volume per atom ($0.394\,\text{Å}^3$ vs. $0.277\,\text{Å}^3$).

We generate baseline `RF` metrics using a random forest regression with the `Magpie` `CBFV` as defined by Matminer [36]. This is done using the `scikit-learn`

Python package. The RF models were trained with $n_{\text{estimators}} = 500$ and default parameters.

# Data Availability

Data is provided in its cleaned and pre-split form to ensure reproducible results, and with the hope that other researchers find it useful when benchmarking their own approaches. The processed data that is used in this study can be found on the GitHub repository [59] at the address https://github.com/anthony-wang/CrabNet. All raw data as well as scripts to process and split the datasets can be found in the GitHub repository [58] at the address https://github.com/kaaiian/mse_datasets.

# Code Availability

We provide detailed instructions for the installation, training, and general usage of the open-source CrabNet on GitHub [59]. In addition, pre-trained network weights for the CrabNet models reported in this work are available for download [64].

The following files are available with this publication: (1) GitHub repository with the CrabNet source code, figures, and example property predictions:

https://github.com/anthony-wang/CrabNet, (2) pre-trained weights for the CrabNet models reported in this work:
https://doi.org/10.5281/zenodo.4633866, and (3) Supplementary Information.

Finally, we recommend that interested readers consult the paper "Machine Learning for Materials Scientists: An introductory guide towards best practices" [4] for a detailed treatment of best practices in machine learning and justification for many of the unmentioned experimental design decisions used in this work.

## 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 number 57438025) for their financial support. Special thanks is given to Dr. Aleksander Gurlo and Dr. Mathias Czasny for advising and supporting Anthony Yu-Tung Wang and for encouraging his collaborative stay at the University of Utah.

The authors thank the creators of AFLOW and Materials Project for the creation of the databases 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 `PyTorch`.

Last but not least, the authors thank OpenAI, the researchers at Hugging Face, and Adam King for their contribution to `TalkToTransformer.com`. The underlying Transformer-powered GPT-2 model was used to generate text for the closing lines of this publication.

## Author Contributions

AYTW and SKK jointly and in equal amounts conceived, developed the concept, and implemented the algorithms, code and visualizations described in this work. AYTW and SKK jointly analyzed the results.

RJM assisted with developing the architecture and provided insight and guidance during model optimization and training.

All authors discussed the results and contributed to the writing of the manuscript.

# Competing Interests

The authors declare no conflicts of interest.

# References

1. Maier, W. F., Stöwe, K. & Sieg, S. Combinatorial and high-throughput materials science. *Angew. Chem. Int. Ed. Engl.* **46,** 6016–6067 (2007).
2. Agrawal, A. & Choudhary, A. Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science. *APL Mater.* **4,** 053208 (2016).
3. Barnard, A. S. Best Practice Leads to the Best Materials Informatics. *Matter* **3,** 22–23 (2020).
4. Wang, A. Y.-T. *et al.* Machine Learning for Materials Scientists: An Introductory Guide toward Best Practices. *Chem. Mater.* **32,** 4954–4965 (2020).
5. Jha, D. *et al.* ElemNet: Deep Learning the Chemistry of Materials From Only Elemental Composition. *Sci. Rep.* **8,** 17593 (2018).
6. Jha, D. *et al.* IRNet: A General Purpose Deep Residual Regression Framework for Materials Discovery, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining – KDD '19,* 2385–2393 (ACM Press, New York, NY, 2019).
7. Xie, T. & Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.* **120,** 145301 (2018).
8. Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A. & Müller, K.-R. SchNet – A deep learning architecture for molecules and materials. *J. Chem. Phys.* **148,** 241722 (2018).
9. Goodall, R. E. A. & Lee, A. A. Predicting materials properties without crystal structure: deep representation learning from stoichiometry. *Nat. Commun.* **11,** 6280 (2020).
10. Ziletti, A., Kumar, D., Scheffler, M. & Ghiringhelli, L. M. Insightful classification of crystal structures using deep learning. *Nat. Commun.* **9,** 2775 (2018).

11. Faber, F. A., Lindmaa, A., von Lilienfeld, O. A. & Armiento, R. Crystal structure representations for machine learning models of formation energies. *Int. J. Quantum. Chem.* **115,** 1094–1101 (2015).

12. Faber, F. A., Lindmaa, A., von Lilienfeld, O. A. & Armiento, R. Machine Learning Energies of 2 Million Elpasolite (ABC2D6) Crystals. *Phys. Rev. Lett.* **117,** 135502 (2016).

13. Kong, C. S. *et al.* Information-theoretic approach for the discovery of design rules for crystal chemistry. *J. Chem. Inf. Model.* **52,** 1812–1820 (2012).

14. Fischer, C. C., Tibbetts, K. J., Morgan, D. & Ceder, G. Predicting crystal structure by merging data mining with quantum mechanics. *Nat. Mater.* **5,** 641–646 (2006).

15. Curtarolo, S., Morgan, D., Persson, K. A., Rodgers, J. & Ceder, G. Predicting crystal structures with data mining of quantum calculations. *Phys. Rev. Lett.* **91,** 135503 (2003).

16. Zhuo, Y., Mansouri Tehrani, A. & Brgoch, J. Predicting the Band Gaps of Inorganic Solids by Machine Learning. *J. Phys. Chem. Lett.* **9,** 1668–1673 (2018).

17. Kauwe, S. K., Graser, J., Vazquez, A. & Sparks, T. D. Machine Learning Prediction of Heat Capacity for Solid Inorganics. *Integr. Mater. Manuf. Innov.* **7,** 43–51 (2018).

18. Oliynyk, A. O. *et al.* High-Throughput Machine-Learning-Driven Synthesis of Full-Heusler Compounds. *Chem. Mater.* **28,** 7324–7331 (2016).

19. Hautier, G., Fischer, C. C., Jain, A., Mueller, T. & Ceder, G. Finding Nature's Missing Ternary Oxide Compounds Using Machine Learning and Density Functional Theory. *Chem. Mater.* **22,** 3762–3767 (2010).

20. Mansouri Tehrani, A. *et al.* Machine Learning Directed Search for Ultraincompressible, Superhard Materials. *J. Am. Chem. Soc.* **140,** 9844–9853 (2018).

21. Graser, J., Kauwe, S. K. & Sparks, T. D. Machine Learning and Energy Minimization Approaches for Crystal Structure Predictions: A Review and New Horizons. *Chem. Mater.* **30,** 3601–3612 (2018).

22. Choudhary, K., DeCost, B. & Tavazza, F. Machine learning with force-field-inspired descriptors for materials: Fast screening and mapping energy landscape. *Phys. Rev. Mater.* **2,** 083801 (2018).

23. Kauwe, S. K., Graser, J., Murdock, R. J. & Sparks, T. D. Can machine learning find extraordinary materials? *Comput. Mater. Sci.* **174,** 109498 (2020).

24. Gaultois, M. W. *et al.* Perspective: Web-based machine learning models for real-time screening of thermoelectric materials properties. *APL Mater.* **4,** 053213 (2016).

25. de Jong, M. *et al.* A Statistical Learning Framework for Materials Science: Application to Elastic Moduli of k-nary Inorganic Polycrystalline Compounds. *Sci. Rep.* **6,** 34256 (2016).

26. Glaudell, A. M., Cochran, J. E., Patel, S. N. & Chabinyc, M. L. Impact of the Doping Method on Conductivity and Thermopower in Semiconducting Polythiophenes. *Adv. Energy Mater.* **5,** 1401072 (2015).

27. Zhang, S. B. 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,** R881–R903 (2002).

28. Sheng, L., Wang, L., Xi, T., Zheng, Y. & Ye, H. Microstructure, precipitates and compressive properties of various holmium doped NiAl/Cr(Mo,Hf) eutectic alloys. *Mater. Des.* **32,** 4810–4817 (2011).

29. Mansouri Tehrani, A. *et al.* Atomic Substitution to Balance Hardness, Ductility, and Sustainability in Molybdenum Tungsten Borocarbide. *Chem. Mater.* **31,** 7696–7703 (2019).

30. Mihailovich & Parpia. Low temperature mechanical properties of boron-doped silicon. *Phys. Rev. Lett.* **68,** 3052–3055 (1992).

31. Qu, Z., Sparks, T. D., Pan, W. & Clarke, D. R. Thermal conductivity of the gadolinium calcium silicate apatites: Effect of different point defect types. *Acta Mater.* **59,** 3841–3850 (2011).

32. Sparks, T. D., Fuierer, P. A. & Clarke, D. R. Anisotropic Thermal Diffusivity and Conductivity of La-Doped Strontium Niobate Sr2Nb2O7. *J. Am. Ceram. Soc.* **93,** 1136–1141 (2010).

33. Grimvall, G. Thermophysical Properties of Materials 1st ed. (North Holland, Amsterdam, 1999).

34. Gaumé, R., Viana, B., Vivien, D., Roger, J.-P. & Fournier, D. A simple model for the prediction of thermal conductivity in pure and doped insulating crystals. *Appl. Phys. Lett.* **83,** 1355–1357 (2003).

35. Murdock, R. J., Kauwe, S. K., Wang, A. Y.-T. & Sparks, T. D. Is Domain Knowledge Necessary for Machine Learning Materials Properties? *Integr. Mater. Manuf. Innov.* **9,** 221–227 (2020).

36. Dunn, A., Wang, Q., Ganose, A., Dopp, D. & Jain, A. Benchmarking materials property prediction methods: the Matbench test set and Automatminer reference algorithm. *npj Comput. Mater.* **6** (2020).

37. Vaswani, A. *et al.* Attention Is All You Need, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., La Jolla, LA, 2017).

38. Tang, G., Müller, M., Rios, A. & Sennrich, R. Why Self-Attention? A Targeted Evaluation of Neural Machine Translation Architectures, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* 4263–4272 (Association for Computational Linguistics, Stroudsburg, PA, 2018).

39. Al-Rfou, R., Choe, D., Constant, N., Guo, M. & Jones, L. Character-Level Language Modeling with Deeper Self-Attention. *AAAI* **33,** 3159–3166 (2019).

40. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics,* 4171–4186 (Association for Computational Linguistics, Stroudsburg, PA, 2019).

41. Yu, A. W. *et al.* QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension, in *Proceedings of the International Conference on Learning Representations (ICLR)* (Online, 2018).

42. Yang, Z. *et al.* XLNet: Generalized Autoregressive Pretraining for Language Understanding, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., La Jolla, LA, 2019).

43. Huang, C.-Z. A. *et al.* Music Transformer, in *Proceedings of the International Conference on Learning Representations (ICLR)* (Online, 2019).

44. Zhang, H., Goodfellow, I., Metaxas, D. & Odena, A. Self-Attention Generative Adversarial Networks, in *Proceedings of the 36th International Conference on Machine Learning (ICML),* 7354–7363 (PMLR, 2019).

45. Dai, T., Cai, J., Zhang, Y., Xia, S.-T. & Zhang, L. Second-Order Attention Network for Single Image Super-Resolution, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 11057–11066 (IEEE, 2019).

46. Zhang, Y. *et al.* Image Super-Resolution Using Very Deep Residual Channel Attention Networks, in *Computer Vision – ECCV 2018.* **11211,** 294–310 (Springer International Publishing, Cham, Switzerland, 2018).

47. Zhang, Y., Li, K., Li, K., Zhong, B. & Fu, Y. Residual Non-local Attention Networks for Image Restoration, in *Proceedings of the International Conference on Learning Representations (ICLR)* (Online, 2019).

48. Kim, T. H., Sajjadi, M. S. M., Hirsch, M. & Schölkopf, B. Spatio-Temporal Transformer Network for Video Restoration, in *Computer Vision – ECCV 2018.* **11207,** 111–127 (Springer International Publishing, Cham, Switzerland, 2018).

49. Wang, X., Chan, K. C. K., Yu, K., Dong, C. & Loy, C. C. EDVR: Video Restoration With Enhanced Deformable Convolutional Networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops,* 1954–1963 (IEEE, 2019).

50. Vinyals, O. *et al.* Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **575,** 350–354 (2019).

51. Baker, B. *et al.* Emergent Tool Use From Multi-Agent Autocurricula, in *Proceedings of the International Conference on Learning Representations (ICLR)* (Online, 2020).

52. Zheng, S., Yan, X., Yang, Y. & Xu, J. Identifying Structure-Property Relationships through SMILES Syntax Analysis with Self-Attention Mechanism. *J. Chem. Inf. Model.* **59,** 914–923 (2019).

53. Schwaller, P. *et al.* Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Cent. Sci.* **5,** 1572–1583 (2019).

54. Clement, C. L., Kauwe, S. K. & Sparks, T. D. Benchmark AFLOW Data Sets for Machine Learning. *Integr. Mater. Manuf. Innov.* **9,** 153–156 (2020).

55. Bartel, C. J. *et al.* A critical examination of compound stability predictions from machine-learned formation energies. *npj Comput. Mater.* **6,** 97 (2020).

56. Kirklin, S. *et al.* The Open Quantum Materials Database (OQMD): assessing the accuracy of DFT formation energies. *npj Comput. Mater.* **1** (2015).

57. Ward, L. *et al.* Matminer: An open source toolkit for materials data mining. *Comput. Mater. Sci.* **152,** 60–69 (2018).

58. Kauwe, S. K. Online GitHub repository for mse_datasets. `https://github.com/kaaiian/mse_datasets` (2020).

59. Wang, A. Y.-T. & Kauwe, S. K. Online GitHub repository for the paper "Compositionally-Restricted Attention-Based Network for Materials Property Prediction". `https://github.com/anthony-wang/CrabNet` (2020).

60. Tshitoyan, V. *et al.* Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **571,** 95–98 (2019).

61. Ward, L., Agrawal, A., Choudhary, A. & Wolverton, C. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Comput. Mater.* **2,** 16028 (2016).

62. Zhang, M. R., Lucas, J., Hinton, G. & Ba, J. Lookahead Optimizer: k steps forward, 1 step back, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., La Jolla, LA, 2019).

63. You, Y. *et al.* Large Batch Optimization for Deep Learning: Training BERT in 76 minutes, in *Proceedings of the International Conference on Learning Representations (ICLR)* (Online, 2020).

64. Wang, A. Y.-T., Kauwe, S. K., Murdock, R. J. & Sparks, T. D. Trained network weights for the paper "Compositionally-Restricted Attention-Based Network (CrabNet)". `https://doi.org/10.5281/zenodo.4633866` (2021).

# Figure Legends

Figure 1: **Benchmark results.** MAE scores of Roost, CrabNet, one-hot encoded CrabNet (HotCrab), and ElemNet on the held-out test datasets, compared with the random forest (RF) baseline for (a) the Matbench dataset and (b) the Extended dataset. Cells are colored according to relative MAE performance within each row (blue is better, and red is worse). A NaN (not a number) value is reported for instances where the models failed to converge on a given material property. Here we present model results trained using chemical information (Roost, CrabNet), no chemical information (HotCrab, ElemNet), and a standard CBFV (RF).

Figure 2: **Visualization of self-attention in one compound.** Displayed are the four attention heads (a, b, c, and d) from the first layer of a CrabNet model trained on mp_bulk_modulus and evaluated on the composition $Al_2O_3$. Each row represents an element in the system. Each column represents an element being attended to. Each element's fractional amount is shown on the $x$-axis. The values in the attention matrix are scores representing element-element interactions for the compound. As an example, in head (a), $Al_{0.4}$ and $O_{0.6}$ are attending strongly to each other, with attention scores of 1.00 between these two elements.

Figure 3: **Visualization of average attention for one dataset.** The average attention from each of the four attention heads (a, b, c, d) from the first layer of a CrabNet model trained on the aflow___Egap data is shown for systems containing Si. The heatmap shows the average amount of attention that Si dedicates to the other elements in Si-containing compounds. The darker the coloring, the more strongly Si attends to that element. We can see that each attention head exhibits its own behavior, and attends to different groups of elements. Interestingly, head (a) attends to common $n$-type dopants and head (c) attends to many transition metals, whereas heads (b) and (d) have unfamiliar element groupings.

Figure 4: **Overall element contribution to property predictions.** Average contribution of all elements to bulk modulus predictions, computed from the AFLOW_bulk_modulus dataset, (a) plotted on a periodic table and (b) plotted as a distribution showing the per-element contribution amounts of Li and W, respectively, in all the compounds. The darker colored elements in the periodic table contribute more towards a compound's bulk modulus value.

Figure 5: **Element contribution to property prediction as a function of composition.** Model predictions over the $Si_xO_{1-x}$ system using a model trained on the aflow___Egap data. The $x$-axis is the fractional amount of Si. The $y$-axis shows the predicted band gap value at a given composition. The blue and red lines are the individual element contributions to the prediction, as predicted by `CrabNet`. The gray shading represents the aleatoric uncertainty for each prediction.

Figure 6: **`EDM` featurization scheme.** Schematic illustration of the element-derived matrix (`EDM`) representation for $Al_2O_3$, where $B$ represents the batch, $d_{\text{model}}$ is the element features, and $n_{\text{elements}}$ represents the number of elements. Composition slices, when concatenated across batch dimension $i$, form an `EDM` tensor which is then used as the model input to `CrabNet`. When a chemical formula has fewer elements than rows in the `EDM`, the extra data rows are filled with zeros.

Figure 7: **Schematic of an attention block in the `CrabNet` architecture.** (a) The initial projection of the input `EDM` into the $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ tensors. (b) The scaled dot-product attention operation obtaining the self-attention matrix and the updated $\mathbf{Z}$ element representation. The batch dimension is not shown in (b) to improve legibility.

Figure 8: **Overall `CrabNet` architecture and prediction of material property and uncertainty.** (a) Schematic of the `CrabNet` architecture including the input `EDM`, the self-attention layers (repeated $N$ times), the updated and final element representations ($\texttt{EDM}'$ and $\texttt{EDM}''$), the residual network, and the final model output. (b) The calculation steps for element-contributions and prediction of the target and uncertainties. The $p'$ and $u'$ vectors represent the element-proto-contributions and the element uncertainties, respectively. $y'$ represent the element-contributions. The material property is obtained by taking the mean of element-contributions ($y'$) for each compound. Similarly, the mean of the element-uncertainties ($u'$) gives us the estimated aleatoric uncertainty.

Table 1: **Benchmark datasets.** List of all 28 material properties used to benchmark the ML models in this work, together with the dataset size and the original training, validation, and test set proportions. The materials properties listed in the top and bottom halves are Matbench and Extended datasets, respectively.

Table 2: **List of default model parameters of `CrabNet`.**