npj | computational materials

Article

Published in partnership with the Shanghai Institute of Ceramics of the Chinese Academy of Sciences

https://doi.org/10.1038/s41524-024-01509-x

Feature engineering descriptors, transforms, and machine learning for grain boundaries and variable-sized atom clusters

C. Braxton Owens[1], Nithin Mathew[2], Tyce W. Olaveson[3], Jacob P. Tavenner[4], Edward M. Kober[2], Garritt J. Tucker[5], Gus L. W. Hart[3] ✉ & Eric R. Homer[6] ✉

Obtaining microscopic structure-property relationships for grain boundaries is challenging due to their complex atomic structures. Recent efforts use machine learning to derive these relationships, but the way the atomic grain boundary structure is represented can have a significant impact on the predictions. Key steps for property prediction common to grain boundaries and other variable-sized atom clustered structures include: (1) describing the atomic structure as a feature matrix, (2) transforming the variable-sized feature matrix to a fixed length common to all structures, and (3) applying a machine learning algorithm to predict properties from the transformed matrices. We examine how these steps and different combinations of engineered features impact the accuracy of grain boundary energy predictions using a database of over 7000 grain boundaries. Additionally, we assess how different engineered features support interpretability, offering insights into the physics of the structure-property relationships.

Due to the impact of grain boundaries (GBs) on material properties[1–4], there is a need to better understand the relationship between the structure of a GB and its corresponding properties. With expanding computing power, increasingly large amounts of data, and advances in data-driven approaches, there has been a push for suitable representations of GBs in order to predict their properties[5–37]. However, accurate property prediction is not the only measure of success. Models and representations that provide insight into structure-property relationships are key to advancing our understanding.

Representing a GB starts with defining its structure, which has both macroscopic and microscopic characteristics. Macroscopically, five degrees of freedom are used to define the GB character: Three to define a misorientation between two crystals, (often given by a rotation axis $[uvw]$ and angle θ), and two to define a boundary plane (hkl). Microscopically, the positions of the atoms result in $3n$ degrees of freedom. Additionally, a GB can assume various metastable configurations under any given set of macroscopic constraints[13,24,38–42].

While there have been impressive developments in macroscopic representations for better understanding GBs[12,35,43–46], the atomic structure is what defines a GB's properties. This article concentrates on the microscopic structure-property relationships since the macroscopic structure acts as a constraint on the microscopic structure.

One microscopic method for defining a GB is the structural unit model[47–50]. This model describes the atomic structure of a quasi-two-dimensional GB as a series of repeating atomic "structural units" that are characteristic of the boundary's local atomic arrangement. This simplifies analysis of its geometry and properties as long as it is quasi-two-dimensional. In recent years, the structural unit model has been modified to more accurately represent a GB by considering the effects of metastable structures[51]. Other early methods for defining local atomic environments for characterizing GBs include: the centrosymmetry parameter (CSP)[52], Voronoi index[53], excess volume[54], common neighbor analysis (CNA)[55], the Polyhedral Unit Model[56], and local entropy[57].

Although the early methods have been used mainly to classify GB atoms[29,36,58], they have also been used for machine learning predictions of atomic level properties[32,37]. Advanced methods, some of which are described below, have also been used to predict atomic-level properties in GBs[59].

[1]Department of Computer Science, Brigham Young University, Provo, 84602 UT, USA. [2]Group T-1, Theoretical Division, Los Alamos National Laboratory, Los Alamos, 87544 NM, USA. [3]Department of Physics and Astronomy, Brigham Young University, Provo, 84602 UT, USA. [4]KBR, Inc., Intelligent Systems Division, NASA Ames Research Center, Moffett Field, 94035 CA, USA. [5]Department of Physics, Baylor University, Waco, 76798 TX, USA. [6]Department of Mechanical Engineering, Brigham Young University, Provo, 84602 UT, USA. ✉e-mail: gus.hart@gmail.com; eric.homer@byu.edu

However, when using machine learning to predict the properties of a GB as a whole, the descriptor must be transformed in some way to achieve a consistent feature size. This notion of transformation is an important one for variable-sized atomic structures such as GBs because different GBs will have different numbers of atoms (features) in their structure. Transformation is one step in a three-step feature engineering process common to almost all machine learning predictions for variable-sized atom structures. These steps are illustrated in Fig. 1 and are described as follows: 1. **Describe** the atomic structure with an encoding algorithm, descriptor, or fingerprint of some kind, which is often represented as a matrix or vector. 2. **Transform** the mapping of the variable length descriptor for each structure to a fixed length descriptor common across all structures in a dataset. 3. Apply **Machine Learning** models or regression algorithms, to learn and then predict the property of a given atomic structure.

In the paragraphs below we illustrate the consistency of these steps, occasionally combined in different orders or sequences, in numerous applications of machine learning to predict properties based on microscopic GB structure. These examples will also serve to highlight the variety of descriptors, transforms, and machine-learning algorithms employed by the community.

Snow et al.[36] utilized the graphs underlying CNA to **describe** the atoms in each GB, categorizing them into 2205 distinct environments. They then performed principal component analysis to reduce these environments into 84 principal components, which constitutes a second **description** step. To standardize the representation size, a **transform** was applied, representing each GB by the proportions of the 84 components present. This transformed representation was used as input to a linear regression **machine learning** model to predict GB energy.

Guziewiski et al.[15] also explore this concept of proportions, utilizing the diamond-structure identification and the polyhedral template matching algorithms to **describe** GB atoms. This was **transformed** into a fixed-length density metric for each GB by counting the number of atoms within each polyhedral template class and normalizing this by the GB area. Random forest **machine learning** models were then used to predict both the GB energy and the tensile strength of the GB.

Gomberg et al.[29] utilized a specialized pair correlation function[60,61] to **describe** their GB structures. This function is unique due to its use of a probability distribution function, allowing equal sampling for each GB and simultaneously **transforming** the descriptor into a fixed length. This representation was further refined to the first two principal components, constituting a second **describe** step, which is then used as input into a regression **machine learning** model. Dang and Yu extended Gomberg's method by incorporating the standard deviation of the pair correlation function through a weight parameter[6].

More recently, the GB community has used atomic structure descriptors developed by the machine-learned interatomic potential community. These descriptors are attractive because they are inspired by the symmetries and physical response inherent to the atoms, as described by Musil et al.[62]. Rosenbrock et al.[30] implemented one of these physics-inspired descriptors, called the smooth overlap of atomic positions (SOAP), to **describe** the GB and then **transformed** the SOAP descriptor into a fixed-length vector by averaging over the atom environments. These are then used to predict GB energy using a support vector **machine learning** model. Later, Fujii et al.[9] used SOAP to calculate a local distortion factor, **describing** how similar a GB atom's environment is to a bulk atom's environment. This was then **transformed** into a fixed-length using complete-linkage clustering. The GB thermal conductivity was then predicted by a ridge regression **machine learning** algorithm.

In this paper, we examine the impact of feature engineering different descriptors, transforms, and machine learning models to predict GB energy, as illustrated in Fig. 1. The exact **descriptors, transforms**, and **machine learning** algorithms we employ are listed in Fig. 2 and described in detail in the Methods Section. The feature engineering is tested on a dataset comprising 7304 aluminum GBs, which provides comprehensive coverage of the 5-dimensional macroscopic space of crystallographic character[38,63]. The interplay of various descriptors, transforms, and machine learning algorithms are analyzed for their effect on the accuracy of their predictions. In addition, we examine how feature reduction impacts the results for select cases. Finally, we assess the interpretative ability of some key descriptors to establish meaningful connections to the inherent structure of the GB. Prioritizing interpretability is imperative to increase our understanding of atomic GB structure-property relationships.

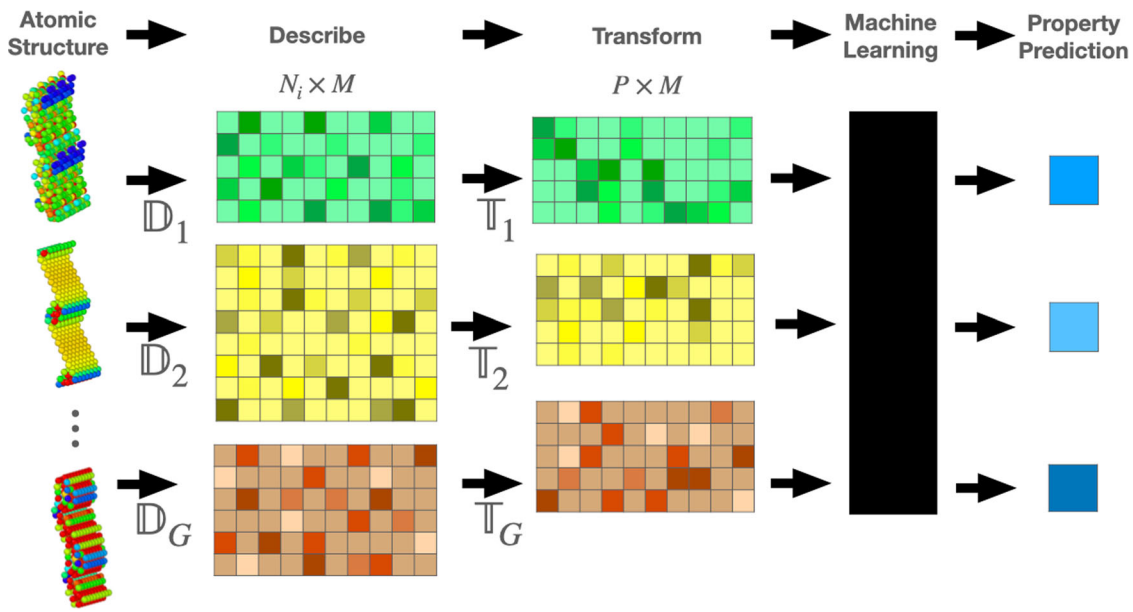

Fig. 1 | The workflow for predicting material properties from variable-sized atomic structures. Starting with various atomic structures (left), each structure is described using a matrix of size $N_i \times M$ that captures relevant features. These matrices are then transformed into a different representation of size $P \times M$. The transformed matrices are input into a machine learning model, which predicts the material properties (right), shown as blue squares. This approach allows for systematic analysis and prediction of properties based on the atomic-level description of materials.

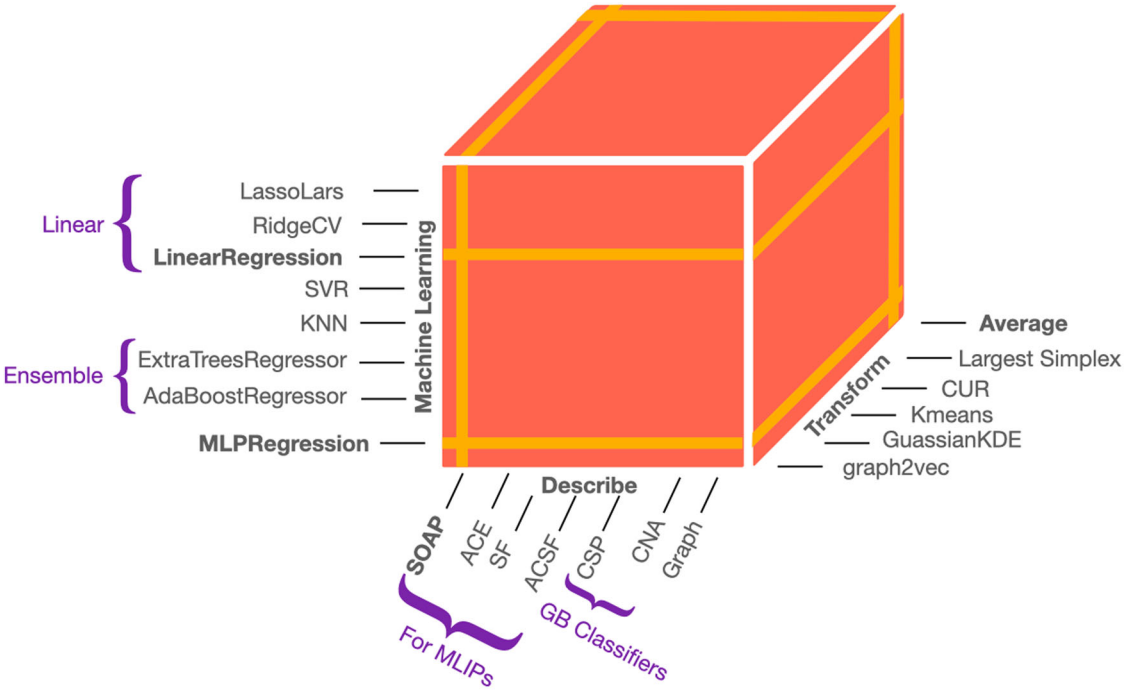

Fig. 2 | Graphic illustrating the various combinations of descriptors, transforms, and machine learning algorithms employed. The light orange planes denote cross-sections of this space that are examined in this work.

Results

Accuracy of Predictions

As illustrated in Fig. 1, there are essentially 3 methods or "knobs" that can be adjusted to improve the accuracy of predictions: **descriptor, transform**, and **machine learning** algorithm. As noted above, the descriptors, transforms, and machine learning algorithms we employ are listed in Fig. 2 and described in detail in the Methods Section.

We start by examining Fig. 3, which shows parity plots comparing machine learning predictions of GB energy against the GB energy values calculated in LAMMPS, as reported in ref. 38. There is one plot for each of the 7 descriptors examined in this work: The Atomic Cluster Expansion (ACE), SOAP, Atom Centered Symmetry Functions (ACSF), Strain Functional (SF) descriptors, a Graph description, CSP, and CNA. For each of these 7 descriptors, the results are reported for the combination of transform and machine learning algorithm that resulted in the highest overall accuracy predicted for GB energy using this dataset. The accuracy of predictions is measured by the mean absolute error (MAE) and R-squared (R^2) values. These metrics are used in tandem to best illustrate the performance of a given model. Finally, these 7 descriptors are accompanied by a parity plot labeled as "Random SOAP" where the SOAP descriptor is used as input, but with the GB energy values shuffled so that every SOAP vector points to a random GB energy value in the set. This serves as a worst-case reference value since a shuffled dataset would be expected to have no correlations.

In examining Fig. 3, the SOAP descriptor combined with LinearRegression achieves the highest accuracy, with a low MAE of 3.89 mJ/m^2 and a high R^2 of 0.99, indicating the near-perfect correlation between predicted and actual values. In contrast, the Random SOAP model, where GB energies are shuffled, has a high MAE of 46.96 mJ/m^2 and a negative R^2 of -0.23, confirming no predictive capability. While ACE and SF descriptors also achieve high accuracy, ACSF exhibit intermediate performance and descriptors like graph (graph2vec), CNA, and CSP exhibit significantly higher MAE and lower R^2, indicating poorer predictive performance. This suggests that the higher complexity descriptors do indeed capture more relevent information for predicting grain boundary energy.

Figure 3 illustrates key aspects of feature engineering used to predict GB energy. First, it can be seen that due to the nature of this dataset, with many GBs concentrated about the mean GB energy value of 497 mJ/m^2, it is possible to obtain a relatively low MAE, even in the case of the "Random SOAP" model. It is for this reason that we report both the MAE and the R^2 values. Caution must be exercised in assuming a model is good just because the MAE is low. One must also see a high R^2 value to show that the model results in correlated predictions; a negative value for the R^2 metric indicates that it would have been better to simply predict the mean.

Second, one can see that the 'average' transform is selected as the transform providing the most accurate predictions in four of seven cases. Third, in three of seven cases, the machine learning algorithm that provides the highest accuracy is LinearRegression. In the other four of seven cases, MLPRegression is the most accurate. But the three cases with LinearRegression have much better predictions than those with MLPRegression. Fourth, one can see that the combination of MAE and R^2 values provide a nice summary of the accuracy that can be visibly seen in the parity plots. Fifth, the stark contrasts of the MAE and R^2 values between the SOAP and "Random SOAP" models illustrates that there is valuable information in the features of the averaged SOAP that is correlated with the GB energy of a given structure. This is strictly true when comparing SOAP and "Random SOAP" and likely true when comparing "Random SOAP" with the other descriptors, which categorize the atomic information of the GBs in distinct manners.

While important insights can be gained from these comparisons in Fig. 3, great care must be exercised because they are not direct comparisons; the models are different. Better comparisons can be made by holding as many variables constant between the models as possible. The three key steps (**describe, transform**, and **machine learning**) represent a 3-dimensional space of combinations, as illustrated in Fig. 2. However since not all combinations were evaluated, we choose to examine 2-dimensional subsets, holding constant one of the three methods, as illustrated by the yellow bands in Fig. 2. The methods that are held constant are chosen because they typically perform better than their counterparts for the GB energy predictions examined in this work.

The first subset we analyze compares different descriptors and machine learning algorithms, keeping the averaging transform constant since it performes well for 4 of 7 descriptors in Fig. 3. The accuracy comparison for

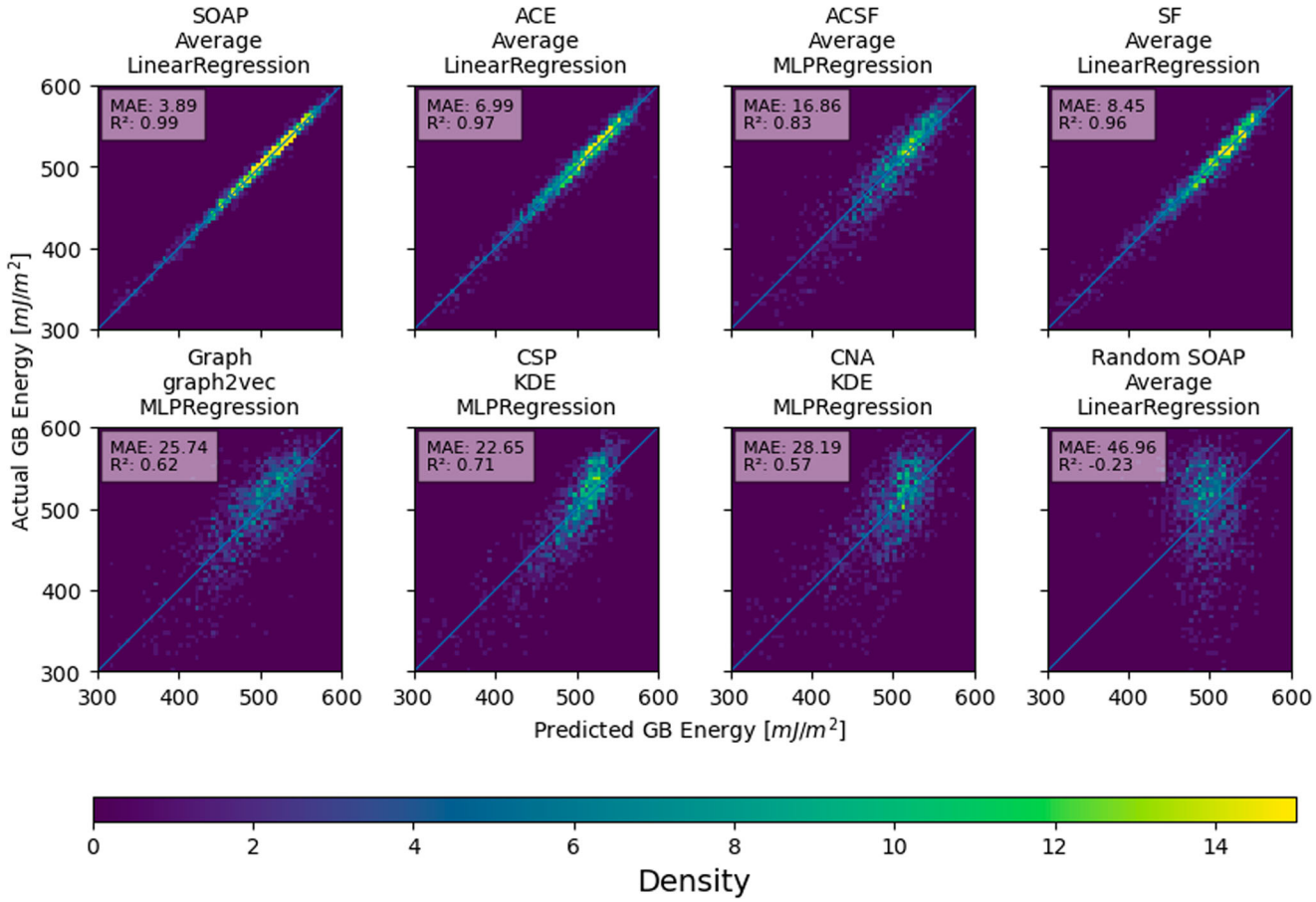

Fig. 3 | Summary figure illustrating the accuracy of each GB descriptor paired with its optimal transform and machine learning algorithm. Each subplot uses color-coded density levels to highlight the relationships between predicted and actual values. MAE is in units of mJ/m^2.

this subset is given in Fig. 4a. The ACE, SOAP, and SF descriptors performed exceptionally well across multiple machine learning algorithms. The R^2 values were consistently above 0.95 in the majority of cases, indicating strong predictive capabilities, while the MAE values were comparatively low, < 10 mJ/m^2 in the majority of cases, reflecting accuracy in the energy predictions. ACSF had higher errors (MAE just less than 20 mJ/m^2 and R^2 around 0.8). CSP, CNA, and the graph descriptor had errors around 25 mJ/m^2 and $R^2 < 0.67$.

One might be tempted to make quick judgements about the quality of the different descriptors, but caution must be exercised because the number of features for each descriptor vary drastically, as shown by Table 2 in the Methods section. Each descriptor forms a unique basis to represent an environment. For the SOAP, ACE, and SF techniques, the user specifies the desired order of radial basis functions, the degree of spherical harmonics, or the polynomial order to be included in the representation. Similarly, for the ASCFs, the user selects sets of 2- and 3-body functions. The graph used in combination with graph2vec creates a nearest-neighbor-connected graph where the user defines a cutoff distance for the nearest neighbors. In contrast, the CNA and CSP descriptors are both singular-valued quantities for each atom, though it is noted that CNA is built on top of local graphs that could be used instead of the integer classification, as in[36].

In the cases where the user picks the level of expansion, the user can determine what cutoff can be used to obtain a desired level of accuracy; this trades computation time and memory required for a larger basis that hopefully captures more of the physics. For example, when we increased the number of ACSF features from 8 to 37, the R^2 value increased from 0.65 to 0.81 but it required significantly more computation time. One could also compare the fact that ACSFs and SFs have approximately the same number of terms, 37 and 36 respectively, but the ACSFs perform worse than SFs. This suggests that more or different interaction pairs are probably required to detect the structural features that determine GB energy with ACSF. In fact, with a greater understanding of the ACSF implementation, it is possible that one could obtain higher accuracy even with the same number of terms. Although we see that this method of "covering your bases" by increasing the number of features does quite well in terms of accuracy, there is power in a pre-training choice of basis based on physical beliefs about the material, which ideas are discussed in the interpretability section of the results.

What is remarkable about the two singular-valued descriptors, CSP and CNA, is that, despite being less accurate than the SOAP, ACE, SF, and ACSF descriptors, they still achieve respectable accuracy (c.f. Fig. 4a). In fact, it is simple enough that we show the equation for the LinearRegression model, $\gamma = 280.26 \times x_{\mathrm{CSP}} + 173.14$, where γ is the GB energy and x_{CSP} is the average of the scalar CSP values for a given boundary. Similarly for CNA, the single coefficient linear function is $\gamma = 505.88 \times x_{\mathrm{CNA}} - 370.71$, where x_{CNA} is the average of the integer CNA values that refer to different structure types.

Finally, in Fig. 4a, we include the graph descriptor, despite its use of a different transform, graph2vec. This descriptor performs worse than the CSP and CNA descriptors, even though it encodes more information than the singular-valued descriptors. However, as with the other descriptors, several parameters could be optimized for better predictions, including the cutoff distance of the graph, the weighting of the graph, and the selection of subgraphs in the graph2vec transform.

In summary, the effect of descriptor on accuracy in Fig. 4a shows that in general, more features is better. The density-based descriptors, many of which are created for machine-learned interatomic potentials, appear to be better at capturing the complex and intricate nature of the local atomic environments.

(a) Fixed Transform: Averaging

ML	SOAP	ACE	SF	ACSF	CSP	CNA	Graph	Random SOAP
LassoLars	5.9/0.98	9.1/0.95	8.4/0.96	19.6/0.77	24.2/0.67	24.15/0.67	25.7/0.62	40.23/-0.02
RidgeCV	4.7/0.98	9.2/0.95	8.4/0.96	18.2/0.8	24.2/0.67	24.15/0.67	25.7/0.62	65.1/-1.54
LinearRegression	3.9/0.99	6.9/0.97	8.4/0.96	18.1/0.81	24.2/0.67	24.2/0.67	25.7/0.62	86.2/-26.3
SVR	6.8/0.97	10.7/0.93	23.7/0.71	22.65/0.69	24.1/0.67	24.1/0.67	25.4/0.61	38.5/-0.3
KNN	19.13/0.78	16.8/0.84	18.0/0.81	26.6/0.63	24.33/0.66	29.1/0.54	31.7/0.42	42.5/-0.07
ExtraTreesRegressor	15.8/0.86	12.2/0.92	11.3/0.93	24.0/0.69	30.5/0.47	30.5/0.48	31.7/0.43	58.7/-1.2
AdaBoostRegressor	25.9/0.64	16.47/0.86	18.8/0.8	29.2/0.58	24.6/0.65	28.9/0.55	37.64/0.31	55.3/-0.3
MLPRegression	9.4/0.94	10.8/0.93	8.5/0.95	17.7/0.82	24.1/0.67	28.9/0.55	33.8/0.39	88.8/-3.19

(b) Fixed Descriptor: SOAP

ML	Average	Largest Simplex	CUR	Kmeans
LassoLars	5.9/0.98	29.2/0.51	29.8/0.52	43.3/0.32
RidgeCV	4.7/0.98	26.8/0.58	30.32/-0.75	43.5/-0.02
LinearRegression	3.9/0.99	69.7/-11	104.4/-7.8	46.0/-0.14
SVR	6.8/0.97	29.1/0.5	DNC	DNC
KNN	19.1/0.78	36.9/0.21	39.1/0.16	39.5/0.14
ExtraTreesRegressor	15.8/0.86	30.8/0.47	31.4/0.44	41.3/0.03
AdaBoostRegressor	25.9/0.64	DNC	DNC	40.1/0.1
MLPRegression	9.4/0.94	40.8/0.12	DNC	69.0/-1.5

(c) Fixed ML: LinearRegression

Descriptor	Average	Largest Simplex	CUR	Kmeans	GaussianKDE	graph2vec
SOAP	3.9/0.99	69.7/-11.1	104.4/-7.8	46.0/-0.14	—	—
ACE	6.9/0.97	33.5/-3.16	42.3/-6.28	91.4/-3.6	—	—
SF	8.4/0.96	36.4/0.24	32.1/0.43	38.0/0.24	—	—
ACSF	18.1/0.8	34.1/0.34	30.5/0.48	32.3/0.28	—	—
CSP	24.2/0.67	—	—	—	22.6/0.71	—
CNA	30.0/0.52	—	—	—	28.8/0.52	—
graph	—	—	—	—	—	25.7/0.62

(d) Fixed ML: MLPRegression

Descriptor	Average	Largest Simplex	CUR	Kmeans	GaussianKDE	graph2vec
SOAP	9.4/0.94	40.8/0.12	DNC	69.6/-1.5	—	—
ACE	10.8/0.93	33.1/0.39	34.1/0.37	65.5/-1.23	—	—
SF	8.5/0.95	70.44/-1.5	67.3/-1.31	88.5/-2.9	—	—
ACSF	17.1/0.82	38.0/0.21	39.4/0.16	66.95/-1.31	—	—
CSP	24.1/0.67	—	—	—	22.2/0.72	—
CNA	28.9/0.55	—	—	—	28.4/0.56	—
graph	—	—	—	—	—	33.8/0.39

Fig. 4 | Tables comparing the accuracy of different combinations of descriptors, transforms, and machine learning techniques. The entries in each cell list the MAE/R^2 as a measure of the accuracy (with MAE units in mJ/m^2), and the cell is colored according to this accuracy. Note that in all cases the Graph descriptor uses the graph2vec transform. The Random SOAP is included for reference to illustrate the accuracy on a randomized list of GB energies; values near this accuracy are considered to be no better than predicting a random output. If the models did not converge in a reasonable time their results were left as blank cells. Approaches that did not converge are labeled with DNC.

In examining the role of machine learning algorithms for a given descriptor, one can see that the linear models (LinearRegression, LASSO, RidgeCV) generally outperform other types of machine learning models for nearly all descriptors when using the average transform (c.f. Fig. 4a). The SVM (SVR) and Nearest Neighbor (KNN) models frequently rank next in performance, followed closely by Neural Network (MLPRegression) and ensemble (AdaBoostRegressor) methods. All these models perform better than the "Random SOAP" input, demonstrating their ability to identify meaningful correlations between the features and GB energy.

The second subset we examine in the 3-D space holds the SOAP descriptor constant and varies the transform and machine learning methods, as illustrated in Fig. 2. The results for this subset are illustrated in Fig. 4b, where it should be noted that the average column in Fig. 4b is the same as the SOAP column in Fig. 4a due to the intersection of the two 2-D cross-sections. This analysis shows that in all cases in Fig. 4b, the averaging transform significantly outperforms the other methods; MAE is less than 10 mJ/m^2 for the average transform in most cases and greater than 26 mJ/m^2 for the other transforms in all cases.

All the transforms assume some prior on the important features to transform. The average transform assumes that an average environment is the most important information to preserve. KMeans clustering assumes that the clustering in the dataset and the locations of those clusters is the most important information to preserve. Largest simplex assumes that it is important to represent the data with subsections of the data that are far apart and maximize the simplex volume. CUR assumes that specific subsets, actual rows and columns of the original data, are critical and obtains these through matrix decomposition.

Given that GB energy is calculated from the sum of the energy of all the atoms divided by the area of the boundary, it is not surprising that the best transform is an average of the atomic environments. In other words, the assumption behind the average transform aligns closely with the calculation method for GB energy, making it a suitable representation. Conversely, the

assumptions underlying other transforms do not as accurately capture the relationship between the atomic collection and GB energy. However, there may be other cases where a different transform better matches with the property of interest. For example, in some cases, extreme values of a distribution control the behavior, such as in fracture, and a different transform may better capture that relationship. Therefore, we hypothesize that the best transform for accurate predictions is one that preserves the relationship between the way a collection of atoms relates to the property of interest.

However, it is also possible that the choices made in this work about the transform hyperparameters resulted in poor predictions since we did not seek to optimize these hyperparameters. For example, the target rank used in the CUR transform is 20 and the largest simplex transform employed 10 dimensions; it is not clear that these values are sufficient or insufficient. Similarly, the KMeans transform employed 100 clusters and it is not clear that this is representative of the number of clusters in any of the descriptors.

It is also worth noting that the SOAP descriptor appears to fall victim to the curse of high dimensionality when transformed with KMeans clustering. On average, the KMeans clustering transform performs worse when applied to SOAP. This is likely because the high dimensionality of SOAP descriptors leads to a phenomenon known as "distance uniformity"[64]. In high-dimensional spaces, feature values tend to be equidistant from each other, making it difficult for clustering algorithms to distinguish between similar and dissimilar data points.

The third and fourth 2-D subsets we examine from the 3-D space illustrated in Fig. 2 hold a different machine-learning algorithm constant. Specifically, the LinearRegression model was picked because of its high accuracy and MLPRegression was also picked because of the popularity of deep learning models. The results for these subsets are presented in Fig. 4c,d where the effect of different descriptors and transforms can be seen. First, in comparing the two tables, the average transform is better with LinearRegression in all but the case of ACSF. The higher accuracy between LinearRegression and MLPRegression is evenly divided for the largest simplex transform. MLPRegression is better in two of three cases for the CUR transform and in one of four cases for the KMeans transform. However, in many of these cases, the accuracy values approach or exceed predictions by "Random SOAP", making it difficult to judge the value of the improvements. Furthermore, these all perform worse than the average transform.

In the case of CSP and CNA, the GaussianKDE transform performs better than the average transform for both LinearRegression and MLPRegression, with the exception of CNA by LinearRegression. Also for these, MLPRegression performs better than LinearRegression for three of four cases considered. In these singular-valued descriptors, the more sophisticated GaussianKDE transform and MLPRegression allow it to obtain slightly better predictions.

This examination of the effect of all three key steps (**describe, transform, machine learning**) shows that the descriptor plays an outsize role in the quality of the predictions. However, the transform of the features also plays an important role and some important information can be lost at this step if care is not exercised. Finally, the machine learning algorithm appears to play more of a secondary role; if the features are correlated with the property of interest, multiple algorithms can often extract the relationship (though some methods appear to perform better than others depending on the circumstance).

The accuracy of the model will also be affected by the dataset used as input to training of the model, as evidenced by recent work on machine-learned interatomic potentials[65]. In this work, we examined 7174 minimum energy GB structures. However, in generating the dataset, more than 43 million metastable GB structures were generated and preserved. Since training and validating such a large set would represent a 6000-fold increase, we chose to evaluate the trained model on the metastable variants of two-grain boundaries, one Σ5 and one Σ103e. These evaluations, detailed in the supplementary material, suggest that the energies encountered during training were sufficiently representative for the model to generalize effectively. However, the worst predictions were for GB structures with energies far outside the range of energies on which the dataset was trained. For a more robust model, one could increase the dataset with a small sampling of the metastable structures and ensure the model sees a more diverse set of GB structures and energies.

Feature Selection

Although SOAP achieves the highest accuracy when predicting GB energy, in our implementation SOAP also uses the most features of any of the descriptors (c.f. Table 2). ACE and SF achieve comparable accuracy but only use 121 and 36 terms, respectively. In fact, these two occasionally outperform SOAP. It should be noted that any of these could be adjusted to use more or fewer terms to achieve higher or lower levels of accuracy. One can also use feature selection methods to remove redundant or irrelevant information in the machine-learned structure-property models. Feature selection is an important step towards interpretable machine learning models because of the challenge of interpreting the meaning of high-dimensional descriptors[66].

To identify lower dimensional representations we implement a feature selection method that uses the least absolute shrinkage and selection operator (LASSO) to identify what terms are most important for retaining high accuracy. LASSO is formulated as a minimization of a least squares error plus an L1 norm regularization term scaled by the parameter λ which controls the trade-off between fitting the model accurately to the training data and keeping the model coefficients (parameters) small and sparse. This LASSO model is defined by

$$\min_{\beta}\left\{\frac{1}{N}\parallel y - X\beta\parallel_2^2 + \lambda \parallel \beta\parallel_1\right\} \quad (1)$$

For this feature selection analysis, we use SOAP due to its prominence in GB predictions and SF for its accuracy and interpretability, as detailed in the following section. Figure 5 plots the number of non-zero terms (blue) and the model accuracy (red) of LASSO as the size of λ increases for both SOAP (Fig. 5a) and SF (Fig. 5b). The sparsity of the parameters increases with increased λ values. This model is trained using the average transform and illustrates just how many parameters can be neglected while maintaining a high R^2 value. The 'elbow' of the R^2 curve marks where the model begins to significantly drop in accuracy.

Figure 5 illustrates that at low λ values SF retains all 36 terms and achieves an R^2 value of 0.95. SOAP starts with more terms but can be reduced to 209 terms with a comparable R^2 value. At the high end of λ values, SOAP and SF achieve an R^2 value of 0.43 using 4 and 2 terms, respectively,.

This illustrates that although the accuracy decreases, both of these descriptors can be reduced to a very small representation space while preserving the most important information. It is noteworthy that averaged values of both CSP and CNA achieve higher accuracy predictions of GB energy with a single scalar value with R^2 values of 0.67 for both. Thus, these two singular-valued descriptors are quite expressive and are better than just a few terms of the other descriptors. But, perhaps this is not surprising since these descriptors were designed to easily identify defects and other changes in structure with a single value, while the other descriptors were created to provide a more nuanced description of an environment with a much larger number of terms. Consequently, while feature selection can be helpful in removing redundant and irrelevant information, the descriptors selected have a big impact on identifying important features that contribute to the machine-learned structure-property model.

Interpretability

The goal in this work is to obtain structure-property models that are both accurate and interpretable. Interpretability is of critical interest for the advancement of science since machine learning models could easily become black box models that can't be understood. We have examined the impacts that the descriptor, transform, and machine learning model have on accuracy and methods to select the features that have the biggest impact on the models. We now focus on extracting interpretable information from these models.

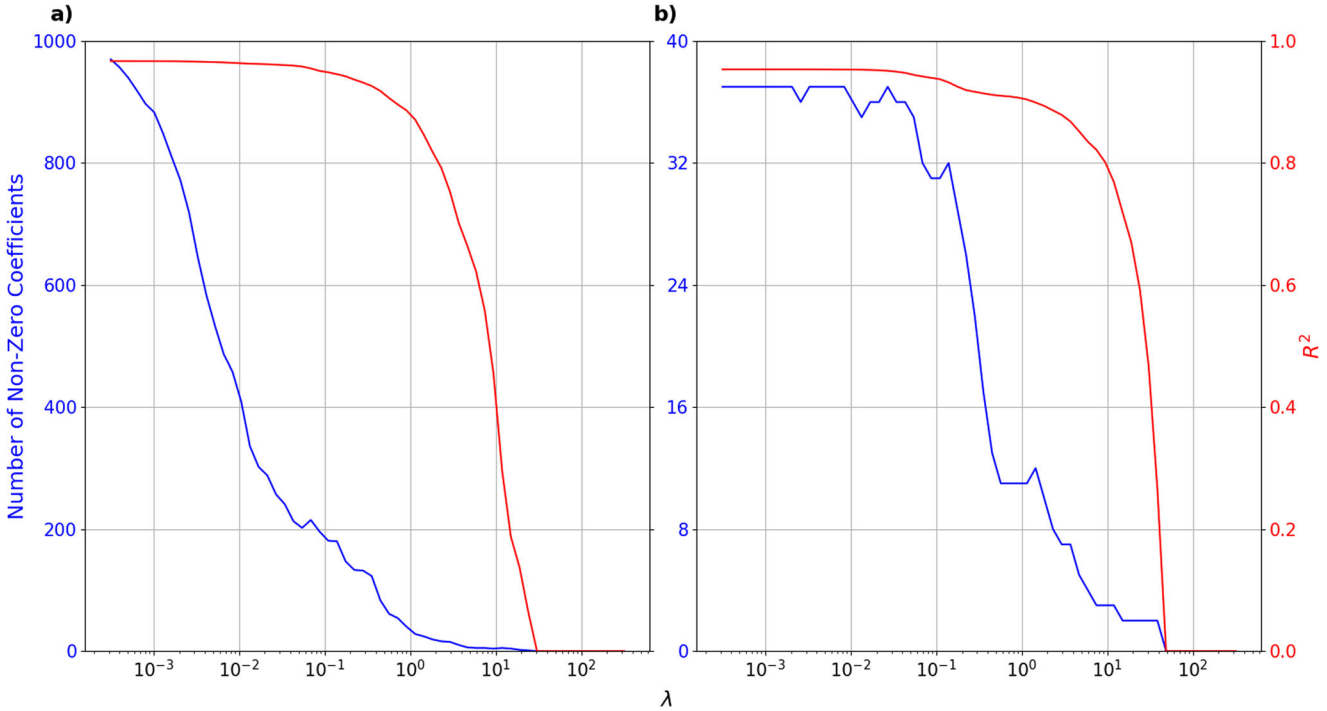

Fig. 5 | Summary of LASSO model results for two different descriptors. The LASSO model is trained on averaged SOAP (**a**) and SF (**b**). The *x*-axis is the λ value that scales the regularization term. The regularization term is the L1 norm of vector of coefficients. Increasing λ pushes the model to send coefficients to zero. The red line shows the R^2 value calculated from the predictions of the model. The blue line represents the number of non-zero coefficients.

Descriptors. ACE, SOAP, SF, ACSF, CSP, and CNA all represent the local environment around each atom. These representations often involve sums over neighboring atoms, where each term in the sum depends on the distance and/or angular relationship between the central atom and its neighbors. ACE, SOAP, and SF all represent the angular relationships in terms of spherical harmonic expansions, while ACSF uses a more explicit *n* body expansion of the angular terms. SOAP utilizes a truncated Gaussian function for the radial distribution and then recasts its basis function in a bispectrum approach. This results in the mixing of angular and radial nodes in the invariant basis functions. ACE utilizes multiple radial functions within a specified cutoff radius to capture the radial environments. In SF, the radial and angular nodes of the expansion are kept separate resulting in descriptors that are analogous to an atomic orbital expansion. SF then represents this information in the minimal set of invariants required to characterise deformations up to the 4th order (i.e., second derivatives or curvature of the strain). ACSF uses multiple Gaussian radial terms which are then convolved with the angular terms. All of the methods carry out their expansions to a level of accuracy that can be defined by the user.

In contrast to the other methods, the graph description (used in the graph2vec transform) characterizes the GB in a periodic, weighted graph. In this framework, the nodes of the graph correspond to the spatial positions of atoms and the edges are weighted by the distance between neighboring atoms within a cutoff distance. Thus, this method captures the complex arrangements of the atoms in a GB as a whole.

All the descriptors transform the 3D spatial coordinates of atoms into a high-dimensional feature space. This transformation is designed to capture complex interactions and symmetries, but it also means that the resulting descriptors are often far removed from the intuitive, three-dimensional space in which atoms actually exist. The use of physical descriptors (such as bond bending terms or spherical harmonics) leaves open the possibility that terms *could* be interpretable. However, the derivation of the terms often renders this difficult, and most approaches do not offer any interpretations of their descriptors. This is not a criticism of these descriptors as they were not designed for the purpose of interpretability.

Many of these descriptors were designed for machine-learned interatomic potentials or for classification of atom neighborhoods. In other words, many of these descriptors were meant for forward modeling only, but to extract interpretable information, we need to go backwards through forward models. Therefore, pointing out weaknesses in backwards applications to extract specific terms is not a completely fair criticism. Nevertheless, because we wish to understand which descriptors provide this backwards path to interpretability we examine them anyway.

In a backwards application, ACE and SOAP are less than ideal because they do not retain knowledge of how their rotationally invariant descriptors are oriented in space. In contrast, the SF approach explicitly retains such terms. Additionally, for ACE and SOAP, one can know which degree of the spherical harmonics are important as illustrated in Supplemental Figure S2, but finer structural detail within that degree is lost. It may be possible to preserve some of that information if interpretability is desired. In a backwards application of ACSF, it is not clear to the authors how one would interpret the values of the 2- and 3-body functions. At a minimum the summation over neighbors makes it difficult to connect with specific atomic structures. However, it is possible that improved application of ACSF descriptors could be useful in extracting the local symmetries and deviations therefrom that could result in interpretability.

Graph2vec's interpretability suffers due to its abstract representation of graph embeddings. In its current implementation, there is no method by which to go backwards and make use of the subgraphs that were extracted, but this could be added for improved interepretability. CSP and CNA are non-unique descriptors where multiple atomic configurations can produce identical values, making it challenging to pinpoint specific structural features responsible for observed properties.

As a result, the inherent complexity of these descriptors makes it challenging to reverse-engineer the key features of the structure-property relationships. Most of these descriptors do not support easy interpretability because they were not designed for this purpose. However, the SF approach is distinct in characterizing the invariants of physical deformations up to the 4th order. This descriptor is examined in additional detail in a case study below.

Transforms. Just like descriptors, the ability to go backwards through transforms is crucial for interpretability. Averaging, the best transform for the most accurate descriptors, can not be applied backwards. One can only make conclusions about the average atomic environments. As explained in the Accuracy subsection, there are assumptions inherent to each of the transforms. Averaging best matches the physics of how GB energy is calculated.

On the other hand, the largest simplex and CUR methods actually preserve specific features from the input matrix. By selecting the most important rows, these techniques effectively fix the size of the matrix that describes the GB. In other words, they choose the atomic environments that, even before training on data, are likely to preserve valuable information based on the assumptions of these transforms listed earlier. This deliberate selection ensures that the matrix representation of each GB remains concise and meaningful. Therefore, both CUR and largest simplex methods identify original atomic environments that can be analyzed for their interpretability and impact on the predictions.

The KMeans clustering transform, while fixing the matrix representation, does not preserve the original rows of the matrix but rather a number of cluster centers. Thus it is like averaging where information is lost in the transform application. In the case of KMeans clustering the original rows can be identified by finding the nearest neighbors to the cluster centers. Although this can be difficult in higher dimensions.

Finally, it is important to note that other properties of interest might have different relationships with the atoms involved. This will impact which transform might work best, as discussed in the Accuracy subsection above.

Machine Learning Models. The ability to extract interpretable information from the machine learning models depends heavily on the model used. When linear models are used, one can readily identify the features that have the greatest impact on the predictions based on values of the coefficients. The relationships are easily defined and understood. One can even reduce the features using a feature selection or regularization approach[66], as discussed above, to more easily identify the important features. However, in non-linear models identifying the most important features is not as simple.

If a KNN performs well, that may be indicative of clustering in the input features, which would suggest that the clustering is relevant to the property of interest. Ensemble methods like AdaBoostRegressor fit the data multiple times, focusing more on difficult cases with additional iterations. While decision trees provide inherent interpretability, ensembles of them make that more difficult. However, these algorithms can export importance scores to learn about which features are of greatest interest.

In our Support Vector Regression (SVR) model, we utilized a linear kernel to fit the data with a hyperplane. The choice of a linear kernel allowed us to maintain a straightforward relationship between the features and the output. This approach ensured that the model remained easily interpretable, as each feature's effect on the output could be independently assessed through the corresponding coefficients. By employing a linear kernel, we avoided the complexities associated with high-dimensional transformations, which are common in non-linear kernel methods.

A neural network model uses a series of layers whose connectedness and construction can be highly variable. In the neural network certain layers have non-linear activation functions to learn non-linear relationships. While this may improve predictions, the overall structure of the neural network makes it difficult to extract interpretable information.

However, in any of these cases, one can use additional tools, such as those that fit into the category of Explainable AI[67], to extract the features of greatest impact or importance.

The preceding discussion of how the method selected for each step in the process of predicting structure-property relationships impacts interpretability has remained theoretical. In the following subsection, we examine a case study where we can be more specific about the ability to extract meaningful information from machine-learned models.

Case Study of Interpretability

Here we examine how a combination of one descriptor, one transform, and features selection in two different machine learning models provide interpretability. We employ the SF descriptor because, as noted above, it was defined with the express purpose of retaining a physical meaning. We employ the average transform because, as noted above, it retains a connection to how our property of interest, GB energy, is calculated. Finally, we examine feature selection in two different machine learning methods to illustrate the differences related to interpretability.

As discussed above, regularized linear models allow easy identification of the most important features in a model. We revisit the results of the LASSO application to the SF illustrated in Fig. 5 and described in the Feature Selection subsection above. As the model complexity is reduced, the terms that remain can be considered the most important for GB energy prediction and interpretability. The last five SF terms to be removed by LASSO, and therefore the top five terms for predicting GB energy, are listed in Table 1. Next to each term in Table 1 is the sign of the correlation of that term in the model with its effect on the GB energy prediction.

To understand the impact of a non-linear model, we employ Extra Trees regression. For interpretability, this is used in conjunction with SHapley Additive exPlanations (SHAP) analysis[68]. SHAP values explain the prediction of an instance by computing the contribution of each feature to the prediction. This method is based on game theory and helps in attributing the prediction output to individual features, offering a fair and consistent way to understand the model behavior.

Figure 6 plots the SHAP analysis of an Extra Trees regression, showcasing the top 10 SF features sorted by their average SHAP value. A SHAP value indicates how much a given feature changes the output of the model compared to the baseline prediction; positive and negative SHAP values correspond to a positive and negative effect on the predicted property, respectively. The colors represent the values of the features for each data point; with red and blue values corresponding to high and low values of a particular feature, respectively. For example, if a dot is blue and located on the right side (positive SHAP value), the low value of that feature increases the value predicted by the model. The top five features from the SHAP analysis are also listed in Table 1, along with the sign of the correlation between the term and its effect on the predicted GB energy in that model.

There is significant alignment between the LASSO and SHAP analysis, as illustrated in Table 1. Four of the top five SF terms are the same in both analyses. Furthermore, the sign on the correlation of these four terms is the same. These four terms are P4I8, P1I0, P2I0, and P4I9, and are accompanied by the P3I4 and O2I0 terms that only appear in one model.

To further confirm the expected correlation of these top terms from the LASSO and SHAP analysis, we plot the average SF values for each GB against both excess volume and energy in Fig. 7. Excess volume is included as it is known to have a positive correlation with GB energy[38]. It is noted that correlation plots for all SF terms are plotted in Supplemental Figures S9-S11. The top 5 features from these supplemental plots with the highest

Table 1 | Comparison of the top five features identified by LASSO and SHAP for the SF model to linear correlations with GB energy

Rank	LASSO	SHAP	Correlation
1	P4I8 −	P4I8 −	P1I0 +
2	P1I0 +	P2I0 +	P3I4 +
3	P2I0 +	P1I0 +	P4I8 −
4	P4I9 −	P3I4 +	P2I2 −
5	O2I0 +	P4I9 −	P3I0 +

Each feature is ranked based on its influence on the model's predictions, where the + or − indicating a positive or negative influence, pushing the model's output higher or lower, respectively. This is accompanied by a column listing the features with the highest correlation with GB energy along with the sign of the correlation.

correlation with GB energy are listed in Table 1 along with the sign of the correlation. It is worth noting that five of the six correlations plotted in Fig. 7 have the same sign as that of the models listed in Table 1. The exception is the O2I0 term, which has an opposite sign in the model but only has a weak correlation with energy.

As noted earlier, SF comes with an added interpretability benefit since each SF descriptor characterizes something unique about the deformation. Each SF term can be classified into one of five categories: density, deformation magnitude, deformation type, and internal and external orientation. Supplemental Figures S9-S11 identify the categories for all 36 terms, which are related to the categories listed in[69].

The P4I8 term is one of three density metrics, and is an r^4 average for all the atoms in the neighborhood of a given atom. This term is orthogonal to the other two density metrics (P2I2 and P0I0). As the GB regions are defined using a finite thickness around the non-FCC atoms, the different density terms will have different relative contributions from the disordered atoms in the GB region and P4I8 should have highest relative contribution due to faster decay. Figure 7 shows that there are reasonable correlations with excess volume and GB energy, which have R^2 values of 0.53 and 0.72, respectively. Note that among the three density terms, P4I8 displays the highest R^2 for correlation with GB energy (see also Supplemental Figure S10).

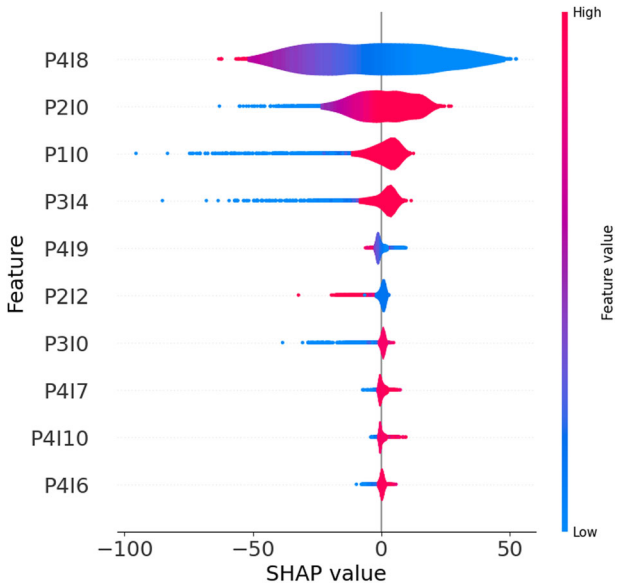

Fig. 6 | SHAP analysis of an Extra Trees regression on SF results. This showcases the first 10 SHAP values sorted by their impact on model output.

The P1I0, P2I0, and P3I4 terms are all categorized as deformation magnitude. P1I0 characterizes the gradient in the density. P2I0 characterizes the net deviatoric strain, akin to the von Mises strain invariant of the neighborhood. There should be substantial deformations of the fcc structures in the lattice neighboring the GB and shearing will be one of the primary means for this. The P3I4 term measures the extensional contribution of the strain gradient. The importance of this term can be attributed to the presence of strain gradients at GBs, especially in the case of GBs that can be represented as dislocation arrays. It is similar to the P1I0 term, where it is a gradient term, but it is weighted by r^3 rather than r^1. As illustrated in Fig. 7, the P1I0 and P3I4 have strong correlations with excess volume and energy, while P2I0's correlations are weak.

The P4I9 metric is an internal orientation metric that defines the orientation between the local lattice and the P4I6 measure of the shear. The latter is similar to the net shear metric P2I0, except that it is weighted by r^4 rather than r^2. Finally, the O2I0 is an external orientation metric that is also highlighted by the LASSO method; this terms defines the orientation of the lattice shearing (measured by P2I0) with respect to the normal of the GB. Thus, it has some similarity to P4I9. Given that aluminum is not isotropic (albeit with a relatively small Zener ratio), it is not surprising that the amount of shear necessary to accommodate the mismatch at a GB will be related to the direction of that shear i.e., the crystal orientation.

Interestingly, the P4I9 and O2I0 terms both have little to no correlation with excess volume and energy on their own. But in conjunction with the other terms in the models, they are deemed more important for the prediction of GB energy than other terms that have strong correlation with energy. For example, the P2I2 term (a density metric based on r^2 weighting) has much stronger correlation with energy than the P4I9 term, but the models are not predicting based on any single term alone, but the combined effect of multiple terms. Thus, some of these terms may provide more of a secondary effect that can distinguish nuanced variations of the GB energy and such effects may not be apparent in 2-D cross-sections examining single-value correlations.

The consistency of the top features and their identifiable correlations with energy in all but one case illustrate that this combination of SF descriptor, average transform, and both linear and non-linear machine learning models is capturing useful trends. The average transform tells about the general trends of the atoms in the GB but not about specific local atomic environments. The linear model provides detailed insight into the influence of each parameter. The non-linear model provides this insight through the SHAP analysis. Most importantly, the SF descriptor connects the features with physical, interpretable attributes of the GB structure-property relationship.

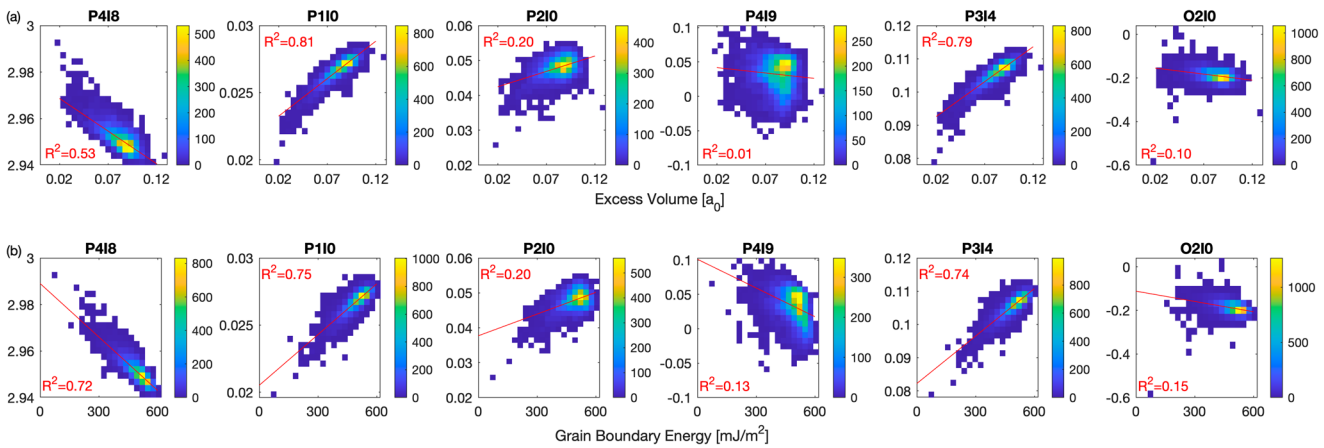

Fig. 7 | Comparison of SF metrics to GB excess volume and energy. 2D histograms of SF terms from Table 1 with **a** excess volume, measured in lattice parameter units a_0, and **b** GB energy.

However, the analysis also uncovers the fact that different models will extract different features. While there are only minor differences in the top five features listed in Table 1, the difference becomes more dramatic as more terms are compared. In addition, other models will extract entirely different features; a Bayesian linear model applied to the data showed correlations with many of the SF metrics listed above, but sometime with the same effect and sometimes with an opposite effect on the GB energy. However, the Bayesian analysis, which is described in the Supplemental Material, required an additional step in describing the data (subtracting the mean and dividing by the standard deviation). The model input is slightly different and leads to different correlations as a result. Care must be taken since the machine learning is only finding correlations in the data that has been through feature engineering, it is not finding the causal relationships for the predictions.

Discussion

Prediction in variable-sized atom-clustered structures consistently require three steps: **describe, transform, machine learning**. Each of these steps, illustrated in Fig. 1, plays a role in the resulting model accuracy and interpretation. Such feature engineering is frequently employed to provide more accurate predictions or better interpretation of the results. In the fast growing environment of machine learning and artificial intelligence models, the diversity in steps taken by different groups makes it challenging to know which methods lead to improved accuracy and interpretability. We have examined this challenge in GBs, which, like all variable-sized atom clusters, require a transform to obtain consistent feature sizes. By attempting to standardize the various steps of the feature engineering process, we have aimed to understand how each step in the process affects the accuracy and interpretability of the resulting model predictions.

Descriptors play an essential role in taking atom structure information, most often represented by Cartesian coordinates, and mapping that to a feature vector that encodes the most critical information. Our findings underscore the robustness of physics-inspired structural representations[62] in capturing the intricate behaviors of GBs. Notably, descriptors such as the SOAP, ACE, and SF demonstrated superior predictive accuracy, underscoring their potential in advancing computational materials science. SF stands out in this group of accurate predictors because it has a low feature count and each feature has a physical meaning in terms of the strain in the neighborhood of each atom. One of the interesting conclusions is that higher order deformations (i.e., strain gradients and higher) should be considered for accurate predictions of GB energy. This is likely the main reason behind higher predictive capability of SOAP and ACE compared to SF, as the current implementation of SF considers only up to 4^{th} order terms, whereas spherical harmonics up to 12^{th} order were considered for both SOAP and ACE. This is also corroborated by LASSO analysis for SOAP descriptors (Supplemental Figures S1 and S2), which shows persistence of $l = 8, 10$ terms.

In short, it appears that better accuracy can always be achieved with additional descriptor information. For example, there are numerous cases where concatenation of one or more descriptor improves the learning[18,28,31]. But, longer feature vectors complicate interpretability. Additionally, some descriptors that can encode a lot of information cannot be processed in reverse to provide physical insight from model predictions. Therefore, those descriptors that encode physics that can be readily extracted from a model prediction are likely to provide the greatest insight into the resulting structure-property models.

The transforms applied to obtain consistent feature sizes from the variable-sized input data impact both the accuracy and interpretability of the resulting model. If per-atom quantities are to be predicted, the transformation step can be eliminated. However, for our goal of predicting overall grain boundary properties, a transformation step remains necessary. As hypothesized in this work, the average transform provides the best accuracy because it is the most similar to the procedure used to calculate GB energy from the atomic structure; the excess energy, relative to the bulk energy, for all the atoms is summed and divided by the area of the boundary. We further hypothesize that other properties of interest may benefit from transforms that capture the important features of that property. For example, properties controlled by extreme values in a distribution may benefit from a transform that captures the structural aspects of those extreme values. However, while some transforms might provide more accurate predictions, they may make interpretation difficult. The average transform is one of these and it can not be processed in reverse to tell us how the distribution of values that were averaged might be critical to a certain structure-property relationship.

The machine-learning models and algorithms also play a significant role in both accuracy and interpretability. As discussed above, simpler models are preferred to complex models and the linear models provide high accuracy in many cases. The linear models are easier to interpret because the contribution of any given feature can be easily discerned. However, non-linear models can make use of Explainable AI tools[67], such as the SHAP analysis discussed above, to extract interpretable meaning from the resulting model. Models can sometimes be overly complex; we suggest opting for simpler linear models whenever feasible.

From an accuracy standpoint, we recognize that we have not considered all combinations of descriptors, transforms, and machine learning models. Neither have we done an exhaustive search of the hyperparameters for each of the descriptors, transforms, and machine learning models beyond a simple gridsearch of hyperparameters for the machine learning models. We cannot guarantee that higher accuracy couldn't be achieved with adjustments to the models we examined. Furthermore, we have predicted grain boundary energy but other properties, such as grain boundary mobility, might result in different architectures and feature engineering to incorporate relevant features and their time-dependence. Thus, each unique property may require a unique approach for optimal predictions. In any case, by providing a systematic approach, and a consistent dataset across all the models evaluated in this work, we provide a framework and benchmark against which new and improved models can be tested, like the MNIST dataset has served for benchmarking machine learning efforts in optical character recognition[70–72]. Also, by providing standard steps and language for the comparison of models along with a deliberate attempt to employ principles of the machine learning community, we hope the grain boundary community can identify the best methods to obtain structure-property relationships that will drive innovation.

Feature selection is a tool that can be used in conjunction with machine learning models to reduce the feature vector to those items that are the most critical for the accuracy in the model. The assumption from an interpretability standpoint is that these selected features are the most important for the model and can therefore be used to obtain insight into the structure-property models.

The final case study illustrated how the SF descriptor provides insight into the density and deformations that correlate with GB energy. Four of the top five features were shared between a linear and non-linear model. This nuanced view, where some features are consistently highlighted across methods while others are unique to specific approaches, offers a richer understanding of the predictive landscape. It suggests that while some attributes of GBs are universally recognized by various predictive models, others may be more method-dependent, possibly due to underlying assumptions or mathematical formulations inherent to each technique. This insight not only enriches our understanding of feature selection dynamics but also guides further investigation into the specific roles these features play in material structure-property relationships.

The implications of these results offer a pathway towards more precise and efficient predictive models that can be instrumental in materials design and engineering. By enhancing our ability to predict GB properties, these findings could facilitate the development of materials with optimized mechanical properties, thereby having a profound impact on various industrial applications.

Methods

In the application of machine learning grain boundaries (GBs), feature engineering[73] is crucial for enhancing model performance by tailoring input data to more accurately reflect the underlying problem and prepare for

predictive modeling. Figure 1 illustrates the three key steps taken for predicting GB properties: employing descriptors to encode or **describe** the structure (step 1), **transforms** to standardize the data size across different GBs (step 2), and **machine learning** or prediction algorithms to predict a GB property from the input data (step 3). The methods and techniques employed in this work for the three key steps are described below.

Descriptors

The descriptors employed in this work are inspired by a phylogenetic tree of structural representations shown in Fig. 8. The tree is reproduced and adapted from a paper titled "Physics-Inspired Structural Representations for Molecules and Materials," by Musil et al.[62]. They emphasized the requirements for developing descriptors that map the atomic positions in Cartesian coordinates to a new metric space commonly called the feature space, the representation[62], descriptor[74], or fingerprint[16,75–80]. This mapping is required for predictive modeling because atom locations represented as Cartesian coordinates cannot uniquely characterise materials[62]. Specifically, atom locations do not preserve symmetries and any translation or rotation of a material will change the Cartesian coordinates. Thus, a descriptor should address requirements of completeness, symmetry, smoothness, and additivity. These requirements and the general process is discussed in additional detail in the Supplemental Materials.

In this work, we considered descriptors from all the branches; additional discussion of the selection process is provided in the Supplemental Materials. In the end, we selected the following methods listed in the tree by Musil et al.[62]: smooth overlap of atomic positions (SOAP), atomic cluster expansion (ACE)[81], and Behler-Parinello's atom-centered symmetry functions (ACSF)[82,83]. These three methods are boxed in green in Fig. 8. We also include several other methods. First, we include a recent Strain Functional (SF) Descriptor, which define the minimal set of invariants required to characterise deformations up to the 4th order (i.e., second derivatives or curvature of the strain)[84]. This method is similar to SOAP in having smoothly declining neighborhood function, compared to the sharper neighborhood cutoffs of ACE, MTP and SNAP. Second, we include two methods commonly used to identify GB atoms: Centrosymmetry Parameter (CSP) and Common Neighbor Analysis (CNA). Finally, we include a periodic graph description of a boundary where every atom in the GB is defined by a node, which is connected by edges to neighboring nodes (atoms) within a certain cutoff distance. Each of these additional methods is listed in Fig. 8 in green under the branch we identified as most relevant. Techniques considered but not selected from other branches are discussed in the Supplemental Materials, with reasons being that in some cases they were not appropriate for characterization of single species large GB structures and in other cases, code was not readily available for implementation.

The SOAP, ACE, and ACSF implementations all required user defined parameters that impact how many terms are used to represent an atom environment. SOAP and ASCF are implemented using the 'DScribe' python library[85], a powerful tool designed for creating descriptors in atomistic systems. For SOAP, all default parameters were used excluding `rcut=3.74`, `lmax=12`, `nmax=12`, and `sigma=.575`. ACSF are unique in the sense that the user defines a list of interactions i.e. pairs, triplets for the various functions used by ACSF. We used 11 G2 term, 8 G3 terms, 8 G4 terms, and 8 G5 terms. The implementation of the ACE algorithm was achieved using the 'ace.jl' package within the Julia programming environment[81,86]. A basis was created using the following parameters: `N=3`, `maxdeg=12`, `r0=2.86`, `rin=0.1`, `rcut=3.25`, and `pin=2`. The SF descriptors are currently 4th order, but can be extended to higher orders. SF used the following parameters: `sigma=1.017837` and `cutoff=5.699887`. The graph description has the nodes of the periodic graph listed as their spatial coordinates. The edges are weighted by the distance to their neighbors that are within `rcut=3.74` Å of the atom (node) of interest. CNA[55] and CSP[52] both represent each atom's environment with a single value, which is calculated during the creation of the dataset using LAMMPS[87]. Each descriptor results in an $N_i \times M$ matrix where N_i is the number of atoms in the ith GB structure and M is the feature length

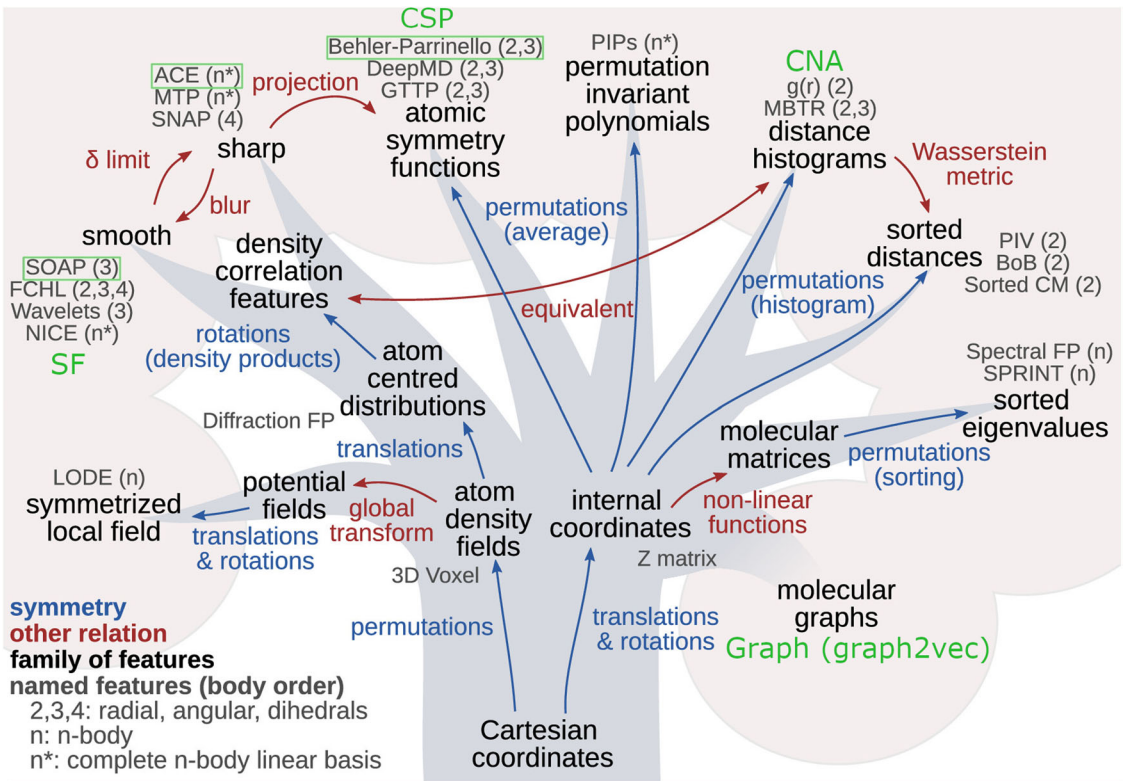

Fig. 8 | Philogenetic tree of structural representations adapted from Musil et al.[62] under the CC-BY-4.0 license. All additions are marked in green to identify the descriptors used in the present work.

Table 2 | Table of output size of various descriptors

Descriptor	M Value
ACE	121
ACSF	37
SOAP	1014
SF	36
Graph	n/a
CSP	1
CNA	1

Table 3 | Table of input and output shapes of various transform methods and the parameter *P* used to create the shapes

Transform	Input	Output	*P* Choice
Average	$N_i \times M$	$1 \times M$	n/a
Largest Simplex	$N_i \times M$	$P \times M$	10
CUR	$N_i \times M$	$P \times M$	20
Kmeans	$N_i \times M$	$P \times M$	100
KDE	$N_i \times 1$	P	100
graph2vec	Graph	$P \times M$	128

of the descriptor. Table 2 shows the feature lengths of each of our descriptors.

Transforms

Since GBs and other variable-sized atom-clustered structures can have variable numbers of features, the feature size must be standardized through some sort of transform. This will transform the $N_i \times M$ feature representation from the descriptor to a $P \times M$ feature representation that is identical for all atomic structures.

In this work we examine 6 possible transforms : average, CUR or skeleton matrices[88], KMeans clustering[89], the largest simplex[16], kernel density estimation (GaussianKDE)[89], and graph2vec[90]. The motivation behind this set of transforms is provided in the Supplemental Materials.

The Average, largest simplex, CUR, and Kmeans transforms are used with almost all descriptors. GaussianKDE is used only with CSP and CNA and graph2vec is used only with the graph descriptor. Table 3 lists the theoretical input and output size of each transform along with the P values used in this work. It is noted that no attempt was made to find the optimal P value for each implementation. Explanations for why each P value was chosen is given in the Supplemental Materials.

Machine Learning

Machine learning is oftentimes yet another mapping to a different feature space. To understand the impact of this step, the research employed a comprehensive and systematic approach to compare a diverse set of machine learning algorithms. This set includes three linear models: LinearRegession, Lasso, and RidgeCV; one support vector machine: SVR; two ensemble methods: Extra Trees and AdaBoost; one nearest neighbor method: KNN; and one neural network (deep learning) model: MLPRegresssion. All of these methods are available for implementation via the sklearn python library[89]. This set of algorithms were selected using the sklearn documentation where various supervised methods are grouped by methodology.

The training and validation process used in this work is illustrated in Fig. 9. In order to train and validate the various machine learning models, the dataset was split into training and validation subsets of 80% and 20% of the GBs, respectively. The exact same subsets of GBs were used in the training and validation of each model to ensure a consistent and fair comparison between models.

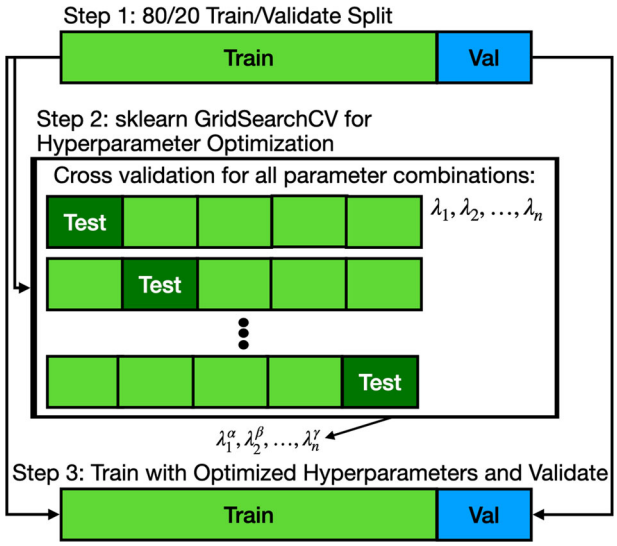

Fig. 9 | Illustration of the steps involved in training each machine learning model. Step 1: 80%/20% split of the data into training and validation sets. Step 2: Use of sklearn GridSearchCV to optimize hyperparameters in the machine learning model with 5-fold cross-validation. Step 3: Training of the machine learning model using the optimized hyperparameters and the 80% training set followed by validation using the 20% validation set.

To optimize each model's performance, we used scikit-learn's GridSearchCV. This tool systematically searches for the best combination of hyperparameters by applying cross-validation on the training set. Hyperparameters, which are model parameters that must be set before the learning process, such as regularization strength or tree depth, can significantly affect a model's outcome. GridSearchCV performs an exhaustive search over a user-defined grid of hyperparameter values, training and validating the model for each possible combination.

The cross-validation splits the training data into several subsets (or folds), trains the model on a subset, and validates it on the remaining fold. This process is repeated across all folds, ensuring the model is tested on all portions of the training data to prevent overfitting. Once GridSearchCV identifies the optimal hyperparameters based on the average cross-validation performance, the model is retrained on the entire training set using these best parameters.

Finally, the trained model is evaluated on the validation set, which remains unseen during the training process, to assess its performance. This pipeline was implemented in an effort to provide a uniform and equitable basis for comparison between models. For performance assessment, both the Mean Absolute Error (MAE) and the Coefficient of Determination (R^2) were recorded. The MAE provided an understanding of the average absolute error made by the models, while the R^2 offered insights into the proportion of variance in the dependent variable that could be explained by the independent variables, acting as another method for measuring the accuracy of the model.

We disclose that beyond the grid search, which required user input, little additional effort was taken for the optimization of the hyperparameter values. Typically convergence was costly from a time perspective for the non-averaging transforms due to the increased complexity, so the range of hyperparameters defined in the grid search was customised according to convergence time and not accuracy. While we recognize that this may limit the accuracy of these predictions, our data consisted of only a train and test set, so this also minimized the chances of overfitting.

Comprehensive documentation of all aspects of the model training process, including hyperparameter values, cross-validation details, and performance metrics, was maintained for the reproducibility of the research. A code base with a sample dataset is included to ensure that the experiments can be replicated and validated by others in the scientific community.

Dataset

To compare the diverse **descriptors, transforms**, and **machine learning** techniques, we employ a recently published atomistic dataset of aluminum GBs[38,63]. This dataset provides a comprehensive sampling of the five degrees of crystallographic character. The datasest includes 7304 pure Aluminum GB structures and their corresponding energies. The construction and process of obtaining the minimum energy structure for each GB is described in detail[38] and the structures are available for download[63]. These GB structures were created with the emperical embedded atom model (EAM) potential created by Mishin et al.[91]. Thus, the physics inherent to these structures is limited to the accuracy of this emperical potential and the methods used to construct the GBs. For purposes of limiting data storage, each GB includes ± 15 Å of atoms relative to the expected location of the GB. Because the size of some of the GBs in the original dataset were too large for some of the descriptor implementations, we use a subset of 7174 GBs, which are those GB structures that contain less than 35,000 atoms. It is expected that this dataset contains sufficient diversity both in crystallographic character and atomic structure to serve as a robust basis for the comparisons and interpretations provided in this work.

Data availability

The datasets generated and/or analysed during the current study are available in the Mendeley Data repository, https://doi.org/10.17632/4ykjz4ngwt[38,63].

Code availability

The underlying code for this study is available at https://github.com/braxtonowens/gbcompare.

Received: 30 July 2024; Accepted: 9 December 2024;
Published online: 26 January 2025

References

1. Palumbo, G., Lehockey, E. M. & Lin, P. Applications for grain boundary engineered materials. *JOM* **50**, 40–43 (1998).
2. Randle, V. Grain boundary engineering: An overview after 25 years. *Mater. Sci. Technol.* **26**, 253–261 (2010).
3. Watanabe, T. Grain boundary engineering: Historical perspective and future prospects. *J. Mater. Sci.* **46**, 4095–4115 (2011).
4. Rohrer, G. S. Grain boundary energy anisotropy: A review. *J. Mater. Sci.* **46**, 5881–5895 (2011).
5. Echeverri Restrepo, S., Tamayo Giraldo, S. & Thijsse, B. J. Using artificial neural networks to predict grain boundary energies. *Comput. Mater. Sci.* **86**, 170–173 (2014).
6. Dang, R. & Yu, W. Standard Deviation Effect of Average Structure Descriptor on Grain Boundary Energy Prediction. *Materials* **16**, 1197 (2023).
7. Noh, W. & Chew, H. B. Dislocation descriptors of low and high angle grain boundaries with convolutional neural networks. *Extrem. Mech. Lett.* **68**, 102138 (2024).
8. Lortaraprasert, C. & Shiomi, J. Robust combined modeling of crystalline and amorphous silicon grain boundary conductance by machine learning. *npj Comput. Mater.* **8**, 219 (2022).
9. Fujii, S., Yokoi, T., Fisher, C. A. J., Moriwake, H. & Yoshiya, M. Quantitative prediction of grain boundary thermal conductivities from local atomic environments. *Nat. Commun.* **11**, 1854 (2020).
10. Priedeman, J. L., Rosenbrock, C. W., Johnson, O. K. & Homer, E. R. Quantifying and connecting atomic and crystallographic grain boundary structure using local environment representation and dimensionality reduction techniques. *Acta Mater.* **161**, 431–443 (2018).
11. Kiyohara, S., Oda, H., Miyata, T. & Mizoguchi, T. Prediction of interface structures and energies via virtual screening. *Sci. Adv.* **2**, e1600746 (2016).
12. Zhang, S. et al. Predicting grain boundary damage by machine learning. *Int. J. Plast.* **150**, 103186 (2022).
13. Zhu, Q., Samanta, A., Li, B., Rudd, R. E. & Frolov, T. Predicting phase behavior of grain boundaries with evolutionary search and machine learning. *Nat. Commun.* **9**, 467 (2018).
14. Yokoi, T., Noda, Y., Nakamura, A. & Matsunaga, K. Neural-network interatomic potential for grain boundary structures and their energetics in silicon. *Phys. Rev. Mater.* **4**, 014605 (2020).
15. Guziewski, M., Montes de Oca Zapiain, D., Dingreville, R. & Coleman, S. P. Microscopic and Macroscopic Characterization of Grain Boundary Energy and Strength in Silicon Carbide via Machine-Learning Techniques. *ACS Appl. Mater. Interfaces* **13**, 3311–3324 (2021).
16. Parsaeifard, B., Tomerini, D., De, D. S. & Goedecker, S. Maximum volume simplex method for automatic selection and classification of atomic environments and environment descriptor compression. *J. Chem. Phys.* **153**, 214104 (2020).
17. Cui, Y. & Chew, H. B. Machine-Learning Prediction of Atomistic Stress along Grain Boundaries. *Acta Mater.* **222**, 117387 (2022).
18. Homer, E. R., Hensley, D. M., Rosenbrock, C. W., Nguyen, A. H. & Hart, G. L. W. Machine-Learning Informed Representations for Grain Boundary Structures. *Front. Mater.* **6**, 168 (2019).
19. Trujillo, D., Guziewski, M. & Coleman, S. Machine learning for predicting properties of silicon carbide grain boundaries. Technical Report AD1087034, US Army Combat Capabilities and Development Command, Army Research Laboratory, Aberdeen Proving Ground, United States (2019).
20. Sharp, T. A. et al. Machine learning determination of atomic dynamics at grain boundaries. *Proc. Natl Acad. Sci.* **115**, 10943–10947 (2018).
21. Chesser, I., Francis, T., De Graef, M. & Holm, E. A. Learning the grain boundary manifold: Tools for visualizing and fitting grain boundary properties. *Acta Mater.* **195**, 209–218 (2020).
22. Wagih, M., Larsen, P. M. & Schuh, C. A. Learning grain boundary segregation energy spectra in polycrystals. *Nat. Commun.* **11**, 6376 (2020).
23. Wagih, M. & Schuh, C. A. Learning Grain-Boundary Segregation: From First Principles to Polycrystals. *Phys. Rev. Lett.* **129**, 046102 (2022).
24. Han, J., Vitek, V. & Srolovitz, D. J. Grain-boundary metastability and its statistical properties. *Acta Mater.* **104**, 259–273 (2016).
25. Zheng, H. et al. Grain boundary properties of elemental metals. *Acta Mater.* **186**, 40–49 (2020).
26. Ratanaphan, S. et al. Grain boundary energies in body-centered cubic metals. *Acta Mater.* **88**, 346–354 (2015).
27. Hu, C., Zuo, Y., Chen, C., Ping Ong, S. & Luo, J. Genetic algorithm-guided deep learning of grain boundary diagrams: Addressing the challenge of five degrees of freedom. *Mater. Today* **38**, 49–57 (2020).
28. Tamura, T. et al. Fast and scalable prediction of local energy at grain boundaries: Machine-learning based modeling of first-principles calculations. *Model. Simul. Mater. Sci. Eng.* **25**, 075003 (2017).
29. Gomberg, J. A., Medford, A. J. & Kalidindi, S. R. Extracting knowledge from molecular mechanics simulations of grain boundaries using machine learning. *Acta Mater.* **133**, 100–108 (2017).
30. Rosenbrock, C. W., Homer, E. R., Csányi, G. & Hart, G. L. W. Discovering the building blocks of atomic systems using machine learning: Application to grain boundaries. *npj Comput. Mater.* **3**, 29 (2017).
31. Montes de Oca Zapiain, D., Guziewski, M., Coleman, S. P. & Dingreville, R. Characterizing the Tensile Strength of Metastable Grain Boundaries in Silicon Carbide Using Machine Learning. *J. Phys. Chem. C.* **124**, 24809–24821 (2020).
32. Song, X. & Deng, C. Atomic energy in grain boundaries studied by machine learning. *Phys. Rev. Mater.* **6**, 043601 (2022).
33. Wu, X. et al. Application of Machine Learning to Predict Grain Boundary Embrittlement in Metals by Combining Bonding-Breaking and Atomic Size Effects. *Materials* **13**, 179 (2020).
34. Nishiyama, T., Seko, A. & Tanaka, I. Application of machine learning potentials to predict grain boundary properties in fcc elemental metals. *Phys. Rev. Mater.* **4**, 123607 (2020).

35. Ye, W., Zheng, H., Chen, C. & Ong, S. P. A Universal Machine Learning Model for Elemental Grain Boundary Energies. *Scr. Mater.* **218**, 114803 (2022).
36. Snow, B. D., Doty, D. D. & Johnson, O. K. A Simple Approach to Atomic Structure Characterization for Machine Learning of Grain Boundary Structure-Property Models. *Front. Mater.* **6**, 120 (2019).
37. Huber, L., Hadian, R., Grabowski, B. & Neugebauer, J. A machine learning approach to model solute grain boundary segregation. *npj Comput. Mater.* **4**, 64 (2018).
38. Homer, E. R. et al. Examination of computed aluminum grain boundary structures and energies that span the 5D space of crystallographic character. *Acta Mater.* **234**, 118006 (2022).
39. Hart, E. W. Grain Boundary Phase Transformations. In Hu, H. (ed.) *The Nature and Behavior of Grain Boundaries: A Symposium Held at the TMS-AIME Fall Meeting in Detroit, Michigan, October 18-19, 1971*, 155–170 (Springer US, New York, 1972).
40. Frolov, T., Olmsted, D. L., Asta, M. & Mishin, Y. Structural phase transformations in metallic grain boundaries. *Nat. Commun.* **4**, 1899 (2013).
41. Hickman, J. & Mishin, Y. Extra variable in grain boundary description. *Phys. Rev. Mater.* **1**, 010601 (2017).
42. Meiners, T., Frolov, T., Rudd, R. E., Dehm, G. & Liebscher, C. H. Observations of grain-boundary phase transformations in an elemental metal. *Nature* **579**, 375–378 (2020).
43. Orme, A. D. et al. Insights into twinning in Mg AZ31: A combined EBSD and machine learning study. *Comput. Mater. Sci.* **124**, 353–363 (2016).
44. Francis, T., Chesser, I., Singh, S., Holm, E. A. & De Graef, M. A geodesic octonion metric for grain boundaries. *Acta Mater.* **166**, 135–147 (2019).
45. Zhou, T., Jog, A. & Gall, D. First-principles prediction of electron grain boundary scattering in fcc metals. *Appl. Phys. Lett.* **120**, 241603 (2022).
46. Baird, S. G., Homer, E. R., Fullwood, D. T. & Johnson, O. K. Five degree-of-freedom property interpolation of arbitrary grain boundaries via Voronoi fundamental zone framework. *Comput. Mater. Sci.* **200**, 110756 (2021).
47. Sutton, A. P., Vitek, V. & Christian, J. W. On the structure of tilt grain boundaries in cubic metals I. Symmetrical tilt boundaries. *Philos. Trans. R. Soc. Lond. Ser. A* **309**, 1–36 (1983).
48. Sutton, A. P., Vitek, V. & Christian, J. W. On the structure of tilt grain boundaries in cubic metals II. Asymmetrical tilt boundaries. *Philos. Trans. R. Soc. Lond. Ser. A* **309**, 37–54 (1983).
49. Sutton, A. P., Vitek, V. & Christian, J. W. On the structure of tilt grain boundaries in cubic metals. III. Generalizations of the structural study and implications for the properties of grain boundaries. *Philos. Trans. R. Soc. Lond. Ser. A* **309**, 55–68 (1983).
50. Bishop, G. H. & Chalmers, B. A coincidence — Ledge — Dislocation description of grain boundaries. *Scr. Metall.* **2**, 133–139 (1968).
51. Han, J., Vitek, V. & Srolovitz, D. J. The grain-boundary structural unit model redux. *Acta Mater.* **133**, 186–199 (2017).
52. Kelchner, C. L., Plimpton, S. J. & Hamilton, J. C. Dislocation nucleation and defect structure during surface indentation. *Phys. Rev. B* **58**, 11085–11088 (1998).
53. Żydek, A., Wermiński, M. & Trybula, M. E. Description of grain boundary structure and topology in nanocrystalline aluminum using Voronoi analysis and order parameter. *Comput. Mater. Sci.* **197**, 110660 (2021).
54. Elliott, J. R. & Lira, C. T. *Introductory Chemical Engineering Thermodynamics* 2nd ed. (Prentice Hall, Upper Saddle River, 2012).
55. Honeycutt, J. D. & Andersen, H. C. Molecular dynamics study of melting and freezing of small Lennard-Jones clusters. *J. Phys. Chem.* **91**, 4950–4963 (1987).
56. Banadaki, A. D. & Patala, S. A three-dimensional polyhedral unit model for grain boundary structure in fcc metals. *npj Comput. Mater.* **3**, 13 (2017).
57. Lejček, P., Hofmann, S., Všianská, M. & Šob, M. Entropy matters in grain boundary segregation. *Acta Mater.* **206**, 116597 (2021).
58. Tsuzuki, H., Branicio, P. S. & Rino, J. P. Structural characterization of deformed crystals by analysis of common atomic neighborhood. *Comput. Phys. Commun.* **177**, 518–523 (2007).
59. Zhou, X., Marchand, D., McDowell, D. L., Zhu, T. & Song, J. Chemomechanical Origin of Hydrogen Trapping at Grain Boundaries in fcc Metals. *Phys. Rev. Lett.* **116**, 075502 (2016).
60. Billinge, S. & Kanatzidis, M. Beyond crystallography: The study of disorder, nanocrystallinity and crystallographically challenged materials with pair distribution functions. *Chem. Commun.* **4**, 749–760 (2004).
61. Kalinin, S. V., Sumpter, B. G. & Archibald, R. K. Big–deep–smart data in imaging for guiding materials design. *Nat. Mater.* **14**, 973–980 (2015).
62. Musil, F. et al. Physics-Inspired Structural Representations for Molecules and Materials. *Chem. Rev.* **121**, 9759–9815 (2021).
63. Homer, E. R., Hart, G. L. W., Hensley, D., Owens, C. B. & Serafin, L. H. Computed al grain boundary structures and energies covering 5d space. Mendeley Data V1, https://doi.org/10.17632/4ykjz4ngwt (2022).
64. Michael, S., Vipin, Ertöz, L. & Kumar, V. New Directions in Statistical Physics. In The Challenges of Clustering High Dimensional Data (ed. Wille, L.T.) 273-309 (Springer, Berlin, Heidelberg, 2004).
65. Poltavsky, I. et al. Crash Testing Machine Learning Force Fields for Molecules, Materials, and Interfaces: Molecular Dynamics in the TEA Challenge 2023. *ChemRxiv* (2024).
66. Wright, J. & Ma, Y.*High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications* (Cambridge University Press, Cambridge, 2022).
67. Longo, L. et al. Explainable Artificial Intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Inf. Fusion* **106**, 102301 (2024).
68. Lundberg, S. M. & Lee, S.-I. A unified approach to interpreting model predictions. In Guyon, I. et al. (eds.) *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing, Long Beach CA*, 4765–4774 (Curran Associates Inc., Red Hook, 2017).
69. Mishra, A., Suresh, S. A., Fensin, S. J., Mathew, N. & Kober, E. M. Learning from metastable symmetric-tilt grain boundaries using physics-based descriptors. *Phys. Rev. Mat.* **8**, 123605 (2024).
70. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998).
71. Deng, L. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Process. Mag.* **29**, 141–142 (2012).
72. Bengio, Y., Courville, A. & Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1798–1828 (2013).
73. Hastie, T., Tibshirani, R. & Friedman, J. H.*The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, New York, 2009).
74. Bartók, A. P., Kondor, R. & Csányi, G. On representing chemical environments. *Phys. Rev. B* **87**, 184115 (2013).
75. Kearnes, S., McCloskey, K., Berndl, M., Pande, V. & Riley, P. Molecular Graph Convolutions: Moving Beyond Fingerprints. *J. Computer-Aided Mol. Des.* **30**, 595–608 (2016).
76. Sadeghi, A. et al. Metrics for measuring distances in configuration spaces. *J. Chem. Phys.* **139**, 184118 (2013).
77. Parsaeifard, B. & Goedecker, S. Manifolds of quasi-constant SOAP and ACSF fingerprints and the resulting failure to machine learn four-body interactions. *J. Chem. Phys.* **156**, 034302 (2022).
78. Batra, R. et al. General Atomic Neighborhood Fingerprint for Machine Learning-Based Methods. *J. Phys. Chem. C.* **123**, 15859–15866 (2019).
79. Piaggi, P. M. & Parrinello, M. Entropy based fingerprint for local crystalline order. *J. Chem. Phys.* **147**, 114112 (2017).

80. Zhu, L. et al. A fingerprint based metric for measuring similarities of crystalline structures. *J. Chem. Phys.* **144**, 034203 (2016).
81. Drautz, R. Atomic cluster expansion for accurate and transferable interatomic potentials. *Phys. Rev. B* **99**, 014104 (2019).
82. Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **134**, 074106 (2011).
83. Behler, J. & Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).
84. Kober, E. M., Tavenner, J. P., Adams, C. M. & Mathew, N. Strain Functionals: A Complete and Symmetry-adapted Set of Descriptors to Characterize Atomistic Configurations 2402.04191 (2024).
85. Himanen, L. et al. DScribe: Library of descriptors for machine learning in materials science. *Computer Phys. Commun.* **247**, 106949 (2020).
86. Dusson, G. et al. Atomic Cluster Expansion: Completeness, Efficiency and Stability. *J. Comp. Phys*. **454**, 110946 (2022).
87. Thompson, A. P. et al. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comp. Phys. Comm.* **271**, 108171 (2022).
88. Virtanen, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **17**, 261–272 (2020).
89. Pedregosa, F. et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
90. Rozemberczki, B., Kiss, O. & Sarkar, R. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*, 3125-3132 (ACM, 2020).
91. Mishin, Y., Farkas, D., Mehl, M. J. & Papaconstantopoulos, D. A. Interatomic potentials for monoatomic metals from experimental data and ab initio calculations. *Phys. Rev. B* **59**, 3393–3407 (1999).

Acknowledgements

This work was primarily supported by the U.S. National Science Foundation (NSF) under Award #DMR-1817321. EMK and NM were supported by internal LANL LDRD funding (XX9A, XXG0).

Author contributions

C. Braxton Owens: Conceptualization, Methodology, Software, Formal analysis, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization; Gus L. W. Hart, Eric R. Homer - Conceptualization, Methodology, Formal analysis, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision, Funding acquisition; Tyce W. Olaveson: Conceptualization, Software, Formal analysis; Nithin Mathew, Jacob P. Tavenner, Edward M. Kober, Garritt J. Tucker - all performed the following related to the Strain Functional analysis: Formal analysis, Writing - Review & Editing.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41524-024-01509-x.

Correspondence and requests for materials should be addressed to Gus L. W. Hart or Eric R. Homer.