Machine Learning in Material Property Prediction

By
Lane E. Schultz

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
(Materials Science and Engineering)

at the
UNIVERSITY OF WISCONSIN-MADISON
2024

Date of final oral examination: 08/19/2024

The dissertation is approved by the following members of the Final Oral Committee:

Dane Morgan, Principal Investigator Professor, Materials Science and Engineering
Izabela Szlufarska, Professor, Materials Science and Engineering
John H. Perepezko, Professor, Materials Science and Engineering
Paul M. Voyles, Professor, Materials Science and Engineering
Mark D. Ediger, Professor, Chemistry

Thesis Abstract

This thesis explores the application of machine learning techniques to predict properties of metallic glasses and assesses the applicability domain of machine learning models. The work is primarily based on three papers focusing on predicting glass forming ability through experimental data, molecular dynamics simulations, and other computational methods. A fourth paper proposes a general approach for determining the applicability domain of machine learning models.

Chapter 2 introduces computational and machine learning approaches to predict the glass forming ability of metallic alloys. The first study investigates the use of characteristic temperatures to predict critical casting diameters using various machine learning models. The second study employs molecular dynamics to calculate characteristic temperatures for metal alloys, which were then used as features in glass forming ability models. The third study explores the prediction of critical cooling rates using a combination of elemental properties and simulated features. Simulated features utilize machine-learned interatomic potentials. This work demonstrates the potential of combining machine learning with physics-based simulations for accelerating the discovery of metallic glasses.

Chapter 3 focuses on developing a general approach for assessing the applicability domain of machine learning models. The proposed method uses kernel density estimation to measure training data density at feature values for inference points. Points from low-density regions have large dissimilarities (i.e., the features come from differing spaces), while those from high-density regions have small dissimilarities. High dissimilarity measures are associated with poor model performance and unreliable uncertainty estimation.

In summary, this thesis makes two significant contributions: first, it enhances the understanding and creation of metallic glass forming ability models through computational techniques; second, it addresses the challenge of determining the applicability domains of models. The findings highlight the potential of machine learning in guiding the discovery of new metallic glasses while emphasizing the need for careful consideration of model limitations.

Acknowledgments

Throughout my career as an engineer, I have been influenced by exemplary individuals from diverse backgrounds, each with their own fascinating stories to share. Their knowledge and support have been invaluable to me over the years, and they deserve special recognition.

Prof. Dane Morgan has been an exceptional guide throughout my graduate career. I am consistently amazed by the depth of his knowledge across a wide range of topics, even those beyond the scope of materials science. He has always encouraged those working under his guidance to develop and rigorously defend their own ideas. Dane has respected personal freedoms and promoted a healthy work-life balance. I am especially grateful to him for allowing me to be with my family during my father's problems with health. I would also like to express my gratitude to other professors who have provided insightful feedback on my research: Prof. Izabela Szlufarska, Prof. Paul Voyles, Prof. Mark Ediger, and Prof. John Perepezko.

I have greatly enjoyed my time with many of my colleagues in the computational materials group (CMG). Dr. Ajay Annamareddy was the first person I became close with, and he was always there to help when I had questions. Having someone to lean on is crucial for someone new to a program. We eventually formed a social group with Dr. Waqas Qureshi, Chiyoung Kim, Dr. Siamak Attarian, and Dr. Sarif Sekh. It was a pleasure to explore the city of Madison, spend time together, and learn from a diverse set of cultures (e.i., Pakistani, South Korean, Iranian, and Indian). I can already feel the nostalgia setting in. A strong appreciation for Dr. Maciej Piotr Polak was also developed. He was the first and only person to impress me with his computer skills, and I aspire to become his peer one day. I learned a great deal from our collaboration in constructing and administering scientific-purpose clusters. The homemade food that Maciej and his wife, Anna, brought me one Thanksgiving still occupies my thoughts. Finally, Dr. Ryan Jacobs was an interesting friend to have in the office. We were often the only ones in the office during the early mornings. Our morning discussions and walks during lunchtime

will be missed.

I firmly believe that I would have never attended graduate school if it were not for both Dr. William Nollet and Kelly Burton. Dr. Nollet was always enthusiastic about lecturing, and I was consistently entertained by his energetic mannerisms. He exposed me to opportunities in undergraduate school that I would have otherwise never known about. He was the one who connected me with Kelly Burton, the soon-to-be-retired manager of the Graduate Engineering Research Scholars (GERS) program. Kelly was always open to guiding anyone in GERS and connecting them to other opportunities. But to be honest, my favorite thing about Kelly was her organizing of social events during the summer. Those were fun times when I made other close friends like Carlos Andrade, with whom I enjoy occasional phone calls and had the pleasure of attending his wedding in Puerto Rico.

My longest-lasting friend is Dr. James Schneider. We have been through tough times together ever since undergraduate school. I would have never guessed at the time that we would both pursue PhDs at the same university. You were the person who allowed me to crawl out of my shell by pestering me to hang out. We have had many adventures, and I think of you constantly. During the quarantine of COVID-19, you were my only tether to the outside world. I would have been much lonelier without you. I learned a lot from you and believe I would have never learned to enjoy life as much as I do now without you. Thank you, my honorary brother, James.

I would like to share a special note for my parents. They are quite literally the reason I am able to compose this document. Both Robert Daniel Schultz and Maria Angela Schultz Altamirano have supported all of my endeavors ever since I was small. My father is from a small town in Ohio and was adopted by his grandparents. He always had fun stories about family, growing up, and later becoming a marine biologist. Living life vicariously through his stories was always entertaining. My mother hails from another small town, but in Colombia. She had a tough upbringing but persevered and taught high school in both Colombia and the United States. Her true passion was art, and I hope she can find

pleasure in her projects during her retirement. I love both of my parents and hope to enjoy their presence for many years to come.

Of course, none of my research would be possible without funding. I am grateful for the Bridge to the Doctorate: Wisconsin Louis Stokes Alliance for Minority Participation National Science Foundation (NSF) award number HRD-1612530, the University of Wisconsin–Madison GERS fellowship program, and the PPG Coating Innovation Center for financial support. Support from the NSF Collaborative Research: Framework: Machine Learning Materials Innovation Infrastructure award number 1931306 and the NSF Designing Materials to Revolutionize and Engineer our Future (DMREF) program, Division of Materials Research (DMR), METAL & METALLIC NANOSTRUCTUREs, award number #1728933 were also appreciated. Many machine learning calculations for research were performed with the computational resources provided by XSEDE 2.0: Integrating, Enabling and Enhancing National Cyberinfrastructure with Expanding Community Involvement Grant ACI-1548562.

List of Publications

1. Published:
   (a) L. E. Schultz *et al.*, "Molecular dynamic characteristic temperatures for predicting metallic glass forming ability," *Computational Materials Science*, vol. 201, p. 110 877, 2022, ISSN: 0927-0256. DOI: https://doi.org/10.1016/j.commatsci.2021.110877. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927025621005899
   (b) L. E. Schultz *et al.*, "Exploration of characteristic temperature contributions to metallic glass forming ability," *Computational Materials Science*, vol. 196, p. 110 494, 2021, ISSN: 0927-0256. DOI: https://doi.org/10.1016/j.commatsci.2021.110494. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927025621002196
   (c) B. T. Afflerbach *et al.*, "Machine Learning Prediction of the Critical Cooling

Rate for Metallic Glasses from Expanded Datasets and Elemental Features," *Chemistry of Materials*, acs.chemmater.1c03542, Mar. 2022, ISSN: 0897-4756. DOI: 10.1021/acs.chemmater.1c03542. [Online]. Available: https://pubs.acs.org/doi/10.1021/acs.chemmater.1c03542

(d) B. T. Afflerbach *et al.*, "Molecular simulation-derived features for machine learning predictions of metal glass forming ability," *Computational Materials Science*, vol. 199, Nov. 2021, ISSN: 09270256. DOI: 10.1016/j.commatsci.2021.110728

(e) J. Xi *et al.*, "Microalloying effect in ternary al-sm-x (x=ag, au, cu) metallic glasses studied by ab initio molecular dynamics," *Computational Materials Science*, vol. 185, p. 109 958, 2020, ISSN: 0927-0256. DOI: https://doi.org/10.1016/j.commatsci.2020.109958. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927025620304493

(f) K. Schmidt *et al.*, "Foundry-ml - software and services to simplify access to machine learning datasets in materials science," *Journal of Open Source Software*, vol. 9, no. 93, p. 5467, 2024. DOI: 10.21105/joss.05467. [Online]. Available: https://doi.org/10.21105/joss.05467

2. Submitted:

(a) L. E. Schultz *et al.*, "A general approach for determining applicability domain of machine learning models," 2024. arXiv: 2406.05143

(b) V. Agrawal *et al.*, "Accelerating ensemble error bar prediction with single models fits," 2024. arXiv: 2404.09896

(c) J. Meng *et al.*, "Ultra-fast oxygen conduction in sillén oxychlorides," 2024. arXiv: 2406.07723

(d) R. Jacobs *et al.*, *Machine learning materials properties with accurate predictions, uncertainty estimates, domain guidance, and persistent online accessibility*, 2024. arXiv: 2406.15650 [cond-mat.mtrl-sci]. [Online]. Available: https://arxiv.org/abs/2406.15650

3. In Preparation:

(a) L. E. Schultz *et al.*, "Machine learning metallic glass critical cooling rates through elemental and molecular simulation based featurization,"

(b) S. Huang *et al.*, "Composition-resolved dynamics in metallic supercooled liquids from momentum-resolved electron correlation microscopy,"

List of Presentations

- "Molecular dynamic characteristic temperatures for predicting metallic glass forming ability," Materials Science & Technology, Columbus, OH, 2021
- "Molecular dynamics features for predicting metallic glass critical casting thickness," Virtual Materials Research Society Spring/Fall Meeting & Exhibit, Virtual, 2020

Contents

List of Figures

List of Tables

Chapter 1

Methods

The works mentioned in this thesis employ a range of computational techniques, including classical molecular dynamics, ab initio molecular dynamics, and machine learning. Due to the complexity of each of these computational methods, a brief introduction and explanation of their use is provided.

1.1 Material Modeling

Methods for modeling materials, from smaller to larger scales, include: ab initio methods at the electronic scale, molecular dynamics at the atomic scale, phase-field modeling at the mesoscale, finite element analysis at the macroscale, etc. For the work in this thesis, both ab initio and molecular dynamics methods were employed (Chapter 2). A brief explanation of these methods is provided in this section.

1.1.1 Ab Initio

To model atoms, it is essential to understand the behavior of both nuclei and electrons. First, modeling electrons can be done with the time-independent form of the Schrödinger equation [15]. In a stationary state, a system can be described by

$$E\Psi = H\Psi = \left[\frac{\hbar^2}{2m}\sum_{i=1}^{N}\nabla_i^2 + \sum_{i=1}^{N}V(\vec{r}_i) + \sum_{i=1}^{N}\sum_{j<i}U(\vec{r}_i,\vec{r}_j)\right]\Psi \tag{1.1}$$

where $\hbar$ is the reduced Planck constant, m is the electron mass, H is the Hamiltonian, Ψ represents the eigenstates of H, E is the adiabatic potential energy surface of the atoms, $\vec{r}$ is the coordinate of an electron, $V(\vec{r}_i)$ describes the energy between an electron and a set of nuclei, $U(\vec{r}_i,\vec{r}_j)$ describes the energy between two electrons, and the gradient operator ∇i is used to calculate the kinetic energy of electrons [15]. Note that the Born-Oppenheimer approximation, which relies on the significantly greater mass of nuclei compared to electrons, allows for the omission of nuclei-nuclei interactions and nuclei velocities in the model. This approximation assumes that the response of electrons to changes in nuclei positions is much greater than the response of nuclei to changes in electrons. When modeling static systems, Eq. (1.1) must be solved. More complex details regarding the assumptions and techniques employed to solve Eq. (1.1) are necessary but extend beyond the scope of this section's description. The subsequent section explains how to propagate the nuclei through time using molecular dynamics.

1.1.2 Molecular Dynamics

Classical physics can be used to describe the motion of atoms. To commence a simulation of moving atoms, the atoms are assigned initial velocities that correspond to a temperature of interest. Then, equations of motion have to be solved. The force interactions between atoms are calculated using a potential function. This potential can come in many forms, including some that are tuned from experimental data, like the embedded atom method, or those built from machine learning on ab initio data, such as the moment tensor potential. For large simulation cells, calculating the force interaction between every pair of atoms can be computationally expensive. Often, only the atoms within a cutoff radius of the atom in question are used for calculating pairwise forces [16]. Integrating classical equations of motion provides both the positions and velocities of atoms based on the force calculations. The Verlet algorithm provides equations for updating positions

and velocities, but others exist [17].

$$\vec{r}_i(t + \Delta t) = 2\vec{r}_i(t) - \vec{r}_i(t - \Delta t) + \vec{a}_i(t)\Delta t^2 \tag{1.2}$$

$$\vec{v}_i(t) = \frac{\vec{r}_i(t + \Delta t) - \vec{r}_i(t - \Delta t)}{2\Delta t} \tag{1.3}$$

In Eqs. 1.2 and 1.3, i represents the atom in question, $\vec{r}$ is the position, t is time, Δt is the time step, $\vec{a}$ is the acceleration (derived from forces), and $\vec{v}$ is the velocity. Δt is a tunable parameter that needs to be adjusted based on two factors. First, it must be sufficiently small to accurately capture the properties of interest. Second, it must be large enough to probe longer time scales efficiently. Many other aspects are involved in simulating the movement of atoms, including periodic boundary conditions, atomic bond modeling, thermodynamic ensembles, etc. However, the provided descriptions offer a high-level overview of how atoms are moved in a simulation.

1.2 Machine Learning

Machine learning encompasses various branches, each with its own set of techniques and applications. Generally, models can be categorized as either unsupervised or supervised. Unsupervised techniques exploit the underlying relations in data without focusing on a specific value of interest or a target to predict. These methods aim to discover patterns, structures, or groupings within the data itself. On the other hand, supervised models learn from accessible data, known as features, to predict values of interest that are more difficult to acquire, referred to as the target. The values for both features and the target can be either discrete or continuous, and specific methods exist to address both types of data. However, the majority of the techniques employed in this work are supervised and deal with continuous values, so the focus will be placed on these particular approaches. By leveraging the relationships between the features and the target, supervised learning algorithms can learn from labeled examples and make predictions or decisions based on

new, unseen data points.

1.2.1 Basic Notation and Goal of Supervised Learning

Data are typically stored in the form of either vectors or matrices. The features or properties measured are commonly denoted as X. The matrix X is an $n \times m$ matrix, where n represents the number of observations and m denotes the number of properties or features. The terms data point, observation, and case are used interchangeably to refer to a single instance in a data set. In supervised learning, the target property for a single observation is denoted as y, and the set of all y values under consideration is represented by $\vec{y}$. Each row of the matrix X, denoted $\vec{x}$, corresponds to a single entry in the vector $\vec{y}$, establishing a one-to-one relationship between the input features and the target variable. The primary objective of supervised learning is to predict the target variable y, denoted as $\hat{y}$, in such a way that the difference between y and $\hat{y}$ (called $Error$ here) is minimized (Eq. 1.4).

$$\min_{\theta \in \Theta} \{Error\} \tag{1.4}$$

In Eq. 1.4, a machine learning model learns the fitting parameters, θ, from all possible parameters, Θ, by minimizing the $Error$. The functional form of $Error$ is not unique and depends on the task of a model (i.e., regression versus classification or sensitivity to outliers). The remainder of this section covers some of the different types of models used supervised regression with the aim of computing Eq. 1.4.

1.2.2 Linear Regression

One of the simplest and most important methods for fitting data is linear regression. For each column in the matrix X, the linear regression method assigns a weight such that $\hat{y}$ obtained from Eq. 1.5 produce an estimate of the dependent variable y with the smallest possible error [18].

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_m x_m \tag{1.5}$$

In Eq. 1.5, each feature, denoted as $\{x_i | i \in 1, 2, \ldots, m\}$, is multiplied by a corresponding weight, represented by $\{\theta_i | i \in 1, 2, \ldots, m\}$, to yield a prediction. The weight θ_0 serves as a bias term, which allows for a non-zero intercept in the linear relationship between the features and the dependent variable. The model learns by selecting weights via Eq. 1.4 for a set of data (i.e., more than one observation). For example, one could use the Residual Sum of Squares (RSS) in Eq. 1.6 for *Error* and find the corresponding set of weights that minimize RSS.

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{1.6}$$

A desirable property of the linear regression model is that the magnitude of the weights indicates the significance of each feature in the model's prediction. This assumes that the features are scaled to be within similar ranges, which can be achieved through pre-processing data.

1.2.3 Least Absolute Shrinkage and Selection Operator

Least Absolute Shrinkage and Selection Operator (LASSO) is a linear model that shares similarities with linear regression. However, LASSO includes an additional term that penalizes the model's complexity, effectively reducing the risk of overfitting. The additional penalty, or regularization term, increases with the number of fitting weights for the model. The regularization term can also be tuned by multiplying it by a regularization parameter, λ, as seen in Eq. 1.7 [18]. When $\lambda = 0$, LASSO becomes linear regression. After the model parameters are found, Eq. 1.6 can then be used for predictions.

$$\min_{\theta \in \Theta} \left\{ RSS + \lambda \sum_{j=1}^{m} |\theta_j| \right\} \tag{1.7}$$

1.2.4 Support Vector Regression

Instead of fitting a single line to data like linear regression, Support Vector Regression (SVR) develops a "tube" within which many points lie. The center of the tube represents the model's predictions. Only points on or outside the tube (i.e., the support vectors) of width ε are considered for calculation of its corresponding minimization function and is called the ε-insensitive loss function [19]. ε is a tune-able parameter. Many more details surrounding the tolerance for the number of points that can be outside the SVR tube (i.e., slack variables), the kernel trick, the type of kernels, and other aspects are included in SVM but are beyond the scope of this work. Refer to Ref. [19] for more information. A key takeaway from SVR is that it tolerates outliers better than ordinary linear regression and can learn non-linear relationships through the use of the kernel trick.

1.2.5 Decision Tree

Decision trees are a type of non-linear model based on the partition of features into J hyper-rectangles. A tree is built using recursive binary splitting. First, a feature is selected, and the training data are split into two sets: $\{X|X_j < \theta_j\}$ and $\{X|X_j \geq \theta_j\}$, where θ_j is the value of column X_j used for splitting. θ_j is chosen such that Eq. 1.6 is minimized (Eq. 1.4) when using the mean y of each partition as a prediction. Other error metrics can be considered and change depending on the tree implementation. To grow the tree, values of X_j and corresponding θ_j are selected until a stopping criteria is met (e.g., a minimum number of points are reached for each divided set, tree depth, etc.). Because growing a tree in this manner is prone to overfitting, trees tend to be pruned (i.e., the number of splits are reduced) [18].

1.2.6 Random Forest

A Random Forest (RF), as the name suggests, is composed of many decision trees. Each tree in an RF is built by bootstrapping the training data (i.e., make copies of data and repeat entries), selecting a subset of features for each split of a tree (i.e., columns of X),

growing several trees, and then averaging the final decision of each tree to produce a mean prediction (or majority voting if the task is classification). By aggregating the predictions of many weaker models, RF adds additional resilience to overfitting the training data [18].

1.2.7 Boosting

Similar to RF, boosting is a technique that uses trees to produce predictions. Unlike RF, the trees are grown sequentially. Once one tree is grown, the predictions from that tree are fed into the next tree, and so on. Each tree in the sequence corrects the errors in predictions of the previous one. In other words, the predictions of trees are boosted by each subsequent tree [18]. Several implementations of boosting exist, but the most robust (at least given present knowledge) is the implementation of Extreme Gradient Boosting (XGBoost) [20].

1.2.8 Neural Network

A neural network (NN) consists of several key components. The input layer of a NN contains a number of nodes corresponding to the number of features for each data point. The values from these nodes are connected to nodes in hidden layers (i.e., layers between the input and output layers) through weighted connections. The outputs from hidden layers are determined by activation functions, which aggregate all the input values from preceding neurons and produce a single value. These values from each activation function in each neuron are then connected to the next layer with weights, which has its own set of activation functions. Finally, the output layer operates similarly, but provides the final output used by the researcher [18]. In the case of single-target regression, only one continuous output is of interest. However, NNs can also be designed for multi-target regression and classification tasks, which are beyond the scope of this description.

NNs are trained by finding the set of connecting weights between layers that minimize the error on the target variable. A common method for training is stochastic gradient descent, where weights are initialized, subsets of the training data are collected, and the

predictions from the features of those data are obtained. The errors in predictions are then calculated, and the chain rule is used to compute the necessary updates to the weights to minimize the error. This process is repeated for a number of iterations until further updates to the weights do not significantly improve predictions [18].

Chapter 2

Machine Learning Metallic Glass Formation Ability

2.1 What Are Metallic Glasses?

Metallic glasses are a unique class of amorphous material that possess disordered atoms, similar to traditional glasses, but are composed of metallic elements. Unlike conventional crystalline metals, metallic glasses lack the long-range periodic atomic order, resulting in distinctive properties and behaviors [21]. Metallic glasses can be manufactured through various methods, including physical vapor deposition, suction casting, and cold rolling [22]. The formation of metallic glasses through melt-quench methods requires rapid cooling of molten alloys to avoid crystallization and maintain the disordered form of the liquid state. A time-temperature transformation (TTT) diagram (Fig. 2.1) illustrates this process, showing how rapid cooling from a liquid state forms a glass by avoiding the crystalline region.

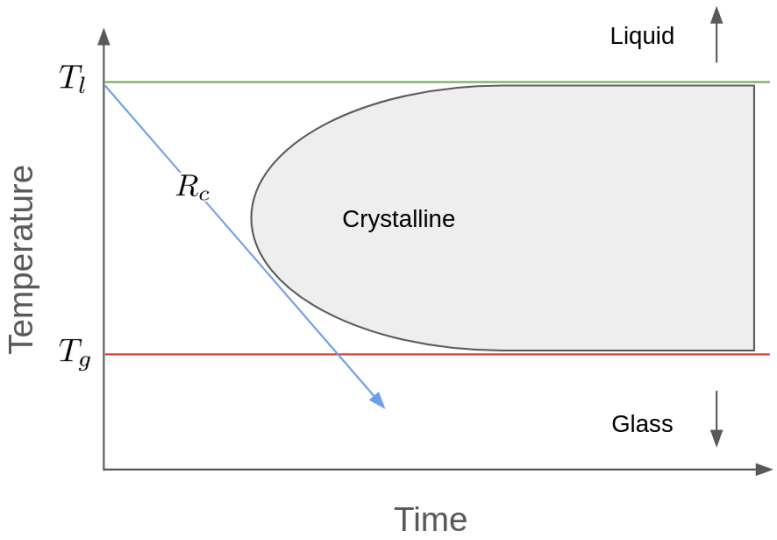


Figure 2.1: A conceptual TTT diagram for glass formation is shown (drawing re-drawn from [23]). T_l, T_g, and R_c are the liquidus temperature, glass transition temperature, and critical cooling rate, respectively.

The critical cooling rate (R_c), the slowest rate at which a molten material can cool while avoiding crystallization, is a measure of glass-forming ability (GFA). Alternative GFA measures exist, such as the maximum casting thickness (Z_{max}). Alloys with lower R_c and higher Z_{max} are considered to have better GFA, as they can form glassy structures more easily and at larger dimensions [24, 25].

2.2 Brief Overview of Applications

The atomic disorder of metallic glasses gives rise to a combination of desirable properties, such as high hardness, good corrosion resistance, and favorable magnetic properties [26–28]. Two engineering applications of metallic glasses are covered in this section. We study metallic glasses because their properties have useful applications.

Metallic glasses are a promising material for use as temporary implants to provide mechanical support during a healing process [27]. One of the key advantages of these materials is their lack of grain boundaries, which allows for more uniform dissolution within the body. This property also makes metallic glasses more resistant to issues like pitting corrosion, which could otherwise lead to premature failure. Ideally, the implants would

gradually dissolve as the surrounding bone or tissue grows and replaces it, providing support until the healing process is complete.

Transformer cores can be manufactured using crystalline, amorphous, or semi-amorphous metals. One of the main challenges in transformer core design is the formation of eddy currents induced by alternating current. These eddy currents reduce the transformer's efficiency by generating heat. Studies have demonstrated that semi or fully amorphous metallic transformer cores exhibit lower eddy current losses compared to crystalline cores [29], which is crucial given the increasing global demand for electricity.

The examples mentioned above represent just a few of the numerous potential applications for metallic glasses. However, the difficulty in forming large samples of these materials has limited their widespread use. Furthermore, the properties of metallic glasses are highly dependent on their alloying constituents, and the effort to map specific alloys to their GFA has been ongoing since the discovery of these unique materials.

2.3 Chapter Summary

Predicting the GFA of metallic alloys had been a long-standing challenge. Many properties, including the glass transition temperature (T_g), crystallization temperature (T_x), liquidus temperature (T_l), fragility, short-range order, enthalpy differences between crystalline and amorphous phases, and many more had been used in attempts to quantify GFA through both experimental and computational endeavors. Measuring certain properties (e.g., T_g) presented bottlenecks in predictive ability due to the necessity of first producing a glassy sample before measurement. Computational approaches for predicting certain properties had been limited by their accessible time and length scales (e.g., viscosity versus temperature to measure fragility with ab initio methods). Properties accessible through classical Molecular Dynamics (MD) simulations suffered from limitations on available interatomic potentials. Our work was an effort to first understand these limits and then find alternate methods to acquire properties related to GFA. First, the study in Sec. 2.4 examined the efficacy of using experimental characteristic temperatures

to model GFA. Next in Sec. 2.5, we simulated our own set of characteristic temperatures using classical molecular dynamics to predict GFA. Finally, we simulated properties with machine-learned interatomic potentials or calculated properties directly to predict R_c in Sec. 2.6.

Sec. 2.4 investigated the use of characteristic temperatures such as T_g, T_x, and T_l to predict D_{max}. Previous efforts to model R_c made use of simple arithmetic functions of T_g, T_x, and T_l (i.e., products, ratios, sums, and differences) [24]. We investigated if these types of functions would improve models for a larger data set with critical casting diameters, D_{max}. Comparison of models using simple functions of characteristic temperatures as features versus only the temperatures themselves showed no statistically significant difference in predictive ability. The inclusion of arithmetic functions of characteristic temperatures as features in the examined models resulted in an increased number of fitting parameters without providing significant improvements in model performance.

The work in Sec. 2.5 conducted MD to calculate characteristic temperatures for metal alloys, which were then used as features to fit GFA regression and classification models. Temperatures derived from cooling curves of self-diffusion, viscosity, and energy were used. Regression models using the logarithm of critical casting thickness showed only weak correlation. However, binary classification models distinguishing poor and good glass formers achieved a maximum $F1$ score of 0.82 ± 0.01 for the best model type. While the predictive performance was modest, this work demonstrated the potential of using MD simulations and machine learning to predict metallic glass forming ability.

The work in Sec. 2.6 explored the prediction of R_c for metallic glasses using a combination of easily computed elemental properties and more complex simulated features. Machine-learned interatomic potentials were used to simulate properties across diverse chemical systems. Various features for 34 materials were produced, including those derived from rheological behavior, characteristic temperatures, short-range order, energy comparisons between crystalline and amorphous phases, etc. The best model for predicting $log_{10}(R_c)$ was constructed using four features: one derived from elemental properties and three from

simulations. This model achieved an R^2 of 0.78 when evaluated through a demanding cross-validation test based on leaving out full chemical systems. While promising, the main contribution lay in demonstrating a versatile approach to systematically explore previously inaccessible material properties across diverse chemical systems using machine-learned potentials. This methodology could be applied to other high-throughput studies of material properties, combining the efficiency of classical interatomic potential methods with the chemical diversity accessible through ab initio techniques.

This work investigated the efficacy of using characteristic temperatures, molecular dynamics simulations, and machine-learned interatomic potentials to develop predictive models for GFA. While much work remains to explore other diverse sets of metallic alloys, the present work establishes a foundation for high-throughput computational characterization and investigation of difficult-to-acquire properties for metallic glasses.

2.4 "Exploration of characteristic temperature contributions to metallic glass forming ability"

Note: This paper has been published as L. E. Schultz *et al.*, "Exploration of characteristic temperature contributions to metallic glass forming ability," *Computational Materials Science*, vol. 196, p. 110 494, 2021, ISSN: 0927-0256. DOI: `https://doi.org/10.1016/j.commatsci.2021.110494`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0927025621002196`, and has been adapted for use in this thesis.

2.4.1 Abstract

Various combinations of characteristic temperatures, such as the glass transition temperature, liquidus temperature, and crystallization temperature, have been proposed as predictions of the glass forming ability of metal alloys. We have used statistical approaches from machine learning to systematically explore a wide range of possible characteristic temperature functions for predicting glass forming ability in the form of critical cast-

ing diameter, D_{max}. Both linear and non-linear models were used to learn on the largest database of D_{max} values to date consisting of 747 compositions. We find that no combination of temperatures for features offers a better prediction of D_{max} in a machine learning model than the temperatures themselves, and that regression models suffer from poor performance on standard machine learning metrics like root mean square error (minimum value of 3.3 ± 0.1 mm for data with a standard deviation of 4.8 mm). Examination of the errors vs. database size suggest that a larger database may improve results, although a database significantly larger than that used here would likely be required. Shifting a focus from regression to categorization models learning from characteristic temperatures can be used to weakly distinguish glasses likely to be above vs. below our database's median D_{max} value of 4.0 mm, with a mean F1 score of 0.77 ± 0.02 for this categorization. The overall weak results on predicting D_{max} suggests that critical cooling rate might be a better target for machine learning model prediction.

2.4.2 Introduction

Physically motivated models built using powers and ratios of sums and differences of experimental measures of the glass transition T_g, onset to crystallization, T_x, and liquidus, T_l, temperatures (so-called characteristic temperatures (CTs)) have been explored for decades to predict metallic glass forming ability (GFA). We will call these powers and ratios of sums and differences PRSD functions in this paper. There exist several quantitative measures of GFA. The critical cooling rate, R_c, is the slowest a molten metal can be cooled to produce a glass. The smallest dimension of the largest glassy sample for a composition is defined to be Z_{max} whereas the maximum rod diameter of a glassy specimen manufactured through suction casting is the critical casting diameter, D_{max}. Both D_{max} and Z_{max} denote the maximum reachable thickness for a glass but differ by geometry. Hence, predictions on either D_{max} or Z_{max} denote the ability for a model to quantify the maximum thickness for a glassy metal sample. Starting in 1969, the ratio between T_g and T_l was introduced to quantify the ease of forming a bulk metallic glass [30]. This was the original form of the reduced glass transition temperature, $T_{rg} = T_g/T_l$.

Attempts to use the melting temperature instead of T_l were made but resulted in worse models for $log_{10}(R_c)$ [31].

More modern efforts to model GFA with CTs have only modestly increased in complexity and still generally focus on correlating a metric of GFA with PRSD functions of CTs. For instance, $\gamma = T_x/(T_g + T_l)$ was introduced in 2002 and is a single term ratio between two quantities [32]. The γ parameter was constructed by using the devitrification range, $(T_x - T_g)$, and simplifications based on classical crystal growth and nucleation theory. Lu and Liu [32] showed that γ shows a strong relationship with $log_{10}(R_c)$ ($R^2 = 0.91$) but much weaker correlation with critical section thickness Z_{max}, ($R^2 = 0.57$).

In 2005, another parameter based on PRSD functions of CTs was introduced as $\alpha = T_x/T_l$ [33]. The difference between this parameter and T_{rg} was the substitution of T_x for T_g. In the same work, $\beta = T_{rg} + T_x/T_g$ was also introduced. The R^2 scores for α and β were 0.90 and 0.93 against $log_{10}(R_c)$ respectively. Similar to the parameter γ, both α and β degraded in R^2 performance for prediction on Z_{max}, with R^2 of 0.48 and 0.54, respectively.

Another PRSD function of CTs used as a feature for GFA is $\omega = T_g/T_x - 2T_g/(T_g + T_l)$ introduced in 2008 [24]. This parameter takes into account the devitrification range and liquid stability. With an R^2 of 0.93 against $log_{10}(R_c)$ for 53 metallic alloys, ω provides the best performing model for $log_{10}(R_c)$ of which we are aware to date. Since D_{max} should increase with a decreasing R_c, the comparison between $1/\omega$ and D_{max} was performed in the same work. The R^2 was 0.41 which follows previous performance trends of relatively poor R^2 of PRSD functions of CTs for D_{max} even when good performance for R_c was obtained.

A common theme throughout attempts to model GFA from CTs is that they are based on linear models of PRSD functions, and usually just a single PRSD function. They also have a tendency to do well with learning on $log_{10}(R_c)$ but not on D_{max} or Z_{max}. See Refs. [34, 35] for more comparisons between these types of linear models and their correlations against D_{max}.

Recently, there have been machine learning (ML) efforts to learn D_{max} from CTs that go beyond simple linear correlations with PRSD functions. Because of the availability of more D_{max} data compared to R_c measurements, more advanced ML techniques can be implemented and reliably assessed for learning on D_{max}. Specifically, Xiong, et al. [35] used a Gaussian process (GP) model to learn D_{max} from CTs. The study predicted D_{max} on 442 metallic glasses with an R^2 of 0.76 for the training set. Even more recently, Deng and Zhang [36] used the random forest (RF) method to learn D_{max} from CTs and some other features for the same dataset as used by Xiong, et al. [35]. Deng and Zhang found an R^2 of 0.64.

Thus far, the largest ML attempt to quantify D_{max} as a function of CTs was done by Xiong, et al. [37] with 674 compositions. RF models were trained on the three CTs mentioned in this study. They used 100-fold cross-validation (CV) where an RF model was trained on 99 folds while its performance was measured by the leave out set. Their model had an R^2 of 0.60 ($R = 0.77$) and a root mean squared error ($RMSE$) of 2.89 mm. However, this result was attained by excluding six erroneous compositions which showed large residuals in D_{max} predictions for another one of their models. In the spirit of generating a PRSD function of CTs, Xiong, et al. also used symbolic regression with CTs to generate a three-term model on their dataset (Equation 2.1). The symbolic regression model scored an R^2 of 0.45 ($R = 0.67$) and an $RMSE$ of 3.37 mm [37]. Whether PRSD functions of CTs can aid in quantifying D_{max} through other ML methods remains an open question.

$$D_{max} = f\left(\frac{T_g^2}{T_l^3}, \frac{1940}{136263T_l - 1426T_l(T_x - T_g)}, \frac{T_g^2}{39T_l(14T_l - (T_x - T_g)^2)}\right) \tag{2.1}$$

While the ML approaches to learning D_{max} from CTs appear to be performing better than linear PRSD functions of CTs, many ML studies to date suffer from several shortcomings. First, model performances tend to be measured with R^2. Other error metrics such as mean average error (MAE) and $RMSE$ are generally not provided. Second, and more importantly, the reported R^2 for some previous efforts are for predicting back on

training data, rather than assessment on test data not seen in the fitting process. Thus, there was no assessment of the extent to which trained models have predictive power outside their training sets. Having just training data results is common when modeling D_{max} using simple linear models of PRSD functions of CTs, and while this may lead to some overfitting and underestimation of errors, the simplicity of these models may make these underestimations negligible. However, more complex ML are particularly subject to overfitting and careful assessment with test data is essential for robust assessment of predictive ability.

Our work focuses on providing multiple metrics of assessment for ability to predict on D_{max} on carefully excluded validation data, avoiding data leakage through nested cross-validation. Our approach generalizes previous use of select PRSD functions as features by generating a comprehensive set of PRSD functions up to reasonable powers. We then explore the efficacy of these features in multiple ML model types, including least absolute shrinkage and selection operator (LASSO), Gaussian kernel ridge regression (GKRR), RF, and Gradient Boosting (GB). The benefit of using ensemble models is the inherent feature selection that they provide. The L1 norm for LASSO tends to penalize arbitrary feature weights to zero which is also a form of feature selection. This approach can therefore assess whether features based on PRSD functions of CTs yield effective predictive models of D_{max} and to what extent the use of PRSD functions provide better predictions than just using the CTs themselves. We also test whether learning on the $log_{10}(D_{max})$ is more effective than learning on D_{max} and if applying principal component analysis (PCA) to transform our features had any added benefit. Our dataset is comprised of 747 compositions, which is the largest set to date used for building and assessing models to predict D_{max}.

2.4.3 Materials and Methods

Experimental data was provided by Ref. [38]. Measurements of D_{max} are susceptible to differences in experimental setup which could impact cooling rate. The experiments in the database were all melt quenched using similar rod like molds and melting processes

that are standard among experimentalists. Only integer D_{max} values are reported which introduces small uncertainties. The characteristic temperature T_l should be less effected by heating and cooling rates than T_g and T_x. In the database, entries were not constrained to certain heating and cooling rates, but the majority of the heating rates fall into the range of 10-100 degrees per minute as this is fairly standard procedure for the differential scanning calorimetry (DSC) used to measure these values. The impact of variation in heating and cooling rates on the ML accuracy is difficult to asses and is an interesting topic for further study but we have not attempted to explore it here.

A set of unique compositions provided by Ref. [38] with values of T_g, T_x, T_l, and D_{max} were used to generate features. First, differences and summations between pairs of temperatures were taken (e.g., (T_g+T_x) and (T_l-T_g)). These features were then raised to the powers of -4 to 4 (e.g., $(T_g+T_x)^2$ and $(T_l-T_g)^{-3}$). Products between all aforementioned features were then included to produce the feature set of PRSD functions of CTs (e.g. $(T_g+T_x)^2 \cdot (T_l-T_g)^{-3}$). Any instance that resulted in a division by zero was eliminated from the analysis. For compositions that appeared more than once in the database and had a full set of the three CT values, the maximum value for D_{max} and the mean values for CTs were used. However, if any one instance of a CT was more than 50 K from the mean, then that value was excluded and the mean was taken with the remaining points to minimize erroneous CTs values. This data processing reduced the complete database of 6,914 entries to 747 unique compositions, each with a D_{max} value, three unique characteristic temperatures (T_g, T_x, T_l) and 2,628 features from PRSD functions of CTs described above. The processed data can be found in Ref. [39] and Ref. [40].

Features were standardized to have a zero mean and unit variance. A separate feature set was generated by applying principal component analysis (PCA) while keeping all principal components to transform features. Some of the PRSD functions of CTs features are highly correlated which means that they provide similar information. PCA can be used to transform a feature set to a lower dimension or to an equivalent set with linearly independent features. The principal components are orthogonal vectors and explain the maximum variance of data along several directions. The larger a singular value for a

corresponding principal component, the more data that principal component represents from the original dataset. Therefore, data sets with linearly dependent features can be transformed to a linearly independent data set with PCA if all nonzero principal components are kept for the projection.

In ML, having more features (explanatory variables) than observations when building a model can lead to overfitting. An overfit model generalizes poorly and is less likely to correctly predict a target variable for cases withheld from training. The degree of overfitting tends to increase with the number of features included in training. Because the number of generated features in the present study are much larger than the number of observations, we apply models that reduce the number of contributing features via regularization (LASSO), shrinkage (GB), and bagging (RF). We will show that a subset of features that number less than the number of observations contribute to predictions by a GB model (Section 2.4.4) which is likely to hold true for LASSO and RF model types as well.

All assessment done used a nested CV approach [41]. The nested CV used a 5-fold inner and 5-fold outer loop. A grid search of hyperparameters was applied for the inner loop in the nested CV (Table 2.1) to establish optimal hyperparameters for each outer loop fold. Fitting was then done for each outer CV fold on the full training set not in the test fold (80% of the data). Then, the model was applied to the outer CV test set fold. To test whether the generated features provide improved prediction, we also applied nested CV with only T_g, T_x, and T_l. Data were randomized every time nested CV was applied, meaning that the splits for the testing and training sets may have differed for comparisons. All ML was performed with scikit-learn [42].

Table 2.1: The hyperparameter grid for each model type explored is tabulated below. Variable names follow the convention of scikit-learn [42].

Model	Parameter	Values
LASSO	alpha	100 values from 0 to 5 in log10 space
GKRR	alpha	100 values from -5 to 5 in log10 space
	kernel	rbf
	gamma	100 values from -3 to 3 in log10 space
RF	n_estimators	30, 40, 50, 60, 100, 500
	max_features	sqrt, log2, None
	max_depth	2, 3, 4, None
GB	learning_rate	0.001, 0.01, 0.1, 0.2
	n_estimators	30, 40, 50, 60, 100, 500
	max_features	sqrt, log2, None
	max_depth	2, 3, 4

For LASSO, GKRR, RF, and GB, the following set of four tests were performed: fit with non-PCA features and D_{max}, fit with PCA features and D_{max}, fit with non-PCA features and $log_{10}(D_{max})$, and fit with PCA features and $log_{10}(D_{max})$. $RMSE$, MAE, R^2, and $RMSE/\sigma$ were calculated for each fold in the outer loop from the nested CV (where σ is the standard deviation of the true target values in that fold). This gave five values for each metric for each nested CV run, and we performed one nested CV run for each test, for a total of 5 values for each metric (i.e., 5 values of $RMSE$, 5 values of MAE, etc.). These distributions of values for each metric were used to find the mean value, standard deviation (STDEV), and standard error in the mean (SEM), for each metric from each nested CV. All metrics with units are in units of D_{max}, which is in millimeters (mm). If fitting was done with $log_{10}(D_{max})$, then the predicted output was transformed back to

D_{max} before calculating error metrics. All metrics are scores from outer folds of nested CV runs unless explicitly stated otherwise.

Separate from assessing the accuracy of our models, we use nested CV, we would also like to develop the most complete and accurate model possible using the whole database. To do this a GB model was trained using all the data. The optimal hyperparameters were found by applying a grid search using 5-fold CV on the whole data set and call this model GB 1. From GB 1, we can assess which of the generated features provide the most utility for regression prediction. We studied the impact of fitting GB models with the top n features by building a learning curve with 5-fold CV and measuring $RMSE/\sigma$. The curve was averaged over the leave out sets. The uncertainties are in SEM. The choice of hyperparameters were kept from GB 1. The 50 highest ranking features were used to fit a final GB model, named GB 2, because regression performance did not significantly change by including the remaining features. GB 2 can be found at the Materials Data Facility (MDF) online data and code sharing repository at Ref. [39] and figshare at Ref. [40].

To test whether the generated features had a significant impact on learning, we performed a two-sample T-test for the distribution of five scores for each metric obtained above using all model types. One distribution of scores were from the generated features while the others were from using just the three CTs as features. Both feature sets did not have a PCA transformation and learned from D_{max} directly. The aforementioned comparison choice was performed because models tended to degrade in performance with PCA transformed features and with application of a logarithm onto D_{max}.

For LASSO models using PRSD functions of CTs, models had unusually poor regression performance on some test folds for nested CV. Many generated LASSO models were not well conditioned due to numerical problems and gave outlandish predictions. As a result, an outer fold was removed if any of the regression metric values were outside of a multiple of 3 from the optimal GB workflow metrics. This left three outer folds for any LASSO metric reported for all combinations of logarithm application on D_{max} and PCA application on the feature set. Due to the low number of individual observations,

LASSO models were excluded from the two-sided T-tests. All LASSO models trained during hyperparameters grid searches showed acceptable convergence.

A learning curve was generated to test if improved learning would result from more data. Nested CV was performed for the best GB workflow for 10% up to 100% for the 747 compositions by increments of 10%. Each subset of data was randomly sampled. Learning curves were built from the predictive performance of the testing sets from nested CV.

To test if the generated features had predictive power for classification, nested CV was performed using a GB classifier. No PCA was applied nor a logarithm onto D_{max}. Any composition with a D_{max} less than 4.0 mm were assigned to be class 0 while all others were assigned class 1. The median D_{max} value for the dataset was 4.0 mm which splits the classes evenly. Binary classification scores for every test set in nested CV were gathered.

We mark our contribution to the field by comparing our best performing regression workflow to the work done by Xiong, et al. in Ref. [37]. We make sure to match the number of folds used in their CV and the same set of features. Additionally, we attempt to predict D_{max} with our dataset using their reported model shown in Equation 2.1. We use ordinary least squares with three fitting coefficients and one intercept term to fit Equation 2.1 to all of our data. The least squares model was then used to predict back on our full dataset.

2.4.4 Results and Discussion

The lowest $RMSE/\sigma$ model types were GB and RF from learning on the generated feature set without PCA and using D_{max} (rather than $log_{10}(D_{max})$) as the target feature. The mean of MAE was slightly lower for the optimal GB workflow (Tables 6.2-6.5). The lowest mean $RMSE/\sigma$ (MAE) for GB and RF were 0.70 (2.18 mm) and 0.70 (2.24 mm) respectively. For all model types, it was found that fitting on D_{max} instead of $log_{10}(D_{max})$ gave better performance for mean $RMSE/\sigma$ from nested CV. Application of PCA was better for GKRR and LASSO, and even in that case the effect was small. All values are

tabulated on Table 2.2. Scores for using the just the three CTs as features are provided in Tables 6.6-6.9.

Table 2.2: The mean $RMSE/\sigma$ scores for each test for each model type along with their standard deviations (STDEV) and standard error in the mean (SEM).

Model	Log_{10}	PCA	Mean	STDEV	SEM
GB	False	False	0.70	0.08	0.04
GB	False	True	0.86	0.10	0.05
GB	True	False	1.37	0.04	0.02
GB	True	True	1.38	0.06	0.03
GKRR	False	False	0.76	0.06	0.03
GKRR	False	True	0.75	0.09	0.04
GKRR	True	False	1.39	0.09	0.04
GKRR	True	True	1.38	0.05	0.02
LASSO	False	False	0.86	0.03	0.02
LASSO	False	True	0.81	0.04	0.02
LASSO	True	False	1.40	0.05	0.03
LASSO	True	True	1.36	0.03	0.02
RF	False	False	0.70	0.04	0.02
RF	False	True	0.92	0.14	0.06
RF	True	False	1.37	0.06	0.03
RF	True	True	1.38	0.05	0.02

GB 2, which was trained on the top 50 ranking PRSD functions of CTs, is shown in Figure 2.2. GB 2 was attained by using the optimal hyperparameters from GB 1 and showed outstanding performance on regression metrics $RMSE/\sigma = 0.30$ and $R^2 = 0.91$ (not from nested CV). While GB 2 is taken as the best overall model for predicting new data as it is fit and optimized on all our present data, the 5-fold CV performance of GB 2 cannot be taken as predictive for new data due to data leakage and overfitting. The error metrics from the nested CV are the best predictor of the expected performance on new

data for GB model types.

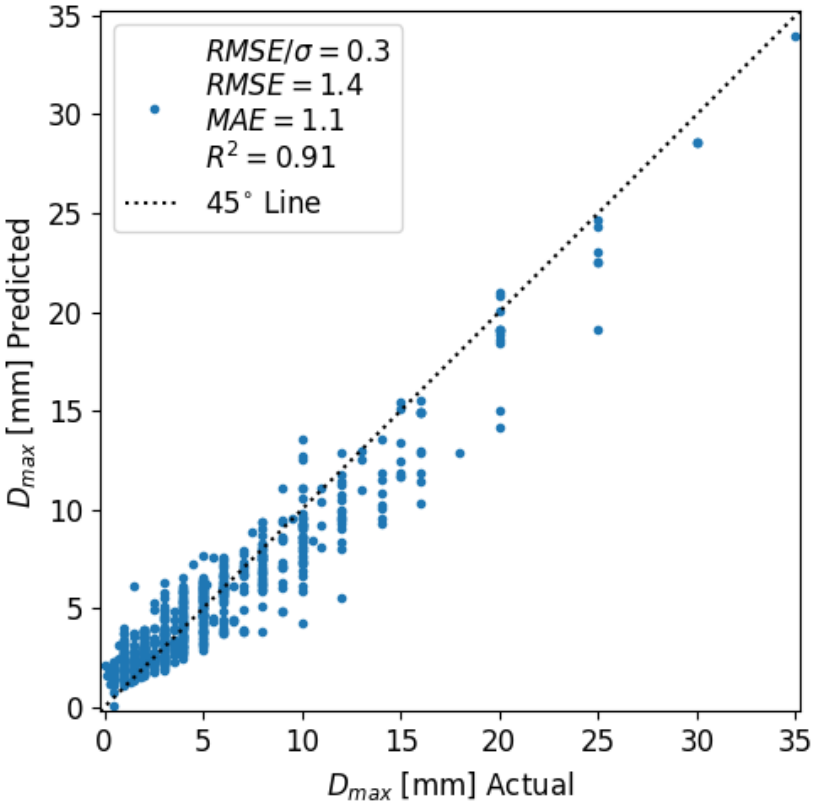


Figure 2.2: The parity plot for training set prediction for GB 2 is shown.

GB 1 ranked 1,042 out of the 2,628 features to be of some nonzero significance. A subset of those features with scores are tabulated in the Appendix. This is a very large number of features, even more than the number of data points, and suggests that the model is very poorly constrained. However, the actual number of significant parameters that impact the model is likely far fewer. To asses the truly significant parameters, GB models were fit incrementally with subsets of features, starting with the highest ranking, then the top two ranking, then the top three ranking, and so on. The GB models used the GB 1 hyperparameters described in the Methods section. $RMSE/\sigma$ with respect to the number of included features, ordered by their ranking, is shown in Figure 2.3. There is essentially no gain in in prediction performance after about 50 features. Thus, a GB model represented with all the features can be equally represented with just the first 50 features as these are all that really contribute to its accuracy. In general, it is a concern

when one has more fitting parameters than data points. Although models mentioned in this work can formally provide fits when they have more fitting features than cases, only a number much less than 747 are found in this work to contribute to predictions when training on PRSD functions of CTs as shown by GB 2.

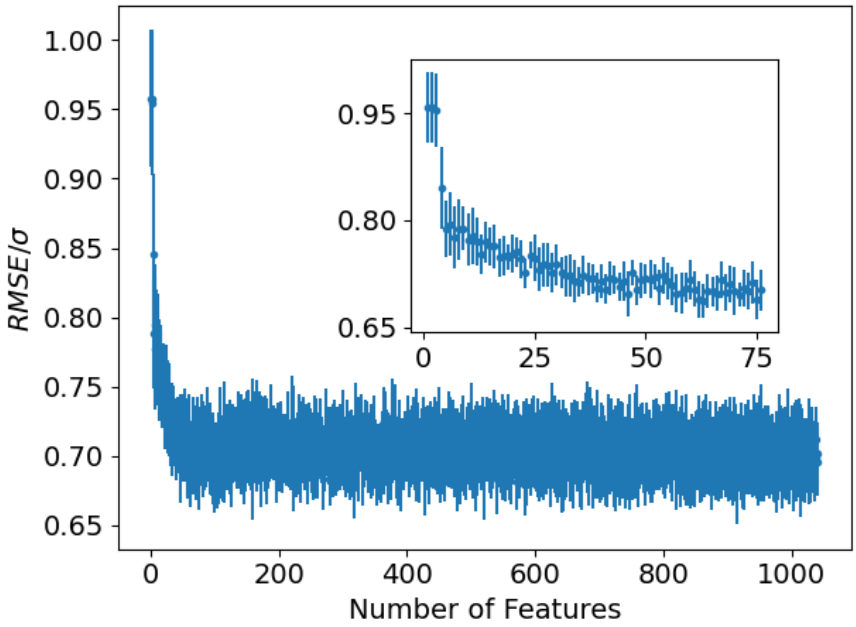


Figure 2.3: The number of features included in GB models as a function of $RMSE/\sigma$.

The p-values for the two-sided T-test for comparing results from using features that are PRSD functions of CTs and just the three CTs are reported in Table 2.3. None of the values for all reported metrics fell below 0.32, far from the value of 0.05 typically used as a cutoff to claim significant difference. Hence, there was no statistically significant difference between learning from the generated features versus the original CTs.

Table 2.3: The p-values for comparing models based on generated features and just three CTs for each ML scoring metric.

Model	MAE	$RMSE$	$RMSE/\sigma$	R^2
GB	0.47	0.67	0.79	0.83
GKRR	0.78	0.52	0.58	0.60
RF	0.63	0.53	0.32	0.33

The optimal GB values of $MAE = 2.18 \pm 0.09\ mm$, $RMSE = 3.28 \pm 0.13\ mm$, $RMSE/\sigma = 0.70 \pm 0.04$, and $R^2 = 0.50 \pm 0.05$ are generally quite poor. In particular, the $RMSE/\sigma = 0.70$ suggests only modest improvement over simply guessing the mean of the dataset (which gives $RMSE/\sigma = 1$) and $R^2 = 0.50$ is well below the qualitative guide of $R^2 \approx 0.7$ that is often used to consider a result of significance. Therefore, these results suggest that it is unlikely that the CTs studied here can be used to provide a quantitative regression model for D_{max} with a data set similar to that we have examined.

One way to potentially improve the models for D_{max} relative to those presented in this work is to add more data. To test the effect of the size of data on learning, a learning curve was produced using the best GB workflow for regression on D_{max} (Figure 2.4). After 50% of the data are included, only minor improvements on $RMSE/\sigma$ were found, with a reduction from 0.77 at 50% of the data to 0.70 at 100% of the data. Consequently, a modest increase in the amount of data is unlikely to significantly improve learning D_{max} using CTs. For example, assuming a linear extrapolation of rate of decrease from 50% to 100%, the total database would have to grow by 300% to get below a reasonable performance target of $RMSE/\sigma = 0.3$.

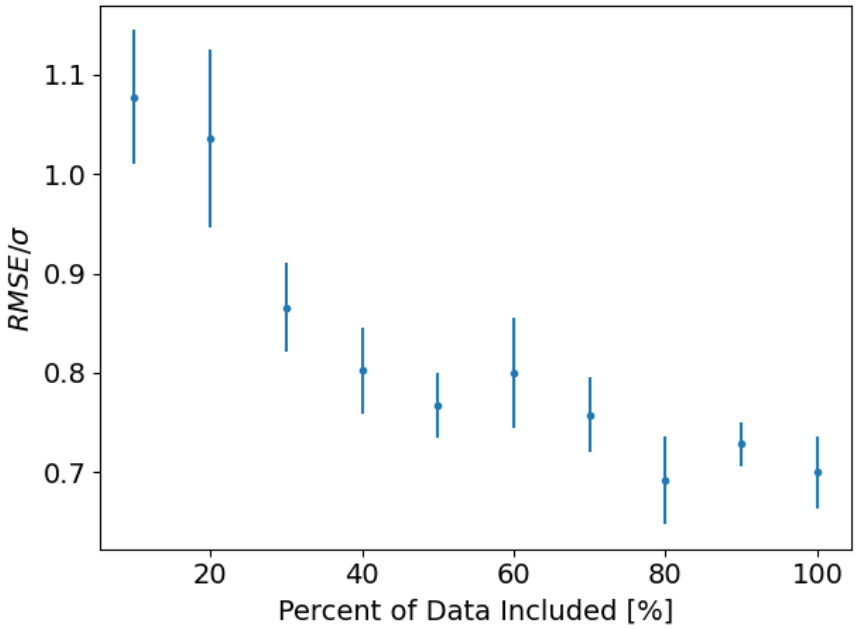


Figure 2.4: $RMSE/\sigma$ decreases with an increase in the amount of data considered. The error bars are the standard error of the mean from all outer loop test sets in nested CV (see Sec. 2.4.3).

Another way to potentially improve the models is to explore a simpler classification in place of the full regression model. Here we consider classification into glasses with $D_{max} < 4\ mm$ or $D_{max} \geq 4\ mm$, where $4\ mm$ is the median of the dataset. The binary classification metrics for the GB classifier are shown in Table 2.4. Since the number of classes are near equal for the outer folds from nested CV, the baseline for F1 and receiver operator characteristic (ROC) area under the curve (AUC) scores is around 0.5. The scores for F1 above or equal to 4.0 mm, F1 below 4.0 mm, and ROC AUC are 0.77, 0.78, and 0.78 respectively, which represent a significant, although not outstanding, predictive ability. Although the regression metrics for predicting D_{max} showed large uncertainties, we have shown that CTs are still potentially useful for classifying glasses above or below the median D_{max}. The success of classification can be understood by examining the parity plot for the GB regression workflow. As seen in Figure 2.5, predicted and actual D_{max} values deviate significantly but show enough correlation that cases below 4.0 mm tend to have predictions below 4.0 mm. Likewise, cases above or equal to 4.0 mm tend to be predicted above or equal to 4.0 mm.

We compared our workflow of GB regression to previous efforts by Xiong, et al. in Ref. [37] because they have the next largest database for metallic glass D_{max}. When applying 100-fold nested CV, our aggregate $RMSE = 3.4\ mm$ and $R^2 = 0.50$ which is comparable to their scores of $RMSE = 2.89\ mm$ and $R^2 = 0.60$ since no compositions with large residuals were excluded from our assessment. When an ordinary least squares model was fit and used to predict back D_{max} from the terms in Equation 2.1 for our dataset of 747 compositions, R^2 and $RMSE$ become 0.07 and 4.6 mm respectively. Each term added a degree of freedom for fitting and we included the fitting intercept. Through further private communication with the authors, the general methodology to train the symbolic regression model may have suffered from data leakage.

Table 2.4: The binary classification metrics for distinguishing metallic glasses above and below the median D_{max} from nested CV are tabulated below.

Metric	Mean	STDEV	SEM
Cases for $D_{max} < 4\ mm$	74.40	6.15	2.75
Cases for $D_{max} \geq 4\ mm$	75.00	6.56	2.93
Accuracy	0.78	0.04	0.02
F1 for $D_{max} < 4\ mm$	0.78	0.04	0.02
F1 for $D_{max} \geq 4\ mm$	0.77	0.04	0.02
Precision for $D_{max} < 4\ mm$	0.76	0.07	0.03
Precision for $D_{max} \geq 4\ mm$	0.80	0.06	0.03
ROC AUC	0.78	0.04	0.02
Recall for $D_{max} < 4\ mm$	0.81	0.06	0.03
Recall for $D_{max} \geq 4\ mm$	0.74	0.08	0.03

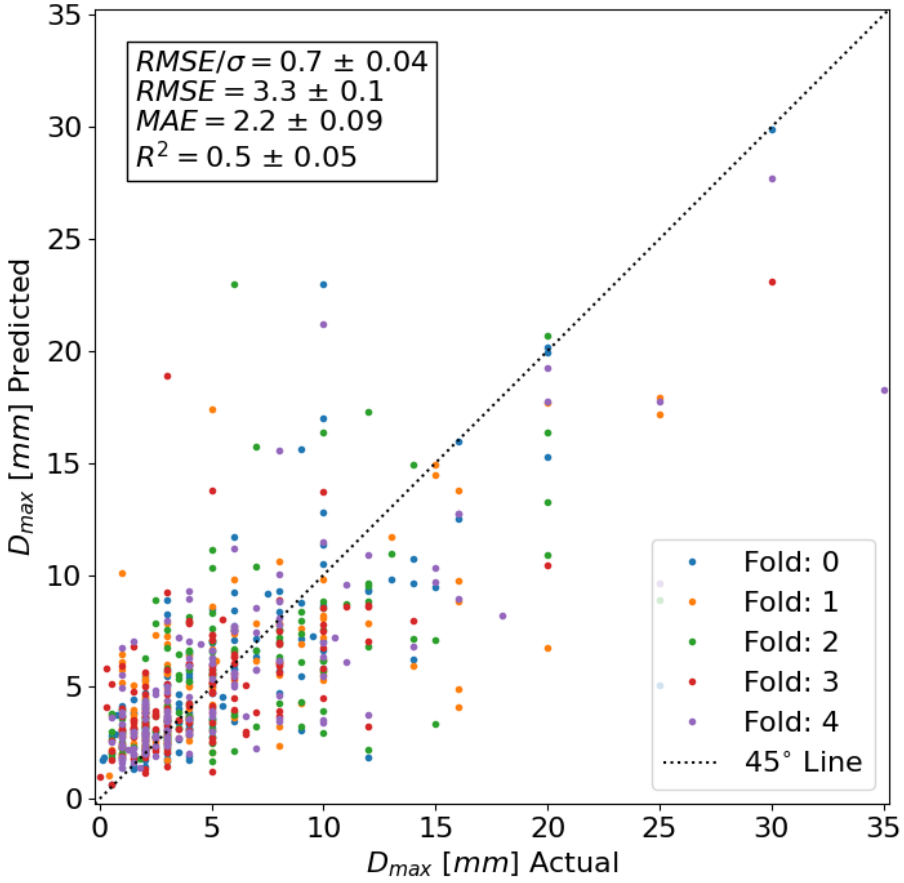


Figure 2.5: The parity plot for test set prediction for the GB workflow is shown. The metrics in the annotation have SEM as the uncertainty.

2.4.5 Conclusion

We have assessed the ability of features based on the characteristic temperatures (CTs) T_g, T_x, and T_l, to predict the critical casting diameter, D_{max}. We explored an extensive search of features based on powers and ratios of sums and differences of CTs, multiple machine learning models, and used nested cross validation to avoid data leakage when assessing the models. We found only weak ability for the models to predict D_{max} and found that to achieve significant improvement from increasing the database size would likely require a few multiples of the present database size. Given that we are already using the largest aggregated database to date, such an increase in amount of data would likely require a very large experimental effort or application of new high-throughput approaches.

We also found that using just T_g, T_x, and T_l directly was not statistically different than

using features based on the powers and ratios of their sums and differences. These results suggest that further efforts adding terms within the examined space of features will not yield better predictive performance outside their training set compared to using the CTs directly. Some success was found in predicting D_{max} above or below its median value from the CTs, suggesting that they can provide some valuable D_{max} information. For example, models using these CTs could be used to screen small glassy samples and determine if larger glasses might be produced. Nevertheless, it appears that D_{max} cannot be quantified with regression models built with the set of CTs examined. Previous linear models using CTs appear to have had more success when quantifying R_c than D_{max}. This suggests that further exploration of R_c models might be more fruitful than D_{max} models. However, more complex models and more thorough assessment are limited by the limited amount of R_c data, and more of such data would help in developing and assessing optimal CTs models.

2.4.6 Data Availability

The raw data required to reproduce these findings are available to download from DOI: 10.18126/7yg1-osf2 version 1.3 through the Materials Data Facility. The processed data required to reproduce these findings are available to download from `https://petreldata.net/mdf/detail/schultz_gb_model_full_fit_v1.1/`.

2.4.7 Impact from Work

- This work showed that using powers, ratios, sums, and differences of characteristic temperatures as features does not significantly improve machine learning models for predicting the critical casting diameter of metallic glasses compared to just using the characteristic temperatures directly.
- The study highlights that using characteristic temperatures in machine learning models provides only weak predictive power for the critical casting diameter, suggesting that a significant increase in data size would be necessary for substantial improvements.

2.5 "Molecular dynamic characteristic temperatures for predicting metallic glass forming ability"

Note: This paper has been published as L. E. Schultz *et al.*, "Molecular dynamic characteristic temperatures for predicting metallic glass forming ability," *Computational Materials Science*, vol. 201, p. 110 877, 2022, ISSN: 0927-0256. DOI: https://doi.org/10.1016/j.commatsci.2021.110877. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927025621005899, and has been adapted for use in this thesis.

2.5.1 Abstract

We explore the use of characteristic temperatures derived from molecular dynamics to predict aspects of metallic Glass Forming Ability (GFA). Temperatures derived from cooling curves of self-diffusion, viscosity, and energy were used as features for machine learning models of GFA. Multiple target and model combinations with these features were explored. First, we use the logarithm of critical casting thickness, $log_{10}(D_{max})$, as the target and trained regression models on 21 compositions. Application of 3-fold cross-validation on the 21 $log_{10}(D_{max})$ alloys showed only weak correlation between the model predictions and the target values. Second, the GFA of alloys were quantified by melt-spinning or suction casting amorphization behavior, with alloys that showed crystalline phases after synthesis classified as Poor GFA and those with pure amorphous phases as Good GFA. Binary GFA classification was then modeled using decision tree-based methods (random forest and gradient boosting models) and were assessed with nested-cross validation. The maximum F1 score for the precision-recall with Good Glass Forming Ability as the positive class was 0.82 ± 0.01 for the best model type. We also compared using simple functions of characteristic temperatures as features in place of the temperatures themselves and found no statistically significant difference in predictive abilities. Although the predictive ability of the models developed here are modest, this work demonstrates clearly that one can use molecular dynamics simulations and machine

learning to predict metal glass forming ability.

2.5.2 Introduction

Determining metal alloy compositions that have good glass forming ability (GFA), and in particular that yield bulk metallic glasses, has been a grand challenge in metallic glass research for decades. Many previous attempts to predict GFA for metals have made use of some function of the temperatures characterizing important aspects of the melt behavior, sometimes called characteristic temperatures. The characteristic temperatures most used for this purpose are the glass transition (T_g), the onset of crystallization (T_x), and liquidus (T_l) temperatures. For example, the reduced glass transition temperature, $T_{rg} = T_g/T_l$, is one of the earliest and most iconic GFA indicator [30]. A very successful model for GFA predicts the critical cooling rate, R_c, as a linear function of $\omega = T_g/T_x - 2T_g/(T_g + T_l)$. When fit to 53 metallic glasses, a linear function between ω and R_c had an R^2 of 0.93 [24]. T_{rg} and ω are just two of over 20 functions of T_g, T_x, and T_l that have been proposed to quantify GFA [34, 35]. Although successful models have only been shown for a small number of alloys, there are clear indications that insights into GFA can be given by characteristic temperatures.

Another GFA indicator, in this case from the melt, is the liquid fragility, m, which is measured by finding the slope of viscosity as a function of temperature near T_g for an alloy. Glasses with higher viscosities when approaching T_g are said to be strong (low m) and are thought to suppress the kinetics for crystallization. Conversely, glasses that experience low viscosities upon cooling are said to be fragile [43, 44]. It was shown that a linear combination of T_{rg} and m fit to $log_{10}(D^2_{max})$ for 42 glassy alloys had an outstanding R^2 score of 0.980 [25]. Here, D_{max} is the critical casting diameter. Hence, D_{max} can be written as a relatively simple function of T_g, T_l, and m. Although m is not strictly a characteristic temperature it is similar in spirit as it represents the temperature dependent physics of the melt, and it can be related to a characteristic temperature as discussed later.

Quantitative relationships between the most important and intrinsic measure of GFA, specifically R_c and D_{max}, and characteristic temperatures (or closely related melt properties) can be established. However, the aforementioned relationships have limited utility for new material discovery because a glass must be synthesized to obtain the features T_g, T_x, and m. T_l must also be obtained, although this temperature can be measured without making a glass and is often accessible through thermodynamic modeling without actual synthesis. An exciting design opportunity would be realized if we could access T_g, T_x, T_l, and m from molecular simulations, as this would allow the above correlations to be used for computational prediction of GFA. It is this opportunity that motivate the present work.

T_l can be predicted from molecular simulations quite accurately [45–47] and is often available from thermodynamic models fit to experiments for relevant alloys, so we will not focus further on this quantity and simply take it from experiments or online phase diagrams when available for the rest of the present study. T_g, T_x, and m are all in theory accessible to molecular simulations but have major practical challenges. T_g from molecular dynamic (MD) studies are strongly impacted by the very fast cooling rates necessitated by the short time scales accessible to MD. The MD values of T_g tend to be higher than experimental values due to heating/cooling rate differences and adopted methodology [21, 48]. See Ref. [48] for a comparison of experimental and MD approaches for finding T_g. Finally, another set of limitations are imposed by T_x calculation. As seen in Ref. [49], the crystallization kinetics in MD for a single composition varies by system size and annealing temperatures explored. The aforementioned indicates that mimicking an experimental T_x value from differential scanning calorimetry (DSC) would pose many challenges in choices of system sizes, cooling rates to attain a glass, heating rates, and the starting temperature to heat a material. m is also difficult to practically calculate from MD as it requires determining viscosity as a function of temperature near T_g, which is impractical due to the slow kinetics near T_g. Equations with reliable extrapolation to low temperature viscosities have their own set of challenges for finding their fitting parameters with MD [50, 51]. Due to these obstacles, direct MD of some experimental

characteristic temperatures is currently impractical.

Nevertheless, MD accessible quantities that approximate or correlate with some of the previously defined material properties exist. For example, Kelton et al. have shown correlations between m and the ratio between T_g and a temperature where a set of compositions cross a set viscosity value, T^* [52, 53]. They further showed that T_g can be captured as a function of T^*, and the crossover from Arrhenius behavior temperature, T_A^* [54]. Specifically, a fit between T_A^*/T_g and T_g/T^* had an R^2 of 0.96 [53]. The results from Kelton et al. and in Ref. [25] together imply that D_{max} should be a function of T_g, T^*, T_A^*, and T_l.

Other alloy characteristic temperatures are MD acquirable and included in this study. We choose to examine self-diffusion derived temperatures because trends between self-diffusion, viscosity, and relaxation times with respect to temperature show strong relationships, as seen in Ref. [55]. Parallel to the way T^* and T_A^* are defined for viscosity, we define $T^{'}$ and $T_A^{'}$ as the temperatures where diffusivity reaches a critical value and where diffusivity deviates from an Arrhenius trend, respectively. We can find an approximate T_g by direct high-rate cooling, which we call T_{gm}. Approximate T_g values can also be estimated by fitting to high-temperature kinetic properties with a simple Vogel-Fulcher-Tamman (VFT) form and extrapolating. We use this approach to define T_g^* as the temperature where extrapolated viscosity reaches $10^{12}\ Pa \cdot s$ [56] and $T_g^{'}$ as the temperature where extrapolated self-diffusion values reach $10^{-12}\ \text{Å}^2/ps$ (which is the method used to define T_g in Ref. [57]).

We argue and show that GFA insights could be gained from the MD characteristic temperatures of T_{gm}, T_g^*, T^*, T_A^*, $T_g^{'}$, $T^{'}$, $T_A^{'}$, and T_l. We use both regression and classification machine learning (ML) models to quantify the ability of models to learn GFA from MD characteristic temperatures. Although the examined data sets were small, preliminary results suggest that MD quantification of GFA is possible.

2.5.3 Methods

2.5.3.1 Molecular Dynamics

All MD simulations were performed with the LAMMPS package [58]. The time step was set to 1 fs. Periodic boundary conditions were applied along all directions. Finnis-Sinclair (FS) and embedded-atom model (EAM) potentials were used to simulate alloys of interest [59–65]. The starting MD supercell structures were built with a repeating simple cubic unit cell with 1,000 atoms. The element type for each atom was assigned randomly to match compositions of interest. An isothermal hold at 2,000 K was run for 100 ps under the NPT ensemble and was used as the initial trajectory for melt quench simulations. In the NPT ensemble, N is the number of atoms, P is the pressure, and T is the temperature and are all held constant.

Continuing from the starting structures, isothermal holds were constructed by dropping 100 K from the preceding hold. Each of the holds were ran for 10 ps which gave a cooling rate of 10^{13} K/s. The final temperature probed was 100 K. For each of the isothermal holds from the melt-quench simulation, the final trajectory was run for an additional 10 ns under the NVT ensemble. In the NVT ensemble, N is the number of atoms, V is the system volume, and T is the temperature and are all held constant. For self-diffusion and viscosity, the reference time for calculations was at the beginning of the 10 ns isothermal hold. Mean squared displacement (MSD) for self-diffusion, viscosity, and averaged thermodynamic data were attained from the last 2 ns of the 10 ns isothermal hold. Each composition was run 2 times with different starting atomic positions for averaging characteristic temperature measurements to reduce uncertainty.

2.5.3.2 Kinetic Properties

Self-diffusion for each isothermal hold from quench runs was calculated through the long-time limit of MSD with Equation 2.2 [66]. In Equation 2.2, D is the self-diffusion, N is the total number of atoms, t is the time, and r is the position of an atom i. If the average mean squared displacement of atoms is less than 1 Å^2, we assumed that the atoms cannot

be reliably identified as diffusing rather than just vibrating in place and the associated self-diffusion value was excluded from all future analysis and our VFT fits to data.

$$D = \lim_{t\to\infty} \frac{1}{6Nt} \left\langle \sum_{i=1}^{N} \left[r_i(t) - r_i(t=0) \right]^2 \right\rangle \tag{2.2}$$

The Green-Kubo formalism was used to calculate viscosity (Equation 2.3) in an equilibrated system by integrating the autocorrelation of the pressure tensor off-diagonals [55, 66–68]. In Equation 2.3, k_B is the Boltzmann's constant, T is the temperature, V is the system volume, t_0 is the starting time, t is a time value, and $P_{ij} \in \{P_{xy}, P_{xz}, P_{yz}\}$ are the elements of the pressure tensor. For a three-dimensional simulation, the integral of the autocorrelation of P_{xy}, P_{xz}, and P_{yz} can be averaged together due to their symmetry equivalence in the liquid state.

$$\eta = \frac{V}{k_B T} \int_0^\infty \langle P_{ij}(t_0) P_{ij}(t_0 + t) \rangle \, dt \tag{2.3}$$

Although several expressions exist to fit dynamic properties for a fluid, we find that the VFT expression was the simplest to fit and use and we did not see any advantage using more complex functional forms [50]. The VFT function was used to fit the resulting self-diffusion and viscosity data (Equation 2.4). In the VFT expression, A , B, and T_0 are fitting constants. x is either self-diffusion or viscosity depending on which data was used for fitting.

$$log_{10}(x) = A + \frac{B}{T - T_0} \tag{2.4}$$

2.5.3.3 Characteristics Temperatures

Seven temperatures were calculated from MD: T_{gm}, T_g^*, T^*, T_A^*, $T_g^{'}$, $T^{'}$, and $T_A^{'}$. T_l was also considered but was taken from experimental data or the Alloy Phase Diagram Database from the American Society for Metals (ASM) International. Table 6.10 lists the system and the corresponding diagram used from ASM for T_l values. Here we describe

how each characteristic temperature was determined. T_{gm} was calculated via methods used in Refs. [69, 70] which use a change in the potential energy, E_{pot}, slope between high and low temperature regimes to determine T_{gm}. The change of slope is typically acquired through a "knee" in a heat capacity curve with respect to temperature but we did not use this approach due to issues with numerical noise. We find the change in slope as follows. We perform a fit to E_{pot} vs. temperature using three piecewise linear fits, which represent the behavior of the system in the liquid, supercooled, and glassy phases. The three fitted lines are chosen to minimize the squared residuals using the package in Ref. [71]. An example of the fitting is shown in Figure 2.6. We then determine T_{gm} as the intersection of the two lower temperature lines (the glassy and supercooled liquid lines). We note that the intersection of the two higher temperature lines generally gives a dynamical slow down temperature, T_s, as defined in Ref. [70]. We were not able to determine a robust T_s for all systems and we did not use this value as a characteristic temperature in this study. We used this approach to determine T_{gm} for all 95 compositions studied. The uncertainty in our estimate of T_{gm} is found from the standard error of the mean (SEM) across the 2 cooling runs. The average SEM for T_{gm} across multiple compositions were 19 K, which is adequately low given the other uncertainties in this overall analysis.

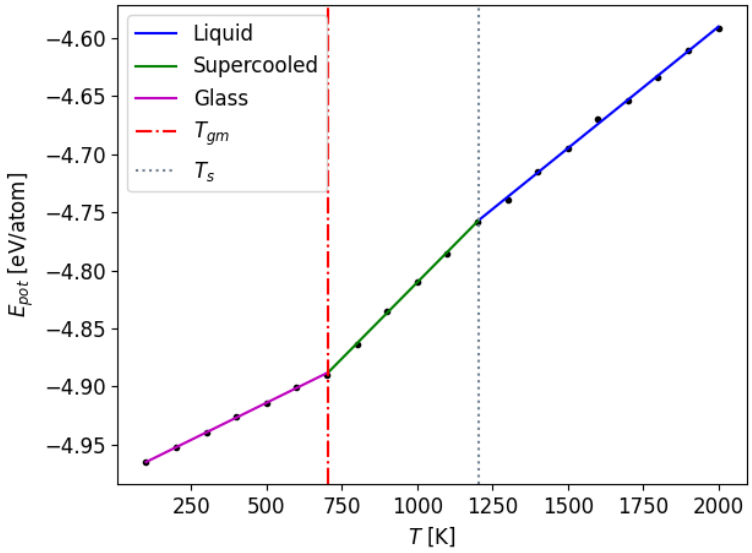


Figure 2.6: The results of a three part piecewise linear fit to the potential energy vs. temperature for $Cu_{50}Zr_{50}$. The transition from low-temperature glassy to mid-temperature supercooled liquid regimes is shown by the red vertical line which denotes the molecular dynamic glass transition temperature for a single run.

Self-diffusion and viscosity follow an Arrhenius behavior at high temperatures. At lower temperatures, cooperative motion becomes more significant leading to a deviation from Arrhenius behavior [44, 72]. A simple algorithm was applied to consistently compute $T_A^{'}$ and T_A^* across multiple compositions. First, self-diffusion and viscosity values were fit to VFT and a linear function. We then compared the two fits to see where they deviated and used that measure to determine the characteristic temperature. More specifically, we first started with all the data points. Then, if the mean absolute residuals (MAR) of the linear fit were more than the VFT fit by an amount greater than 0.005 for either self-diffusion or viscosity in their respective log_{10} units, the lowest temperature points were excluded from linear fits until the linear minus VFT MAR equaled or fell below the threshold. The computed $T_A^{'}$ and T_A^* denote the lowest temperatures where VFT and Arrhenius fits are approximately indistinguishable. The present approach provides a consistent definition for $T_A^{'}$ and T_A^* across the 95 studied compositions. The uncertainty in our estimate of these characteristic temperatures is found similarly to T_{gm} above from SEM across the 2 cooling runs. The algorithm used to compute $T_A^{'}$ and T_A^* have average SEM values of 21 K and 11 K respectively for all compositions studied. More averaging could reduce uncertainties, but we find the uncertainties sufficient for the current work. A sample calculation of $T_A^{'}$ and T_A^* are shown in Figures 2.7 and 2.8 respectively.

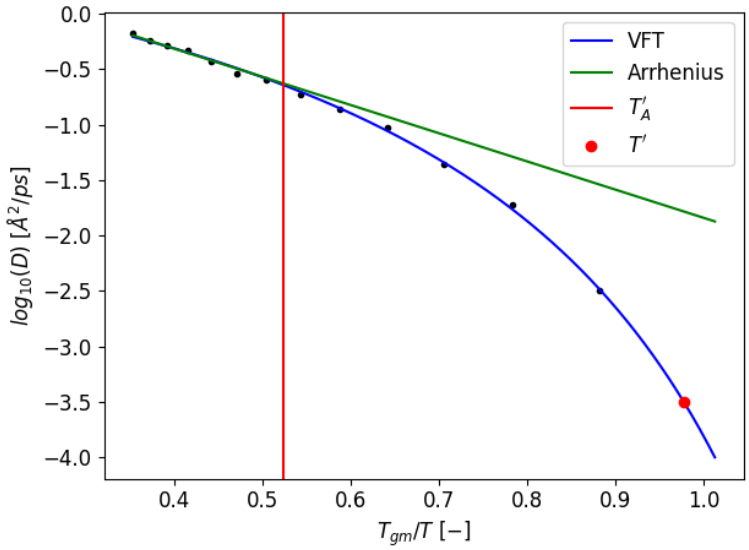


Figure 2.7: The self-diffusion behavior with respect to temperature for a single run of $Cu_{50}Zr_{50}$. The black points are the molecular dynamic self-diffusion for NVT isothermal holds. The blue and green curves are the VFT and high temperature Arrhenius fit to self-diffusion data respectively. The red point denotes the temperature at a user specified self-diffusion cutoff and the vertical line represent the temperature where self-diffusion deviates from Arrhenius behavior.

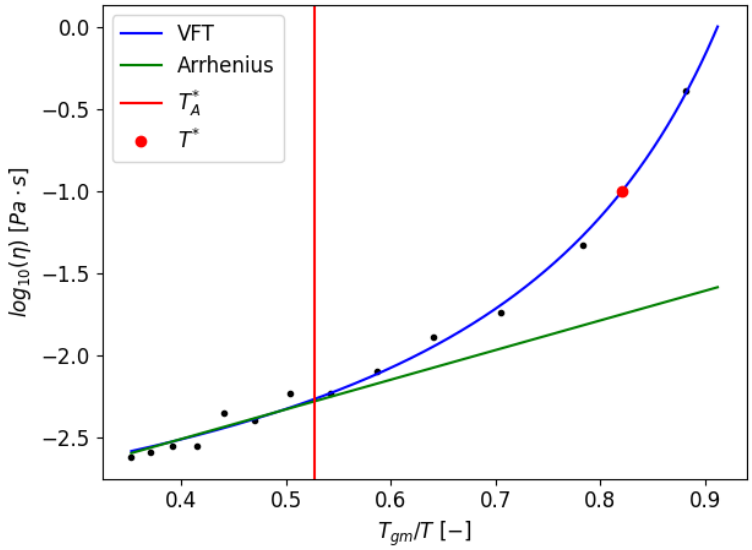


Figure 2.8: The viscosity behavior with respect to temperature for a single run of $Cu_{50}Zr_{50}$. The black points are the molecular dynamic viscosity for NVT isothermal holds. The blue and green curves are the VFT and high temperature Arrhenius fit to viscosity data respectively. The red point denotes the temperature at a user specified viscosity cutoff and the vertical line represent the temperature where viscosity deviates from Arrhenius behavior.

In a similar manner to T'_A and T^*_A calculation, T' and T^* were determined using fits to the

VFT equation of self-diffusion and viscosity data respectively. $T^{'}$ was defined here as the temperature at which each composition reached a self-diffusion value of $10^{-3.5}$ Å$/ps$ (Figure 2.7). Similarly, T^{*} was defined as the temperature where each composition reached a viscosity value of 10^{-1} $Pa \cdot s$ (Figure 2.8). The choice of cutoffs were as large as practical given a goal of being near data points acquired through MD to reduce extrapolation error from VFT. Using the same error methods for the characteristic temperatures described above gave average SEM for $T^{'}$ and T^{*} of 18 K and 9 K respectively. The values of $T_g^{'}$ and T_g^{*} were also determined from VFT fits to self-diffusion and viscosity. Again, using the same error methods, when the VFT functions were extrapolated to 10^{-12} Å$^2/ps$ and 10^{12} $Pa \cdot s$ for self-diffusion and viscosity, respectively, the average SEM for $T_g^{'}$ and T_g^{*} were 35 K and 31 K, respectively. Each CT along with its description can be seen in Table 2.5.

Table 2.5: The definitions for all CTs included in this study.

CT	Description
T_{gm}	The glass transition temperature acquired from a potential energy versus temperature relationship.
T_g^{*}	The glass transition temperature acquired by extrapolating VFT viscosity to 10^{12} $Pa \cdot s$.
T^{*}	The temperature where viscosity reaches 10^{-1} $Pa \cdot s$.
T_A^{*}	The deviation from high temperature Arrhenius behavior in a viscosity versus temperature relationship.
$T_g^{'}$	The glass transition temperature acquired by extrapolating VFT self-diffusion to 10^{-12} Å$^2/ps$.
$T^{'}$	The temperature where self-diffusion reaches $10^{-3.5}$ Å$/ps$.
$T_A^{'}$	The deviation from high temperature Arrhenius behavior in a self-diffusion versus temperature relationship.
T_l	The liquidus temperature.

2.5.3.4 Data

Experimental T_l, D_{max}, and melt-spun classification data were acquired through an online database with citations to papers included in the Data Availability section. Missing values of T_l were replaced by values read from ASM phase diagrams (see Table 6.10). Although T_l did not come from MD, previous studies show that MD calculation of T_l is possible, as discussed in the Introduction. When multiple D_{max} or T_l values were available for the same composition, the mean of the values were used. If any one of the melt-spun sheets resulted in fully or partially crystalline sheets, then that alloy was classified as a Poor GFA alloy. Fully amorphous sheets and alloys that had a D_{max} measure were classified as Good GFA alloys. As noted above, the data set containing only D_{max} data has 21 compositions, is a subset of the 95 total alloys, and is called the regression data set. The full set of 95 compositions is called the classification data and contains classes of Poor GFA (39 compositions = 41% of the data) and Good GFA (56 compositions = 59% of the data).

For our classification data, we can generate simple new features from products, ratios, summations, and differences (PRSDs) of the characteristic temperatures, which we call PRSD features. We can then compare the effect of the classification models using the PRSD features to models using our original set of characteristic temperatures as features. We are motivated to explore the PRSD features by the long history of studies using linear functions of PRSD features to predict different aspects of GFA, as discused in the Introduction. To construct the PRSD features, we follow the approach of Ref. [2]. We start with a set of characteristic temperatures defined as $A = \{T_{gm}, T_g^*, T^*, T_A^*, T_g^{'}, T^{'}, T_A^{'}, T_l\}$. From set A, summations and differences were taken between each of the features to construct set B. A sample feature contained in set B would be $(T_{gm} - T_l)$. Now define $C = A \cup B$. From C, we can take powers up to n for every element to produce set D. For instance, $(T_{gm} - T_l)^2$ is an element in set D. For our current work, we limited $n \in \{1, 2\}$. Define set E as follows: $E = C \cup D$. For every element in E, we can take the inverse to produce set F. Continuing from our example, a possible element produced would be $1/(T_{gm} - T_l)^2$. We then construct another set $G = E \cup F$. The final operation to gener-

ate features involves products between every combination of two elements from set G to produce set H. The final feature set was defined as $X = G \cup H$. This process produced a total of 32,896 features. As noted in the Results and Discussion section, PRSD features do not provide statistically significant improvements in learning GFA compared to using CTs alone. Hence, a larger space of features was not explored (e.g., $n \in \{1, 2, 3, 4, ...\}$ for power features). All data sets used for machine learning, models, and figures can be found at figshare and GitHub at Refs. [73] and [74] respectively.

2.5.3.5 Machine Learning

Scikit-learn was used for ML applications [42]. The XGBoost model type was attained from Ref. [20]. A standard scaler was used to transform all our features to have a mean of zero and a standard deviation of 1 for each dataset. When used in nested CV, the scaler is trained only on the training set and then used to transform both the training and test features.

Since the number of fitting points for our regression data are small, we fit a simple model to our characteristic temperatures of greatest importance. We use $log_{10}(D_{max})$ as the target feature. Raising D_{max} to a power within a logarithm like in Ref. [25] has no impact on the fitting ($log_{10}(D^x_{max}) = xlog_{10}(D_{max})$ and ML models can account for the multiple by a real number x) so no power is included. First, we trained a Least Absolute Shrinkage and Selection Operator (LASSO) model that minimized root mean squared error ($RMSE$) through a grid search of α hyperparameter values [75]. The α values considered were 10^{-5} to 10^{5} in a log_{10} grid of 100 values. The absolute value of weights from the fitted model denote the magnitude of the model's response with respect to a change in feature value which is a measure of the contribution each feature makes to predicting $log_{10}(D_{max})$. Using the two features with the highest weights, an OLS regression model was fit with an intercept term to produce the final model (using more features led to overfitting and reduced the CV accuracy, as might be expected given the complex physics and limited training data in the model). To assess the OLS model, 20 repeats of 3-fold CV were performed to view the effects of prediction on data outside of

the training set. Metrics reported for final regression models are the mean average error (MAE), coefficient of determination (R^2), $RMSE$, and the $RMSE/\sigma$ where σ denotes the spread on predicted target values.

We implemented repeated nested CV to assess classification models. Model types included are Gradient Boosting (GB), eXtreme Gradient Boosting (XGBoost), and Random Forest (RF) which are ensemble models [20, 76, 77]. We used ensemble models because of their tendency to outperform linear models in D_{max} predictions in our past research [2]. Models were trained on PRSD feature and original characteristic temperature feature sets. Data were shuffled and split into 3 outer and 3 inner folds. The choice of hyperparameters that minimized $RMSE$ were determined from the inner folds via a grid search (Table 2.6). The outer folds were used to assess a model's performance on data not used for training. Nested CV was performed 20 times which gave 60 test sets for each model and data set combination. Classification scores were averaged between all leave out sets. A two-sample T-test was performed between models trained on the original versus the PRSD feature sets to show if there was a statistically significant difference between training on the two feature sets.

Table 2.6: The grid of hyperparameters for XGBoost, GB, and RF models. The conventions of Scikit-learn and XGBoost were used for parameter names [20, 42].

Model	Parameter	Values
RF	n_estimators	30, 40, 50, 60, 100, 500
	max_features	sqrt, log2, None
	max_depth	2, 3, 4, None
GB	learning_rate	0.001, 0.01, 0.1, 0.2
	n_estimators	30, 40, 50, 60, 100, 500
	max_features	sqrt, log2, None
	max_depth	2, 3, 4
XGBoost	learning_rate	0.001, 0.01, 0.1, None
	max_depth	2, 3, 4, 5, None
	subsample	0.5, 0.8, 1.0, None
	gamma	0, 1, 5, None

All classification models were assessed with the area under the curve (AUC) from Precision-Recall (PR) curves and maximum $F1$ scores. $F1$ is defined as the harmonic mean between precision and recall. The baseline AUC for any PR curve is defined as $P/(P+N)$ with P being the number of positive and N being the number of negative cases [78]. For the nested CV tests, PR curves where averaged together for the outer loops. The averaging was performed by first building a grid of horizontal values (recall for PR) from 0 to 1 with 1,000 linearly spaced values. Then data are linearly interpolated. This ensured that all averaged values were gridded equally for vertical averaging. For $F1$ scores, the maximum values were averaged between all outer folds.

SHapley Additive exPlanations (SHAP) provide feature interpretations by fairly allocating feature contributions via game theory, an approach developed by Professor Lloyd Shapley [79]. SHAP values were attained with the shap package in Ref. [80] and used to analyze feature contributions for an XGBoost model trained on all 95 cases (called the full-fit model) of our classification data set. We will show that the prediction contribu-

tions of the top 3 features agree with physical intuition of GFA.

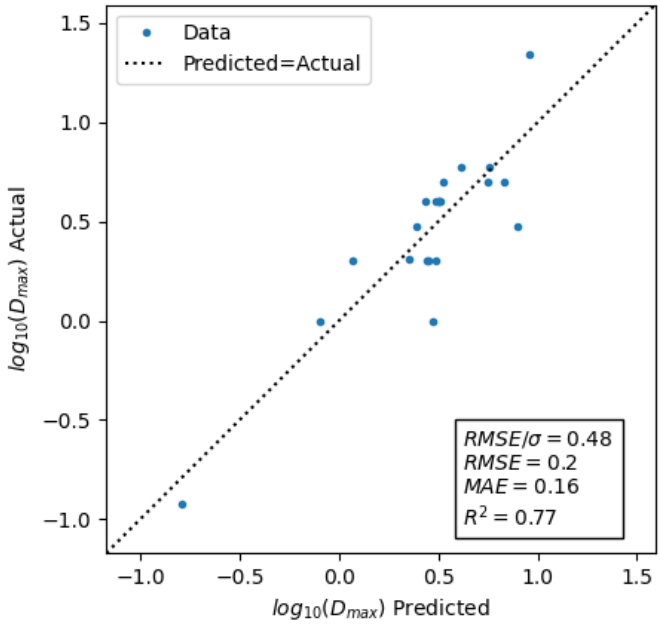
2.5.4 Results and Discussion

First, we consider the regression data set and models. The OLS regression model fit from LASSO selected features was used to predict back onto the training set to produce the parity plot in Figure 2.9a. The $RMSE/\sigma$ was 0.48 which means that the $RMSE$ of predicted $log_{10}(D_{max})$ values are well-below the spread in true values, σ. The characteristic temperatures with the highest absolute weights and therefore used in the model were T_l and $T_A^{'}$. The sign of weights for T_l and $T_A^{'}$ were negative and positive respectively which follow expected theories. Assuming the experimental T_g is similar across studied compositions, a higher T_l represents a larger range of temperatures through which the supercooled liquid must remain stable without nucleating crystalline phases before producing a metallic glass. This argument is similar in spirit to that supporting T_{rg} as correlating with GFA, as proposed by Turnbull [30]. Conversely, a higher $T_A^{'}$ denotes a dynamic slowdown in a system at a higher temperature, which suppresses the ability of atoms to arrange themselves into an ordered structure and therefore suppresses crystallization and stabilizes glass formation. Higher T_l values should therefore reduce GFA while higher values of $T_A^{'}$ should increase GFA.

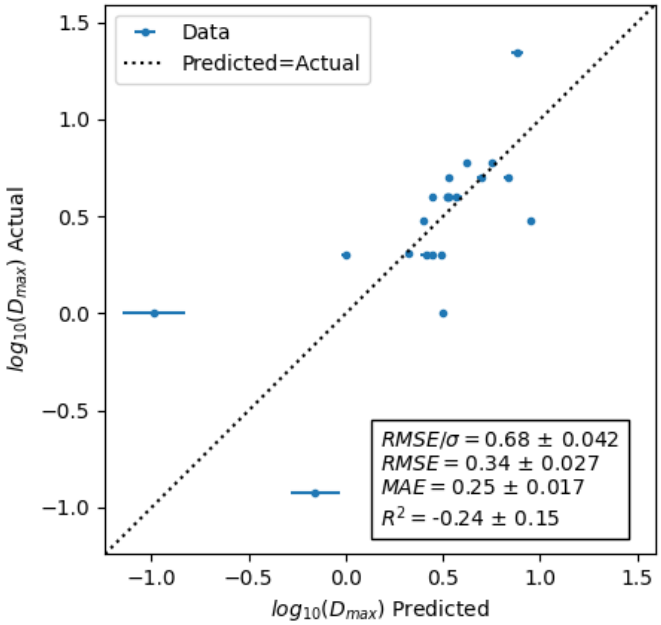
The closest model in literature to our OLS regression model was proposed in Ref. [25] as a linear combination of T_{rg} and m. Our model qualitatively agrees with the model in Ref. [25] in two ways. First, lower T_l for the OLS regression model generally results in higher T_{rg} and better GFA. Second, m denotes the viscosity of a system as it approaches experimental T_g. A larger m corresponds to less resistance to movement when cooling and vice versa which is similar to $T_A^{'}$. Although the parameters across models are different, they both describe the degree of undercooling along with the mobility of atoms with respect to GFA. The model in Ref. [25] had an R^2 score of 0.980 while we had an R^2 score of 0.77. Our scores may have been worse due to the use of different variables, but perhaps also just because of the smaller number of compositions we could study along with our mixing of MD with approximate potentials and experimental data. Some of

these aspects could be remedied in future work and potentially approach the outstanding accuracy of the model from Ref. [25] while still using MD derived features.

A more rigorous test of our OLS regression model was performed. The average of predicted values from leave-out compositions in 20 repeats of 3-fold CV along with their SEM is shown Figure 2.9. As expected, the $RMSE/\sigma$ increased, with values changing from 0.48 to 0.68 ± 0.042. The decrease in prediction performance can be explained by the generally complex dependence that might be expected for $log_{10}(D_{max})$ on the features, and in part by the lack of cases with low D_{max} values. The models fit only on high D_{max} cases will fail to predict the cases on the lower extreme. It is not surprising that fitting to few, unevenly distributed data results in a relatively inaccurate (Figure 2.9) model, even when it may have first appeared promising during the full-fit without cross validation (Figure 2.9a).



(a) Predicted Back



(b) CV Averaged

Figure 2.9: The parity plots for OLS models along with standard ML performance metrics. Each of the blue points denotes a prediction of D_{max} from a model. The black dotted line represents where ideal predictions would fall. We note a reduction in prediction ability of OLS models when a cross validation test was performed (predicted back compared to CV averaged).

Second, we consider the classification data set and models. Through nested CV with XGBoost, RF, and GB models fit on the original classification set and then the PRSD classification set, PR curves were produced as detailed in Sec. 2.5.3.5 and the scores were

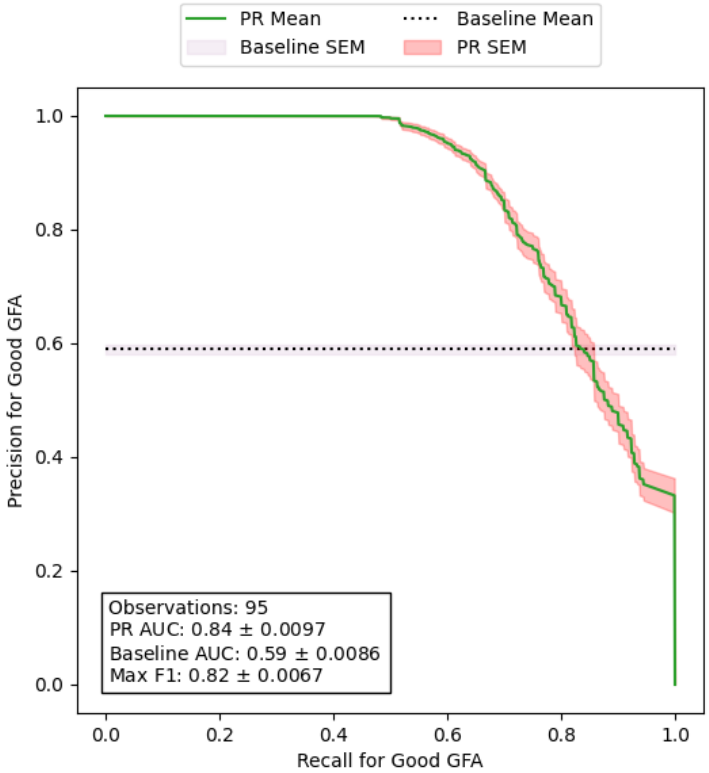
tabulated in Table 2.7. The use of XGBoost models yielded slightly better results than other models and is shown in Figure 2.10a. The AUC score for PR with Good GFA as positive was 0.84 ± 0.0097 with a baseline AUC of 0.59 ± 0.0086. The average maximum $F1$ score was 0.82 ± 0.0067. One way to understand the implications of the PR curve (Figure 2.10a) is that if one starts with a list of compositions similar to the training data, and one can accept finding just half the good GFA alloys in the list (50% recall), then one can be almost certain that the compositions one predicts as good GFAs are correctly identified (100% precision). As an example, consider searching a list of 100 compositions for GFA with 50 favorable glass formers. The XGBoost model might be expected to predict 25 of the alloys as good glass formers. Almost all 25 would likely be correct, but the model would find only 25 out of the 50 glass formers. A researcher could move along the PR curve to define an acceptable threshold to tune for the number of missed good glass formers while ensuring that a tolerable fraction of glasses studied will produce a glass.

Although the above results are encouraging, they are almost certainly optimistic when using the present model for predicting GFA of general alloys. First, one needs to consider the compositions in the data. Simulations were performed on only 17 chemical systems with a varying number of compositions in each system but 5-6 examples from each system on average. The outer folds used for model assessment therefore most likely contain compositions close to, although not exactly the same as, those used for model training. This will bias the CV scores to be much better than expected on data for totally new systems. Furthermore, new compositions outside the 17 studied here may be quite different in their underlying mechanisms, further reducing the applicability of the model. Finally, it should be noted that the present data is fairly well balanced, with about 59% of the data with good GFA. Even if the model accuracy is fairly well represented by the present PR on a new test data set, if the fraction of good GFA alloys is much lower, then the model will have much lower precision than found here. We therefore believe that the PR curve obtained here is exciting as it is evidence that GFA can be predicted from MD features based on characteristic temperatures, but it is not a robust guide for expected

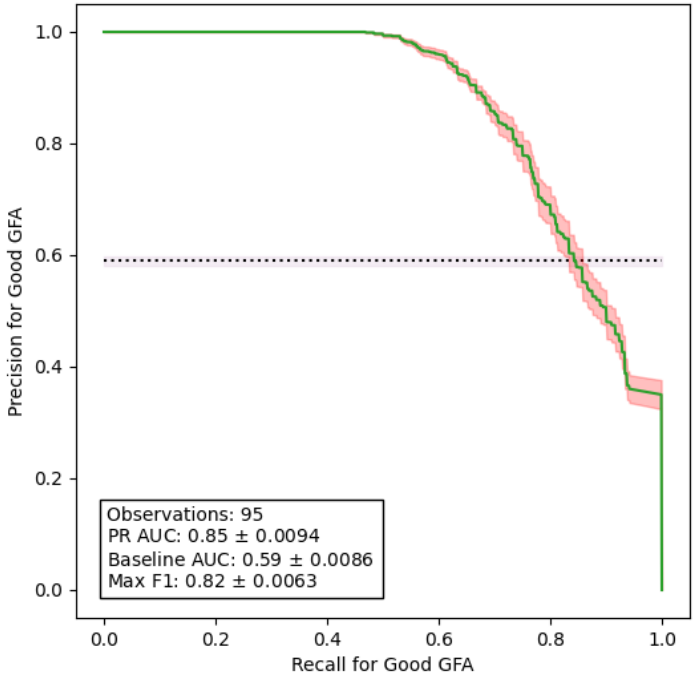
results on a general screening of alloys for GFA.

Table 2.7: The classification scores for all model types. The PRSD classification data contains generated features from the classification set as outlined in Sec. 2.5.3.5.

Model	Feature Set	Metric	Score	SEM
GB	Classification	Precision Recall AUC for Good GFA	0.82	0.0086
GB	Classification	Max F1 for Good GFA	0.80	0.0061
RF	Classification	Precision Recall AUC for Good GFA	0.84	0.0078
RF	Classification	Max F1 for Good GFA	0.81	0.0059
XGBoost	Classification	Precision Recall AUC for Good GFA	0.84	0.0097
XGBoost	Classification	Max F1 for Good GFA	0.82	0.0067
GB	PRSD Classification	Precision Recall AUC for Good GFA	0.84	0.0095
GB	PRSD Classification	Max F1 for Good GFA	0.82	0.0065
RF	PRSD Classification	Precision Recall AUC for Good GFA	0.86	0.0096
RF	PRSD Classification	Max F1 for Good GFA	0.83	0.0069
XGBoost	PRSD Classification	Precision Recall AUC for Good GFA	0.85	0.0094
XGBoost	PRSD Classification	Max F1 for Good GFA	0.82	0.0063



(a) Original Classification Set



(b) PRSD Classification Set

Figure 2.10: The comparison between XGBoost models trained on the original versus PRSD classification data sets. The green curve represents the precision and recall given a classification threshold averaged across outer fold test sets. The shaded red area is the SEM between averaged sets. The horizontal line is the average baseline from class counts from outer fold test sets along with the purple shaded region SEM.

The use of the PRSD features to fit models (Figure 2.10b) showed some improvement over the original classification feature set. Although initially promising, p-values computed from a two-sample T-test show that there was no statistically significant difference between learning from PRSD and original classification feature sets from 60 test observations. None of the p-values (Table 2.8) fell below 0.05 which is a commonly used cutoff for statistical significance. Because of the lesser complexity of the models fit with the original characteristic temperature feature set, the final evaluation metrics of XGBoost without PRSD features were reported.

Table 2.8: The p-values from two-sample T-tests.

Model	Metric	P-Value
GB	Precision Recall AUC for Good GFA	0.13
GB	Max F1 for Good GFA	0.22
RF	Precision Recall AUC for Good GFA	0.22
RF	Max F1 for Good GFA	0.21
XGBoost	Precision Recall AUC for Good GFA	0.59
XGBoost	Max F1 for Good GFA	0.51

The SHAP values for our full-fit XGBoost model are shown on Figure 2.11. The features that have the highest impact on the final prediction are at the top. Conversely, the bottom features have the lowest contribution on final predictions. Feature values to the right of the vertical line in Figure 2.11 push the final prediction to Good GFA while values to the left contribute to Poor GFA classification. The color of values denotes the scale of the feature values. The SHAP values are generally consistent with physical intuition, as can be seen by considering the trends of T^*, T_g^*, and T_l. The SHAP values show that higher values of T^* and T_g^* and lower values of T_l generally correlate with better GFA. Materials that have higher values of T^* experience higher viscosities at higher temperatures. These higher viscosities may slow down the kinetics of crystallization when cooling a molten alloy, supporting better GFA. However, lower values of T_l might decrease the range an alloy must cool through before vitrification in a time temperature transformation (TTT)

diagram [21], also supporting better GFA. Similarly, higher values of T_g^* could narrow the amorphous cooling range on a TTT diagram, again supporting better GFA. These correlations between the SHAP values and these physically sound trends suggest that the model has captured some of the underlying physics behind GFA.

Figure 2.11: The SHAP values for our 8 characteristic temperatures for an XGBoost model. SHAP values from the figure denote the impact from each feature on the prediction of the model. Highest ranked features are displayed from the top to the bottom of the visual.

2.5.5 Conclusion

We used MD to calculate a set of characteristic temperatures for 95 metal alloys which were used to fit GFA for cases containing D_{max} and melt-spinning data. When using $log_{10}(D_{max})$ as our target, we find a 3-fold cross-validation $RMSE$ score of 0.34 ± 0.027 for an OLS regression model, which was not significantly below the standard deviation of 0.43 for the 21 training cases. Only T_l and T_A' were used in the OLS model. Through nested CV, we assessed the capacity to learn poor from good glass formers for several models. XGBoost was slightly better than all other classification model types. Our average maximum $F1$ score for our RF classification predictions was 0.82 ± 0.0067. Additionally, the AUC for classifying Good GFA on our PR curve was 0.84 ± 0.0097 which was greater than the baseline of 0.59 ± 0.0086 by 0.25. Our XGBoost models predict significantly higher than random guessing. We also determined that learning from PRSD

features had no statistically significant effect for any classification tasks compared to learning on the characteristic temperatures themselves. Classification scores suggest that characteristic temperatures from MD can be used as features in machine learning models with predictive ability for GFA. This result provides a potential pathway to discovering new metallic glass alloys based on only simulations. However, to support such screening it is necessary to develop a larger training database that can train more quantitative models with larger ranges of chemistry in their domains of applicability.

2.5.6 Data Availability

The raw and processed data required to reproduce these findings are available to download from figshare at `https://doi.org/10.6084/m9.figshare.14502135.v1` and GitHub at `https://github.com/leschultz/Molecular-Dynamic-Characteristic-Temperatures-for-Predicting-Metallic-Glass-Forming-Ability`.

2.5.7 Impact from Work

- This work used molecular dynamics simulations to calculate characteristic temperatures for metal alloys and fit machine learning models to predict glass forming ability, demonstrating the potential of using simulated properties to predict experimental measures of glass forming ability.
- While the regression models for predicting critical casting thickness showed only modest correlation, the binary classification models distinguishing poor and good glass formers achieved promising performance, highlighting the ability to screen alloys computationally to identify promising glass formers for experimental validation.

2.6 "Machine Learning Metallic Glass Critical Cooling Rates Through Elemental and Molecular Simulation Based Featurization"

Note: This paper is in preparation for submission to the Journal of Materiomics and has been adapted for use in this thesis.

2.6.1 Abstract

We have developed a machine learning model for critical cooling rates for metallic glasses based on featurization using elemental properties and molecular simulations. We compare results for features derived from easy-to-compute functions of elemental properties to more complex physically motivated properties using ab initio, machine-learning potential, and empirical potential molecular dynamics methods. We use the flexibility of machine-learning potentials to investigate previously computationally inaccessible material properties, like viscosity, across a diverse range of alloys. Analysis of various features for 34 alloys from 20 chemical systems shows that the best model for critical cooling rates was learned from one elemental property-based feature and three simulated features. The elemental property-based feature is a simple ideal entropy value based on alloy stoichiometry. The simulated features were acquired from estimates of energies above the convex hull, changes in heat capacity above and below the fictive temperature, and the fraction of icosahedra-like Voronoi polyhedra. Models were assessed through a demanding cross validation test based on repeatedly leaving out full chemical systems as test sets and had an R^2 of 0.78 and a mean average error of 0.76 in units of $[log_{10}(K/s)]$. We demonstrate with Shapley additive explanation analysis that the most impactful features have physically reasonable influence on the critical cooling rate. This methodology can be applied to other high-throughput studies of material properties of diverse compositions.

2.6.2 Introduction

Unlike their crystalline counterparts, metallic glasses are a type of metallic material characterized by a lack of long-range atomic order. Metallic glasses can have exceptional hardness, increased strength, superior corrosion resistance, smaller magnetic hysteresis, and higher resistivity when compared to their crystalline counterparts. These properties have led to a range of applications, including structural supports within the human body [27, 28], voltage transformers [29], and golf clubs [81]. Each of these applications depend on the composition of the material in question and the ability for that composition to form a glass instead of a crystal during processing.

One significant obstacle to the more widespread adoption of metallic glasses for engineering applications is their size limitation, which is primarily due to the greater thermodynamic stability of their crystalline counterparts. When metallic glasses are manufactured through melt-quench methods, heat must be quickly dissipated from samples or else crystalline phases are formed, which limits the maximum size the quenched samples can reach and remain a glass. The slowest rate at which a molten material can be cooled to produce a glass is called the critical cooling rate (denoted by R_c) and is an intrinsic measure of glass forming ability (GFA). The size of a glass that can be produced is closely related to R_c. This size is typically quantified by the maximum potential length of the shortest sample section of a produced glass before nucleation of crystalline phases occurs (often represented as Z_{max}). If the geometry is cylindrical, then the diameter, D_{max}, can be referenced specifically. Any length above Z_{max} will lead to the emergence of crystalline phases. Z_{max} and D_{max} are also widely used as measures of GFA, although they can depend on the quenching method and are not as intrinsic to the material as R_c.

Many previous efforts to quantify GFA relied on arithmetic operations of the glass transition (T_g), crystallization (T_x), and the liquidus (T_l) temperatures [2, 24]. A significant limitation of these methods is the prerequisite formation of a glass to measure T_g and T_x. Nonetheless, models derived from these temperatures can offer valuable insights into the potential size of metallic glass samples, once they are established as glass formers.

The rheology of a material, typically characterized by viscosity or relaxation time, is suggested by other studies to play a pivotal role in explaining GFA [25, 43, 44]. Relaxation time and viscosity are frequently used interchangeably because they are directly proportional to each other [82]. As a material cools from a molten state and transitions into a glass, atoms with increasingly restrained movement, reflected by higher viscosity at lower temperatures, tend to remain amorphous. The change in viscosity (or relaxation time) with respect to temperature when the material is near T_g is called fragility, m, [56]. The conventional understanding is that high m indicates poor glass formers, while low m suggests favorable glass formers. There are alternate properties that correlate with m, as highlighted in the work by Gangopadhyay et al. [53]. Two noteworthy properties include a cross-over from Arrhenius behavior temperature during cooling to non-Arrhenius behavior, T_a, and the temperature at which materials reach a chosen viscosity cutoff, T_c, as described in Ref. [53]. Ratios of the aforementioned temperatures and T_g were shown to have significant correlations with m. However, m alone does not describe GFA. As discussed in Ref. [83], m sometimes exhibits the opposite relationship with GFA than anticipated. In the studied $MgCuY$ system, many compositions displayed high GFA despite having high m values. Additionally, Ref. [25] uses T_g, T_l, and m to predict D_{max} because none of these properties cannot robustly predict D_{max} alone. Together, however, these properties produced a model for $log_{10}(D^2_{max})$ with a coefficient of determination of $R^2 = 0.980$. While these correlations are often extremely good, any correlation based on experimentally measured values has a severe limitation of needing to make and characterize the material to predict R_c. Several efforts have been made to correlate GFA to descriptors that can be trivially computed from properties of elements or properties that can be computed from molecular simulations.

Among these properties are those based on pre-tabulated values of simple alloying properties. These properties are nearly instantaneous to evaluate and are very easy to use. Some authors have combined experimentally produced physical descriptors with easy to compute elemental properties (i.e., properties built from chemical information of each element). In Ref. [84], the experimental descriptors of T_g, T_x, and T_l were combined with

elemental features to produce a model for D_{max} with a score of $R^2 = 0.763$. While this model shows promise, the use of experimental characteristic temperatures introduces the limitations noted above. More general models that use only composition as features to predict GFA have been produced. A notable example is from Ward et al. [85] who built a D_{max} model with a correlation coefficient (not R^2) of $R = 0.89$ when assessing their model through 10-fold cross validation (CV). Their mean average error (MAE) was 0.21 mm. However, 10-fold CV produces an overly confident assessment of model performance due to similar compositions appearing in both the training and test sets. When the authors assess their model by iteratively excluding binary systems from training, the MAE rises to 0.81 mm for tests (a 286% increase). A MAE of 0.81 mm accounts for 68% of the mean absolute deviation of test sets, so there is considerable room for improvement. Afflerbach et al., combined experimental data containing Z_{max}, D_{max}, R_c, T_g, T_x, T_l, and melt-spun ribbon information to produce approximate R_c values (denoted as R_a here) for 2,125 materials [3]. When manufactured, ribbons can form fully amorphous, partially amorphous, or fully crystalline samples. The phases present in a ribbon are indicative of the quality of the GFA. The authors assigned R_c values of $10^{5.5}$ and 10^7 $[K/s]$ for partially amorphous and fully crystalline ribbons, respectively. However, fully amorphous sheets were not included in their analysis. Through systematically excluding complete chemical systems as test sets (i.e., materials were grouped by chemical systems and iteratively left out as the test set), they were able to produce random forest regression models of $log_{10}(R_a)$ with a MAE of 0.82 $[log_{10}(K/s)]$. A lower MAE of 0.36 $[log_{10}(K/s)]$ was obtained from a 5-fold CV assessment because it is a less demanding test. When applying 5-fold CV, many of the test compositions may be similar to those included in the training set (e.g., $Zr_{49}Cu_{51}$ in training and $Zr_{50}Cu_{50}$ in test). The features used to build the model were from elemental properties generated with the Materials Simulation Toolkit for Machine Learning (MAST-ML) [86]. The studies in Refs. [85] and [3] highlight the challenge of GFA features being applicable across chemical systems, as model predictions deteriorate markedly for compositions significantly different from those used in training. As demonstrated in Ref. [87], models constructed using physically motivated

features exhibit greater extrapolation capabilities compared to models built with features acquired through numerical brute force. The authors examined models trained separately on compositions, elemental features, and three physically motivated features to predict whether a composition is likely to produce a large metallic glass. When assessed through 10-fold CV, all models showed similar prediction ability. When assessed with chemistries more different from those used for training models, they find that models built from physically motivated features correctly classified 20 times more materials of interest than models built from elemental features. Models built from compositions alone identified none. This result leads us to heavily emphasize the physical relevance of features used to build models in this present work.

The computational approaches mentioned previously either rely on pre-existing or easy to generate data for constructing models. More sophisticated computational methods exist for extracting physical properties from materials and present an opportunity to enhance GFA prediction. From a structural standpoint, exploring the short-range order of glasses through Voronoi polyhedra (VPs) could provide valuable insights into their GFA. To construct VPs bisecting planes are positioned between an atom and its nearest neighbors. The type of VP is determined by the number of faces and their corresponding edges that enclose the atom. The work in Ref. [88] argues that the variance of VPs from a high temperature liquid (at 1,478 $[K]$ and above T_l) can be used as an indicator of GFA. Follow up work in Ref. [89] uses clustering techniques on VPs. Studies have also focused on VPs resembling icosahedra (ICO-like, characterized by having at least 10 faces with 5 edges each [90]), demonstrating that ICO-like VPs exhibit slow movement during cooling, consequently enhancing GFA [90, 91]. However, none of these works were assessed on a large database of GFA for their ability to provide accurate regression or classification predictions.

Structural features in the form of VPs, among many others, were studied in Ref. [4] in a regression task. Molecular dynamics (MD) was used to measure R_c by studying crystallization rates, and a set of simulated material properties was used to predict $log_{10}(R_c)$. Elemental features were also included from the elemental properties gener-

ated with MAST-ML. While differences exist between experimentally acquired R_c and its simulated counterpart (in particular, the simulated values were limited by computational constraints to above 10^{11} $[K/s]$), the results appear promising given that the model validation yielded an R^2 of 0.769. Features from VPs, rheology, and elemental properties were found to significantly impact the model. In a similar spirit to Ref. [4], characteristic temperatures obtained through rheology, temperature, energies, and phase diagrams were used to construct regression and classification models within a limited composition space [1]. Regression models yielded unsatisfactory results for D_{max} prediction, while classification metrics for high or low D_{max} were more promising. Both Ref. [4] and [1] relied on classical interatomic potentials, which are difficult and time-consuming to construct. The need to have acceptably accurate potentials available for each system being studied greatly limited the chemical space accessible to the studies. Ab initio methods can be used for many chemical compositions, but several of the studied properties were impractical to obtain through ab initio methods because of the computational expense associated with the necessary system sizes and simulation time scales. In this work, we take advantage of recent developments in machine learning potentials (MLPs), which allow for the rapid construction of accurate potentials. These MLPs were used to study 34 systems with diverse chemical compositions. Employing MLPs constructed from ab initio data with machine learning can enable practical simulations at the necessary length and time with near ab initio accuracy [92, 93].

Here, we integrate the best practices of previous efforts to quantify GFA and use them to guide developing descriptors from material properties that are computationally accessible. We extend the previous chemical domain of properties that can be obtained with MLPs. Then we evaluate the ability of these descriptors to predict R_c. The properties we use include those that are trivial to compute but also ones derived from simulations of rheological behavior, energy differences across phases, and many other physically motivated aspects. We obtain encouraging results, with a cross validation test based on repeatedly leaving out full chemical systems yielding an R^2 of 0.78 and MAE of 0.76 in units of $[log_{10}(K/s)]$. We use SHapley Additive exPlanations (SHAP) values to extract

correlations of R_c with physically motivated features and show that these correlations align with physical expectations. SHAP values quantify the contribution of each feature to a machine learning model's prediction, based on game theory principles [80].

2.6.3 Methods

2.6.3.1 Software

The MLPs in this study were constructed using the Machine-Learning Interatomic Potentials 2 (MLIP-2) package, which fits a moment tensor potential (MTP) [93]. For all other machine learning tasks, MAST-ML was employed along with scikit-learn [42, 86]. Ab initio calculations were carried out using the Vienna Ab initio Simulation Package (VASP) [94] and classical MD simulations utilized the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) [95]. OVITO was used for the calculation of VP types through its python application program interface (API) [96].

2.6.3.2 Database

Two sources of data for R_c were used in this study. One dataset contained 299 entries (where some are duplicates) gathered with the assistance of large language models [97, 98]. Another set of data was taken from a metallic glass database containing R_c and other properties from Ref. [99]. These last data were curated by researchers sifting through various publications manually. If duplicates existed across both sources of data, then the entry from Ref. [99] was kept and the other was discarded. For duplicate compositions within each dataset, the mean R_c was taken to produce a one-to-one mapping between composition and R_c. The merged data set contained 177 entries. For the portion of the study that includes fitting MLPs and computationally expensive simulations, only 34 out of the 177 entries were studied. The subset of 34 compositions was acquired from Ref. [99], and none of them originated from the data obtained using large language models.

2.6.3.3 Summary of Explored Properties

The properties studied in this work include those acquired using MLPs and those gathered from elemental properties. Due to the extensive number of features and potential for confusion in feature nature and naming conventions, we present a comprehensive summary of features and their corresponding descriptions in Table 2.9. Secs. 2.6.3.6, 2.6.3.7, and 2.6.3.8 will elaborate on the acquisition of features and which set of features and their connection to the different feature sets (i.e., X_i where $i \in \{long, mtp, crys, mastml\}$).

Table 2.9: The features included in this study are tabulated below. Formulas are provided for features that are further discussed from other sources.

Feature	Set	Description
$\bar{x}$	X_{long}	Sum of the multiple of Pauling electronegativity and element fractions [84]
γ	X_{long}	A function of atomic radii [84]
H_{mix}	X_{long}	A function of molar mixing enthalpy [84]
δ	X_{long}	A function of atomic radii and element fractions [84]
$\bar{r}$	X_{long}	Sum of the multiple of atomic radii and element fractions [84]
S_{mix}	X_{long}	$-R\sum_{i=1}^{N} c_i ln(c_i)$, where c is the atomic fraction, R is the gas constant, and i is the element in question [84]
ΔC	X_{mtp}	Difference in dE/dT between a frozen glass and liquid
T_f	X_{mtp}	The fictive temperature
T_s	X_{mtp}	High temperature dynamical transition
T_{a_diff}	X_{mtp}	The cross-over temperature denoting the start of non-Arrhenius behavior for self-diffusion
T_{a_visc}	X_{mtp}	The cross-over temperature denoting the start of non-Arrhenius behavior for viscosity
T_{g_diff}	X_{mtp}	The temperature where a material reaches a self-diffusion of 10^{-12} [Å/ps]
T_{g_visc}	X_{mtp}	The temperature where a material reaches a viscosity of 10^{12} [Pa/s]
T_{c_diff}	X_{mtp}	The temperature where a material reaches a self-diffusion of 10^{-6} [Å/ps]
T_{c_visc}	X_{mtp}	The temperature where a material reaches a viscosity of 10^{4} [Pa/s]
D_a	X_{mtp}	The MYEGA fitting parameter a in Eq. 2.8 for self-diffusion
D_b	X_{mtp}	The MYEGA fitting parameter b in Eq. 2.8 for self-diffusion
D_c	X_{mtp}	The MYEGA fitting parameter c in Eq. 2.8 for self-diffusion
η_a	X_{mtp}	The MYEGA fitting parameter a in Eq. 2.8 for viscosity

η_b	X_{mtp}	The MYEGA fitting parameter b in Eq. 2.8 for viscosity
η_c	X_{mtp}	The MYEGA fitting parameter c in Eq. 2.8 for viscosity
D_m	X_{mtp}	The MD fragility index for self-diffusion
η_m	X_{mtp}	The MD fragility index for viscosity
ICO-like	X_{mtp}	The fraction of Voronoi polyhedrons where number of faces with 5 edges are greater than 9 at 1500 $[K]$ [90]
$var(VP)$	X_{mtp}	The variance of Voronoi polyhedrons at 1500 $[K]$ [88]
E_{form}	X_{crys}	The formation energy for the amorphous phase
E_{above}	X_{crys}	The energy above the convex hull
Many	X_{mastml}	Algebraic operations of elemental properties (not explored in feature selection)
Many	X_{long}	Features from Ref. [84]
Many	X_{mtp}	Features derived from simulation with MTPs only
Many	X_{crys}	Features derived from simulation with MTPs and ab initio
Many	X_{tot}	$X_{long} \cup X_{mtp} \cup X_{crys}$
Many	X_{best}	$X_{best} \subset X_{tot}$

2.6.3.4 Creation and Validation of MLP Fitting Methodology

Prior to calculating computationally expensive properties for learning R_c, MLPs were fit and the approach carefully validated. Our properties of interest stem from the relationships between viscosity, self-diffusion, and potential energies with respect to temperature. Any MLP must have almost the same behavior across these properties when compared to first principles MD for its training to be considered accurate. For validation of our approach, we needed a ground truth system for which all properties could be robustly determined. This ruled out validating on ab initio data since some key properties, like viscosity, were not practical to compute fully with ab initio. Therefore, we fit MLPs to data from embedded atom method (EAM) potentials and compared the MLP predictions to those from EAM potentials [60–62, 65, 70, 100, 101]. Specifically, the potential for the systems of $AlCuZr$, $PdSi$, and NiP are from Refs. [60], [100, 102], and [70], respectively. While the EAM potentials may not be highly accurate vs. experiments, they can still be used as a physically realistic materials system for validation. We assessed our fitting of MLPs using atomic positions, energies, forces, and stresses (i.e., the set of which form one configuration) generated through classical interatomic potentials for three materials:

$Ni_{80}P_{20}$, $Pd_{75}Si_{25}$, and $Al_{10}Cu_{40}Zr_{50}$.

For producing data to train potentials, LAMMPS was used to simulate the aforementioned compositions. The data generation process was chosen so that it was practical for ab initio molecular dynamics (MD) and therefore includes fairly short MD simulation times. Materials were created randomly in a cubic configuration, then melted at 3,000 $[K]$ for 1 $[ps]$. Then, materials were held for another 1 $[ps]$ at 3,000 $[K]$ and iteratively cooled by 300 $[K]$ with 1 $[ps]$ holds until a final temperature of 300 $[K]$ was reached. At the beginning of each hold, 61 configurations with volumes expanded and contracted by $\pm$ 15% the cubic side length were taken. We conducted these expansions and contractions to ensure that the training data for potentials encompassed a broad range of energies and forces. While our choices regarding the number of configurations and the extent of contractions and expansions may not be optimal, we find them sufficient for our work because they yield excellent agreement for properties of interest. The 61 configurations making out the equation of state might seem particularly excessive but we found that just using a modest number like five configurations significantly degraded the results. All simulation was done with a time step of 1 $[fs]$. Energies, forces, and stresses were taken from all the volumetric contraction/expansion data along with every 100th frame from isothermal holds to produce the data used to fit level 08 MTPs (the level 08 and other levels are described in Ref. [93]). The final number of MD frames used for training was 724. All configurations used for building MLPs were from amorphous phases (liquid and solid). None of the potentials have crystalline information used in the fitting. Constructed MTPs were validated with separate MD. We performed a new set of melt quench runs following the approach previously discussed but without the volume compression/expansion grid using EAM potentials. For each set of atomic positions, we used MTPs to predict the energies and forces. We then compared them directly to values from classical EAM potentials. All simulation described so far in this section included system sizes of 100 atoms in the NPT ensemble.

Subsequently, MD simulations were conducted for both the classical potentials and MLPs to collect data on potential energy, self-diffusion, and viscosity as functions of tempera-

ture (methodology covered in Sec. 2.6.3.6). While MLPs are employed to acquire other properties in our study, we find that acquiring these properties was particularly challenging, and their validation serves as an effective check. Trends for these properties were compared between the MLP and EAM potentials for simulated systems of 1,000 atoms. The comparison was done with values of potential energy, self-diffusion, and viscosity averaged over 10 independent runs separately for each potential to assume robust converged values. We show in Sec. 2.6.4.1 that potential energy, self-diffusion, and viscosity were modeled accurately with the MLP vs. EAM potentials and therefore apply a similar methodology to MLPs fit with ab initio data in Sec. 2.6.3.5.

2.6.3.5 Applying MLP Fitting Methodology to Materials of Interest

We selected 34 compositions from 20 different chemical systems for fitting MLPs. The number of explored compositions was decreased from 177 because of the computational cost of fitting potentials and simulation of properties. The R_c for the chosen compositions span eight orders of magnitude and exhibit variations in the number of constituent element types. The smallest systems studied comprised two elements, while the largest system investigated involved six elements. The total number of chemical species considered in this study amounted to 19 and formed 20 chemical systems (with some chemical systems being subsets of others). Training data for MLPs were created with the use of ab initio MD in VASP. We employed a 1x1x1 Γ mesh for the k-points in our calculations. The energy cutoffs were determined by selecting the largest value of ENMAX defined in the POTCAR file for each composition (the default for VASP). Ab initio calculations were conducted for the 34 compositions following as closely as possible to the methods described for the EAM and MLP comparisons (Sec. 2.6.3.4). The only notable difference between the ab initio and EAM training approaches was a switch from the NPT to the NVT ensemble. The Nose-Hoover thermostat was used for NVT. The only thermostat compatible with NPT simulation in VASP is the Langevin thermostat, which requires defining friction coefficients for each atomic species. However, determining these coefficients for all atomic species across the 34 studied compositions was impractical for

generalized approaches. For holds under the NVT ensemble, the grid of volumetric contractions/expansions were used to find the zero pressure volume that was used for each respective isothermal hold. Some MLPs trained on the ab initio data obtained in this manner were unstable, possibly because of the change in ensemble from the previous validation on EAM potentials to the approach used in fitting with ab initio. NPT allows for many more perturbations to volumes in the system and may provide more information for potentials. The issue of MLP instability was solved with the use of active learning. NPT runs described in Sec. 2.6.3.4 with isothermal holds of 100 $[ps]$ instead of 1 $[ps]$ were simulated using the MLIP-2 interface with LAMMPS. Whenever a configuration of atoms was considered uncertain based on the default criterion outlined in Ref. [93], an additional ab initio calculation was carried out on that configuration. The information obtained from these supplementary ab initio calculations was subsequently incorporated into the training data. MD was repeated until potentials were stable. Because of this process, differing number of training configurations were used for each composition. Additional data were acquired to train MLPs compared to the method outlined in Sec. 2.6.3.4, so we anticipate improvements in behavior across properties. Errors in forces and energies were acquired in the same manner as in Sec. 2.6.3.4. The fitting errors are described in Sec. 2.6.4.1. All MD in this study was conducted with a time step of 1 $[fs]$.

2.6.3.6 Features Generated from MLP MD

This section describes how we extracted properties from simulations of 34 compositions to be used to machine learn R_c. In Sec. 2.6.2, we discussed the significance of viscosity and characteristic temperatures in prior GFA studies. As self-diffusion data can be obtained from the same MD simulations used for measuring viscosity and signifies the mobility of atoms within a system, we incorporate it into our set of properties of interest. For MD in this section, viscosity, self-diffusion, energies, etc. were acquired from the average of 5 independent simulations for each of the 34 studied materials. Each composition of interest underwent an initial equilibration at 2,000 $[K]$ for 100 $[ps]$ with a system size of 1,000 atoms. Subsequently, materials were iteratively quenched by decreasing

the temperature by 50 $[K]$ instantaneous decrements and holding for 110 $[ps]$ until a final temperature of 100 $[K]$. The first 10 $[ps]$ were used for equilibration and were discarded from any data analysis. Each isothermal hold was continued from the previous hold. The average cooling rate was approximately 0.45 $[K/ps]$. As an example of one iteration, an isothermal hold at 1,500 $[K]$ would decrease instantly to 1,450 $[K]$ after 110 $[ps]$. All these processes were conducted under the NPT ensemble and the potential energy was measured for each temperature. A Nose-Hoover thermostat and barostat were used for the NPT ensemble. The average potential energy versus temperature were used to measure the fictive temperature, T_f, and a dynamic slowdown temperature, T_s [70]. T_s indicates the onset of non-Arrhenius relaxation for a liquid material [70]. Both T_f and T_s were determined by fitting the data with three optimal linear segments using the PieceWise Linear Functions (PWLF) package [71]. Essentially, the two most drastic changes in the slope of potential energy with respect to temperature denote T_f and T_s, with T_f occurring at a lower temperature than T_s. The difference between slopes of total energies with respect to temperature above and below T_f effectively gives change in the heat capacity, ΔC. ΔC was used as a feature for learning R_c. This feature was included based on arguments made in Ref. [103], which relate change in heat capacity to m with random first order transition theory. Generally, an increase in ΔC results in a higher value of m and vice versa (See Eq. 43 or Fig. 3b in Ref. [103]).

We included two features regarding VPs by measuring atom positions at 1,500 $[K]$ for each composition. Note that the atom positions at 1,500 $[K]$ were near the temperature of 1,478 $[K]$ used in Ref [88] and is above all T_f explored in our study. These positions were determined at the final step of the NPT quenching process at 1,500 $[K]$ specified at the beginning of this section. We measure the fraction of ICO-like Voronoi polyhedra (VP) and the variance of VP as defined in Ref.s [90] and [88], respectively. ICO-like VP are VP whose number of faces with 5 edges are greater than 9. Variance of VP (denoted as $var(VP)$) is defined by

$$var(VP) = \frac{1}{G}\sum_{i}^{G}(f_i - \mu)^2 \tag{2.5}$$

where i is the type of VP, f is the fraction of i in the system, G is the number of VP types, and μ is the average of all f_i. The python3 API from OVITO was used to calculate VP with an edge threshold of 0.1 [Å] [96]. These are not necessarily the optimal types of VPs, temperatures to explore VPs, etc. for predicting GFA. However, such detailed exploration would involve an impractical level of effort given all the features being modeled and was therefore outside the scope of this work.

For each of the explored temperatures above $1.1T_f$, simulations were extended for 10 $[ns]$ under the NVT ensemble and were used to measure self-diffusion and viscosity. We explored temperatures near T_f but above because atoms are sufficiently mobile to acquire converged measurements of self-diffusion and viscosity. Viscosity can be determined using the Green-Kubo formalism [55, 66]. The autocorrelation of pressure tensor components ($P_{ij} \in \{P_{xy}, P_{xz}, P_{yz}\}$) under the NVT ensemble was integrated, following Eq. 2.3. Here, η is the viscosity, V represents the system volume, T is the temperature, k_B denotes Boltzmann's constant, and t represents a specific time value. The integrals of P_{xy}, P_{xz}, and P_{yz} were averaged together.

$$\eta = \frac{V}{k_B T}\int_0^\infty \langle P_{ij}(t_0)P_{ij}(t_0+t)\rangle \, dt \tag{2.6}$$

To measure self-diffusion, the long-time limit of mean squared displacement (MSD) was taken (Eq. 2.7) [66]. In Eq. 2.7, D is self-diffusion, N is the number of atoms, t is the time, r is the position, and i indicates the atom in question. We measured self-diffusion because it describes how rapidly atoms move in a system and is acquired from the same simulations needed to compute viscosity.

$$D = \lim_{t\to\infty}\frac{1}{6Nt}\left\langle \sum_{i=1}^{N}\left[r_i(t) - r_i(t=0)\right]^2 \right\rangle \tag{2.7}$$

Both high temperature self-diffusion and viscosity, as functions of temperature, were fitted using the MYEGA equation [104, 105]. The MYEGA function is a three parameter model that has superior extrapolation properties compared to the more commonly used Vogel-Fulcher-Tammann (VFT) equation. In Eq. 2.8, ν_a, ν_b, and ν_c are fitting parameters to map a temperature, T, to a given property, ν. $\nu = D$ for discussions on self-diffusion and $\nu = \eta$ for discussions on viscosity.

$$\log_{10}(\nu) = \nu_a + \frac{\nu_b}{T} e^{\nu_c/T} \tag{2.8}$$

From the rheological data, we derived various features. The fitting parameters from the MYEGA function for self-diffusion and viscosity for each composition were obtained. Subsequently, we employed MYEGA to extrapolate self-diffusion values to 10^{-12} [Å/ps] and viscosity to 10^{12} [$Pa \cdot s$]. The temperatures corresponding to these extrapolated values were designated as the glass transition temperatures (T_{g_visc} and T_{g_diff} for viscosity and self-diffusion, respectively), based on experimental results and approaches. In particular, the glass transition temperature is determined experimentally when a material attains a viscosity of 10^{12} [$Pa \cdot s$] [56]. Regarding diffusion, we utilized the self-diffusion value observed at a glass transition temperature in Ref. [57]. The slopes immediately preceding T_{g_visc} and T_{g_diff} were adopted as a proxy for experimental m. Another feature derived from rheology is T_c, which is determined at the temperature where materials reach a cut-off in viscosity [53]. We extracted the temperatures at which self-diffusion and viscosity reach values of $10^{-4.0}$ [Å/ps] and $10^{2.0}$ [$Pa \cdot s$], respectively for each composition. The chosen values ensured that the functions from Eq. 2.8 were not significantly extrapolated beyond the temperature range used to fit the equation's parameters for most compositions. Additionally, deviations from Arrhenius behavior for self-diffusion and viscosity were measured. For viscosity, this matches the definition of the temperature denoted as T_a. For diffusion, we are not aware of any specific naming convention for the temperature at which it is measured. However, we included this temperature as a feature because it seemed plausible that it could provide useful information. The features discussed so far

will be denoted as X_{mtp} as they all come from the MTPs.

2.6.3.7 Features Generated by Mixing MLP Potentials and Ab Initio

Enthalpy of formation, E_{form}, and energy above the convex hull, E_{above}, constitute the feature set X_{crys} and are an approximation of the stability of the amorphous phase (given in Table 2.9). Differences between energies of phases have previously been shown to be important for GFA models as seen by the top features in Ref. [4]. We generated X_{crys} to capture information about crystalline metals through a combination of MLPs and ab initio. We added pure ab initio because the MLPs were not fit to any crystalline structures and might be unreliable for those atomic arrangements. All simulations were conducted with 100 atoms. Amorphous structures were obtained through simulated quenches from 3,000 $[K]$ down to 100 $[K]$ at 0.5 $[K/fs]$ using MLPs. The resulting structures were relaxed with ab initio approaches to obtain the enthalpy of amorphous structures (denoted as E_{amorph}). This process was repeated 3 times for each material to give an average E_{amorph}. E_{form} was calculated with a face-centered cubic (FCC) crystal reference for each element in the compound (each of which is denoted by E_i for a total of N chemical species). Specifically, E_{form} was determined with Eq. 2.9.

$$E_{form} = E_{amorph} - \sum_{i}^{N} E_i \tag{2.9}$$

For each chemical system, an estimated convex hull was generated using the Materials Project database of crystal structures [106]. Comparing the enthalpy of formation for each specific composition to the hull energy gives the resulting E_{above} for each amorphous material, which can be taken as an approximation to the driving force for crystallization.

2.6.3.8 Elemental Property Based Features

Here we discuss the more easily computed features based on elemental properties. For an alloy, one can take functions of the constituent elements in ways that represent the alloy [107]. Specifically, simple to generate features based on minimum, maximum, averaged,

and composition weighed averaged properties of elements were acquired through MAST-ML. 409 elemental properties included, such as thermal conductivity, atomic sizes, and electronegativities. We refer to these features as X_{mastml}. Additional descriptors involving specific physically motivated functions of atomic sizes, enthalpies, Pauling electronegativity, and mole fractions have shown promise in the work of Long et al. [84]. These descriptors were also used in the present study, following the formulae and naming conventions provided in Ref. [84]. We refer to these features as X_{long}. Table 2.9 provides the formulation for features from X_{long} that were identified as important in the SHAP analysis (see Sec. 2.6.3.10). Features from X_{mastml} and X_{long} can be easily generated for all 177 materials in Sec. 2.6.3.2.

2.6.3.9 Regression Models and Corresponding Assessment

An eXtreme Gradient Boosting (XGBoost) model type was employed throughout the study [20] due to its ability to efficiently and accurately learn complex relationships. The target of regression models was $log_{10}(R_c)$ and were evaluated through a Chemical System Leave-Out (CSLO) CV. For CSLO CV, we group each material by its chemical system. We then perform leave one group out CV. As an illustration, assume a dataset has material entries with elements e_1, e_2, and e_3. The following are all possible combinations of left out sets of materials: $\{e_1\}$, $\{e_2\}$, $\{e_3\}$, $\{e_1, e_2\}$, $\{e_1, e_3\}$, $\{e_2, e_3\}$, and $\{e_1, e_2, e_3\}$. As an illustration of 1 iteration, an evaluation focused on the $ZrCuAl$ system could use data from the systems of $LaAu$ and $ZrCu$ for training and predict $ZrCuAl$ as left out validation data. Note that $ZrCu$ is in the $ZrCuAl$ system but has a different number of components and is therefore considered to be in another group. Test metrics were gathered across all iterations of leaving chemical systems out.

Various metrics were employed to evaluate models, including the coefficient of determination (R^2), mean average error (MAE), root mean squared error ($RMSE$), and $RMSE$ normalized by the target standard deviation ($RMSE/\sigma_y = \sqrt{1 - R^2}$). In comparing model assessments, $RMSE/\sigma_y$ was favored due to its intuitive nature. If $RMSE/\sigma_y$ is close to one, it suggests that the errors of the model are comparable to a naive model

that predicts only the mean of the property of interest. $RMSE/\sigma_y$ is defined by

$$RMSE/\sigma_y = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{2.10}$$

where i denotes a specific material, y denotes a property of interest, $\hat{y}$ denotes the prediction of y, and $\bar{y}$ denotes the mean value calculated from all y.

2.6.3.10 Comparison of Feature Effectiveness

XGBoost models constructed using X_{mastml}, X_{long}, and a union of both (i.e. $X_{mastml} \cup X_{long}$) were assessed with CSLO CV for 177 materials, with results shown in Sec. 2.6.4.3. Models fit with X_{long} alone tended to outperform models fit to X_{mastml} or $X_{mastml} \cup X_{long}$. Consequently, only X_{long} was kept for subsequent analyses and compared to more intricate, simulated features. To assess the value added by our set of simulated features to GFA models, we train XGBoost models on 34 materials (i.e., materials that have MLPs) with the combined set of $X_{long} \cup X_{mtp} \cup X_{crys}$. We will refer to $X_{long} \cup X_{mtp} \cup X_{crys}$ as X_{tot} for simplicity. We also train models on X_{long} alone as a comparison.

SHAP values were used to compare the relative importance of physically motivated features for X_{tot} [80]. These values assess feature contributions to predictions for a given model, with principles rooted in game theory. We create a feature learning curve, ordered in importance by their SHAP values, to select the subset of X_{tot} with the lowest $RMSE/\sigma_y$ which we will call X_{best}. CSLO CV was used in feature selection because we desire models that can predict GFA across chemical systems. To demonstrate that the selection of X_{best} was not due to chance, given the similar number of features and observations, we repeated the feature selection process with shuffled $log_{10}(R_c)$ values. This approach demonstrated that achieving an $RMSE/\sigma_y$ as low as the model from the original feature selection is improbable when randomly fitting a model to $log_{10}(R_c)$ with features unrelated to GFA (see Sec. 2.6.4.5).

2.6.4 Results

2.6.4.1 Errors from MLPs

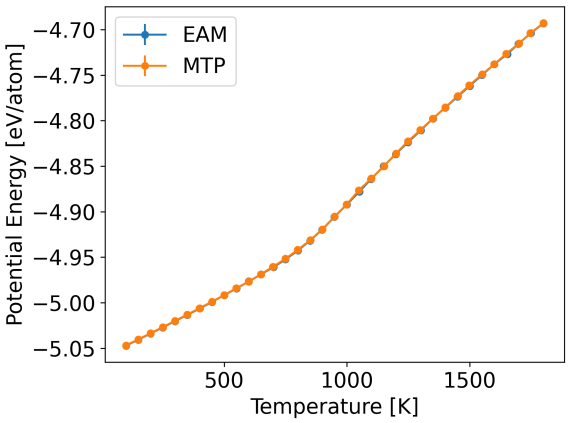
For the three test systems where we are comparing the MLPs to EAM potentials, the fitting methodology outlined in Sec. 2.6.3.4 produced the $RMSE$ for forces and energies shown in Table 2.10. Our $RMSE$ values are generally comparable to other MLPs on metal alloys (0.1 $[eV/\text{Å}]$ for forces and 0.01 $[eV/atom]$ for energies), except maybe for the force errors being a bit higher. However, the MLPs are sufficiently accurate to produce properties of interest as shown subsequently.

Table 2.10: The errors from MTP potentials compared with EAM potentials are tabulated below.

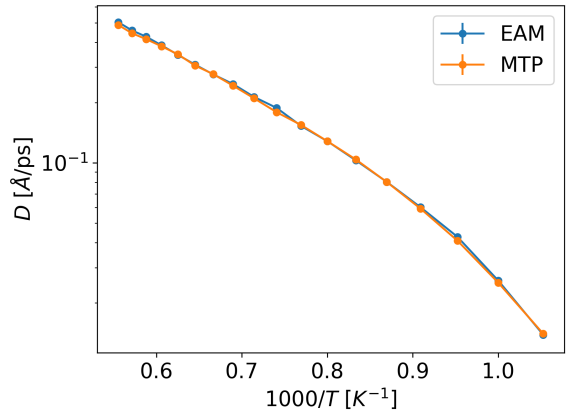
Composition	Property	Units	RMSE
$Ni_{80}P_{20}$	Force	$eV/\text{Å}$	0.187
$Pd_{75}Si_{25}$	Force	$eV/\text{Å}$	0.124
$Al_{10}Cu_{40}Zr_{50}$	Force	$eV/\text{Å}$	0.080
$Pd_{75}Si_{25}$	Energy	$eV/atom$	0.012
$Ni_{80}P_{20}$	Energy	$eV/atom$	0.007
$Al_{10}Cu_{40}Zr_{50}$	Energy	$eV/atom$	0.004

Fig. 2.12 shows an example temperature dependence of potential energy, self-diffusion, and viscosity for $Al_{10}Cu_{40}Zr_{50}$. Parallel data for the other materials may be found in the Supplemental Materials. All three properties are similar between EAM potentials and MLPs. In other words, we can simulate properties of interest with sufficient accuracy with the potential errors in Table 2.10.

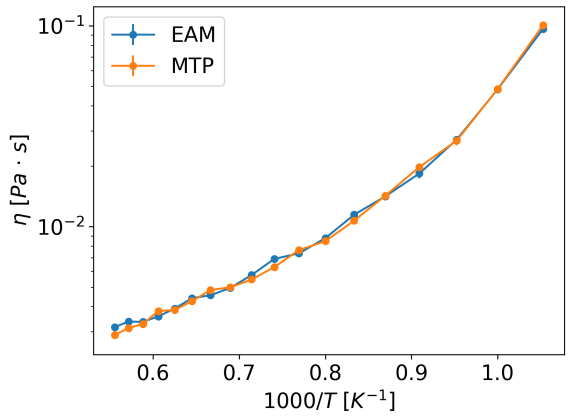
For the 34 MLPs used to extract features for machine learning, the errors in energies and forces were acquired with the method outlined in Sec. 2.6.3.4 and tabulated in Table 2.11. Generally, chemical systems with more types of elements require more training configurations. The maximum obtained $RMSE$ across all MLPs for energies and forces was 0.031 $[eV/atom]$ and 0.290 $[eV/\text{Å}]$, respectively. These values are larger than ideal and



(a) Potential Energy



(b) Self-Diffusion



(c) Viscosity

Figure 2.12: Potential energy, self-diffusion, and viscosity for $Al_{10}Cu_{40}Zr_{50}$ as functions of temperature are shown. MTP and classical EAM potentials show excellent agreement across all shown properties. Note that simulated properties were averaged across 10 independent runs.

larger than our worst test case in Table 2.10, but only by up to at most 2-3 times. We therefore believe that these MLPs will produce similarly accurate model results as those used in the EAM comparison. While more accurate MLPs can certainly be developed using more complex potential functions and additional training data, such refinements were beyond the scope of this work, given the large number of systems studied.

Table 2.11: The energy $RMSE$, force $RMSE$, and number of training configurations from MTP potentials are tabulated below.

Composition	Energy $[eV/atom]$	Force $[eV/\text{Å}]$	Configurations
$Mg_{90}Nd_5Ni_5$	0.007	0.084	992
$Mg_{77}Nd_5Ni_{18}$	0.008	0.092	990
$Pd_{95}Si_5$	0.008	0.109	893
$Cu_{30}Ni_{10}P_{20}Pd_{40}$	0.009	0.143	1160
$Pd_{82}Si_{18}$	0.009	0.144	881
$Ni_{40}P_{20}Pd_{40}$	0.009	0.156	1039
$Cu_{25}Mg_{65}Y_{10}$	0.010	0.094	889
$Cu_{25}Gd_{10}Mg_{65}$	0.010	0.100	889
$Mg_{65}Nd_{15}Ni_{20}$	0.010	0.111	1010
$Cu_{25}Ni_{10}P_{20}Pd_{45}$	0.010	0.145	1173
$Cu_{25}Ni_{15}P_{20}Pd_{40}$	0.010	0.149	1172
$Mg_{80}Nd_{10}Ni_{10}$	0.011	0.098	978
$Mg_{70}Nd_{15}Ni_{15}$	0.012	0.113	1003
$Pd_{75}Si_{25}$	0.012	0.193	870
$Nb_7Ni_{59}Si_3Sn_2Ti_{13}Zr_{16}$	0.012	0.202	2541
$Ca_{65}Mg_{15}Zn_{20}$	0.013	0.113	1081
$Cu_6Pd_{77}Si_{17}$	0.013	0.156	917
$Cu_{47}Ni_8Ti_{34}Zr_{11}$	0.017	0.198	1066
$Cu_{30}Ni_5P_{20}Pd_{45}$	0.019	0.188	1046
$Nb_{40}Ni_{60}$	0.019	0.240	861
$Al_{25}Cu_{20}La_{55}$	0.020	0.158	1020
$Al_9Cu_{16}Ni_9Zr_{66}$	0.020	0.230	1223
$Al_{25}Cu_{15}La_{55}Ni_5$	0.022	0.167	1195
$Al_8Cu_{12}Ni_{14}Zr_{66}$	0.023	0.239	1255
$Al_{14}Cu_{20}La_{66}$	0.024	0.162	1011
$Be_{25}Cu_{10}Ni_{10}Ti_{11}Zr_{44}$	0.024	0.237	1510
$Al_8Cu_7Ni_{19}Zr_{66}$	0.025	0.233	1203
$Be_{35}Zr_{65}$	0.026	0.250	870
$Al_{25}Cu_{10}La_{55}Ni_{10}$	0.028	0.175	1230
$Al_{25}Cu_5La_{55}Ni_{15}$	0.028	0.177	1247
$Al_{25}La_{55}Ni_{20}$	0.029	0.187	955
$Al_8Ni_{26}Zr_{66}$	0.030	0.235	946
$Al_{25}Co_5Cu_{10}La_{55}Ni_5$	0.031	0.183	1773
$Be_{37}Ti_{63}$	0.031	0.290	903

2.6.4.2 Visualization of Some Properties Attained Through MD

The potential energy versus temperature curve for $Cu_{25}Mg_{65}Y_{10}$ is shown in Fig. 2.13. Figures for all other compositions are available as described in the Data Availability section. Fig. 2.13 also shows the values of T_f and T_s.

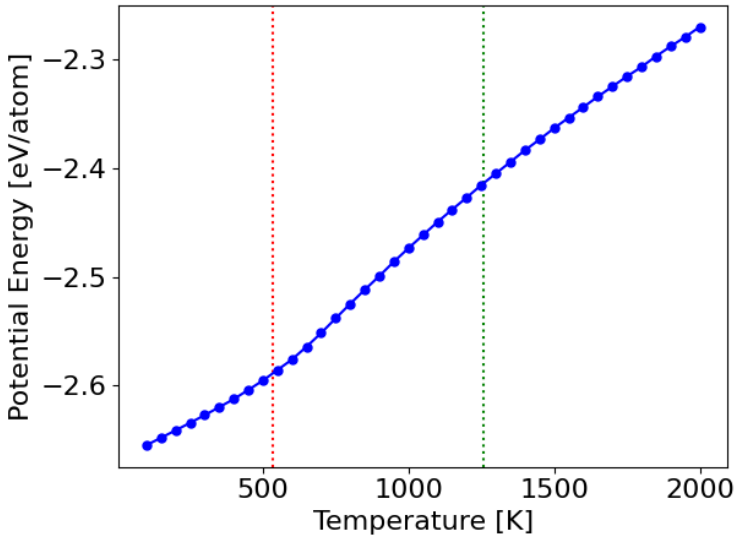
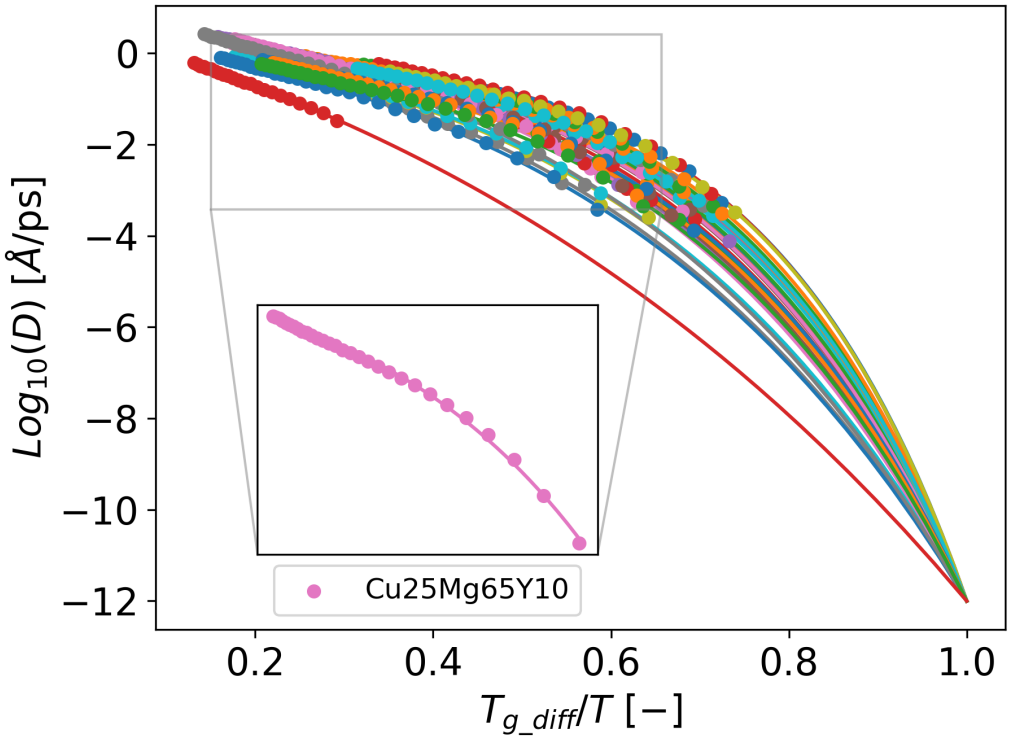
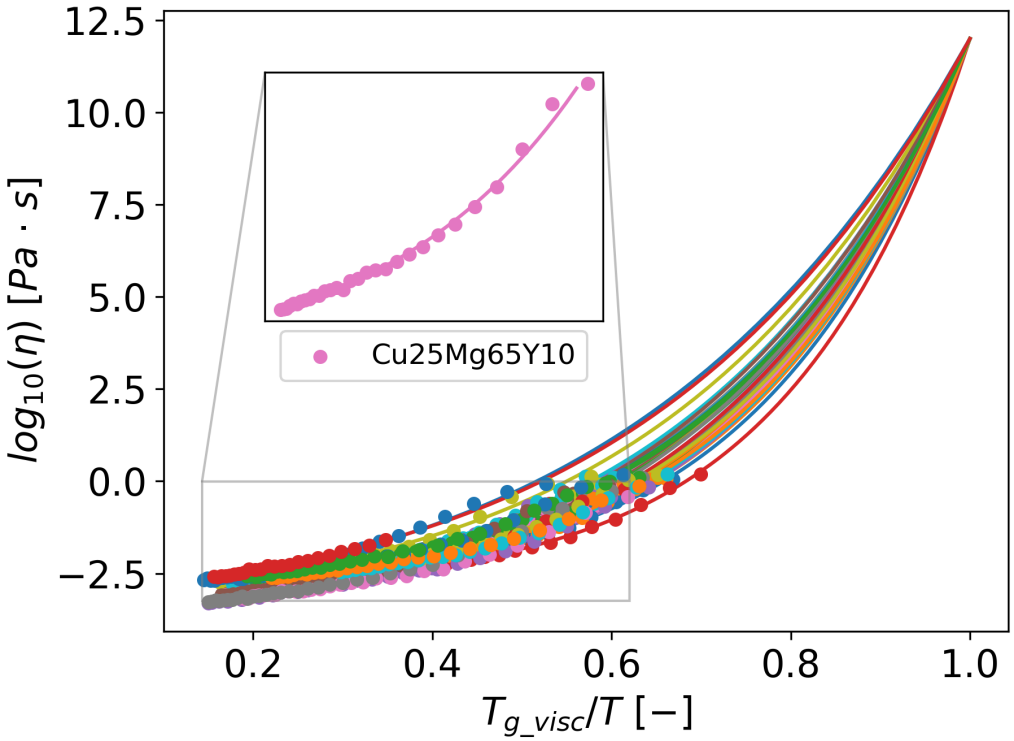


Figure 2.13: The potential energy relationship with respect to temperature is shown for $Cu_{25}Mg_{65}Y_{10}$. There are high and low temperature transitions denoted by T_f and T_s.

Self-diffusion data for 34 compositions are shown in Fig. 2.14a. Viscosity values are shown in Fig. 2.14b. The data points denote isothermal holds where averaging was performed, and continuous lines show the MYEGA fits to the data.



(a) Self-Diffusion



(b) Viscosity

Figure 2.14: Self-diffusion and viscosity for 34 compositions are shown. As materials cool, they experience restrained movement at varying degrees with respect to T_{g_diff} or T_{g_visc}. The points represent measured viscosity values and solid lines are the MYEGA fits. The MYEGA function fit to points was extrapolated to -12 and 12 on the vertical axis for self-diffusion and viscosity, respectively. Note that all fits converge at $T_{g_visc}/T = 1$ and $T_{g_diff}/T = 1$ because of our definitions of T_{g_visc} and T_{g_diff} for viscosity and diffusion, respectively.

2.6.4.3 Exploring Models Fit to Elemental Features

For 177 compositions, we evaluated the $RMSE/\sigma_y$ for $log_{10}(R_c)$ across three feature sets, X_{mastml}, X_{long}, and $X_{mastml} \cup X_{long}$. XGBoost models were evaluated with CSLO CV. The feature set that yielded the lowest $RMSE/\sigma_y$ was found to be X_{long} as shown in Fig. 2.15. Other parity plots are supplied in the Supplemental Materials. Models fit to X_{long} had an $RMSE/\sigma_y$ of 0.85 whereas models fit to X_{mastml} and $X_{mastml} \cup X_{long}$ had $RMSE/\sigma_y$ of 1.00 and 0.92, respectively. None of these models are particularly accurate, as can be seen by the fact that $RMSE/\sigma_y$ for all feature sets are close to 1.00, which is the score that would be obtained by a naive model that predicts just the mean of y. Therefore, it is useful to explore other properties that can improve our $log_{10}(R_c)$ model.

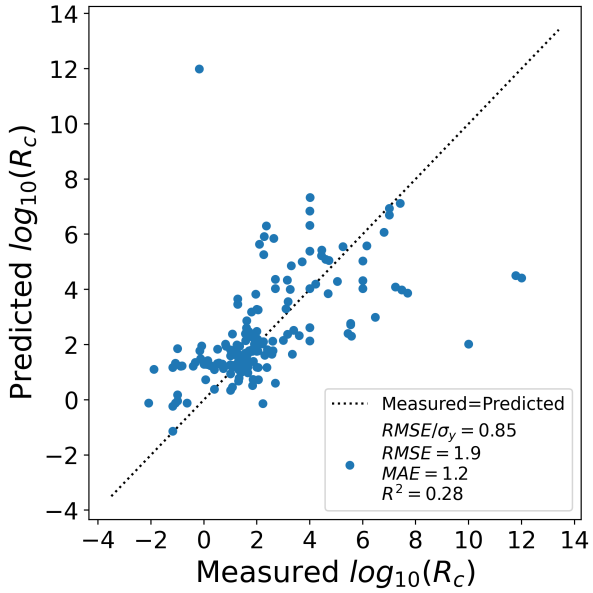
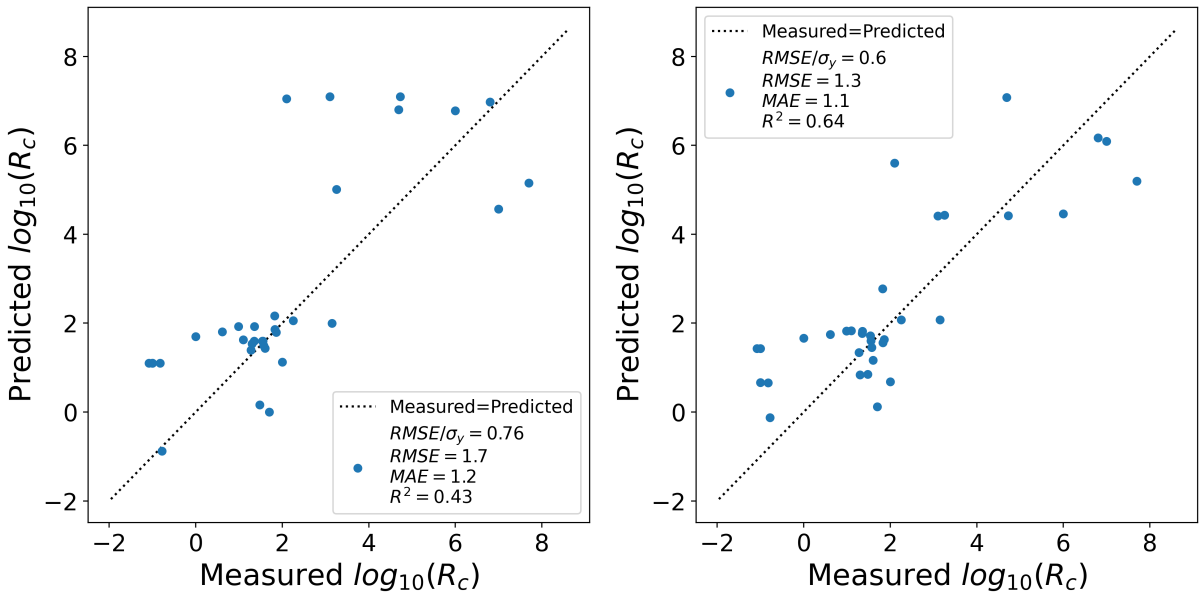


Figure 2.15: The parity plot for XGBoost models fit to X_{long} for 177 materials. The models produced are better than other sets of features, but still have high errors. Note that test data were produced by CSLO CV.

2.6.4.4 Exploring Models Fit with Simulated Features

XGBoost models were assessed using the CSLO CV for 34 materials that had MLPs. Models were built by using X_{long} alone and the combined feature set of X_{tot} (defined in Sec. 2.6.3.10 and Table 2.9). The resulting $RMSE/\sigma_y$ was 0.76 and 0.60 for X_{long} and X_{tot} respectively (Fig. 2.16). These results suggest that the inclusion of simulated features improve a model's ability to predict GFA. We refined our model by implementing

feature selection via SHAP values to show which features in X_{tot} are most important for predicting $log_{10}(R_c)$.



(a) X_{long} (b) X_{tot}

Figure 2.16: The CSLO CV results are shown here for XGBoost models fit to X_{long} and X_{tot} for 34 materials. The models built from these features has some predictive ability.

2.6.4.5 Ranking Features Based with Feature Selection

Shapley Additive Explanations (SHAP), a method based on game theory principles, provides valuable insights into feature contributions. Fig. 2.17 illustrates the SHAP values for X_{tot} with respect to an XGBoost model. The vertical axis of Fig. 2.17 represents the features, while the horizontal axis displays the SHAP values, indicating whether the prediction of a model using that feature increases or decreases the predicted value of $log_{10}(R_c)$. The mean absolute SHAP values for each feature in Fig. 2.17 determine the importance and ranking of the features. Higher values signify greater influence on a model's prediction. Each dot from each feature value corresponds to a specific material (e.g., $Al_{25}Co_5Cu_{10}La_{55}Ni_5$). By examining the color bar, we can assess whether the feature value for a specific material is high or low and then observe if those values drive the model's predictions to be higher or lower. Note that out of the 4 features shown, only S_{mix} was derived from the easy-to-compute elemental features in X_{long} and is ranked first. Other features derived from more complex simulations significantly contribute to a

model's prediction on $log_{10}(R_c)$.

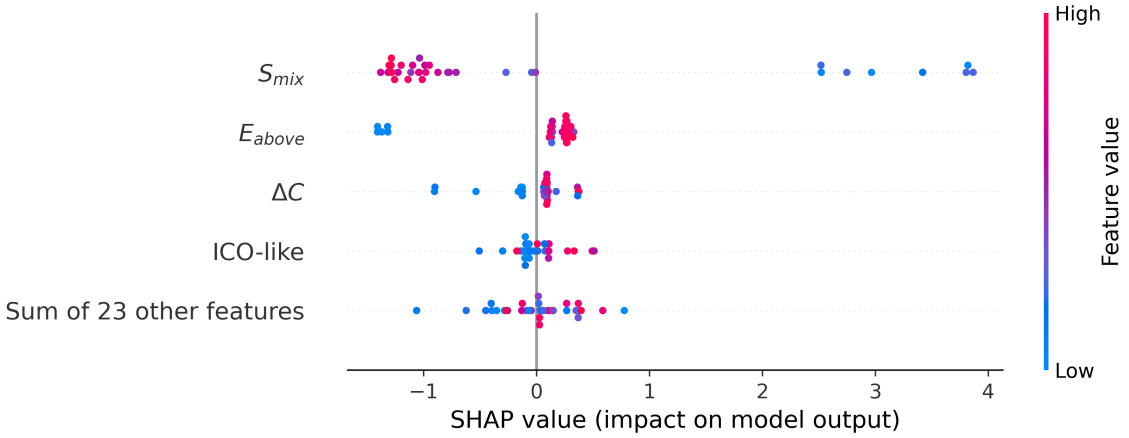


Figure 2.17: The SHAP values for XGBoost is shown for X_{tot}. Higher values of $log_{10}(R_c)$ are denoted by higher values of model output on the horizontal axis and vice versa.

We show $RMSE/\sigma_y$ as a function of the number of features used when the features are ordered based on SHAP values in Fig. 2.18. Any feature after the top 4 did not lead to regression improvements. The addition of the fifth top feature arbitrarily increased model complexity which is concerning for a small data set of 34 points and causes overfitting. Overfitting can be seen by the increase of $RMSE/\sigma_y$ starting at 5 features and beyond. The top 4 features are denoted as X_{best}, which are the ones shown in Fig. 2.17.

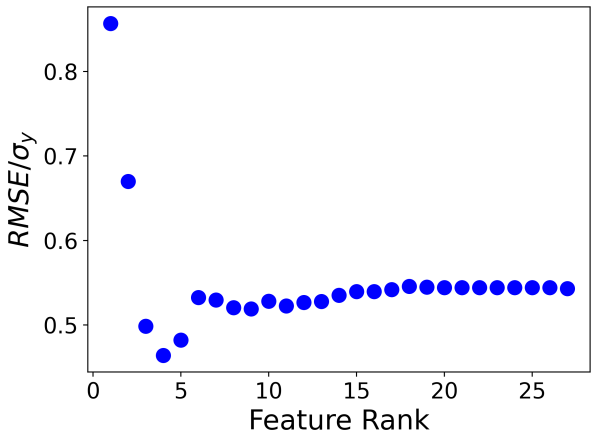


Figure 2.18: $RMSE/\sigma_y$ as a function included features sorted with SHAP values for XGBoost models are shown.

Assessing models with X_{best} yields an $RMSE/\sigma_y$ (R^2) of 0.46 (0.78). The associated parity plot is shown in Fig. 2.19. It is worth noting that the best cited $RMSE/\sigma_y$ is

an overestimation of what might be expected on test data due to feature selection being conducted without an explicit test set. Nevertheless, the number of features in X_{best} is much smaller than the 34 studied materials and the assessment is tested with CSLO CV, which provides some protection against significant overfitting. The top four features are discussed in detail in Sec. 2.6.5.

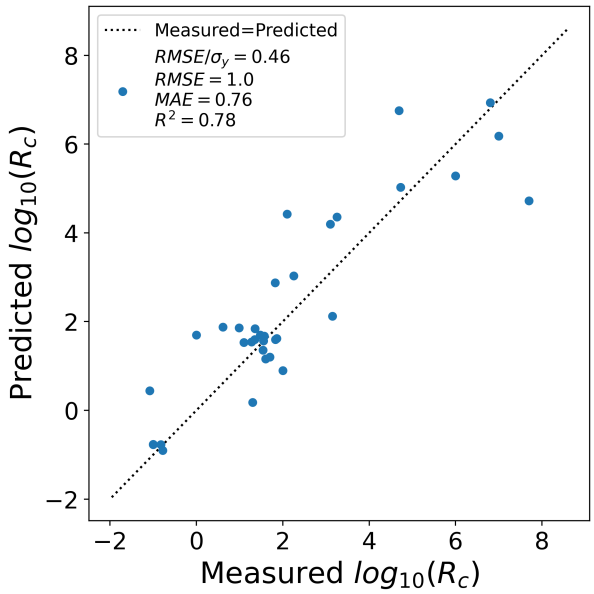


Figure 2.19: The parity plot using X_{best} for the 34 compositions that had MLPs are shown. Note that the built model appears to predict R_c across chemical systems well.

The feature learning curve obtained from the shuffling feature selection procedure (see the end of Sec. 2.6.3.10) is shown in Fig. 2.20. The best $RMSE/\sigma_y$ in Fig. 2.20 is about 1.36, significantly higher than the 0.46 achieved in Fig. 2.19. These results indicate that the selection of X_{best} from our extensive feature set was not random and that X_{best} is crucial for constructing a $log_{10}(R_c)$ model.

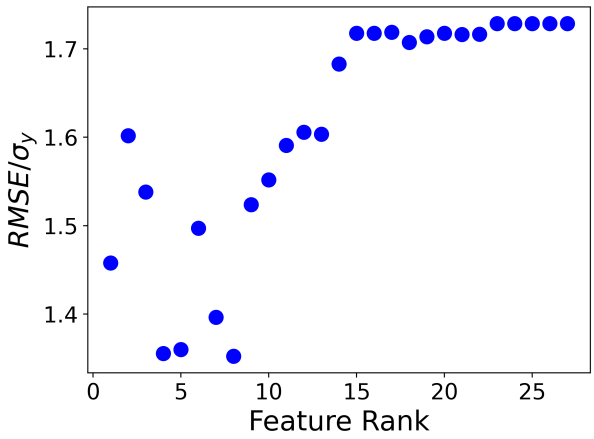


Figure 2.20: $RMSE/\sigma_y$ as a function included features sorted with SHAP values for XGBoost models are shown. This feature selection process was performed with shuffled $log_{10}(R_c)$.

2.6.5 Discussion

X_{long} emerged as the feature set yielding the lowest model errors of $RMSE/\sigma_y$ among the 177 values of R_c compared to X_{mastml}. It is perhaps not surprising that a set of physically motivated features would yield superior predictions compared to those derived from brute force arithmetic operations on elemental properties, although this does not always occur (e.g., see [108]). Nevertheless, models built from the 177 compositions using just X_{long} have a high $RMSE/\sigma_y$ of 0.85.

SHAP value rankings show that both simulated properties and X_{long} could produce viable models for predicting $log_{10}(R_c)$. The best performing models were built from only the top four features. The selected features were the mixing entropy (S_{mix} defined in Sec. 2.6.3.8), the energy of the amorphous phase above the convex hull (E_{above} defined in Sec. 2.6.3.7), the fraction of icosahedral-like local VP environments in the high-temperature liquid (ICO-like defined in Sec. 2.6.3.6), and the difference in heat capacity between the glass and liquid phases (ΔC defined in Sec. 2.6.3.6).

The important of S_{mix} is not surprising and it likely reflects the confusion principle associated with GFA [81]. In other words, including more chemical species in a material may hinder crystalline phase formation because atoms struggle to find suitable sites to form crystals, which shows up as a higher value of S_{mix}. Given this argument, we would

expect higher values of S_{mix} to correspond to lower values $log_{10}(R_c)$ (higher GFA), which is exactly the trend in Fig. 2.17.

The importance of E_{above} is also expected. E_{above} is a measure of the stability of the amorphous phase of our final glass relative to stable crystal structures and is therefore a qualitative guide to the drive for crystallization, a crucial factor controlling R_c. Higher values of E_{above} indicate less stable glasses with more driving force to crystallization, while lower values imply greater stability glasses with less driving force for crystallization. Given this argument we would expect higher E_{above} values to correspond to higher values of $log_{10}(R_c)$ (lower GFA), which is exactly the trend in Fig. 2.17.

The importance of ΔC is perhaps easiest to understand when we consider its correlation with fragility (see Ref. [103]), which is known to correlate with measures of GFA such as the R_c. An increase in ΔC is expected to correlate with an increase in fragility, which, in turn, is expected to correlate with a decrease in GFA and higher values of $log_{10}(R_c)$. This trend is precisely what is observed in Fig. 2.17.

The importance of ICO-like is consistent with results of previous research that have demonstrated that these types of polyhedral structures slow the liquid and relate to GFA [5, 90, 109, 110]. However, previous studies suggest a positive correlation between the icosahedral VP and GFA, which implies a larger ICO-like value would give a lower $log_{10}(R_c)$, which is the opposite of the trend in Fig. 2.17. It is worth noting that the impact of this variable in the SHAP analysis is modest, the negative trend of $log_{10}(R_c)$ with ICO-like value is weak and the overall impact of this variable on the model accuracy in Fig. 2.18 is very modest. Taken together it is reasonable to say that the impact of this variable on $log_{10}(R_c)$ is not well determined by the preset model and data, so disagreement with trends is not unexpected. However, it may also be that icosahedral fractions do not have simple correlations with GFA and $log_{10}(R_c)$ across so many widely varying chemistries, especially when evaluated at a single fixed temperature (e.g., rather than a fixed homologous temperature). Even the definition of VP that are considered ICO-like is somewhat ambiguous and can vary because of distortions [111]. In summary, when

one looks deeply, the possibly somewhat unphysical behavior of the ICO-like variable correlations is not evidence of any significant issue with the model.

We highlight the significance of developing a model of experimental $log_{10}(R_c)$ from features developed computationally, especially since many of the features from X_{best} are more simple to acquire computationally compared to their experimental counterparts. Take E_{above} and ΔC as examples. While ΔC can be obtained through calorimetry, it remains challenging unless a large quantity of a high GFA alloy is accessible (i.e., this quantity cannot be easily experimentally measured for low GFA alloys). Determining E_{above} experimentally is even more difficult due to the complexities in establishing a baseline for differential scanning calorimetry. Thus, leveraging computational models not only streamlines the process of measuring critical material properties but also offers an alternative to experimental characterization and their inherent challenges.

Together the 4 features yield an apparently fairly capable model of $log_{10}(R_c)$, with an R^2 of 0.78 (although we reiterate that this is likely an overestimation due to some data leakage). Regardless of model metrics, it is crucial to note that the model should not be relied upon for any significant materials screening. Further tuning of model parameters with a larger dataset than the current 34 observations is necessary to assure the model has a reasonable domain of applicability. However, these initial results are quite encouraging. Furthermore, generating larger data sets is largely limited by computation, which will become much easier with increases in computational power and more efficient computational techniques. This study suggests a promising path for integrating elemental and simulated features for predicting GFA in metals and illustrates how researchers can explore and identify physically motivated features with the flexibility now provided by MLPs.

2.6.6 Conclusion

In this work we explored an extensive range of elemental and molecular simulation calculated properties to develop a simple model for predicting critical cooling rate, R_c for

metallic glasses. In particular, we employed machine-learned interatomic potentials to facilitate the simulation of materials properties inaccessible via traditional ab initio or classical interatomic potential methods and difficult or expensive to obtain through experiments, all while maintaining accuracy comparable to ab initio and computational efficiency comparable to potentials. We conducted simulations on 34 compositions to collect properties potentially linked to the glass forming ability in metallic systems. We investigated self-diffusion, viscosity, characteristic temperatures, short range order, and energy comparisons between material crystalline and amorphous phases. Additionally, we explored easily computed features based on elemental properties that do not require simulation. Out of this feature set, we identified that four features were particularly significant for constructing regression models to predict R_c. Notably, one of the features was obtained through trivial computation, while the other three required simulation with machine-learned interatomic potentials. Evaluated models could predict $log_{10}(R_c)$ with an R^2 of 0.78 for chemical systems excluded from model training. While our model had an impressive R^2, caution is warranted due to potential overfitting from the feature selection process and the small data sets involved. More broadly, we have demonstrated a versatile approach to systematically explore previously inaccessible material properties across diverse chemical systems using machine learned potentials. Almost all properties used as features for our model followed physical intuition outlined by previous research. Other computational endeavors can benefit from adopting a similar approach to quantify desirable characteristics across multiple materials, rather than focusing solely on properties within individual chemical systems.

2.6.7 Data Availability

The raw and processed data required to reproduce these findings are available to download from figshare at doi: 10.6084/m9.figshare.26142382.

2.6.8 Impact from Work

- This work advances the field by developing a model that successfully predicts the critical cooling rate of metallic glasses using a combination of elemental and simulated features, which avoids experimental characterization of materials for glass forming ability prediction.
- The methodology of machine learning potentials from ab initio data to increase the length and time scales of simulated properties is a versatile approach that can be broadly applied to investigate the properties of various materials.

Chapter 3

Machine Learning Model Domain

Note: This paper has been submitted for peer-reviewed publication in npj Computational Materials and has been adapted for use in this thesis.

3.1 Abstract

Knowledge of the domain of applicability of a machine learning model is essential to ensuring accurate and reliable model predictions. In this work, we develop a new and general approach of assessing model domain and demonstrate that our approach provides accurate and meaningful domain designation across multiple model types and material property data sets. Our approach assesses the distance between data in feature space using kernel density estimation, where this distance provides an effective tool for domain determination. We show that chemical groups considered unrelated based on chemical knowledge exhibit significant dissimilarities by our measure. We also show that high measures of dissimilarity are associated with poor model performance (i.e., high residual magnitudes) and poor estimates of model uncertainty (i.e., unreliable uncertainty estimation). Automated tools are provided to enable researchers to establish acceptable dissimilarity thresholds to identify whether new predictions of their own machine learning models are in-domain versus out-of-domain.

3.2 Introduction

Machine learning (ML), as one component of the larger umbrella of artificial intelligence (AI), is one of the fastest evolving technologies in the world today. In the context of materials science, thousands of papers using ML are now published each year, and the number of publications using ML has been growing exponentially since around 2015 [112, 113]. The applications of ML in materials science takes many forms, including materials property prediction [84], computer vision-based defect detection and microstructure segmentation [114, 115], assimilation of data and knowledge from publications using natural language processing and large language models [97, 116], and fitting of ML-based interatomic potentials representing nearly all elements in the periodic table to enable fast and accurate atomistic simulations [117].

Useful ML models generally require some form of prediction quality quantification because models can experience significant performance degradation when predicting on data that falls outside the model's domain of applicability. This performance degradation can manifest as high errors, unreliable uncertainty estimates, or both. Without some estimation of model domain, one does not know, a priori, whether the results are reliable when making predictions on new test data. More precisely, useful ML models ideally have at least the following three characteristics regarding their prediction quality: (i) accurate prediction, meaning the model has low residual magnitudes, (ii) accurate uncertainty in prediction, meaning the model produces some useful quantification of uncertainty on new predictions (note this requirement does not stipulate that the uncertainties should be small), and (iii) domain classification, meaning the model can reliably determine when predictions are inside a domain (ID) versus outside a domain (OD) of feature space where the model is trustworthy.

Separate from determining whether data are ID or OD are the techniques used in domain adaptation. In certain situations, domain adaptation techniques enable the fine-tuning of a model or data to transform originally OD data into ID data. The objective of domain adaptation is to adapt a model for prediction on a property (denoted as M^{prop} here) to

new data whose distribution may be shifted from training data [118, 119]. There would be no need to identify OD data if domain adaptation were always effective, but adapting models to initially OD data can be a challenging and intricate process. First, many techniques require re-training models, involving a substantial effort in tuning parameters and validating models. Second, once a model is adapted to one target domain, it may still fail on other unknown domains. It is therefore useful to have a method to identify when a model is applied to problematic domains without having to adapt the model. In this work, we develop a domain classification technique that identifies when predictions are likely ID or OD (equivalently ID/OD).

The domain classification problem, at least for materials property prediction and many other similar problems, can be formulated as follows: given a trained model M^{prop} and the features of an arbitrary test data point, how can we develop a model to predict if the test data point is ID/OD for M^{prop}? In this work, we frame this challenge as a classic supervised ML problem for categorization. To develop such a model, we need training data with input features and labels, which labels are ID/OD, as well as some ML modeling approach for making the label prediction. We will denote this ML model for domain as M^{dom} to distinguish it from M^{prop}. Note that the labeled training data for M^{dom} does not necessarily have to match the original M^{prop} model training data.

There is no unique, universal definition for the domain of an M^{prop} model, and therefore no unambiguously defined labels for the M^{dom} training data. In other words, we do not have an absolute ground truth labeling on which to train the M^{dom} model [120, 121]. This problem can be solved by imposing some reasonable definition of ground truth for ID/OD based on model reliability, as quantified by, e.g., small residual magnitudes and stable predictions under changes in data. In many cases ID/OD data points are described in terms of a region of feature space, in which case M^{dom} becomes a trivial check if a data point is in the region or not [122]. Predictions of ID/OD are often checked against chemical intuition of whether data are somehow "similar" to training data, and therefore ID, or not, and therefore OD. Such checks are an effective way of using field-specific knowledge of similarity to provide a ground truth for ID/OD classification.

Two sophisticated approaches have been developed in Refs. [123, 124] that effectively find a region in feature space where an M^{prop} model shows performance above some cutoff. These approaches are potentially very powerful but have limitations. For methods that create a single connected region to denote as ID, multiple disjointed regions in space that could yield perfectly reasonable predictions from M^{prop} will be excluded. A method by which non-connected ID regions are established without a single, pre-defined shape would be advantageous (e.g., with kernel density estimation as done in this work). In addition, the sophisticated approaches in these models introduce significant complexity, which makes them challenging to implement. It is therefore useful to revisit simpler approaches that build directly on the intuition that ID regions of feature space are likely to be those regions close to significant amounts of training data [125–127].

There are many techniques for quantifying closeness in feature spaces, including convex hulls, distance measures, and (probability) density estimates [122]. While identifying as ID all data within a convex hull of the training data methods is reasonable, such approaches have the major limitation of potentially including large regions with no training data. For example, the convex hull of points on a circle in a two-dimensional feature space includes the entirely empty middle of the circle as ID. Distance measures are also a reasonable approach to measure closeness. In Ref. [128], distance measures from a number of nearest neighbors was used as a dissimilarity score between a point and a model's training data. They showed that target property prediction errors generally increase with increasing distance, which followed intuition. Nevertheless, distance measures have the limitation of there being no unique measure of distance between two points and no unique single distance of a new point from a set of training data. This creates a vast space of possible ways of measuring two-point distances (e.g., Euclidean, Mahalanobis, etc.) and one to N-point distances (closest point distance, closest k-points distance, weighted average of distances, etc.), making it difficult to find a robust method. In general, approaches based on convex hulls or standard distances between two points do not account naturally for data sparsity, and may consider a point near one outlier training data point or many training data points as almost identically likely to be ID.

Kernel density estimation (KDE), and density-based methods in general, offer several advantages vs. other approaches, including (i) a density value that can act as a distance or dissimilarity measure, (ii) a natural accounting for data sparsity, and (iii) trivial treatment of arbitrarily complex geometries of data and ID regions. Techniques utilizing Gaussian process have comparable advantages. But unlike Gaussian process, KDE is relatively fast to fit and evaluate, at least of modest size data sets that are common in materials (see the Appendix for comparisons between KDE and Gaussian process regression). The work in Ref. [129] employed KDEs to demonstrate, using a projection of features, that many assessments of machine learning models were in regions where models had a significant number of training data. Their research demonstrated that numerous assessments previously categorized as extrapolation were, upon closer examination, actually instances of interpolation. The authors also showed that model residuals generally increased in regions of the feature space with little to no training data. While the authors employed KDE in their study, they do not utilize it as a means to categorize new predictions as ID/OD. In contrast, our research establishes a definition for ID/OD classification based on prediction errors. Later, we demonstrate how KDE can effectively differentiate data points that fall within the ID category and those that are considered OD. We therefore focus on KDE, as it provides a natural solution to all the issues around topology, distance measures, data sparsity, and has been effectively shown to have a relationship with prediction errors in the past.

Given the absence of any unique ground truth, as noted above, we explore four different approaches for defining ID/OD. Specifically, we define four domain types, each based on a corresponding ground truth, which are: (i) a chemical domain where test data materials with similar chemical characteristics to the training data are ID, (ii) a residual domain where test data with residuals below a chosen threshold are ID, (iii) another residual domain where groups (i.e., not single cases) of test data with residuals below a chosen threshold are ID, and (iv) an uncertainty domain where groups of test data with differences between predicted and expected uncertainties below a chosen threshold are ID. For each of our four domain types, we assessed our models on sets of test data

that were increasingly distinct from the training data. Generally, test cases that had low KDE likelihoods were chemically dissimilar, had large residuals, and had inaccurate uncertainties, just as one would hope for an effective method of domain determination.

Here we summarize the structure of the paper. Materials and Methods has some basic details on software, ML models, our definitions of domain, the methodology behind assessing domains, and our five data sets with details on their featurization and chemical properties. Four materials property data sets and one commonly used synthetic data set were studied. Various model types including random forest, bagged support vector regressor, bagged neural network, and bagged ordinary least squares models were trained with the aforementioned data sets. We then show the assessment of our domain predictions. We finish with conclusions, data and code resources, and acknowledgements. Our findings indicate that KDE likelihoods can provide valuable insights of model applicability domain, enabling effective classification of ID/OD for most data sets and models examined here. Importantly, our approach is expected to be generally applicable to regression-based prediction problems involving tabular data and can be easily applied to other regression tasks.

3.3 Material and Methods

3.3.1 Software Tools Used

In our study, we utilized several software tools for analysis. We employed the Materials Simulation Toolkit for Machine Learning (MAST-ML) to generate and select features, X, across multiple data sets [86]. Furthermore, we utilized various models and subroutines from scikit-learn [42]. For neural network (NN) implementations, we relied on Keras [130]. Visualizations from projecting X onto lower dimensions used the Uniform Manifold Approximation and Projection (UMAP) package [131].

3.3.2 Definition of Model Types

We outline here the types of models, their applications in this work, and define the nomenclature for each. First, M^{prop} is a regression model that uses X to predict a property, y. Second, M^{dom} is a classification model that predicts domain labels ID/OD given the features of a single data point, $\vec{x}$. The predicted labels of domain will henceforth be called $\widehat{ID}$ and $\widehat{OD}$. Construction of M^{dom} requires other additional models. We define a model called $M^{dis}(X,\vec{x})$ (or just M^{dis} for short) which returns a dissimilarity score between $\vec{x}$ and the data X used in training M^{prop}. Finally, we also build a model for predicting uncertainties in M^{prop} predictions, M^{unc}, which uses M^{prop}. The data used to build or train the models of M^{prop}, M^{unc}, and M^{dis} will be referred to as In-The-Bag (ITB). Conversely, the data excluded from the training of these models will be termed Out-Of-Bag (OOB). The following subsections provide explanations for how M^{prop}, M^{unc}, and M^{dis} combine to generate M^{dom}. Because of the interdependence of many kinds of models to produce M^{dom}, a summary of models, their inputs, and their outputs are provided in Table 3.1.

Table 3.1: The summary of variables used in model training, variables used in model prediction, and the outputs from each model are covered here. $\{\cdot\}$ represents a set.

Model	**Description**	**Training Inputs**	**Deployment Inputs**	**Outputs**
M^{prop}	Property prediction	$X, \{y\}$	$\vec{x}$	$\hat{y}, \sigma_u$
M^{unc}	Uncertainty estimation	$\{y, \hat{y}, \sigma_u\}$	σ_u	σ_c
M^{dis}	Measuring dissimilarity	X	$\vec{x}$	d
M^{dom}	Classifying domain	$\{d, ID/OD\}$	d	$\widehat{ID}/\widehat{OD}$

In some cases, these models are very simple, e.g., just a simple function or a check if a value is above or below a cutoff. We make the choice of defining them as models and denoting them with a variable for two reasons. First, it gives a well-defined symbol for each item, which makes the discussion more precise and compact, although at the cost of more variables. Second, defining each of these relationships as models stresses the fact

that the specific models we use in this work could easily be replaced by other models, and these could be much more complex. For example, our domain model M^{dom} for predicting whether a test data point with features $\vec{x}$ belongs to the ID domain is a simple check of whether the kernel density value exceeds a cutoff. However, this could be replaced by a more complex ML model based on the KDE or other features. We hope that these definitions will help make the paper clearer and suggest natural ways to improve our approach in the future.

3.3.2.1 Model for Property Regression (M^{prop})

We investigated a range of model types for M^{prop} including Random Forest (RF), Bagged Support Vector Regressor (BSVR), Bagged Neural Network (BNN), and Bagged Ordinary Least Squares (BOLS). We used bagged versions of all but the RF models (RF is already an ensemble model) as these ensembles were used to generate uncertainty estimates and define the model M^{unc}. Note that M^{dis} depends only on the features of ITB data, X_{ITB}, and does not depend on the form of M^{prop} nor M^{unc}. Two types of errors on property predictions were considered. First, absolute residuals ($|y - \hat{y}|$), normalized by the Mean Absolute Deviation of y (MAD_y) was measured with Eq. 3.1 and was named $E^{|y-\hat{y}|/MAD_y}$.

$$E^{|y-\hat{y}|/MAD_y} = \frac{|y - \hat{y}|}{MAD_y} \tag{3.1}$$

Second, we denote the Root Mean Squared Error ($RMSE$) of predictions from M^{prop} normalized by the standard deviation of y (σ_y) by the symbol E^{RMSE/σ_y} (see Eq. 3.2). In both Eqs. 3.1 and 3.2, "E" denotes that it is a type of error, $\hat{y}$ is a prediction from M^{prop}, and $\bar{y}$ is the mean of y. $E^{|y-\hat{y}|/MAD_y}$ can be measured for any individual data point considered, but can be randomly low for a data point known to be OD (i.e., residuals are stochastic). E^{RMSE/σ_y} considers residuals for groups of data, so data points with low and high values of $E^{|y-\hat{y}|/MAD_y}$ are included in a statistical measure. Both $E^{|y-\hat{y}|/MAD_y}$ and E^{RMSE/σ_y} were used in producing ID/OD labels to train M^{dom} (see the Defining Ground Truths section).

$$E^{RMSE/\sigma_y} = \frac{RMSE}{\sigma_y} = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{3.2}$$

3.3.2.2 Model for Uncertainty Estimates (M^{unc})

Uncertainty calibration has been extensively studied in both classification and regression settings [132–136]. Our study utilized uncertainty estimates for regression and used the calibration implementation from Palmer et al. [137]. In this approach, we started from an ensemble model to acquire individual predictions from each sub-model comprising an ensemble. The mean of predictions from the sub-models is $\hat{y}$. The standard deviation of predictions, σ_u, was calculated and then calibrated to yield the calibrated uncertainty estimates, denoted as σ_c. Repeated 5-fold CV was used throughout the study for producing residual and σ_u values used for calibration.

We borrowed arguments from Ref. [138] for assessing the quality of calibrated uncertainty estimates. The mean and variance of a set of z-scores, Z, should be 0 (unbiased) and 1 (unit-scaled), respectively. Each $z \in Z$ was calculated by $z = (y - \hat{y})/\sigma_c$. We made the assumption that the distribution of Z is standard normal (i.e., had a mean of 0 and a standard deviation of 1) for reliable uncertainty estimates as done in Refs. [139] and [137], which is not done in Ref [138]. We called the cumulative distribution of both Z and the standard normal distribution $O(z)$ and $\Phi(0,1)$, respectively. We calculated how far any $O(z)$ was from $\Phi(0,1)$ via Eq. 3.3, where E^{area} is the miscalibration area or error in uncertainties. Other measures of the quality of uncertainty measures exist like sharpness, dispersion, etc. (see Ref. [140]) that could be used instead. However, we used Eq. 3.3 because it includes information of σ_c, y, and $\hat{y}$. E^{area} was important for determining the quality of uncertainties in future sections and was used to produce ID/OD labels for training M^{dom} (see the Defining Ground Truths section).

$$E^{area} = \int_{-\infty}^{\infty} |O(z) - \Phi(0,1)| dz \tag{3.3}$$

E^{area} is a comparison of two statistical quantities and is inaccurate for small sample sizes.

However, the comparison is still possible. If the sample size of $O(z)$=$\Phi(0,1)$ is 1, then the cumulative distribution will be 0 until the single observation and then become 1. E^{area} will be non-zero in the aforementioned case, but does not represent the population of $O(z)$ well. In other words, even a single value sampled from a standard normal distribution will give a non-zero E^{area}. To prevent issues with small samples sizes, we performed multiple splits and binned data as outlined in Model Assessments section to get many points for statistical comparisons.

3.3.2.3 Model for Dissimilarity Measures (M^{dis})

KDE is a method to approximate a density of points given finite sampling of points from that density. If KDE is fit to a density distribution normalized to an area of one over a region, then KDE can be used to estimate the probability density (often called a likelihood) of observing points from the distribution at a point in that region. KDE works by first placing a local distribution (kernel) with a thickness (bandwidth) on each observed data point. Overlapping regions of kernels are superimposed, providing an estimate of an overall density and can be generalized to any number of dimensions [141]. In our approach, we developed a KDE of X_{ITB} whose values approximately give the likelihood of finding a data point at any coordinate in feature space.

To implement KDE, we took the following steps. First, X_{ITB} was standardized using scikit-learn's StandardScaler fit (i.e., data were rescaled to have a mean of zero and a standard deviation of one for each feature) [42]. This was done because the KDE implementation in scikit-learn uses a single bandwidth parameter, so all dimensions of X_{ITB} should be set to a similar length scale. Then, that single bandwidth parameter was used with the Epanechnikov kernel to construct the KDE of X_{ITB} data of M^{prop}. The bandwidth was estimated automatically using a nearest-neighbors algorithm as implemented in scikit-learn through the sklearn.cluster.estimate bandwidth method [42]. One could potentially obtain better results by more carefully optimizing the bandwidth for each data set and model combination studied. However, we opted for a straightforward automated approach to avoid any possible data leakage into our assessment and to keep

the method simple to understand and implement. We used the Epanechnikov kernel because it is widely used and is a bounded kernel with a value of zero for any observation outside the bandwidth. This means that the likelihood of far away data will be exactly zero. Other kernels like the Gaussian and exponential kernels are unbounded and have non-zero values for any point. More details for these hyperparameter choices are covered in the Appendix.

Scikit-learn uses the natural logarithm of likelihoods as the inference outputs of KDE. We converted any logarithmic output to likelihoods by exponentiating the values. Eq. 3.4 was then employed to transform the likelihood of any inference data point $\vec{x}$ with respect to the maximum likelihood observed from X_{ITB}. The result obtained from Eq. 3.4 served as our dissimilarity measure, d. It is important to note that $\vec{x}$ was transformed using the same scaler utilized to build the KDE previously mentioned prior to attaining its likelihood with $KDE(\vec{x})$.

$$d = 1 - \frac{KDE(\vec{x})}{\max\limits_{\forall \vec{a} \in X_{ITB}} (KDE(\vec{a}))} \tag{3.4}$$

This transformation did not alter the information provided by the KDE and was performed only to produce an easy to interpret number, which ranges from 0 to 1. A d value of 0 corresponds to $\vec{x}$ being in the region of feature space most densely sampled by X_{ITB} (i.e., the peak of the KDE) and a d value of 1 corresponds to $\vec{x}$ being far from X_{ITB}, where the density is zero for our choice of kernel.

3.3.2.4 Model for Domain (M^{dom})

M^{dom} is a classifier which relies only on d as an input and produces $\widehat{ID}/\widehat{OD}$ given a cutoff, d^t. Values $d < d^t$ result in the label $\widehat{ID}$ and values $d \geq d^t$ result in the label $\widehat{OD}$ (Eq. 3.5). d^t is a value chosen from interval $[0, 1]$. Each d^t is associated with a specific set of $\widehat{ID}/\widehat{OD}$ predictions. The effectiveness for a given d^t is assessed by precision and recall given a ground truth, which we define based on chemistry, $E^{|y-\hat{y}|/MAD_y}$, E^{RMSE/σ_y}, or E^{area} (see the Defining Ground Truths section for ground truth labeling).

$$\text{Predicted Label} = \begin{cases} \widehat{ID}, & \text{if } d < d^t \\ \widehat{OD}, & \text{otherwise} \end{cases} \tag{3.5}$$

Here, we explain the procedure for training M^{dom}, which amounts to determining a single value called d_c^t. Note that M^{dom} can be trained with data from $E^{|y-\hat{y}|/MAD_y}$, E^{RMSE/σ_y}, E^{area}, or chemical information, and each will produce a different M^{dom}. Assume we have a data set split into ITB and OOB sets. We generated labels on OOB data for training M^{dom} by following the procedures outlined in the Defining Ground Truths section, which produced the class labels of ID and OD. For each d^t, precision and recall were measured by comparing ID/OD and $\widehat{ID}/\widehat{OD}$ (Eq. 3.6). The number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) were acquired, corresponding to when the (ground truth, prediction) are (ID, $\widehat{ID}$), (OD, $\widehat{OD}$), (OD, $\widehat{ID}$), and (ID, $\widehat{OD}$), respectively. Note that TN is not used in Eq. 3.6, but it is included for completeness.

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN} \tag{3.6}$$

M^{dom} learns by selecting d_c^t such that desirable properties are acquired. d_c^t can be selected to maximize the harmonic mean between precision and recall ($F1_{max}$) or to maximize recall while retaining a precision above a desired value. We used $F1_{max}$ for selecting d_c^t in this study unless explicitly stated otherwise. We performed many splits of data into ITB and OOB sets using the approaches described in the Model Assessments section to obtain a large set of OOB data. We then find a single d_c^t to optimize $F1$ for all OOB data together. If all the OOB is OD, then d_c^t is set to less than zero (we effectively use $-\infty$) so that nothing is predicted $\widehat{ID}$. Conversely, d_c^t is set to greater than 1 (we effectively use $+\infty$) if all OOB data is ID so that all data are predicted $\widehat{ID}$. Each value of d was acquired from the KDE built from each X_{ITB} for each split.

Deployment of M^{dom} only relies on $\vec{x}$ of a single data point and can produce $\widehat{ID}/\widehat{OD}$

for that point by checking the d produced by M^{dis}. If $d < d_c^t$, then the prediction is $\widehat{ID}$ and $\widehat{OD}$ otherwise (Eq. 3.5 when d^t=d_c^t). This describes how someone can use M^{dom} for domain prediction after training. Note that this prediction does not involve $E^{|y-\hat{y}|/MAD_y}$, E^{RMSE/σ_y}, E^{area}, nor chemical intuition after the learning process is complete. The overall precision and recall on splits used to find d_c^t can be stored and provided at inference along with $\widehat{ID}/\widehat{OD}$, giving guidance to the user on the confidence they should put in the domain determination.

3.3.2.5 Why not More Complex M^{dis} and M^{dom}?

In this section, we discuss the question "Why use this particular approach to get domain when there are clear opportunities for something potentially more accurate?" The composite function $M^{cdom} = M^{dom}(M^{dis}(\vec{x}))$ takes a feature vector for a given data point and returns a class value of $\widehat{ID}/\widehat{OD}$. In our approach, we used $M^{dis}(\vec{x})$ and a simple cutoff as the learned parameter for M^{dom}. One could easily replace this approach with a full ML model for M^{cdom}, presumably with much more ability to predict ID/OD. We did not do this for two reasons. First, we were concerned that it would be difficult to avoid overfitting a complex ML model for M^{cdom} with limited data. The present choice for M^{cdom} has almost no adjustable parameters and a strong foundation in our understanding that ML models learn best where there is more training data, which protects from overfitting. Second, we wanted to have a simple approach that was easy to use and reproduce by other researchers. We expect that more complex versions of M^{cdom} will be explored in the future, and we hope our work can provide a baseline for such studies, as our work has yielded a simple but promising method, as shown in the Results section.

3.3.3 Defining Ground Truths

We highlighted the absence of an absolute, unique definition for data being ID in our Introduction section. As we are proposing M^{dom} to predict if a data point will be ID and wish to assess its effectiveness, we must provide some precise quantitative ground truth definitions for our OOB data points being ID/OD. We will generally define a

ground truth for an observation to be either ID or OD of a model based on privileged information, by which we mean information never seen during the creation of M^{prop}, M^{unc}, nor M^{dis}. Chemical differences, residuals, $RMSE$s, and accuracy of uncertainty estimates are examples of privileged information and are represented in our study by errors denoted as E^{chem}, $E^{|y-\hat{y}|/MAD_y}$, E^{RMSE/σ_y}, and E^{area}. Each of these errors has an associated cutoff to provide a ground truth ID/OD labeling. The cutoffs for E^{chem}, $E^{|y-\hat{y}|/MAD_y}$, E^{RMSE/σ_y}, and E^{area} are denoted E_c^{chem}, $E_c^{|y-\hat{y}|/MAD_y}$, E_c^{RMSE/σ_y}, and E_c^{area}, respectively. When $E^i < E_c^i$ ($i \in \{chem, |y-\hat{y}|/MAD_y, RMSE/\sigma_y, area\}$), the ground truth label is ID and OD otherwise (Eq. 3.7).

$$\text{Ground Truth} = \begin{cases} ID, & \text{if } E^i < E_c^i \\ OD, & \text{otherwise} \end{cases} \tag{3.7}$$

We outline our rationale behind these sets of privileged information and how we used them to produce our ground truth labels in subsequent sections. Note that there are multiple ways one could use different privileged information and define these ground truths, but we feel these form a logical and broad set of ground truths. The key result of this paper is to show that our dissimilarity measure d (see Eq. 3.4), determined only from the data features, can predict these ground truth ID/OD categorizations and therefore clearly contain essential aspects of privileged information. We demonstrate this ability of d through the assessments outlined in the Model Assessments section.

3.3.3.1 Chemical Intuition

Here we define E^{chem} and associated cutoffs. We define E^{chem} as the mismatch between the chemistries used to build M^{prop} and those seen during deployment of M^{prop}. E_c^{chem} implicitly denotes when chemical mismatch (E^{chem}) becomes sufficiently large such that the physics governing materials defined as ID are vastly different from those that are OD. Both E^{chem} and E_c^{chem} depend on the set of studied materials, their governing physics, and empirical observations. By choosing very similar (e.g., materials selected

from ITB data) and very different (e.g., materials with totally different composition, phases, controlling physics, etc.) materials, it is easy to produce data that are mostly ID/OD. For example, an M^{prop} trained on steel alloys (labeled ID) should not be applicable to polymers (labeled OD). Note that this approach does not require the value for the property being predicted, y. Therefore, we can assess M^{dom} on any data point for which we can write down the feature vector ($\vec{x}$) and have a useful intuition about its chemical similarity to ITB data. We view M^{dom} being successful at delineating ID/OD based on simple chemical intuition as a necessary, but not sufficient, condition to use KDE for domain determination. In other words, if M^{dom} struggles to delineate basic intuitive chemical domains, it is likely that the approach is not particularly useful for domain determination. Specifics on chemical groups, which implicitly define E_c^{chem}, are covered in the Data Curation Section.

3.3.3.2 Normalized Absolute Residuals and $RMSE$

Here we define the cutoffs $E_c^{|y-\hat{y}|/MAD_y}$ and E_c^{RMSE/σ_y} as the associated error metrics were already defined (see Eqs. 3.1 and 3.2). For an M^{prop} whose predictions represent the mean of all y (denoted as $\bar{y}$), both $E^{|y-\hat{y}|/MAD_y}$ and E^{RMSE/σ_y} are 1.0 as calculated from Eqs. 3.1 and 3.2. We consider this "predicting the mean" to be a baseline naïve model with respect to which any reasonable M^{prop} model should perform better. If a fit M^{prop} yields better predictions than a baseline, we considered those data to be ID, otherwise, those data were considered OD (Eq. 3.7). This defines $E_c^{|y-\hat{y}|/MAD_y}$=E_c^{RMSE/σ_y}=1. Using either $E_c^{|y-\hat{y}|/MAD_y}$ or E_c^{RMSE/σ_y} as the ground truth cutoff for ID/OD labeling captures the widely invoked idea that ID cases should be predicted better compared to the OD cases. These definitions require residuals and can therefore be assessed on data with X and y for M^{prop}.

3.3.3.3 Errors in Predicted Uncertainties

Here we define the cutoff E_c^{area} as the associated error metric was already defined (see Eq. 3.3). Similar to $E_c^{|y-\hat{y}|/MAD_y}$ and E_c^{RMSE/σ_y}, we set E_c^{area} such that M^{unc} will show

better performance than a naïve baseline case. For this case, we assume a baseline M^{prop} model that provides $\bar{y}$ as the prediction for all cases and a baseline M^{unc} that provides the standard deviation of the training target values, σ_y, as the uncertainty for all cases. These baseline predictions can be used in Eq. 3.3 to measure E^{area} for any given set of data, and we take this predicted baseline E^{area} to be E_c^{area}. We defined ID cases as those for which the $E^{area} < E_c^{area}$ (OD otherwise). Unlike both $E_c^{|y-\hat{y}|/MAD_y}$ and E_c^{RMSE/σ_y}, E_c^{area} changes depending on the evaluated set of y and ranged from approximately 0.2 to 0.3 in our study. In other words, Eq. 3.3 does not yield a single number for naïve models.

It is worth relating that this criterion of $E^{area} < E_c^{area}$ for a set of data points tells us that M^{unc} is more accurate than the naïve baseline, but not that M^{prop} has small residuals. If M^{prop} has high E^{RMSE/σ_y} on a set of data points but M^{unc} has accurate estimates of those residuals, then M^{unc} clearly knows significant and useful information about the data, and the data is therefore in some sense ID. This situation can be contrasted with the case where M^{prop} has high residuals and M^{unc} has very inaccurate estimates of those residuals, in which case M^{prop} and M^{unc} appear to know nothing about the data, and the data is therefore reasonably considered OD. The extent a quality ID/OD ground truth definition can be defined by just E^{area}, whether E^{area} needs to be combined with the other errors considered, or even whether E^{area} should be excluded entirely from consideration is not established at this point. Regardless, we show that M^{dom} provides an excellent ability to predict an ID/OD ground truth label based on E^{area}.

3.3.3.4 Summary of Ground Truth Definitions

The above detailed definitions of ID/OD can be confusing, so we summarize the idea again here. We considered four intuitive ways of thinking about being ID/OD of a trained model based on intuitive chemical similarity to ITB data, model residuals, model $RMSE$s, and model uncertainty estimates. For each way of thinking, we defined a score of the closeness of the data to being ID, which we denoted E^{chem}, $E^{|y-\hat{y}|/MAD_y}$, E^{RMSE/σ_y}, and E^{area}, respectively. We then used chemical intuition (for chemical dissimilarity) or naïve baseline model behavior (for model residuals and model uncertainty

estimates) to define a cutoff (denoted E_c^i) for each corresponding E^i ($i \in \{chem, |y - \hat{y}|/MAD_y, RMSE/\sigma_y, area\}$) such that it was reasonable to assume that $E^i < E_c^i$ meant data were ID (Eq. 3.7). This allowed us to assign the labels of ID/OD for data so that we could evaluate the utility of d. It is important to realize that these ground truth definitions make extensive use of privileged information (i.e., information not expected to be available during application of the model). However, these ground truth definitions make no direct use of d and d makes no use of the privileged information. The value of our results is that we gain access to ID/OD measures that require privileged information by using d.

We note that while E^{chem} relies on a researcher's intuition regarding the chemical field and is specific to given fields, definitions relying on $E^{|y-\hat{y}|/MAD_y}$, E^{RMSE/σ_y}, and E^{area} are automated numerical approaches and quite general. Therefore, the success of M^{dom} on methods relying on $E^{|y-\hat{y}|/MAD_y}$, E^{RMSE/σ_y}, and E^{area} suggests our approach is not limited to materials and chemistry problems, but that in general d gives valuable access to the domain implications of knowledge of residuals. Note that the way we define E^{RMSE/σ_y} and E^{area} for our automated numerical ground truths require averaging over groupings based on d and is covered in the next section. Thus, there is a very modest and indirect path by which the nature of d can influence the ground truth categorization based on E^{RMSE/σ_y} and E^{area}. Although we think this effect is modest, it represents a path of data leakage that could potentially lead to bias in the assessments of the approach. To counteract this concern, we have included $E^{|y-\hat{y}|/MAD_y}$, which does not require any binning and is immune to data leakage. This quantity has other limitations as it is quite stochastic, but our success on assessments using the ground truth categorization from E_c^{chem} and $E_c^{|y-\hat{y}|/MAD_y}$ demonstrates that the other successes were not dominantly influenced by data leakage from binning on d. In the next section, we outline assessments for each definition of ground truth.

3.3.4 Model Assessments

Now that we have defined ground truths, we can assess how well M^{dom}, or equivalently, categorizing the data based on d (Eq. 3.5), can predict ID/OD. Specifically, we evaluate the ability of d obtained from M^{dis} to predict domain labels by building precision-recall curves on OOB data sets. For each precision-recall curve, a naïve baseline area under the curve (AUC) was constructed by considering a baseline model that predicts ID for every case, which yields a baseline AUC of the ratio of ID cases over the total number of cases [78]. The difference between the true and baseline AUC was used as an assessment metric, which we call AUC-Baseline. AUC was calculated using the sklearn.metrics.average_precision_score from scikit-learn [42]. AUC-Baseline shows how much additional information d provided for domain classification above the naïve baseline model. Any AUC-Baseline above zero indicates domain information retained by d. The values of precision, recall, and $F1$ were reported for the acquired metric of $F1_{max}$, which evaluates how effectively the ID points were separated from the OD points by the dissimilarity measure d.

3.3.4.1 Assessing Our Domain Prediction Based on A Ground Truth Determined by Chemical Intuition

In this section, we describe our method to evaluate how d provides information with respect to E^{chem} and call this assessment A^{chem}. We started with a set of ITB data that were used to construct M^{prop} and additional data with chemically distinct groups. These data are called the Original and non-Original sets, respectively. The non-Original set can be further subdivided into other chemistries and was always treated as OOB. Each case from all data was labeled as ID/OD based on chemical intuition. We wanted to evaluate the ability of d to discern ID/OD from groups of chemistries that are OOB. To do this, we built sets of OOB data that contained ID/OD labels. For every observation in the Original set, i, the following sets of steps were performed: i was taken out of the Original set (i.e., placed in the OOB data), an M^{dis} model was trained on the remaining Original ITB data, then M^{dis} was used to calculate d on all OOB data (i.e., i and all cases from

the non-Original set). This formed one prediction set. We repeated the procedure for all data points in the Original set and aggregated that data. In other words, we performed Leave-One-Out (LOO) CV on the Original set while treating the non-Original set as an OOB set for every iteration. If there are n cases from the Original set, then there should be n different models and prediction sets by the time the procedure finished. See Fig. 3.1a for a diagram of the splitting procedure. Violin plots were generated to show the distribution of d values for each chemical group.

This methodology often produced a large class imbalance between the number of ID and OD cases, which can make interpreting the assessment of the domain classifier difficult. Therefore, we used a resampling procedure to obtain a balanced number of ID and OD data points in the OOB aggregated set. More specifically, if the number of ID cases exceeded the number of OD cases, we randomly sampled a subset of the ID data such that the number of ID cases was equal to the number of OD cases. Conversely, if the number of OD cases surpassed the number of ID cases, we randomly sampled a subset of the OD data to match the number of ID cases. However, if the ID and OD data sets contained an equal number of cases, no subsampling was performed. These samplings of data were then used to assess the ability of d to predict ID/OD with precision and recall.

3.3.4.2 Assessing Our Domain Prediction Based on A Ground Truth Determined by Normalized Residuals and Errors in Predicted Uncertainties

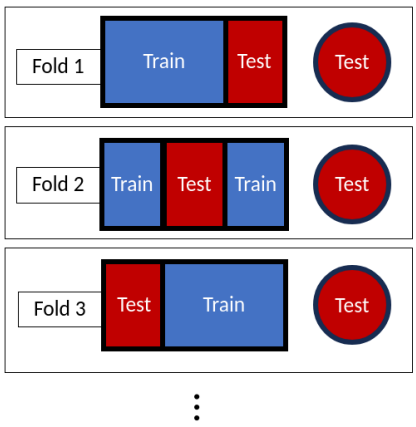
In this section we describe our method to evaluate how d provides information with respect to $E^{|y-\hat{y}|/MAD_y}$, E^{RMSE/σ_y}, and E^{area} and call these assessments $A^{|y-\hat{y}|/MAD_y}$, A^{RMSE/σ_y} and A^{area}, respectively. We must generate a set of OOB data that contain ID/OD labels to assess the ability of d to separate ID/OD. To do this, data containing both X and y were split using several methods for CV. First, data were split by 5-fold CV where models were iteratively fit on 4 folds and then predicted $\hat{y}$, σ_c, and d on OOB data. Second, data were pre-clustered using agglomerative clustering from scikit-learn

[42]. One cluster was left out for OOB data, and all other ITB clusters were used to train models. Then, models predicted $\hat{y}$, σ_c, and d on the OOB data. Like the 5-fold CV methodology, we sequentially left out each cluster, which was similar to the approaches applied in Refs. [142, 143]. However, applying this clustering to the original data gave only a limited number of clusters. To generate many more OOB data, we generated a series of bootstrap data sets from clusters and then applied the same leave out cluster approach. To clarify, we first clustered the data, and then performed bootstrapping separately on each cluster, rather than bootstrapping on all the data and then clustering. We call this a Bootstrapped Leave-One-Cluster Out (BLOCO) approach. We performed BLOCO with 3 and 2 total clusters with the aim of providing a large amount of OOB data that are increasingly dissimilar to ITB data and hopefully OD. See Fig. 3.1b for an illustration of BLOCO.

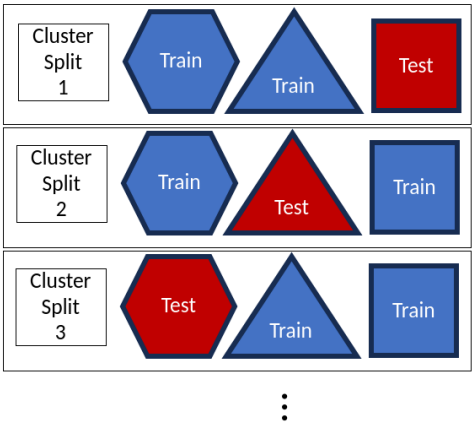
Both 5-fold CV and the BLOCO splitting strategies were repeated 5 times for nested CV on most data and model combinations. Only BNN models had the number of repeats decreased to 3, 3, and 2 for the Fluence, Friedman, and Superconductor data sets, respectively. The amount of RAM consumed by BNNs for these larger data sets lead to a practical limitation on acquirable statistics. The workflow for producing the OOB data is shown in Fig. 3.1c. OOB data were aggregated and binned with respect to d into N bins with equal (or close to equal) number of points. If data could not be divided evenly (e.g., many repeated points with d=1), then the extra points went to the bin with the highest average d. Our choice of $N = 10$ gave robust results. The number of bins should strike a balance between being small enough for each bin to encompass a substantial number of data points for reliable statistics, yet large enough to effectively discern shifts in trends across different bins of OOB data. E^{RMSE/σ_y} was calculated for all OOB data in each bin. We subsequently compared $O(z)$ to $Phi(0, 1)$ for each bin using Eq. 3.3, which provided E^{area} for each bin. The measure of $E^{|y-\hat{y}|/MAD_y}$ did not require binning and was measured for each individual data point. Note that normalization of $E^{|y-\hat{y}|/MAD_y}$ and E^{RMSE/σ_y} used the MAD_y and σ_y, respectively, from ITB data, not the OOB data. To avoid data leakage, we prevented models from learning on OOB data by incorporat-

ing information solely from ITB data during the model development phase, which may yield slightly different values for MAD_y and σ_y according to the splits considered. Class labels were acquired, $E^{|y-\hat{y}|/MAD_y}$ versus d plots were generated, E^{RMSE/σ_y} versus d plots were generated, E^{area} versus d plots were generated, and precision and recall scores were recorded. Each M^{dom} for $E^{|y-\hat{y}|/MAD_y}$, E^{RMSE/σ_y}, and E^{area} were built similarly. Any M^{dom} model checks the value of M^{dis} and returns a corresponding $\widehat{ID}/\widehat{OD}$ value.

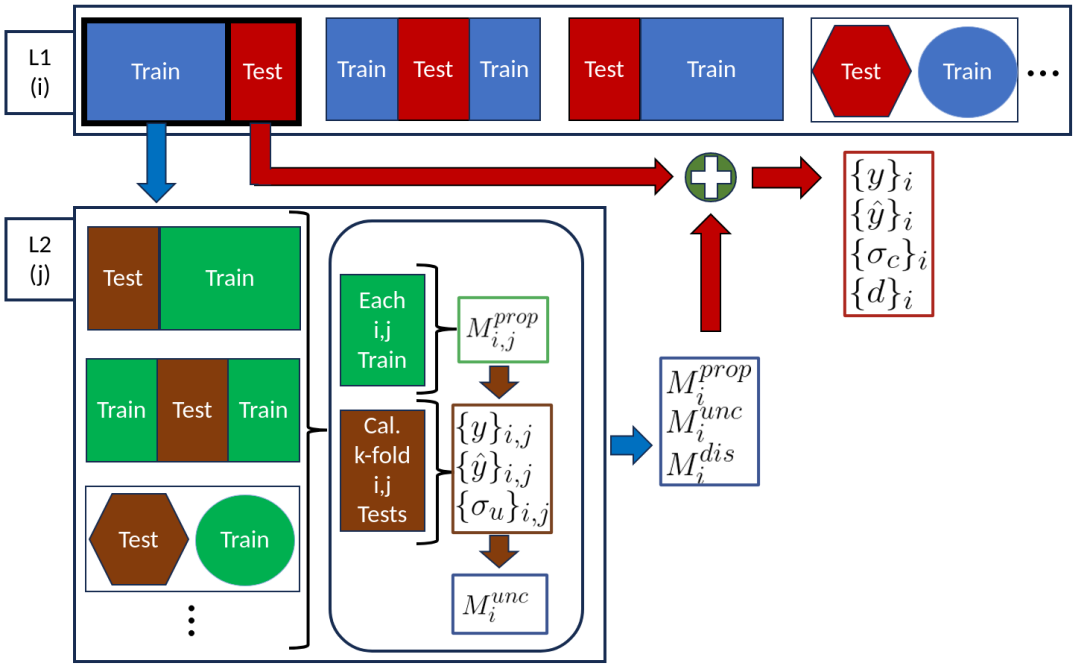
Unfortunately, the above steps meant that the choice of N impacted which data points were averaged together and therefore impacted the values of E^{RMSE/σ_y} and E^{area} and the grid of points on which d can be evaluated. Thus, N impacts both this assessment of M^{dom} and the M^{dom} that would be developed for a model to be used at inference. While we have found that using 10 bins can be effective, it is important to note that this choice may not necessarily be optimal for these or general models. Future work should establish an approach for this binning that automatically selects N to obtain optimal results, or avoids binning altogether in the determination of d_c^t for M^{dom}. Note that $E^{|y-\hat{y}|/MAD_y}$ is unaffected by N.



(a) Generating OOB for A^{chem}



(b) Generating OOB from BLOCO



(c) Nested CV

Figure 3.1: **Methods to generate OOB data.** Shown in Fig. 3.1a is the splitting methodology for A^{chem}. The square and circle sets are from LOO CV and a fixed OOB set, respectively. Shown in Fig. 3.1b is the BLOCO splitting methodology. Each shape represents a specific cluster. Clusters were iteratively swapped between ITB and OOB sets. Shown in Fig. 3.1c is the depiction of nested CV. The upper level (L1) was used to produce ITB (blue Train) and OOB (red Test) data from k-fold and BLOCO splits. From each L1 Train, we further divided data in a nested manner (L2). From L2, we calibrated model uncertainties on a set of k-fold splits and produced M_i^{unc}. M_i^{prop} and M_i^{dis} were fit to all Train for each split in L1. The predictions of M_i^{prop}, M_i^{unc} and M_i^{dis} on each respective L1 Test set produced the OOB data used to measure $E^{|y-\hat{y}|/MAD_y}$, E^{RMSE/σ_y}, and E^{area}.

3.3.5 Data Curation

We applied all previous definitions of domain and assessments with five data sets consisting of four physical data sets from the field of materials science and one synthetic data set (all discussed in detail below). Among these, the four physical property data sets exhibit well-defined chemical domains, with samples that are readily categorized as ID/OD. The assigned categories are shown in Table 3.2. For three of the physical data sets, the data curation process involved generation of features, followed by a careful down-selection of a relevant X that held significance for y (see Appendix for the feature learning curves). It is important to remember that certain groups were denoted as Original and Perturbed Original across various data sets. It is essential to interpret these labels within the specific context of the material data set under investigation.

3.3.5.1 Diffusion Data Set

Diffusion activation energies (y) of single atom (dilute solute) impurities in metal hosts were acquired from Refs. [144, 145]. The activation energies were calculated using Density Functional Theory (DFT) methods. The total number of host-impurity pairs are 408. We refer to these compositions as the Original set. We generated host and impurity combinations across many chemical groups in the periodic table, but made sure to only include compositions outside the Original data set (no y are available). Chemical groups were assigned ID/OD based on intuition (see Table 3.2). The Original group was labeled ID because they were data used to build M^{prop} models. Host metals for the Original group include elements in the following groups: alkaline earth, transition, and post-transition metals. Any data from the aforementioned groups that were not included in the Original data are considered to be ID due to their chemical proximity and shared physical properties. None of the remaining chemical groups in Table 3.2 were included as the host element in our data, and are therefore considered to be OD. Furthermore, we included a Manual group which mixed elements across chemical groups designated as [host][impurity] and are as follows: $[Fe][O]$, $[Fe][Cl]$, $[Na][Cl]$, $[Al][Br]$, $[Ca][P]$, and $[Sr][I]$. This provided another OD group to test our d measure against. Mixing elements

Table 3.2: Tabulated are the ground truth labels for A^{chem}.

Data Set	Chemical Group	Material	Label
Diffusion			
	Original	Many	ID
	Alkaline earth metals	Many	ID
	Transition metals	Many	ID
	Post-transition metals	Many	ID
	Metalloids	Many	OD
	Alkali metals	Many	OD
	Reactive nonmetals	Many	OD
	Noble gases	Many	OD
	Lanthanides	Many	OD
	Actinides	Many	OD
	Manual	Many	OD
Steel Strength and Fluence			
	Original	Many	ID
	Perturbed Original	Many	ID
	Copper-Based	$Cu98.05Be1.7Co0.25$	OD
	Copper-Based	$Cu85Zn15$	OD
	Copper-Based	$Cu89.8Sn10P0.2$	OD
	Copper-Based	$Cu92Ni4Sn4$	OD
	Copper-Based	$Cu85Zn5Pb5Sn5$	OD
	Aluminum-Based	$Al95Si5$	OD
	Aluminum-Based	$Al90Mg10$	OD
	Aluminum-Based	$Al92.5Mg1.5Ni2Cu4$	OD
	Aluminum-Based	$Al83.5Si12Mg1Ni2.5Cu1$	OD
	Aluminum-Based	$Al88Cu3.5Si8.5$	OD
	Iron-Based-Far	$Fe99.1C0.2Mn0.45Si0.25$	OD
	Iron-Based-Far	$Fe96.49C0.21Mn0.75Si0.25Ni2.3$	OD
	Iron-Based-Far	$Fe60.6Ni35Si2C2.4$	OD
	Iron-Based-Far	$Fe42.45Cr18Ni39C0.55$	OD
	Iron-Based-Far	$Fe63Cr28Ni9$	OD
Cuprates			
	Cuprates	Many	ID
	Iron-Based	Many	OD
	Low-T_c	Many	OD
Iron-Based			
	Iron-Based	Many	ID
	Cuprates	Many	OD
	Low-T_c	Many	OD
Low-T_c			
	Low-T_c	Many	ID
	Iron-Based	Many	OD
	Cuprates	Many	OD

across the periodic table groups created a set of data with distinct physics compared to our ID data.

Features were generated using the MAST-ML elemental property feature generator (named ElementalFeatureGenerator in the package) [86]. The resulting features consist of the composition average, arithmetic average, maximum, and minimum values of elemental features for the host and impurities. While the DFT calculations used to create this database investigated dilute solutes in a metallic host, the host and impurity elements were weighted equally, following previous work in Refs. [144, 145]. The EnsembleModelFeatureSelector from MAST-ML was used to reduce the number of features to 25 (X) for the data containing y. We select the same X for data without y. The X we study has a strong relation with the y of interest. We do not concern ourselves with possible overfitting from feature selection on all the data as our aim is not to make a maximally robust model but to study how well we can predict the domain of the model. We refer to data in this section as "Diffusion".

3.3.5.2 Fluence Data Set

Data of Reactor Pressure Vessel (RPV) steel embrittlement were acquired from Ref. [146] and described in detail from the work in Ref. [147], where the ductile-to-brittle transition temperature shift, DT41J [C], is y. The total number of cases is 2,049. These steels are over 96.5 wt% Fe with small amounts of alloying additions less than around 3.5 wt%. For materials that should be ID, each element from each material of the Original set was incremented by ± 0.01 and then normalized such that the sum of element fractions is 1. These data have X that were slightly perturbed from the Original set and are called Perturbed Original. For OD data, a subset of materials without y from Ref. [148] were added (Table 3.2). These are randomly chosen metal alloys that have been previously manufactured, but contain a majority of either Al, Cu, or Fe. These data should be OD since much of the data contain elements not present in the Original set and contain less Fe in all cases except one. The exception has the element Be, which is not in our Original set. For these data, elemental feature generation was not needed and thus not conducted.

Instead, the features for the Fluence data include the weight percentages of elements in each alloy (Fe, Cr, Al, Be, Co, Si, Mn, Zn, Sn, Pb, Ni, P, Cu, Mg, and C). Additional features include irradiation fluence (in *log* values), flux (in *log* values), and temperature for a total of 18 features used for A^{chem}. Note that this feature set is a natural extension of the successfully used features in Ref. [147]. For cases that do not provide irradiation fluence or flux, we performed mean imputation. For $A^{|y-\hat{y}|/MAD_y}$, A^{RMSE/σ_y}, and A^{area}, only the features that are chemically relevant for the Original set were used. Features of elemental weights for elements not seen in the Original set were dropped since the feature columns would be equal to zero for all data points. We also omitted the weight percent of Fe because it trivially represented the residual weight percent in the alloy composition. In other words, the weight percentages for Cu, Ni, Mn, P, Si, and C were kept with fluence, flux, and temperature, which constitute 9 features. We refer to data in this section as "Fluence".

3.3.5.3 Steel Strength Data Set

Data of steel strengths were acquired from Ref. [149] and our y is yield strength. The total number of cases is 312. These alloys are majority Fe with the minimum (maximum) percent of Fe being 62% (86%). The minimum (maximum) number of alloying components is 10 (13). Similar to our Fluence data, we create an ID set of cases by following the same elemental perturbation method. The same cases of OD materials from Table 3.2 were used as OD. The OD alloys either have majority Cu, Fe, or Al, or, if they have majority Fe, then the Fe percent is generally outside the range of our Original data (i.e., either Fe alloying percent is less or greater than the Fe percent in Ref. [149]). Furthermore, the minimum number of alloying components of 10 for ID alloys is much greater than the maximum of 5 for OD alloys. Similar to our study of the Diffusion data set, we used MAST-ML to generate elemental features [86]. However, we have the percent composition of constituent elements, so we included the weighted averages of properties based on elemental fractions. Application of the same feature selection method as used on the Diffusion data yielded 15 final features (X). We refer to data in this section as

"Steel Strength".

3.3.5.4 Superconductor Data Set

Material compositions and superconducting critical temperatures were acquired from Ref. [150] and totaled 6,253 cases. Our y was the maximum temperature at which a material is capable of superconduction, T_c. We split our data into three sets for A^{chem} according to Ref. [150]. These sets are cuprates (called Cuprates), materials containing Fe (called Iron-Based), and materials left out by the previous two classifications (called Low-T_c). Stanev et al. have demonstrated that M^{prop} models trained exclusively on one of the aforementioned material classes cannot accurately predict T_c for the others [150]. We compared d from M^{dis} on models trained only on subsets of each aforementioned material class and how it compared to the other two. As an example, consider Cuprates. We perform A^{chem} treating Cuprates as ID, Iron-Based as OD and Low-T_c as OD (see Table 3.2). Features were generated and selected similar to our Steel Strength data which yielded 25 features (X). We refer to data in this section as "Superconductor".

3.3.5.5 Friedman Data Set

Most data considered in our study are from the physical sciences. To test the generalizability of our developed methods and assessments, we also study a synthetic data set constructed from Eq. 3.8.

$$y = 10sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 \tag{3.8}$$

Eq. 3.8 is a Friedman function from Ref. [151]. Each x_i is the feature for the dimension i. Like the original Friedman data set, each x_i was generated from a uniform distribution on the $[0.0, 1.0)$ interval to generate y. We chose 500 samples. Because we wanted distinct clusters to exist in our data, we also uniformly generated 500 points for each x_i from the following intervals: $[0.0, 0.2)$, $[0.2, 0.4)$, $[0.4, 0.6)$, $[0.6, 0.8)$, and $[0.8, 1.0)$. Data generated from these intervals do not overlap, effectively creating distinct subspaces. For example,

a sample where $x_1 = 0.1$ and $x_2 = 0.4$ cannot exist for the exclusive intervals. We call these data "Friedman". Additionally, we generated 3,000 points on the $[0.0, 1.0)$ interval to explain one of our failure modes discussed in the Notes of Caution for Domain Prediction section. We call these data Friedman WithOut Distinct Clusters (FWODC). Note that FWODC is only used in the context of explaining when our method fails. No feature selection was performed because the 5 features completely explain y.

Results and Discussion

Now that the conceptual tools and methodology used in this study are established, we can apply those concepts to various data sets and models. We start with A^{chem} and show that d mostly separated OD materials from ID materials. Although the separation is not perfect in many cases, it provided a clear way to flag OD materials. We then cover more automated numerical methods which do not require a priori knowledge of domains from chemical intuition. We show that d can be used to separate cases with high $E^{|y-\hat{y}|/MAD_y}$ (i.e., data that are OD) and cases with lower $E^{|y-\hat{y}|/MAD_y}$ (i.e., data that are ID). A similar observation was made with the separation of high and lower E^{RMSE/σ_y} and E^{area} groups of data, respectively. We end by providing notes of caution for use cases where developed methods may not yield desired behavior. All results can be seen in Table 3.3.

Table 3.3: Classification metrics are tabulated for A^{chem}, $A^{|y-\hat{y}|/MAD_y}$, A^{RMSE/σ_y}, and A^{area}. The d_c^t, precision, recall, and $F1$ are for $F1_{max}$. A^{chem} entries do not require M^{prop} and are left empty.

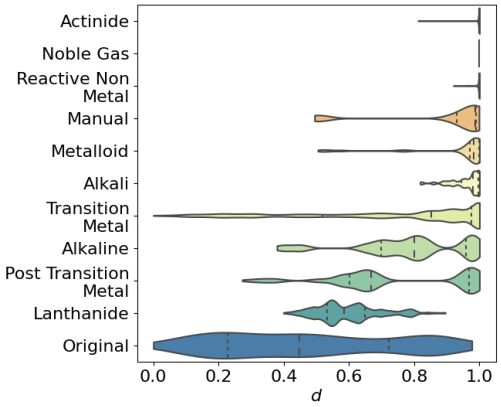
Data	M^{prop}	Assessment	Baseline	AUC	AUC-Baseline	d_c^t	Precision	Recall	$F1$
Diffusion		A^{chem}	0.50	0.68	0.18	1.00	0.64	0.80	0.71
Steel Strength		A^{chem}	0.50	0.99	0.49	1.00	1.00	0.98	0.99
Fluence		A^{chem}	0.50	1.00	0.50	0.99	1.00	1.00	1.00
Cuprates		A^{chem}	0.50	0.97	0.47	0.99	0.94	0.98	0.96
Iron-Based		A^{chem}	0.50	0.84	0.34	0.97	0.58	0.98	0.73
Low-T_c		A^{chem}	0.50	0.96	0.46	0.80	0.98	0.80	0.88
Diffusion	BNN	$A^{|y-\hat{y}|/MAD_y}$	0.53	0.90	0.37	1.00	0.85	0.86	0.86
Diffusion	BOLS	$A^{|y-\hat{y}|/MAD_y}$	0.59	0.93	0.34	0.99	0.95	0.83	0.88
Diffusion	BSVR	$A^{|y-\hat{y}|/MAD_y}$	0.48	0.96	0.48	1.00	0.88	0.98	0.93
Diffusion	RF	$A^{|y-\hat{y}|/MAD_y}$	0.57	0.95	0.38	1.00	0.96	0.89	0.92

Fluence	BNN	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.65	0.92	0.26	1.00	0.88	0.91	0.89
Fluence	BOLS	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.71	0.86	0.15	1.00	0.71	1.00	0.83
Fluence	BSVR	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.63	0.86	0.23	1.00	0.80	0.82	0.81
Fluence	RF	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.74	0.90	0.16	1.00	0.74	1.00	0.85
Friedman	BNN	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.71	0.97	0.27	1.00	0.98	0.89	0.93
Friedman	BOLS	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.75	0.96	0.21	1.00	0.94	0.89	0.91
Friedman	BSVR	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.64	0.98	0.34	1.00	0.99	0.91	0.95
Friedman	RF	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.73	0.95	0.22	1.00	0.93	0.86	0.89
Steel Strength	BNN	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.49	0.68	0.19	1.00	0.64	0.80	0.71
Steel Strength	BOLS	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.45	0.69	0.25	1.00	0.62	0.87	0.72
Steel Strength	BSVR	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.49	0.62	0.13	1.00	0.49	1.00	0.66
Steel Strength	RF	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.64	0.83	0.19	1.00	0.64	1.00	0.78
Superconductor	BNN	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.57	0.74	0.18	1.00	0.57	1.00	0.72
Superconductor	BOLS	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.43	0.61	0.19	0.99	0.61	0.68	0.65
Superconductor	BSVR	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.62	0.70	0.08	1.00	0.62	1.00	0.77
Superconductor	RF	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.62	0.79	0.17	1.00	0.62	1.00	0.77
Diffusion	BNN	A^{RMSE/σ_y}	0.47	1.00	0.53	0.96	1.00	1.00	1.00
Diffusion	BOLS	A^{RMSE/σ_y}	0.53	1.00	0.47	1.00	1.00	1.00	1.00
Diffusion	BSVR	A^{RMSE/σ_y}	0.53	1.00	0.47	1.00	1.00	1.00	1.00
Diffusion	RF	A^{RMSE/σ_y}	0.53	1.00	0.47	1.00	1.00	1.00	1.00
Fluence	BNN	A^{RMSE/σ_y}	0.64	1.00	0.36	1.00	1.00	1.00	1.00
Fluence	BOLS	A^{RMSE/σ_y}	0.57	0.99	0.41	1.00	0.89	1.00	0.94
Fluence	BSVR	A^{RMSE/σ_y}	0.43	1.00	0.57	0.91	1.00	1.00	1.00
Fluence	RF	A^{RMSE/σ_y}	0.64	1.00	0.36	1.00	1.00	1.00	1.00
Friedman	BNN	A^{RMSE/σ_y}	0.65	1.00	0.35	1.00	1.00	1.00	1.00
Friedman	BOLS	A^{RMSE/σ_y}	0.71	1.00	0.29	1.00	1.00	1.00	1.00
Friedman	BSVR	A^{RMSE/σ_y}	0.62	1.00	0.38	1.00	1.00	1.00	1.00
Friedman	RF	A^{RMSE/σ_y}	0.67	1.00	0.33	1.00	1.00	1.00	1.00
Steel Strength	BNN	A^{RMSE/σ_y}	0.47	1.00	0.53	0.88	1.00	1.00	1.00
Steel Strength	BOLS	A^{RMSE/σ_y}	0.36	0.96	0.61	0.78	0.83	1.00	0.91
Steel Strength	BSVR	A^{RMSE/σ_y}	0.20	1.00	0.80	0.43	1.00	1.00	1.00
Steel Strength	RF	A^{RMSE/σ_y}	0.47	1.00	0.53	0.85	1.00	1.00	1.00
Superconductor	BNN	A^{RMSE/σ_y}	0.36	1.00	0.64	0.97	1.00	1.00	1.00
Superconductor	BOLS	A^{RMSE/σ_y}	0.36	1.00	0.64	0.97	1.00	1.00	1.00
Superconductor	BSVR	A^{RMSE/σ_y}	0.54	0.85	0.31	0.97	1.00	0.67	0.80
Superconductor	RF	A^{RMSE/σ_y}	0.45	0.78	0.34	0.93	1.00	0.61	0.76
Diffusion	BNN	A^{area}	0.06	0.31	0.25	0.31	0.50	1.00	0.67
Diffusion	BOLS	A^{area}	0.53	0.50	-0.03	1.00	0.53	1.00	0.69
Diffusion	BSVR	A^{area}	0.18	1.00	0.82	0.46	1.00	1.00	1.00
Diffusion	RF	A^{area}	0.18	1.00	0.82	0.46	1.00	1.00	1.00
Fluence	BNN	A^{area}	0.43	1.00	0.57	0.92	1.00	1.00	1.00
Fluence	BOLS	A^{area}	0.29	0.43	0.14	0.97	0.57	1.00	0.73
Fluence	BSVR	A^{area}	0.43	1.00	0.57	0.91	1.00	1.00	1.00
Fluence	RF	A^{area}	0.43	1.00	0.57	0.92	1.00	1.00	1.00
Friedman	BNN	A^{area}	0.21	0.30	0.09	0.98	0.43	1.00	0.60
Friedman	BOLS	A^{area}	0.00	0.00	0.00	$-\infty$	0.00	0.00	0.00
Friedman	BSVR	A^{area}	0.23	1.00	0.77	0.34	1.00	1.00	1.00

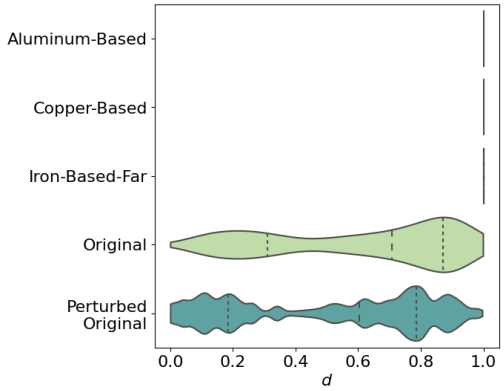
Friedman	RF	A^{area}	0.29	1.00	0.71	0.65	1.00	1.00	1.00
Steel Strength	BNN	A^{area}	0.13	0.13	-0.01	0.99	0.22	1.00	0.36
Steel Strength	BOLS	A^{area}	0.14	0.29	0.15	0.78	0.33	1.00	0.50
Steel Strength	BSVR	A^{area}	0.07	0.31	0.24	0.30	0.50	1.00	0.67
Steel Strength	RF	A^{area}	0.33	1.00	0.67	0.65	1.00	1.00	1.00
Superconductor	BNN	A^{area}	0.18	0.32	0.14	0.99	0.40	1.00	0.57
Superconductor	BOLS	A^{area}	0.27	0.90	0.63	0.97	0.75	1.00	0.86
Superconductor	BSVR	A^{area}	0.09	0.31	0.22	0.56	0.50	1.00	0.67
Superconductor	RF	A^{area}	0.27	1.00	0.73	0.93	1.00	1.00	1.00

Relationship Between Chemical Dissimilarity (E^{chem}) and Distance (d) from the Chemical Assessment (A^{chem})

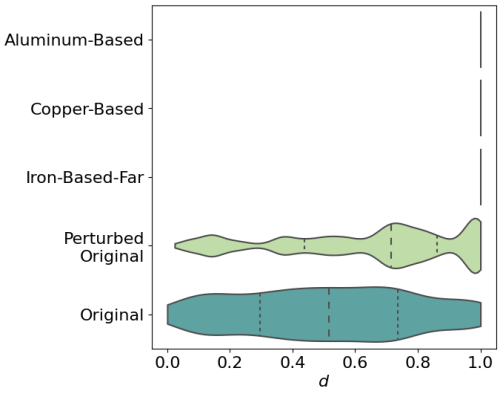
For all chemical data sets analyzed through A^{chem}, violin plots for d with respect to chemical groups were generated and shown in Fig. 3.2. A positive result shows that ID materials have lower values of d compared to OD materials. Values to the left are more likely to be observed (i.e., be ID), and values to the right become less likely to be observed (i.e., be OD). All sets of A^{chem} generally show the aforementioned trend. Data are organized based on their median d values, with lower values positioned at the bottom and higher values at the top in Fig. 3.2. The classification metrics are tabulated in Table 3.3. Our $F1_{max}$ scores range from 0.71 to 1.00, which are rather high. We show how d is useful for flagging OD materials from predictions in subsequent text.



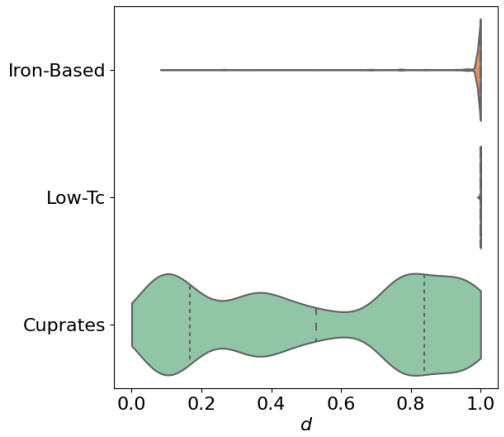
(a) Diffusion



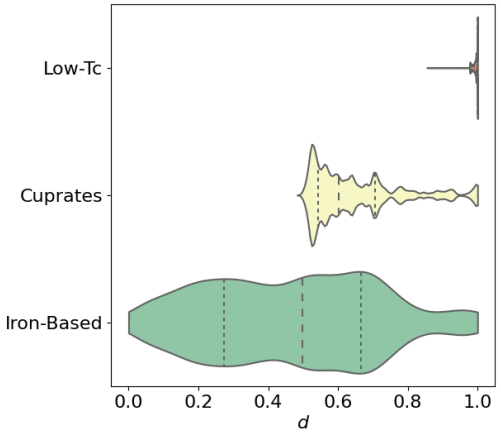
(b) Fluence



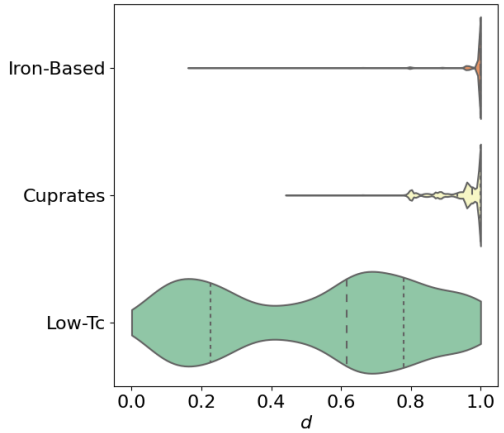
(c) Steel Strength



(d) Cuprates



(e) Iron-Based



(f) Low-T_c

Figure 3.2: **KDE separates distinct materials.** We show the violin plot for all the d scores separated by chemical groups. The first, second, and third vertical lines within each violin denote the separations between the first, second, third, and fourth quartiles. Values to the left are more likely to be observed compared to values to the right. All violins were forced to have the same width for visual purposes (i.e., the actual number of observations are not reflected by the visual).

The violin plot for the Diffusion data is shown in Fig. 3.2a. Intuitively dissimilar materials to the Original data subset, such as noble gases, are to the very right of the figure. Materials that were mixed across chemical groups (Manual set) also show high dissimilarities compared to the Original set. Conversely, more similar materials like transition metals, alkaline earths, and post transition metals have their median d value closer to the Original subset compared to the aforementioned groups. Lanthanides, however, appear closer to our Original set than expected. If we want 95% of data to come from a trustworthy chemical domain while maximizing recall for the Diffusion data, then the corresponding threshold of d_c^t=0.45 can act as a decision boundary. Using this threshold for M^{dom}, chemical groups such as noble gases, actinides, reactive non-metals, etc. are mostly excluded from $\widehat{ID}$, but our recall is only 0.20. At a value slightly below 1.00 for d_c^t, our $F1$ score is maximized and can still exclude most noble gases, reactive non-metals, and actinides. At $F1_{max}$, our recall and precision are 0.80 and 0.64 respectively. One can increase the precision of predictions by choosing a lower value of d_c^t at the cost of recall and vice versa according to the needs of M^{dom} application.

We have a similar observation for our Fluence data (see Fig. 3.2b). Note that the Perturbed Original set have their quartiles shifted to the left of the Original data, meaning that values from the Perturbed Original data are predicted by the KDE to be more likely to be observed than test cases from the Original data, which is surprising. This is likely just an anomaly due to the modest size of the data and is not a concern given that both sets of data are ID. The important consideration is that all OD data have d values at or near 1.00. All OD data are far away from ID data. Any prediction with $d_c^t \approx 0.99$ is deemed to be untrustworthy and separates ID/OD essentially perfectly.

As for our Steel Strength data in Fig. 3.2c, the Perturbed Original data have a median d value further to the right than the Original set, as expected. Most Perturbed Original data from the Steel Strength data are to the left of other OD groups. In other words, OD data are further away than ID data. The Iron-Based-Far data have cases that are closer to the Original data than the Copper-Based and Aluminum-Based sets, although it is not evident from the figure. The observation is reasonable considering most of the

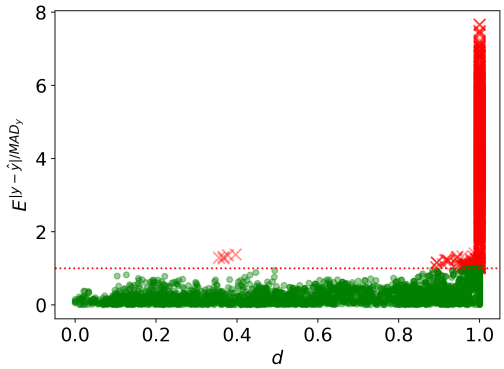
Original data are iron based, but outside the range of Fe weight percentages in the Iron-Based-Far data. Like our Fluence data, any prediction at or near $d_c^t \approx 1.00$ was deemed to be untrustworthy. But unlike Fluence, ID/OD labels were not separated perfectly. Our recall is 0.98 because of a few FN predictions and the precision is 1.00.

Regarding the Superconductor data, we divided data into Cuprates, Iron-Based, and Low-T_c groups and assessed the behavior where each set was taken as the Original data (see the Model Assessments section) with the results shown in Figs. 3.2d, 3.2e, and 3.2f, respectively. In all cases, the ID materials in the Original data generally have d values to the left of OD data. $F1_{max}$ scores are not lower than 0.7, which is generally good. Note that we can tune d_c^t (i.e., non-$F1_{max}$ d_c^t) such that the corresponding precision is near or at 1.00 at the cost of a lower recall for these data.

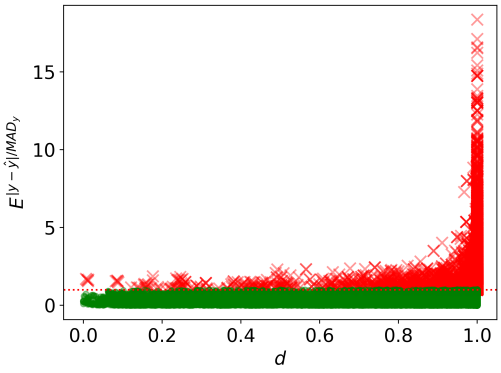
While our method does not flawlessly differentiate all chemical cases, it serves as a robust tool to eliminate a substantial portion of unreliable predictions. Furthermore, our chemical domains, like all the domain definitions in this paper, are approximate, and it is not expected that any method will provide perfect ID/OD predictions. Predicting domain with d_c^t derived from data analysis like those shown in Figs. 3.2a through 3.2f required being able to label a large set of data as chemically similar (ID) or dissimilar (OD), which requires extensive domain expertise and is not always feasible. Our other methods for domain determination rely purely on statistics of the trained model for assigning labels for domain, which makes them more practical for deployment. A^{chem} serves as an initial, intuitive, and rudimentary assessment, and passing this assessment was necessary for any domain method to be considered useful for materials. These results show that d effectively provided privileged information regarding the chemical classes and physics that separate materials. We are confident that any positive outcomes observed for $A^{|y-\hat{y}|/MAD_y}$, A^{RMSE/σ_y}, or A^{area} are not solely attributable to numerical artifacts. These readily interpretable results from A^{chem} establish a firm, intuitive, and chemically informed foundation before exploring more general numerical approaches, which we do now.

Relationship Between Absolute Residuals ($E^{|y-\hat{y}|/MAD_y}$) and Distance (d) from the Residual Assessment ($A^{|y-\hat{y}|/MAD_y}$)

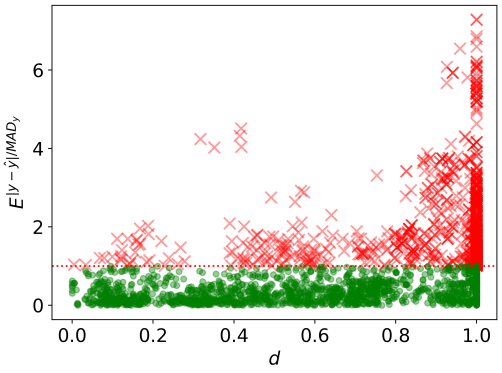
In Fig. 3.3, we illustrate how $E^{|y-\hat{y}|/MAD_y}$ is related to d for the RF model type. We summarize the results for other model types in Table 3.3. We observe that $E^{|y-\hat{y}|/MAD_y}$ increases when the likelihood of observing similar points to ITB data decreases (larger d). All AUC-Baseline scores in Table 3.3 are positive for the relevant assessment, meaning that our d measure gives more information than a naïve guess provided by our baseline. Nearly all $F1_{max}$ scores are above 0.7, which means that d can be used to significantly separate ID and OD points. Only 2 out of 20 measures of $F1_{max}$ fall below 0.7. As an example of application, we can examine Fig. 3.3a and select d_c^t=1.00 for which we reject predictions (i.e., label as $\widehat{OD}$) from any M^{prop} according to Eq. 3.5. Most points with large $E^{|y-\hat{y}|/MAD_y}$ tend to occur at d=1.00. By selecting a d_c^t=1.00, a user can filter out the majority of those points from a study, thereby saving time by excluding untrustworthy cases. Other data sets in Figs. 3.3b, 3.3c, and 3.3d have more OD cases above $E_c^{|y-\hat{y}|/MAD_y}$ at lower d compared to the previous example. However, the fraction of OD cases greatly exceeds that of ID cases when d=1.00 and the fraction of ID cases is higher than the number of OD cases when d<1.00. Therefore, the $F1_{max}$ occurs at d_c^t=1.00, where data have a zero likelihood based on the KDE constructed from ITB data. A similar methodology can be applied for d_c^t values across other data set and model combinations. Essentially, we can select predictions likely to be better than naïve. We note that $E^{|y-\hat{y}|/MAD_y}$ is a statistical quantity that will almost certainly be challenging to predict by d or any other domain method as it is quite stochastic. Even if one has a method that was essentially perfect at identifying data that was ID/OD, some residuals would likely be large for ID data and small for OD data by chance. Given this intrinsic limitation of predicting $E^{|y-\hat{y}|/MAD_y}$, we consider the present results quite strong. The statistical quantity E^{RMSE/σ_y} is essentially the same information as $E^{|y-\hat{y}|/MAD_y}$, but averaged over many points and therefore less stochastic, and we will see below that E^{RMSE/σ_y} is predicted more robustly by d.



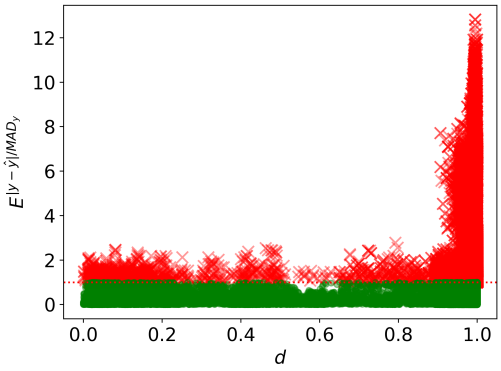
(a) Diffusion



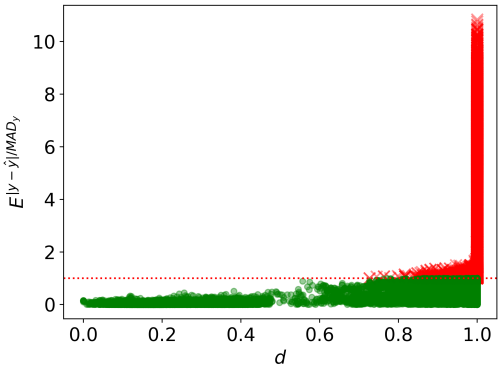
(b) Fluence



(c) Steel Strength



(d) Superconductor

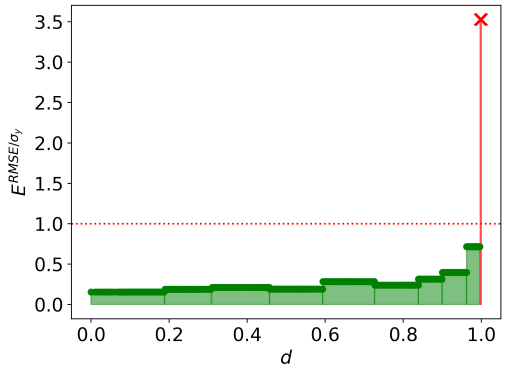


(e) Friedman

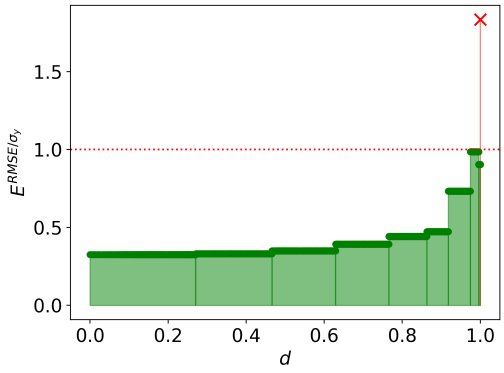
Figure 3.3: **Absolute residuals grow as OOB data becomes increasingly dissimilar.** The relationship between $E^{|y-\hat{y}|/MAD_y}$ and d for the RF model type is shown. Generally, $E^{|y-\hat{y}|/MAD_y}$ increases with an increase in d. $E_c^{|y-\hat{y}|/MAD_y}$ is shown by the horizontal red line, which separates our OD (red) and ID (green) cases.

Relationship Between $RMSE$ (E^{RMSE/σ_y}) and Distance (d) from the $RMSE$ Assessment (A^{RMSE/σ_y})

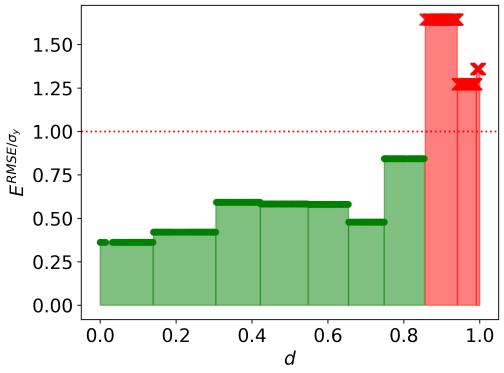
In Fig. 3.4, we illustrate how E^{RMSE/σ_y} is related to d for the RF model type. We summarize the results for other model types in Table 3.3. Similar to the results from the previous section, we observed that E^{RMSE/σ_y} increased when the likelihood of observing similar points to ITB data decreased (larger d). All AUC-Baseline scores in Table 3.3 are positive for the relevant assessment, meaning that our d measure gives more information than a naïve guess provided by our baseline. Nearly all $F1_{max}$ scores are 1.00, which means that ID and OD bins were nearly perfectly separated (i.e., only 4 out of 20 were not perfect but still above 0.7). As an example of application, we can examine Fig. 3.4c and select d_c^t=0.85 for Eq. 3.5. Three bins at d>0.85 have E^{RMSE/σ_y} above E_c^{RMSE/σ_y} and are OD. All other bins have E^{RMSE/σ_y} below E_c^{RMSE/σ_y} and are ID. If M^{dis} yields d=0.8 for a data point, the point falls in a bin that is ID. If M^{dis} yields d=0.95 for a data point, the point falls in a bin that is OD. A similar methodology can be applied for d_c^t values across other data set and model combinations. We can use d to distinguish points where E^{RMSE/σ_y} is expected to be low from those likely to have high E^{RMSE/σ_y} (i.e., we can discern predictions likely to be better than a naïve).



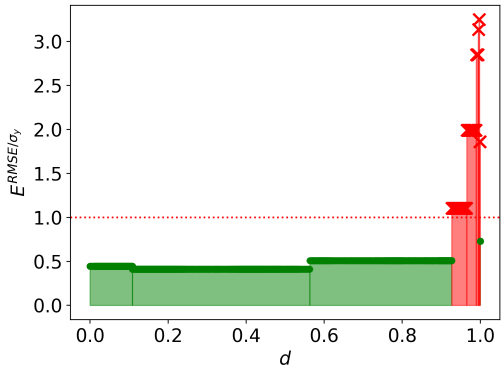
(a) Diffusion



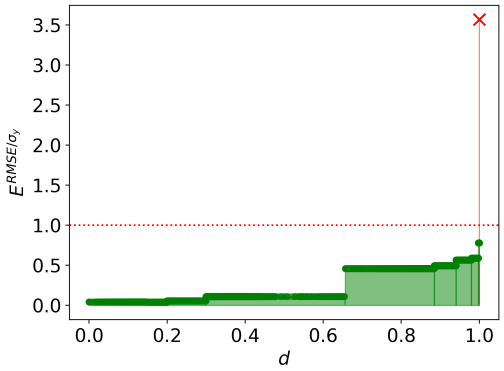
(b) Fluence



(c) Steel Strength



(d) Superconductor

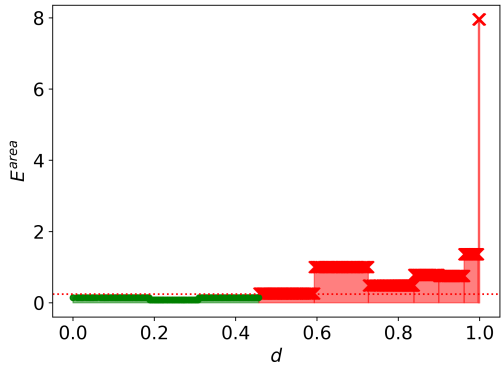


(e) Friedman

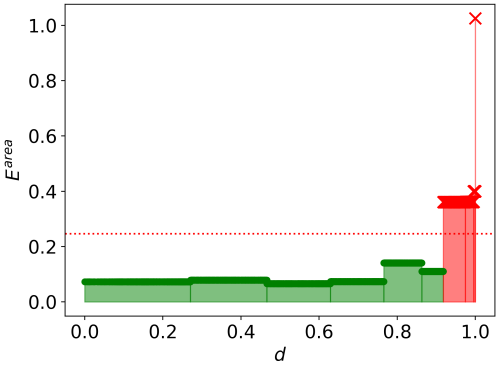
Figure 3.4: **RMSE grows as OOB data becomes increasingly dissimilar.** The relationship between E^{RMSE/σ_y} and d for the RF model type is shown. Generally, E^{RMSE/σ_y} increases with an increase in d. E_c^{RMSE/σ_y} is shown by the horizontal red line, which separates our OD (red) and ID (green) bins.

Relationship Between Miscalibration Area (E^{area}) and Distance (d) from the Uncertainty Quality Assessment (A^{area})

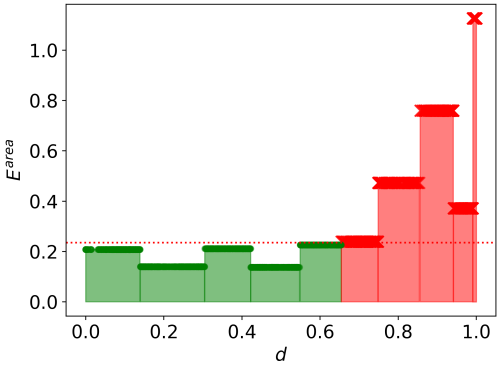
We now provide a similar analysis for E^{area} as previously provided for E^{RMSE/σ_y}. In Fig. 3.5, we illustrate how E^{area} and d are related for the RF model type. We summarize the results for other model types in Table 3.3. We observed that E^{area} for bins with small d values tend to be relatively small compared to bins with larger d values for all the data presented in Fig. 3.5. All AUC-Baseline scores are greater than zero except for two, indicating that two data set and model combinations performed worse than naively predicting the number of ID bins (i.e., only 2 out of 20 results were undesirable). For the most part, our d measure offers substantial insight into the quality of E^{area} for the reported data. Approximately half (9 out of 20) of the model type and data combinations yield precision and recall scores of 1.00, indicating their ability to perfectly discern points in ID/OD bins. A total of 11 out of the 20 entries report $F1_{max}$ above 0.7, which is relatively good but not nearly as great as the results from $A^{|y-\hat{y}|/MAD_y}$ and A^{RMSE/σ_y}. An additional 4 entries were only slightly below 0.7. The only entries with $F1_{max}$ scores worse than naïve were Steel Strength with BNN and Diffusion with BOLS. No data for the entry for Friedman with BOLS was ID, which was the reason for an AUC-Baseline of zero. The quality of M^{unc} affects the ability of M^{dom} to discern domain, a problem we discuss in detail as case (ii) in the Notes of Caution for Domain Prediction section and in the Appendix. As an example of application, we can examine Fig. 3.5b and select d_c^t=0.92 for as a threshold to discern between ID/OD bins. If M^{dis} yields d=0.5 for a point, the point falls in a bin that is ID. If M^{dis} yields d=0.99 for a point, the point falls in a bin that is OD. Choices for d_c^t can be similarly made for other models. These results show that we can usually use d to distinguish points where E^{area} is expected to be low from those likely to have high E^{area} (i.e., we can discern predictions likely to be better than a naïve). It is important to note that each bin utilizes the exact same data to calculate E^{RMSE/σ_y} and E^{area}. This implies that bins that were ID based on E_c^{RMSE/σ_y} but OD based on E_c^{area} had low absolute residuals but poor uncertainty quantification accompanying the predictions.



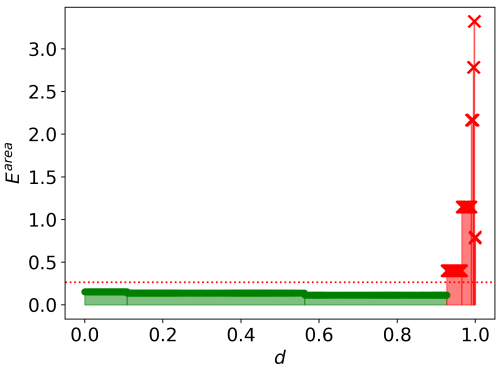
(a) Diffusion



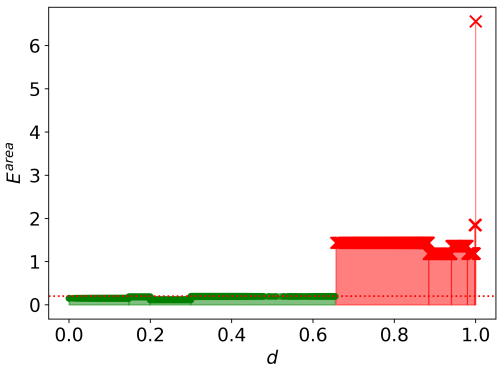
(b) Fluence



(c) Steel Strength



(d) Superconductor

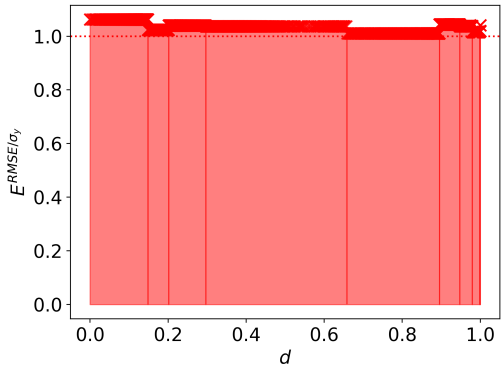


(e) Friedman

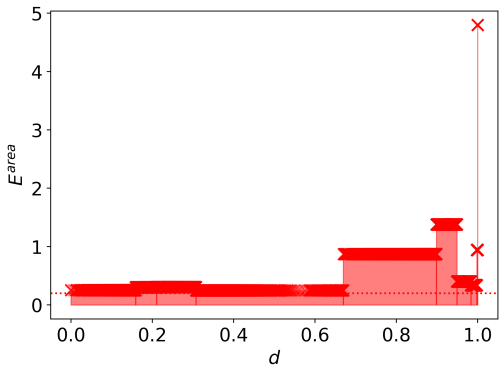
Figure 3.5: **Uncertainty estimates deteriorate as OOB data becomes increasingly dissimilar.** The relationship between E^{area} and d for the RF model type is shown. Generally, E^{area} increases with an increase in d. E_c^{area} is shown by the horizontal red line, which separates our OD (red) and ID (green) bins.

Notes of Caution for Domain Prediction

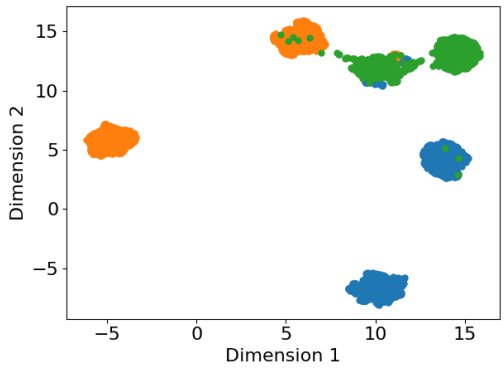
We have identified four scenarios that could, but not necessarily, lead to the breakdown of developed models (although others likely exist): (i) the trivial case where OOB y cannot be learned from X (i.e., M^{prop} breaks), (ii) the case where uncertainties are bad (i.e., M^{unc} breaks), (iii) the case where BLOCO fails to select regions of X that are sufficiently distinct as OOB sets, and (iv) the case where KDE does not provide adequate information about ITB data because of a very large number of features. We will show the first 3 failure modes with easily controlled data generated with Eq. 3.8 (Friedman and FWODC). To show case (i), we shuffled y with respect to X. M^{prop} could not learn y from X in this scenario. E^{RMSE/σ_y} will be above E_c^{RMSE/σ_y} for all (or nearly all) bins of d. To show case (ii), we used uncalibrated (σ_u) instead of calibrated (σ_c) uncertainties. If M^{unc} is not accurate, then d will show no reasonable trend between E^{area} and d. To show case (iii), we used UMAP to show how insufficient clustering of distinct spaces provided values of d that could not be used to build M^{dom}. This is a failure in producing ID/OD labels for OOB data. In cases (i)-(iii), the failure is not really a problem with the fundamental domain method, as we now explain. For case (i), it is reasonable to assume an M^{prop} model has no domain (or, equivalently, all data is OD) if its predictions are poor, and therefore no domain prediction method can reasonably be expected to work. Similarly, an M^{unc} incapable of generating accurate uncertainties as outlined in case (ii) would have most, if not all, data as being OD. For case (iii), we do not separate data with our BLOCO procedure. This is not a failure of the fundamental approach, but a failure of our specific strategy for splitting leading to poor generation of ID/OD samples. However, case (iv) would be a true failure of the underlying KDE approach, although we have not seen it occur in our tests.



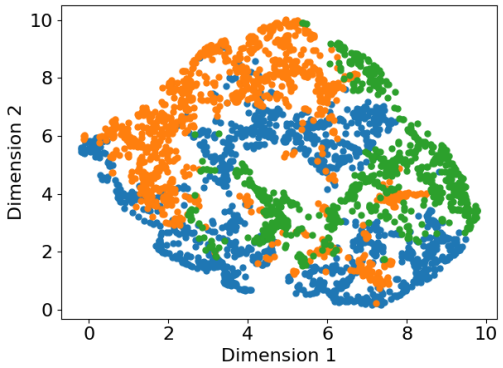
(a) Shuffled y



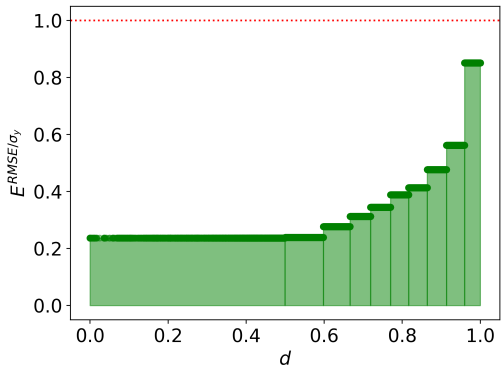
(b) Uncertainties from σ_u Instead of σ_c



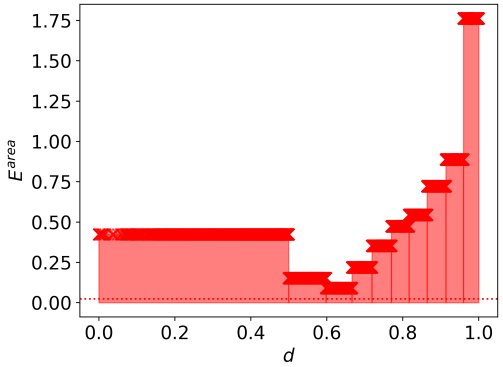
(c) UMAP of Friedman



(d) UMAP of FWODC



(e) A^{RMSE/σ_y} on FWODC



(f) A^{area} on FWODC

Figure 3.6: **Explanation of the limits of KDE to determine domain.** Fig. 3.6a shows the assessment of the Friedman data fit with an RF model type where we purposefully shuffled y to acquire an M^{prop} with no predictive ability. Because X no longer has a strong relationship with y, all data are OD. Fig. 3.6b shows the relationship between d and E^{area} for using σ_u instead of σ_c for M^{unc}. Because M^{unc} is poor at estimating uncertainties, all data are OD. The UMAP projections of Friedman and FWODC onto two-dimensions are shown in Figs. 3.6c and 3.6d, respectively. One sampling of X yields distinct regions in features (left) and the other does not (right). The colors represent the labels for three clusters acquired through agglomerative clustering. Figs. 3.6e and 3.6f show the relationship between d, E^{RMSE/σ_y}, and E^{area} for the poorly clustered FWODC data. Note that at least the bin with the highest d for A^{RMSE/σ_y} should be OD. We observe that data closest to our X_{ITB} for A^{area} are marked as OD domain by E_c^{area}, but should ideally be ID.

The RF model type was used for M^{prop} for cases (i)-(iii) with either Friedman or FWODC data. We start with case (i) using the Friedman data. By shuffling y with respect to X, the results from our assessments of A^{RMSE/σ_y} breakdown (Fig. 3.6a). Note that all predictions from M^{prop} are worse than naively predicting $\bar{y}$ for all points. A d_c^t cannot be established for separating ID/OD cases from E^{RMSE/σ_y} since all data are OD. Fig. 3.6b shows how M^{dom} fails if M^{unc} fails via case (ii). We use σ_u instead of σ_c as the estimates provided by M^{unc}. Errors in our uncertainties are high, and no value of d_c^t separates ID/OD points well (M^{dom} failure) because no bins exist that are ID. Note that the portion of M^{dom} trained with E^{RMSE/σ_y} data should remain unaffected as long as M^{prop} functions well. Failure case (iii) for our domain methodology stems from a specific failure case of the BLOCO procedure. First, we use the UMAP approach to project X into a two-dimensional space and visualize the cluster labels assigned by agglomerative clustering for Friedman (Fig. 3.6c) and FWODC (Fig. 3.6d) data. Each color from the figures represents the cluster labels for three clusters. Because there are intervals of sampling for X that exclude other intervals for Friedman in Fig. 3.6c, we can cluster subspaces of X that are distinct. If we consider FWODC as shown in Fig. 3.6d, all data that are OOB by BLOCO are close to each other, which results in an insufficient distinctiveness of clusters for pseudo-label generation (ID/OD). The X space from Fig. 3.6c led to the previously seen results on A^{RMSE/σ_y} and A^{area} (Figs. 3.4e and 3.5e). Data used to make Fig. 3.6d led to the results shown in Figs. 3.6e and 3.6f. For Fig. 3.6e, we know that data at d=1.00 can produce OD points as seen in Fig. 3.4e. For Fig. 3.6f, E^{area} is much larger for our first bin than that from Fig. 3.5e. Conversely, E^{area} is much smaller for our last bin in Fig. 3.6f compared to Fig. 3.5e. Unusual trends of d with respect to E^{area} emerge when the data set lacks diversity (i.e., data that cannot be clustered well).

Finally, we discuss failure mode (iv). We do not illustrate case (iv), as we did not run into this problem in our studies, but is worth mentioning. KDE is known to have challenges in getting accurate representations for a large number of dimensions [152]. For models with many features, it is possible that KDE will give a poor representation of the distance to the X_{ITB} data and the approach taken in this work will break down (i.e., yield inaccurate

$\widehat{ID}/\widehat{OD}$ labels). We do not have any example of where this occurs but note the concern for completeness. Feature selection to obtain a modest feature number for which the KDE methods work is suggested. We do not know how many features may cause issues, but the models studied here use a maximum of 25 features.

In summary, there are three known conditions and a fourth one hypothesized that will lead to a non-functional M^{dom}. First, M^{prop} cannot properly predict y from X. Second, M^{unc} fails to provide accurate measures of uncertainty. Third, the data cannot be clustered to produce ID/OD labels. Fourth, the KDE fails to provide a good density due to a large number of features. So long as the aforementioned conditions are not met, M^{dom} has been found to effectively predict domain in our tests.

3.4 Conclusion

Our work addresses a significant concern in the deployment of machine learning models: the potential for these models to produce inaccurate or imprecise predictions without warning. We have shown that kernel density estimates provide valuable insights into where in the feature space model performance degrades significantly or where predictions fall outside the model's domain of applicability. The central idea of our approach is to use kernel density estimation to define a dissimilarity measure d and then classify data as in-domain or out-of-domain based on d. We assessed the approach with a range of machine learning models (random forest, bagged neural network, bagged support vector regressor, and bagged ordinary least squares) on data sets with diverse physical properties and chemistries (dilute solute diffusion in metals, ductile-to-brittle transition temperature shifts in irradiated steel alloys, yield strengths of steel alloys, and superconducting critical temperatures) as well as the synthetic Friedman data set. We demonstrated qualitatively that as d increased, data became more chemically distinct, residual magnitudes increased, and uncertainty estimates deteriorated, validating that d was a powerful descriptor for these critical ways of thinking about domain. Our quantitative assessment compared our predicted domain categorization to ground truth values based on chem-

istry, residuals, and uncertainty estimates and generally found good improvement over naïve models and high $F1$ scores. This approach can be easily applied to many problems and allows categorization of in-domain or out-of-domain for any test data point during inference. The approach can be applied easily through its stand-alone implementation or its implementation in the MAterials Simulation Toolkit - Machine Learning (MAST-ML) package (see Data and Code Availability). Researchers can easily use this method to provide automated guardrails for their machine learning models, greatly enhancing their reliable application.

3.5 Data and Code Availability

The raw and processed data required to reproduce these findings are available to download from figshare at doi: https://doi.org/10.6084/m9.figshare.25898017.v1. The developed code is available in four places: a static code version in GitHub at `https://github.com/leschultz/materials_application_domain_machine_learning.git`, a continuously developed code in GitHub at `https://github.com/uw-cmg/materials_application_domain_machine_learning.git`, an implementation in MAST-ML at `https://github.com/uw-cmg/MAST-ML.git`, and a PyPI package at `https://pypi.org/project/madml/`.

Chapter 4

Collaborative Publications

While the primary focus of this thesis is on developing machine learning models for predicting the glass formation of metallic glasses and providing guidelines to prevent poor deployment of machine learning models, the author has also provided supporting guidance and computation to several other works beyond those presented in Chapters 2 and 3. This chapter briefly introduces and describes these collaborative efforts.

4.1 "Molecular simulation-derived features for machine learning predictions of metal glass forming ability"

Note: This section has been published as B. T. Afflerbach *et al.*, "Molecular simulation-derived features for machine learning predictions of metal glass forming ability," *Computational Materials Science*, vol. 199, Nov. 2021, ISSN: 09270256. DOI: 10.1016/j.commatsci.2021.110728, and has been adapted for use in this thesis.

This research paper explored the use of molecular dynamics simulations to generate features for machine learning models that predict the glass-forming ability of metallic alloys. Molecular dynamics simulations were performed on 11 binary alloy systems to generate a database of critical cooling rates as the target. The study combined easily accessible elemental features with more complex, experimentally inspired simulated features to

build models. These features included the glass transition temperature, atomic packing density, and icosahedral-like fraction, among others.

Two models were built using the LASSO algorithm: a baseline model using only features generated from properties of elements and a second model incorporating simulated features. Models fitted only to the elemental features had an R^2 of 0.198 ± 0.168, while the inclusion of all features increased the R^2 to 0.769 ± 0.089. Analysis of the LASSO coefficients revealed that the two most important features were the enthalpy of crystallization and the icosahedral-like Voronoi polyhedra fraction at 100 [K], both derived from simulated features. The third most important feature was the average Mendeleev number, which came from the simple-to-compute elemental features.

This work demonstrates that including features derived from simulations can significantly improve machine learning models for predicting the glass-forming ability of metallic glasses. The computational accessibility of the covered features enables more accurate prediction of new glass-forming alloys across a broader design space.

4.2 "Machine Learning Prediction of the Critical Cooling Rate for Metallic Glasses from Expanded Datasets and Elemental Features"

Note: This section has been published as B. T. Afflerbach *et al.*, "Machine Learning Prediction of the Critical Cooling Rate for Metallic Glasses from Expanded Datasets and Elemental Features," *Chemistry of Materials*, acs.chemmater.1c03542, Mar. 2022, ISSN: 0897-4756. DOI: 10.1021/acs.chemmater.1c03542. [Online]. Available: https://pubs.acs.org/doi/10.1021/acs.chemmater.1c03542, and has been adapted for use in this thesis.

This study focused on developing a machine learning model to predict the experimental critical cooling rates for glass formation in various metallic alloys using only elemental features. The limited critical cooling rate data set was expanded by integrating multiple

sources of direct and indirect critical cooling rate data. This increased the number of data points from 77 to 2,125. Authors assessed a random forest model trained on this expanded data set with 5-fold cross-validation and showed a root mean square error of 0.69 ± 0.09 in log base 10 units of $[K/s]$. When assessing models with leave-out-one-group cross-validation, where alloy systems defined the groupings, models yielded a root mean square error of 0.88 in the same units. The errors in predictions from these assessments are sufficiently low to distinguish materials likely to be favorable glass formers from those that are not.

Subsequently, a model trained on all available 2,125 data points was used to predict the critical cooling rate for other material systems. The predictions identified several potential new bulk metallic glass systems for future study. Model predictions were also compared to previously established empirical rules for identifying metallic alloys with high glass-forming ability. The empirical rules had some general agreement with alloys to be predicted as favorable glass formers. The study highlights the potential of machine learning models in accelerating the discovery of new bulk metallic glasses when combined with human guidance and existing knowledge.

4.3 "Microalloying effect in ternary Al-Sm-X (X=Ag, Au, Cu) metallic glasses studied by ab initio molecular dynamics"

Note: This section has been published as J. Xi *et al.*, "Microalloying effect in ternary al-sm-x (x=ag, au, cu) metallic glasses studied by ab initio molecular dynamics," *Computational Materials Science*, vol. 185, p. 109 958, 2020, ISSN: 0927-0256. DOI: `https://doi.org/10.1016/j.commatsci.2020.109958`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0927025620304493`, and has been adapted for use in this thesis.

The study investigates the relationship between the fraction of icosahedral-like (ICO-

like) Voronoi polyhedral clusters and the glass-forming ability of the metallic glasses $Al_{90}Sm_8X_2$ (where X = Al (binary), Cu, Ag, Au) using ab initio molecular dynamics simulations. The work investigated if ICO-like fractions could serve as a useful guide for screening and identifying good glass formers.

The authors first demonstrated that the ICO-like fraction could be determined with adequate precision. They performed convergence tests on the system sizes and simulation times accessible ab initio, ensuring that the ICO-like fraction could be calculated with a standard error of the mean below 1% using systems of 256 atoms and averaging over approximately 20 simulations.

Next, the authors explored the correlation between the ICO-like fraction and the critical cooling rate. The critical cooling rate was estimated from experimental data on the critical thickness for forming amorphous ribbons and the thermodynamic properties of the alloys. The results showed a clear relationship between the ICO-like fraction and the estimated critical cooling rate across the four alloys studied. The authors also investigated the relationship between the ICO-like fraction and other parameters related to glass forming ability such as the average coordination number, atomic radius, and chemical bond strength of the alloying elements. While these parameters exhibited some trends with glass forming ability, the relationships were not monotonic.

Overall, the study demonstrated that simulations could be used to calculate the ICO-like fraction with sufficient accuracy and that this structural descriptor correlated well with the critical cooling rate in the $Al - Sm - X$ metallic glasses. These findings suggest that the ICO-like fraction from simulations may offer a useful guide for searching and screening for good glass formers, enabling the in-silico design of metallic glasses.

4.4 "Accelerating Ensemble Error Bar Prediction with Single Models Fits"

Note: This section is in preparation to be published but is available as V. Agrawal *et al.*, "Accelerating ensemble error bar prediction with single models fits," 2024. arXiv: 2404.09896, and has been adapted for use in this thesis.

This research paper introduces an approach to efficiently estimate ensemble-based uncertainty in machine learning models for materials science applications. The common method of using ensemble models to estimate prediction uncertainties can be computationally demanding, as an ensemble of N models requires approximately N times more computational resources compared to a single model during inference. The goal of this work was to produce a single model for uncertainty predictions that mimicked the ensemble approach.

The proposed methodology involved three models: Model A for property prediction, Model A_E for ensemble-based uncertainty prediction, and Model B to estimate uncertainties using a single model. The three data sets were augmented by adding small random values to each observation for each feature. These augmented features were then fed into Model A_E. Model B was built using the outputs of Model A_E, and its performance was assessed on left-out sets of uncertainty predictions of Model A_E. The results demonstrate that Model B can accurately reproduce the uncertainties for augmented data sets that are close to the original features.

This work demonstrates a practical approach to achieve uncertainty estimation with accuracy approaching that of ensemble methods using just a single model. The proposed method can enhance the speed and reduce the memory required to produce uncertainties, supporting greater use of uncertainty quantification in machine learning for materials science applications.

4.5 "Ultra-fast Oxygen Conduction in Sillén Oxychlorides"

Note: This section is in preparation to be published as but is available as J. Meng *et al.*, "Ultra-fast oxygen conduction in sillén oxychlorides," 2024. arXiv: 2406.07723, and has been adapted for use in this thesis.

In this study, researchers found lanthanum bismuth oxychloride ($LaBi_2O_4Cl$) to be an ultra-fast oxygen conductor from the MBi_2O_4X family. M and X are rare-earth and halogen elements, respectively. This discovery was made with the aid of a structure similarity analysis of simulated X-Ray diffraction analysis patterns and radial distribution functions of approximately 62,000 oxygen-containing compounds.

$LaBi_2O_4Cl$ is a layered structure that offers efficient oxygen ion transport and had a low migration barrier of 0.1 eV for oxygen vacancies calculated through ab initio. Experimental studies on synthesized $LaBi_2O_4Cl$ and strontium-doped $LaBi_2O_4Cl$ demonstrated comparable or higher oxygen conductivity than the widely used yttria-stabilized zirconia and lanthanum strontium gallium magnesium oxide below 400 [$°C$]. The discovery of the MBi_2O_4X material family as promising room-temperature fast oxygen conductors emphasize the significant potential for discovering new, efficient conductors from structural information.

4.6 *Machine Learning Materials Properties with Accurate Predictions, Uncertainty Estimates, Domain Guidance, and Persistent Online Accessibility*

Note: This section is in preparation to be published but is available as R. Jacobs *et al.*, *Machine learning materials properties with accurate predictions, uncertainty esti-*

mates, domain guidance, and persistent online accessibility, 2024. arXiv: 2406.15650 [cond-mat.mtrl-sci]. [Online]. Available: https://arxiv.org/abs/2406.15650, and has been adapted for use in this thesis.

In this work, random forest machine learning models were fit to 33 distinct materials properties spanning a diverse array of data sources (computational and experimental) and property types (electrical, mechanical, thermodynamic, etc.). All models could provide target property prediction, uncertainty estimates, and a measure feature dissimilarity via kernel density estimation. The fitted models were publicly hosted on the Garden infrastructure, providing an intuitive, persistent interface for model deployment.

A subset of models were used to screen for perovskite oxide catalyst materials predicted to be promising for fuel cell and electrolyzer applications. The authors employed machine learning models for approximately 19 million candidate perovskite compositions and screened materials based on predicted stability, area-specific resistance, conductivity, and thermal expansion coefficient values. Kernel density estimated dissimilarities were used to filter problematic predictions from models. This screening process identified several promising perovskite catalyst materials with properties comparable to well-studied materials. The combination of materials data and models made accessible through the Garden infrastructure enable synergistic searches of materials with lucrative properties.

Chapter 5

Concluding Remarks

5.1 Summary

In this thesis, we have explored the application of machine learning techniques to predict the glass forming ability of metallic alloys and to assess the applicability domain of machine learning models. The work primarily focuses on three papers that predict the glass forming ability through properties acquired from experimental data, molecular dynamics simulations, and other computational methods. A fourth and final work proposes a general approach for determining the applicability domain of machine learning models.

Chapter 2 has significantly advanced our understanding of the glass forming ability prediction of metallic glasses through the integration of machine learning techniques with both experimental and computational data. The studies conducted have demonstrated that characteristic temperatures, molecular dynamics simulations, and machine-learned interatomic potentials can be effectively utilized to predict glass forming ability. The knowledge gathered from these works culminated to produce a model capable of predicting experimental R_c with an $R^2 = 0.78$ from features derived computationally (Sec. 2.6).

In the broader context of machine learning applications in materials science (see Chapter 3), this thesis has made significant contributions by developing a general approach for determining the applicability domain of machine learning models. The proposed method

uses kernel density estimation to measure the training data density at the feature values for an inference point. The research has shown that high dissimilarity measures, as determined by the kernel density estimation-based approach, are associated with poor model performance. By providing automated tools to establish acceptable dissimilarity thresholds, this work has equipped researchers with the means to identify whether new predictions fall within the applicability domain of their machine learning models.

This thesis showcases the successful application of machine learning techniques in predicting the properties of metallic glasses and assessing the applicability domain of machine learning models. The methodologies developed and insights gained from this work contribute to the advancement of materials science and engineering, facilitating the discovery of novel materials with desirable properties.

5.2 Suggestions for Future Work

To improve the predictive accuracy of machine learning models for glass forming ability, it is essential to expand the database of metallic glass compositions and their properties. Larger data sets will enable the development of more robust models and enhance generalizability. Future research should also explore additional features beyond those considered in this work. Fortunately, the development of more accurate and universal machine learned interatomic potentials will facilitate the exploration of more material properties for a greater number of metallic alloys. Experimental validation is still needed. To avoid bias from model tuning, glass formation ability models should be validated using unseen experimental data. Specifically, a model should be developed, predictions should be acquired, and then those predictions should be checked with experimental synthesis of potential glasses. Combining high throughput computation with new simulation techniques will streamline the screening process for potential glass-forming alloys.

Further refinement of the kernel density estimation-based approach for determining the applicability domain of ML models can improve its accuracy and reliability for domain determination. The current implementation overlooks the relevance of each feature for

model prediction. Weighting density estimates based on feature relevance should yield more precise regions of feature space that result in robust model predictions. The domain method's applicability can be expanded to diverse data types and tasks. So far, the method only works on tabular data with continuous feature values. The methodology used in our domain determination implementation could be adapted for other machine learning tasks, such as image classification where the features are based on image pixels. By addressing these limitations and expanding the method's versatility, researchers can enhance the overall effectiveness and widespread adoption of kernel density estimation-based approaches for defining applicability domains across various machine learning models and data types.

Chapter 6

Appendix

6.1 Appendix for Section 2.4

Scikit-learn metrics were used to assess the performance of ML models with the exception of $RMSE/\sigma$ [42]. Relevant metrics that have functional forms are defined in this section. Additional tables and a figure used in our work are presented here.

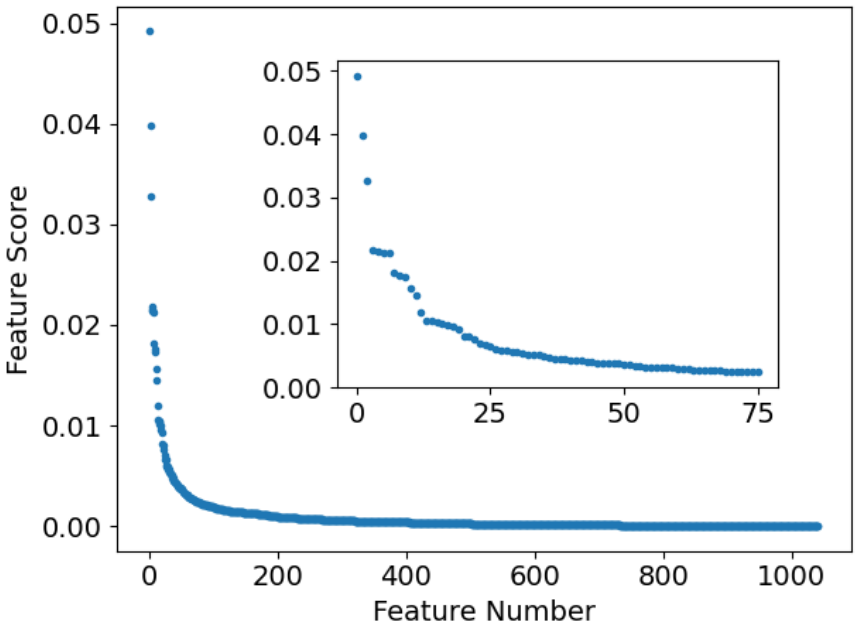


Figure 6.1: The feature rankings for all PRSD functions of CTs for the GB model fit to all data.

Table 6.1: The features scores for the full fit GB model. Higher score is better. Features where the score rounds to zero for two decimal places are excluded.

Features	Scores
[(tg-tx)^4]*1/[(tg-tl)^4]	0.05
(tg-tx)*1/(tg-tl)	0.04
[(tl-tx)^4]*1/[(tg-tx)^4]	0.03
[tg^4]*1/[(tg-tl)^4]	0.02
(tl+tx)*1/[tl^2]	0.02
[(tg-tl)^4]*1/[(tl-tx)^2]	0.02
[(tg-tx)^2]*1/[(tl-tx)^2]	0.02
(tg-tl)*1/(tg-tx)	0.02
[(tl+tx)^4]*1/[tg^3]	0.02
[(tg-tl)^4]*1/[(tg-tx)^4]	0.02
[(tg-tx)^4]*1/[(tl-tx)^4]	0.02
[(tl+tx)^4]*1/[(tg-tx)^3]	0.01
[tl^4]*[(tg-tx)^2]	0.01
[(tg-tl)^2]*1/[(tl-tx)^3]	0.01
[(tl+tx)^3]*1/[tx^2]	0.01
[(tg-tx)^2]*1/(tl+tx)	0.01
[tg^3]*1/[(tg+tx)^3]	0.01
[(tg+tx)^2]*1/[(tl-tx)^4]	0.01
[(tg-tl)^2]*1/[(tg-tx)^2]	0.01
[tl^3]*1/[(tg+tl)^3]	0.01
[(tg+tl)^4]*1/[(tg-tl)^4]	0.01
[(tl+tx)^2]*1/[tx^4]	0.01
[(tg+tx)^3]*1/[(tg-tl)^4]	0.01
tg*1/[(tg+tx)^3]	0.01
[tx^3]*[(tl+tx)^4]	0.01
[(tl-tx)^4]*1/[(tg-tx)^2]	0.01
(tl-tx)*1/[(tg-tx)^2]	0.01
1/tl*1/[tg^4]	0.01
tl*1/(tg-tx)	0.01
[(tg+tx)^2]*1/[(tg-tx)^2]	0.01
[(tg-tx)^2]*1/[tx^3]	0.01
[tx^4]*1/[(tg-tx)^3]	0.01
[(tg-tx)^3]*1/[tl^2]	0.01
[(tg-tl)^3]*1/[tg^2]	0.01
1/(tg-tx)*1/[(tg+tl)^3]	0.01

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{6.1}$$

where:

- R^2 is the coefficient of determination
- i is the sample number
- n is the number of samples
- y_i is the true target value for a case i
- $\hat{y}_i$ is the predicted target value for a case i
- $\bar{y}$ is the mean of true target values

$$MAE = \frac{1}{n}\sum_{i=0}^{n-1}|y_i - \hat{y}_i| \tag{6.2}$$

where:

- MAE is the mean absolute error
- i is the sample number
- n is the number of samples
- y_i is the true target value for a case i
- $\hat{y}_i$ is the predicted target value for a case i

$$MSE = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2 \tag{6.3}$$

where:

- MSE is the mean squared error
- i is the sample number
- n is the number of samples
- y_i is the true target value for a case i
- $\hat{y}_i$ is the predicted target value for a case i

$$RMSE = \sqrt{MSE} \tag{6.4}$$

where:

- $RMSE$ is the root mean squared error
- MSE is the mean squared error

$$RMSE/\sigma = \frac{RMSE}{\sigma} \tag{6.5}$$

where:

- $RMSE/\sigma$ is the ratio between RMSE and σ
- $RMSE$ is the root mean squared error
- σ is the standard deviation in the true target values

$$accuracy = \frac{1}{n}\sum_{i=0}^{n-1} 1(y_i = \hat{y}_i) \tag{6.6}$$

where:

- $accuracy$ is the accuracy
- i is the sample number
- n is the number of samples
- y_i is the true target value for a case i
- $\hat{y}_i$ is the predicted target value for a case i

$$precision = \frac{tp}{tp + fp} \tag{6.7}$$

where:

- $precision$ is the precision
- tp is the number of true positives
- fp is the number of false positives

$$recall = \frac{tp}{tp + fn} \tag{6.8}$$

where:

- $recall$ is the recall

- tp is the number of true positives
- fn is the number of false negatives

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{6.9}$$

where:

- F_1 is the harmonic mean between precision and recall
- $precision$ is defined in Equation 6.7
- $recall$ is defined in Equation 6.8

Table 6.2: The mean and standard deviation for the outer loops in nested cross validation for the generated set of features for GB models.

Metric	Log_{10}	PCA	Mean	STDEV	SEM
MAE	False	False	2.18	0.21	0.09
MAE	False	True	2.89	0.32	0.14
MAE	True	False	4.61	0.16	0.07
MAE	True	True	4.63	0.32	0.14
$RMSE$	False	False	3.28	0.29	0.13
$RMSE$	False	True	4.06	0.48	0.22
$RMSE$	True	False	6.51	0.41	0.18
$RMSE$	True	True	6.53	0.54	0.24
$RMSE/\sigma$	False	False	0.70	0.08	0.04
$RMSE/\sigma$	False	True	0.86	0.10	0.05
$RMSE/\sigma$	True	False	1.37	0.04	0.02
$RMSE/\sigma$	True	True	1.38	0.06	0.03
R^2	False	False	0.50	0.11	0.05
R^2	False	True	0.26	0.18	0.08
R^2	True	False	-0.87	0.10	0.05
R^2	True	True	-0.91	0.17	0.08

Table 6.3: The mean and standard deviation for the outer loops in nested cross validation for the generated set of features for GKRR models.

Metric	Log_{10}	PCA	Mean	STDEV	SEM
MAE	False	False	2.36	0.14	0.06
MAE	False	True	2.40	0.24	0.11
MAE	True	False	4.63	0.23	0.10
MAE	True	True	4.63	0.44	0.20
$RMSE$	False	False	3.57	0.25	0.11
$RMSE$	False	True	3.58	0.37	0.16
$RMSE$	True	False	6.52	0.58	0.26
$RMSE$	True	True	6.52	0.61	0.27
$RMSE/\sigma$	False	False	0.76	0.06	0.03
$RMSE/\sigma$	False	True	0.75	0.09	0.04
$RMSE/\sigma$	True	False	1.39	0.09	0.04
$RMSE/\sigma$	True	True	1.38	0.05	0.02
R^2	False	False	0.43	0.09	0.04
R^2	False	True	0.43	0.13	0.06
R^2	True	False	-0.93	0.25	0.11
R^2	True	True	-0.90	0.14	0.06

Table 6.4: The mean and standard deviation for the outer loops in nested cross validation for the generated set of features for LASSO models.

Metric	Log_{10}	PCA	Mean	STDEV	SEM
MAE	False	False	2.79	0.23	0.13
MAE	False	True	2.75	0.12	0.07
MAE	True	False	4.76	0.18	0.10
MAE	True	True	4.57	0.40	0.23
$RMSE$	False	False	3.96	0.50	0.29
$RMSE$	False	True	3.97	0.22	0.13
$RMSE$	True	False	6.81	0.32	0.19
$RMSE$	True	True	6.66	0.68	0.39
$RMSE/\sigma$	False	False	0.86	0.03	0.02
$RMSE/\sigma$	False	True	0.81	0.04	0.02
$RMSE/\sigma$	True	False	1.40	0.05	0.03
$RMSE/\sigma$	True	True	1.36	0.03	0.02
R^2	False	False	0.25	0.05	0.03
R^2	False	True	0.33	0.07	0.04
R^2	True	False	-0.96	0.13	0.07
R^2	True	True	-0.85	0.09	0.05

Table 6.5: The mean and standard deviation for the outer loops in nested cross validation for the generated set of features for RF models.

Metric	Log_{10}	PCA	Mean	STDEV	SEM
MAE	False	False	2.24	0.19	0.09
MAE	False	True	3.20	0.33	0.15
MAE	True	False	4.61	0.17	0.08
MAE	True	True	4.64	0.30	0.14
$RMSE$	False	False	3.30	0.33	0.15
$RMSE$	False	True	4.30	0.48	0.21
$RMSE$	True	False	6.51	0.42	0.19
$RMSE$	True	True	6.52	0.66	0.30
$RMSE/\sigma$	False	False	0.70	0.04	0.02
$RMSE/\sigma$	False	True	0.92	0.14	0.06
$RMSE/\sigma$	True	False	1.37	0.06	0.03
$RMSE/\sigma$	True	True	1.38	0.05	0.02
R^2	False	False	0.51	0.06	0.03
R^2	False	True	0.14	0.26	0.12
R^2	True	False	-0.89	0.17	0.07
R^2	True	True	-0.91	0.14	0.06

Table 6.6: The mean and standard deviation for the outer loops in nested cross validation for the three characteristic temperatures feature set for GB models.

Metric	Log_{10}	PCA	Mean	STDEV	SEM
MAE	False	False	2.30	0.30	0.13
MAE	False	True	2.52	0.19	0.09
MAE	True	False	4.60	0.27	0.12
MAE	True	True	4.60	0.60	0.27
$RMSE$	False	False	3.38	0.43	0.19
$RMSE$	False	True	3.74	0.31	0.14
$RMSE$	True	False	6.51	0.46	0.20
$RMSE$	True	True	6.49	0.81	0.36
$RMSE/\sigma$	False	False	0.71	0.06	0.02
$RMSE/\sigma$	False	True	0.80	0.10	0.05
$RMSE/\sigma$	True	False	1.37	0.06	0.03
$RMSE/\sigma$	True	True	1.38	0.04	0.02
R^2	False	False	0.49	0.08	0.04
R^2	False	True	0.34	0.16	0.07
R^2	True	False	-0.89	0.17	0.08
R^2	True	True	-0.90	0.11	0.05

Table 6.7: The mean and standard deviation for the outer loops in nested cross validation for the three characteristic temperatures feature set for GKRR models.

Metric	Log_{10}	PCA	Mean	STDEV	SEM
MAE	False	False	2.33	0.16	0.07
MAE	False	True	2.29	0.37	0.17
MAE	True	False	4.61	0.39	0.18
MAE	True	True	4.62	0.24	0.11
$RMSE$	False	False	3.44	0.34	0.15
$RMSE$	False	True	3.51	0.77	0.34
$RMSE$	True	False	6.49	0.80	0.36
$RMSE$	True	True	6.51	0.50	0.22
$RMSE/\sigma$	False	False	0.73	0.08	0.04
$RMSE/\sigma$	False	True	0.75	0.06	0.03
$RMSE/\sigma$	True	False	1.38	0.05	0.02
$RMSE/\sigma$	True	True	1.38	0.05	0.02
R^2	False	False	0.46	0.12	0.05
R^2	False	True	0.44	0.09	0.04
R^2	True	False	-0.91	0.13	0.06
R^2	True	True	-0.89	0.15	0.07

Table 6.8: The mean and standard deviation for the outer loops in nested cross validation for the three characteristic temperatures feature set for LASSO models.

Metric	Log_{10}	PCA	Mean	STDEV	SEM
MAE	False	False	3.37	0.23	0.10
MAE	False	True	3.37	0.23	0.10
MAE	True	False	4.62	0.59	0.26
MAE	True	True	4.62	0.37	0.16
$RMSE$	False	False	4.76	0.60	0.27
$RMSE$	False	True	4.75	0.59	0.27
$RMSE$	True	False	6.59	0.95	0.43
$RMSE$	True	True	6.61	0.72	0.32
$RMSE/\sigma$	False	False	1.01	0.01	0.00
$RMSE/\sigma$	False	True	1.00	0.00	0.00
$RMSE/\sigma$	True	False	1.41	0.05	0.02
$RMSE/\sigma$	True	True	1.40	0.07	0.03
R^2	False	False	-0.01	0.01	0.01
R^2	False	True	-0.00	0.00	0.00
R^2	True	False	-0.98	0.14	0.06
R^2	True	True	-0.98	0.20	0.09

Table 6.9: The mean and standard deviation for the outer loops in nested cross validation for the three characteristic temperatures feature set for RF models.

Metric	Log_{10}	PCA	Mean	STDEV	SEM
MAE	False	False	2.31	0.20	0.09
MAE	False	True	2.40	0.24	0.11
MAE	True	False	4.59	0.42	0.19
MAE	True	True	4.60	0.38	0.17
$RMSE$	False	False	3.43	0.28	0.13
$RMSE$	False	True	3.50	0.55	0.24
$RMSE$	True	False	6.48	0.63	0.28
$RMSE$	True	True	6.48	0.78	0.35
$RMSE/\sigma$	False	False	0.73	0.06	0.03
$RMSE/\sigma$	False	True	0.74	0.07	0.03
$RMSE/\sigma$	True	False	1.37	0.03	0.01
$RMSE/\sigma$	True	True	1.38	0.07	0.03
R^2	False	False	0.46	0.09	0.04
R^2	False	True	0.45	0.10	0.05
R^2	True	False	-0.87	0.09	0.04
R^2	True	True	-0.92	0.20	0.09

6.2 Appendix for Section 2.5

We explain in detail the viscosity calculations used in this work. Because practical MD has a finite run time, the integral form Equation 2.3 has an upper time limit of t_s. MD is also discrete so integration was represented with summation instead. Reformulation of Equation 2.3 yields the following:

$$\eta = \lim_{t_s \to \infty} \frac{V}{k_B T} \int_0^{t_s} \langle P_{ij}(t_0) P_{ij}(t_0 + t) \rangle_{t_0} \, dt \tag{6.10}$$

$$\eta = \lim_{t_s \to \infty} \eta_{t_s} \tag{6.11}$$

$$\eta_{t_s} = \frac{V}{k_B T} \int_0^{t_s} \langle P_{ij}(t_0) P_{ij}(t_0 + t) \rangle_{t_0} \, dt \tag{6.12}$$

$$\eta_{t_s} = \frac{V}{k_B T} \lim_{\delta t \to 0} \sum_{t=0}^{t_s} \delta t \, \langle P_{ij}(t_0) P_{ij}(t_0 + t) \rangle_{t_0} \tag{6.13}$$

In Equation 6.12, η_{t_s} is the approximate value of viscosity when the integral is only taken for a finite time t_s. The thermodynamic average $\langle \rangle_{t_0}$ of pressure values was performed by averaging over several time origins, t_0, separated by a time lag. We take this average over different t_0 values, each separated by multiples of 0.1 ps, to obtain a value of η_{t_s} every 100 ps. Pressure values with equal time separations are then averaged and integrated 100 times to get the total η_{t_s} over the full 10 ns isothermal hold. To further explain, consider the following for autocorrelation of a quantity $P = P_{ij}$:

$$\langle P(t_0) P(t_0 + t) \rangle = C_{PP}(t_0, t_0 + t) \tag{6.14}$$

(because of time-translation invariance)

$$C_{PP}(t_0, t_0 + t) = C_{PP}(j \Delta t) \tag{6.15}$$

(because of ergodicity)

$$C_{PP}(j \Delta t) = \frac{1}{N - j} \sum_{i=0}^{N-1-j} P(i \Delta t) P((i + j) \Delta t) \tag{6.16}$$

where j is the separation between frames, Δt is the sample interval, and N is the total number of frames for a 100 ps period. See Refs. [153, 154] for further details. As an example, consider separations of $j \in \{0, 1, 2, N - 1\}$ with $\Delta t = 100$ frames (equivalently

0.1 ps):

Average with zero lag at j = 0

$$C_{PP}(0) = \frac{1}{N}\sum_{i=0}^{N-1} P(100i)P(100i)$$

Average with 100 frame lag at j = 1

$$C_{PP}(100) = \frac{1}{N-1}\sum_{i=0}^{N-2} P(100i)P((i+1)100)$$

Average with 200 frame lag at j = 2

$$C_{PP}(200) = \frac{1}{N-2}\sum_{i=0}^{N-3} P(100i)P((i+2)100)$$

Average with maximum lag at j = N-1

$$C_{PP}(100(N-1)) = P(0)P((N-1)100)$$

Each time lag from $C_{PP}(j\Delta t)$ was averaged. For example, the very first step in MD produces one autocorrelation measure of P_{ij} with zero time lag. Then, the first 100 ps interval generates another 1000 values that are averaged with the previous for a mean value from 1001 observations with zero time lag. After 200 ps, there are a total of 2001 values with zero time lag to average. The same procedure was repeated for each possible time lag for the total 10 ns per isothermal hold. The integral with respect to each time lag average of the autocorrelation function of P_{ij} was used to compute viscosity with Equation 6.17 where $\tau = j\Delta t$.

$$\eta_{t_s} = \frac{V}{k_B T} \lim_{\delta\tau \to 0} \sum_{\tau=0}^{t_s} \delta\tau C_{PP}(\tau) \tag{6.17}$$

To ensure a settled viscosity measurement, the final 2 ns were gathered from NVT simulations, the gradient of viscosity with respect to time was taken to acquire slopes, and then the mean of the slopes was taken. If the mean slope was below 10^{-5} Pa, then data were considered stable and therefore converged. For the converged cases, the average over the final 2 ns was used to determine our viscosity measurement. We average values because there are some minor viscosity fluctuations as seen in Figure 6.2.

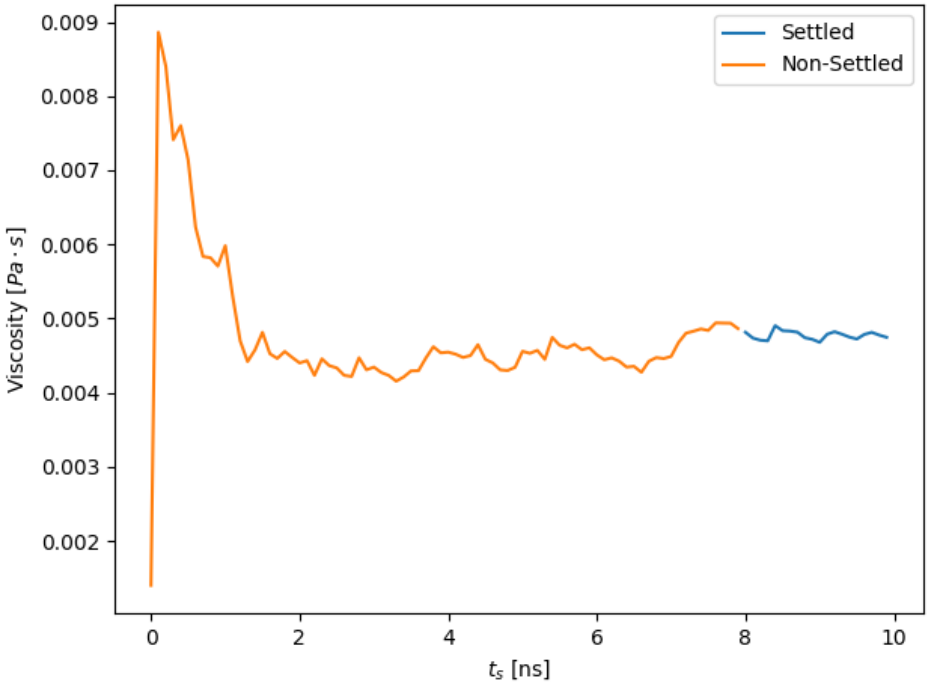


Figure 6.2: The running integral for viscosity of a single run of $Cu_{50}Zr_{50}$ at 1100 K. The curve is the pressure autocorrelation integrals taken every of 100 ps. The blue portion of the curve contains data used for convergence analysis of viscosity.

Table 6.10: The phase diagrams used from ASM International for T_l values.

System	Reference
Ag-Cu	Silver-Copper Binary Phase Diagram (2007 Cao W.)
Al-La	Aluminum-Lanthanum Binary Phase Diagram (2000 Okamoto H.)
Al-Zr	Aluminum-Zirconium Binary Phase Diagram (2002 Okamoto H.) a
Al-Cu	Aluminum-Copper Binary Phase Diagram (1991 Chen S.)
Al-Ti	Aluminum-Titanium Binary Phase Diagram (2012 Wang H.)
Al-Ni	Aluminum-Nickel Binary Phase Diagram (2005 Miettinen J.)
Al-Sm	Aluminum-Samarium Binary Phase Diagram (2007 Delsante S.)
Al-Co	Aluminum-Cobalt Binary Phase Diagram (2004 Ohtani H.)
Cu-Zr	Copper-Zirconium Binary Phase Diagram (2010 Kang D.H.) a
Fe-Ni	Iron-Nickel Binary Phase Diagram (1991 Swartzendruber L.J.) d
Mg-Y	Magnesium-Yttrium Binary Phase Diagram (2008 Guo C.)
Nb-Ni	Niobium-Nickel Binary Phase Diagram (2007 Tokunaga T.)
Ni-Ti	Nickel-Titanium Binary Phase Diagram (2010 Agraval P.G.)
Ni-Zr	Nickel-Zirconium Binary Phase Diagram (2007 Wang N.)
Pd-Si	Palladium-Silicon Binary Phase Diagram (2006 Du Z.) b

6.3 Appendix for Section 2.6

6.3.1 Validation of Viscosity Computation

Measuring viscosity via the Green-Kubo formalism is known to have long convergence times and can yield noisy results for individual runs. Therefore, we compared our acquisition of viscosity data with previously published work to ensure the quality of our calculations. For a single composition of $Cu_{50}Zr_{50}$, viscosities were measured through the same means described in the main text. We compared viscosity to Ref. [55] for the potential in Ref. [155]. The only point of major disagreement was the lowest temperature point near T_f, which is reasonable given the sluggish kinetics nearing material freezing

(Figure 6.3). Our data was averaged between 10 independent runs.

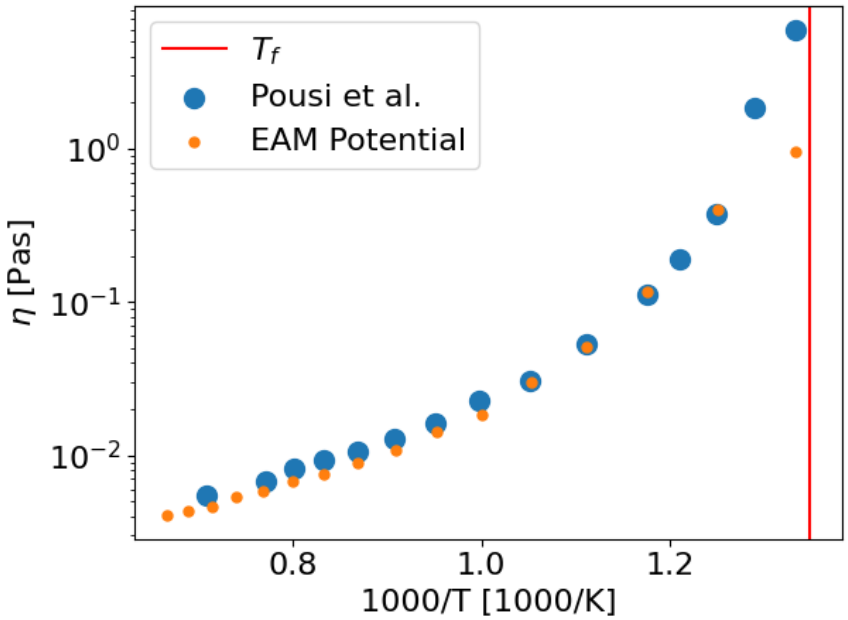


Figure 6.3: Comparison of viscosity.

6.3.2 Validation of MTP Fitting Methodology

Both classical potentials and MTPs were used to simulate the properties of potential energy, self-diffusion, and viscosity with respect to temperature and have been found to have excellent agreement (see figures below). The system size was 1,000 atoms when measuring these properties. Only $Ni_{80}P_{20}$ has slight, constant off-sets between the MLP and EAM potential for potential energy and viscosity with respect to temperature, and the offsets are small. More importantly, the relationship between temperature and potential energy, viscosity, and self-diffusion are similar. Properties were measured in the same manner described in Sec. 2.6.3.6.

6.3.2.1 Potential Energy

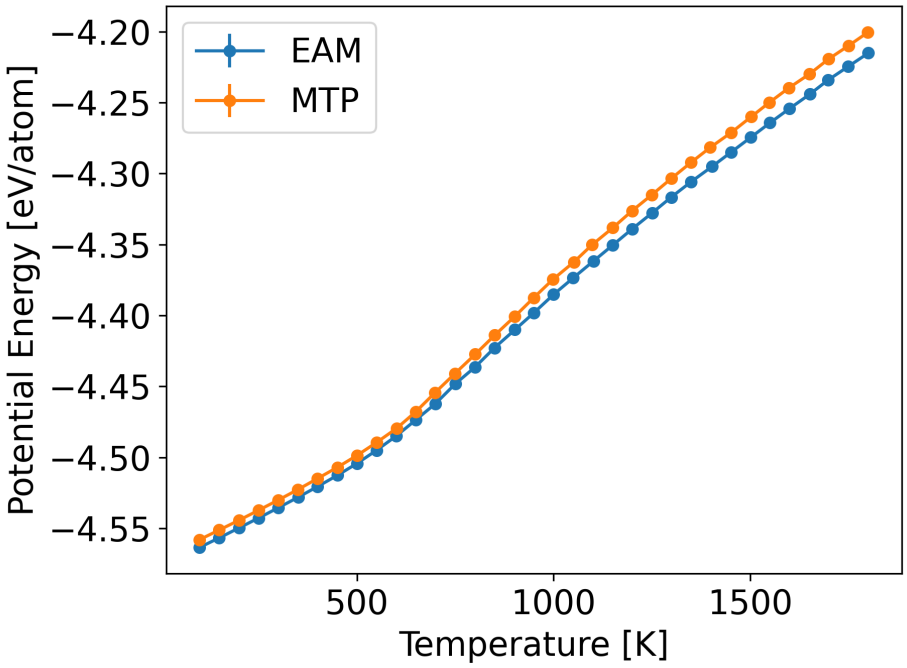


Figure 6.4: Comparison of potential energy between MLP and EAM potentials for $Ni_{80}P_{20}$.

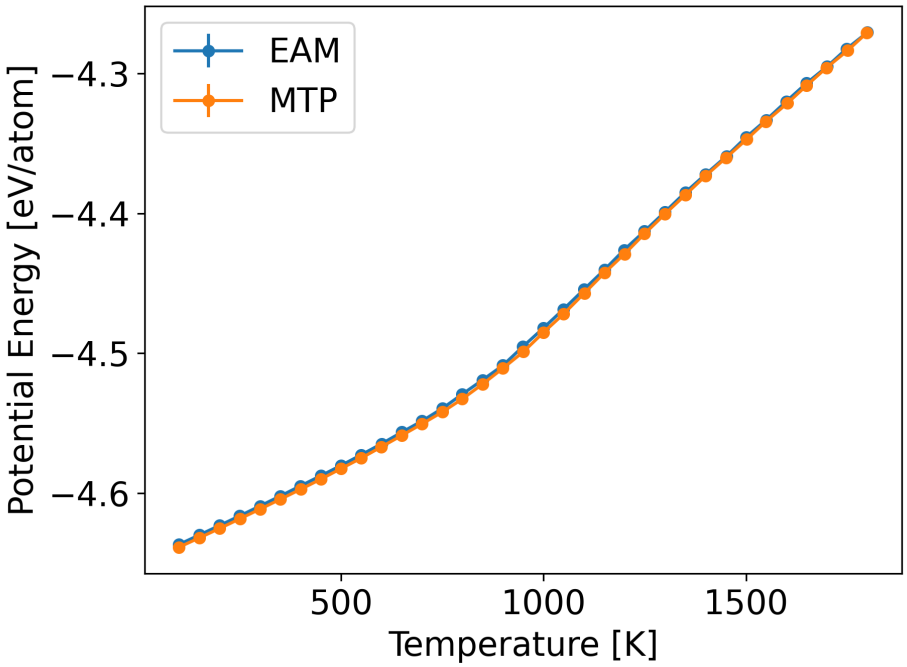


Figure 6.5: Comparison of potential energy between MLP and EAM potentials for $Pd_{75}Si_{25}$.

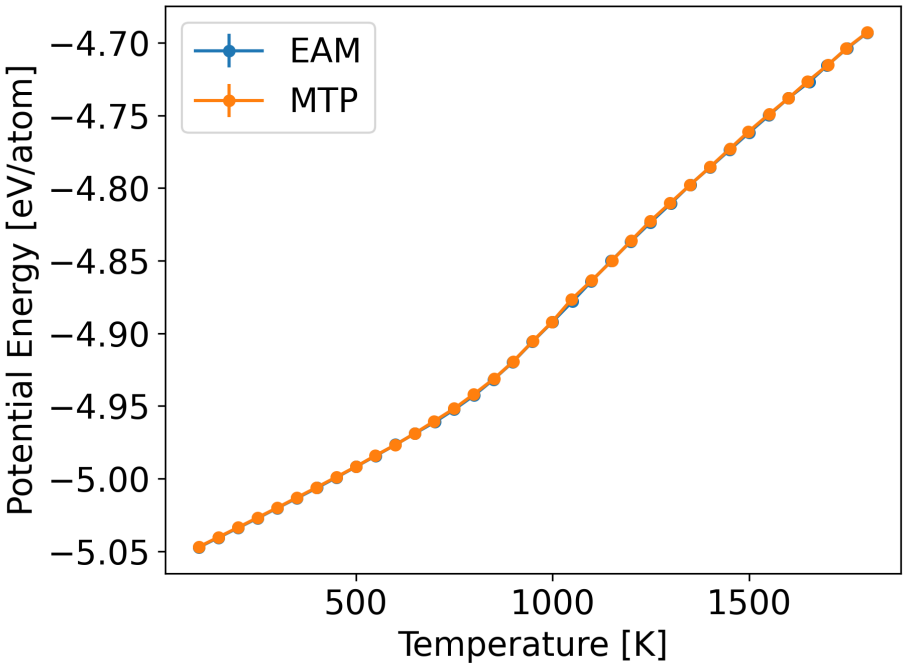


Figure 6.6: Comparison of potential energy between MLP and EAM potentials for $Al_{10}Cu_{40}Zr_{50}$.

6.3.2.2 Self-Diffusion

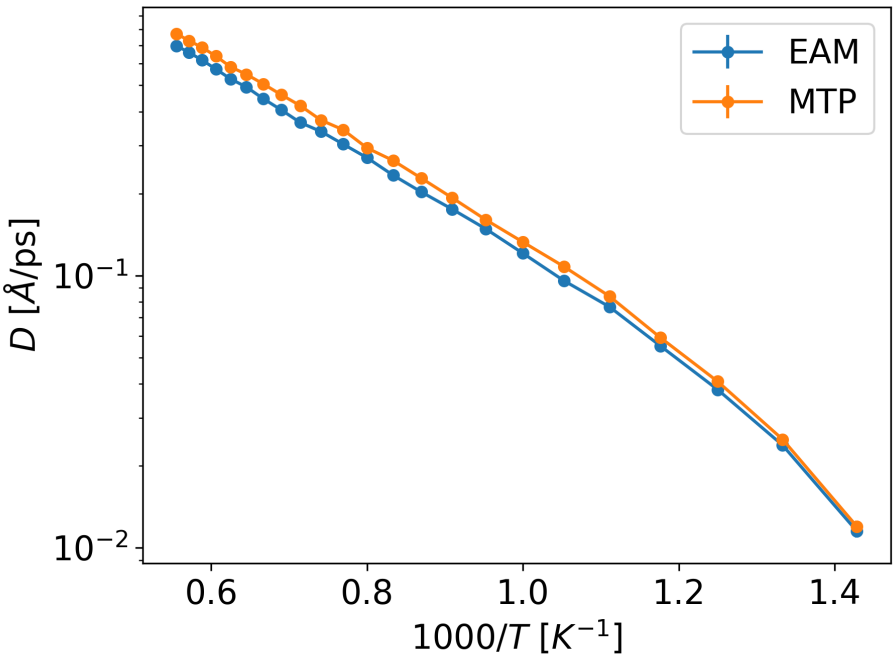


Figure 6.7: Comparison of self-diffusion between MLP and EAM potentials for $Ni_{80}P_{20}$.

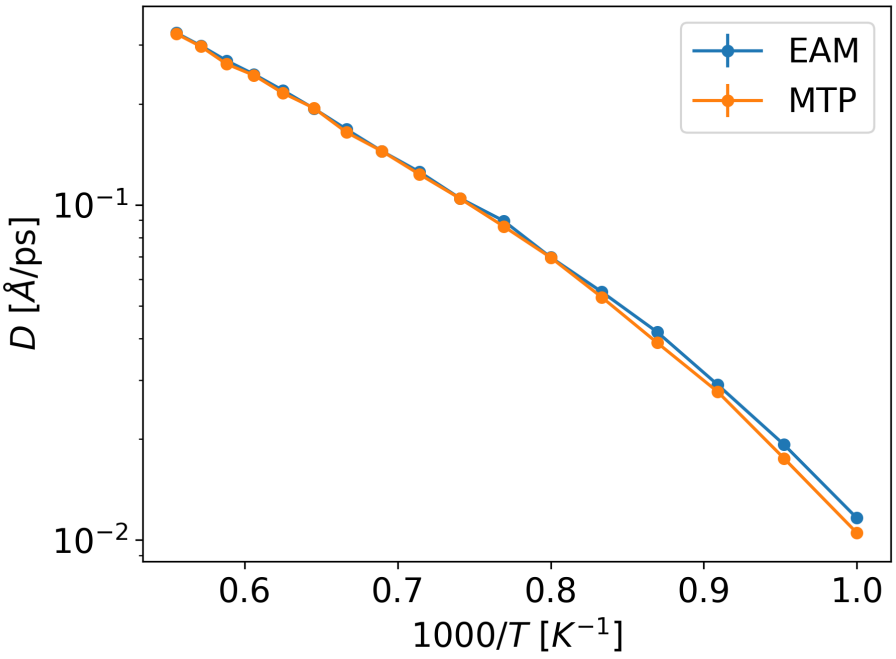


Figure 6.8: Comparison of self-diffusion between MLP and EAM potentials for $Pd_{75}Si_{25}$.

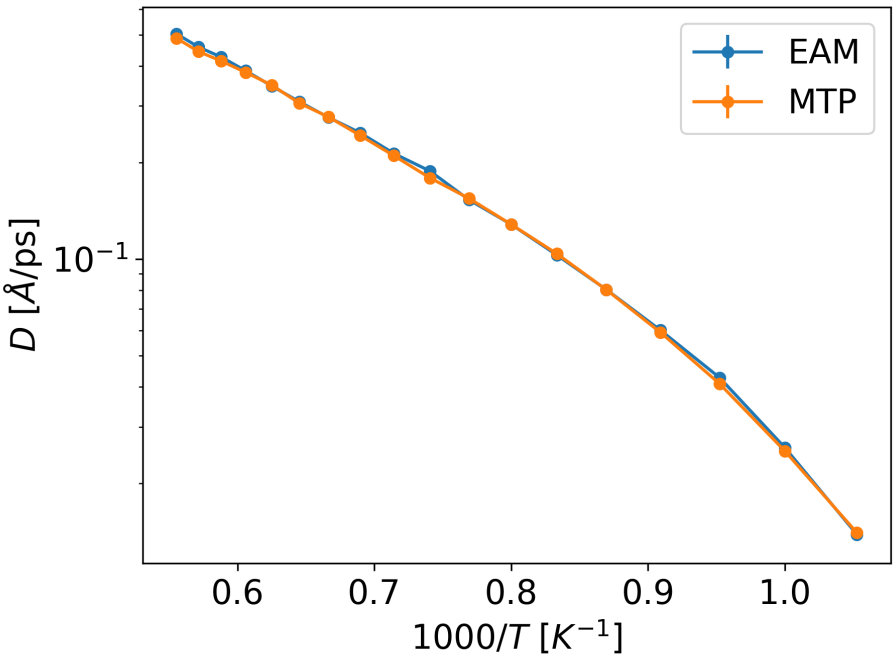


Figure 6.9: Comparison of self-diffusion between MLP and EAM potentials for $Al_{10}Cu_{40}Zr_{50}$.

6.3.2.3 Viscosity

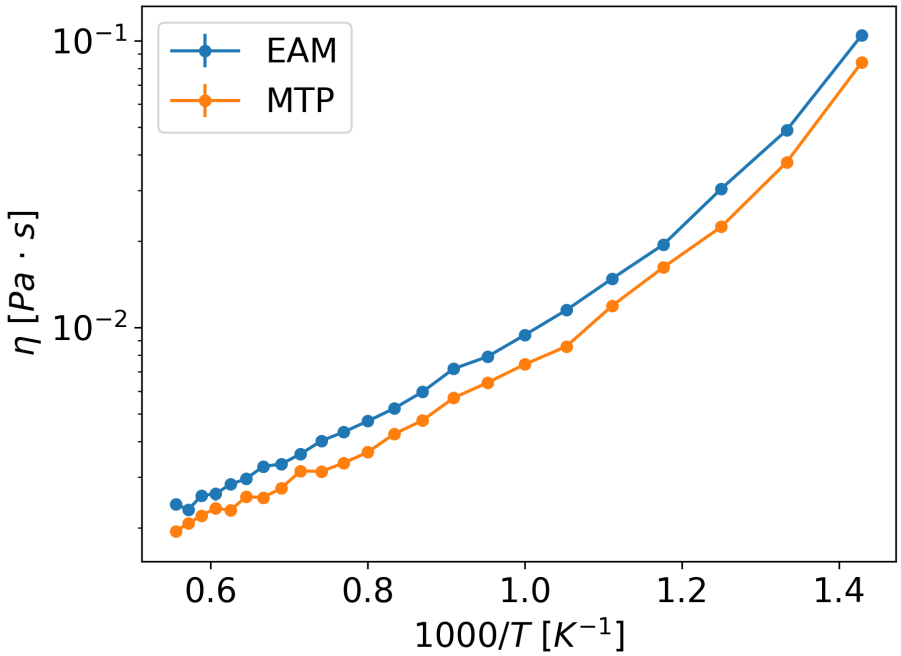


Figure 6.10: Comparison of viscosity between MLP and EAM potentials for $Ni_{80}P_{20}$.

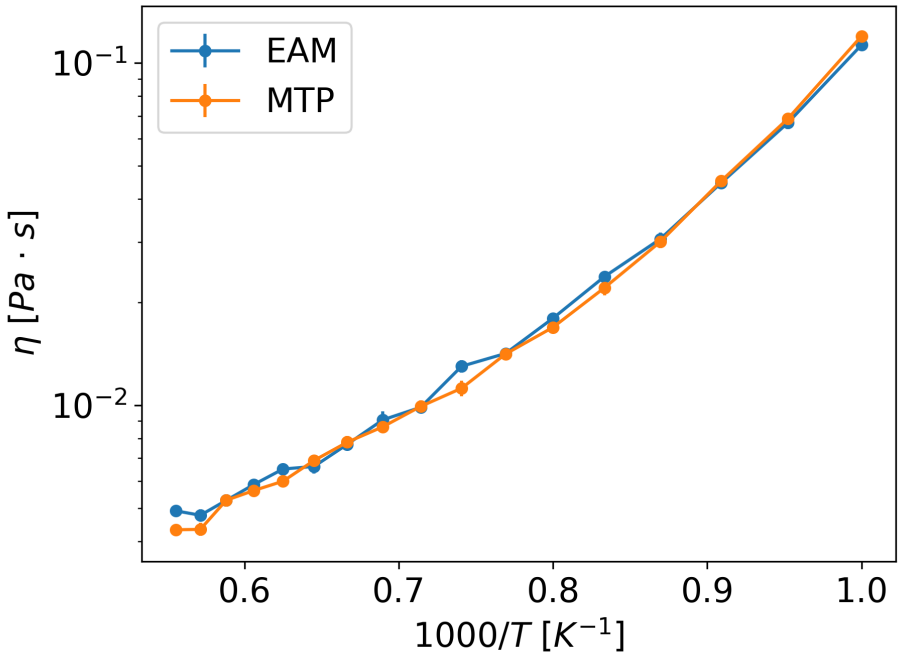


Figure 6.11: Comparison of viscosity between MLP and EAM potentials for $Pd_{75}Si_{25}$.

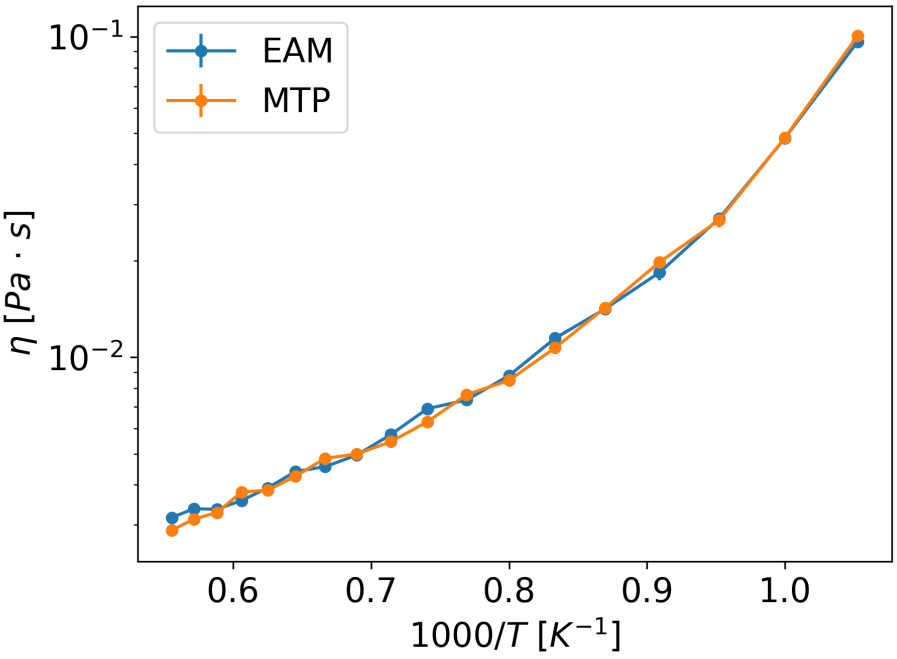


Figure 6.12: Comparison of viscosity between MLP and EAM potentials for $Al_{10}Cu_{40}Zr_{50}$.

6.3.3 CSLO CV Chemical Systems

The exact chemical system groups are tabulated in this section. For each iteration of CSLO CV, one group is selected and left out as a test set. For assessments on the 34 materials with MLP, entries from Table 6.11 are used. CV for larger assessments of 177 materials use entries from Table 6.12.

Table 6.11: The chemical systems considered for CSLO CV for the 34 materials with MLPs.

Chemical System	Count
MgNdNi	5
CuNiPPd	4
PdSi	3
AlCuNiZr	3
AlCuLaNi	3
AlCuLa	2
NiPPd	1
NbNiSiSnTiZr	1
NbNi	1
CuPdSi	1
CuNiTiZr	1
CuMgY	1
CuGdMg	1
CaMgZn	1
BeZr	1
BeTi	1
BeCuNiTiZr	1
AlNiZr	1
AlLaNi	1
AlCoCuLaNi	1

Table 6.12: The chemical systems considered for CSLO CV for 177 materials.

Chemical System	Count
CuNiPPd	15
AlCuNiZr	15
BeCuNiTiZr	11
MgNdNi	7
CuTiZr	6
CuZr	5
CuPdSi	4
CuGdMg	4
CaMgZn	4
BFe	4
AlCuZr	4
AlCuNiTiZr	4
AlCuLaNi	4
PdSi	3
NiZr	3
NbNi	3
BCCoCrFeMoY	3
AlFeNd	3
AlCoCuLaNi	3
NiPd	2
CuNiTiZr	2
CuNiPPt	2
AlCuNiPdZr	2
AlCuNbNiZr	2

Continued on next page

Table 6.12: The chemical systems considered for CSLO CV for 177 materials.

Chemical System	Count
AlCuLa	2
AlCoZr	2
AgAuCuPdSi	2
P	1
NiSiTiZr	1
NiSiSnTiZr	1
NiPPt	1
NiPPd	1
NbNiSiSnTiZr	1
GeSbTe	1
FeNiZr	1
CuPPdPt	1
CuPPd	1
CuNiSiTiZr	1
CuMgYZn	1
CuMgY	1
CuLa	1
CuHfTiZr	1
CoCuPPt	1
CFeP	1
BeZr	1
BeTi	1
BeCuNbNiZr	1
BZr	1
BNiSi	1

Continued on next page

Table 6.12: The chemical systems considered for CSLO CV for 177 materials.

Chemical System	Count
BFeSi	1
BFeNiP	1
BFeNi	1
BFeNbNiSi	1
BCoSi	1
BCoFeNbSi	1
BCoCrFeMoYZr	1
BCCrFeMoY	1
BCCrFeMnMoSiW	1
BCCrFeGaMoP	1
AuGeSi	1
AlNiZr	1
AlMg	1
AlLaNi	1
AlGdNiZn	1
AlGdNiSn	1
AlGdMnNi	1
AlCuNbZr	1
AlCuHfNi	1
AlCuGdZr	1
AlCuGdNi	1
AlCoSm	1
AlCoLaNiY	1
AlCoGd	1
AlCoFeNiY	1

Continued on next page

Table 6.12: The chemical systems considered for CSLO CV for 177 materials.

Chemical System	Count
AlCoCuZr	1
AlCoCuFeSm	1
AlCeNi	1
AgCuZr	1
AgCuMgNiYZn	1
AgCuGdMgPd	1
AgCuGdMgNiYZn	1
AgAlCuZr	1
AgAlCuNiZr	1

6.3.4 Parity Plots

We provide the parity plots for several sets of data and assessments in this section. We emphasized the use of CSLO CV throughout the main text because we wanted models that could predict well across chemical systems, and include all relevant data here. However, 5-fold CV (repeated 10 times) is a common way to assess models, so we provide those parity plots as well. Also, because of the small set of data with properties acquired from 34 MLPs, we provide plots for leave one out CV for the cases where only 34 data points are fit. The number of points in these data are small and will lead to poor performance in 5-fold CV.

6.3.4.1 CSLO CV

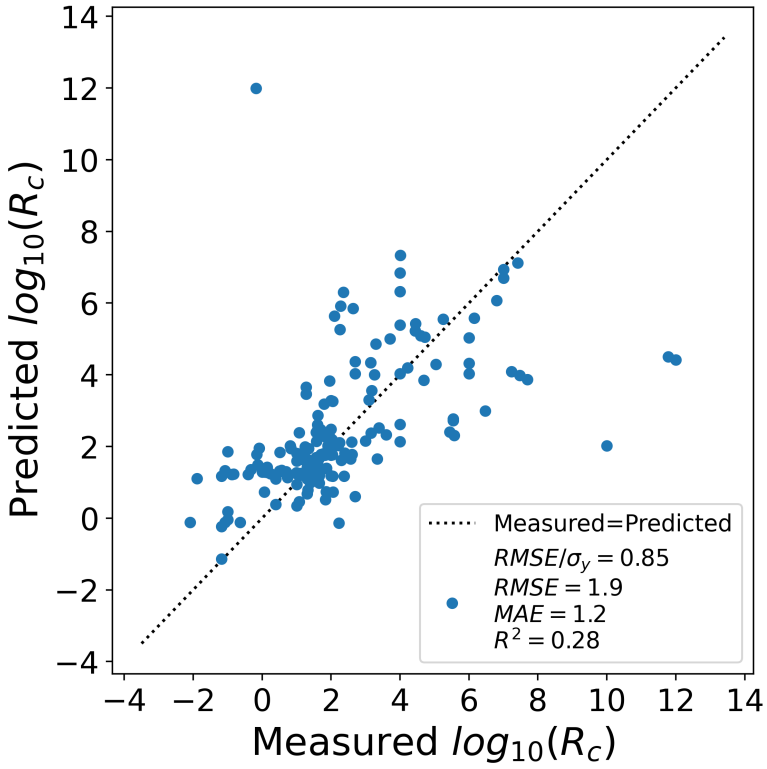


Figure 6.13: The parity plot for XGBoost models fit to X_{long} for 177 materials. Note that test data were produced by CSLO CV.

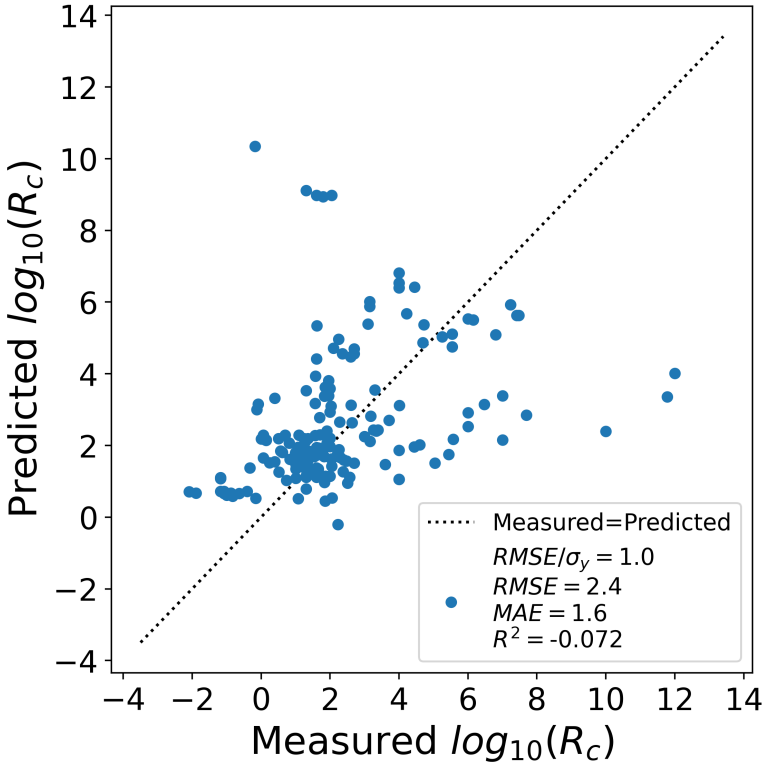


Figure 6.14: The parity plot for XGBoost models fit to X_{mastml} for 177 materials. Note that test data were produced by CSLO CV.

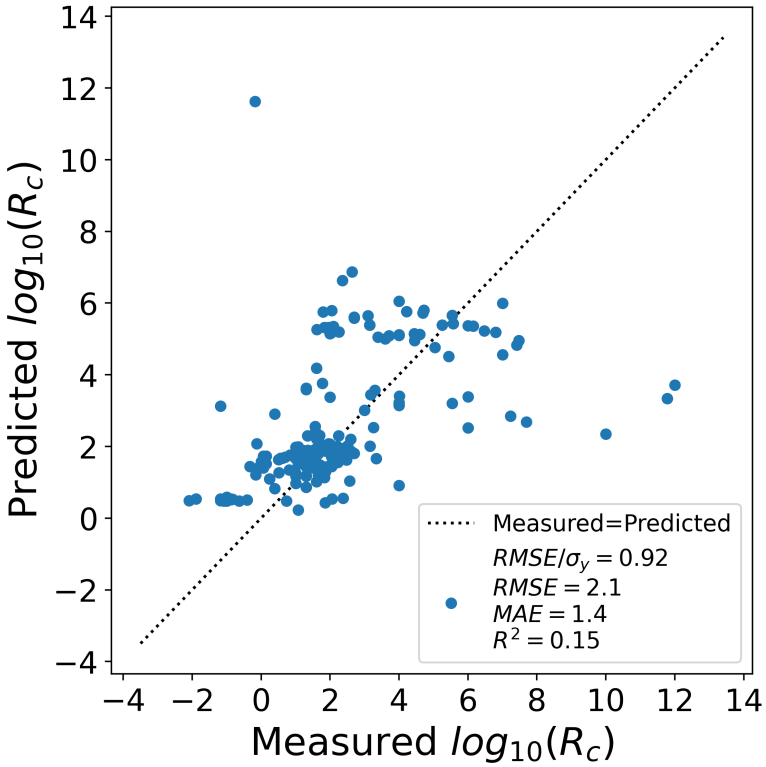


Figure 6.15: The parity plot for XGBoost models fit to $X_{mastml} \cup X_{long}$ for 177 materials. Note that test data were produced by CSLO CV.

6.3.4.2 5-Fold CV Repeated 10 Times

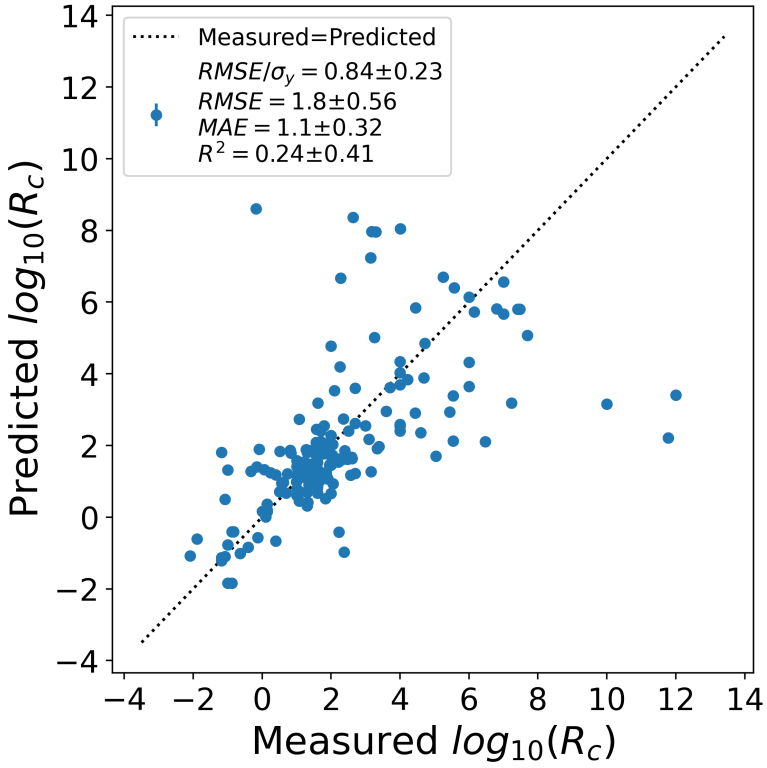


Figure 6.16: The parity plot for XGBoost models fit to X_{long} for 177 materials. Note that test data were produced by 5-fold CV repeated 10 times.

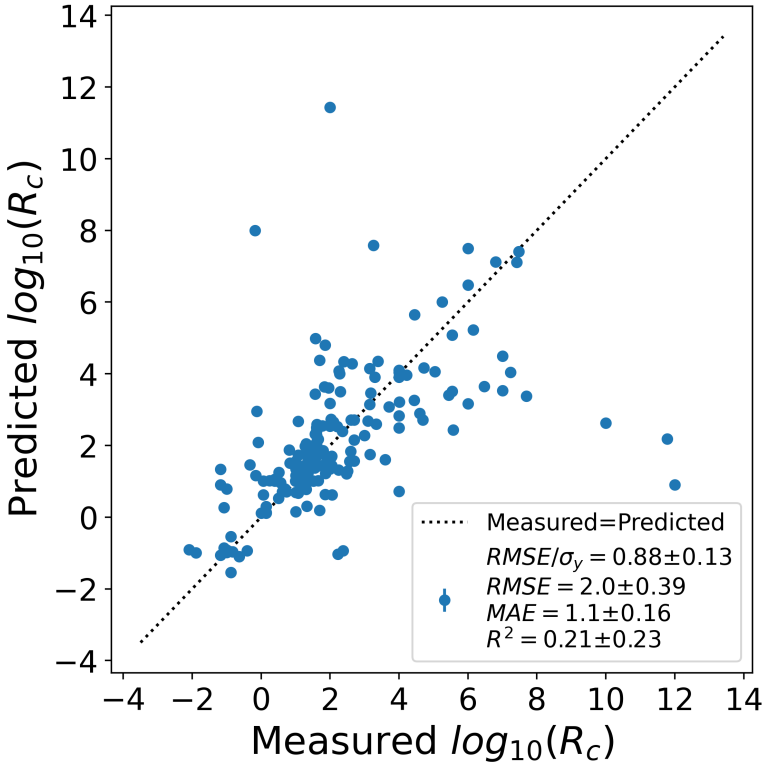


Figure 6.17: The parity plot for XGBoost models fit to X_{mastml} for 177 materials. Note that test data were produced by 5-fold CV repeated 10 times.

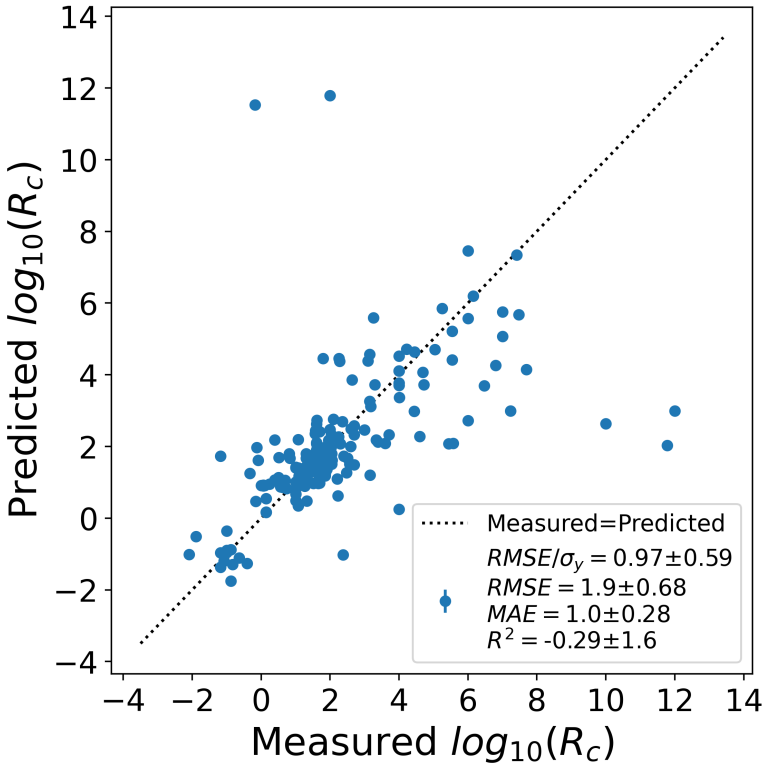


Figure 6.18: The parity plot for XGBoost models fit to $X_{mastml} \cup X_{long}$ for 177 materials. Note that test data were produced by 5-fold CV repeated 10 times.

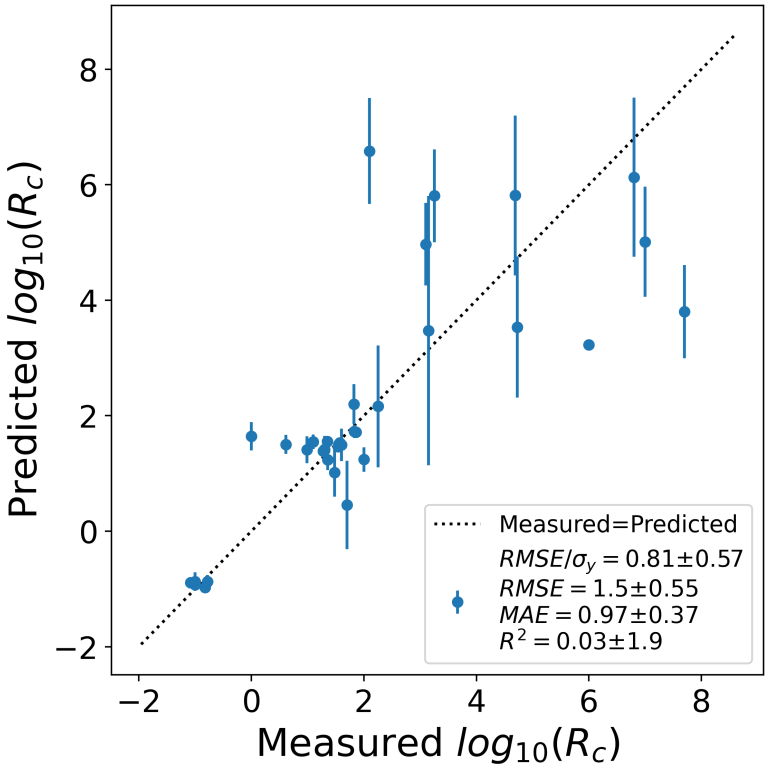


Figure 6.19: The parity plot for XGBoost models fit to X_{long} for 34 (materials with a built MLP). Note that test data were produced by 5-fold CV repeated 10 times.

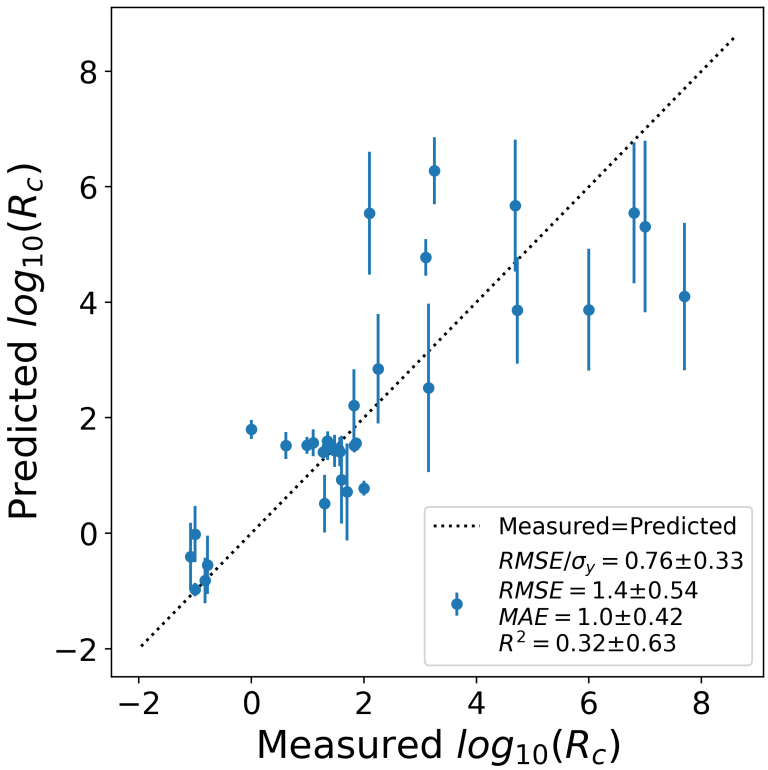


Figure 6.20: The parity plot for XGBoost models fit to X_{tot} for 34 (materials with a built MLP). Note that test data were produced by 5-fold CV repeated 10 times.

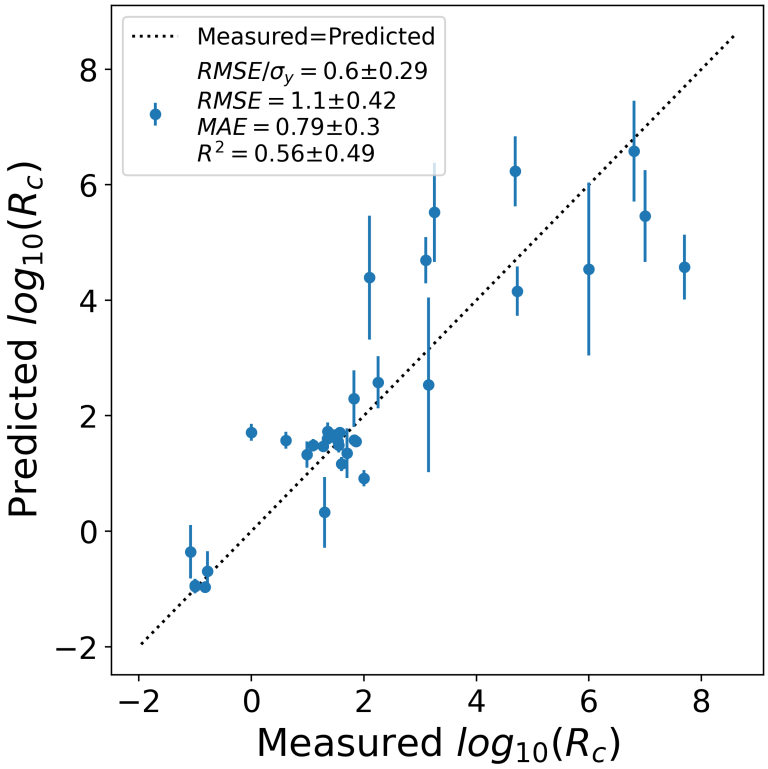


Figure 6.21: The parity plot for XGBoost models fit to X_{best} for 34 (materials with a built MLP). Note that test data were produced by 5-fold CV repeated 10 times.

6.3.4.3 Leave One Out CV

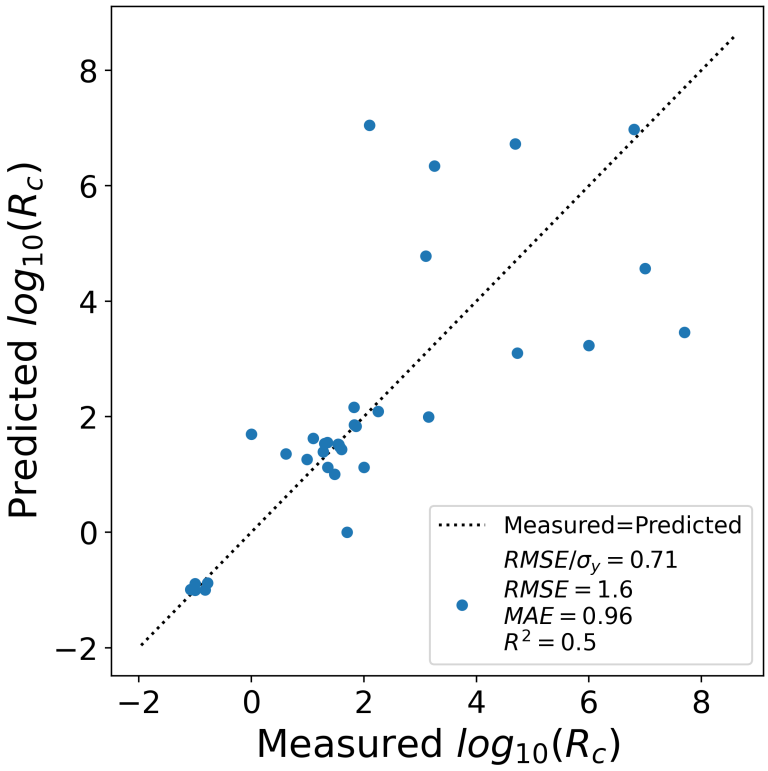


Figure 6.22: The parity plot for XGBoost models fit to X_{long} for 34 (materials with a built MLP). Note that test data were produced by Leave One Out CV.

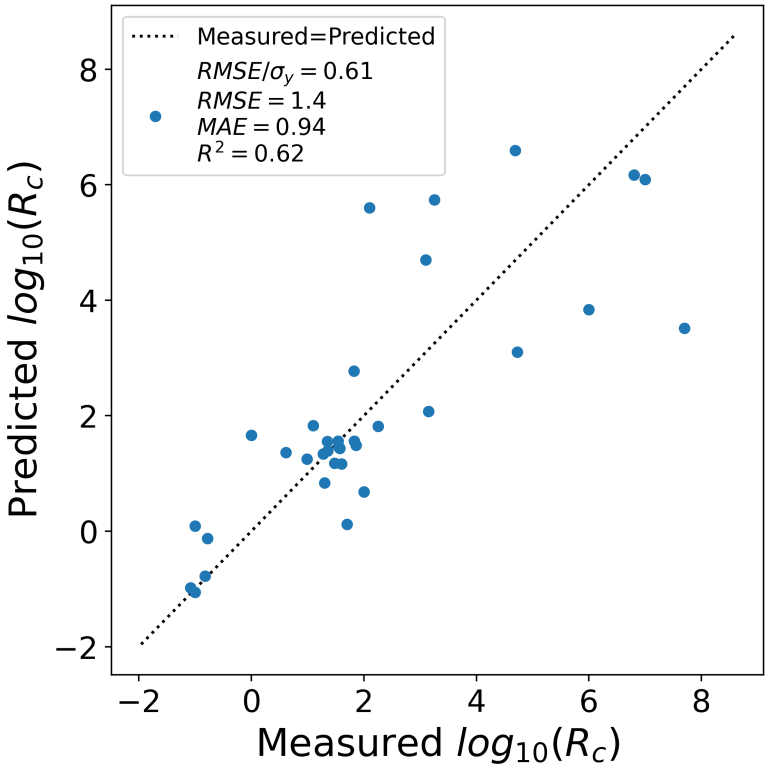


Figure 6.23: The parity plot for XGBoost models fit to X_{tot} for 34 (materials with a built MLP). Note that test data were produced by Leave One Out CV.

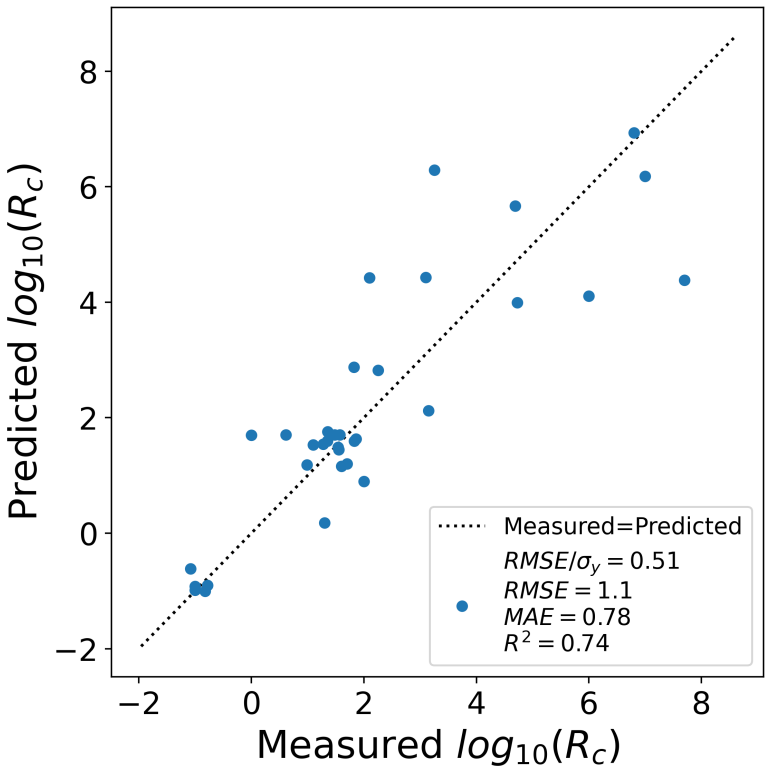


Figure 6.24: The parity plot for XGBoost models fit to X_{best} for 34 (materials with a built MLP). Note that test data were produced by Leave One Out CV.

6.4 Appendix for Chapter 3

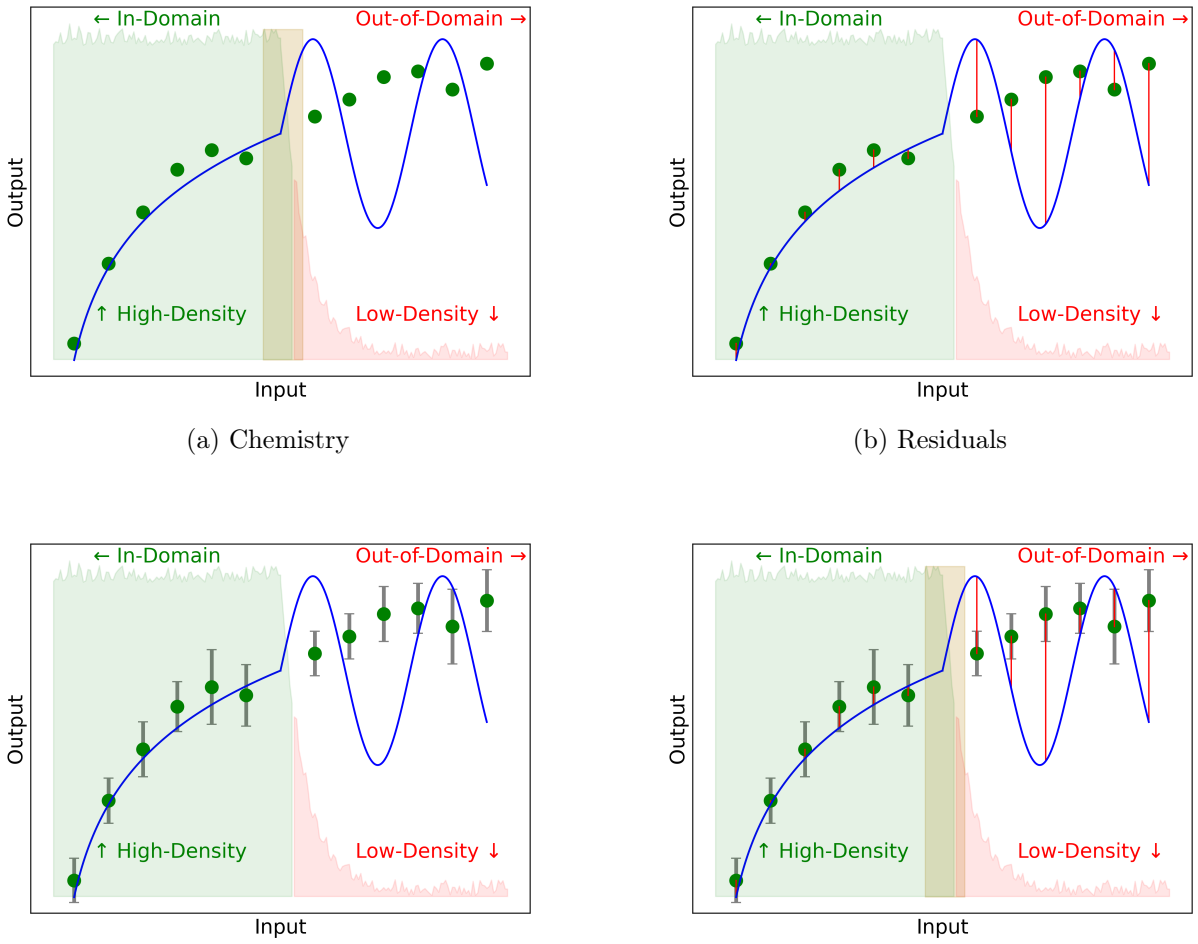
6.4.1 Abstract Illustration of Domain

Figure 6.25a through 6.25c illustrate the rationale for sets of privileged information derived from chemical intuition, residuals, and uncertainties. We provide an abstract illustration of determining domains from chemical information in Figure 6.25a. The true behavior of a physical relationship is shown by the blue line, while the estimates of the measure by M^{prop} are represented by the green dots. The green area depicts the region with ample amounts of ITB data and likely ID, while the red areas indicate regions with few ITB data and likely OD. Specialists have an intuition that the physical response changes between ID and OD regions due to a physical change (e.g., phase transition), which is represented by the brown shaded region. M^{prop} fails to correctly predict the true behavior in the low-density (or OD) region of space, but an expert in materials science may have an intuition that the model would fail. The brown shaded region, in essence, represents the separation of materials that are ID from those that are OD via chemical intuition.

As an alternative to chemical intuition, we provide methodologies to discern domains through residual information. An abstract illustration for domain based on $E^{|y-\hat{y}|/MAD_y}$ and E^{RMSE/σ_y} are shown in Figure 6.25b. The green dots, blue line, and shaded regions have the same meaning as in Figure 6.25a. As seen by the residuals (i.e., the red vertical lines) in Figure 6.25b, the predictions of M^{prop} would, on average, have larger E^{RMSE/σ_y} for the entirety of an OD region compared to an ID region. However, individual measures of $E^{|y-\hat{y}|/MAD_y}$ can be randomly low for some data points in an OD region. The cutoffs based on residuals provide a bound for how poor predictions on a target property become before designating those predictions as OD.

Finally, domain can be considered in terms of the quality of uncertainty estimates. The abstract illustration for domain based on E^{area} is shown in Figure 6.25c. The green dots, blue line, and shaded regions have the same meaning as in Figure 6.25a and 6.25b. Each green dot in the ID region has an uncertainty estimate from M^{unc} that includes the true

behavior (blue) within its bounds. Conversely, the OD region has many uncertainties that may not include the true behavior within their bounds. The cutoff based on uncertainty estimate quality (i.e., E^{area}) separates regions where uncertainties are unreliable (i.e., OD regions) from other regions where uncertainties are more reliable (i.e., ID regions).



(a) Chemistry

(b) Residuals

(c) Uncertainties

(d) All Together

Figure 6.25: Conceptual illustrations of domain through various types of privileged information are shown. The ground truth data are represented by the blue line and is assumed to suddenly change its behavior from a square root type function to an oscillating sinusoidal type function in the middle of each figure (Figure 6.25a-6.25d). Examples of model inference are represented by green points. It is assumed that there is a large amount of training data (shown by the large amount of green area) in the region on the left side of each figure. It is also assumed that there is a small amount of training data (shown by the small amount of red area) in the region on the right side of each figure and decreases as one moves away from the boundary with the left side. Inference on data on the right side of each figure is OD, as illustrated by the clear change in the nature of underlying ground truth data, which might occur due to changes in chemistry in material applications (brown division in Figure 6.25a), the large magnitude of residuals (red vertical lines in Figure 6.25b), and the incorrect uncertainty estimates (gray error bars in Figure 6.25c). Figure 6.25d shows all these aspects together.

6.4.2 Poor M^{unc} Lead to Poor F1 Scores in A^{area}

We further illustrate the point that poor uncertainty models (M^{unc}) impact M^{dom}'s ability to discern domain through A^{area}. Ideally, measures of σ_c normalized by σ_y correlate with E^{RMSE/σ_y} with a slope of 1 and an intercept of 0. We show a near ideal relationship with the Fluence data set using RF in Figure 6.26. These OOB data were generated through Repeated 5-fold CV only (i.e., excluding the data from the BLOCO procedure), as described in the Materials and Methods section in the main text. Data were binned into 10 bins.

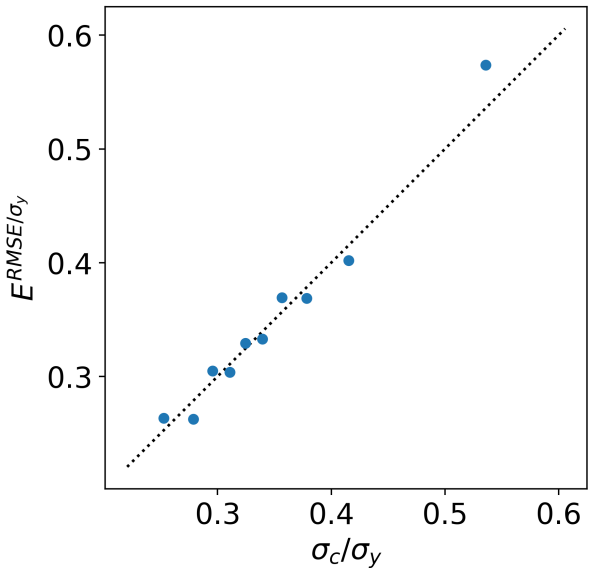


Figure 6.26: The correlation between E^{RMSE/σ_y} and σ_c/σ_y is shown for the Fluence data set with RF.

We gathered slopes and intercepts of M^{unc} models computed similarly to the previous example and compared $F1_{max}$ scores for A^{area}. As depicted in Figure 6.27, there is a relationship between the quality of σ_c and an M^{dom}'s ability to discern domains via the $F1$ score. Any data with $F1_{max}$=1.0 occur near an intercept of 0 and slope of 1. In other words, our capability to distinguish ID from OD deteriorates when the quality of M^{unc} is worse, which is further reinforced by case (ii) in the Notes of Caution for Domain Prediction section.

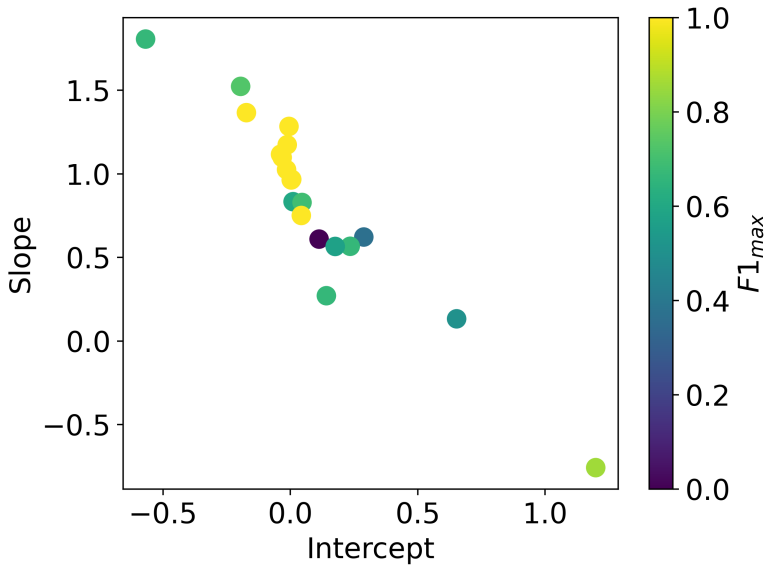


Figure 6.27: The relationship between the slopes, intercepts, and $F1_{max}$ scores for fit M^{unc} models and their ability to correlate E^{RMSE/σ_y} and σ_c/σ_y is shown.

6.4.3 The Use of MinMaxScaler instead of the StandardScaler from Scikit-Learn

We show that our method is not overly sensitivity to the method of feature (X) scaling in this section. We substituted the use of scikit-learn's StandardScaler with the MinMaxScaler [42]. The MinMaxScaler scales features to be in the range $[0, 1]$. The results are shown in Table 6.13 for the RF M^{prop} model type. We took the $F1_{max}$ scores from the results using a StandardScaler (i.e., the results reported in the main text) and subtracted them from the $F1_{max}$ scores using the MinMaxScaler and called the result $\Delta F1$. All $\Delta F1$ are close to zero and show that the use of either the MinMaxScaler or StandardScaler will provide similar results.

Table 6.13: Precision and recall data are tabulated for use the MinMaxScaler instead of the StandardScaler from scikit-learn. The d_c^t, precision, recall, and $F1$ are for $F1_{max}$.

Data	Assessment	Baseline	AUC	AUC-Baseline	d_c^t	Precision	Recall	$F1$	$\Delta F1$
Friedman	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.78	0.96	0.18	1.00	0.97	0.85	0.91	-0.02
Friedman	A^{RMSE/σ_y}	0.65	1.00	0.35	1.00	1.00	1.00	1.00	0.00
Friedman	A^{area}	0.29	1.00	0.71	0.64	1.00	1.00	1.00	0.00
Fluence	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.74	0.90	0.16	1.00	0.74	1.00	0.85	0.00
Fluence	A^{RMSE/σ_y}	0.62	1.00	0.38	0.99	1.00	1.00	1.00	0.00
Fluence	A^{area}	0.46	1.00	0.54	0.90	1.00	1.00	1.00	0.00
Diffusion	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.56	0.95	0.38	1.00	0.96	0.88	0.92	0.00
Diffusion	A^{RMSE/σ_y}	0.50	1.00	0.50	1.00	1.00	1.00	1.00	0.00
Diffusion	A^{area}	0.22	1.00	0.78	0.56	1.00	1.00	1.00	0.00
Steel Strength	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.62	0.82	0.21	1.00	0.62	1.00	0.76	0.02
Steel Strength	A^{RMSE/σ_y}	0.47	1.00	0.53	0.83	1.00	1.00	1.00	0.00
Steel Strength	A^{area}	0.33	1.00	0.67	0.61	1.00	1.00	1.00	0.00
Superconductor	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.62	0.76	0.14	1.00	0.62	1.00	0.77	0.00
Superconductor	A^{RMSE/σ_y}	0.45	0.78	0.32	0.89	1.00	0.60	0.75	0.01
Superconductor	A^{area}	0.27	1.00	0.73	0.89	1.00	1.00	1.00	0.00

6.4.4 KDE Hyper Parameter Selection

Only the Diffusion data with an RF model type was used to examine parameters such as KDE bandwidth and kernel type. While it is up to the user to select appropriate parameters for their use case, we nevertheless provide what we believe are the best recommended values for each aforementioned parameter: the KDE bandwidth should be determined by scikit-learn's bandwidth estimator and the kernel type should be Epanechnikov.

6.4.4.1 Kernel Choice

We used the automated bandwidth selection method and studied the effect of kernel choice on our precision and recall scores. Cosine, linear, tophat, and Epanechnikov kernels decay to zero after some distance away from an observation (i.e., they are bounded). We wanted a KDE to have a value of zero for unobserved cases outside a bandwidth, which would not occur with Gaussian and exponential kernels. Out of the remaining kernels, the AUC-Baseline scores were comparable. We opted for the Epanechnikov kernel because of its

common usage, yet other kernels may be equally suitable for our application. Table 6.14 contains the results for $A^{|y-\hat{y}|/MAD_y}$, A^{RMSE/σ_y}, and A^{area}.

Table 6.14: Kernel types were compared with the automated bandwidth selection method. Precision and recall scores were tabulated. The d_c^t, precision, and recall are for $F1_{max}$.

Assessment	Baseline	AUC	AUC-Baseline	d_c^t	Precision	Recall	$F1$	Kernel
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.58	0.99	0.41	0.97	0.96	1.00	0.98	gaussian
A^{RMSE/σ_y}	0.60	1.00	0.40	0.94	1.00	1.00	1.00	gaussian
A^{area}	0.20	1.00	0.80	0.21	1.00	1.00	1.00	gaussian
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.57	0.95	0.39	0.98	0.97	0.90	0.93	tophat
A^{RMSE/σ_y}	0.53	1.00	0.47	0.98	1.00	1.00	1.00	tophat
A^{area}	0.12	0.45	0.33	0.37	0.67	1.00	0.80	tophat
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.58	0.95	0.37	1.00	0.98	0.90	0.94	epanechnikov
A^{RMSE/σ_y}	0.53	1.00	0.47	1.00	1.00	1.00	1.00	epanechnikov
A^{area}	0.18	1.00	0.82	0.46	1.00	1.00	1.00	epanechnikov
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.58	0.99	0.41	0.85	0.97	1.00	0.98	exponential
A^{RMSE/σ_y}	0.60	1.00	0.40	0.84	1.00	1.00	1.00	exponential
A^{area}	0.20	1.00	0.80	0.23	1.00	1.00	1.00	exponential
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.57	0.96	0.39	1.00	0.96	0.91	0.94	linear
A^{RMSE/σ_y}	0.53	1.00	0.47	0.99	1.00	1.00	1.00	linear
A^{area}	0.24	1.00	0.76	0.60	1.00	1.00	1.00	linear
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.57	0.95	0.38	1.00	0.95	0.90	0.92	cosine
A^{RMSE/σ_y}	0.53	1.00	0.47	1.00	1.00	1.00	1.00	cosine
A^{area}	0.24	1.00	0.76	0.59	1.00	1.00	1.00	cosine

6.4.4.2 Bandwidth Selection

To check if the automated bandwidth selection method was appropriate, $A^{|y-\hat{y}|/MAD_y}$, A^{RMSE/σ_y}, and A^{area} were performed with a grid of bandwidths using the Epanechnikov kernel. The precision and recall scores are in Table 6.15. Our precision and recall for the automated bandwidth selection method are comparable to the best from the gridding procedure but have the additional benefit of preventing over-fitting due bandwidth tuning from a user. As shown in Table 6.15, the precision and recall scores for $F1_{max}$ for the automated method is highest for the A^{RMSE/σ_y} and A^{area} and nearly the highest for $E^{|y-\hat{y}|/MAD_y}$. This means that ID/OD points are separated well with the added benefit of no user input for the bandwidth.

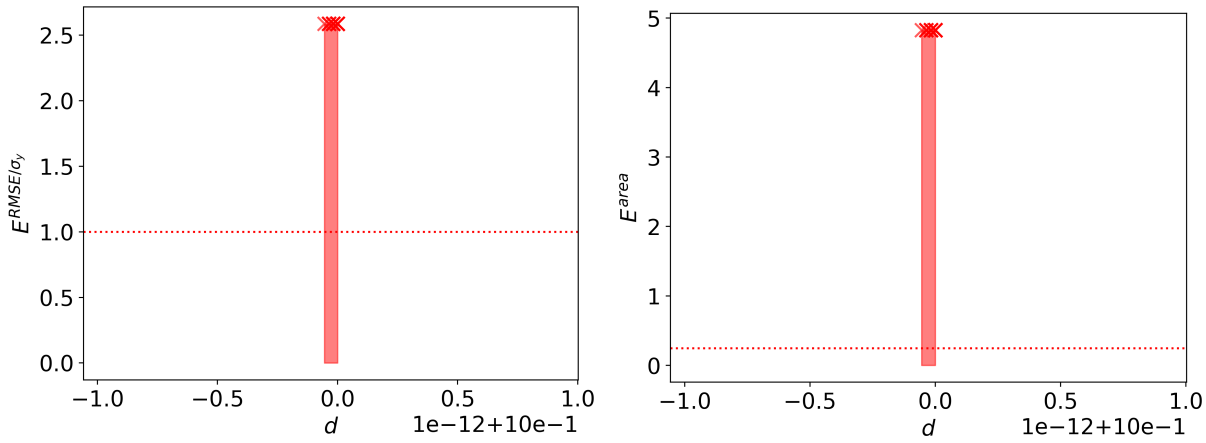
Table 6.15: The effects of bandwidth on precision and recall scores are tabulated. The d_c^t, precision, and recall are for $F1_{max}$. The kernel of choice was Epanechnikov. The automated bandwidth selection method was used for all bandwidth entries that are not 0.001, 0.01, 0.1, 1.0, 10.0, 100.0, or 1000.0. The automated bandwidth shown is from all X. For each fold, the bandwidth can change depending on the subset of data analyzed. A value of negative infinity for d_c^t means all data are OD. A value of infinity for d_c^t means all data are ID.

Assessment	Baseline	AUC	AUC-Baseline	d_c^t	Precision	Recall	$F1$	Bandwidth
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.54	0.55	0.01	1.00	0.54	1.00	0.70	0.001
a^{RMSE/σ_y}	0.00	0.00	0.00	$-\infty$	0.00	0.00	0.00	0.001
A^{area}	0.00	0.00	0.00	$-\infty$	0.00	0.00	0.00	0.001
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.68	0.68	0.00	1.00	0.68	1.00	0.81	0.01
A^{RMSE/σ_y}	0.00	0.00	0.00	$-\infty$	0.00	0.00	0.00	0.01
A^{area}	0.00	0.00	0.00	$-\infty$	0.00	0.00	0.00	0.01
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.65	0.67	0.02	1.00	0.65	1.00	0.78	0.1
A^{RMSE/σ_y}	0.04	1.00	0.96	1.00	1.00	1.00	1.00	0.1
A^{area}	0.01	1.00	0.99	0.76	1.00	1.00	1.00	0.1
A^{RMSE/σ_y}	0.35	1.00	0.65	1.00	1.00	1.00	1.00	1.0
A^{area}	0.31	1.00	0.69	1.00	1.00	1.00	1.00	1.0
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.57	0.83	0.26	1.00	0.99	0.61	0.75	1.0
A^{RMSE/σ_y}	0.53	1.00	0.47	0.99	1.00	1.00	1.00	4.283
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.57	0.96	0.39	1.00	0.97	0.92	0.94	4.283
A^{area}	0.18	1.00	0.82	0.46	1.00	1.00	1.00	4.283
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.59	0.99	0.40	0.89	0.97	0.99	0.98	10.0
A^{RMSE/σ_y}	0.60	1.00	0.40	0.96	1.00	1.00	1.00	10.0
A^{area}	0.20	0.39	0.19	0.16	0.60	1.00	0.75	10.0
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.57	0.96	0.39	0.02	0.95	0.99	0.97	100.0
A^{RMSE/σ_y}	0.60	1.00	0.40	0.02	1.00	1.00	1.00	100.0
A^{area}	0.20	0.45	0.25	0.00	0.67	1.00	0.80	100.0
$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.57	0.97	0.40	0.00	0.95	0.99	0.97	1000.0
A^{RMSE/σ_y}	0.60	1.00	0.40	0.00	1.00	1.00	1.00	1000.0
A^{area}	0.10	0.19	0.09	0.00	0.33	1.00	0.50	1000.0

6.4.5 Comment on Binning and Effect on Ground Truth Assignment

The assignment of ID/OD labels to bins with E^{RMSE/σ_y} or E^{area} is dependent on the bins into which they are grouped. It is important to note that any ground truth cutoff value remains unchanged (E_c^{chem}, $E_c^{|y-\hat{y}|/MAD_y}$, E_c^{RMSE/σ_y}, and E_c^{area}). The change in ground truth assignment is reflected by the baseline column in Table 6.15, which represents the

ratio of the number of ID bins to the total number of bins (10 in our study). Figure 6.28 illustrates A^{RMSE/σ_y} and A^{area} for the entries with a bandwidth of 0.001 in Table 6.15. All data points have the same value of d=1.0, and those that do not are extremely close to 1.0, resulting in their grouping into a single bin. The mixing of "good" and "bad" predictions lead to all data being defined as OD. With more intelligent values of d (e.g., KDE with automated bandwidth method covered in the main text), we can group more bins that are ID to the left of OD. It is crucial to note that $A^{|y-\hat{y}|/MAD_y}$, unlike A^{RMSE/σ_y} or A^{area}, does not require binning. Consequently, the baseline value remains relatively consistent for $E^{|y-\hat{y}|/MAD_y}$ entries in Table 6.15.

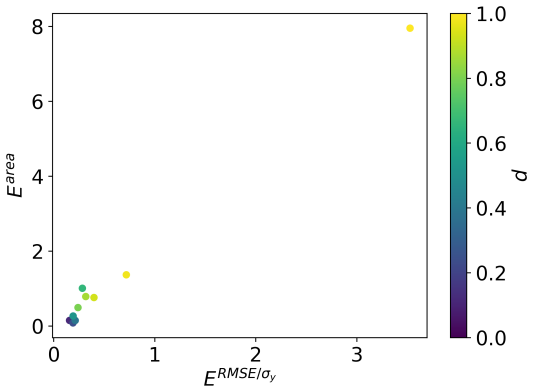


(a) A^{RMSE/σ_y} (b) A^{area}

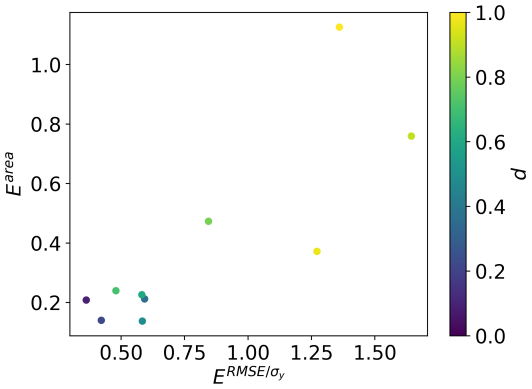
Figure 6.28: A^{RMSE/σ_y} and A^{area} are shown for the bandwidth value of 0.001 covered in Bandwidth Selection section.

6.4.6 Relationship Between E^{RMSE/σ_y}, E^{area}, and d

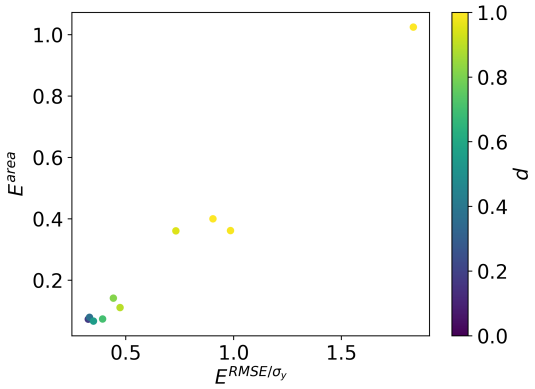
Here we describe the results of assessments A^{RMSE/σ_y} and A^{area} for RF. Both tests rely on the intuition that predictions on data increasingly dissimilar to that used for model training should be increasingly unreliable. As d increases, so should E^{RMSE/σ_y} and E^{area} as shown in Figure 6.29 for the RF model type. While E^{area} shares some similarities with E^{RMSE/σ_y}, it is important to clarify that they are not identical. E^{area} offers additional insights by incorporating information about the quality of uncertainty estimates. The information given by these trends can be used to learn d_c^t for when E^{RMSE/σ_y} (or E^{area}) grows beyond its corresponding E_c^{RMSE/σ_y} (or E_c^{area}).



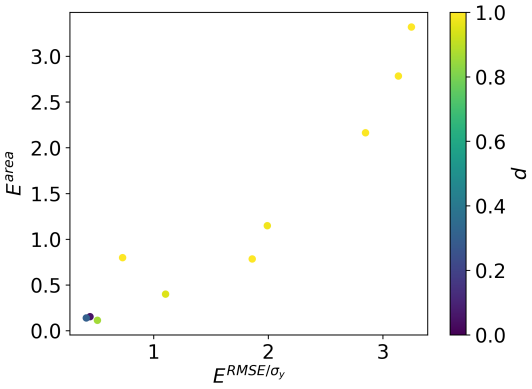
(a) Diffusion



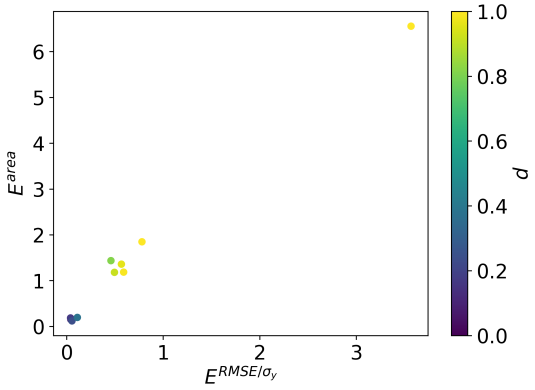
(b) Fluence



(c) Strength



(d) Superconductor



(e) Friedman

Figure 6.29: The relationship between d, E^{RMSE/σ_y}, and E^{area}. When d increases, so does E^{RMSE/σ_y} and E^{area} generally increase. Note that the binning strategy used was covered in the Materials and Methods section of the main text. Lower values of d correspond with values being closer to the X_{ITB} data. Data shown are for the RF model type.

6.4.7 Comparison of Dissimilarities from KDE and GPR measures

Gaussian Process Regression (GPR) is a probabilistic, non-parametric regression model that can provide uncertainties. These uncertainties provide a natural measure of distance from training data, with smaller uncertainties for points close to the training data. The uncertainties are like KDE and might provide a domain determination method. Furthermore, these uncertainties are very widely used to guide active learning, which relies to some extent on the ability of these uncertainties to identify regions of feature space dissimilar to the training data [156]. To determine if GPR uncertainties might provide an improvement on the KDE approach used here, we studied the effect of replacing KDE with GPR uncertainties for M^{dis}. We use the GPR implementation from scikit-learn [42]. The same bandwidth estimator used for KDE in the main text was used for any GPR kernel. However, there is no equivalent GPR kernel type to the Epanechnikov kernel used in the main text for KDE. Instead, we use the kernel combination "ConstantKernel()*Matern()+WhiteNoise()". From the experience of the authors, the previous kernel was found to work well across numerous materials data sets. The parameter n_restarts_optimizer was set to 10. For the uncertainty output of GPR, we scaled values with Eq. 6.18. $GPR(\vec{x})$ denotes the uncertainty of $\vec{x}$ from GPR and X_{ITB} is the ITB feature set. Using Eq. 6.18 has the property of preserving the order of GPR uncertainties, but sets the range of values to be in the interval $[0, 1)$. Smaller values of d correspond to smaller GPR uncertainties and vice versa.

$$d = \frac{GPR(\vec{x})}{GPR(\vec{x}) + \max\limits_{\forall \vec{a} \in X_{ITB}} (GPR(\vec{a}))} \tag{6.18}$$

We then ran the assessments of $A^{|y-\hat{y}|/MAD_y}$, A^{RMSE/σ_y}, and A^{area}. We then took the $F1_{max}$ scores from the results using KDE (i.e., the results reported in the main text) and subtracted the $F1_{max}$ scores using GPR and called the result $\Delta F1$. The scores are shown in Table 6.16. Note that when GPR is better than KDE, it is represented by a

negative value. Overall, the GPR approach performs quite well. However, the results show only one case where GPR outperforms KDE and the improvement is extremely small, multiple cases where both provide essentially perfect results, and a few cases where KDE outperforms GPR significantly (for the Diffusion and Streel Strength data sets).

Table 6.16: Precision and recall data are tabulated for use of GPR scaled uncertainties instead of our KDE measure. The d_c^t, precision, recall, and $F1$ are for $F1_{max}$.

Data	Assessment	Baseline	AUC	AUC-Baseline	d_c^t	Precision	Recall	$F1$	$\Delta F1$
Fluence	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.73	0.93	0.19	1.00	0.86	0.84	0.85	0.00
Fluence	A^{RMSE/σ_y}	0.60	1.00	0.40	0.85	1.00	1.00	1.00	0.00
Fluence	A^{area}	0.50	1.00	0.50	0.82	1.00	1.00	1.00	0.00
Diffusion	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.57	0.89	0.33	1.00	0.73	1.00	0.85	0.07
Diffusion	A^{RMSE/σ_y}	0.40	0.88	0.48	0.50	1.00	0.75	0.86	0.14
Diffusion	A^{area}	0.30	1.00	0.70	0.50	1.00	1.00	1.00	0.00
Steel Strength	$A^{\lvert y-\hat{y}\rvert/MAD_y}$	0.66	0.83	0.17	1.00	0.66	1.00	0.80	-0.02
Steel Strength	A^{RMSE/σ_y}	0.66	0.89	0.23	1.00	0.69	1.00	0.81	0.19
Steel Strength	A^{area}	0.40	1.00	0.60	0.50	1.00	1.00	1.00	0.00

A more direct comparison of KDE and GPR can be performed with confidence curves [138]. In these plots, data were included in the measure of E^{RMSE/σ_y} in the order of the corresponding measure. See Figure 6.30 as an example for the Diffusion data with an RF model type for M^{prop}. Each color represents the measure by which data are added for the calculation of E^{RMSE/σ_y} and the measure's performance can be measured as an AUC. For confidence curves, a higher value of AUC is worse than a lower value. The best possible AUC is the one acquired by absolute residuals (blue curve). As a side note, notice that in Figure 6.30 d from KDE outperforms σ_c for providing privileged knowledge for reducing absolute residuals.

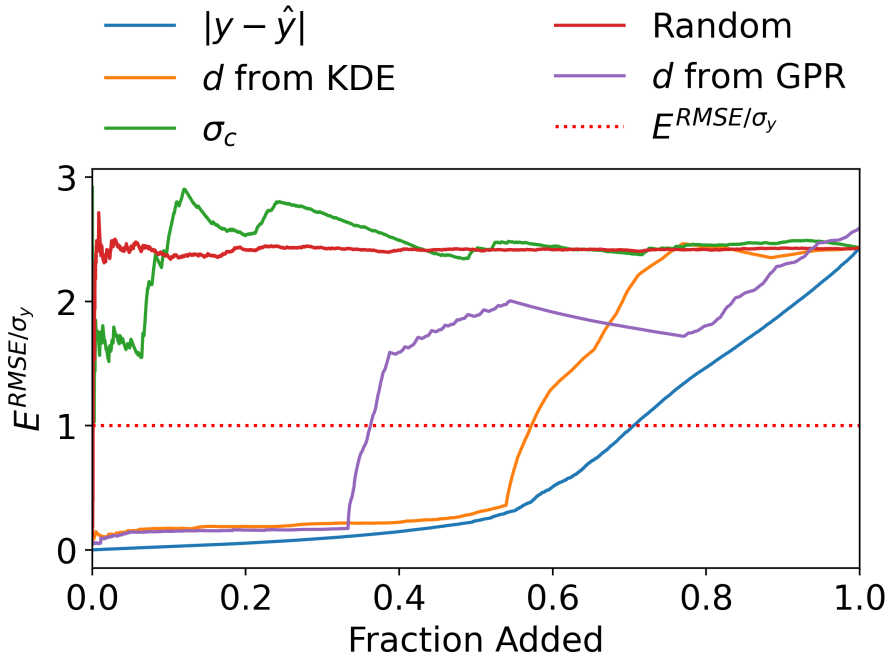


Figure 6.30: The confidence curves for the Diffusion data using the RF model type for M^{prop} are shown. The measure of d from KDE is represented by the yellow line. We included d as the purple line from another independent A^{RMSE/σ_y} using GPR for M^{dis}. Note that d from GPR does not intersect with the other curves to the right because the set of residuals was from another independent calculation and vary slightly.

We tabulated AUC scores for confidence curves in Table 6.17. With this assessment, each method performs better on one data set and both methods are essentially tied for Streel Strength. However, KDE improved over GPR for Diffusion by a much larger margin than GPR improved over KDE for the Fluence data. Given that KDE improved over GPR for more tests and by larger margins, we conclude that assessments for domain should be performed with KDE instead of GPR.

Table 6.17: The AUC scores from confidence curves are tabulated. The M^{prop} type used was RF.

Data	AUC (KDE)	AUC (GPR)
Fluence	0.63	0.58
Diffusion	1.05	1.30
Strength	0.74	0.75

KDE has two other advantages over GPR. First, KDE does not require information of

the target variable (y) unlike GPR. Second, KDE is much faster to compute compared to training a GPR model. The computation times across sets of calculations from this work are shown in Table 6.18 using the same computational resources for the workflows of $A^{|y-\hat{y}|/MAD_y}$, A^{RMSE/σ_y}, and A^{area}. The longer time required for GPR vs. KDE is expected, intrinsic to the methods, and likely to be worse for larger data sets. More specifically, the time complexity for GPR is generally $\mathcal{O}(n^3)$ with n being the size of the input to the model [157]. Methods exist to reduce the required computational time for GPR, but none (to the best of the knowledge of the authors) that are lower than $\mathcal{O}(n^2)$ [158]. In comparison to GPR, KDE is much faster. The time complexity for KDE, at worst, is $\mathcal{O}(n^2)$ [159]. Tree based methods exist to reduce the time complexity of KDE [160]. Two tree algorithms supported by the KDE implementation in scikit-learn are the Ball and k-d tree algorithms, which have time complexities of $\mathcal{O}(nlog(n))$ and $\mathcal{O}(log(n))$ (at least for a low number of dimensions for k-d tree), respectively [161]. These observations strongly suggest that KDE will be significantly, perhaps massively, faster than GPR for almost all cases. Other methods to reduce the time complexity for both KDE and GPR exist which could in theory alter this balance in some cases, but a more detailed discussion is outside the scope of this paper.

Table 6.18: The calculation time comparison between KDE and GPR are shown here. The letters m and s represent minutes and seconds, respectively.

Data	Time (KDE)	Time (GPR)
Fluence	24m50s	1905m22s
Diffusion	8m14s	213m49s
Strength	4m38s	63m8s

6.4.8 Assessment on M^{dom}

We have shown in the main text how OOB data from M^{dis} can be used to separate ID/OD. However, there is also an uncertainty with respect to the exact threshold chosen for $F1_{max}$ for the subset of data analyzed. We provide confusion matrices for $\widehat{ID}/\widehat{OD}$

from each M^{dom} built from repeated 5-fold CV at the end of this document. Each M^{dom} was built with the procedure outlined in the main text. The assessment work flow is shown in Figure 6.31. This is not proper CV since the binned data used for testing are from all OOB predictions, yet is still an assessment on the sensitivity of selecting d_c^t for $A^{|y-\hat{y}|/MAD_y}$, A^{RMSE/σ_y}, and A^{area}. Note that we still include BLOCO to create OOB data, but we do not assess M^{dom} created from those folds because of failure case (iii) outlined in the main text. The confusion matrices are supplied with other assessment plots at the end of this document.

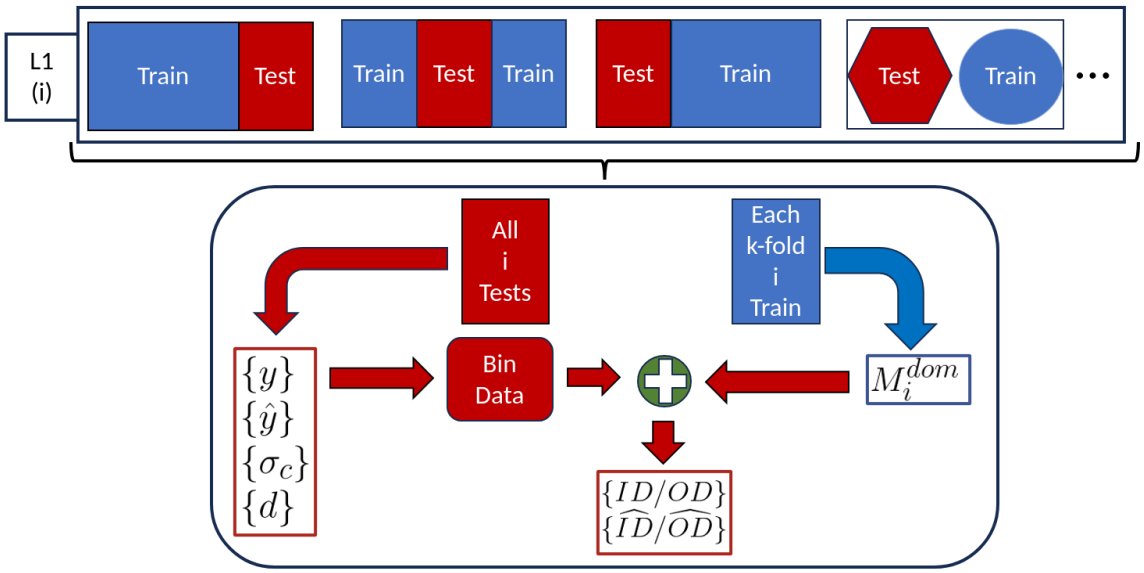


Figure 6.31: The upper level (L1) is used to produce ITB (blue Train) and OOB (red Test) data from k-fold and BLOCO splits. From each repeated k-fold L1 Train, we apply the method used to fit M^{dom} in the main text.

6.4.9 Notes on Model Parameters

All default parameters were used for base estimators for all models covered (unless explicitly stated otherwise). For the ensemble, 100 estimators were used for all model types except for BNN. BNN used 10 base estimators in ensembles because of the computational expense in training. Specifics on the BNN base estimator architecture are detailed below.

```
from keras.layers import Dense, Dropout, BatchNormalization
from scikeras.wrappers import KerasRegressor
from keras.models import Sequential
```

```
def keras_model(shape):

  n = 100
  model = Sequential()
  model.add(Dense(
          n,
          input_dim=shape,
          kernel_initializer='normal',
          activation='relu'
          ))
  model.add(Dropout(0.3))
  model.add(Dense(
          n,
          kernel_initializer='normal',
          activation='relu'
          ))
  model.add(Dropout(0.3))
  model.add(Dense(
          1,
          kernel_initializer='normal'
          ))
  model.compile(
        loss='mean_squared_error',
        optimizer='adam'
        )

  return model
```

6.4.10 Feature Learning Curves

This section contains the feature learning curves. The number of selected features are mentioned in the main text.

6.4.10.1 Diffusion

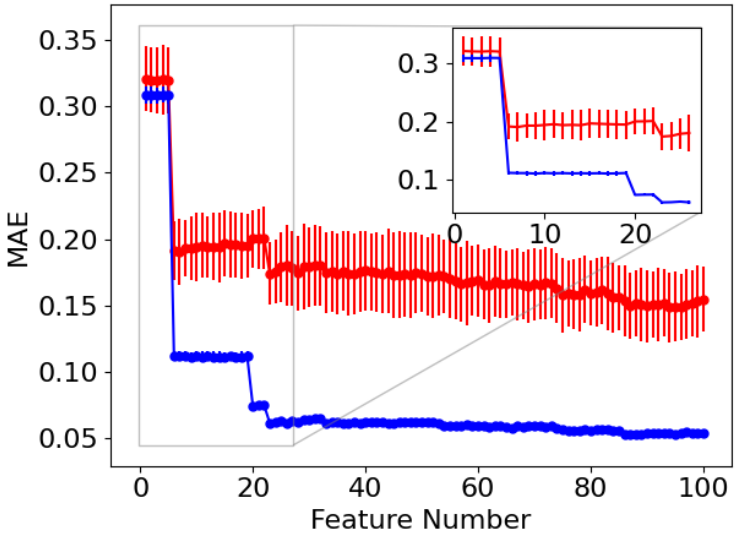


Figure 6.32: Leave out mean average error are shown by the red data (Validation) while blue denotes leave in mean average error (Train). The inset denotes an enlarged visual of a subset of data relevant for finding a feature number cutoff.

6.4.10.2 Steel Strength

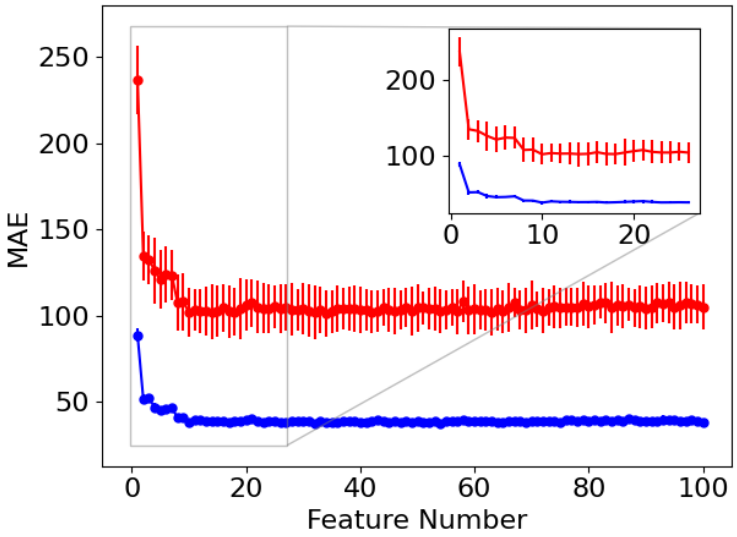


Figure 6.33: Leave out mean average error are shown by the red data (Validation) while blue denotes leave in mean average error (Train). The inset denotes an enlarged visual of a subset of data relevant for finding a feature number cutoff.

6.4.10.3 Superconductor

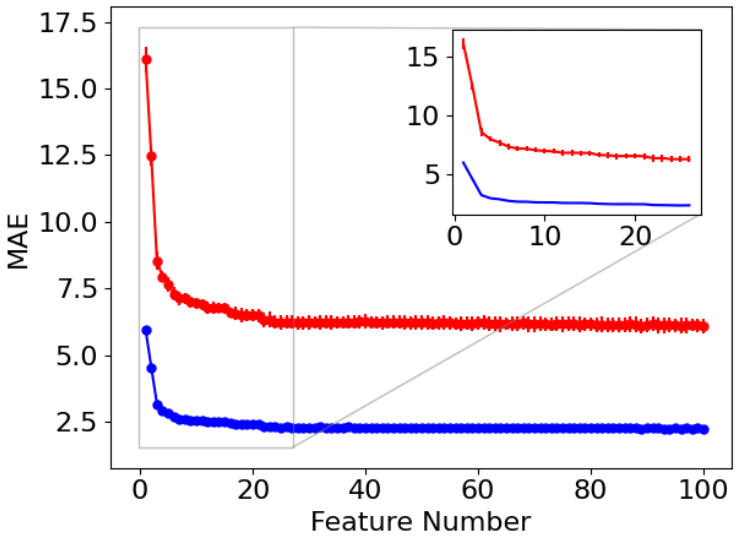


Figure 6.34: Leave out mean average error are shown by the red data (Validation) while blue denotes leave in mean average error (Train). The inset denotes an enlarged visual of a subset of data relevant for finding a feature number cutoff.

Bibliography

[1] L. E. Schultz, B. Afflerbach, I. Szlufarska, and D. Morgan, "Molecular dynamic characteristic temperatures for predicting metallic glass forming ability," *Computational Materials Science*, vol. 201, p. 110 877, 2022, ISSN: 0927-0256. DOI: https://doi.org/10.1016/j.commatsci.2021.110877. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927025621005899.

[2] L. E. Schultz, B. Afflerbach, C. Francis, P. M. Voyles, I. Szlufarska, and D. Morgan, "Exploration of characteristic temperature contributions to metallic glass forming ability," *Computational Materials Science*, vol. 196, p. 110 494, 2021, ISSN: 0927-0256. DOI: https://doi.org/10.1016/j.commatsci.2021.110494. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927025621002196.

[3] B. T. Afflerbach *et al.*, "Machine Learning Prediction of the Critical Cooling Rate for Metallic Glasses from Expanded Datasets and Elemental Features," *Chemistry of Materials*, acs.chemmater.1c03542, Mar. 2022, ISSN: 0897-4756. DOI: 10.1021/acs.chemmater.1c03542. [Online]. Available: https://pubs.acs.org/doi/10.1021/acs.chemmater.1c03542.

[4] B. T. Afflerbach, L. Schultz, J. H. Perepezko, P. M. Voyles, I. Szlufarska, and D. Morgan, "Molecular simulation-derived features for machine learning predictions of metal glass forming ability," *Computational Materials Science*, vol. 199, Nov. 2021, ISSN: 09270256. DOI: 10.1016/j.commatsci.2021.110728.

[5] J. Xi *et al.*, "Microalloying effect in ternary al-sm-x (x=ag, au, cu) metallic glasses studied by ab initio molecular dynamics," *Computational Materials Science*, vol. 185, p. 109 958, 2020, ISSN: 0927-0256. DOI: https://doi.org/10.1016/j.commatsci.2020.109958. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927025620304493.

[6] K. Schmidt *et al.*, "Foundry-ml - software and services to simplify access to machine learning datasets in materials science," *Journal of Open Source Software*, vol. 9, no. 93, p. 5467, 2024. DOI: 10.21105/joss.05467. [Online]. Available: https://doi.org/10.21105/joss.05467.

[7] L. E. Schultz, Y. Wang, R. Jacobs, and D. Morgan, "A general approach for determining applicability domain of machine learning models," 2024. arXiv: 2406.05143.

[8] V. Agrawal, S. Zhang, L. E. Schultz, and D. Morgan, "Accelerating ensemble error bar prediction with single models fits," 2024. arXiv: 2404.09896.

[9] J. Meng *et al.*, "Ultra-fast oxygen conduction in sillén oxychlorides," 2024. arXiv: 2406.07723.

[10] R. Jacobs *et al.*, *Machine learning materials properties with accurate predictions, uncertainty estimates, domain guidance, and persistent online accessibility*, 2024. arXiv: 2406.15650 [cond-mat.mtrl-sci]. [Online]. Available: https://arxiv.org/abs/2406.15650.

[11] L. E. Schultz, B. Afflerbach, and D. Morgan, "Machine learning metallic glass critical cooling rates through elemental and molecular simulation based featurization,"

[12] S. Huang *et al.*, "Composition-resolved dynamics in metallic supercooled liquids from momentum-resolved electron correlation microscopy,"

[13] "Molecular dynamic characteristic temperatures for predicting metallic glass forming ability," Materials Science & Technology, Columbus, OH, 2021.

[14] "Molecular dynamics features for predicting metallic glass critical casting thickness," Virtual Materials Research Society Spring/Fall Meeting & Exhibit, Virtual, 2020.

[15] D. S. Sholl and J. A. Steckel, *Density Functional Theory: A Practical Introduction*. John Wiley & Sons, Inc, Mar. 2009, ISBN: 9780470447710. DOI: 10.1002/9780470447710. [Online]. Available: https://doi.org/10.1002/9780470447710.

[16] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, English, 2nd ed. San Diego: Academic Press, 2002, Copyright © 2002 Elsevier Inc. All rights reserved, ISBN: 978-0-12-267351-1. DOI: 10.1016/B978-0-12-267351-1.X5000-7. [Online]. Available: https://doi.org/10.1016/B978-0-12-267351-1.X5000-7.

[17] L. Verlet, "Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules," *Phys. Rev.*, vol. 159, pp. 98–103, 1967.

[18] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, *An Introduction to Statistical Learning: with Applications in Python*, English, 2023rd. Springer, 2023, ISBN: 978-3031387463.

[19] M. Awad and R. Khanna, *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, 1st ed. Berkeley, CA: Apress, 2015, pp. XIX, 268, ISBN: 978-1-4302-5989-3. DOI: 10.1007/978-1-4302-5990-9. [Online]. Available: https://doi.org/10.1007/978-1-4302-5990-9.

[20] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, San Francisco, California, USA: ACM, 2016, pp. 785–794, ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785.

[21] J. D. Musgraves, J. Hu, and L. Calvez, Eds., *Springer Handbook of Glass*, 1st ed. West Henrietta, NY, USA: Springer International Publishing, 2019, pp. XXXVI, 1841, ISBN: 978-3-319-93728-1. DOI: 10.1007/978-3-319-93728-1.

[22] S. Sohrabi *et al.*, "Manufacturing of metallic glass components: Processes, structures and properties," *Progress in Materials Science*, vol. 144, p. 101 283, 2024, ISSN: 0079-6425. DOI: https://doi.org/10.1016/j.pmatsci.2024.101283. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079642524000525.

[23] J. Schroers, "Processing of bulk metallic glass," *Advanced Materials*, vol. 22, no. 14, pp. 1566–1597, 2010. DOI: https://doi.org/10.1002/adma.200902776. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adma.200902776. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.200902776.

[24] Z. Long, H. Wei, Y. Ding, P. Zhang, G. Xie, and A. Inoue, "A new criterion for predicting the glass-forming ability of bulk metallic glasses," *Journal of Alloys and Compounds*, vol. 475, no. 1, pp. 207–219, 2009, ISSN: 0925-8388. DOI: https://doi.org/10.1016/j.jallcom.2008.07.087. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925838808012206.

[25] W. L. Johnson, J. H. Na, and M. D. Demetriou, "Quantifying the origin of metallic glass formation," *Nature Communications*, vol. 7, no. 1, p. 10 313, Dec. 2016, ISSN: 20411723. DOI: 10.1038/ncomms10313. [Online]. Available: http://www.nature.com/articles/ncomms10313.

[26] S. Sarker *et al.*, "Discovering exceptionally hard and wear-resistant metallic glasses by combining machine-learning with high throughput experimentation," *Applied Physics Reviews*, vol. 9, no. 1, p. 011 403, Jan. 2022, ISSN: 1931-9401. DOI: 10.1063/5.0068207. eprint: https://pubs.aip.org/aip/apr/article-pdf/doi/10.1063/5.0068207/19809298/011403_1_online.pdf. [Online]. Available: https://doi.org/10.1063/5.0068207.

[27] H. F. Li and Y. F. Zheng, "Recent advances in bulk metallic glasses for biomedical applications," 2016, ISSN: 18787568. DOI: 10.1016/j.actbio.2016.03.047.

[28] M. Jafary-Zadeh, G. Praveen Kumar, P. Branicio, M. Seifi, J. Lewandowski, and F. Cui, "A Critical Review on Metallic Glasses as Structural Materials for Cardiovascular Stent Applications," *Journal of Functional Biomaterials*, vol. 9, no. 1, p. 19, 2018, ISSN: 2079-4983. DOI: 10.3390/jfb9010019. [Online]. Available: http://www.mdpi.com/2079-4983/9/1/19.

[29] G. Herzer, "Modern soft magnets: Amorphous and nanocrystalline materials," *Acta Materialia*, vol. 61, no. 3, pp. 718–734, Feb. 2013, ISSN: 1359-6454. DOI: 10.1016/J.ACTAMAT.2012.10.040. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359645412007872?via%3Dihub.

[30] D. Turnbull, "Under What Conditions Can A Glass Be Formed?" *Contemporary Physics*, vol. 10, no. 5, pp. 473–488, Sep. 1969, ISSN: 13665812. DOI: 10.1080/00107516908204405. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/00107516908204405.

[31] Z. P. Lu, H. Tan, Y. Li, and S. C. Ng, "Correlation between reduced glass transition temperature and glass forming ability of bulk metallic glasses," *Scripta Materialia*, vol. 42, no. 7, pp. 667–673, 2000, ISSN: 13596462. DOI: 10.1016/S1359-6462(99)00417-0.

[32] Z. P. Lu and C. T. Liu, "A new glass-forming ability criterion for bulk metallic glasses," *Acta Materialia*, vol. 50, no. 13, pp. 3501–3512, Aug. 2002, ISSN: 13596454. DOI: 10.1016/S1359-6454(02)00166-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359645402001660?via%7B%5C%%7D3Dihub.

[33] K. Mondal and B. S. Murty, "On the parameters to assess the glass forming ability of liquids," *Journal of Non-Crystalline Solids*, vol. 351, no. 16-17, pp. 1366–1371, 2005, ISSN: 00223093. DOI: 10.1016/j.jnoncrysol.2005.03.006.

[34] R. Deng, Z. Long, L. Peng, D. Kuang, and B. Ren, "A new mathematical expression for the relation between characteristic temperature and glass-forming ability of metallic glasses," *Journal of Non-Crystalline Solids*, vol. 533, no. January, p. 119 829, 2020, ISSN: 00223093. DOI: 10.1016/j.jnoncrysol.2019.119829. [Online]. Available: https://doi.org/10.1016/j.jnoncrysol.2019.119829.

[35] J. Xiong, T. Y. Zhang, and S. Q. Shi, "Machine learning prediction of elastic properties and glass-forming ability of bulk metallic glasses," *MRS Communications*, vol. 9, no. 2, pp. 576–585, 2019, ISSN: 21596867. DOI: 10.1557/mrc.2019.44.

[36] B. Deng and Y. Zhang, "Critical feature space for predicting the glass forming ability of metallic alloys revealed by machine learning," *Chemical Physics*, vol. 538, p. 110 898, 2020, ISSN: 03010104. DOI: 10.1016/j.chemphys.2020.110898. [Online]. Available: https://doi.org/10.1016/j.chemphys.2020.110898.

[37] J. Xiong, S. Q. Shi, and T. Y. Zhang, "A machine-learning approach to predicting and understanding the properties of amorphous metallic alloys," *Materials and Design*, vol. 187, p. 108 378, 2020, ISSN: 18734197. DOI: 10.1016/j.matdes.2019.108378. [Online]. Available: https://doi.org/10.1016/j.matdes.2019.108378.

[38] P. M. Voyles, L. E. Schultz, D. Morgan, and F. Carter, *Metallic glasses and their properties*, 2021. DOI: 10.18126/NC04-IBUT. [Online]. Available: https://petreldata.net/mdf/detail/voyles_mdf_dmref_glasses_v1.3.

[39] B. Afflerbach, C. Francis, I. Szlufarska, D. Morgan, P. M. Voyles, and L. E. Schultz, *Characteristic temperature model for metallic glass critical casting diameter*, 2021. DOI: 10.18126/SBCM-IN13. [Online]. Available: https://petreldata.net/mdf/detail/schultz_gb_model_full_fit_v1.1.

[40] D. Morgan, L. Schultz, P. Voyles, I. Szlufarska, C. Francis, and B. Afflerbach, *Exploration of Characteristic Temperature Contributions to Metallic Glass Forming Ability*, Mar. 2021. DOI: 10.6084/m9.figshare.14171255.v1. [Online]. Available: https://figshare.com/articles/dataset/Exploration_of_Characteristic_Temperature_Contributions_to_Metallic_Glass_Forming_Ability/14171255.

[41] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, pp. 2079–2107, 2010, ISSN: 15324435.

[42] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[43] R. Dai, R. Ashcraft, A. K. Gangopadhyay, and K. F. Kelton, "Predicting metallic glass formation from properties of the high temperature liquid," *Journal of Non-Crystalline Solids*, vol. 525, no. October, p. 119 673, 2019, ISSN: 00223093. DOI: 10.1016/j.jnoncrysol.2019.119673. [Online]. Available: https://doi.org/10.1016/j.jnoncrysol.2019.119673.

[44] A. Jaiswal, T. Egami, K. F. Kelton, K. S. Schweizer, and Y. Zhang, "Correlation between Fragility and the Arrhenius Crossover Phenomenon in Metallic, Molecular, and Network Liquids," 2016. DOI: 10.1103/PhysRevLett.117.205701. [Online]. Available: https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.117.205701.

[45] J. Hafner, "Theory of formation of metallic glasses. II," *Physical Review B*, vol. 28, no. 4, pp. 1734–1739, Aug. 1983, ISSN: 01631829. DOI: 10.1103/PhysRevB.28.1734. [Online]. Available: https://journals.aps.org/prb/abstract/10.1103/PhysRevB.28.1734.

[46] Q. J. Hong and A. Van De Walle, "Prediction of the material with highest known melting point from ab initio molecular dynamics calculations," *Physical Review B - Condensed Matter and Materials Physics*, vol. 92, no. 2, pp. 1–6, 2015, ISSN: 1550235X. DOI: 10.1103/PhysRevB.92.020104.

[47] R. Jinnouchi, F. Karsai, and G. Kresse, "On-the-fly machine learning force field generation: Application to melting points," *Physical Review B*, vol. 100, no. 1, 2019, ISSN: 2469-9950. DOI: 10.1103/physrevb.100.014105. arXiv: 1904.12961.

[48] L. C. R. Aliaga, L. V. Lima, G. M. Domingues, I. N. Bastos, and G. A. Evangelakis, "Experimental and molecular dynamics simulation study on the glass formation of Cu–Zr–Al alloys," *Materials Research Express*, vol. 6, no. 4, p. 045 202, Jan. 2019, ISSN: 20531591. DOI: 10.1088/2053-1591/aaf97e. [Online]. Available: https://doi.org/10.1088/2053-1591/aaf97e.

[49] D. V. Louzguine-Luzgin and A. I. Bazlov, "Crystallization of fcc and bcc liquid metals studied by molecular dynamics simulation," *Metals*, vol. 10, no. 11, pp. 1–11, 2020, ISSN: 20754701. DOI: 10.3390/met10111532.

[50] M. E. Blodgett, T. Egami, Z. Nussinov, and K. F. Kelton, "Proposal for universality in the viscosity of metallic liquids," *Scientific Reports*, vol. 5, pp. 1–8, 2015, ISSN: 20452322. DOI: 10.1038/srep13837. arXiv: 1407.7558. [Online]. Available: http://dx.doi.org/10.1038/srep13837.

[51] C. Chen *et al.*, "A novel viscosity-temperature model of glass-forming liquids by modifying the eyring viscosity equation," *Applied Sciences (Switzerland)*, vol. 10, no. 2, 2020, ISSN: 20763417. DOI: 10.3390/app10020428.

[52] A. K. Gangopadhyay *et al.*, "Correlation of the fragility of metallic liquids with the high temperature structure, volume, and cohesive energy," *The Journal of Chemical Physics*, vol. 146, no. 15, p. 154 506, Apr. 2017, ISSN: 0021-9606. DOI: 10.1063/1.4981011. [Online]. Available: http://aip.scitation.org/doi/10.1063/1.4981011.

[53] A. Gangopadhyay and K. Kelton, "Recent progress in understanding high temperature dynamical properties and fragility in metallic liquids, and their connection with atomic structure," *Journal of Materials Research*, vol. 32, pp. 2638–2657, 14 Jul. 2017, ISSN: 0884-2914. DOI: 10.1557/jmr.2017.253. [Online]. Available: https://www.cambridge.org/core/product/identifier/S0884291417002539/type/journal_article.

[54] R. Dai, A. Gangopadhyay, R. Chang, and K. Kelton, "A method to predict the glass transition temperature in metallic glasses from properties of the equilibrium liquid," *Acta Materialia*, vol. 172, pp. 1–5, Jun. 2019, ISSN: 1359-6454. DOI: 10.1016/J.ACTAMAT.2019.04.034. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359645419302368.

[55] F. Puosi, N. Jakse, and A. Pasturel, "Dynamical, structural and chemical heterogeneities in a binary metallic glass-forming liquid," *Journal of Physics Condensed Matter*, vol. 30, no. 14, Mar. 2018, ISSN: 1361648X. DOI: 10.1088/1361-648X/aab110.

[56] A. C. Angell, "Formation of Glasses from Liquids and Biopolymers," *Science*, vol. 267, no. 5206, pp. 1924–1935, 1995. DOI: 10.1126/science.267.5206.1924. [Online]. Available: papers2://publication/uuid/46ED95A9-CC48-4242-A7C5-9438976B8C42.

[57] Y. Chen, W. Zhang, and L. Yu, "Hydrogen Bonding Slows Down Surface Diffusion of Molecular Glasses," *Journal of Physical Chemistry B*, vol. 120, no. 32, pp. 8007–8015, 2016, ISSN: 15205207. DOI: 10.1021/acs.jpcb.6b05658.

[58] S. Plimpton, "Fast parallel algorithms for short-range molecular dynamics," *Journal of Computational Physics*, vol. 117, no. 1, pp. 1–19, Mar. 1995, ISSN: 00219991. DOI: 10.1006/jcph.1995.1039.

[59] C. A. Becker, F. Tavazza, Z. T. Trautt, and R. A. Buarque De Macedo, "Considerations for choosing and using force fields and interatomic potentials in materials science and engineering," *Current Opinion in Solid State and Materials Science*, vol. 17, no. 6, pp. 277–283, 2013, ISSN: 13590286. DOI: 10.1016/j.cossms.2013.10.001. [Online]. Available: http://dx.doi.org/10.1016/j.cossms.2013.10.001.

[60] Y. Q. Cheng, E. Ma, and H. W. Sheng, "Atomic level structure in multicomponent bulk metallic glass," *Phys. Rev. Lett.*, vol. 102, p. 245 501, 24 Jun. 2009. DOI: 10.1103/PhysRevLett.102.245501. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.102.245501.

[61] Y. Q. Cheng, H. W. Sheng, and E. Ma, "Relationship between structure, dynamics, and mechanical properties in metallic glass-forming alloys," *Phys. Rev. B*, vol. 78, p. 014 207, 1 Jul. 2008. DOI: 10.1103/PhysRevB.78.014207. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.78.014207.

[62] T. Fujita, P. F. Guan, H. W. Sheng, A. Inoue, T. Sakurai, and M. W. Chen, "Coupling between chemical and dynamic heterogeneities in a multicomponent bulk metallic glass," *Phys. Rev. B*, vol. 81, p. 140 204, 14 Apr. 2010. DOI: 10.1103/PhysRevB.81.140204. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.81.140204.

[63] L. M. Hale, Z. T. Trautt, and C. A. Becker, "Evaluating variability with atomistic simulations: The effect of potential and calculation methodology on the modeling of lattice and elastic constants," *Modelling and Simulation in Materials Science and Engineering*, vol. 26, no. 5, 2018, ISSN: 1361651X. DOI: 10.1088/1361-651X/aabc05.

[64] Q. J. Li, H. Sheng, and E. Ma, "Strengthening in multi-principal element alloys with local-chemical-order roughened dislocation pathways," *Nature Communications*, vol. 10, no. 1, pp. 1–11, 2019, ISSN: 20411723. DOI: 10.1038/s41467-019-11464-7. arXiv: 1904.07681. [Online]. Available: http://dx.doi.org/10.1038/s41467-019-11464-7.

[65] H. W. Sheng, M. J. Kramer, A. Cadien, T. Fujita, and M. W. Chen, "Highly optimized embedded-atom-method potentials for fourteen fcc metals," *Phys. Rev. B*, vol. 83, p. 134 118, 13 Apr. 2011. DOI: 10.1103/PhysRevB.83.134118. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.83.134118.

[66] D. Rapaport, *The Art of Molecular Dynamics Simulation*. Cambridge University Press, 2007, ISBN: 9780521825689.

[67] B. Hess, "Determining the shear viscosity of model liquids from molecular dynamics simulations," *Journal of Chemical Physics*, vol. 116, no. 1, pp. 209–217, 2002, ISSN: 00219606. DOI: 10.1063/1.1421362.

[68] R. W. Zwanzig, "TIME-CORRELATION FUNCTIONS AND TRANSPORT COEFFICIENTS IN STATISTICAL MECHANICS National Bureau of Standards, Washington," *Annu. Rev. Phys. Chem.*, vol. 16, pp. 67–102, 1964, ISSN: 0020-7136.

[69] Y. Q. Cheng and E. Ma, "Indicators of internal structural states for metallic glasses: Local order, free volume, and configurational potential energy," *Applied Physics Letters*, vol. 93, no. 5, pp. 1–4, 2008, ISSN: 00036951. DOI: 10.1063/1.2966154.

[70] H. W. Sheng, E. Ma, and M. J. Kramer, "Relating dynamic properties to atomic structure in metallic glasses," *JOM*, vol. 64, no. 7, pp. 856–881, Jul. 2012, ISSN: 1543-1851. DOI: 10.1007/s11837-012-0360-y. [Online]. Available: https://doi.org/10.1007/s11837-012-0360-y.

[71] C. F. Jekel and G. Venter, *pwlf: a python library for fitting 1d continuous piecewise linear functions*, 2019. [Online]. Available: https://github.com/cjekel/piecewise_linear_fit_py.

[72] T. Iwashita, D. M. Nicholson, and T. Egami, "Elementary Excitations and Crossover Phenomenon in Liquids," 2013. DOI: 10.1103/PhysRevLett.110.205504. [Online]. Available: https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.110.205504.

[73] D. Morgan, L. Schultz, I. Szlufarska, and B. Afflerbach, *Molecular dynamic characteristic temperatures for predicting metallic glass forming ability*, Apr. 2021. DOI: 10.6084/m9.figshare.14502135.v1. [Online]. Available: https://figshare.com/articles/dataset/Molecular_Dynamic_Characteristic_Temperatures_for_Predicting_Metallic_Glass_Forming_Ability/14502135/1.

[74] L. E. Schultz, *Molecular-Dynamic-Characteristic-Temperatures-for-Predicting-Metallic-Glass-Forming-Ability*, Apr. 2021. [Online]. Available: https://github.com/leschultz/Molecular-Dynamic-Characteristic-Temperatures-for-Predicting-Metallic-Glass-Forming-Ability.git.

[75] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996, ISSN: 00359246. [Online]. Available: http://www.jstor.org/stable/2346178.

[76] Y. L. Pavlov, "Random forests," *Random Forests*, pp. 1–122, 2019. DOI: 10.1201/9780429469275-8.

[77] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001, ISSN: 00905364. DOI: 10.1214/aos/1013203451.

[78] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, pp. 1–21, 2015, ISSN: 19326203. DOI: 10.1371/journal.pone.0118432.

[79] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and Information Systems*, vol. 41, no. 3, pp. 647–665, 2014, ISSN: 02193116. DOI: 10.1007/s10115-013-0679-x.

[80] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon *et al.*, Eds., Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[81] W. Wang, C. Dong, and C. Shek, "Bulk metallic glasses," *Materials Science and Engineering: R: Reports*, vol. 44, no. 2, pp. 45–89, 2004, ISSN: 0927-796X. DOI: https://doi.org/10.1016/j.mser.2004.03.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927796X04000300.

[82] Q. Gao and Z. Jian, "Fragility and Vogel-Fulcher-Tammann parameters near glass transition temperature," *Materials Chemistry and Physics*, vol. 252, no. May, p. 123 252, 2020, ISSN: 02540584. DOI: 10.1016/j.matchemphys.2020.123252. [Online]. Available: https://doi.org/10.1016/j.matchemphys.2020.123252.

[83] S. A. Kube *et al.*, "Compositional dependence of the fragility in metallic glass forming liquids," *Nature Communications*, vol. 13, no. 1, p. 3708, Dec. 2022, ISSN: 2041-1723. DOI: 10.1038/s41467-022-31314-3. [Online]. Available: https://www.nature.com/articles/s41467-022-31314-3.

[84] T. Long, Z. Long, and Z. Peng, "Rational design and glass-forming ability prediction of bulk metallic glasses via interpretable machine learning," *Journal of Materials Science*, vol. 58, no. 21, pp. 8833–8844, May 2023, ISSN: 15734803. DOI: 10.1007/s10853-023-08528-x. [Online]. Available: https://link.springer.com/10.1007/s10853-023-08528-x.

[85] L. Ward, S. C. O'Keeffe, J. Stevick, G. R. Jelbert, M. Aykol, and C. Wolverton, "A machine learning approach for engineering bulk metallic glass alloys," *Acta Materialia*, vol. 159, pp. 102–111, 2018, ISSN: 1359-6454. DOI: https://doi.org/10.1016/j.actamat.2018.08.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359645418306268.

[86] R. Jacobs *et al.*, "The materials simulation toolkit for machine learning (mast-ml): An automated open source toolkit to accelerate data-driven materials research," *Computational Materials Science*, vol. 176, October 2019 2020, ISSN: 09270256. DOI: 10.1016/j.commatsci.2020.109544.

[87] G. Liu *et al.*, "Machine learning versus human learning in predicting glass-forming ability of metallic glasses," *Acta Materialia*, vol. 243, Jan. 2023, ISSN: 13596454. DOI: 10.1016/j.actamat.2022.118497.

[88] J. Wang, A. Agrawal, and K. Flores, "Are hints about glass forming ability hidden in the liquid structure?" *Acta Materialia*, vol. 171, pp. 163–169, Jun. 2019, ISSN: 13596454. DOI: 10.1016/j.actamat.2019.04.001.

[89] W. P. Weeks and K. M. Flores, "Using characteristic structural motifs in metallic liquids to predict glass forming ability," *Intermetallics*, vol. 145, Jun. 2022, ISSN: 09669795. DOI: 10.1016/j.intermet.2022.107560.

[90] G. B. Bokas, L. Zhao, J. H. Perepezko, and I. Szlufarska, "On the role of sm in solidification of al-sm metallic glasses," *Scripta Materialia*, vol. 124, pp. 99–102, Nov. 2016, ISSN: 13596462. DOI: 10.1016/j.scriptamat.2016.06.045.

[91] G. B. Bokas, Y. Shen, L. Zhao, H. W. Sheng, J. H. Perepezko, and I. Szlufarska, "Synthesis of sm–al metallic glasses designed by molecular dynamics simulations," *Journal of Materials Science*, vol. 53, pp. 11 488–11 499, 16 Aug. 2018, ISSN: 15734803. DOI: 10.1007/s10853-018-2393-2.

[92] Y. Zuo *et al.*, "Performance and cost assessment of machine learning interatomic potentials," *Journal of Physical Chemistry A*, vol. 124, pp. 731–745, 4 2020, ISSN: 15205215. DOI: 10.1021/acs.jpca.9b08723.

[93] I. S. Novikov, K. Gubaev, E. V. Podryabinkin, and A. V. Shapeev, "The mlip package: Moment tensor potentials with mpi and active learning," *Machine Learning: Science and Technology*, vol. 2, p. 025 002, 2 2021, ISSN: 2632-2153. DOI: 10.1088/2632-2153/abc9fe.

[94] G. Kresse and J. Hafner, "Ab initio molecular dynamics for liquid metals," *Phys. Rev. B*, vol. 47, pp. 558–561, 1 Jan. 1993. DOI: 10.1103/PhysRevB.47.558. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.47.558.

[95] A. P. Thompson *et al.*, "LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," *Comp. Phys. Comm.*, vol. 271, p. 108 171, 2022. DOI: 10.1016/j.cpc.2021.108171.

[96] A. Stukowski, "Visualization and analysis of atomistic simulation data with ovito-the open visualization tool," *Modelling and Simulation in Materials Science and Engineering*, vol. 18, 1 2010, ISSN: 09650393. DOI: 10.1088/0965-0393/18/1/015012.

[97] M. P. Polak and D. Morgan, "Extracting accurate materials data from research papers with conversational language models and prompt engineering," *Nature Communications*, vol. 15, 1 Dec. 2024, ISSN: 20411723. DOI: 10.1038/s41467-024-45914-8.

[98] M. P. Polak *et al.*, "Flexible, model-agnostic method for materials data extraction from text using general purpose language models," 2023. arXiv: 2302.04914 [cond-mat.mtrl-sci].

[99] P. M. Voyles, L. E. Schultz, D. Morgan, C. Francis, B. Afflerbach, and A. Hakeem, *Metallic glasses and their properties*, Data set. DOI: 10.18126/7yg1-osf2. [Online]. Available: https://foundry-ml.org/#/datasets/10.18126%2F7yg1-osf2.

[100] H. Sheng. [Online]. Available: https://sites.google.com/site/eampotentials/.

[101] Q.-J. Li, H. Sheng, and E. Ma, "Strengthening in multi-principal element alloys with local-chemical-order roughened dislocation pathways," *Nature Communications*, vol. 10, no. 1, p. 3563, Aug. 2019, ISSN: 2041-1723. DOI: 10.1038/s41467-019-11464-7. [Online]. Available: https://doi.org/10.1038/s41467-019-11464-7.

[102] H. Sheng, *Pdsi potential table*, https://sites.google.com/site/eampotentials/table/pdsi?authuser=0.

[103] G. Biroli and J. P. Bouchaud, "The random first-order transition theory of glasses: A critical assessment," 2009. arXiv: 0912.2542 [cond-mat.dis-nn].

[104] J. C. Mauro, Y. Yue, A. J. Ellison, P. K. Gupta, and D. C. Allan, "Viscosity of glass-forming liquids," 2009.

[105] R. M. Reis *et al.*, "Relationship between viscous dynamics and the configurational thermal expansion coefficient of glass-forming liquids," *Journal of Non-Crystalline Solids*, vol. 358, pp. 648–651, 3 2012, ISSN: 00223093. DOI: 10.1016/j.jnoncrysol.2011.11.029. [Online]. Available: http://dx.doi.org/10.1016/j.jnoncrysol.2011.11.029.

[106] A. Jain *et al.*, "The Materials Project: A materials genome approach to accelerating materials innovation," *APL Materials*, vol. 1, no. 1, p. 11 002, 2013, ISSN: 2166532X. DOI: 10.1063/1.4812323. [Online]. Available: http://link.aip.org/link/AMPADS/v1/i1/p011002/s1%5C&Agg=doi.

[107] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, "A general-purpose machine learning framework for predicting properties of inorganic materials," *npj Computational Materials*, vol. 2, no. 1, p. 16 028, 2016, ISSN: 2057-3960. DOI: 10.1038/npjcompumats.2016.28. [Online]. Available: https://doi.org/10.1038/npjcompumats.2016.28.

[108] R. Jacobs, J. Liu, H. Abernathy, and D. Morgan, "Machine learning design of perovskite catalytic properties," *Advanced Energy Materials*, 2024, ISSN: 16146840. DOI: 10.1002/aenm.202303684.

[109] Y. Cheng and E. Ma, "Atomic-level structure and structure–property relationship in metallic glasses," *Progress in Materials Science*, vol. 56, no. 4, pp. 379–473, 2011, ISSN: 0079-6425. DOI: https://doi.org/10.1016/j.pmatsci.2010.12.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079642510000691.

[110] M. Chen, "A brief overview of bulk metallic glasses," *NPG Asia Materials*, vol. 3, no. 9, pp. 82–90, Sep. 2011, ISSN: 1884-4057. DOI: 10.1038/asiamat.2011.30. [Online]. Available: https://doi.org/10.1038/asiamat.2011.30.

[111] J. Ding, Y.-Q. Cheng, and E. Ma, "Full icosahedra dominate local order in cu64zr34 metallic glass and supercooled liquid," *Acta Materialia*, vol. 69, pp. 343–354, 2014, ISSN: 1359-6454. DOI: https://doi.org/10.1016/j.actamat.2014.02.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359645414000858.

[112] D. Morgan and R. Jacobs, "Opportunities and challenges for machine learning in materials science," *Annual Review of Materials Research*, vol. 50, no. Volume 50, 2020, pp. 71–103, 2020, ISSN: 1545-4118. DOI: https://doi.org/10.1146/annurev-matsci-070218-010015. [Online]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-matsci-070218-010015.

[113] D. Morgan and R. Jacobs, "Exploring the role of machine learning in materials science and engineering," *Open Access Government*, no. September, pp. 1–4, 2023. DOI: 10.56367/OAG-040-11110.

[114] Y. Gao, X. Li, X. V. Wang, L. Wang, and L. Gao, "A Review on Recent Advances in Vision-based Defect Recognition towards Industrial Intelligence," *Journal of Manufacturing Systems*, vol. 62, no. November 2020, pp. 753–766, 2022, ISSN: 02786125. DOI: 10.1016/j.jmsy.2021.05.008. [Online]. Available: https://doi.org/10.1016/j.jmsy.2021.05.008.

[115] R. Jacobs, "Deep learning object detection in materials science: Current state and future directions," *Computational Materials Science*, vol. 211, Aug. 2022, ISSN: 09270256. DOI: 10.1016/j.commatsci.2022.111527.

[116] J. Mavračić, C. J. Court, T. Isazawa, S. R. Elliott, and J. M. Cole, "Chemdataextractor 2.0: Autopopulated ontologies for materials science," *Journal of Chemical Information and Modeling*, vol. 61, pp. 4280–4289, 9 Sep. 2021, ISSN: 1549960X. DOI: 10.1021/acs.jcim.1c00446.

[117] C. Chen and S. P. Ong, "A universal graph deep learning interatomic potential for the periodic table," *Nature Computational Science*, vol. 2, no. 11, pp. 718–728, 2022, ISSN: 2662-8457. DOI: 10.1038/s43588-022-00349-3. [Online]. Available: https://doi.org/10.1038/s43588-022-00349-3.

[118] R. Stahlbock *et al.*, *Transactions on Computational Science and Computational Intelligence Advances in Data Science and Information Engineering*. Springer, 2021. [Online]. Available: http://www.springer.com/series/11769.

[119] A. de Mathelin, F. Deheeger, G. Richard, M. Mougeot, and N. Vayatis, "Adapt: Awesome domain adaptation python toolbox," *arXiv preprint arXiv:2107.03049*, 2021.

[120] S. Mohseni, M. Pitale, J. Yadawa, and Z. Wang, "Self-supervised learning for generalizable out-of-distribution detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 5216–5223, Apr. 2020. DOI: 10.1609/aaai.v34i04.5966. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5966.

[121] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," 2024. arXiv: 2110.11334 [cs.CV].

[122] J. Jaworska, N. Nikolova-Jeliazkova, and T. Aldenberg, "Qsar applicability domain estimation by projection of the training set in descriptor space: A review," *ATLA Alternatives to Laboratory Animals*, vol. 33, pp. 445–459, 5 2005, ISSN: 02611929. DOI: 10.1177/026119290503300508.

[123] G. Panapitiya and E. Saldanha, "Outlier-based domain of applicability identification for materials property prediction models," *ArXiv*, vol. abs/2302.06454, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:256826836.

[124] C. Sutton, M. Boley, L. M. Ghiringhelli, M. Rupp, J. Vreeken, and M. Scheffler, "Identifying domains of applicability of machine learning models for materials science," *Nature Communications*, vol. 11, pp. 1–9, 1 2020, ISSN: 20411723. DOI: 10.1038/s41467-020-17112-9.

[125] J. Caldeira and B. Nord, "Deeply uncertain: Comparing methods of uncertainty quantification in deep learning algorithms," *Machine Learning: Science and Technology*, vol. 2, p. 015 002, 1 Dec. 2020, ISSN: 2632-2153. DOI: 10.1088/2632-2153/aba6f3.

[126] J. P. Janet, C. Duan, T. Yang, A. Nandy, and H. J. Kulik, "A quantitative uncertainty metric controls error in neural network-driven chemical discovery," *Chemical Science*, vol. 10, pp. 7913–7922, 34 2019, ISSN: 20416539. DOI: 10.1039/c9sc02298h.

[127] E. M. Askenazi, E. A. Lazar, and I. Grinberg, "Identification of high-reliability regions of machine learning predictions based on materials chemistry," *Journal of Chemical Information and Modeling*, vol. 63, no. 23, pp. 7350–7362, 2023, PMID: 37983482. DOI: 10.1021/acs.jcim.3c01684. eprint: https://doi.org/10.1021/acs.jcim.3c01684. [Online]. Available: https://doi.org/10.1021/acs.jcim.3c01684.

[128] E. Askanazi and I. Grinberg, "Analysis of machine learning prediction reliability based on sampling distance evaluation with feature decorrelation," *Machine Learning: Science and Technology*, vol. 5, no. 2, p. 025 030, May 2024. DOI: 10.1088/2632-2153/ad4231. [Online]. Available: https://dx.doi.org/10.1088/2632-2153/ad4231.

[129] K. Li *et al.*, "Probing out-of-distribution generalization in machine learning for materials," 2024. arXiv: 2406.06489.

[130] F. Chollet *et al.*, *Keras*, https://keras.io, 2015.

[131] T. Sainburg, L. McInnes, and T. Q. Gentner, "Parametric umap embeddings for representation and semisupervised learning," *Neural Computation*, vol. 33, no. 11, pp. 2881–2907, 2021.

[132] M. Abdar *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021, ISSN: 15662535. DOI: 10.1016/j.inffus.2021.05.008.

[133] G. Scalia, C. A. Grambow, B. Pernici, Y. P. Li, and W. H. Green, "Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction," *Journal of Chemical Information and Modeling*, vol. 60, pp. 2697–2717, 6 Jun. 2020, ISSN: 1549960X. DOI: 10.1021/acs.jcim.9b00975.

[134] K. Tran, W. Neiswanger, J. Yoon, Q. Zhang, E. Xing, and Z. W. Ulissi, "Methods for comparing uncertainty quantifications for material property predictions," *arXiv*, 2019, ISSN: 23318422. DOI: 10.1088/2632-2153/ab7e1a.

[135] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd International Conference on Machine Learning*, ser. ICML '05, Bonn, Germany: Association for Computing Machinery, 2005, pp. 625–632, ISBN: 1595931805. DOI: 10.1145/1102351.1102430. [Online]. Available: https://doi.org/10.1145/1102351.1102430.

[136] M. Kull, T. M. S. Filho, and P. Flach, "Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration," *Electronic Journal of Statistics*, vol. 11, pp. 5052–5080, 2 2017, ISSN: 19357524. DOI: 10.1214/17-EJS1338SI.

[137] G. Palmer *et al.*, "Calibration after bootstrap for accurate uncertainty quantification in regression models," *npj Computational Materials*, vol. 8, p. 115, 1 Dec. 2022, ISSN: 2057-3960. DOI: 10.1038/s41524-022-00794-8. [Online]. Available: https://www.nature.com/articles/s41524-022-00794-8.

[138] P. Pernot, "The long road to calibrated prediction uncertainty in computational chemistry," *The Journal of Chemical Physics*, vol. 156, no. 11, p. 114 109, Mar. 2022, ISSN: 0021-9606. DOI: 10.1063/5.0084302. eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/5.0084302/16539431/114109_1_online.pdf. [Online]. Available: https://doi.org/10.1063/5.0084302.

[139] J. Ling, M. Hutchinson, E. Antono, S. Paradiso, and B. Meredig, "High-dimensional materials and process optimization using data-driven experimental design with well-calibrated uncertainty estimates," *Integrating Materials and Manufacturing Innovation*, vol. 6, no. 3, pp. 207–217, 2017, ISSN: 2193-9772. DOI: 10.1007/s40192-017-0098-z. [Online]. Available: https://doi.org/10.1007/s40192-017-0098-z.

[140] C. J. Gruich, V. Madhavan, Y. Wang, and B. R. Goldsmith, "Clarifying trust of materials property predictions using neural networks with distribution-specific uncertainty quantification," *Machine Learning: Science and Technology*, vol. 4, no. 2, p. 025 019, May 2023. DOI: 10.1088/2632-2153/accace. [Online]. Available: https://dx.doi.org/10.1088/2632-2153/accace.

[141] S. Węglarczyk, "Kernel density estimation and its application," *ITM Web Conf.*, vol. 23, 2018. DOI: 10.1051/itmconf/20182300037. [Online]. Available: https://doi.org/10.1051/itmconf/20182300037.

[142] E. A. Pogue *et al.*, "Closed-loop superconducting materials discovery," *npj Computational Materials*, vol. 9, no. 1, p. 181, 2023, ISSN: 2057-3960. DOI: 10.1038/s41524-023-01131-3. [Online]. Available: https://doi.org/10.1038/s41524-023-01131-3.

[143] B. Meredig *et al.*, "Can machine learning identify the next high-temperature superconductor? examining extrapolation performance for materials discovery," *Molecular Systems Design and Engineering*, vol. 3, pp. 819–825, 5 Oct. 2018, ISSN: 20589689. DOI: 10.1039/c8me00012c.

[144] H. J. Lu, N. Zou, R. Jacobs, B. Afflerbach, X. G. Lu, and D. Morgan, "Error assessment and optimal cross-validation approaches in machine learning applied to impurity diffusion," *Computational Materials Science*, vol. 169, Nov. 2019, ISSN: 09270256. DOI: 10.1016/j.commatsci.2019.06.010.

[145] H. Wu *et al.*, "Robust fcc solute diffusion predictions from ab-initio machine learning methods," *Computational Materials Science*, vol. 134, pp. 160–165, Jun. 2017, ISSN: 09270256. DOI: 10.1016/j.commatsci.2017.03.052.

[146] *Adjunct for e900-15 technical basis for the equation used to predict radiation-induced transition temperature shift in reactor vessel materials*, Jan. 2015. [Online]. Available: https://www.astm.org/adje090015-ea.html.

[147] R. Jacobs, T. Yamamoto, G. R. Odette, and D. Morgan, "Predictions and uncertainty estimates of reactor pressure vessel steel embrittlement using machine learning," *Materials and Design*, vol. 236, Dec. 2023, ISSN: 18734197. DOI: 10.1016/j.matdes.2023.112491.

[148] *Properties of some metals and alloys*, Nickel Institute, 1982. [Online]. Available: https://nickelinstitute.org/media/1771/propertiesofsomemetalsandalloys_297_.pdf.

[149] S. Bajaj, *Citrination dataset 153092*, 2017. [Online]. Available: https://citrination.com/datasets/153092/show_files/.

[150] V. Stanev *et al.*, "Machine learning modeling of superconducting critical temperature," *npj Computational Materials*, vol. 4, 1 Dec. 2018, ISSN: 20573960. DOI: 10.1038/s41524-018-0085-8.

[151] J. H. Friedman, "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1–67, 1991. DOI: 10.1214/aos/1176347963. [Online]. Available: https://doi.org/10.1214/aos/1176347963.

[152] T. Nagler and C. Czado, "Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas," *Journal of Multivariate Analysis*, vol. 151, pp. 69–89, 2016, ISSN: 0047-259X. DOI: https://doi.org/10.1016/j.jmva.2016.07.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0047259X16300471.

[153] *Definitions, properties, and examples of correlation functions*, Chemistry LibreTexts, Dec. 2020. [Online]. Available: https://chem.libretexts.org/@go/page/107272.

[154] *Correlation functions*, GROMACS, 2019. [Online]. Available: https://manual.gromacs.org/documentation/2019-rc1/reference-manual/analysis/correlation-function.html.

[155] M. I. Mendelev, Y. Sun, F. Zhang, C. Z. Wang, and K. M. Ho, "Development of a semi-empirical potential suitable for molecular dynamics simulation of vitrification in Cu-Zr alloys," *The Journal of Chemical Physics*, vol. 151, no. 21, p. 214502, Dec. 2019, ISSN: 0021-9606. DOI: 10.1063/1.5131500. eprint: https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.5131500/13362744/214502_1_online.pdf. [Online]. Available: https://doi.org/10.1063/1.5131500.

[156] J. Chang, J. Kim, B.-T. Zhang, M. A. Pitt, and J. I. Myung, "Data-driven experimental design and model development using gaussian process with active learning," *Cognitive Psychology*, vol. 125, p. 101360, 2021, ISSN: 0010-0285. DOI: https://doi.org/10.1016/j.cogpsych.2020.101360. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S001002852030089X.

[157] S. Das, S. Roy, and R. Sambasivan, "Fast gaussian process regression for big data," *Big Data Research*, vol. 14, pp. 12–26, 2018, ISSN: 2214-5796. DOI: https://doi.org/10.1016/j.bdr.2018.06.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214579617301909.

[158] J. Kocijan, *Modelling and Control of Dynamic Systems Using Gaussian Process Models* (Advances in Industrial Control), 1st ed. Springer Cham, 2016, pp. XVI, 267, ISBN: 978-3-319-21020-9. DOI: 10.1007/978-3-319-21021-6.

[159] A. Gramacki, *Nonparametric Kernel Density Estimation and Its Computational Aspects* (Studies in Big Data), 1st ed. Springer Cham, 2018, pp. XXIX, 176, ISBN: 978-3-319-71687-9. DOI: 10.1007/978-3-319-71688-6.

[160] A. G. Gray and A. W. Moore, "Nonparametric density estimation: Toward computational tractability," in *Proceedings of the 2003 SIAM International Conference on Data Mining (SDM)*, 2003, pp. 203–211. DOI: 10.1137/1.9781611972733.19. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611972733.19. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611972733.19.

[161] A. M. Kibriya and E. Frank, "An empirical comparison of exact nearest neighbour algorithms," in *Knowledge Discovery in Databases: PKDD 2007*, J. N. Kok, J. Koronacki, R. Lopez de Mantaras, S. Matwin, D. Mladenič, and A. Skowron, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 140–151, ISBN: 978-3-540-74976-9.