MatFold: systematic insights into materials discovery models' performance through standardized cross-validation protocols

Matthew D. Witman[1,*] and Peter Schindler[2,*]

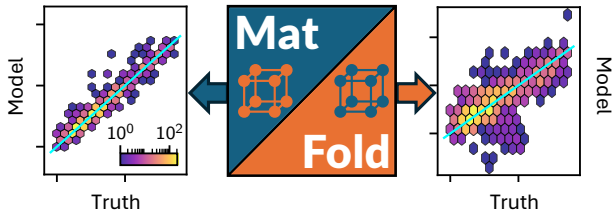
[1]Sandia National Laboratories, Livermore, California 94551, United States

[2]Northeastern University, Boston, Massachusetts 02115, United States

[*]*mwitman@sandia.gov*; *p.schindler@northeastern.edu*

Abstract

Machine learning (ML) models in the materials sciences that are validated by overly simplistic cross-validation (CV) protocols can yield biased performance estimates for downstream modeling or materials screening tasks. This can be particularly counterproductive for applications where the time and cost of failed validation efforts (experimental synthesis, characterization, and testing) are consequential. We propose a set of standardized and increasingly difficult splitting protocols for chemically and structurally motivated CV that can be followed to validate any ML model for materials discovery. Among several benefits, this enables systematic insights into model generalizability, improvability, and uncertainty, provides benchmarks for fair comparison between competing models with access to differing quantities of data, and systematically reduces possible data leakage through increasingly strict splitting protocols. A general-purpose, model-agnostic toolkit, MatFold, is provided to automate the construction of these CV splits and encourage further community use.



Introduction

Understanding and quantifying the generalizability, improvability, and uncertainty of machine learning (ML)-based materials discovery models is critical, especially in applications where downstream experimental validation (synthesis, characterization, and testing) is often time- and cost-intensive. Careful, and sometimes extensive, cross-validation (CV) is required to both avoid erroneous conclusions regarding a model's capabilities and to fully understand its limitations.[1] Withholding randomly selected test data is often insufficient for quantifying a model's performance as this sub-set is drawn from the same distribution that potentially suffers from data leakage. This in-distribution (ID) generalization error is typically minimized during model training and hyperparameter tuning to avoid over/underfitting. However, the out-of-distribution (OOD) generalization error constitutes a more useful performance metric for assessing a model's true ability to generalize to unseen data. This error originates from either lack of knowledge (*e.g.*, imbalance in data, or poor data representation) or sub-optimal model architecture and is referred to as being *epistemic*.[2] Evaluating OOD generalization, however, requires more careful considerations during data splitting.

One approach to constructing OOD test sets is to utilize unsupervised clustering with a chosen materials featurization and then conduct leave-one-cluster-out CV (LOCO-CV). For example, on compositional models for superconducting transition temperatures, LOCO-CV revealed how generalizability and expected accuracy are drastically overestimated due to data leakage in random train/test splits.[3] Omee et al. have investigated the performance of OOD prediction tasks on MatBench datasets (refractive index, shear modulus, and formation energy) utilizing structure-based graph neural network (GNN) models and LOCO-CV (k-means clustering and t-distributed stochastic neighbor embedding).[4] Hu et al. similarly have utilized LOCO-CV to study the benefit of various domain adaptation algorithms for materials property predictions (experimental band gaps and bulk metallic glass formation ability).[5] Further examples of studying

Supplementary Information: the ΔH_V dataset is provided in the supplementary_files_defects.zip. Additional CV analysis showing inference performance for additional hold-out strategies. MAE heatmaps and parity plots for leave-one-element-out splits.

generalization error based on featurization and clustering algorithms include kernel density estimate[6] and uniform manifold approximation and projection.[7]

Another approach to assess OOD generalization is based on ensembling ML models to obtain a set of predictions. The averaged predictions can exhibit more robust OOD generalization behavior and the standard deviation yields an uncertainty metric. Ensembling can be effectively applied to any bagged regressor ML model[6,8] and has also been implemented for GNNs and other deep neural networks.[2,9] Other recent work includes fitting a single model to estimate ensemble error bars by leveraging synthetic data augmentation,[10] mitigating data bias arising from uneven coverage of materials families by entropy-targeted active learning,[11] and a study on OOD generalization of formation energy models with structural and chemical hold-outs.[12]

To further encourage standardized reporting of these types of detailed insights into generalization performance and limitations of ML-based models in the materials sciences, here we provide "MatFold" as a *model-agnostic* programmatic tool for automatically generating CV splits for arbitrary materials datasets and model architectures, such as structure-based[13] or composition-based[14] models. Specifically, we propose a standardized series of CV splits based on increasingly difficult chemical/structural hold-out criteria, dataset size reduction, nested vs. non-nested splits, and others. By assessing model performance across various combinations of MatFold splitting criteria, one could, for example, more fairly compare the performance of differing approaches with the same modeling objectives. This approach allows for a better understanding of how well models' predictions generalize with increasingly difficult chemical or structural hold-out criteria. Additionally, it can determine the expected model improvement with continued data acquisition and assess whether this improvement depends on the splitting criteria used for OOD generalization. Furthermore, the method evaluates whether nested CV ensembles enhance OOD predictions and quantifies the extent of this improvement. It also examines the reliability of uncertainty estimates derived from nested CV ensembles and whether this reliability varies based on the splitting criteria used for assessing generalization.

For practically demonstrating the utility of insights derived from MatFold, we select ML exemplars from our previous work (modeling vacancy formation energies[15] and surface work functions[16]). These are examples in structure-based ML where data leakage can be very problematic since multiple training examples are derived from the same base crystal structure. For example, many structures may contain vacancy sites that are determined to be unique but are in fact nearly identical because they are only slightly above the symmetry tolerance. Similarly, Miller surfaces from the same base crystal structure may be extremely similar. In either exemplar, the expected model error for inference (*i.e.*, materials screening) can vary by factors of 2-3, depending on the splitting criteria. Through detailed insights into expected model performance in these exemplars and how it compares/differs across various splitting criteria, dataset sizes, and the exemplars themselves, we motivate MatFold as an easy-to-use and open-source tool for the materials ML community to deliver greater insights into model generalizability, improvability, and uncertainty.

MatFold procedure

MatFold serves as a convenient and automated tool to process a user's materials data and systematically generate increasingly difficult CV splits to test a modeling approach's generalizability (Figure 1). MatFold offers two split methods, $S = \{K\text{-fold or nested } (K, L)\text{-fold}\}$, where K and L are integers chosen by the user. If that value is chosen to be equal to the number of unique split labels then the created folds are leave-one-out (LOO). Outer K-folds can be split on a variety of criteria, C_K ={Random, Structure, Composition, Chemical system=Chemsys, Element, Periodic table (PT) group, PT row, Space group number=SG#, Point group, or Crystal system}, while inner L-folds can be split either randomly or utilizing the same split criteria as the outer splits (C_L ={Random, C_K}). We note that for datasets where each target label corresponds to a unique bulk crystal structure (*e.g.*, Materials Project ID, mpid) the splitting strategies "Random" and "Structure" coincide (which is not the case for the two datasets considered in this work). As shown in Table 1, MatFold provides functionality to artificially reduce the dataset size by a fractional amount D. Furthermore, materials with a specified number of unique chemical elements can be assigned to the training set by default thereby exempting them from the split criteria. This could be, for example, whether the automatic assignment of all binary compounds to the training data is performed, T ={None or Binary} (the motivation for which is discussed in the next section).

Options	Abbr.	Possibilities
Data Fraction	D	$\mathbb{R} \in (0, 1]$
Default Train Assignment	T	{None, Elemental, Binary, Ternary, ...}
Split Method	S	{K-fold, (K, L)-fold} $K, L \in \mathbb{N}^+$ (fixed or LOO)
Criteria (outer)	C_K	{Random, Structure, Composition, Chemsys, Element, PT Group, PT Row, Space Group, Point Group, Crystal System}
Criteria (inner)	C_L	{Random, C_K }

Table 1: Description of available options and criteria for creating splits with MatFold. PT and LOO stand for periodic table and leave-one-out, respectively.

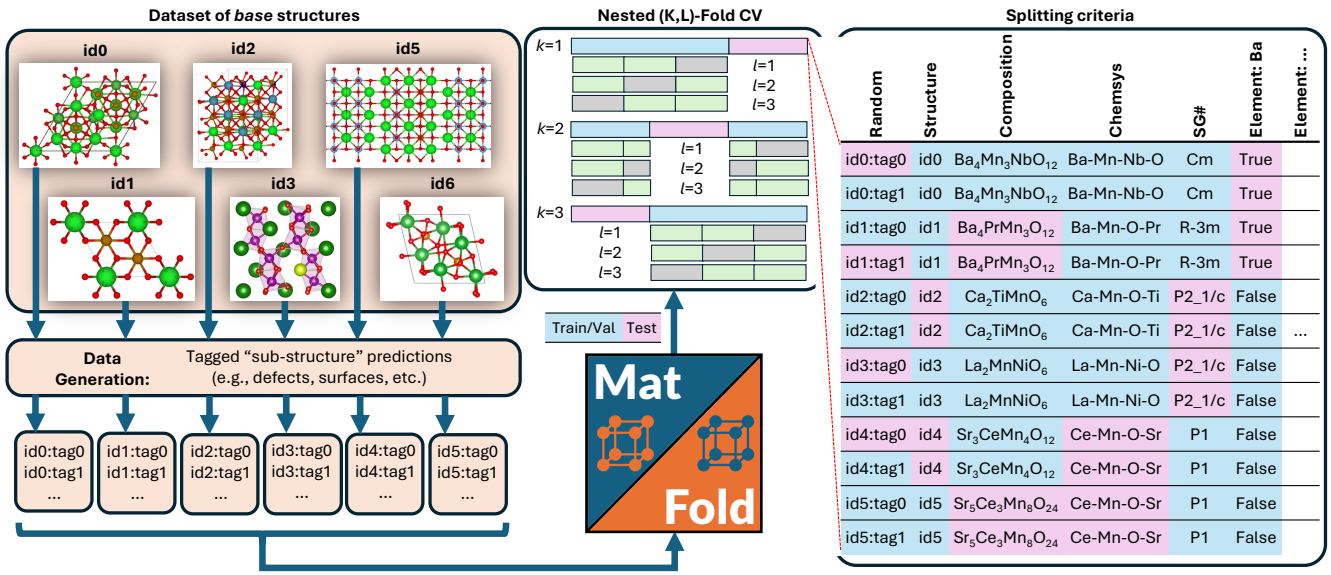


Figure 1: MatFold processes a set of base crystal structures, each of which may have multiple unique target values per structure (i.e., defect formation energies for unique symmetry sites, work functions for unique Miller surfaces, etc.). Nested (K, L)-fold CV train/test splits are automatically generated according to a variety of splitting criteria.

Based on the user's choices of D, T, K, L, and C_K, MatFold can typically create thousands of splits. The feasibility of training this many models may depend on the dataset size and modeling approach and may be less feasible, for example, in the training of recently developed universal ML potentials.[17–20] However, dataset and model sizes are often small enough for more specialized ML-based materials discovery models to perform splits across at least some subset of the criteria summarized in Table 1. Subsequent exemplars based on our previous work (modeling vacancy defect formation energies[15] and surface work functions[16]), we are able to train thousands of model splits generated by MatFold to obtain improved insights into our model's generalizability and limitations. An overview of the two datasets and the chosen MatFold split protocols for each are listed in Table 2. Model hyperparameters are fixed at the optimal conditions as determined in the respective previous work.[15,16]

To evaluate the model performances, we denote the mean absolute error of an outer test set MAE $= 1/N_k \sum_i |\hat{p}_i - p_i|$, where N_k is the number of samples in outer fold k, $\hat{p}_i$ is the model prediction of sample i, and p_i the truth value. The expected model performance is given as the ensemble average over the set of all K folds, $\langle\{\text{MAE}\}_K\rangle$. For non-nested CV, *i.e.*, K-fold, $\hat{p}_i$ in the k^{th} test set is predicted by a single model trained on the k^{th} train set. For nested CV, *i.e.*, (K, L)-fold, the final prediction is the ensemble average over the set of inner model predictions on the outer test set, $\langle\{\hat{p}_i\}_L\rangle$. The deviation of that ensemble average prediction from the true value is referred to as *residuals*, calculated as $|p_i - \langle\{\hat{p}_i\}_L\rangle|$. Importantly, nested CV also yields an uncertainty metric via the standard deviation over the set of inner model predictions on the outer test set, $\sigma(\{\hat{p}_i\}_L)$.

We note that for datasets with strong imbalances in splitting labels (*e.g.*, an element present in almost the entire dataset vs. another element being present only in a tiny fraction of the dataset) the MAE and its standard

	ΔH_V	Work Function
# data points	1,670	58,332
# of unique:		
Structures	250	3,716
Compositions	230	3,623
Chemsys	114	2,832
Space Groups	35	62
Elements	18	77
⟨target⟩ [eV]	5.8	3.92
σ(target) [eV]	3.5	0.86
Model type	dGNN	RF
# model param.	1721	15
D	{0.1, 0.5, 1.0}	{0.05, 0.1, 0.5, 1.0}
T	{None, Binary}	None
S	$\{K, (K, L)\}$	$\{K, (K, L)\}$
C_K		
Random	$K = 10, L = 10$	$K = 10, L = 10$
Structure	$K = 10, L = 10$	$K = 10, L = 10$
Composition	$K = 10, L = 10$	$K = 10, L = 10$
Chemsys	$K = 10, L = 10$	$K = 10, L = 10$
Elements	$K = \text{LOO}, L = 10$	$K = \text{LOO}, L = \text{LOO}$
PT Group	—	$K = \text{LOO}, L = \text{LOO}$
Space Group	$K = 10, L = 10$	$K = 10, L = 10$
Point Group	—	$K = \text{LOO}, L = \text{LOO}$
Crystal Sys	—	$K = \text{LOO}, L = \text{LOO}$
C_L	Random	C_K
# total splits	2,700	3,271

Table 2: Overview of both datasets considered in this work and description of the utilized splitting strategies implemented with MatFold for each.

deviation may be affected by the random seed during split generation. This can be mitigated in MatFold by specifying a minimum and maximum threshold of split label prevalence that determines whether that label is considered during the CV procedure or is always enforced to be in the training set. For example, if oxygen is present in 90% of structures in the dataset and the user specifies a maximum threshold of 0.9, then oxygen-containing structures will be part of the training set by default during CV.

Vacancy formation energy exemplar

Recently we developed a defect GNN (dGNN) modeling approach to directly predict relaxed vacancy formation energies, ΔH_V, from their respective bulk crystal structures.[15] The accompanying open-source dataset[21] specifically computes neutral cation and oxygen vacancies in ∼200 compounds, to which we added the neutral oxygen vacancy formation energies for ∼50 more structures from the work by Wexler et al[22] in this study. Now, we use MatFold to generate ∼2,700 possible splits and train/test our model performance, as summarized in Figure 2 and Figure 4, to better understand the modeling approach's generalizability, improvability, and uncertainty.

Figure 2(a) shows density parity plots of all outer test set predictions for $C_K = \{$Random, Structure, Composition, Chemsys, SG#, Elements$\}$ and $D = 1.0$, $T =$ None, and $S = K$-fold, while Figure 2(b) shows the same but for $T =$ Binary and $S = (K, L)$-fold. The color code is on a logarithmic scale with respect to the number of predictions at that grid point. Note that for this dataset, we are able to compute all ΔH_V for at least one of each binary oxide in the chemical space of interest, motivating the investigation of automatically assigning binaries to the training data. Immediately noticeable in Figure 2(b) is the mitigation of over-fitting and substantial error reduction for some outliers observed in Figure 2(a). Additional analysis in the Supplementary Information, applicable only to this exemplar, investigates the $C_K =$ Elements parity plots at a more granular level and further reveals insights into the generalization capabilities of dGNN.

Figure 2(c) further quantifies the dependence of the expected model error as a function of T, S, and C_K. The expected MAE generally increases with Random < Structure < Composition < Chemsys < SG# << Elements, where error bars correspond to $\sigma(\{\text{MAE}\}_K)$. Several key conclusions arise. For this particular dataset and model, using a single training model for inference (blue bars) generally produces an expected MAE ∼10-20% higher than using the ensemble of models from nested CV (green bars) across all C_K. From a different perspective, one would *overestimate* the expected MAEs by ∼10-20% if using the ensemble of non-nested K-fold models to perform inference for materials screening exercises, compared to the MAEs calculated by nested (K, L)-folds.

More importantly, the choice of C_K has a very strong influence on the expected MAE. The goal of this and many other specialty ML models for materials discovery, trained on small- to medium-sized datasets (∼100s-1000s of examples), is to screen properties of structures that represent entirely new compositions, or even chemical systems, that are outside the training data. For this use case, performing a purely random split introduces substantial data leakage which leads to a ∼ 30% underestimation of the expected MAE when, for example, predicting defects in a structure that represents an unseen chemical system in the training data. As an even more extreme example, $C_K =$ Elements reveals a ∼2.5 times higher expected MAE than a purely random split, although ensembling can reduce expected MAE by ∼30% relative to a non-ensemble prediction.

Figure 2(d) confirms that the standard deviation of predictions over model ensembles is a useful uncertainty metric[10,23] in this modeling application, but with some limitations. The individual residuals for any given test prediction (blue circles) are only very weakly correlated with $\sigma(\{\text{MAE}\}_K)$. However, computing the average and standard deviation of residuals within a given bin of $\sigma(\{\text{MAE}\}_K)$ (red markers and error bars, respectively) collapses the data onto the $y = x$ parity line (cyan). Therefore, *on average* a low $\sigma(\{\text{MAE}\}_K)$ is correlated with a low residual, but there is a non-negligible probability of individual predictions with very large residuals despite low uncertainty.

The final key insight from the MatFold analysis stems from the dependence of expected MAE on *both* C_K and D. Figure 4 plots expected MAE for $D = \{0.1, 0.5, 1.0\}$, expressed on the x-axis in units of number of defect examples in the training data. Data leakage and underestimation of expected MAE are even more pronounced for the smallest dataset, and the rapid plateauing of the expected MAE with increasing data is potentially indicative of the absolute accuracy limit of the model. For more realistic screening criteria, *i.e.*, $C_K = \{$Composition, Structure, Chemsys$\}$, large accuracy gains are and will continue to be obtained with increasing data collection. Interestingly (and perhaps intuitively), for $C_K =$ Chemsys the improvement qualitatively appears to be saturating before the other criteria, but will only be confirmed with additional data collection. Finally, $C_K =$ Elements reveals that additional data collection does not increase the accuracy during inference on compounds containing unseen elements. In fact, the error slightly increased with additional data collection because it may have introduced compounds with new test set elements which are even more difficult to extrapolate to from the elements contained in the train set (see Supplementary Information for more details).

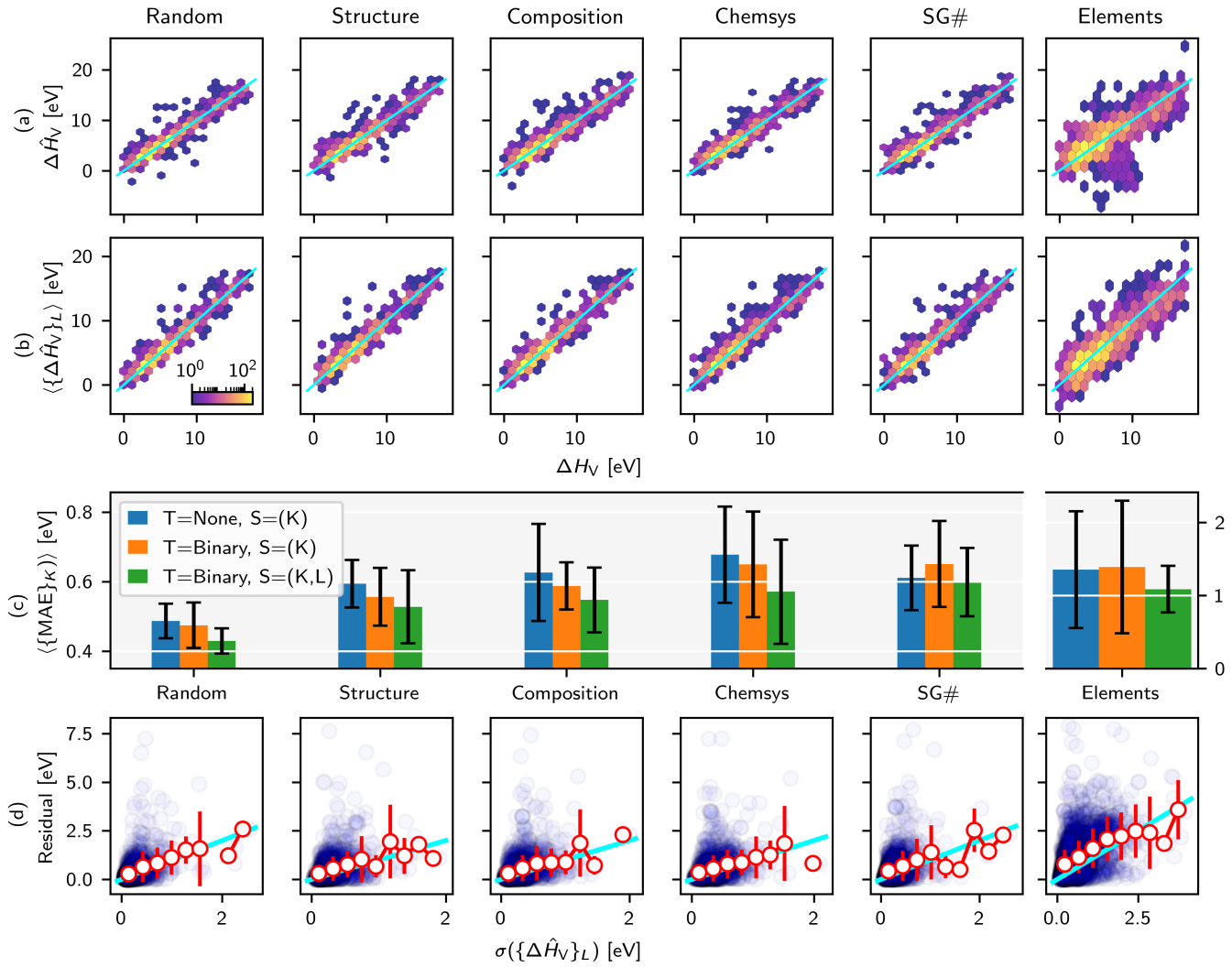


Figure 2: For ΔH_V models, we show: (a) Test set predictions from non-nested K-fold CV for various split criteria. (b) Test set predictions from nested (K, L)-fold CV for various split criteria. (c) Expected MAE for various split criteria and combinations of other MatFold options including binary hold-out or nested CV. (d) Residual vs. standard deviation of individual inner model predictions (purple circles). Here, 9 bins are created for the standard deviation, and the average and standard deviation of residuals in that bin are shown with white circles and red error bars, respectively. The cyan line represents $y = x$.

Surface work function exemplar

To gain insights into generalization error trends for a different type of dataset and ML model, we utilize MatFold on our dataset of 58,332 work functions, ϕ, of surfaces (generated from 3,716 bulk crystals that have a zero band gap) calculated by density functional theory (DFT).[16] On average, each unique bulk crystal structure has $\sim$ 15 derived surfaces. The dataset contains work functions of elemental (261), binary (14,623), and ternary (43,448) crystalline surfaces. The ML model trained on this dataset was based on a random forest (RF) model and a physics-motivated custom featurization of the topmost three atomic surface layers considering their electron affinities, atomic radii, ionization energy, Mendeleev number as well as structural information in the form of area packing fraction and interlayer spacing (details explained in our previous work[16]). The final RF model trained with 15 features has a test-MAE of 0.09 eV utilizing a random 90/10 split and 5-fold CV for hyperparameter optimization. This MAE is about 4–5 times better than the best benchmarking model and more than six times better than the random baseline. The model enabled the discovery of surfaces with extreme work functions for thermionic energy conversion[24] and high-brightness photocathodes.[25,26] Studying this dataset with MatFold is especially interesting as it significantly differs from the defect dataset in size, classes of materials, and model architecture.

We utilize similar split possibilities as for the defect dataset (see Table 2), except we do not automatically assign binary compounds to the training set (*i.e.*, here we use only T = None) and an additional data fraction ($D = 0.05$) which leads to a total of 3,271 unique splits. As discussed in the previous section and Figure 2 for the defect dataset, Figure 3(a) and (b) show density parity plots of all outer test set ($D = 1.0$) predictions for C_K = {Random, Structure, Composition, Chemsys, SG#, Elements} for non-nested $S = K$-fold and nested $S = (K, L)$-fold, respectively.

Figure 3(c) summarizes the MAEs for the parity plots displayed in (a) and (b). Unlike the defect dataset, the MAEs and their standard deviations for the work function dataset are very similar between the non-nested and nested splitting strategies. This likely stems from the GNN-based model being more prone to overfitting compared to the 15-feature RF model. Hence, the GNN model benefits more from statistical averaging during nested splitting. Like the defect dataset, the MAEs increase in the order Random < Structure < Composition < Chemsys < Elements. However, an interesting difference is that the SG# split exhibits the highest MAE, less than the MAE for the Elements split (219 and 149 meV, respectively for nested splits). Compared to the MAE of the random split this is an increase of 133% and 59%, respectively. This agrees well with the RF model features being largely comprised of elemental properties (*e.g.*, electron affinity) while containing little structural information. The work function model generalizes better outside the elemental training distribution and worse outside the structural training distribution. Among all splits that leave one element out, the MAEs are significantly larger for holding out F, H, O, or Cl (1178, 959, 708, 657 meV, respectively; *cf.* periodic table heat map in Supplementary Figure 5). These elements typically exhibit complex chemical behavior that may not be well captured in other chemistries. Compared to random splitting the MAE (94 meV) increases by only $\sim$ 17% for structural, compositional, and chemical systems splitting (all three have an MAE of $\sim$ 110 meV for nested splits). This surprisingly small increase in MAE may be explained by the work function strongly depending on the element present in the topmost surface layer – hence, as long as an element is present in *any* chemical system (or composition) in the train set, the RF model is able to learn the elemental trend for the work function and can then extrapolate well for an unseen chemical system. The average MAE increases (218 meV) by holding out groups of the periodic table compared to holding out just Elements (149 meV). Similarly, the MAEs increase by holding out point groups (227 meV) and crystal systems (272 meV) compared to just holding out space groups (219 meV). Supplementary Figure 4 displays the parity plots, MAE trends, and residuals for these additional hold-out strategies.

Similar to Figure 2(d), the residuals $|\phi_{\mathrm{DFT}} - \langle\{\phi_{\mathrm{ML}}\}_L\rangle|$ are plotted against the standard deviation of the work function predictions over model ensembles in Figure 3(d). Interestingly, the averages of the residuals within a given bin of $\sigma(\{\mathrm{MAE}\}_K)$ (red markers) tend to have a slightly greater slope than the $x = y$ parity line (cyan). The *overconfidence* of this bootstrapped uncertainty metric appears to be typical of tree-based models using hand-engineered features and therefore requires re-calibration[8] such that the expectation value of the residual for a given σ bin is closer to parity.

Figure 4(b) shows the dataset size dependence of the MAEs and their standard deviations for the work function dataset. A roughly linear decrease in the MAEs is observed with a logarithmic increase in the dataset size for splitting strategies C_K = {Random, Structure, Composition, and Chemsys}. The standard deviations of the MAEs typically decrease with increasing dataset size. However, for splitting strategies C_K = {SG#, and Elements}, the MAEs start to plateau with increasing data size, indicating that additional data may no longer improve the RF model's capability to infer OOD samples accurately.

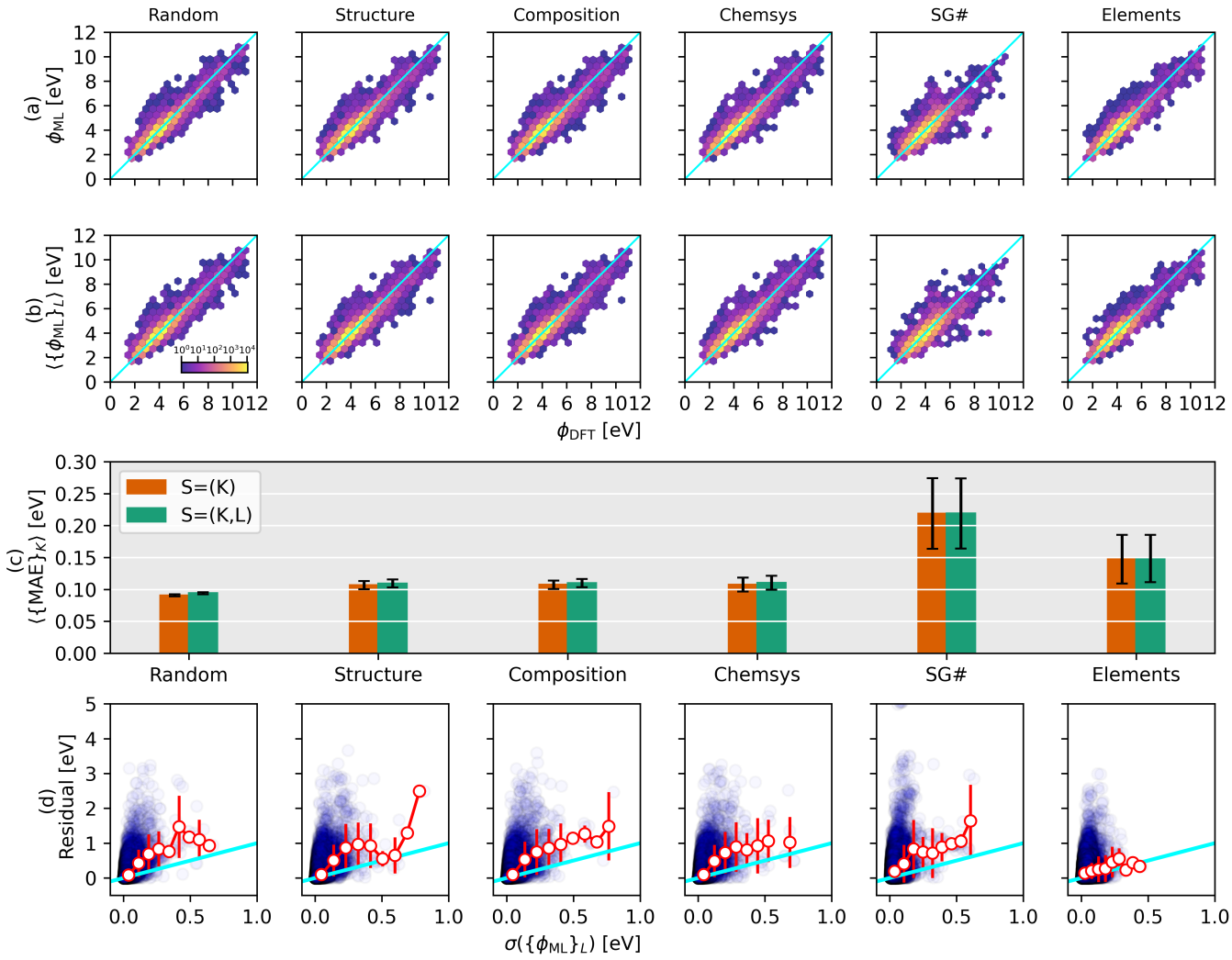
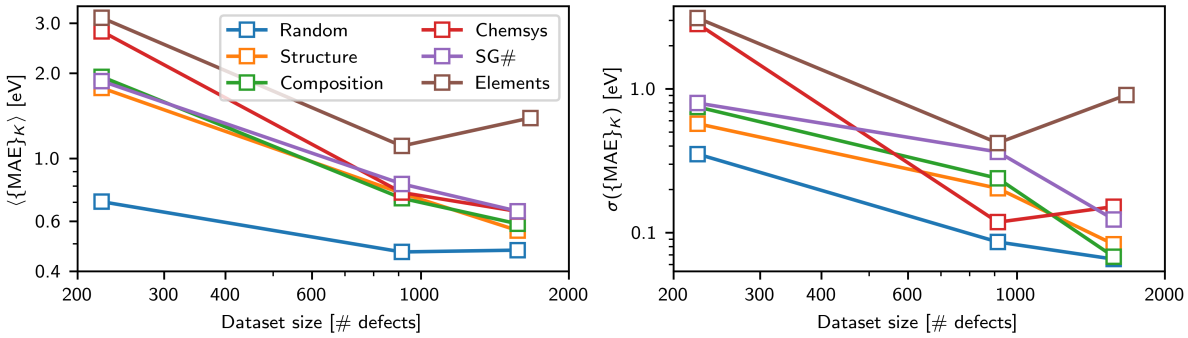


Figure 3: Parity plots of DFT-calculated vs. ML-predicted work functions are shown for (a) K-fold and (b) nested (K, L)-fold splits for different splitting strategies. The color scale is on a logarithmic scale w.r.t. the number of structures at that grid point. The corresponding MAEs are displayed in (c) for K-fold and nested (K, L)-fold splits in green and orange, respectively. The residuals, $|\phi_{\mathrm{DFT}} - \langle\{\phi_{\mathrm{ML}}\}_L\rangle|$, are plotted vs. the standard deviation of the work function predictions (nested K-fold) in (d) alongside the average and standard deviation of residuals in 9 bins (white circles and red error bars, respectively). All units are in eV and the $x = y$ line is highlighted in cyan.

(a) ΔH_V dataset



(b) Work function dataset

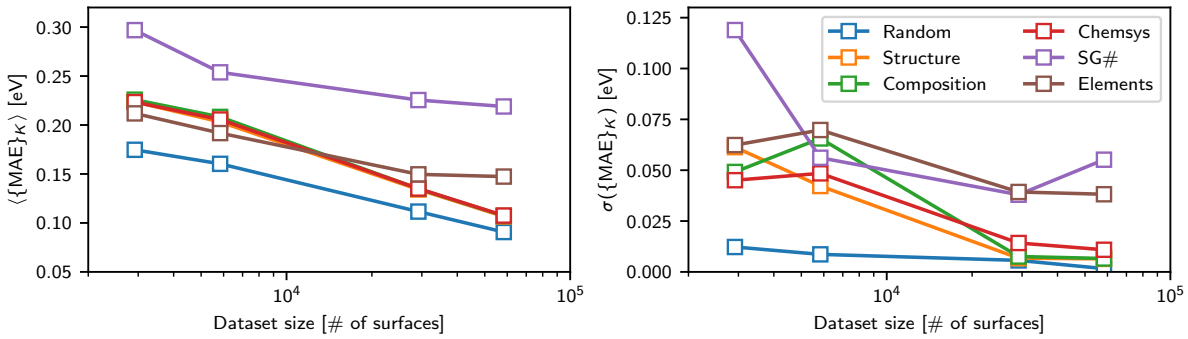


Figure 4: The MAEs (left panels) and standard deviations of the MAEs (right panels) are plotted as a function of dataset size and splitting strategy for (a) the defect dataset and (b) the work function dataset.

Discussion

MatFold provides an automated, easy-to-use tool for generating CV splits of materials data and ultimately enables deeper insights into a data-driven modeling approach's generalizability, uncertainty, and improvability. By computing expected error as a function of the splitting criteria in MatFold, one can both estimate OOD performance (via model agnostic CV splits) and readily and systematically decouple the expected generalization performance of a given modeling approach from its training dataset size. This can be combined with nested CV and bootstrapped model ensembles to ascertain the potential to mitigate over-fitting of high error outliers and the fidelity of uncertainty estimates. Finally, combining all of the above with fractional data hold-out indicates whether continued data collection is beneficial, and most importantly, how it depends on the OOD inference task probed by the different splitting criteria.

Crucially, similarities and differences in MatFold trends can be observed between different modeling approaches and data domains, as demonstrated in our two exemplars, to draw deeper conclusions about their respective strengths and weaknesses. As expected in both exemplars, purely random splits provide the most biased underestimation of expected MAE, but the evolution of expected MAE with increasingly strict splitting criteria is heavily dependent on the modeling approach and data domain. For GNN's predictions of ΔH_V (a direct crystal structure input model), the expected MAE on structures with unseen elements is nearly double that of structures with unseen space groups. Yet the opposite is true for RF predictions of ϕ (a hand-engineered feature input model). Therefore these ΔH_V GNN models generalize better to unseen structural motifs than unseen chemistry, the exact opposite of ϕ RF models.

The ΔH_V GNN predictions also benefit substantially from bootstrapped model ensembling to reduce over-fitting and mitigate outliers in test set prediction parity, while no benefit is observed in the ϕ RF models. Consequently, we observed the need to re-calibrate the bootstrapped uncertainty metric derived for the ϕ RF models, but not for the ΔH_V GNN models. It should be noted that re-tuning the hyperparameters during model ensembling could further reduce over-fitting but comes at a large computational cost (*e.g.*, tuning 2 hyperparameters with 10 possible values each would already require training 100 times more models). Finally, in both exemplars, we generally observe continued improvement in model performance with more training data for moderately difficult OOD inference (*e.g.*, structure, composition, or chemsys splits). However, for their weakest inference task (Elements for ΔH_V GNN and SG# for ϕ RF models), neither is likely to improve

further with additional data indicating fundamental limitations of the respective model architectures.

We anticipate that the splitting criteria and other functionality introduced by MatFold will lower the bar for better and more automated CV of data-driven materials models. Practitioners will have a better understanding of their expected accuracy for materials discovery in increasingly difficult OOD inference, regardless of their modeling approach because MatFold CV splits are only material dependent and entirely model agnostic. This will also enable deeper insights of materials discovery performance spanning differing modeling approaches and data domains and, if widely adopted, provide more grounded evidence for which modeling approaches may be more appropriate in various materials discovery situations.

Acknowledgements

The authors gratefully acknowledge research support from the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Fuel Cell Technologies Office through the HydroGEN Consortium. This work was supported by the Laboratory Directed Research and Development (LDRD) program at Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly-owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration (DOE/NNSA) under contract DE-NA0003525. This written work is authored by an employee of NTESS. The employee, not NTESS, owns the right, title and interest in and to the written work and is responsible for its contents. Any subjective views or opinions that might be expressed in the written work do not necessarily represent the views of the U.S. Government. The publisher acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this written work or allow others to do so, for U.S. Government purposes. The DOE will provide public access to results of federally sponsored research in accordance with the DOE Public Access Plan. P.S. gratefully acknowledged the start-up funds from Northeastern University, Department of Mechanical and Industrial Engineering.

Author Contributions

M. D. W. performed writing – original draft (lead); review and editing (equal); software (equal); methodology (equal); visualization (lead); investigation (equal); data curation (equal); conceptualization (lead); supervision (equal); resources (equal). P. S. performed writing – original draft (supporting); review and editing (equal); software (equal); methodology (equal); visualization (supporting); investigation (equal); data curation (equal); conceptualization (supporting); supervision (equal); resources (equal).

Code Availability

The code will be made available for peer review and open-sourced on GitHub (at github.com/d2r2group/MatFold) and become pip installable upon manuscript publication. A frozen version of the code will also be permanently accessible on Zenodo via this link: 10.5281/zenodo.13147391.

Data Availability

The work function database is available for download at 10.5281/zenodo.10381505. The ΔH_V data,[15,22] is available for download from its original source at 10.5281/zenodo.8087871[21] and 10.1021/jacs.1c05570. We summarize the data in supplementary_files_defects.zip file that contains all .cif files and a .csv file with the corresponding structure name, index of the vacancy defect, and vacancy formation energy.

References

(1) Morgan, D.; Jacobs, R. Opportunities and Challenges for Machine Learning in Materials Science. *Annu. Rev. Mater. Res.* **2020**, *50*, 71–103.

(2) Jiang, S.; Qin, S.; Van Lehn, R. C.; Balaprakash, P.; Zavala, V. M. Uncertainty quantification for molecular property predictions with graph neural architecture search. *Digit. Discov.* **2024**, DOI: 10.1039/D4DD00088A.

(3) Meredig, B.; Antono, E.; Church, C.; Hutchinson, M.; Ling, J.; Paradiso, S.; Blaiszik, B.; Foster, I.; Gibbons, B.; Hattrick-Simpers, J.; Mehta, A.; Ward, L. Can machine learning identify the next high-temperature superconductor? Examining extrapolation performance for materials discovery. *Mol. Syst. Des. Eng.* **2018**, *3*, 819–825.

(4) Omee, S. S.; Fu, N.; Dong, R.; Hu, M.; Hu, J. Structure-based out-of-distribution (OOD) materials property prediction: a benchmark study. *npj Comput. Mater.* **2024**, *10*, 144.

(5) Hu, J.; Liu, D.; Fu, N.; Dong, R. Realistic material property prediction using domain adaptation based machine learning. *Digital Discovery* **2024**, *3*, 300–312.

(6) Jacobs, R.; Schultz, L. E.; Scourtas, A.; Schmidt, K. J.; Price-Skelly, O.; Engler, W.; Foster, I.; Blaiszik, B.; Voyles, P. M.; Morgan, D. Machine Learning Materials Properties with Accurate Predictions, Uncertainty Estimates, Domain Guidance, and Persistent Online Accessibility. *arXiv* **2024**, DOI: 10.48550/arXiv.2406.15650.

(7) Li, K.; DeCost, B.; Choudhary, K.; Greenwood, M.; Hattrick-Simpers, J. A critical examination of robustness and generalizability of machine learning prediction of materials properties. *npj Comput. Mater.* **2023**, *9*, 1–9.

(8) Palmer, G.; Du, S.; Politowicz, A.; Emory, J. P.; Yang, X.; Gautam, A.; Gupta, G.; Li, Z.; Jacobs, R.; Morgan, D. Calibration after bootstrap for accurate uncertainty quantification in regression models. *npj Comput. Mater.* **2022**, *8*, 115.

(9) Gawlikowski, J.; Tassi, C. R. N.; Ali, M.; Lee, J.; Humt, M.; Feng, J.; Kruspe, A.; Triebel, R.; Jung, P.; Roscher, R.; Shahzad, M.; Yang, W.; Bamler, R.; Zhu, X. X. A survey of uncertainty in deep neural networks. *Artif. Intell. Rev.* **2023**, *56*, 1513–1589.

(10) Agrawal, V.; Zhang, S.; Schultz, L. E.; Morgan, D. Accelerating Ensemble Error Bar Prediction with Single Models Fits. *arXiv* **2024**, *2404.09896*.

(11) Zhang, H.; Chen, W. (; Rondinelli, J. M.; Chen, W. ET-AL: Entropy-targeted active learning for bias mitigation in. *Appl. Phys. Rev.* **2023**, *10*, DOI: 10.1063/5.0138913.

(12) Li, K.; Rubungo, A. N.; Lei, X.; Persaud, D.; Choudhary, K.; DeCost, B.; Dieng, A. B.; Hattrick-Simpers, J. Probing out-of-distribution generalization in machine learning for materials. *arXiv* **2024**, 2406.06489.

(13) Xie, T.; Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.* **2018**, *120*, 145301.

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

(15) Witman, M. D.; Goyal, A.; Ogitsu, T.; McDaniel, A. H.; Lany, S. Defect graph neural networks for materials discovery in high-temperature clean-energy applications. *Nat. Comput. Sci.* **2023**, *3*, 675–686.

(16) Schindler, P.; Antoniuk, E. R.; Cheon, G.; Zhu, Y.; Reed, E. J. Discovery of Stable Surfaces with Extreme Work Functions by High-Throughput Density Functional Theory and Machine Learning. *Adv. Funct. Mater.* **2024**, *2401764*, 1–12.

(17) Chen, C.; Ye, W.; Zuo, Y.; Zheng, C.; Ong, S. P. Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. *Chem. Mater.* **2019**, *31*, 3564–3572.

(18) Choudhary, K.; DeCost, B.; Major, L.; Butler, K.; Thiyagalingam, J.; Tavazza, F. Unified graph neural network force-field for the periodic table: solid state applications. *Digit. Discov.* **2023**, *2*, 346–355.

(19) Deng, B.; Zhong, P.; Jun, K.; Riebesell, J.; Han, K.; Bartel, C. J.; Ceder, G. CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nat. Mach. Intell.* **2023**, *5*, 1031–1041.

(20) Batatia, I. et al. A foundation model for atomistic materials chemistry. *arXiv* **2023**, arXiv:2401.00096.

(21) Witman, M.; Goyal, A.; Ogitsu, T.; McDaniel, A. H.; Lany, S. A database of vacancy formation enthalpies for materials discovery (0.0.1) [dataset]. *Zenodo* **2023**, 10.5281/zenodo.8087871.

(22) Wexler, R. B.; Gautam, G. S.; Stechel, E. B.; Carter, E. A. Factors Governing Oxygen Vacancy Formation in Oxide Perovskites. *J. Am. Chem. Soc.* **2021**, *143*, 13212–13227.

(23) Lu, H.-J.; Zou, N.; Jacobs, R.; Afflerbach, B.; Lu, X.-G.; Morgan, D. Error assessment and optimal cross-validation approaches in machine learning applied to impurity diffusion. *Comput. Mater. Sci.* **2019**, *169*, 109075.

(24) Schindler, P.; Riley, D. C.; Bargatin, I.; Sahasrabuddhe, K.; Schwede, J. W.; Sun, S.; Pianetta, P.; Shen, Z.-X.; Howe, R. T.; Melosh, N. A. Surface Photovoltage-Induced Ultralow Work Function Material for Thermionic Energy Converters. *ACS Energy Lett.* **2019**, *4*, 2436–2443.

(25) Antoniuk, E. R.; Schindler, P.; Schroeder, W. A.; Dunham, B.; Pianetta, P.; Vecchione, T.; Reed, E. J. Novel Ultrabright and Air-Stable Photocathodes Discovered from Machine Learning and Density Functional Theory Driven Screening. *Adv. Mater.* **2021**, *33*, 2104081.

(26) Antoniuk, E. R.; Yue, Y.; Zhou, Y.; Schindler, P.; Schroeder, W. A.; Dunham, B.; Pianetta, P.; Vecchione, T.; Reed, E. J. Generalizable density functional theory based photoemission model for the accelerated development of photocathodes and other photoemissive devices. *Phys. Rev. B* **2020**, *101*, 235447.