Supplementary Information: 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

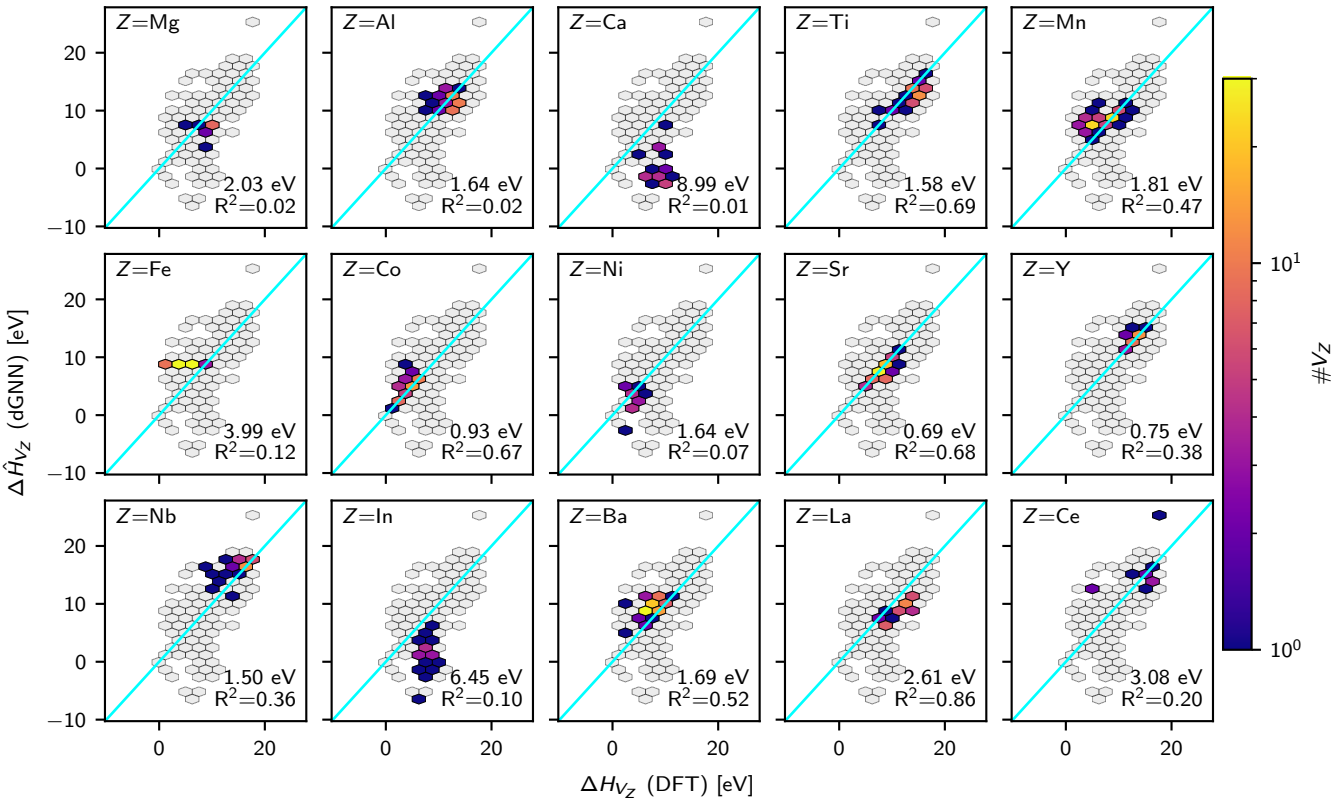
[2]Northeastern University, Boston, Massachusetts 02115, United States

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

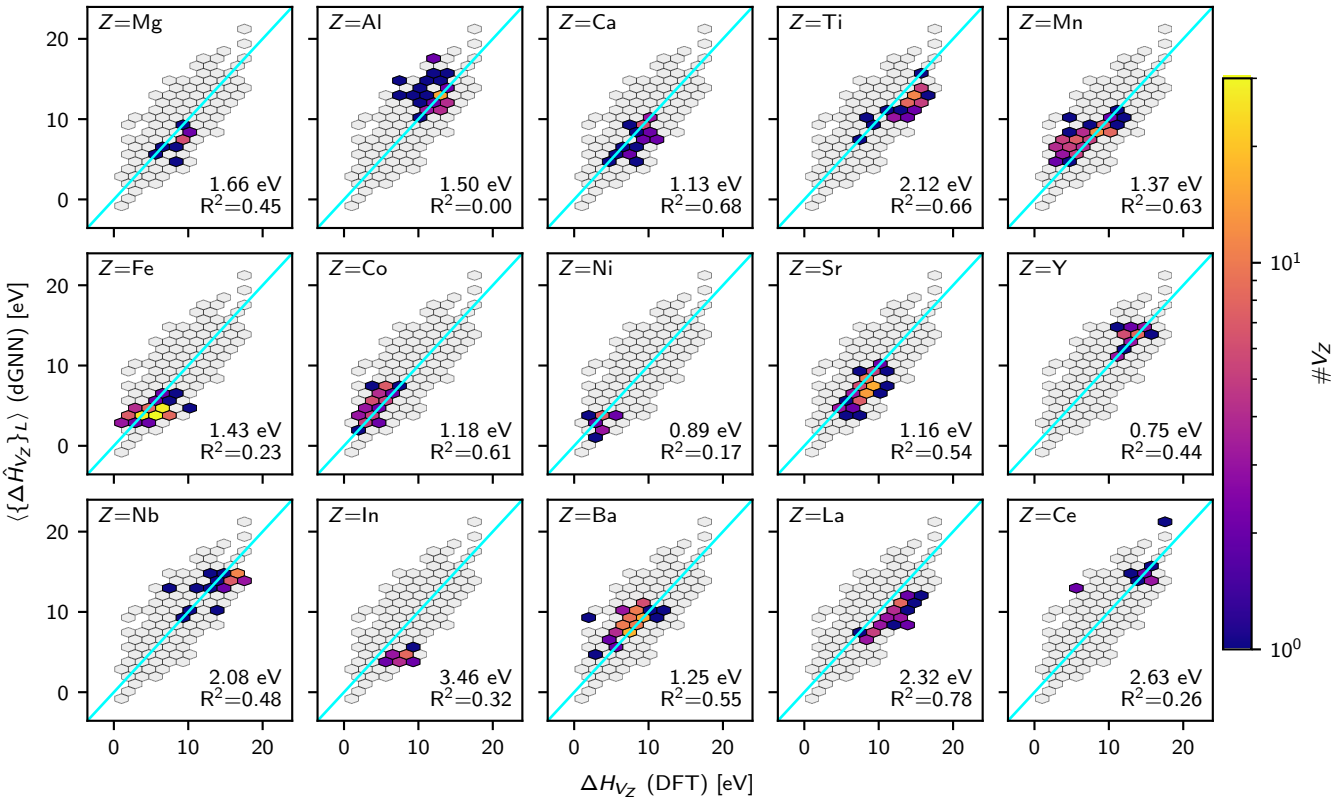
Supplementary Vacancy Formation Energy CV Analysis

Each structure in the vacancy formation energy dataset represents an oxide containing multiple computed neutral vacancy formation energies for various oxygen or cation sites in the crystal structure. Therefore, even C_k = Element test set examples may *possibly* still correspond to a local environment that is fairly similar to a train example. This could occur if the training example corresponds to a vacancy site that is not the hold-out element, or the hold-out element does not populate a site with close interaction with the vacancy site. The test set predictions can be further filtered to keep only those whose vacancy site's elemental identity, V_Z, is the same as the hold-out element, Z. In this extreme case, not only does the test set example correspond to a structure containing an unseen element during the training, but the actual vacancy site also corresponds to that unseen element.

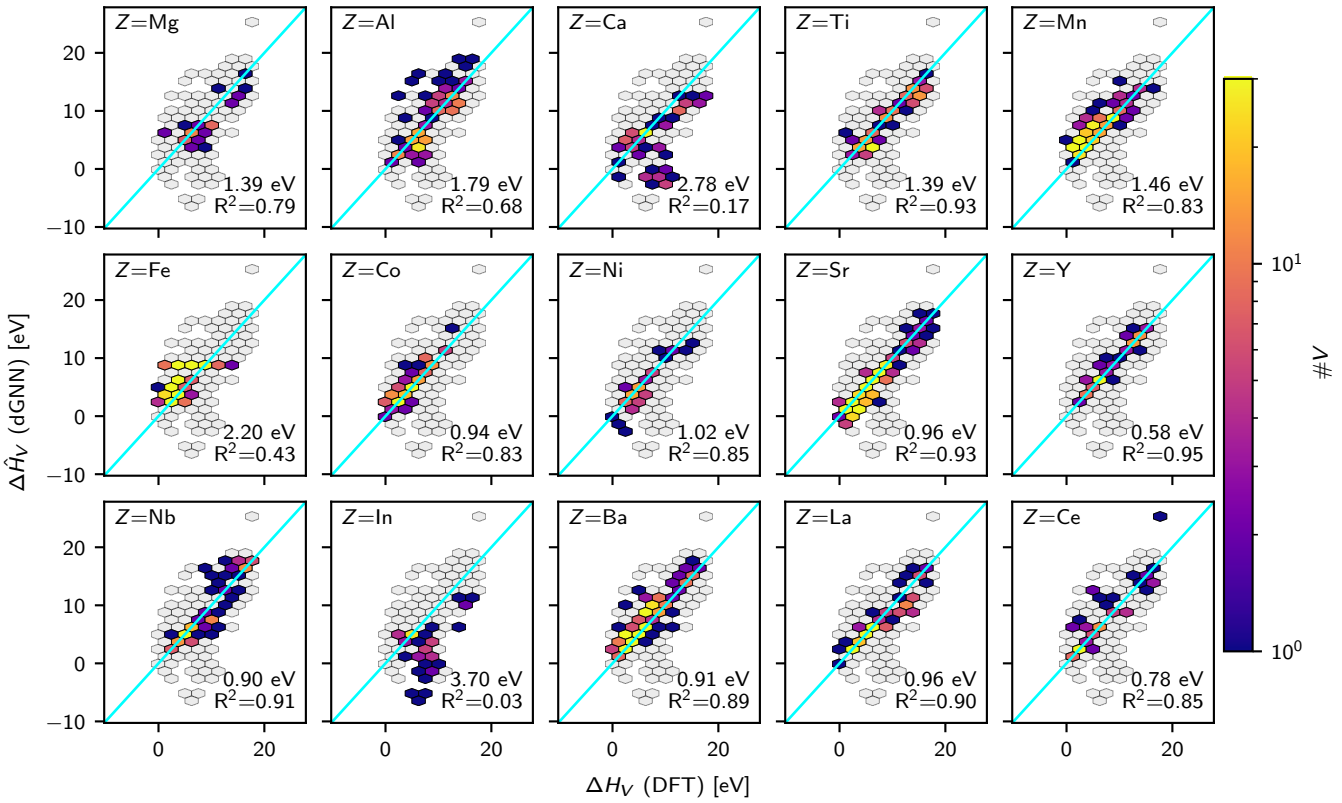
Supplementary Figures 1 and 2 re-visualize Figure 2(a), (b) of the main text, respectively. All test set predictions are shown in gray, but only the subset of $\Delta\hat{H}_{V_Z}$ test set predictions are color-coded, and their MAE and R^2 are reported. As expected the model performance is poor, however, surprisingly high R^2 values can be achieved for some elements like Ti, Co, Sr, Y, and La, while exceptionally poor performance is observed for others like Mg, Al, Fe, Ca, and In. Simply using bootstrapped model ensembles (Supplementary Figure 2) instead of a single model (Supplementary Figure 1), helps correct for these most egregious errors and generally improves the MAE and R^2 across all hold-out elements. Ultimately, Supplementary Figure 2's extreme OOD subsets of test predictions, whose errors are partially masked by all predictions in that test set (Supplementary Figure 4), can yield surprisingly high R^2 and empirical evidence for the excellent generalization capabilities of dGNN and GNNs in general. As an alternative visualization of Fig 2(c), we also plot the expected MAE and R^2 test set statistics as quartile box plots in Supplementary Figure 5.



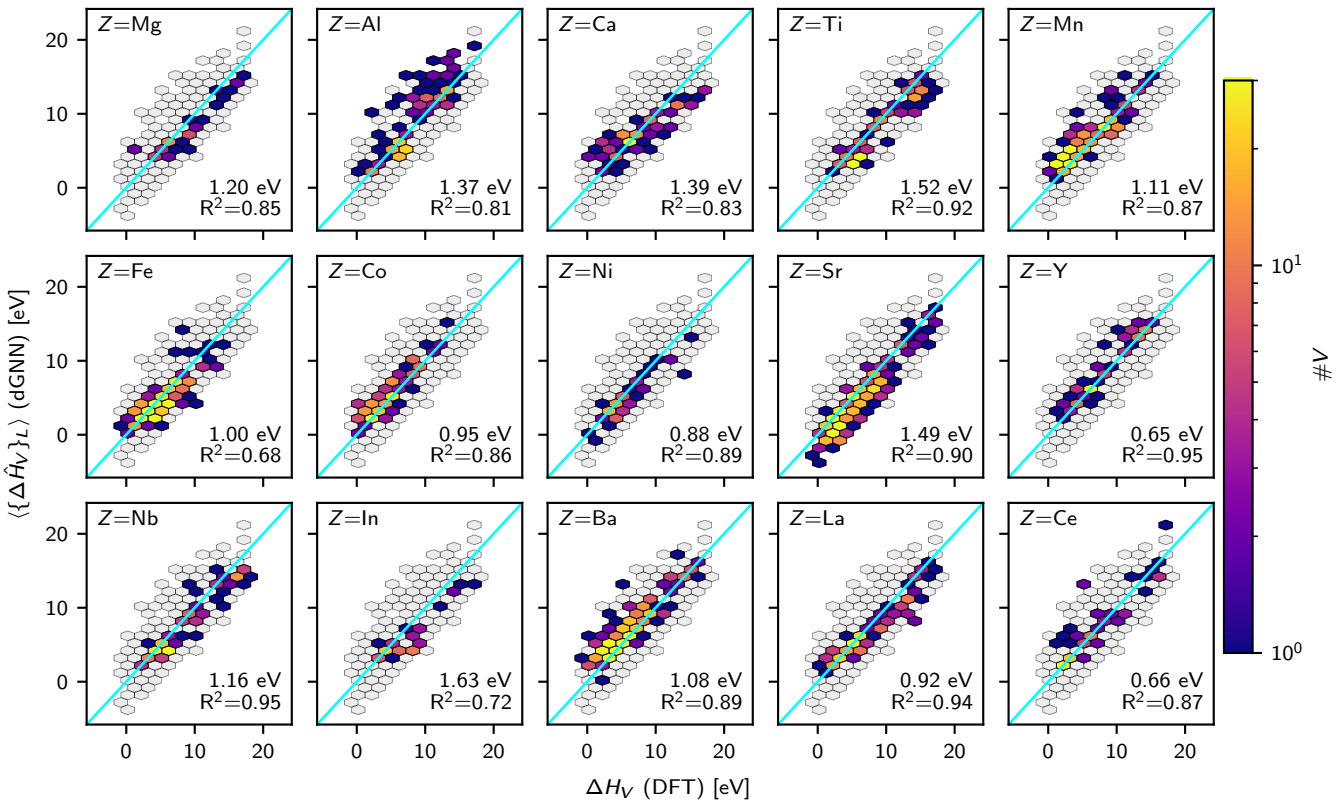
Supplementary Figure 1: Re-visualizing Figure 2(a), corresponding to C_k = Element, T = None, $S = K$, but broken down by each test fold Z, where only V_Z test predictions are color-coded.



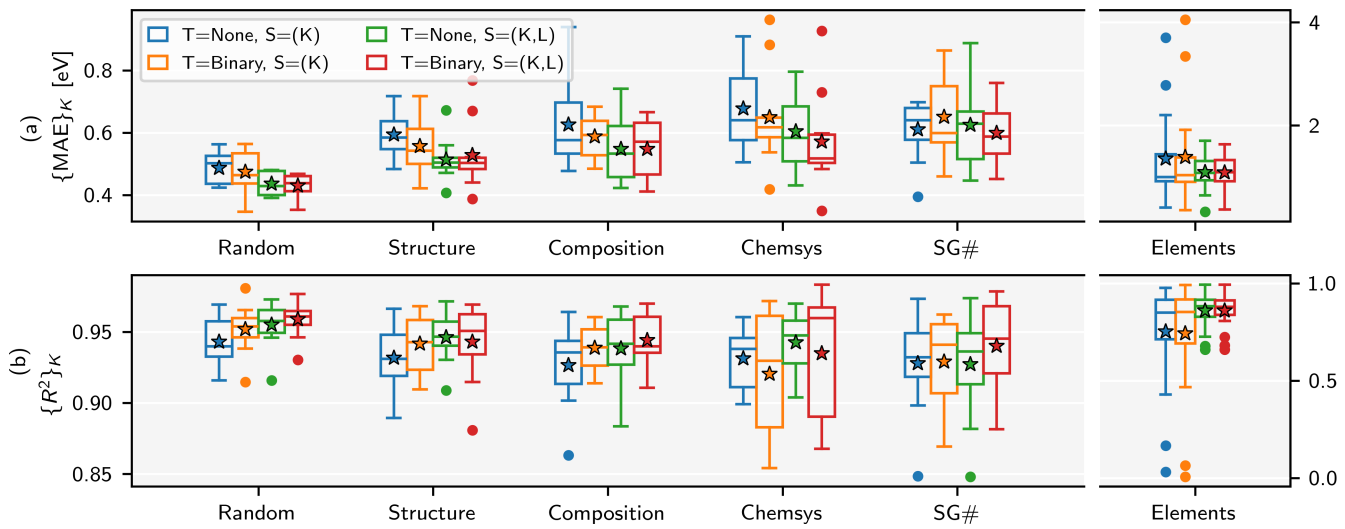
Supplementary Figure 2: Re-visualizing Figure 2(b), corresponding to C_k = Element, T = Binary, $S = (K, L)$, but broken down by each test fold Z, where only V_Z test predictions are color-coded.



Supplementary Figure 3: Re-visualizing Figure 2(a), corresponding C_k = Element, T = None, $S = K$, but broken down by each test fold Z where all test predictions are color-coded.



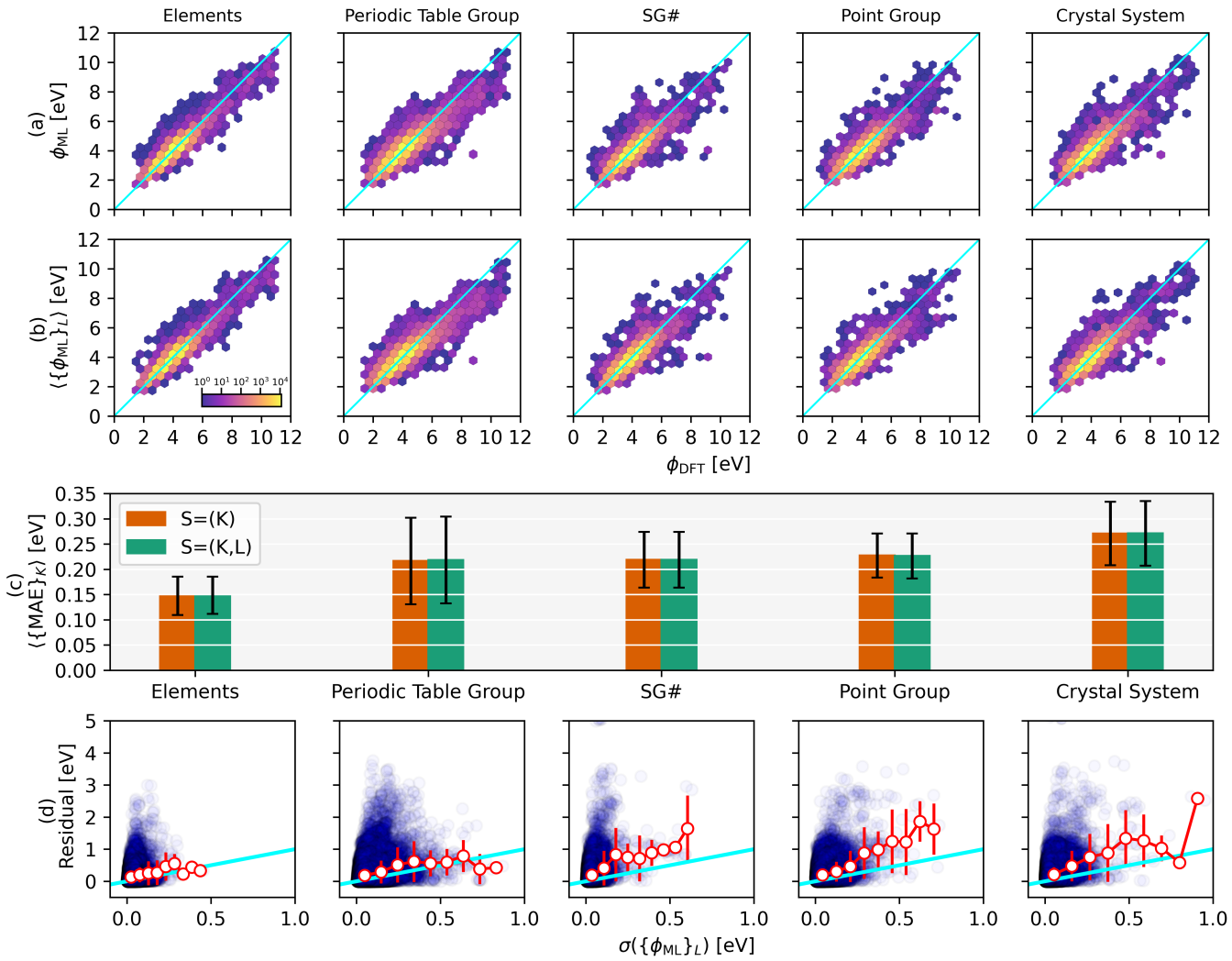
Supplementary Figure 4: Re-visualizing Figure 2(b), corresponding to C_k = Element, T = Binary, $S = (K, L)$, but broken down by each test fold Z where all test predictions are color-coded.



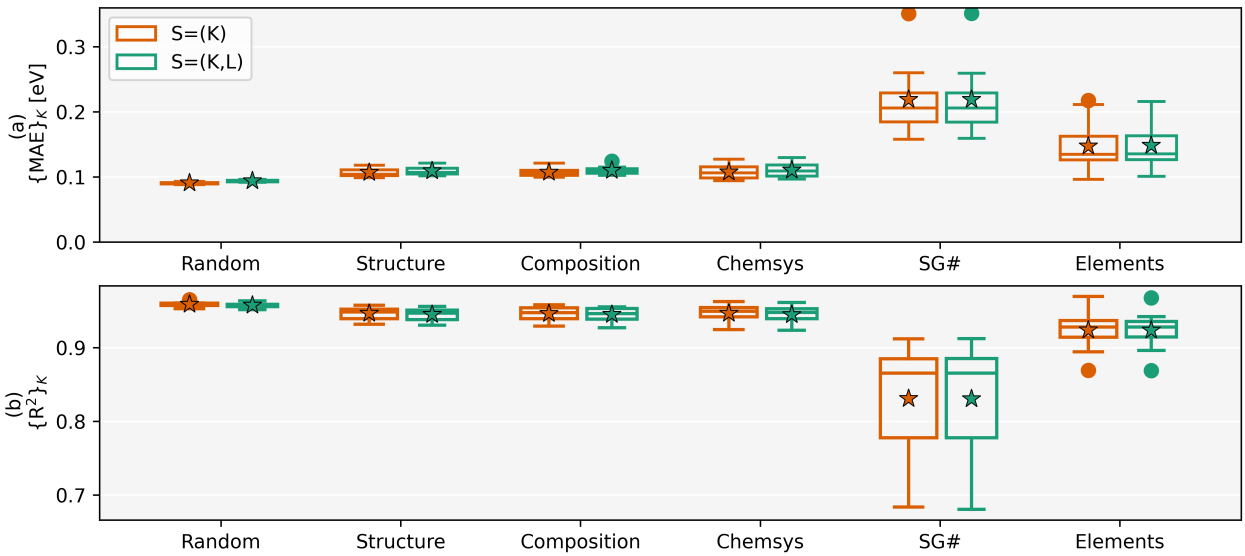
Supplementary Figure 5: Re-visualizing Figure 2(c), but using quartile box plots to represent (a) the expectation MAE and (b) R^2 of the outer test set predictions for the defect dataset. Circles represent outliers, while stars represent the mean.

Supplementary Work Function CV Analysis

Supplementary Figure 6 supports Figure 3 of the main text for splitting strategies $C_K = \{$Elements, Periodic Table Group, SG#, Point Group, and Crystal System$\}$. As an alternative visualization of Fig 3(c), we also plot the expected MAE and R^2 test set statistics as quartile box plots in Supplementary Figure 7. Supplementary Figure 8 shows the periodic table of elements heatmaps of MAEs and test set fraction for the work function dataset utilizing the LOO element split strategy. The largest MAEs are observed for holding out F, H, O, or Cl.

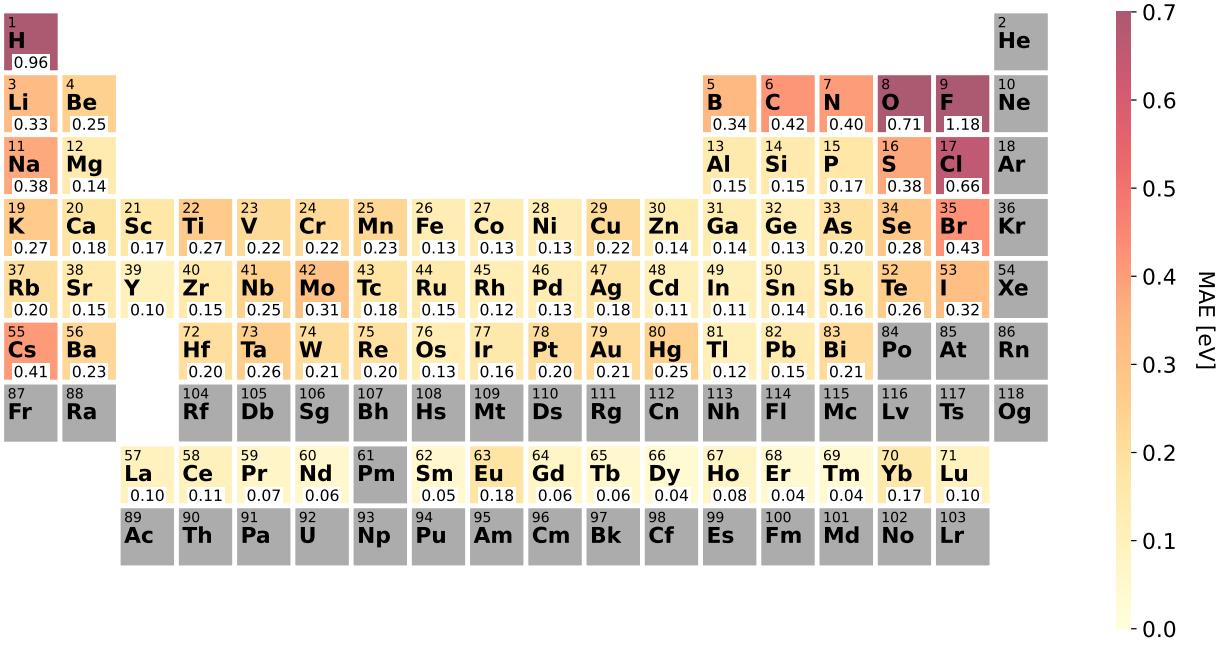


Supplementary Figure 6: Parity plots of DFT-calculated vs. ML-predicted work functions are shown for (a) K-fold and (b) nested (K, L)-fold splits for splitting strategies $C_K = \{$Elements, Periodic Table Group, SG#, Point Group, and Crystal System$\}$. The color scale is on a logarithmic scale w.r.t 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 in eV and the $x = y$ line is highlighted in cyan.

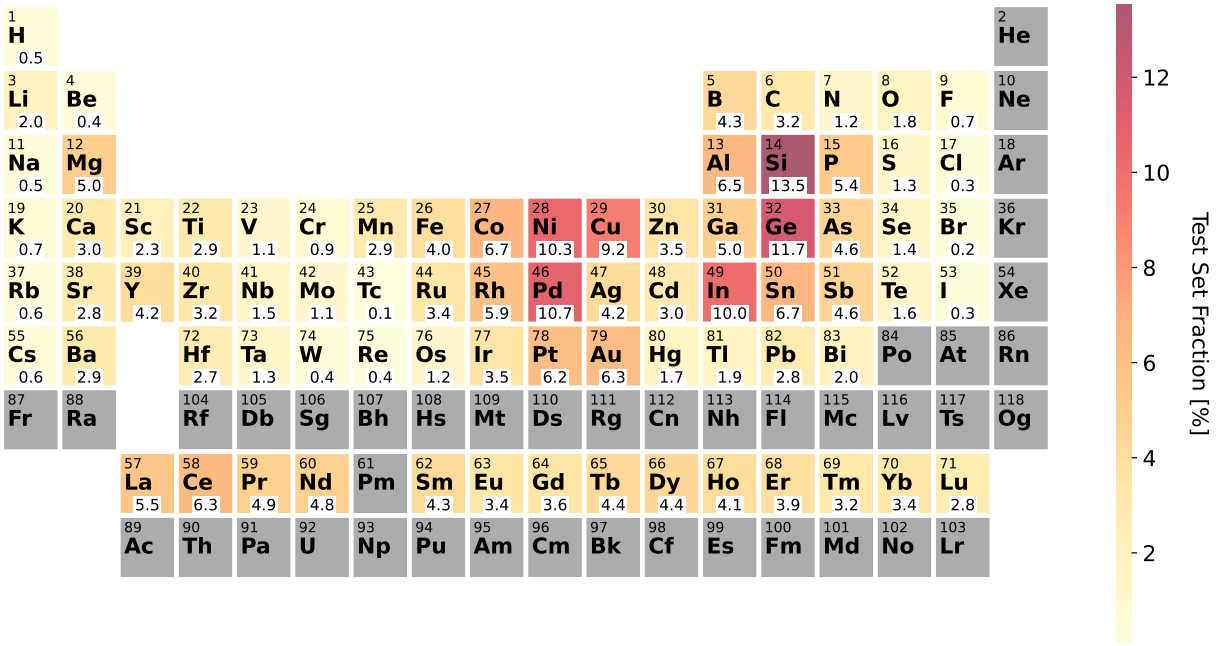


Supplementary Figure 7: Re-visualizing Figure 3(c), but using quartile box plots to represent (a) the expectation MAE and (b) R^2 of the outer test set predictions for the work function dataset. Circles represent outliers, while stars represent the mean.

(a) MAE heatmap



(b) Test set fraction heatmap



Supplementary Figure 8: Periodic table of elements heatmaps of (a) MAEs (in eV) and (b) test set fraction (in percent of total dataset size) are displayed for the work function dataset utilizing the LOO element split strategy.

```
{
    "MatFold": "v1.1.0",
    "data_checksums": [
        164872556148165495537414186909580076184,
        137484994316218657549518869761510670017
    ],
    "__init__": {
        "return_frac": 0.5,
        "always_include_n_elements": null,
        "cols_to_keep": null,
        "seed": 0
    },
    "split_function_called": "create_splits",
    "split_parameters": {
        "split_type": "crystalsys",
        "n_inner_splits": 0,
        "n_outer_splits": 0,
        "fraction_upper_limit": 0.8,
        "fraction_lower_limit": 0.0,
        "keep_n_elements_in_train": 2,
        "min_train_test_factor": null,
        "inner_equals_outer_split_strategy": true,
        "write_base_str": "mf",
        "output_dir": "./output/",
        "verbose": true
    }
}
```


Supplementary Figure 9: MatFold JSON configuration file that stores the necessary class and function variable settings to reproduce identical splits. This file is generated during split creation and can later be loaded through a class function (see code documentation on GitHub). The original materials data is not stored but checksums are generated and a warning is printed if the data has changed since the last split creation.