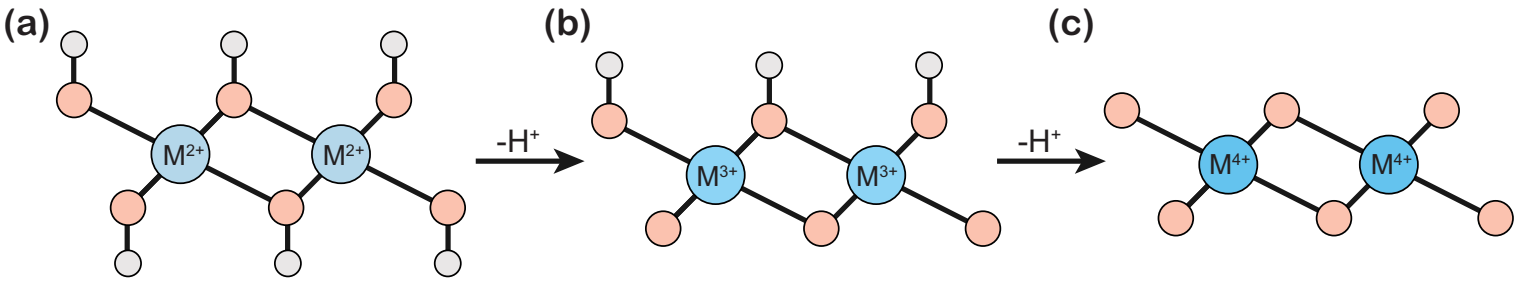
Supplementary Information

Chemical Hardness-Driven Interpretable Machine Learning Approach for Rapid Search of Photocatalysts

Ritesh Kumar[†] and Abhishek K. Singh[*†]

[†]Materials Research Centre, Indian Institute of Science, Bangalore, Karnataka 560012



Supplementary Figure 1: Schematic structures of 2D (a) layered double hydroxides (LDHs), (b) layered oxyhydroxides, and (c) layered double oxides. The orange and gray circles represent O and H atoms, respectively.

Supplementary Note 1: Feature Ranking using Various ML Methods

Linear machine learning algorithms, e.g., linear regression, logistic regression, LASSO and ridge regression fit a model where the prediction is the weighted sum of the input values. They find a set of coefficients to use in the weighted sum in order to make a prediction. These coefficients can be used directly as a crude type of feature importance score. Feature ranking using linear regression works well when the data is not very noisy (or there is a lot of data compared to the number of features) and the features are independent. Therefore, it is not optimal for selecting the top performing features for improving the generalization of a model. Regularization is a method for adding additional constraints or penalty to a model, with the goal of preventing overfitting and improving generalization. L_1 regularization is included in LASSO, which adds a penalty $\alpha \sum_{i=1}^{n} |w_i|$ to the loss function. Since each non-zero coefficient adds to the penalty, it forces weak features to have zero as coefficients. Therefore, it is useful when the sole purpose is to reduce the number of features, but not necessarily for data interpretation, since it might lead to the conclusion that certain features do not have a strong relationship with the output variable. Ridge regression with L_2 regularization adds a penalty $\alpha \sum_{i=1}^{n} w_i^2$ to the loss function. Since the coefficients are squared in the penalty expression, it forces the coefficient values to be spread out more equally. It leads to similar coefficients for the correlated features. The coefficients also do not fluctuate on small data changes as is the case with unregularized or L_1 models. A Random Forest consists of several decision trees, in which every node is a condition on a single feature. It is designed to split the dataset into two so that similar response values end up in the same set. The measure based on which the optimal condition is chosen is called the impurity. Therefore, it is possible to compute the extent to which each feature decreases the weighted impurity in a tree during its training. The decrease in impurity from each feature is averaged for a forest, and the features are ranked according to this measure. RFE is a wrapper-type feature selection method, which uses another model (e.g., linear Regression or SVM) to select the best-performing features. It is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and re-fitting the model. This process is repeated until a specified number of features remains.

[*]abhishek@iisc.ac.in

Supplementary Note 2: Bayesian Hyperparameter Optimization

Hyperparameters, in contrast to model parameters, are set by the users before training, e.g., the number of trees in a random forest is a hyperparameter. The aim is to find the hyperparameters of a given machine learning algorithm that return the best performance as measured on a validation or test set. It can be represented in mathematical form as:

$$x^* = arg\min_{x \in \chi} f(x) \tag{1}$$

where $f(x)$ represents an objective score to minimize (e.g., RMSE) or maximize (e.g., MCC) accuracy metrics on the test set; x^* is the set of hyperparameters that yields the lowest value of the score, and x can take on any value in the domain χ. However, evaluating the objective function to find the best score is extremely expensive. For optimizing each hyperparameter, a model is trained on the training data, predictions made on the test data, and then the accuracy metric is calculated. With a large number of hyperparameters and complex models such as ensembles or deep neural networks, which can take days to train, this process quickly becomes intractable to be performed manually.

Grid search and random search are slightly better than manual tuning because a grid of model hyperparameters is set up and the train-predict-evaluate cycle is performed automatically in a loop. However, even these methods are relatively inefficient because they do not choose the next hyperparameters for evaluation based on the previous results. Grid and random search methods are completely uninformed by past evaluations, and as a result, often spend a significant amount of time evaluating "undesirable" hyperparameters.

Bayesian approaches, in contrast to the random or grid search, keep track of past evaluation results, which they use to form a probabilistic model mapping hyperparameters to a probability of a score on the objective function – P(score|hyperparameters). In the literature, this model is called a "surrogate" for the objective function and is represented as $p(y|x)$. The surrogate is easier to optimize than the objective function and Bayesian methods work by finding the next set of hyperparameters to evaluate on the actual objective function by selecting hyperparameters that perform best on the surrogate function.

Tree-structured Parzen Estimator (TPE): In our study, we utilized TPE as the surrogate function. The Tree-structured Parzen Estimator builds a model by applying Bayes rule. Instead of directly representing $p(y|x)$, it instead uses:

$$p(y|x) = \frac{p(x|y) * p(y)}{p(x)} \tag{2}$$

$p(x|y)$, which is the probability of the hyperparameters given the score on the objective function, in turn is expressed:[1]

$$p(x|y) = \begin{cases} l(x), & \text{if } y < y^* \\ g(x), & \text{if } y \geqslant y^* \end{cases}$$

where $y < y^*$ represents a lower value of the objective function than the threshold. There are two different distributions for the hyperparameters: one where the value of the objective function is less than the threshold, $l(x)$, and one where the value of the objective function is greater than the threshold, $g(x)$.

Selection function: The selection function is the criteria by which the next set of hyperparameters are chosen from the surrogate function. The most common choice of criteria is the expected improvement:[1]

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y) p(y|x) dy \tag{3}$$

where y^* is a threshold value of the objective function, x is the proposed set of hyperparameters, y is the actual value of the objective function using hyperparameters x, and $p(y|x)$ is the surrogate probability model expressing the probability of y given x. The aim is to maximize the expected improvement with respect to x, which means finding the best hyperparameters under the surrogate function $p(y|x)$.

If $p(y|x)$ is zero everywhere for $y < y^*$, then the hyperparameters x are not expected to yield any improvement. On the other hand, if the integral is positive, then the hyperparameters x are expected to yield a better result than the threshold value. On the application of Bayes Rule to the expected improvement, its expression becomes:[1]

$$EI_{y^*}(x) = \frac{\gamma y^* l(x) - l(x) \int_{-\infty}^{y^*} p(y) dy}{\gamma l(x) + (1-\gamma) g(x)} \propto \left(\gamma + \frac{g(x)}{l(x)} (1-\gamma) \right)^{-1} \tag{4}$$

According to the term on the far right, the expected improvement is proportional to the ratio $l(x)/g(x)$ and therefore, to maximize the expected improvement, this ratio has to be maximized. Therefore, the TPE works by drawing sample hyperparameters from $l(x)$, evaluating them in terms of $l(x)/g(x)$, and returning the set that yields the highest value under $l(x)/g(x)$ corresponding to the greatest expected improvement. These hyperparameters are then evaluated on the objective function. If the surrogate function is correct, then these hyperparameters should yield a better value when evaluated.

Plots obtained using Optuna: The hyperparameter importance plot depicts the average importance of each hyperparameter towards the objective in the overall trials. The variation of objective value (e.g., test RMSE) as a function of each hyperparameter with different trials can be seen through slice plots. Finally, the contour plots describe the variation of the objective value against a pair of hyperparameters in the form of contours, where each dot represents a trial.

Supplementary Note 3: Theory of SHAP

SHAP stands for SHapley Additive exPlanations signifying that the concept of SHAP has emerged from Shapley values, which was developed by Lloyd Shapley in the field of game theory.[2] Through Shapley values, one can calculate the contribution of each player in a coalition game, assuming N players and S subset of the N players. Let $\nu(S)$ be the total value of the S players. When a player i join the S players, the player i's marginal contribution is $\nu(S \cup \{i\}) - \nu(S)$. The contribution of the player i can be estimated by taking the average of the contribution over possible different permutations in which the coalition can be formed:

$$\varphi_i(\nu) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} (\nu(S \cup \{i\}) - \nu(S)) \tag{5}$$

There are four axioms proposed by Shapley to achieve a fair contribution:[2]

- Axiom 1: The sum of the Shapley values of all players equals the value of the total coalition.
- Axiom 2: All players have a fair chance to join the game.
- Axiom 3: If a player i contributes nothing to any coalition S, then the contribution of the player i is zero, i.e., $\varphi_i(\nu) = 0$.
- Axiom 4: For any pair of games $\nu, w : \varphi(\nu, w) = \varphi(\nu) + \varphi(w)$ where $(\nu + w)(S) = \nu(S) + w(S)$ for all S. This property enables simple arithmetic summation.

The above concept can be easily extended to machine learning algorithms such as Random forests or gradient boosting in extracting the contribution of each feature towards final predictions. Variables enter the machine learning model sequentially or repeatedly in the trees of the model. In each step of the tree growth, the algorithms evaluate all the variables equally to settle with the variable that contributes the most. Therefore, the marginal contribution of each variable can be calculated.

Supplementary Note 4: Different Hyperparameters Used

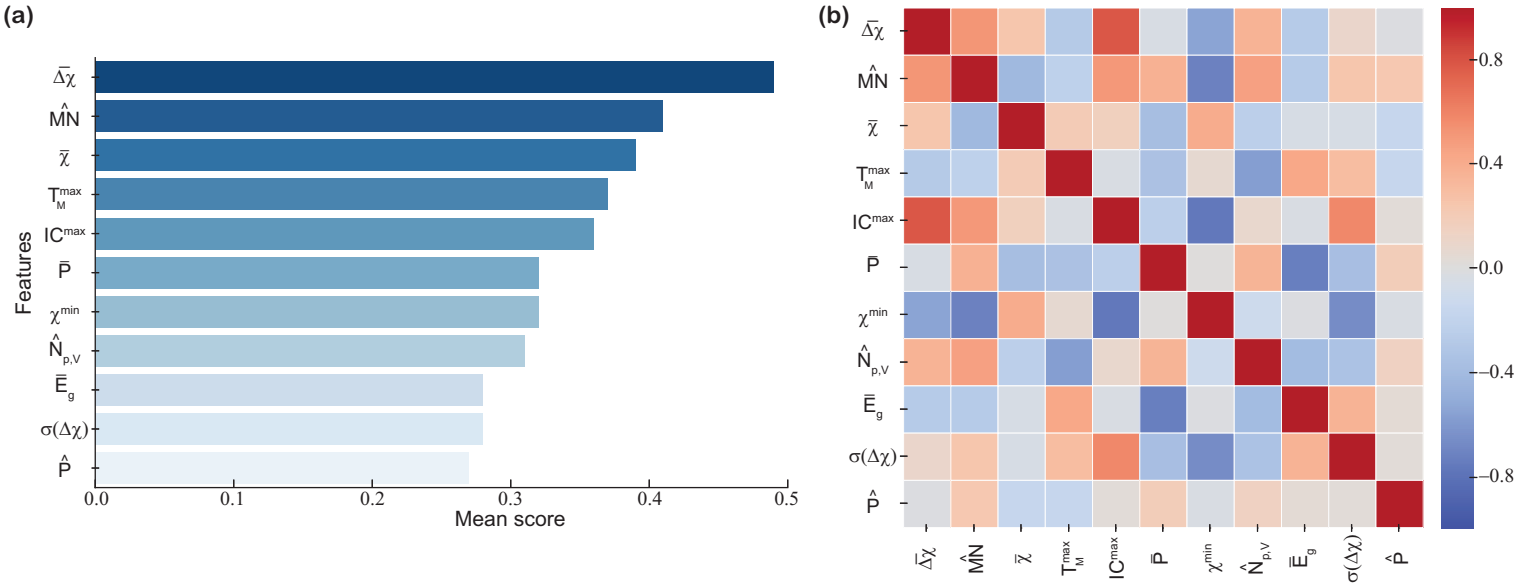
- **LightGBM:**[3]
  - Learning rate. Gradient boosting involves creating and adding trees to the model sequentially. New trees are created to correct the residual errors in the predictions from the existing sequence of trees. It leads to fitting of training dataset by the model, and evenutally overfitting of data. The learning in the LightGBM model can be slowed down by applying a weighting factor for the corrections by new trees when they are added to the model. This weighting factor is called the learning rate.
  - Number of leaves. The number of leaves is one of the most important parameters that controls the complexity of the model. The maximum number of leaves each weak learner has can be set by this hyperparameter. A large number of leaves increases accuracy on the training set and increases the chance of overfitting at the same time.
  - Minimum child samples. It is the minimal number of data, which can be contained in one leaf. Its optimal value depends on the number of training samples and the number of leaves. Setting it to a large value can avoid growing a very deep tree, but may cause underfitting.
  - Bagging fraction. The percentage of rows used per each iteration of tree construction can be speicified with the bagging fraction hyperparameter. Therefore, some rows will be randomly selected for fitting each learner or tree. It improves generalization and the speed of training.
  - Feature fraction. LightGBM randomly selects a subset of features on each iteration (tree) and the feature fraction hyperparameter deals with column sampling. For example, if it is set to 0.6, LightGBM will select 60% of features before training each tree. There are two advantages of using for this feature – speeding up training, and dealing with overfitting.
  - Bagging frequency. A bagging frequency of integer k means performing bagging at every k iteration, while a value of zero will disable the bagging. LightGBM randomly selects a bagging fraction * 100% of the data every kth iteration to use for the next k iterations. It can be used for dealing with overfitting when used along with the bagging fraction hyperparameter.
  - $\lambda_{L_1}/\lambda_{L_2}$. Both hyperparameters are used for controling the L_1 or L_2 regularization to deal with the issue of overfitting.
- **Extra Trees:**[4]
  - Number of estimators. It is the number of trees that are used to grow in a forest. More number of trees should produce a more generalized result. However, choosing more number of trees leads to the increase in complexity of the ET model.

– Maximum depth. It is the maximum depth or number of levels in a decision tree. If it is set to none, then nodes are expanded until all leaves are pure or until all leaves contain less than the minimum samples required for splitting. As the maximum depth of the decision tree increases, the performance of the model over the training set increases continuously. On the other hand, the performance over the test set increases initially but after a certain point, it starts to decrease rapidly due to overfitting.

– Minimum samples at a leaf node. It is the minimum number of data points allowed in a leaf node. A split point at any depth will only be considered if it leaves at least minimum samples at the leaf node of training samples in each of the left and right branches. It is used for controlling the growth of the tree by setting a minimum sample criterion for terminal nodes. This hyperparameter helps in preventing the overfitting as the parameter value increases.

– Minimum samples required for splitting. It is the minimum number of data points placed in a node before the node is split. By increasing the value of the minimum samples required for splitting, the number of splits occuring in the decision tree can be reduced. Therefore, it can prevent the model from overfitting. When its value is increased significantly, there is an overall dip in both the training and test scores. This is due to the fact that the minimum requirement of splitting a node is so high that there are no significant splits observed. As a result, the ET model starts to underfit.

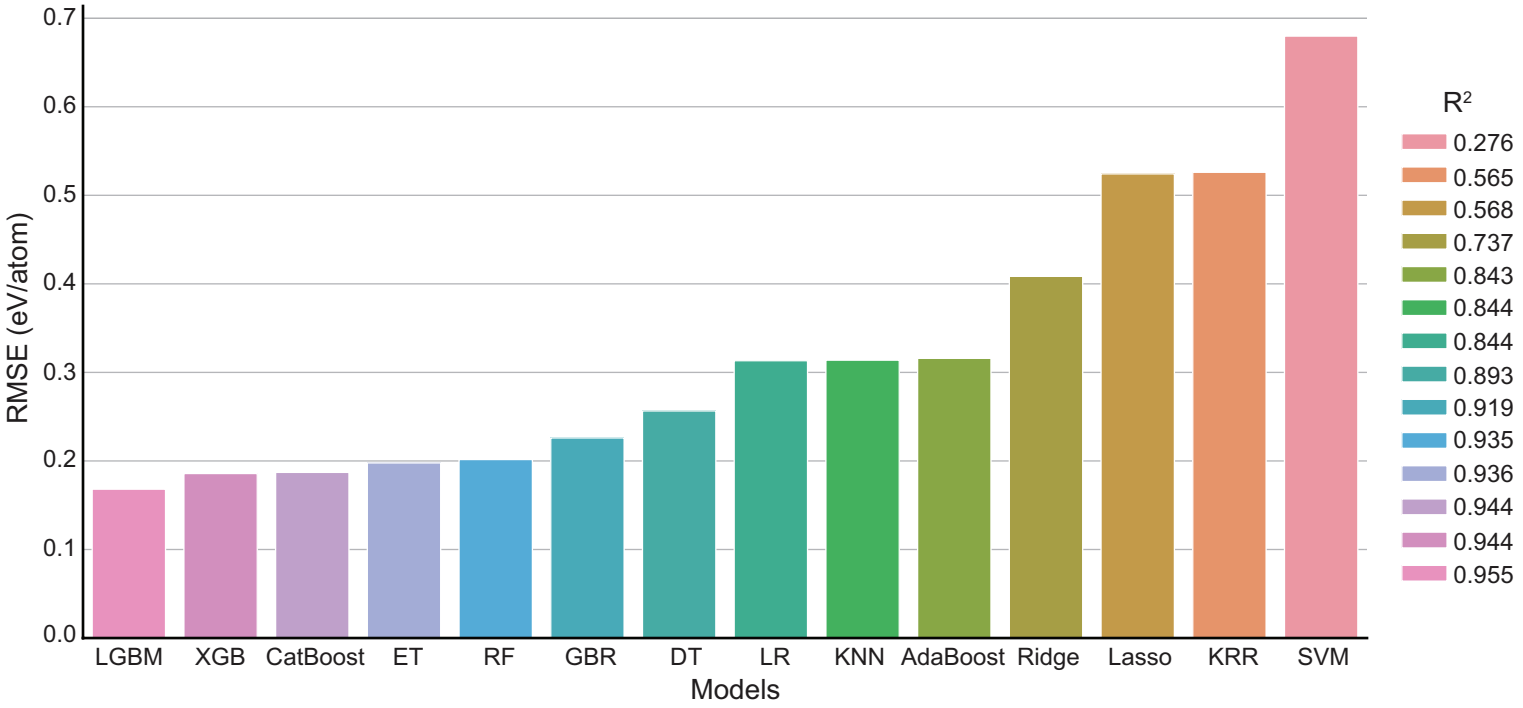
– Minimum weight fraction of leaf. It is the minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node. Samples have equal weight when sample weight is not provided.



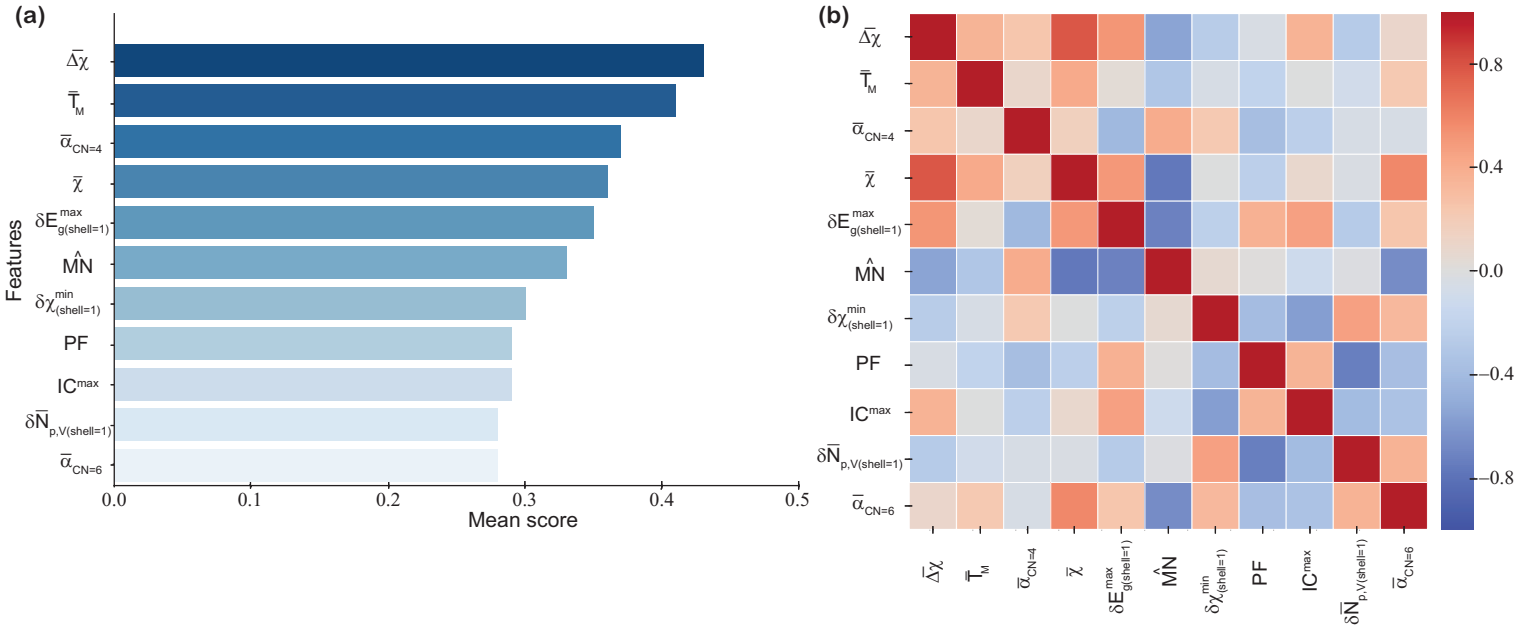
Supplementary Figure 2: (a) Mean score of best features selected from the $\{\mathbf{E}\}$ feature set (highly correlated features removed) for ΔE_f regression. (b) Pearson correlation between the best $\{\mathbf{E}\}$ features (after removing highly correlated features) selected from feature ranking.

Supplementary Table 1: List of all attributes selected from mean feature ranking utilized in the present study

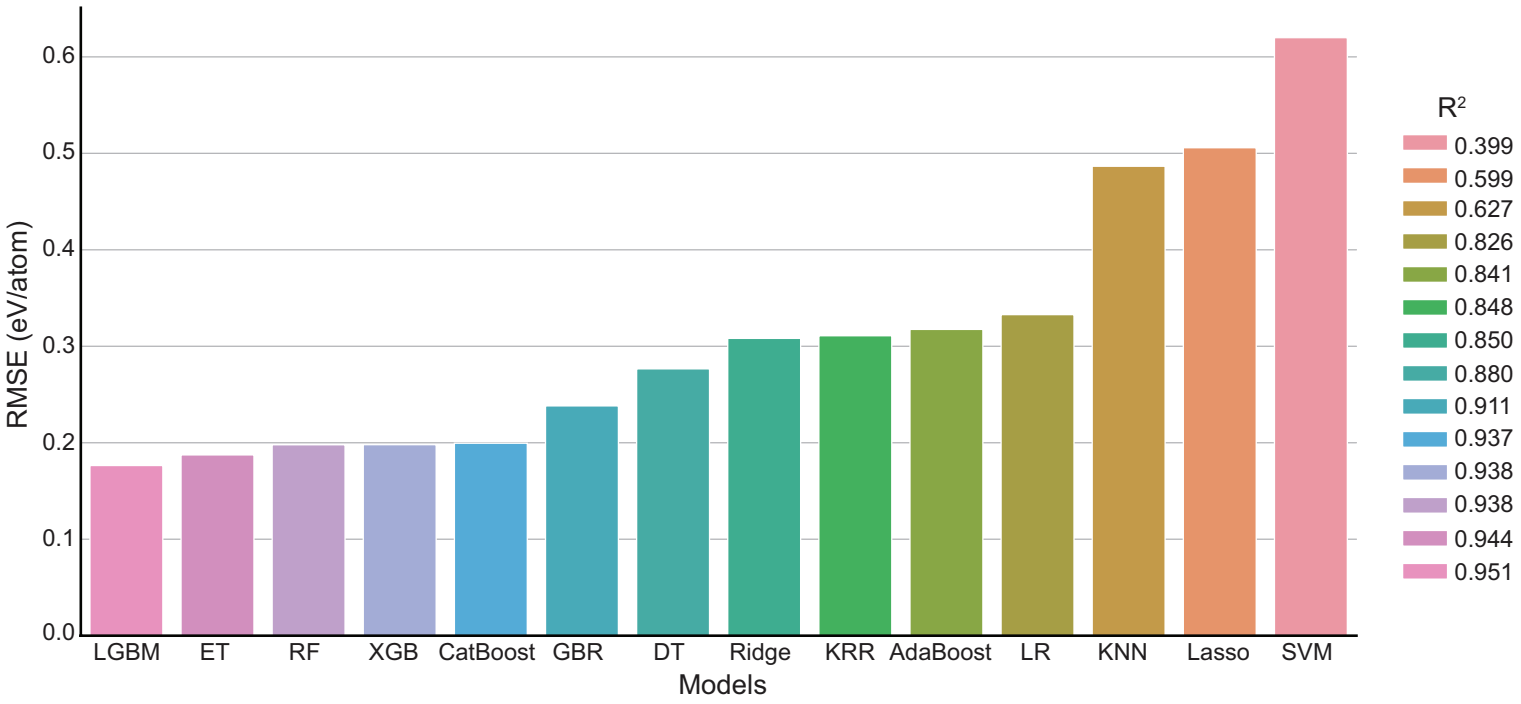
Symbol	Feature name	Feature type
$\overline{\Delta\chi}$	Mean electronegativity difference	Elemental
$\bar{\chi}$	Mean electronegativity	Elemental
$\hat{\chi}$	Average deviation in electronegativity	Elemental
χ^{min}	Minimum electronegativity	Elemental
$\Delta\chi^{min}$	Minimum electronegativity difference	Elemental
$\sigma(\Delta\chi)$	Standard deviation of electronegativity difference	Elemental
$\delta\chi^{min}_{shell=1}$	Minimum electronegativity difference in 1st shell neighbors	Structural
$\hat{MN}$	Average deviation in Mendeleev Number	Elemental
$\overline{MN}$	Mean Mendeleev Number	Elemental
T_M^{max}	Maximum melting point	Elemental
$\overline{T_M}$	Mean melting point	Elemental
$\hat{T_M}$	Average deviation in melting point	Elemental
T_M^{range}	Melting point range	Elemental
IC^{max}	Maximum ionic character	Elemental
$\overline{IC}$	Mean ionic character	Elemental
$\overline{EA}_A$	Mean electron affinity of anions	Elemental
$\bar{P}$	Mean group	Elemental
$\hat{P}$	Average deviation in group	Elemental
A^{min}	Minimum atomic mass	Elemental
Z^{mode}	Mode of atomic numbers	Elemental
$\hat{N}_{p,V}$	Average deviation in number of p-electrons (valence)	Elemental
$\bar{N}_U$	Mean number of unfilled electrons	Elemental
N_V^{min}	Minimum number of valence electrons	Elemental
$\delta\overline{N}_{p,V\,shell=1}$	Mean difference in number of valence p-electrons in 1st shell neighbors	Structural
$\overline{E_g}$	Mean bandgap	Elemental
$\delta E^{max}_{g,shell=1}$	Maximum difference in band gap in 1st shell neighbors	Structural
$\bar{\alpha}(CN=4)$	Mean weighted ordering parameter for atoms having CN = 4	Structural
$\bar{\alpha}(CN=6)$	Mean weighted ordering parameter for atoms having CN = 4	Structural
PF	Packing fraction	Structural
r_{cov}^{max}	Maximum covalent radius	Elemental
V^{range}	Range in volume of unit cell per atom	Elemental
V^{min}	Minimum volume of unit cell per atom	Elemental
$\overline{SG}$	Mean space group number	Elemental
$GM(\eta)$	Geometric mean of local chemical hardness	Chemical hardness
η_L^{max}	Maximum chemical hardness of ligand	Chemical hardness
η_L^{min}	Minimum chemical hardness of ligand	Chemical hardness
$\overline{\Delta\eta}$	Mean chemical hardness difference	Chemical hardness



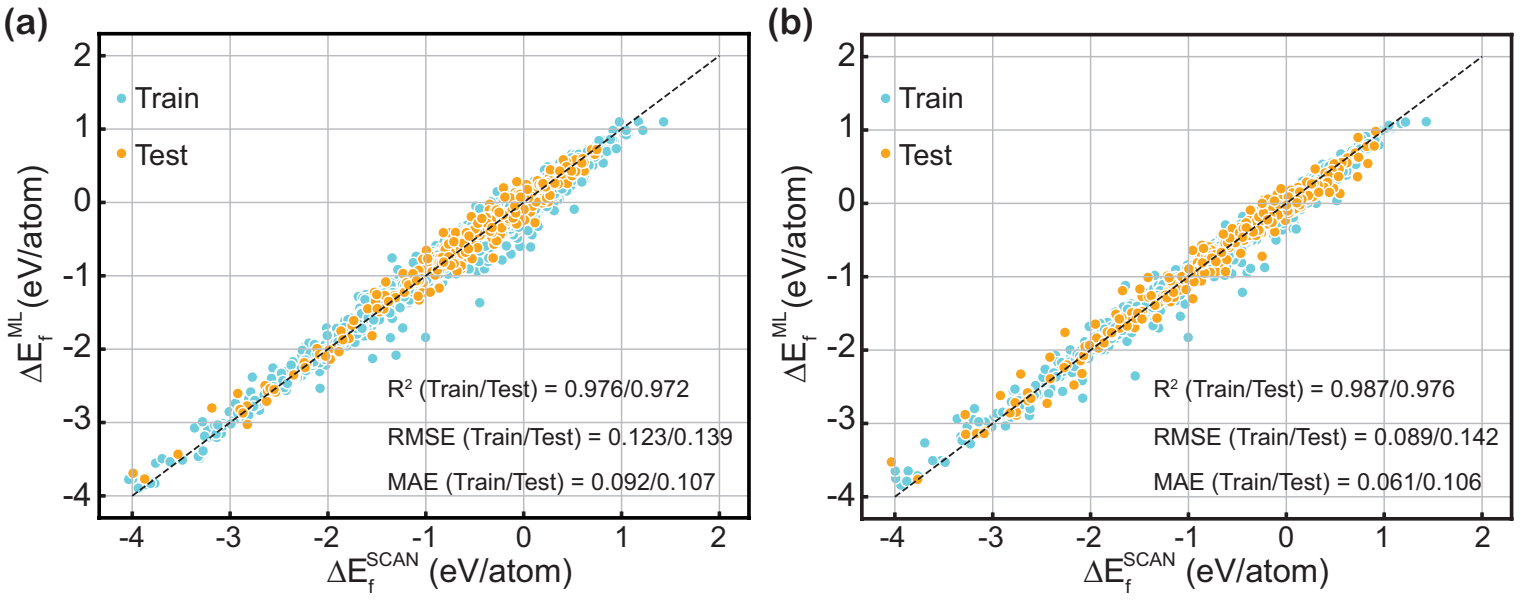
Supplementary Figure 3: Test RMSE and R^2 values (shown in legends) for ΔE_f regression corresponding to all ML algorithms using **{E}** feature set.



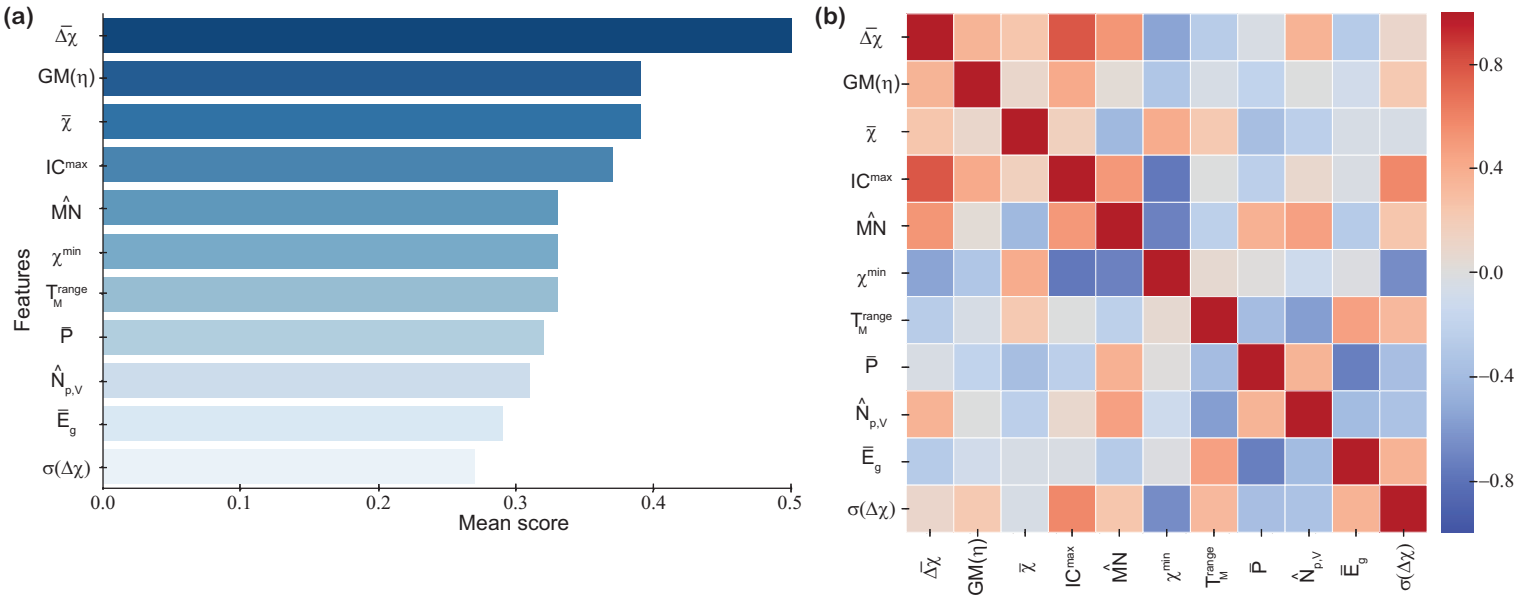
Supplementary Figure 4: (a) Mean score of best features selected from the **{E, S}** feature set (highly correlated features removed) for ΔE_f regression. (b) Pearson correlation between the best **{E, S}** features (after removing highly correlated features) selected from feature ranking.



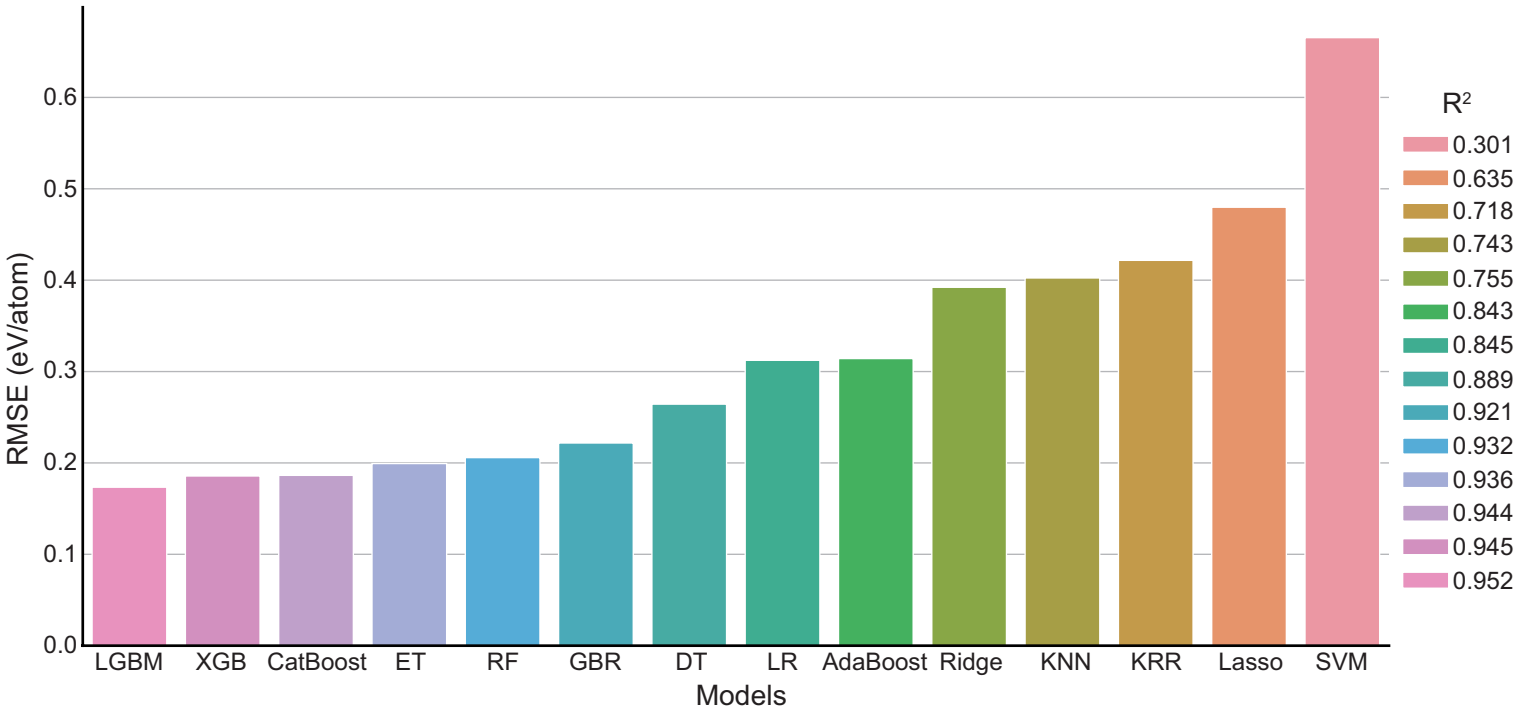
Supplementary Figure 5: Test RMSE and R^2 values (shown in legends) for ΔE_f regression corresponding to all ML algorithms using **{E, S}** feature set.



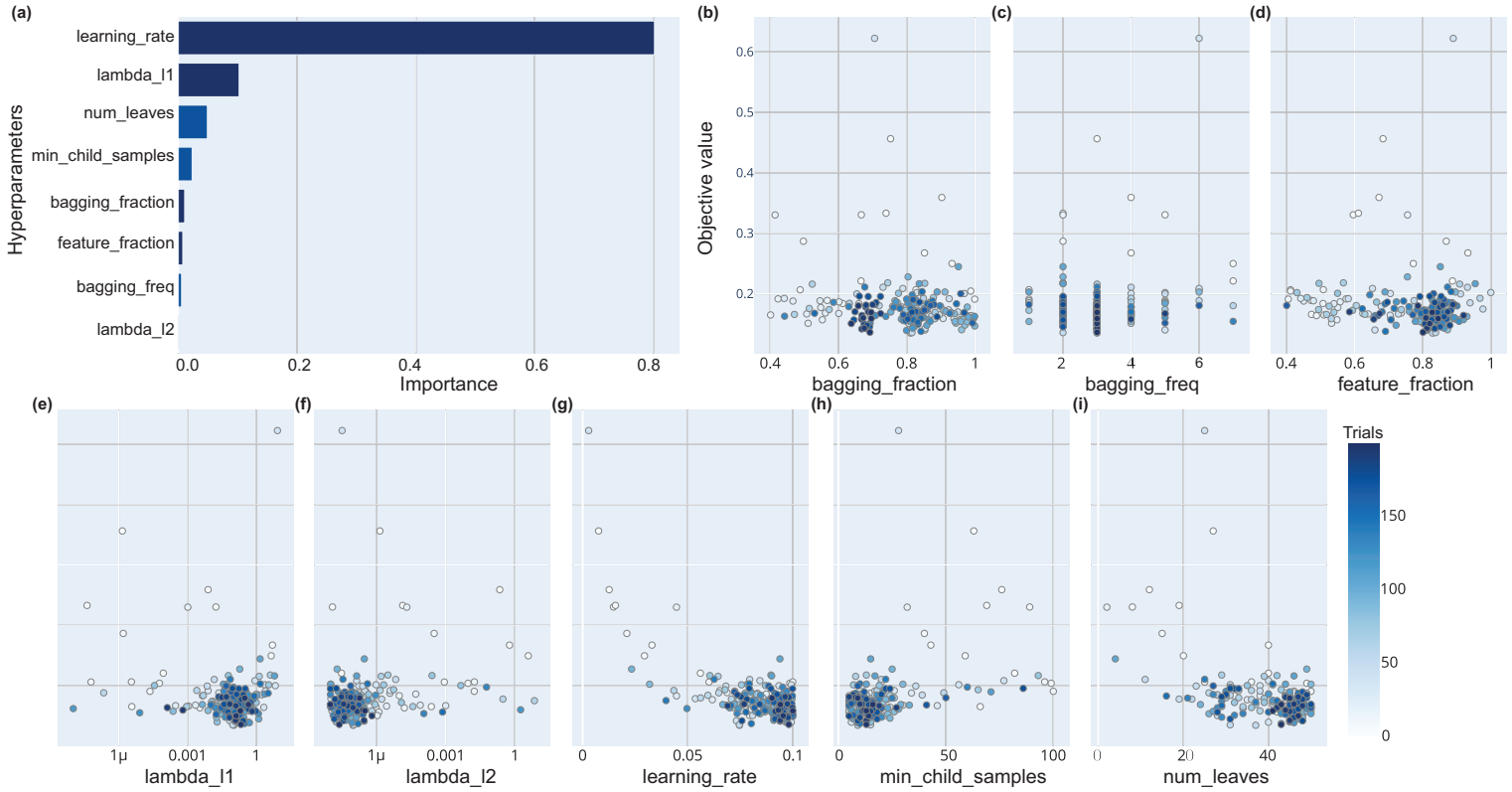
Supplementary Figure 6: Parity plots of DFT and ML(LGBM)-predicted formation energies using selected (a) **{E}** and (b) **{E, S}** features.



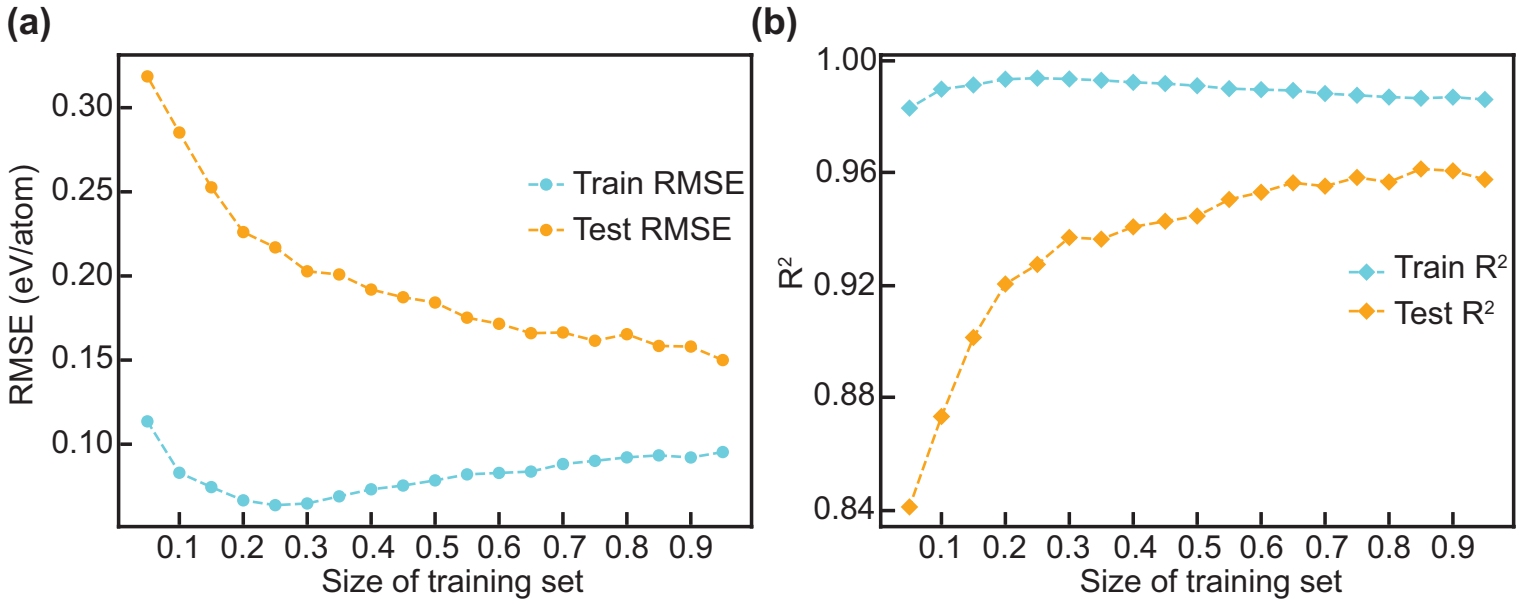
Supplementary Figure 7: (a) Mean score of best features selected from the $\{\mathbf{E}, \boldsymbol{\eta}\}$ feature set (highly correlated features removed) for ΔE_f regression. (b) Pearson correlation between the best $\{\mathbf{E}, \boldsymbol{\eta}\}$ features (after removing highly correlated features) selected from feature ranking.



Supplementary Figure 8: Test RMSE and R^2 values (shown in legends) for ΔE_f regression corresponding to all ML algorithms using $\{\mathbf{E}, \boldsymbol{\eta}\}$ feature set.



Supplementary Figure 9: (a) Hyperparameter importance plot for ML(LGBM) model corresponding to ΔE_f ($\{\mathbf{E}, \boldsymbol{\eta}\}$) regression. (b)-(i) Slice plots for bagging fraction, bagging frequency, feature fraction, λ_{L_1}, λ_{L_1}, learning rate, minimum child samples, and number of leaves hyperparameters for ML(LGBM) model corresponding to ΔE_{hull} ($\{\mathbf{E}, \boldsymbol{\eta}\}$) regression. The legend bar shows number of trials.



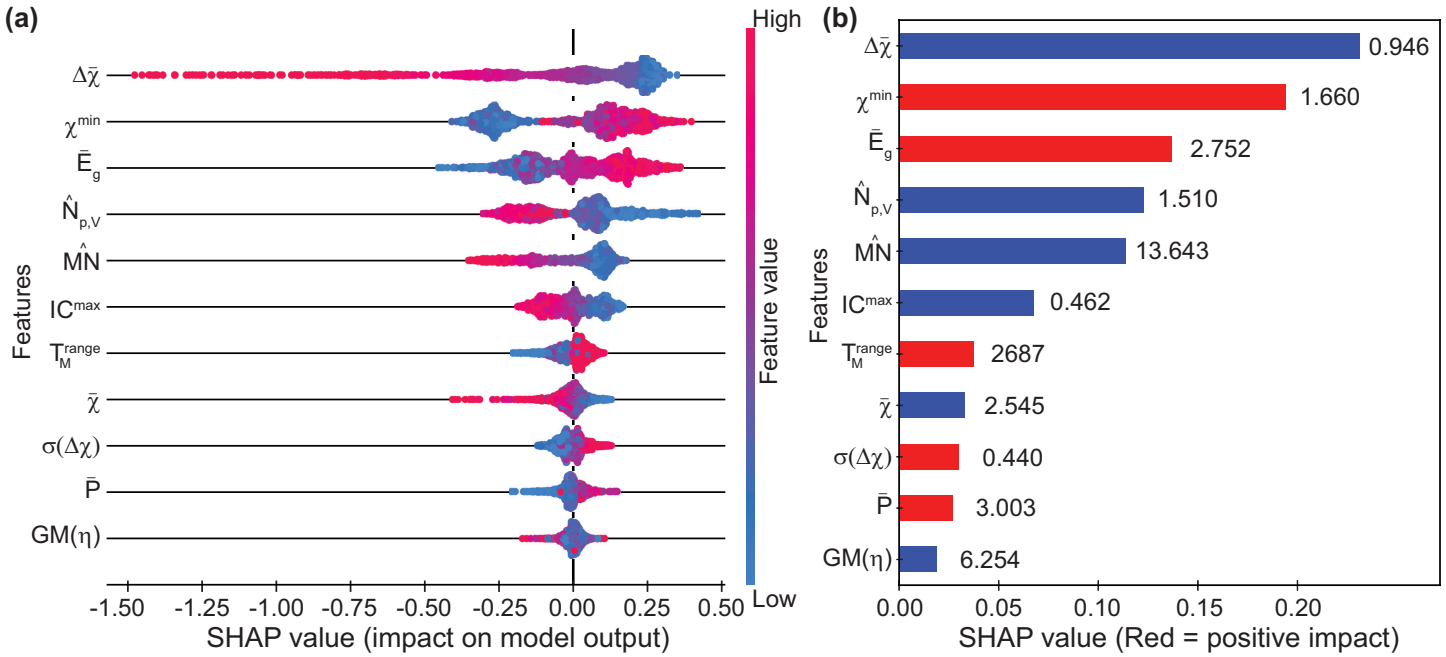
Supplementary Figure 10: Learning curves for the ML(LGBM)-predicted formation energies using the selected $\{\mathbf{E}, \boldsymbol{\eta}\}$ features with (a) RMSE and (b) R^2 as the performance metrics.

Supplementary Table 2: Effect of changing the number of features on the performance of $\Delta E_f^{ML}(\{\mathbf{E, S}\})$

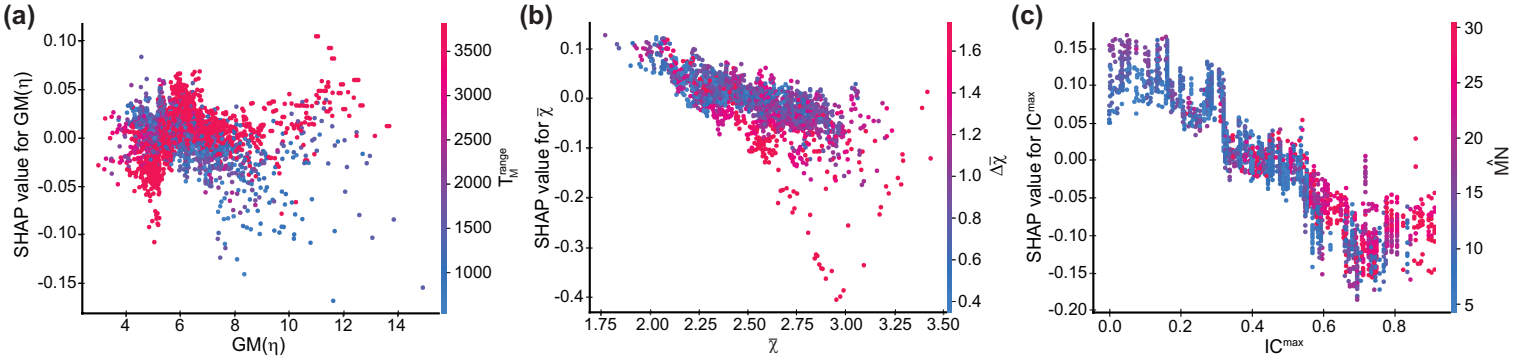
Number of features selected after feature ranking	Number of features selected after removing correlated features	Final features selected	Train RMSE	Test RMSE
10	9	$\overline{\Delta\chi}$, $GM(\eta)$, $\bar{\chi}$, IC^{max}, $\hat{MN}$, χ^{min}, T_M^{range}, $\bar{P}$, $\hat{N}_{p,V}$	0.106	0.146
11	10	$\overline{\Delta\chi}$, $GM(\eta)$, $\bar{\chi}$, IC^{max}, $\hat{MN}$, χ^{min}, T_M^{range}, $\bar{P}$, $\hat{N}_{p,V}$, $\bar{E}_g$	0.098	0.131
12	11	$\overline{\Delta\chi}$, $GM(\eta)$, $\bar{\chi}$, IC^{max}, $\hat{MN}$, χ^{min}, T_M^{range}, $\bar{P}$, $\hat{N}_{p,V}$, $\bar{E}_g$, $\sigma(\Delta\chi)$	0.097	0.131
13	11	$\overline{\Delta\chi}$, $GM(\eta)$, $\bar{\chi}$, IC^{max}, $\hat{MN}$, χ^{min}, T_M^{range}, $\bar{P}$, $\hat{N}_{p,V}$, $\bar{E}_g$, $\sigma(\Delta\chi)$	0.097	0.131
14	11	$\overline{\Delta\chi}$, $GM(\eta)$, $\bar{\chi}$, IC^{max}, $\hat{MN}$, χ^{min}, T_M^{range}, $\bar{P}$, $\hat{N}_{p,V}$, $\bar{E}_g$, $\sigma(\Delta\chi)$	0.097	0.131
15	11	$\overline{\Delta\chi}$, $GM(\eta)$, $\bar{\chi}$, IC^{max}, $\hat{MN}$, χ^{min}, T_M^{range}, $\bar{P}$, $\hat{N}_{p,V}$, $\bar{E}_g$, $\sigma(\Delta\chi)$	0.097	0.131
16	12	$\overline{\Delta\chi}$, $GM(\eta)$, $\bar{\chi}$, IC^{max}, $\hat{MN}$, χ^{min}, T_M^{range}, $\bar{P}$, $\hat{N}_{p,V}$, $\bar{E}_g$, $\sigma(\Delta\chi)$, $\bar{SG}$	0.096	0.134
17	13	$\overline{\Delta\chi}$, $GM(\eta)$, $\bar{\chi}$, IC^{max}, $\hat{MN}$, χ^{min}, T_M^{range}, $\bar{P}$, $\hat{N}_{p,V}$, $\bar{E}_g$, $\sigma(\Delta\chi)$, $\bar{SG}$, $\hat{P}$	0.095	0.133
18	13	$\overline{\Delta\chi}$, $GM(\eta)$, $\bar{\chi}$, IC^{max}, $\hat{MN}$, χ^{min}, T_M^{range}, $\bar{P}$, $\hat{N}_{p,V}$, $\bar{E}_g$, $\sigma(\Delta\chi)$, $\bar{SG}$, $\hat{P}$	0.095	0.133
19	14	$\overline{\Delta\chi}$, $GM(\eta)$, $\bar{\chi}$, IC^{max}, $\hat{MN}$, χ^{min}, T_M^{range}, $\bar{P}$, $\hat{N}_{p,V}$, $\bar{E}_g$, $\sigma(\Delta\chi)$, $\bar{SG}$, $\hat{P}$, $\hat{N}_U$	0.094	0.132
20	15	$\overline{\Delta\chi}$, $GM(\eta)$, $\bar{\chi}$, IC^{max}, $\hat{MN}$, χ^{min}, T_M^{range}, $\bar{P}$, $\hat{N}_{p,V}$, $\bar{E}_g$, $\sigma(\Delta\chi)$, $\bar{SG}$, $\hat{P}$, $\hat{N}_U$, $\hat{\mu}_B$	0.092	0.130

Supplementary Table 3: Comparison of formation energies predicted by different sets of features

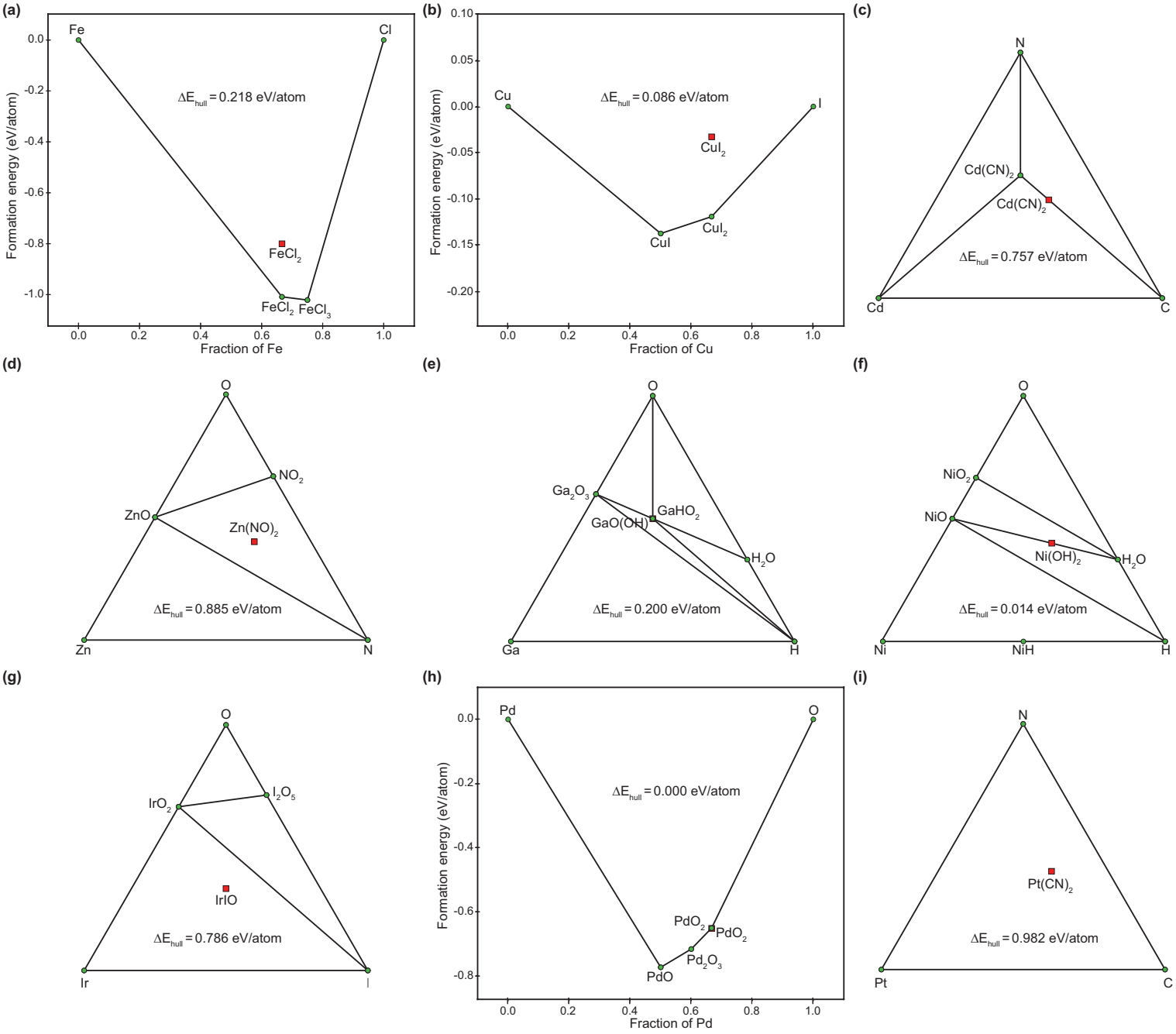
2D material	ΔE_f^{SCAN} [eV/atom]	$\Delta E_f^{ML}(\{\mathbf{E}\})$ [eV/atom]	$\Delta E_f^{ML}(\{\mathbf{E, S}\})$ [eV/atom]	$\Delta E_f^{ML}(\{\mathbf{E, \eta}\})$ [eV/atom]
$Cd(CN)_2$	0.738	0.671	0.628	0.701
$Cd(NC)_2$	0.676	0.671	0.656	0.694
$Ba(NCO)_2$	-1.080	-0.937	-1.069	-0.990
$Ba(OCN)_2$	-0.813	-0.937	-0.849	-0.924
La(CN)(NH)	-0.526	-0.568	-0.547	-0.569
La(NC)(NH)	-0.594	-0.568	-0.584	-0.583
AlO(NCS)	-0.959	-0.708	-0.781	-0.749
AlO(SCN)	-0.615	-0.708	-0.545	-0.710
FeSe(CN)	0.466	0.348	0.454	0.376
FeSe(NC)	0.481	0.348	0.414	0.381
Mg(NCO)(NCS)	-0.616	-0.365	-0.607	-0.413
Mg(OCN)(NCS)	-0.430	-0.365	-0.454	-0.373
Mg(NCO)(SCN)	-0.403	-0.365	-0.403	-0.397
Mg(OCN)(SCN)	-0.198	-0.365	-0.174	-0.363



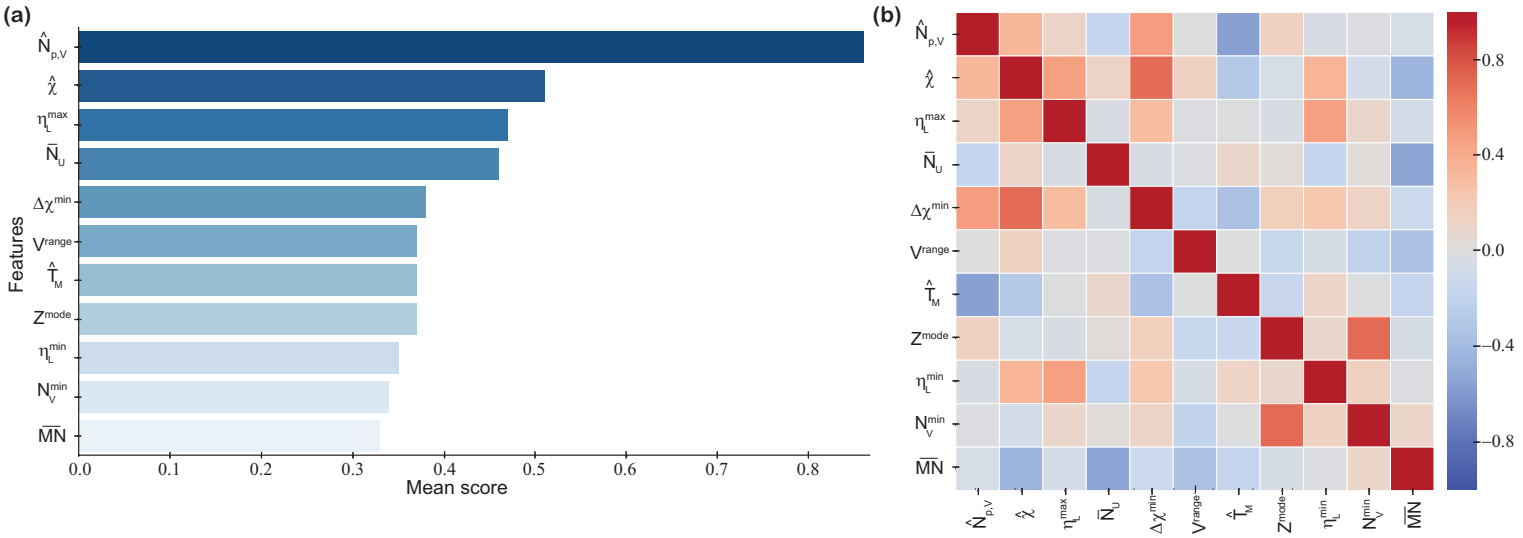
Supplementary Figure 11: (a) SHAP feature importance plot for $\Delta E_f^{ML}(\{\mathbf{E}, \boldsymbol{\eta}\})$ (LGBM model). (b) Simplified version of the SHAP feature importance plot, where features with red and blue bars denote overall positive and negative impacts on ΔE_f^{ML}, respectively. The numbers beside each bar denote the average value of the corresponding feature in the train dataset. The features are arranged in the descending order of their importance.



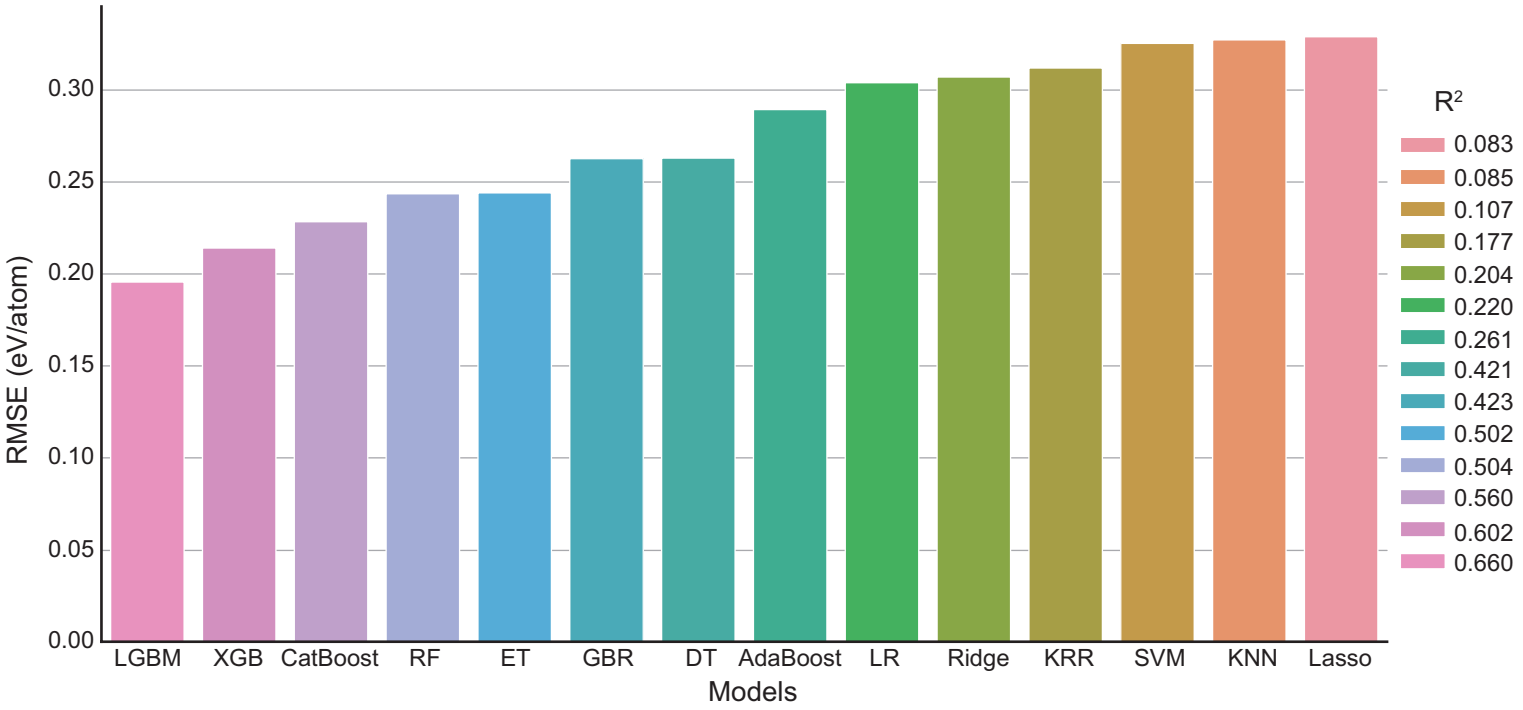
Supplementary Figure 12: SHAP dependence plots for (a) $GM(\eta)$, (b) $\bar{\chi}$, and (c) IC^{max} features for $\Delta E_f^{ML}(\{\mathbf{E}, \boldsymbol{\eta}\})$ (LGBM model).



Supplementary Figure 13: Convex hull constructions for 2D (a) $FeCl_2$, (b) CuI_2, (c) $Cd(CN)_2$, (d) $Zn(NO)_2$, (e) GaO(OH), (f) $Ni(OH)_2$, (g) IrIO, (h) PdO_2, and (i) $Pt(CN)_2$ compounds. The 2D and bulk compounds are depicted by red squares and green circles, respectively. The formation energies of bulk elements in their standard states are taken to be zero.



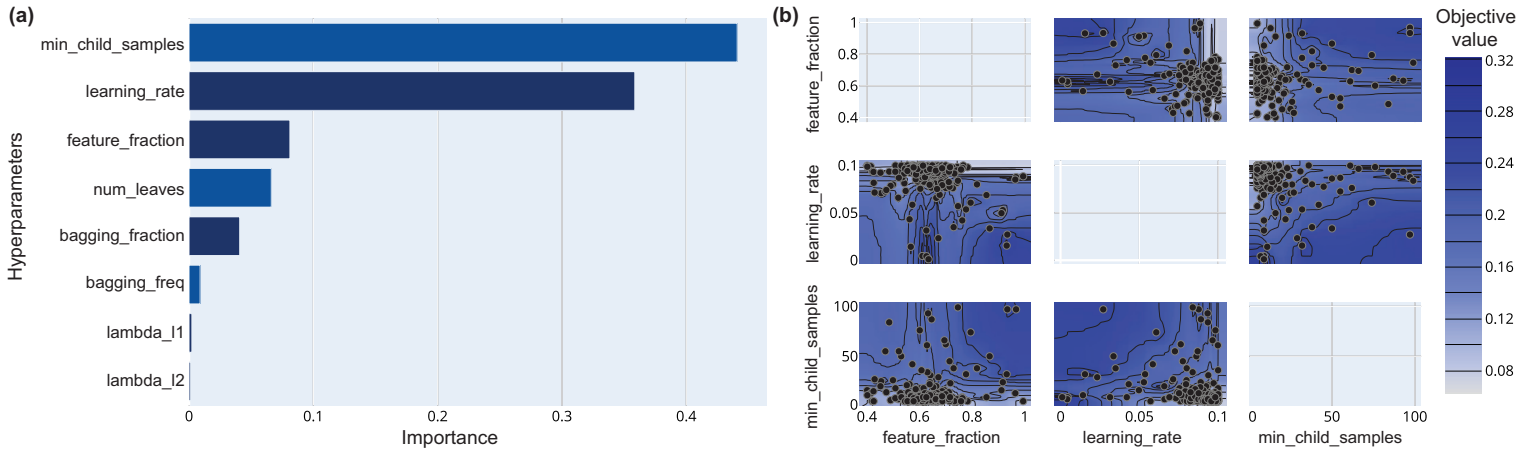
Supplementary Figure 14: (a) Mean score of best features selected from the $\{\mathbf{E}, \boldsymbol{\eta}\}$ feature set (highly correlated features removed) for ΔE_{hull} regression. (b) Pearson correlation between the best $\{\mathbf{E}, \boldsymbol{\eta}\}$ features (after removing highly correlated features) selected from feature ranking.



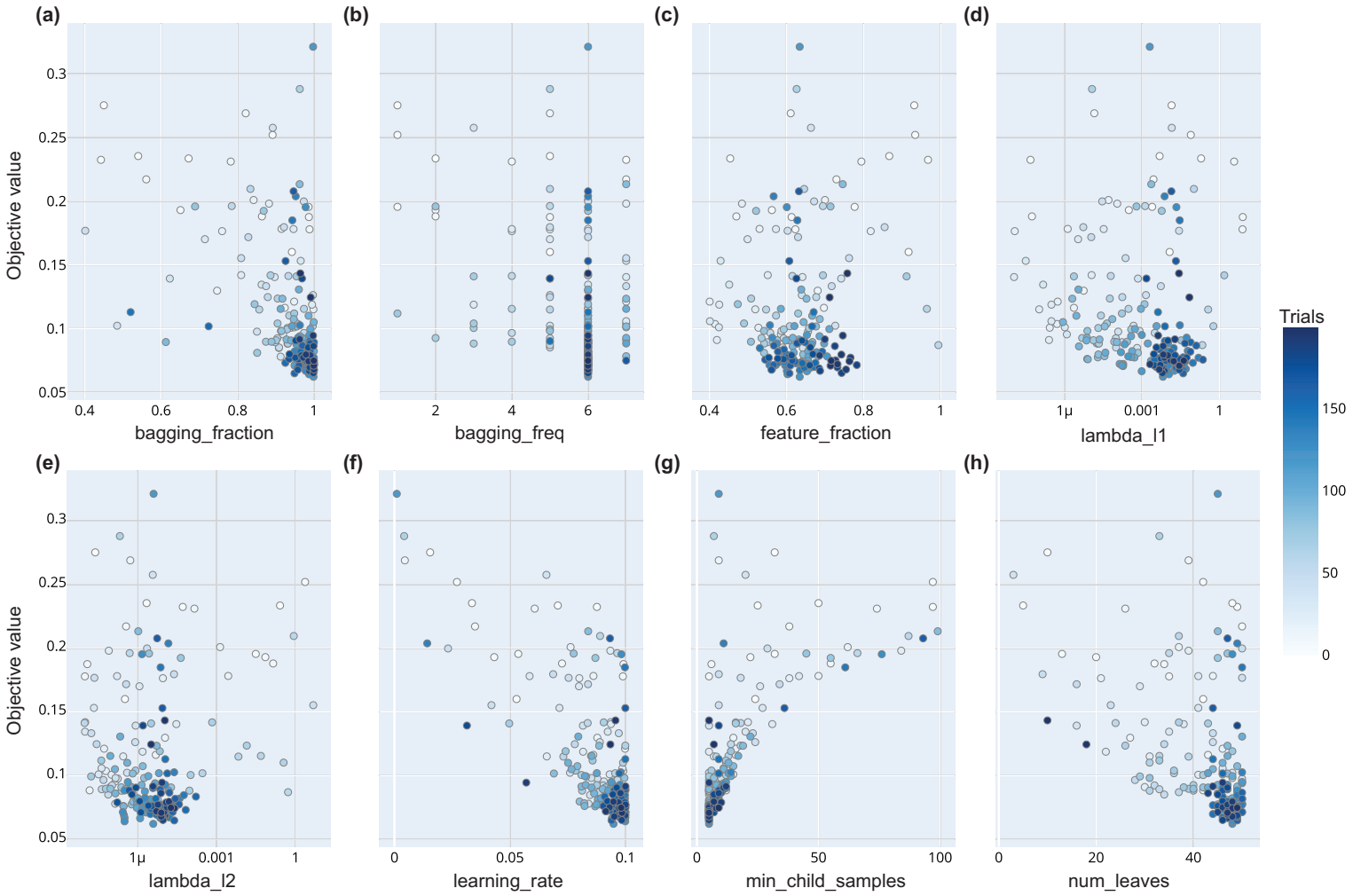
Supplementary Figure 15: Test RMSE and R^2 values (shown in legends) for ΔE_{hull} regression corresponding to all ML algorithms.

Supplementary Table 4: Number of training (90% of total data) samples in each class before and after applying oversampling using SMOTE

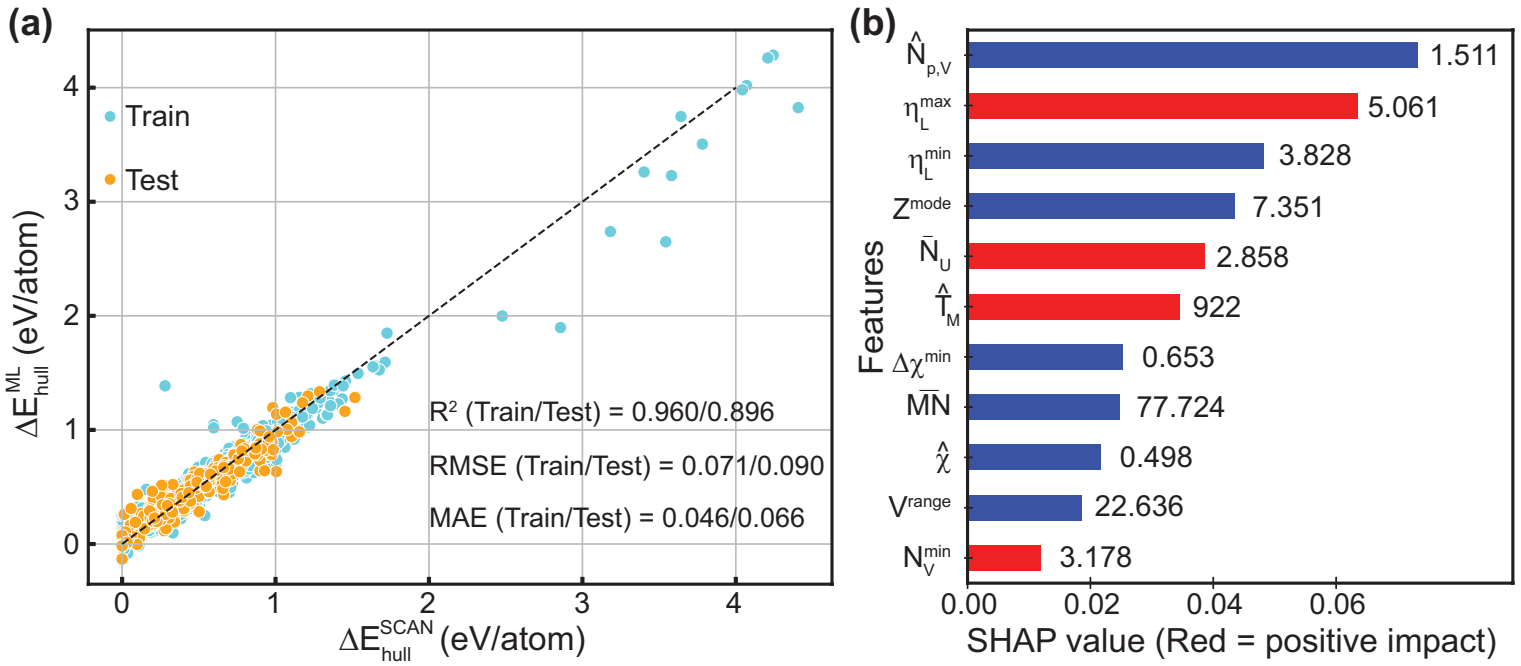
	Hh	**Hm**	**Hl**	**Mh**	**Mm**	**Ml**	**Lh**	**Lm**	**Ll**
Before	103	329	50	139	1372	410	7	206	173
After	103	329	100	139	1372	410	100	206	173



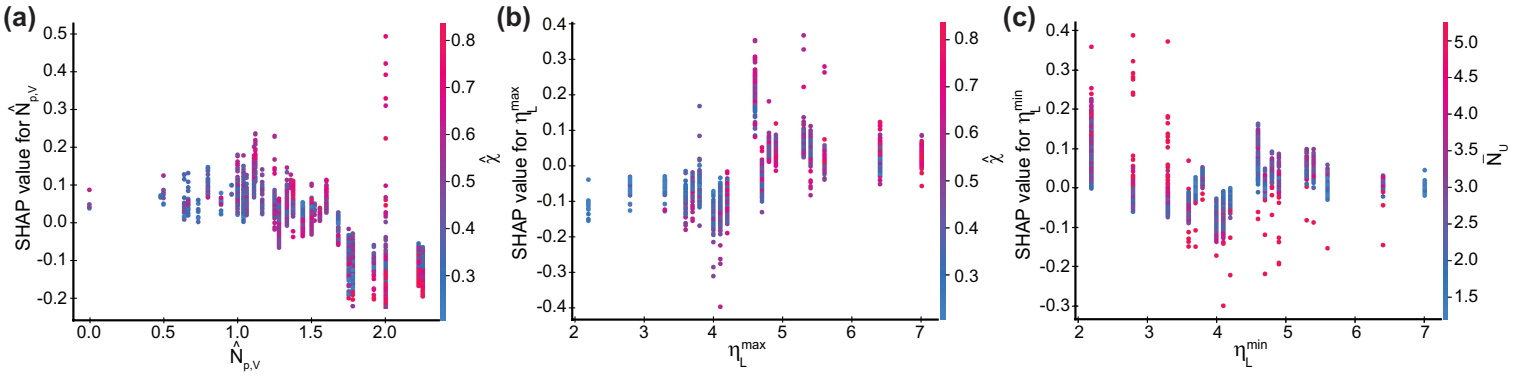
Supplementary Figure 16: (a) Hyperparameter importance plot for ML(LGBM) model corresponding to ΔE_{hull} regression. (b) Contour plot for minimum child samples, learning rate, and feature fraction hyperparameters utilized in the ML(LGBM) model corresponding to ΔE_{hull} regression.



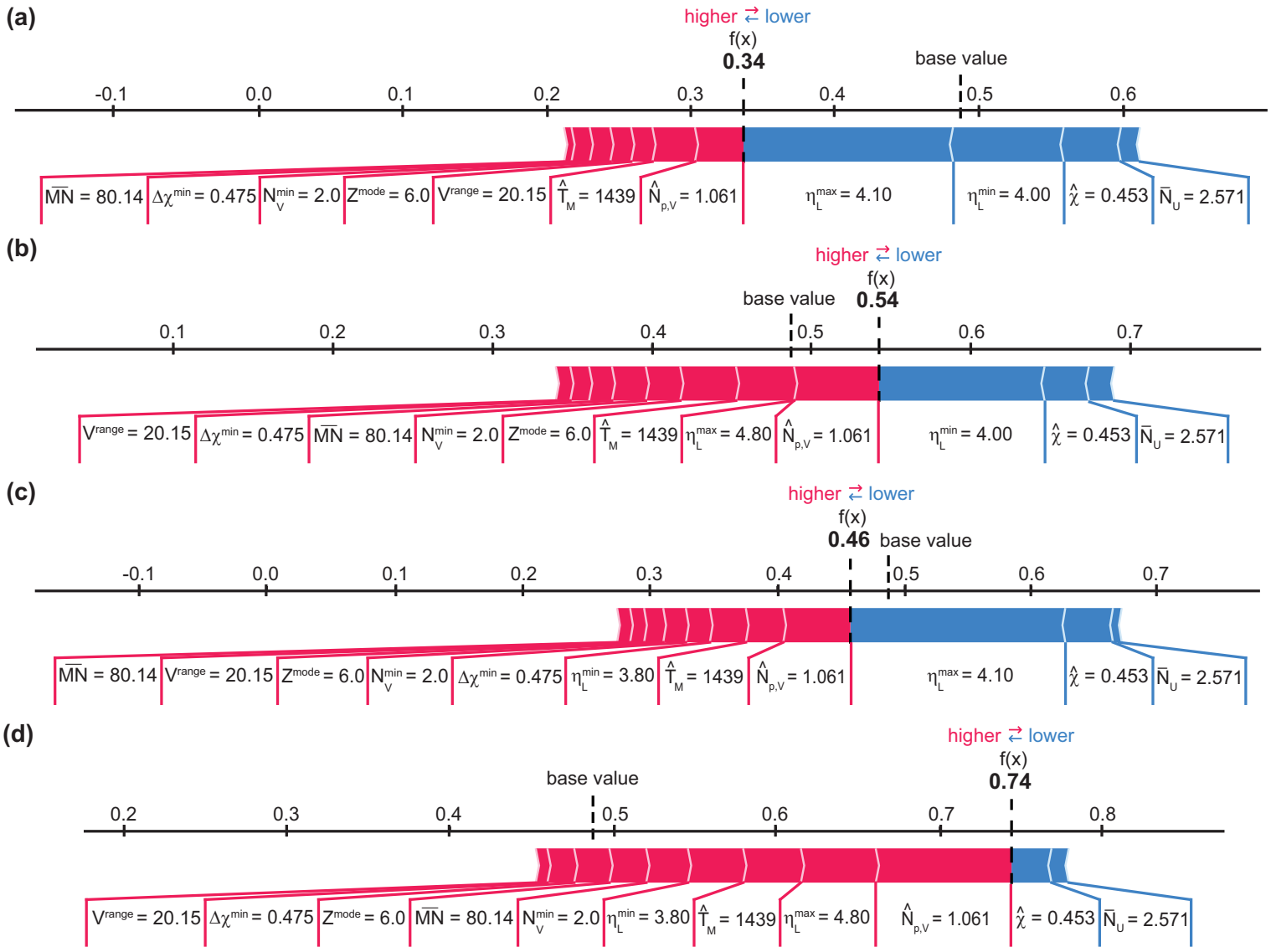
Supplementary Figure 17: (a)-(h) Slice plots for bagging fraction, bagging frequency, feature fraction, λ_{L_1}, λ_{L_1}, learning rate, minimum child samples, and number of leaves hyperparameters for the ML(LGBM) model utilized for ΔE_{hull} regression. The legend bar shows number of trials.



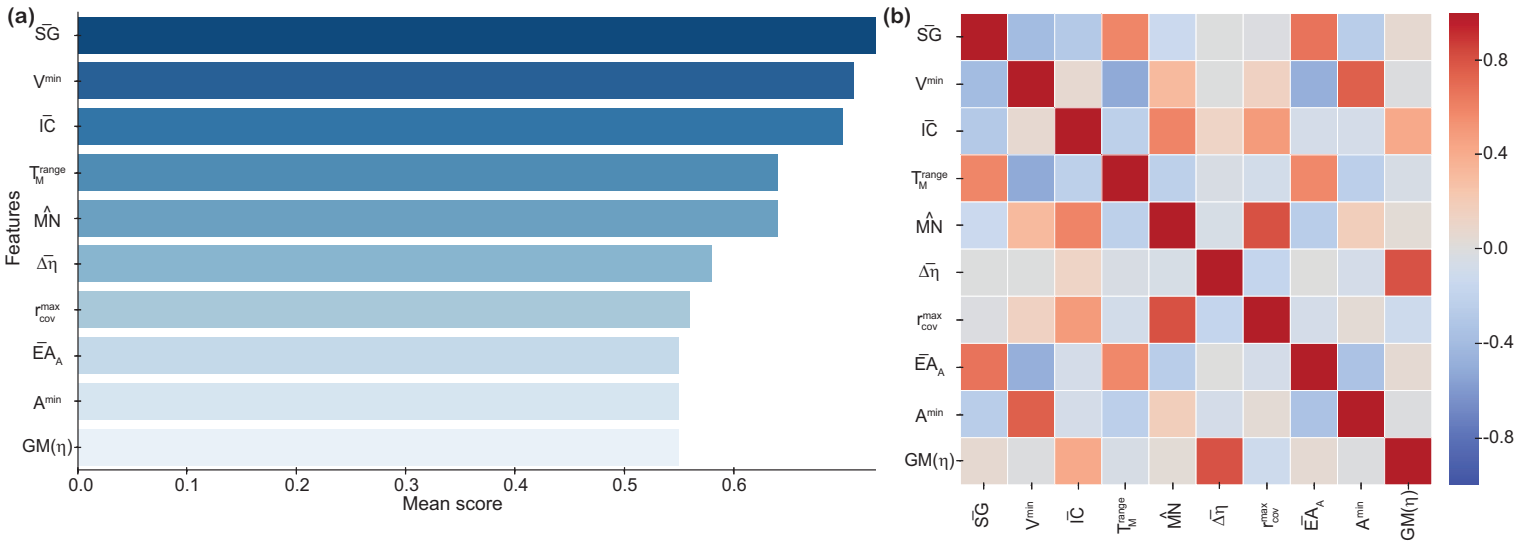
Supplementary Figure 18: (a) Parity plot of DFT and ML(LGBM)-predicted ΔE_{hull}. (b) Simplified version of the SHAP feature importance plot, where features with red and blue bars denote overall positive and negative impacts on ΔE_{hull}^{ML}, respectively. The numbers beside each bar denote the average value of the corresponding feature in the train dataset. The features are arranged in the descending order of their importance.



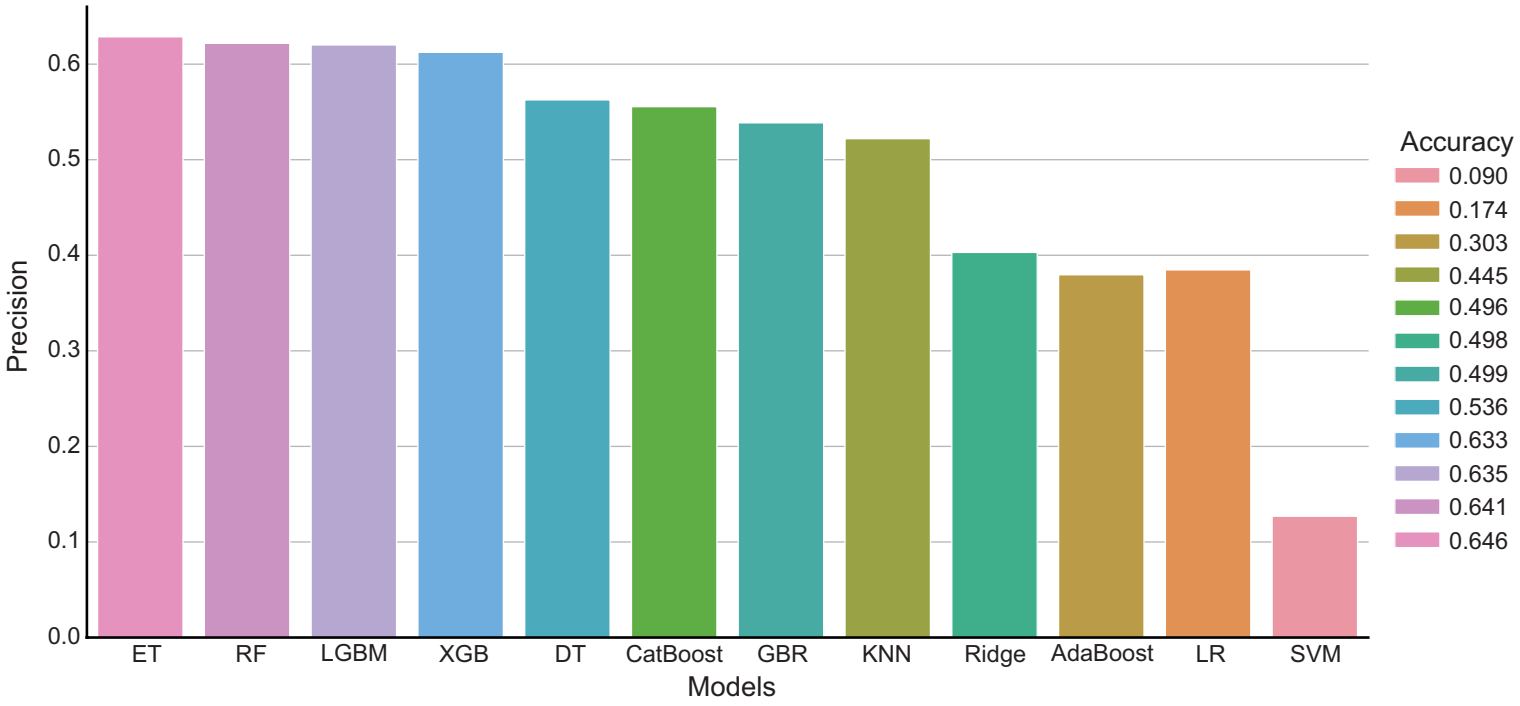
Supplementary Figure 19: SHAP dependence plots for (a) $\hat{N}_{p,V}$, (b) η_L^{max}, and (c) η_L^{min} features for the ΔE_{hull}^{ML} (LGBM model).



Supplementary Figure 20: Individual SHAP plots using the $\Delta E_{hull}^{ML}(\{\mathbf{E}, \boldsymbol{\eta}\})$ for (a) Mg(NCS)(NCO), (b) Mg(NCS)(OCN), (c) Mg(SCN)(NCO), and (d) Mg(SCN)(OCN) 2D compounds. The base value is 0.488 eV/atom.



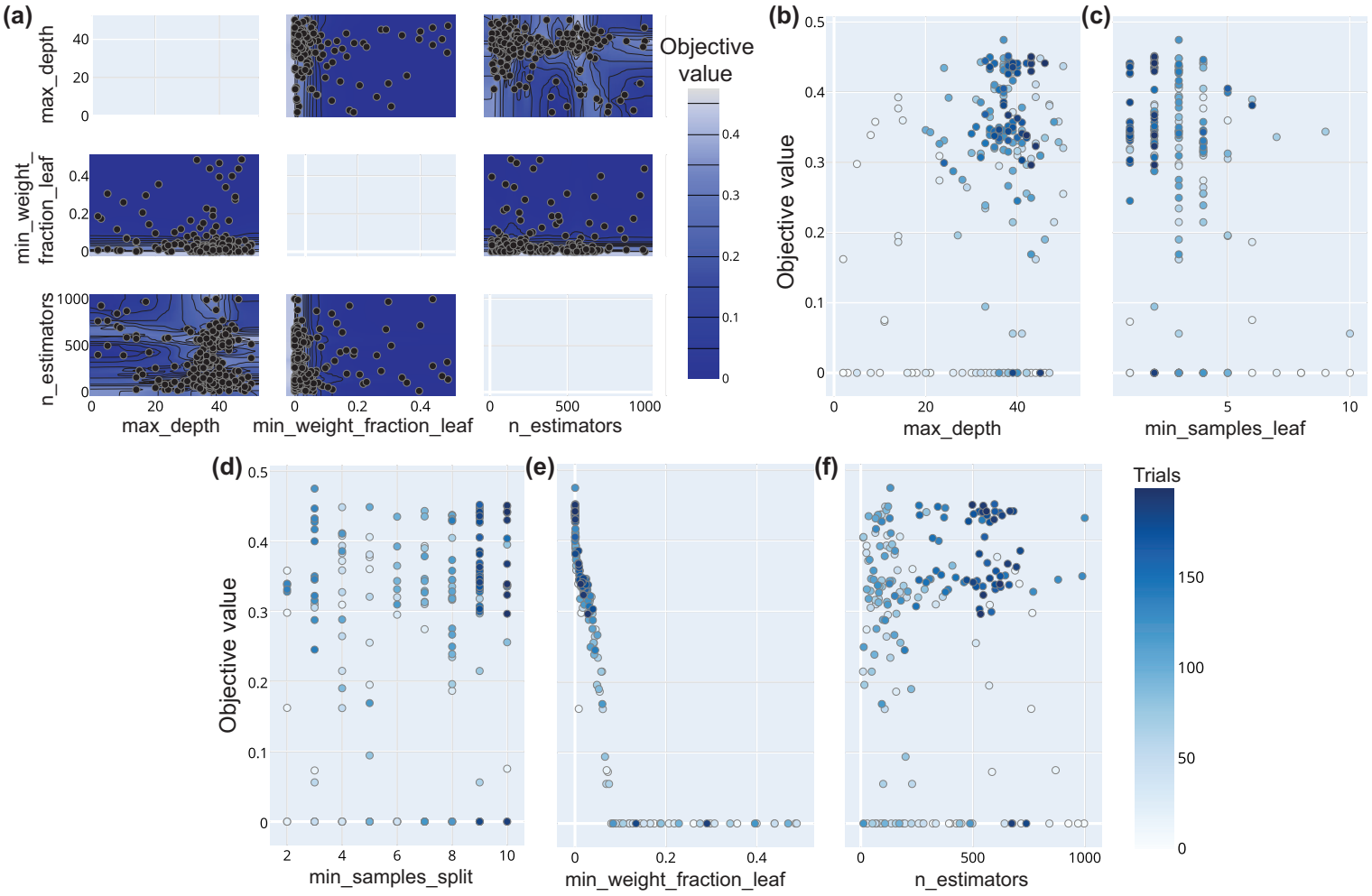
Supplementary Figure 21: (a) Mean score of best features selected from the $\{\mathbf{E}, \eta\}$ feature set (highly correlated features removed) for multiclass classification of overall stability. (b) Pearson correlation between the best $\{\mathbf{E}, \eta\}$ features (after removing highly correlated features) selected from feature ranking.



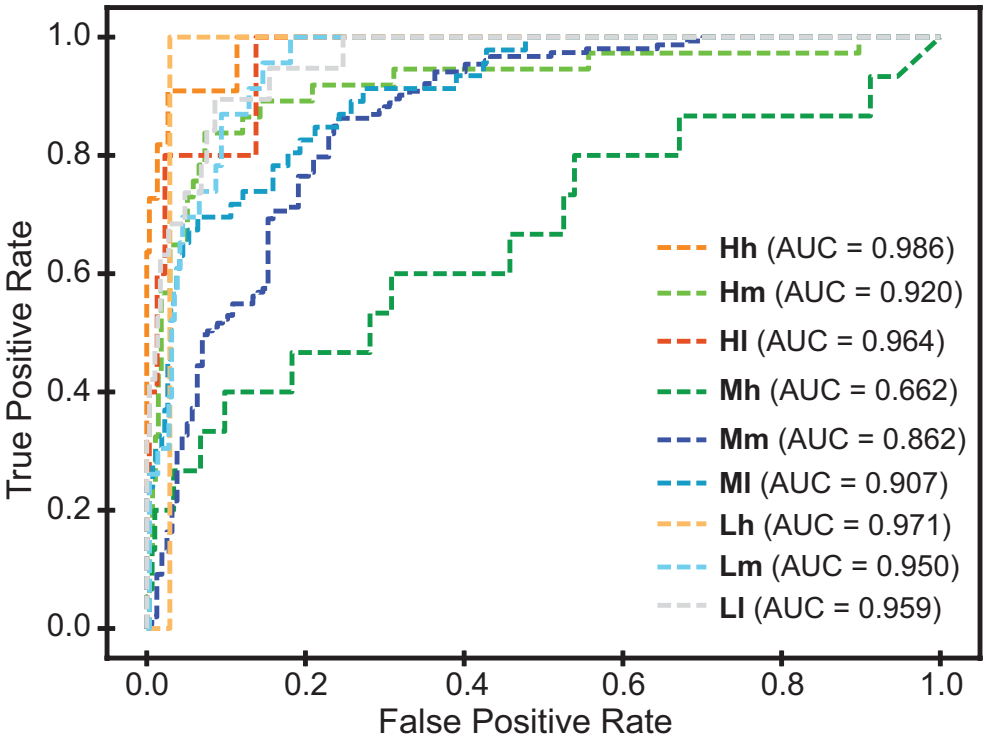
Supplementary Figure 22: Precision and accuracy values (shown in legends) for multiclass classification of overall stability corresponding to all ML algorithms.

Supplementary Note 5: Hyperparameter Optimization for Multiclass Classification of Overall Stability

The contour plot obtained after Bayesian hyperparameter optimization for the ML(ET) model is shown in Figure 23(a) for three hyperparameters – maximum depth, minimum weight fraction of leaf, and number of estimators. The regions yielding highest test MCC scores are the ones in which minimum weight fraction of leaf and number of estimators are less than zero and around 500, respectively, for a wide range of maximum depth values (10 to 50). The slice plots for all the hyperparameters are shown in Figures 23(b) to 23(f), in which the objective values (test MCC scores) are shown as a function of the hyperparameter values and number of trials.



Supplementary Figure 23: (a) Contour plot for the maximum depth, minimum weight fraction of leaf, and number of estimators hyperparameters utilized in the ML(ET) model corresponding to multiclass classification of overall stability. (b)-(f) Slice plots for the maximum depth, minimum samples at a leaf node, minimum samples required for splitting, minimum weight fraction of leaf, and number of estimators hyperparameters for the ML(ET) model. The legend bar shows number of trials.



Supplementary Figure 24: The ROC-AUC plot obtained for multiclass classification of overall stability using the ML(ET) model.

Supplementary Table 5: Number of correctly predicted samples in each class when optimized by different performance metrics for train data (90% of total data)

Metric	True **Hh**	True **Hm**	True **Hl**	True **Mh**	True **Mm**	True **Ml**	True **Lh**	True **Lm**	True **Ll**
Accuracy	100	202	52	111	1373	279	7	179	149
Precision	109	233	52	136	1371	317	8	183	158
F_1	105	209	53	121	1365	283	8	182	158
MCC	108	237	53	136	1369	351	7	195	162

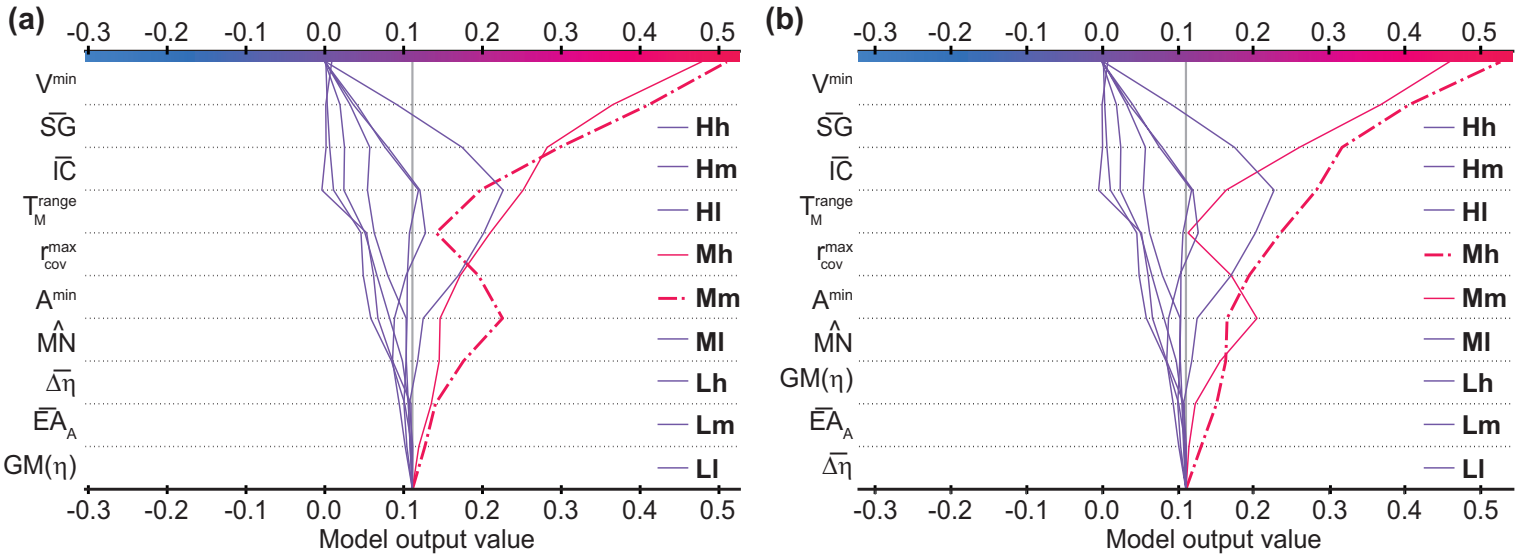
Supplementary Table 6: Performance metrics for each class predicted by the best ML(ET) model

Metric	True **Hh**	True **Hm**	True **Hl**	True **Mh**	True **Mm**	True **Ml**	True **Lh**	True **Lm**	True **Ll**
Precision	0.69	0.89	0.61	0.65	0.90	0.90	0.54	0.77	0.79
Recall	0.95	0.65	0.96	0.88	0.90	0.77	0.88	0.85	0.84
F_1	0.80	0.75	0.75	0.75	0.90	0.83	0.67	0.81	0.82

Supplementary Note 6: SHAP Multioutput Decision Plot

- Both the x-axes represent the model's output in the form of probabilites.
- The plot is centered on the lower x-axis at the base value of 0.11 (=1/9; equal probability taken initially for each of the nine classes). All the SHAP values are relative to this expected value.
- The y-axis lists the model's features. By default, the features are sorted in the descending order of their importance.

- The prediction of each class for an observation is represented by a colored line. At the top of the plot, each line strikes the x-axis at its corresponding predicted value. The class with the highest predicted value at the top x-axis is the class to which the ML model classifies the observation, shown by a dashed line.
- Moving from the bottom of the plot to the top, SHAP values for each feature are added to the model's base value. This shows how each feature contributes to the overall prediction.

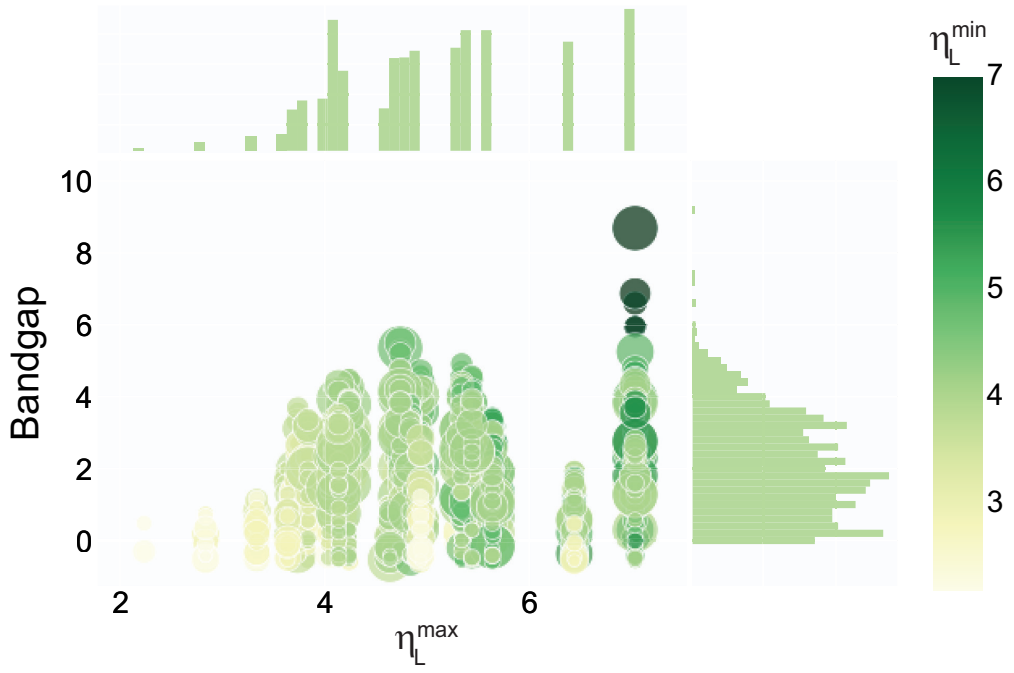


Supplementary Figure 25: SHAP multioutput decision plots for (a) $Ba(CN)_2$ and (b) $Ba(NC)_2$ using the ML(ET) model.

Supplementary Note 7: Determination of E in the STH expression

It is assumed that cocatalysts would be needed to overcome the water-splitting overpotentials. Previous reports have shown that the required overpotentials for OER and HER are below 0.5 V and 0.1 V, respectively, by utilizing several cocatalysts[5,6]. However, energy loss during carrier migration between different materials is to be expected, therefore we assumed required overpotentials for OER and HER to be 0.6 and 0.2 V, respectively[7]. Then the expression for E will be given by:

$$E = \begin{cases} E_g, & (\Delta E_O \geqslant 0.6\text{eV}, \Delta E_R \geqslant 0.2\text{eV}) \\ E_g + 0.6 - \Delta E_O, & (\Delta E_O < 0.6\text{eV}, \Delta E_R \geqslant 0.2\text{eV}) \\ E_g + 0.2 - \Delta E_R, & (\Delta E_O \geqslant 0.6\text{eV}, \Delta E_R < 0.2\text{eV}) \\ E_g + 0.8 - \Delta E_O - \Delta E_R, & (\Delta E_O < 0.6\text{eV}, \Delta E_R < 0.2\text{eV}) \end{cases}$$



Supplementary Figure 26: Scatter-histogram plot for (PBE) band gaps of all (2176) semiconductors in the 2DO database as a function of η_L^{max} (x-axis), η_L^{min}, and η_M. The color bar and circle-sizes are shown according to the η_L^{min} and η_M values, respectively.

Supplementary Table 7: η_{STH} values for 2D materials filtered out from HT-screening

2DO compound	E_g^{GW} [eV]	$\Delta\phi$ [eV]	ΔE_R [eV]	ΔE_O [eV]	η_{STH} (%)
BiSBr	2.97	0.01	0.14	1.61	2.13
BiSCl	3.13	0.27	1.41	0.49	1.20
BiSeBr	2.46	0.32	0.29	0.93	8.06
BiSeCl	2.58	0.62	0.04	1.31	4.34
BiSeI	2.43	0.16	0.42	0.78	8.61
IrSeOH	2.95	2.58	3.98	0.32	1.15
$HfSe_2$	1.82	0.00	0.10	0.49	17.14
$PtSe_2$	2.22	0.00	0.47	0.52	11.09
$ZrSe_2$	1.85	0.00	0.12	0.50	17.14
PtSeTe	1.82	0.77	0.40	0.19	10.84
RhS(NCO)	3.08	1.20	1.27	0.58	1.63
RhS(OH)	2.88	3.30	4.30	0.65	2.88
PtSSe	2.50	0.69	0.29	0.98	7.39
ZrSSe	2.29	0.05	0.60	0.46	8.75
HfS_2	2.92	0.00	0.05	1.64	1.85
PtS_2	2.84	0.00	0.20	1.40	3.47
ZrS_2	2.89	0.00	0.05	1.61	2.03
BiTeBr	2.43	0.78	0.57	0.63	8.25
ScTeBr	2.89	0.66	1.32	0.34	1.45
IrTeCl	2.90	0.93	1.65	0.02	0.57
BiTeI	2.23	0.36	0.88	0.12	4.66

Supplementary Table 8: Comparison of PBE and GW band gaps with the experimental band gaps

2DO compound	E_g^{PBE} [eV]	E_g^{GW} [eV]	E_g^{expt} [eV]
PtS_2	1.71	2.84	1.60[8]
$PtSe_2$	1.31	2.22	2.10[9]
HfS_2	1.19	2.92	2.00[10]
SnS_2	1.70	3.04	2.29[11]

Supplementary References

1. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kègl, B. Algorithms for Hyper-Parameter Optimization. *25th Annual Conference on Neural Information Processing Systems* **2011**, 24, Neural Information Processing Systems Foundation.

2. Roth, A. E. Lloyd Shapley (1923–2016). *Nature* **2016**, 532, 178.

3. `https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html`

4. `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html`

5. McCrory, C. C.; Jung, S.; Ferrer, I. M.; Chatman, S. M.; Peters, J. C.; Jaramillo, T. F. Benchmarking Hydrogen Evolving Reaction and Oxygen Evolving Reaction Electrocatalysts for Solar Water Splitting Devices. *J. Am. Chem. Soc.* **2015**, 137, 4347–4357.

6. Zheng, Y.; Jiao, Y.; Jaroniec M.; Qiao, S. Z. Advancing the Electrochemistry of the Hydrogen-Evolution Reaction Through Combining Experiment and Theory. *Angew. Chem., Int. Ed.* **2015**, 54, 52–65.

7. Fu, C.-F.; Sun, J.; Luo, Q.; Li, X.; Hu, W.; Yang, J. Intrinsic Electric Fields in Two-Dimensional Materials Boost the Solar-to-Hydrogen Efficiency for Photocatalytic Water Splitting. *Nano Lett.* **2018**, 18, 6312–6317.

8. Zhao, Y., *et al*. Extraordinarily Strong Interlayer Interaction in 2D Layered PtS_2. *Adv. Mater.* **2016**, 28, 2399–2407.

9. Wang, Y., *et al*. Monolayer $PtSe_2$, a New Semiconducting Transition-Metal-Dichalcogenide, Epitaxially Grown by Direct Selenization of Pt. *Nano Lett.* **2015**, 15, 4013–4018.

10. Xu, K., *et al*. Ultrasensitive Phototransistors Based on Few-Layered HfS_2. Adv. Mater. **2015**, 27, 7881–7887.

11. Sun, Y., *et al*. Freestanding Tin Disulfide Single-Layers Realizing Efficient Visible-Light Water Splitting. *Angew. Chem. Int. Ed.* **2012**, 51, 8727–8731.