# CS229 Final Report: Learning Chemistry from Moment to Moment

Colin Dickens and Allegra Latimer

December 11th, 2018

## I. Introduction

Density functional theory (DFT) is a quantum mechanical modelling method that facilitates the prediction of ground state energies of atomic systems, thereby yielding valuable insights into many important chemical and physical systems and processes. DFT is particularly valuable in the field of heterogeneous catalysis, where we are often interested in the strength of chemical bonds between a catalyst surface, e.g. metallic platinum, and a molecular fragment, e.g. $CH_2$. Once bound, we refer to this molecular fragment as an adsorbate.

The energy released by bringing these two species together to bond, termed the binding energy, is a function of the electronic structure of the surface and the molecular fragment. For transition metal surfaces, simple chemical bonding theory has been used to demonstrate the existence of a linear correlation between the first moment of the projected density of electronic states (PDOS) of the surface and the binding energy of a given adsorbate,[1] and marginal improvements have been achieved by including the second moment.[2] The moments are defined as:

$$m_0 = \int_{\varepsilon} \rho(\varepsilon)\, d\varepsilon$$

$$m_1 = \frac{1}{m_0}\int_{\varepsilon} \rho(\varepsilon)\varepsilon\, d\varepsilon$$

$$m_{i>1} = \frac{1}{m_0}\int_{\varepsilon} (\rho(\varepsilon) - m_1)\varepsilon^i\, d\varepsilon$$

Additionally, recent work has demonstrated a similar linear correlation between the first moment of the PDOS of oxygen atoms adsorbed on various surfaces, spanning metals and metal oxides, and the binding energy of a hydrogen atom on this surface-bound oxygen.[3] These two works can be summarized mathematically as demonstrating the existence of some functions *f* and *g* such that:

$$TM_{PDOS}, A \xrightarrow{f} \Delta E_{TM-A}$$
$$O_{PDOS} \xrightarrow{g} \Delta E_{O-H}$$

where $TM_{PDOS}$ is the PDOS of the transition metal surface, $A$ is the identity of the molecular fragment to be bound, and $\Delta E_{TM-A}$ is the binding energy of this molecular fragment to the transition metal surface. Similarly, $O_{PDOS}$ is the PDOS of the adsorbed oxygen atom, and $\Delta E_{O-H}$ is the binding energy of a hydrogen atom to this oxygen. While *f* and *g* are quite specific in terms of their inputs, we hypothesize the existence of another, more general function *h*, which takes as arguments the PDOS of the atoms that will participate in bonding between the surface and molecular fragment, and returns the corresponding binding energy, summarized as follows:

$$S_{PDOS}, A_{PDOS} \xrightarrow{h} \Delta E_{S-A}$$

This idea is summarized in Figure 1. Beyond being of fundamental interest, such a function would enable the prediction of the binding energies of $m$ adsorbates at $n$ sites at a given surface from a single surface DFT calculation, as opposed to the $m \times n$ such calculations that would be required in its absence.

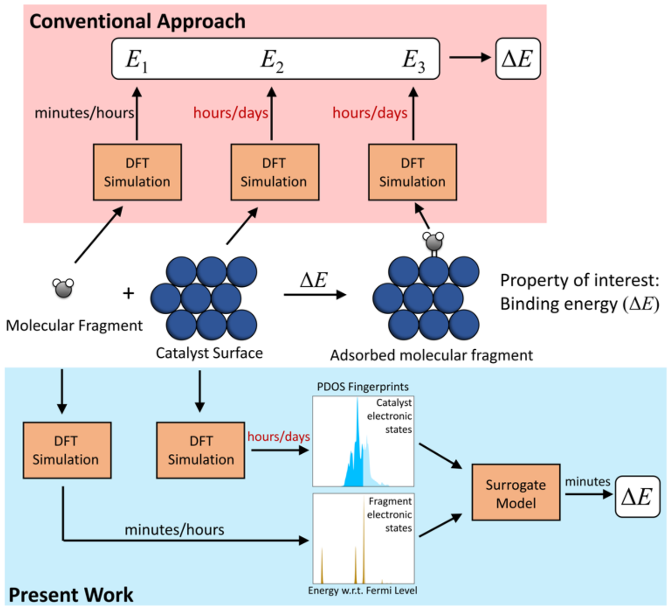

**Figure 1.** Schematic describing the conventional approach to binding energy calculations with DFT and the approach explored in the present work.

This work will investigate whether machine learning (ML) tools can be used to estimate *h*. In essence, we seek a model that can learn the principles of chemical bonding, such that given the electronic structure of two species in isolation, it can predict the energy released when they are brought together to bond. A model that can predict binding energies to within 0.2 eV of the value calculated by DFT will be considered a success, as the expected error of DFT calculations of binding energies on transition metal surfaces is known to be ~0.2 eV.[4]

## II. Data

### A. Data Generation

The initial phase of our project consisted of generating a database of periodic DFT calculations using the open source code, Quantum Espresso,[5] and the RPBE exchange-correlation functional.[6] This process consisted of bulk lattice parameter optimization of 39 different metals spanning body centered cubic (BCC), face centered cubic (FCC), and hexagonal close packed (HCP) crystal structures, followed by cleavage of various surface facets, and finally placement of NO, CO, $CH_x$, $NH_x$, and $OH_x$ adsorbates at all unique surface binding sites (where $x$ can take on values of 0, 1, 2, or 3). The corresponding molecular fragments were also simulated in vacuum with a computational cell of sufficient size to prevent self-interaction through periodic images.

Surface simulations consisted of performing a local optimization of the system's energy with respect to the atomic positions subject to certain constraints, namely that the lower two of four atomic surface layers were fixed to simulate the bulk and the in-plane position of the adsorbates' binding atom was fixed. DFT provides the gradient of the energy with respect to these positions, i.e. the forces on each atom, by solving for the electron positions using an approximate version of the Schrodinger equation. Following this optimization, the PDOS, work function, and system energy were extracted. Next, the resultant data was cleaned by removing systems that behaved unexpectedly upon structural optimization. Specifically, we designed filters that look for cases of molecular fragment dissociation, changes in molecular fragment coordination number, and significant distortion of the surface. We deemed these to be invalid examples and removed them from our analysis.

After cleaning our DFT calculations, we constructed a database for machine learning. A database entry for our machine learning models consist of three valid DFT calculations: an isolated molecular fragment and surface, and the combined system. The keys of the database consist of the bulk structure, composition, surface facet, adsorption site, initial state, and final state. For example:

| Bulk Structure | Composition | Surface Facet | Adsorption Site | Initial State | Final State |
|---|---|---|---|---|---|
| FCC | Au | 111 | Ontop | * | $CH_3$ |
| HCP | Ru | 0001 | Bridge | $CH_2$ | $CH_3$ |

The first key represents the process of $CH_3$ adsorbing at an empty site in an on-top geometry on the (111) surface facet of FCC gold. The relevant PDOS spectra in this case are for an Au surface atom and for the C atom in gas-phase $CH_3$. The second key represents the process of H adsorbing on adsorbed $CH_2$ sitting in a bridge site on the (0001) surface facet of HCP ruthenium. In this case, the relevant PDOS spectra are for the C atom in adsorbed $CH_2$ and an H atom in vacuum. After data cleaning, we were left with a database of 2000 examples on which to train. The data generation process is summarized by the schematic in Figure 2.

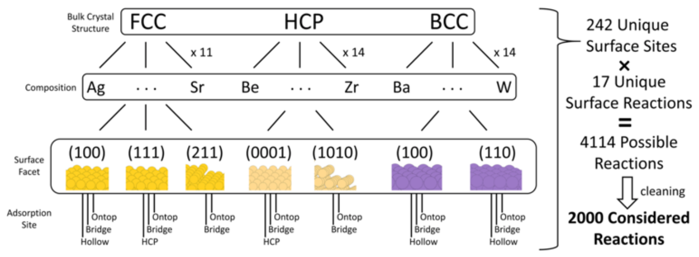

**Figure 2**. Schematic of the data generation process.

## B. *Feature Engineering*

The raw inputs to our ML models, the PDOS of the surface and the molecular fragment, are continuous 1D scalar functions that can be discretized to arbitrary resolution. Based on the aforementioned success of correlating binding energies with first and second moments, we will start by using as feature vectors the first ten moments of the PDOS of both the surface and molecular fragment. Next, we train another set of models on the raw spectra with spacing 0.1 eV, such that the feature vector is a concatenation of the surface and molecular fragment PDOS. In the case of convolutional neural networks, we stack rather than concatenate, creating a two-dimensional array. In instances where the molecular fragment binds to more than one surface atom, the representative surface PDOS is taken to be the sum of the PDOS of every surface atom involved in the bond.

It is important to note that the PDOS is computed relative to the Fermi level of the calculation, which represents the energy that divides filled and unfilled electronic states at zero temperature. However, when considering chemical bonding between the surface and molecular fragment, it is also relevant to know where their electronic states lie on an absolute scale, e.g. relative to vacuum. The energy difference between the Fermi level and vacuum for the surface is defined as the work function (WF), and we will also refer to this quantity as the work function in the case of the molecular fragment, although this is an abuse of nomenclature. Both WFs are included in our feature vectors in addition to the information provided by the PDOS.

## III. Methodology

All models were trained[†] via Python packages Scikit-learn (linear regression, kernel ridge regression, and random forest),[7] Catboost (gradient boosting),[8] or PyTorch (convolutional neural network).[9] In the following model definitions, $X$ is the $m \times n$ design matrix having $m$ examples and $n$ features, $y$ is the length $m$ vector of feature labels, $h_\theta$ is the hypothesis, or the function that predicts values of $y$ given $X$, $J$ is the loss function, and $\theta$ is the vector of model parameters.

## A. *Linear Regression (LR)*

Linear regression utilizes a linear model of the form

$$h_\theta(x) = \theta^T x$$

We can exactly solve for the parameters $\theta$ which minimize the squared error via the normal equations:

$$\theta = (X^T X)^{-1} X^T y$$

## B. *Kernel Ridge Regression (KRR)*

Kernel ridge regression can be viewed as an extension of linear regression with the addition of mechanisms to reduce bias (kernel trick) and variance (regularization). The kernel trick requires reformulating the training process to be purely in terms of inner products of examples such that individual feature vectors need not be explicitly represented. This allows the replacement of inner products with kernel functions that correspond to inner products in an arbitrary high dimensional space, resulting in a nonlinear model when projected back onto the original feature space. Regularization is implemented to combat overfitting by adding the L2 norm of the parameter vector to the squared error lost function:

[†] Code available at https://github.com/colinfd/ChemLearn

$$J = \sum_{i=1}^{m} \left(\theta^T x^{(i)} - y^{(i)}\right)^2 + \lambda \left|\left|\theta\right|\right|^2$$

Because KRR is a distance-based model, it was important to normalize the data before applying the model. We did so by subtracting by the mean and dividing by the standard deviation of each feature.

### C. *Random Forest (RF)*

Decision tree models divide feature space into discrete regions, and for every example that falls within a given region, the same prediction is made. In the case of regression, the prediction is the average of all training labels in that region. The goal is to choose these regions such that the residual sum of squares is minimized:

$$\sum_{j=1}^{J} \sum_{i \epsilon R_j}^{m} \left(y^{(i)} - \hat{y}^{R_j}\right)^2$$

Bagging algorithms, in which the predictions of many independently trained models are averaged, are often employed with decision trees to combat overfitting. The random forest algorithm further reduces variance by encouraging trees in the ensemble to be decorrelated by only allowing a subset of the original *n* features to be used in each tree.

### D. *Gradient Boosting (GB)*

Boosting algorithms also fall under the category of ensemble learning algorithms, where the predictions of multiple models are summed to achieve the overall prediction:

$$F_M(x) = \sum_{i=1}^{M} h_i(x)$$

In the case of boosting, these models are trained in series, where specifically, each new model is trained on the residuals of the sum of the previous models:

$$h_{M+1}(x) \approx y - F_M(x)$$

The boosting algorithm as written above minimizes the squared error of the total model. So-called gradient boosting abstracts away this choice of loss function and instead trains each successive model on "pseudo residuals" of the previous model, which are simply the gradient of the loss function with respect to the predictions of the previous model:

$$h_{M+1}(x) \approx -\frac{\partial J}{\partial F_M(x)}$$

Such that the model update becomes:

$$F_{M+1}(x) = F_M(x) + \alpha h_{M+1}(x)$$

Note the gradient descent form of the expression and the presence of the learning rate, $\alpha$. While these expressions are general in terms of the algorithm used to train each model, decision tree ensembles are typically used, which introduce typical hyperparameters such as tree depth.

### E. *Convolutional Neural Network (CNN)*

Convolutional neural networks are artificial neural networks that utilize one more convolutional layer. Convolutional layers differ from fully connected layers in that they perform convolutions of the input data with a relatively small kernel matrix rather than performing a matrix multiplication with a dense matrix whose dimensionality is similar to the input data. Because the elements of the kernel matrix or dense matrix are parameters to be tuned, the convolutional layer results in much fewer total parameters. However, use of a convolution operation requires the input data have some regular, grid-like structure, which makes it a useful tool for image analysis, or in our case, spectra analysis.

Our CNN was implemented using PyTorch[9] with an architecture consisting of a series of convolutional layers with 8 filters of depth 8 (with the exception of the input layer, whose filters have depth 2). Each convolutional layer was followed by PReLU activation, and the result of the final convolutional layer was passed through a fully connected layer also with PReLU activation before making a scalar prediction at the output layer. The loss function (mean squared error) was minimized with the Adam optimizer. When training, ~150 epochs were sufficient to reach a minimum in the dev set error.

### F. *Error Metric*

As is the convention in the field of heterogenous catalysis, we use mean absolute error (MAE) to quantify the performance of our models, defined as:

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^{m} \left|y^{(i)} - \hat{y}^{(i)}\right|$$

### G. *Train Test Splits*

To prevent model overfitting, the labeled examples were split into train (70%), dev (15%), and test (15%) sets. Training data was used to determine optimal model parameters, and dev data was used to evaluate the model's ability to generalize to previously unseen data and to tune hyperparameters. Finally, the model was judged by its performance on the test set.

We suspect that some of the examples in our dataset are highly correlated (e.g. the same molecular fragment adsorbing on different sites at the same surface), and we anticipate that a typical use case would involve making predictions on data that is unlike the data used to train the model. In the domain of heterogeneous catalysis, we might broadly consider cases to be similar if they involve the same surface composition and/or the same reaction. For the purposes of this work, we group each example by its composition and reaction and then perform the train-dev-test splits such that the model will never be asked to make a prediction on an example whose reactants and surface composition it has already trained on. As a concrete example, if the model has trained on O binding at the ontop site of Au (111), it will not be allowed to test on O binding at any site on any surface of Au. K-fold cross-validation, which was used for hyperparameter tuning, was also performed by making folds in this way. Alternative split schemes will be discussed later.

## IV. Results and Discussion

### A. *Using moments as features*

In Figure 3, we analyze our data in the context of previously published results. Figure 3a demonstrates the correlation between binding energy of C* or O* and the 1st moment of the binding surface atom's PDOS similar to previous results.[3] As expected, the parameters of the linear fit are both adsorbate and site dependent. In Figure 3b, we show another previously observed correlation between the binding energy of H binding on surface-bound O and the 1st moment of the O PDOS. While this is a more general correlation than that shown in Figure 3a, we note that it does not extend to the analogous case of H binding on surface-bound C (Figure 3c).

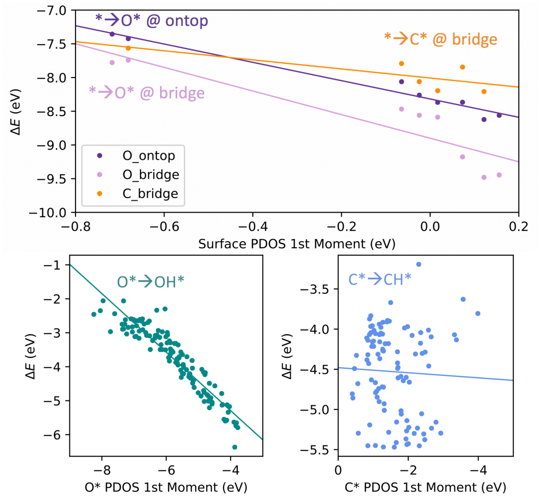

**Figure 3**. Analysis of present calculations in context of previous work. (a) Correlation between binding energy of C* or O* and the 1st moment of the binding surface atom's PDOS, (b) correlation between the binding energy of H binding on surface-bound O* and the 1st moment of the O* PDOS, (c) no analogous correlation exists for H binding on surface-bound C*

In our quest to identify a single model that maps PDOS to binding energy, we begin by extending the univariate linear models shown in Figure 3 to a multivariate linear regression model that takes as inputs the first ten moments of the binding atom PDOS in both the surface and molecular fragment. As shown in the learning curve in Figure 4, linear regression does not appear to be flexible enough to describe the variation in the data, suffering from high bias and achieving a poor training MAE of 0.81 eV compared to our desired threshold of 0.2 eV. Additionally, its performance seems to have plateaued around 800 training examples, suggesting that additional training data will not improve the model. For this reason, we sought the greater representational power of nonlinear models and tested kernel ridge regression (KRR), random forest (RF), and gradient boosting (GB). Cross-validation (3 folds) combined with a randomized grid search was used to tune hyperparameters for KRR and RF. For KRR, the optimal hyperparameters were found to be an RBF kernel with $\alpha$=0.00089 and $\gamma$=0.45. For RF, the optimal hyperparameters were found to be maximum tree depth=18, maximum features allowed per split=11, minimum samples in each split=2, and number of trees=100. Training GB models was too expensive to implement cross-validation, so the dev set was used to determine when to stop training, and manually modifying hyperparameters such as learning rate, tree depth, and regularization parameter was found to have little if any benefit and so the defaults were used. The learning curves for these models are shown in Figure 4.

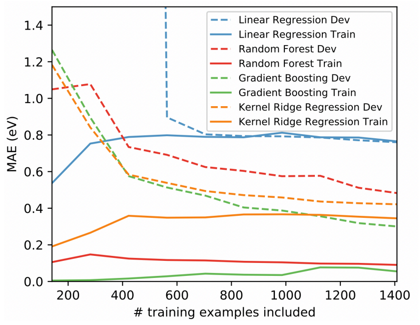

**Figure 4.** Learning curves for LR, RF, GB and KRR.

Nonlinear models generally seem to outperform linear regression. However, both of the tree methods, GB and RF, are overfitting the data as demonstrated by the large difference between train and dev errors. We found that decreasing the tree depth mitigated the overfitting but came at the cost of increased dev set errors. Notably, the learning curves of both GB and RF do not yet seem to have plateaued, suggesting that gathering additional training data could improve the models. KRR, on the other hand, is not overfitting but seems to have already plateaued. Parity plots for the train and test datasets using the optimized models are shown in Figure 5. We find that the test MAEs are similar to the dev MAEs shown in the learning curve and that there do not appear to be any significant outliers.

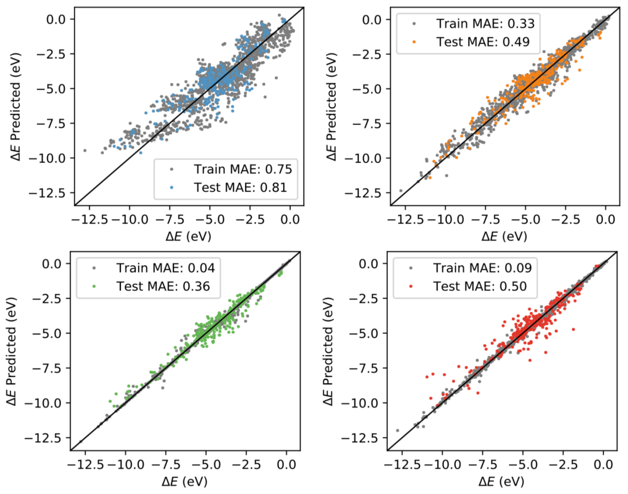

**Figure 5.** Parity plots for Linear Regression (blue), KRR (orange), GB (green) and RF (red).

Finally, we can examine the feature importance of the tree-based methods (Figure 5b). Both GB and RF attribute greater importance to the lower moments, but RF does not attribute as much importance to the surface PDOS as does GB. We also see that GB attributes greater significance to the WFs than RF.

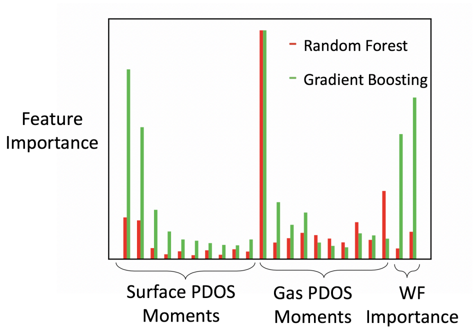

**Figure 6.** Feature importance of tree-based methods.

## *B. Using the full PDOS spectra as features*

While using feature vectors consisting of the first 10 moments of the PDOS and the WFs gives reasonable performance, we recognize that the spectra are not single distributions and are much more complex (*e.g.* Figure 1), suggesting that a simple moment analysis may not be sufficient to capture the richness of the spectra. Therefore, we decided to train several models, namely RF, GB and a convolutional neural network (CNN), on the raw spectra. Again, the hyperparameters for RF and GB were determined as described above. In building the CNN, we sought help from our lab mate, Brian Rohr. We found the model to strongly overfit when regularization techniques were not employed, and we found the "reparameterization trick" first proposed by Kingma and Welling to be an effective defense against overfitting.[10] In this scheme, representations are drawn from a distribution at the output of the final convolutional layer before being passed to the fully connected layers for training examples, effectively adding random noise to the training procedure. By considering the dev set error and manually trying a few iterations on our architecture, we ultimately decided on 4 convolutional layers with 8 filters (of width 5) of depth 8 and a stride of 4 followed by fully connected layer with 30 nodes. Further architecture details can be found in Methods.

Shown in Figure 7 are the learning curves for these models. We see that RF and GB perform similarly to before, albeit with lower train and dev MAEs. The CNN does not overfit and may also benefit from training on additional data. Interestingly, the train error becomes greater than the dev error after 600 training examples, which is a consequence of the fact that when training, the output of the convolutional layers, i.e. the representation, is drawn from a distribution before being passed to the fully connected layer.

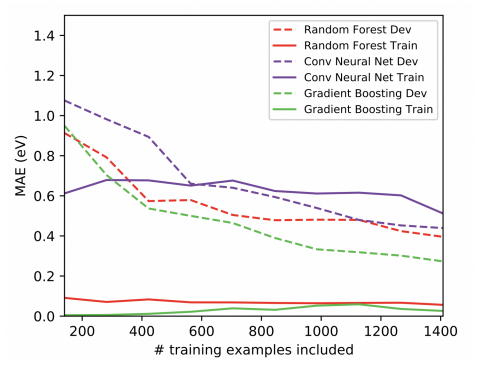

**Figure 7**. Learning curves for RF, GB, and CNN.

As before, the parity plots for the optimized models (Figure 8) show similar test MAEs to the dev MAEs from the learning curves. Notably, we are able to achieve a test MAE of 0.27 eV using GB, which is quite close to our target MAE of 0.2 eV, and acceptable for many screening applications.

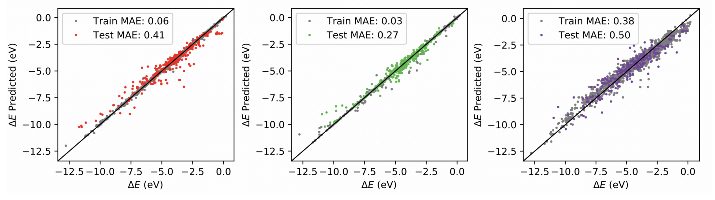

**Figure 8**. Parity plots for RF (red), GB (green), and CNN (purple).

Finally, we can examine the feature importance for the tree-based methods. In Figure 9, we show the feature importance for RF (black) along with the average feature values from the entire dataset. We find that the most important surface features are near the Fermi level and states far below the Fermi level have little importance, which is expected based on basic chemical bonding principles. Interestingly, surface electronic states far above the Fermi level have a significant importance relative to their infrequent occurrence. The importance of the molecular fragment features seems less organized and is reflective of the discrete nature of the electronic states of molecular fragments and the limited number of them in our dataset.

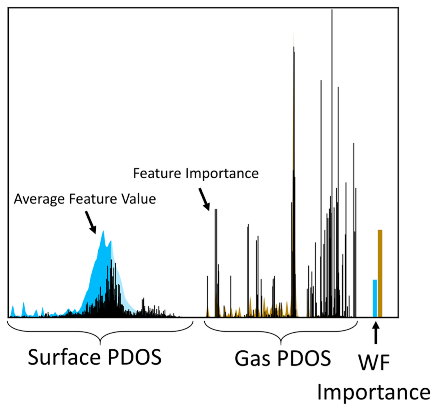

**Figure 9.** Feature importance for RF using the full PDOS spectra as features (black) and average feature values for the surface (blue) and molecular fragment (orange)

## V. Conclusions and Future Work

In this work, we have demonstrated some of the promise for data-driven models to make predictions about the strength of chemical bonds at surfaces when given electronic structural information about the system, namely the PDOS of the atoms involved in the bond. We found a GB model trained on the raw PDOS spectra and reactant work functions yielded optimal performance with a test MAE of 0.27 eV, which is a sufficient accuracy for many screening applications. This performance was achieved using a train-test split such that the model was not allowed to train on any examples that shared the same reaction and surface composition as any example in the test set, which is likely a a typical use case; however we note that model performance can be quite sensitive to the way in which the train-test split is performed as shown in Table 1. Depending on the use case, some of these splits may be more relevant than others.

**Table 1.** Sensitivity of test MAEs on the type of train-test split. The "Reaction" and "Composition" entries correspond to splits where the model is tested on either reactions or surface compositions, respectively, that it has not seen during training. "Composition + Reaction" is the method used throughout the main text and is described there. "Random" is simply a random split without any constraints.

| MAE (eV) | Moments & WF | | | | PDOS (& WF) | | |
|---|---|---|---|---|---|---|---|
| | LR | KRR | RF | GB | RF | GB | CNN |
| Reaction | 5.15 | 1.31 | 0.75 | 0.46 | 0.94 | 1.14 | 0.67 |
| Composition | 1.26 | 0.55 | 0.6 | 0.57 | 0.5 | 0.44 | 0.44 |
| Composition + Reaction | 0.81 | 0.49 | 0.5 | 0.36 | 0.41 | 0.27 | 0.50 |
| Random | 0.8 | 0.44 | 0.38 | 0.29 | 0.41 | 0.28 | 0.35 |

In the future, it would be of interest to investigate whether our model generalizes to more complicated materials such as alloys and metal-oxides. Furthermore, we would like to experiment with other featurization techniques for the molecular fragments, the PDOS of which tend to be sparse with very narrow peaks. A first approach may be to simply replace the PDOS representation with a categorical variable indicating the molecular fragment identity. We expect such a modification to greatly simplify model training. However, it also comes with an inherent loss in generalizability as it necessarily precludes the model's ability to make predictions of reactions that involve molecular fragments it has not seen before.

## Acknowledgment

We are grateful to the instructors of CS229, Andrew Ng and Ron Dror, as well as all of the teaching assistants for their help and support. We also are grateful to our lab-mate, Brian Rohr, who designed and built the convolutional neural networks and provided valuable guidance and advice. Finally, we thank the Stanford Sherlock Computing Resources.

## References

[1] Hammer, B. and Nørskov, J. K. Electronic factors determining the reactivity of metal surfaces. *Surface Science* 343.3 (1995): 211-220.

[2] Vojvodic, A., Nørskov, J. K., & Abild-Pedersen, F. Electronic Structure Effects in Transition Metal Surface Chemistry. *Topics in Catalysis*, *57*(1–4) (2014): 25–32.

[3] Dickens, C. F., Montoya, J. H., Bajdich, M., Kulkarni, A., Nørskov, J. K. An electronic structure descriptor for oxygen reactivity at metal and metal-oxide surfaces. *Surface Science,* 681 (2018): 122-129.

[4] Wellendorff, J., Silbaugh, T. L., Garcia-Pintos, D., Nørskov, J. K., Bligaard, T., Studt, F., & Campbell, C. T. A benchmark database for adsorption bond energies to transition metal surfaces and comparison to selected DFT functionals. *Surface Science*, *640*, (2015): 36–44.

[5] Giannozzi, P., et al. Quantum ESPRESSO: A modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter*, *21*, (2009): 395502–395521.

[6] Hammer, B., Hansen, L. B. and Nørskov, J. K. Improved adsorption energetics within density-functional theory using revised Perdew-Burke-Ernzerhof functionals. *Phys. Rev. B, 59,* (1999): 7413

[7] Pedregosa et al. Scikit-learn: Machine Learning in Python Journal of Machine Learning Research (2011).

[8] Prokhorenkova, Liudmila et al. CatBoost: unbiased boosting with categorical features. *arXiv*:1706.09516 (2018).

[9] Paszke, Adam et al. Automatic differentiation in PyTorch, NIPS-W (2017).

[10] Kingma, D. & Welling, M. Auto-Encoding Variational Bayes. *arXiv*:1312.6114 (2014).