LOCAL ENVIRONMENT INTERACTION FRAMEWORK FOR MACHINE LEARNING MOLECULAR ADSORPTION ENERGY

Li Yifan
Department of Mechanical Engineering,National University of Singapore, 117575, SG
liyifan@nus.edu.sg

Wu Yihan
Department of Mechanical Engineering, National University of Singapore, 117575, SG
wuyihan@u.nus.edu

Han Yuhang
Department of Mechanical Engineering,National University of Singapore, 117575, SG
yuhang_han@u.nus.edu

Lyu Qiujie
Department of Mechanical Engineering,National University of Singapore, 117575, SG
e1110106@u.nus.edu

Wu Hao
Department of Mechanical Engineering,National University of Singapore, 117575, SG
e1010595@u.nus.edu

Zhang Xiuying
Department of Mechanical Engineering,National University of Singapore, 117575, SG
phyxyz@nus.edu.sg

Shen Lei*
Department of Mechanical Engineering,, National University of Singapore, 117575, SG
shelei@nus.edu.sg

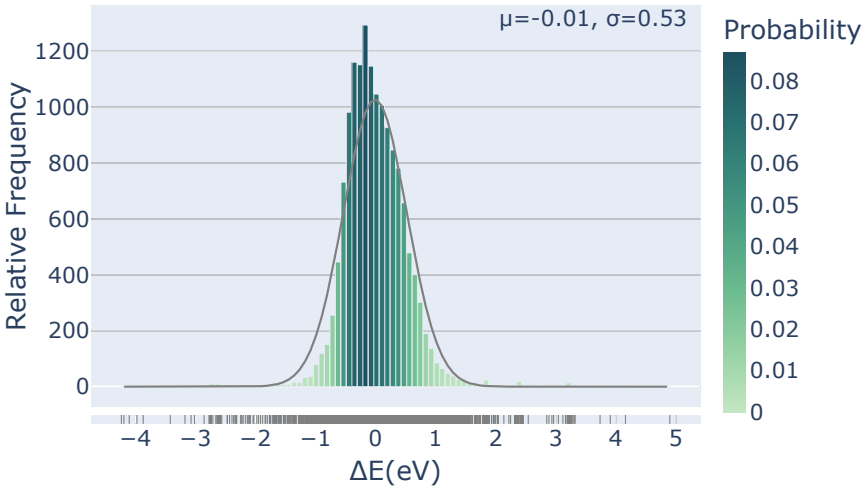


Figure 1: The distribution of the hydrogen dataset

Supplementary Materials

S1

The data shown in Figure 1 appears to follow a normal distribution, with a symmetrical bell-shaped curve and a small standard deviation. This type of distribution is particularly useful in machine learning applications as it enables the use of statistical methods that assume a normal distribution of data. Machine learning algorithms, such as linear regression and decision trees, often rely on normal distribution assumptions to generate accurate predictions and make informed decisions. A normally distributed dataset is also easier to model and analyze, as it follows a predictable pattern that is more straightforward to interpret. As such, the presence of a normal distribution in the data depicted in the icon suggests that machine learning algorithms could be effectively trained using this data. However, further analysis and validation of the data are necessary to confirm its suitability for machine learning purposes.

S2

The CGCNN deep learning framework has demonstrated remarkable performance in several applications. In this work, we propose the use of the modified Voronoi tessellation method to optimize the original structure of cgcnn into a Voronoi structure input. This approach represents our modified VT method gives a significant develop in the field of low data catalysis.

Our results, as depicted in Figure 2, show that Modfied CGCNN achieves superior convergence performance compared to the original CGCNN. The proposed model efficiently captures the necessary information for calculating the adsorption energy with fewer iterations. We attribute this to the enhanced filtering of features by extracting local information, thus facilitating the identification of information localized to the adsorption site. The faster training speed of Modfied CGCNN is particularly relevant for large datasets, given that deep learning algorithms typically require long training cycles due to the high number of hidden layers in neural networks.In methods part, we present a detailed account of our findings and provide supporting evidence for our claims.However, there are some inherent limitations of crystal graph networks in adsorption energy calculations, which we discuss in the Methods section, and therefore we consider the application of feature engineering in this area.

However, the original CGCNN model can still extract local chemical information when the training set is sufficiently large and the number of iterations is high, albeit requiring a significantly larger dataset.

We can prove the reliability of this result through the following process:

Assume that the input sizes of the two CNNs (with and without VT processing) are M and C, respectively, and their regression results are both y. Since the CNN with VT transformation can extract information around the adsorption site faster, it has a faster training speed.

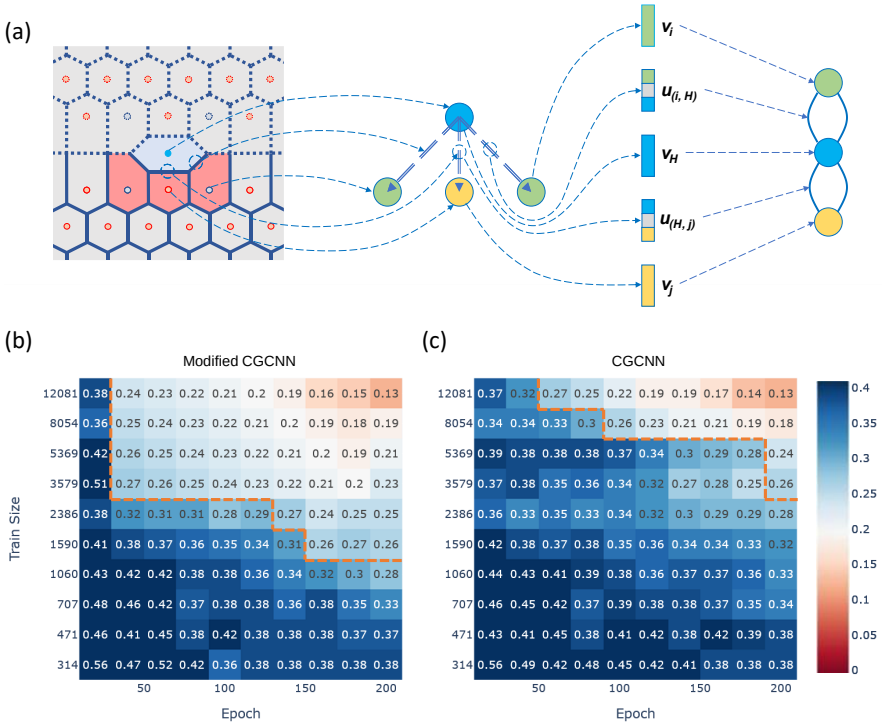


Figure 2: (a) Illustration of Modfied CGCNN crystal graph. (b) and (c) The heatmaps of two models of MAE(eV) corresponding to different training sizes and Epochs.

We can represent the loss functions of the CNNs as L(m) and L(c), which represent the loss functions of Modified CGCNN and CGCNN, respectively. At the end of each epoch(e), we record their loss values and compare their relative change rates, i.e.,

$$r = |\frac{L(m, \mathrm{e}_i) - L(m, \mathrm{e}_{i-1})}{L(m, \mathrm{e}_{i-1})}| - |\frac{L(c, \mathrm{e}_i) - L(c, \mathrm{e}_{i-1})}{L(c, \mathrm{e}_{i-1})}| \tag{1}$$

where $L(m, \mathrm{e}_i)$ and $L(c, \mathrm{e}_i)$ represent the loss function values of Modified CGCNN and CGCNN, respectively, at the i-th epoch, and r represents the difference between their relative change rates.

Since Modified CGCNN has a faster training speed, for any epoch, its loss function value will decrease faster, so that,

$$r_k = |\frac{L(m, \mathrm{e}_k) - L(m, \mathrm{e}_{k-1})}{L(m, \mathrm{e}_{k-1})}| - |\frac{L(c, \mathrm{e}_k) - L(c, \mathrm{e}_{k-1})}{L(c, \mathrm{e}_{k-1})}| \geq 0 \tag{2}$$

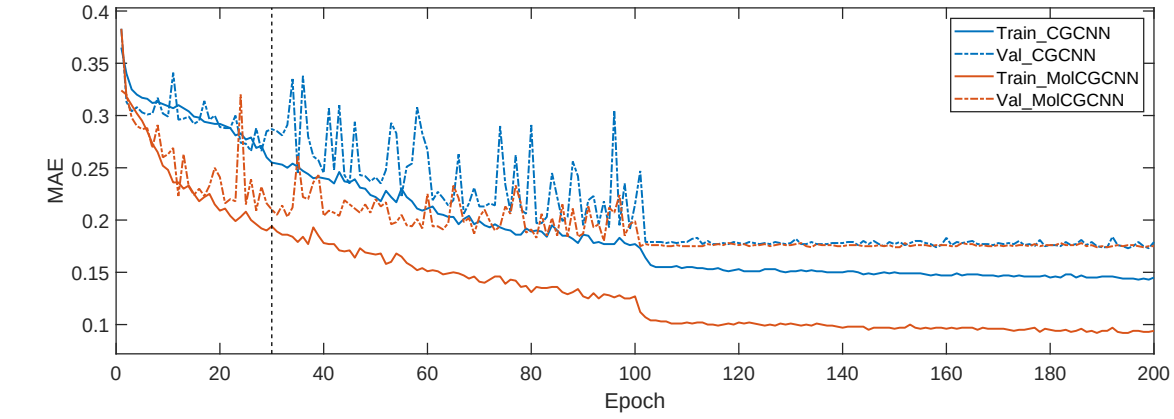
Now, we can use mathematical induction to prove that if the training speed remains constant, the value of r will remain constant for all epochs. That is, assuming the value of r remains constant for any $i < k$, we need to prove that the value of r remains constant at epoch k.

At epoch k, based on the premise assumption, we can obtain,

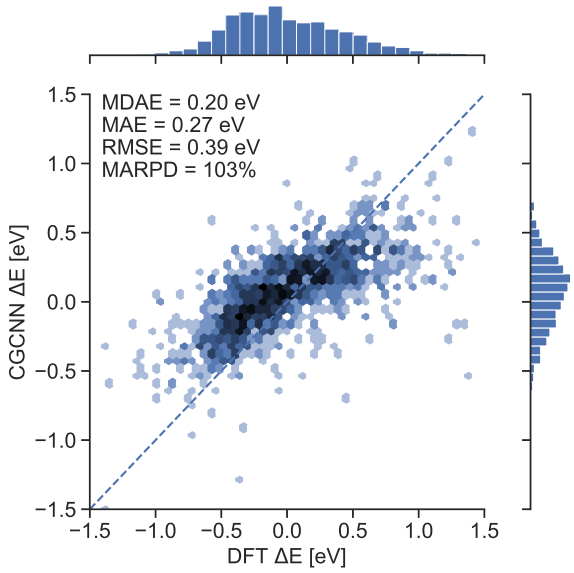
$$|\frac{L(c, \mathrm{e}_k) - L(c, \mathrm{e}_{k-1})}{L(c, \mathrm{e}_{k-1})}| \leq |\frac{L(m, \mathrm{e}_k) - L(m, \mathrm{e}_{k-1}))}{L(m, \mathrm{e}_{k-1})}| \tag{3}$$

And integrating both sides with respect to the time step t=k simultaneously,

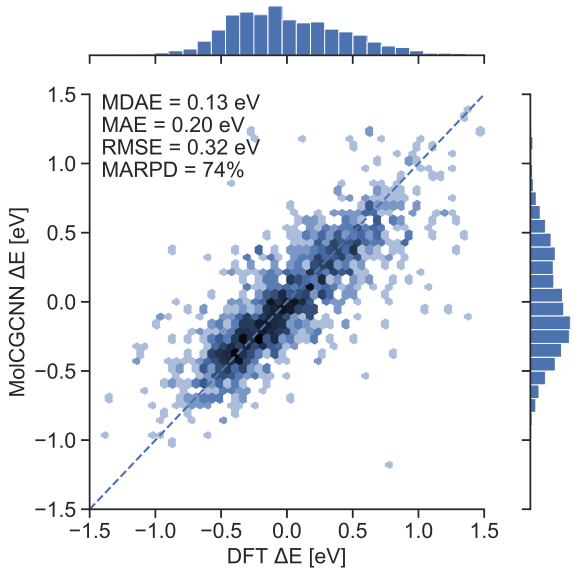
$$\int_{t=k-1}^{t=k} |\frac{L(c, \mathrm{e}_k) - L(c, \mathrm{e}_{k-1})}{L(c, \mathrm{e}_{k-1})}| dt \leq \int_{t=k-1}^{t=k} |\frac{L(m, \mathrm{e}_k) - L(m, \mathrm{e}_{k-1}))}{L(m, \mathrm{e}_{k-1})}| dt \tag{4}$$



(a)

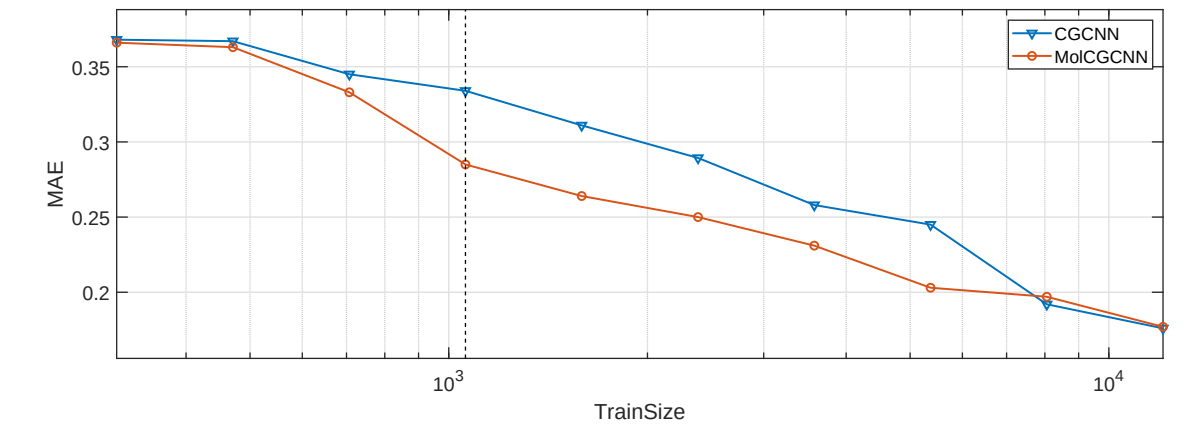


(b)

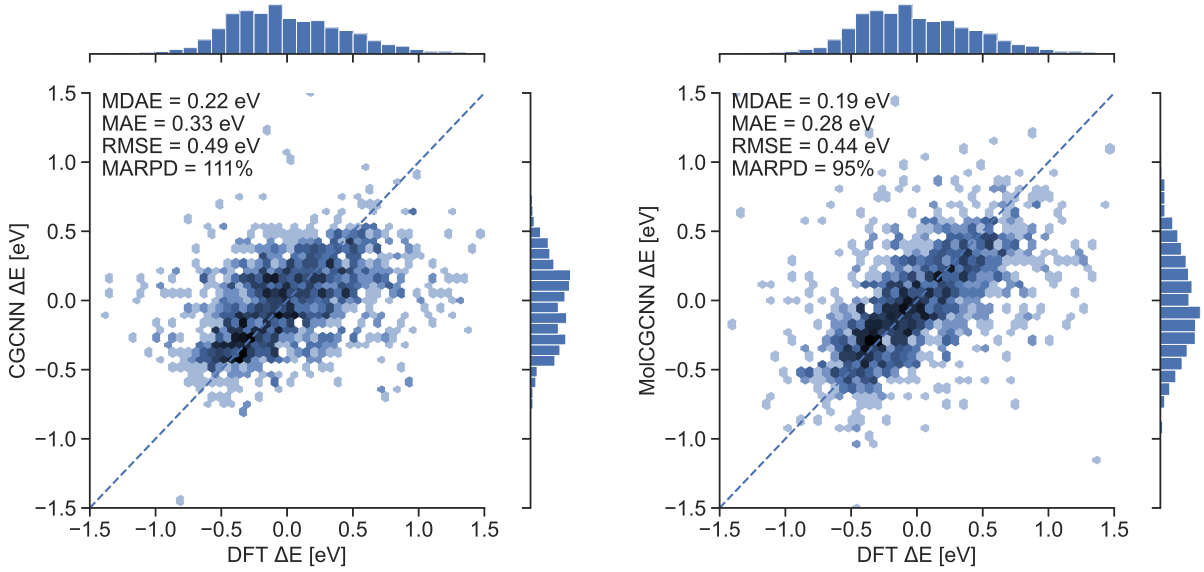


(c)

Figure 3: (a)The change curve of MAE on the training set and verification set when the CGCNN and Modified CGCNN models iterate 200 times respectively. (b)and (c)are scatter plots of predicted values vs. DFT calculated values when the CGCNN and Modified CGCNN models only iterate 30 times on training set respectively.



(a)



(b) (c)

Figure 4: (a) The MAE of the CGCNN and Modified CGCNN models on the test set when the training set sizes are 314, 471, 707, 1060, 1590, 2386, 3579, 5369, 8054, and 12081 respectively (200 epoch). (b) and (c) are scatter plots of predicted values vs. DFT calculated values when the training set size is 1060.

Therefore,

$$|L(m, \mathrm{e}_k) - L(m, \mathrm{e}_{k-1})| \geq |L(c, \mathrm{e}_k) - L(c, \mathrm{e}_{k-1})| \tag{5}$$

Moreover, since the loss decreases with increasing number of iterations during the training process, it is evident that, $L(m, \mathrm{e}_k) < L(m, \mathrm{e}_{k-1})$ and $L(c, \mathrm{e}_k) < L(c, \mathrm{e}_{k-1})$. Then we have,

$$L(c, \mathrm{e}_k) - L(m, \mathrm{e}_k) \geq L(c, \mathrm{e}_{k-1}) - L(m, \mathrm{e}_{k-1}) \tag{6}$$

As a result, it can be inferred that prior to final convergence, the difference between the losses of the two models increases gradually and then remains constant. Therefore, the intermediate segment before the convergence of the two models in the figure can be approximately considered as two parallel line segments with equal slopes. At this point, the MAE of Modified CGCNN is always less than that of CGCNN, and the difference remains constant. Modified CGCNN has higher accuracy with fewer convergence times. This proof also applies to the case where the size of the training set is small.

When the dataset is sufficiently large, assume that the optimal solutions of Modified CGCNN and CGCNN are α_1 and α_2, respectively, and their loss functions are $L(m, \alpha_1)$ and $L(c, \alpha_2)$, respectively. Since both CNNs are initialized with the same random seed during training, they have the same network structure and number of parameters, but different input sizes. Based on the convolutional properties of CNNs, we can divide the layers of CNNs into convolutional layers and fully connected layers. The number of parameters in convolutional layers depends on the size and number of convolutional filters, and is independent of input size. Therefore, Modified CGCNN and CGCNN have the same number of parameters in convolutional layers. We can represent the parameters of the fully connected layers of CNNs as w and b. For Modified CGCNN, its input size is m and output size is y. For CGCNN, its input size is c and output size is the same as the former. Therefore, the sizes of w and b for the fully connected layers of Modified CGCNN and CGCNN are $w_m \in R^{m \times y}$, $b_m \in R^y$, and $w_c \in R^{c \times y}$, $b_c \in R^y$, respectively. Thus, for any input x, the outputs of Modified CGCNN and CGCNN are:

$$\mathrm{Output}_m(x, \alpha_1) = f_1(w_m * x + b_m) \tag{7}$$

$$\mathrm{Output}_c(x, \alpha_2) = f_2(w_c * x + b_c) \tag{8}$$

Since the parameters w_m, b_m and w_c, b_c of the two CNNs are the same, that is, $w_m = w_c$, $b_m = b_c$, the size y is equal, and the local environment extraction retains structural information that has a significant impact on the adsorption energy, without compromising the coherence of the original structure or the information of the effective nodes and bonds, so the mapping function remains the same after convergence, we have:

$$\begin{aligned} &\mathrm{Output}_m(x, \alpha_1) = f(w_m * x + b_m) \\ &= f(w_c * x + b_c) = \mathrm{Output}_c(x, \alpha_2) \end{aligned} \tag{9}$$

Therefore, we have proven that when the dataset is sufficiently large, the final convergence results of these two CNNs are equal.

S3

Figure 5 shows the final feature importance ranking after combining with Matminer.

It should be noted that the numbers following the underscore signify the line numbers in the Fingerprint method. It is important to mention that the importance of these properties is subject to variation depending on the surface placement, thus limiting their reference value.

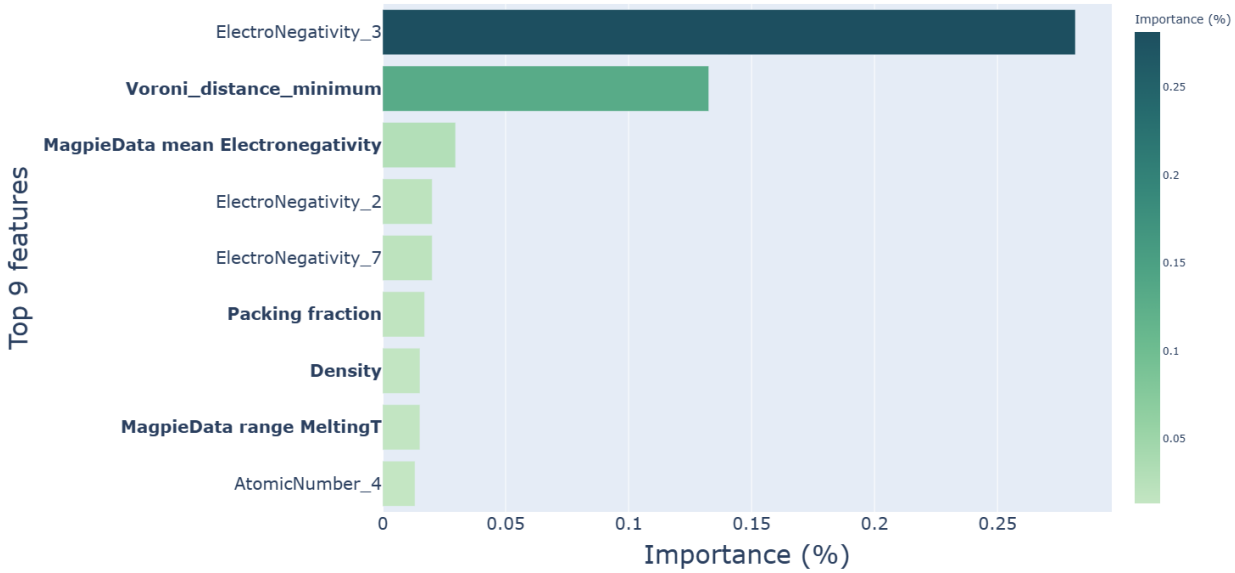


Figure 5: The feature importance