Author(s): Pihlajamäki, Antti; Malola, Sami; Kärkkäinen, Tommi; Häkkinen, Hannu

Title: Graphs and Kernelized Learning Applied to Interactions of Hydrogen with Doped Gold Nanoparticle Electrocatalysts

Year: 2023

Version: Published version

Please cite the original version:

Pihlajamäki, A., Malola, S., Kärkkäinen, T., & Häkkinen, H. (2023). Graphs and Kernelized Learning Applied to Interactions of Hydrogen with Doped Gold Nanoparticle Electrocatalysts. Journal of Physical Chemistry C, 127(29), 14211-14221. https://doi.org/10.1021/acs.jpcc.3c02539

THE JOURNAL OF PHYSICAL CHEMISTRY C
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

pubs.acs.org/JPCC

Article

Graphs and Kernelized Learning Applied to Interactions of Hydrogen with Doped Gold Nanoparticle Electrocatalysts

Published as part of The Journal of Physical Chemistry C virtual special issue "Machine Learning in Physical Chemistry Volume 2".

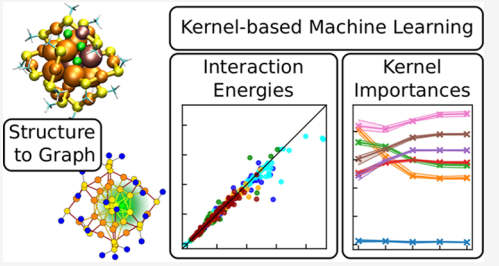
Antti Pihlajamäki, Sami Malola, Tommi Kärkkäinen, and Hannu Häkkinen*

Cite This: *J. Phys. Chem. C* 2023, 127, 14211−14221

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Understanding hydrogen adsorption on metal nanoparticles is a key prerequisite for designing efficient electrocatalysts for water splitting and the hydrogen evolution reaction. However, this seemingly simple elementary reaction step is affected by several factors arising from the chemical environment at the catalyst, and deciphering the most important contributions to optimal interactions requires numerically heavy electronic structure calculations. Here, we combine graph-based representations of the local atomic environment of hydrogen in copper- and palladium-doped 25-atom gold nanoparticles with several kernel-based machine learning (ML) methods to predict the interaction energy between hydrogen and the nanoparticle catalyst. We demonstrate that simple distance-based kernel models are able to predict the interaction energy within 0.1 eV when trained by reference data from state-of-the-art density functional theory calculations. Analyzing the model performance with respect to attributes of the hydrogen node highlights the locality of hydrogen adsorption. This implies the viability of combining graphs with kernel-based ML models for studying hydrogen chemisorption in complex environment data efficiently.



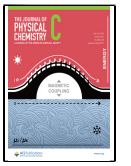
1. INTRODUCTION

The ever-increasing energy demand of the society and the awareness on environmental issues are increasing the interest toward clean energy. Wind, wave, and solar power are good renewable energy sources to produce electricity, but the storage and usage as a fuel are still a challenge. Hydrogen is a strong alternative to solving some of these problems. The production of hydrogen relies on water splitting associated with hydrogen evolution reactions (HERs) and oxygen evolution reactions (OERs), both requiring suitable catalysts.[1,2] The former is the main motivation of this study. In order to design better catalysts for any reaction, one has to understand the reaction mechanisms and the factors of catalytic activity. Various computational methods are irreplaceable tools to probe these properties.[3] Increased computational power and data storage have enabled the increase of machine learning (ML)-based studies on catalysis.[4−8] ML methods can be used to mine underlying dependencies from either experimental or computational data in order to create models for screening catalytic materials,[9] finding binding sites,[10] or creating ML force fields to run dynamic simulations.[11]

One class of materials, with promising catalytic properties on HER and OER, are provided with atomically precise metal nanoparticles, also called monolayer protected clusters (MPCs).[12−17] MPCs are nanoparticles, which consist of a metallic core, an organic ligand layer, and an interface structure between.[18] By controlling the metallic composition and protecting ligands, one can tailor the properties of the MPCs for given tasks.[18−22] MPCs are challenging systems to any computational method due to their chemical complexity. However, they have also been studied on several occasions by ML methods. The metal−ligand interface structures have been predicted with a rule-based method,[23] and positions of the hydrides have been estimated with neural networks.[24,25] Deep learning methods have been utilized to predict from experimental settings whether gold MPCs are formed and to analyze determining factors of the synthesis.[26] Distance-based ML methods have been proven to be able to create realistic potentials.[27,28] Support vector machines have been utilized to evaluate the fluorescence properties of MPCs,[29,30] and recently convolutional neural networks have been able to detect features in UV−vis spectra of mixtures of different sized thiolate protected clusters.[31]

In this study, we utilized four different kernel-based ML methods, minimal learning machine (MLM),[32,33] extreme

Received: April 17, 2023
Revised: June 8, 2023
Published: July 18, 2023



minimal learning machine (EMLM),[34,35] kernelized ridge regression (KRR),[36,37] and learning kernel ridge regression (LKRR),[38] to predict hydrogen interaction energies on $[M_xAu_{25-x}(SCH_3)_{18} + H]^q$ ($M \in \{Pd, Cu\}$, $x \in \{0, 1\}$ and $q \in [-2, 2]$) systems. In the notation, "+H" denotes the adsorbed hydrogen atom on the nanoparticle. These systems have been experimentally shown to have catalytic activity in HER[14] and they have been studied extensively with density functional theory (DFT),[17,39] which offers the data for this study. The motivation is to construct a framework that could reliably predict interaction energies and thus could be later used to find possible catalytic sites on MPCs. At the same time, we compared ML methods with varying complexities from a single "method family", offering a fair comparison of their performance. Artificial neural networks (ANNs) are also a popular method choice in catalysis studies,[6,40−43] but in order to ensure the optimal performance, ANNs often require at least tens of thousands of data points. When data is limited, which is a common situation in catalysis research, kernel-based methods are oftentimes more reliable than ANNs. From kernel-based methods, Gaussian processes, KRR, and support vector machine are commonly used.[41,44,45]

Our graph-based representation approach is inspired by the works of Xu et al., who predicted the binding of molecules on metal surfaces,[44] and Cha et al., where graph features were used to estimate protein interactions.[46] We combine similar graph representation ideas with several kernel-based ML methods not only to predict the interaction energies but also to enable further analysis of the relevant properties dictating the nanoparticle−hydrogen interactions. In addition to features connected to the system geometry and atom types, the methodology also addresses the charge state of the system, which is a crucial parameter for electrocatalysis. Our best models reached a cross-validation RMSE of below 0.1 eV, the stability of which was confirmed by separate validation. The analysis of the multikernel method LKRR revealed the relative importance of data features, which provides interpretability for acquiring chemical knowledge about catalytic systems.

2. THEORETICAL METHODS

In this section, we present our data set and depict the graph representation and ML methods used in the study. We represented the nanoparticle−hydrogen catalytic systems as graphs, which were used to store properties and form connections between the parts of the system. The continuous Weisfeiler−Lehman (WL)[47,48] scheme was applied to propagate data around the system/graph controlling the range of the interaction information. This updated information was used as an input for the ML methods.

2.1. System and Data. The data used in this work originates from an extensive set of DFT calculations by López-Estrada et al.[17,39] for simulating the interaction of hydrogen with the atomically precise nanoparticle $[Au_{25}(SR)_{18}]^-$. The atomic system has a known crystal structure,[49,50] but a simplified ligand structure (methylthiolate) was used as a model thiolate ligand (see Figure 1a). The figure also schematically shows possible atom sites for metal exchange from Au to Pd or Cu (metal doping) and the sites of hydrogen adsorption.

The DFT calculations in refs 17,39 were run on a real-space grid-based DFT code GPAW[51,52] with the GGA-PBE functional[53] and continuum solvation model.[54] From the DFT

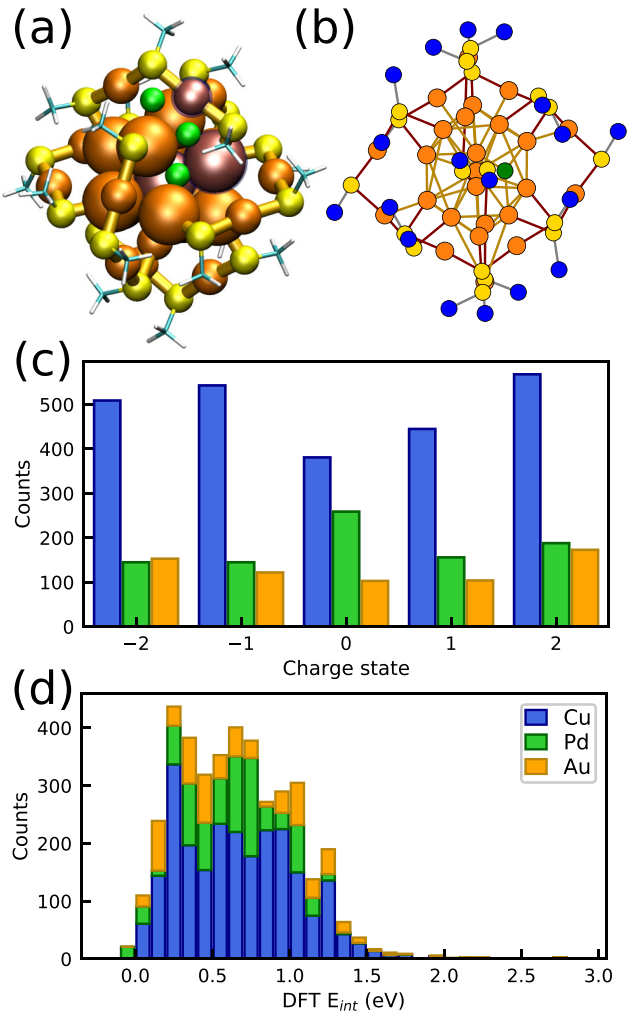


Figure 1. Studied system, graph representation, and DFT data. (a) Visualization of the $[M_xAu_{25-x}(SCH_3)_{18} + H]^q$ ($M \in \{Pd, Cu\}$, $x \in \{0, 1\}$ and $q \in [-2, 2]$) system, where different metal doping and hydrogen adsorption sites are highlighted. Orange: gold; yellow: sulfur; turquoise: carbon; white: methyl hydrogen. Violet-tinted gold atoms point to the example dopant location types. Three green spheres show examples of H adsorption sites. (b) A graph representing the nanoparticle and an adsorbed hydrogen atom. Orange: gold (or other metal atoms); yellow: sulfur; blue: methyl; green: adsorbed hydrogen. (c) Distribution of DFT data points based on the total charge q of the system. (d) Distribution of DFT data points based on the nanoparticle−hydrogen interaction energy E_{int}.

data, we evaluated the nanoparticle−hydrogen interaction energy as

$$E_{int} = E(\mathrm{MPC} + \mathrm{H}) - E(\mathrm{MPC}) - \frac{1}{2}E(\mathrm{H_2}) \quad (1)$$

where $E(\mathrm{MPC} + \mathrm{H})$ is the energy of the nanoparticle with adsorbed hydrogen, $E(\mathrm{MPC})$ is the energy of the optimized nanoparticle, and $E(\mathrm{H_2})$ is the energy of the hydrogen molecule.

In the DFT calculations, a hydrogen atom was initially placed in the vicinity of the desired binding site, and the whole system was optimized until the hydrogen atom was bound to the nanoparticle. These optimization trajectories (atomic structure vs total energy) served as our data set containing in total 3994 configurations, of which 655 structures contained

https://doi.org/10.1021/acs.jpcc.3c02539
J. Phys. Chem. C 2023, 127, 14211−14221

only gold atoms in the nanoparticle, 2446 systems had one Au changed to Cu, and 893 systems included the change of one Au to Pd atom. The nanoparticle may also have a total charge in the range of $-2 \leq q \leq 2$ as follows: 807 systems with −2, 810 with −1, 705 with +1, and 929 with +2 charge and 743 neutral systems. The distributions of the data with respect to the metal content, total charge, and E_{int} are shown in Figure 1c,d.

2.2. Graph Representation of the Nanoparticle−Hydrogen System and Continuous Weisfeiler−Lehman Scheme. The nanoparticle−hydrogen systems were represented as graphs based on their atomistic structure. Au, Cu, Pd, and S atoms as well as methyl (CH_3) groups were considered the nodes of the graph. The nodes were connected via edges if they were within the cutoff radii listed in Supporting Information Table S1. The cutoff radii were selected such that every node was connected to at least one neighboring node. A visualization of the graph is shown in Figure 1b. Every node had a set of attributes, which contains geometric, graph theoretical, and tabulated properties, hence termed GGT features (geometry, graph, tabulated).

As the geometric features, we used the minimum inaccessible radius,[55] accessible shell volume,[55] and Osipov−Pickup−Dunmur chirality indices,[56] calculated with five and seven nearest neighbors. The graph theoretical features were Ollivier−Ricci and Forman−Ricci curvatures,[57−60] multifractal dimensionality (also called the "box-counting dimension"),[61] and Gaussian network mode square sums[62,63] (for further details, see Supporting Information Sections 1.2−1.6). The tabulated features contained atomic numbers, masses, covalent radii, and electronic configurations. The electronic configuration is listed as electron occupations in shells (3d, 4s, 4p, 4d, 5s, 5p, 6s, 4f, 5d). The electronic configuration further distinguished Au, Pd, and Cu atoms from each other.

Graphs and graph kernels could be used directly to represent the MPCs.[64] However, chemical intuition implies that hydrogen adsorption is a "local" chemical event, modifying atomistic interactions only in the vicinity of the adsorption site. Therefore, we utilized the graph structure in the context of the Weisfeiler−Lehman (WL) scheme[47] in its continuous form,[48] collecting the relevant information on the graph node representing the adsorbed hydrogen atom. As shown in ref 65, the WL scheme is at least as good in separating the nonisomorphic graphs as the popular graph neural network architectures.

We start with a graph g containing nodes (vertexes) v. Every node v has the initial attribute $a^0(v)$, composed of the GGT features as described above. In the WL scheme, the node attributes are updated iteratively as

$$a^{i+1}(v) = \frac{1}{2}\left(a^i(v) + \frac{1}{\deg(v)} \sum_{u \in \mathcal{N}(v)} w(v, u) a^i(u)\right) \tag{2}$$

where the superscript i refers to the WL iteration in question. Notation $\deg(v)$ tells the degree of the node v, and $\mathcal{N}(v)$ denotes the neighboring nodes of the v. Weights $w(v, u)$ for edges connecting nodes v and u are set here to unity. The final node attributes are collections of h WL iterations, $a_{\text{final}}(v) = [a^0(v), a^1(v), \cdots, a^{h-1}(v), a^h(v)]$, where $a^0(v)$ are the initial attribute values. Here, we use the notation WL-i to indicate the attributes from the ith WL update, and WL-i-j denotes the collection of attributes from ith up to the jth update ($i, j \in [0, h]$). In this study, the updated hydrogen node attributes serve as the input to ML methods.

Aside from the chemical intuition, our approach is motivated also by the previous works by Jäger et al., who showed that the local descriptors are more suitable for prediction of the hydrogen adsorption events than the global descriptors,[66] and by Xu et al., who used the WL scheme to predict binding energies of molecules on metal surfaces.[44] The work by Linja et al. also stressed the importance of selecting only the most useful features/variables to optimize the performance and accuracy of the ML models,[35] thus supporting the idea of using only the most important node of the graph.

2.3. Extreme Minimal Learning Machine and Kernel Ridge Regression. EMLM and KRR share the same prediction model structure, but EMLM uses an Euclidean distance kernel, while KRR can use any kernel. In this study, we use both EMLM and KRR derived in a similar manner as presented by Kärkkäinen.[34] The input data is contained in a vector $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N \in \mathbb{R}^{N \times n_x}$, and the corresponding output data is in $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N \in \mathbb{R}^{N \times n_y}$. From $\mathbf{X}$, we sample M reference points into $\mathbf{Q} = \{\mathbf{q}_j\}_{j=1}^m \in \mathbb{R}^{M \times n_x}$. For every ML method in this study, we used the RS-maximin method[33,67] to sample the reference points. RS-maximin picks the data point closest to the mean as an initial reference, and then the following points are required to maximize the Euclidean distances to the previously selected ones. This forms an even sampling over the data set. The training of the EMLM and KRR is done via regularized least-squares as

$$\min_{\mathbf{W} \in \mathbb{R}^{M \times n_y}} J(\mathbf{W}) = \frac{1}{2N} \sum_{i=1}^{N} |\mathbf{d}_i^T \mathbf{W} - \mathbf{y}_i^T|^2 + \frac{\beta}{2M} \sum_{i=1}^{K} \sum_{j=1}^{n_y} |W_{ij}|^2 \tag{3}$$

Here, $\mathbf{d}_i \in \mathbb{R}^M$ is a vector containing kernel values between the ith input and every M reference. The aim is to find the optimal weight matrix $\mathbf{W} \in \mathbb{R}^{M \times n_y}$, which minimizes the prediction error and satisfies the regularization determined by the parameter β. Equation 3 can be solved by writing it in the matrix form and finding the zero of the first derivative. Hence

$$\frac{1}{N}\mathbf{D}^T(\mathbf{D}\mathbf{W} - \mathbf{Y}) + \frac{\beta}{K}\mathbf{W} = 0 \tag{4}$$

$$\left(\mathbf{D}^T\mathbf{D} + \frac{\beta}{K}\mathbf{I}\right)\mathbf{W} = \mathbf{D}^T\mathbf{Y} \tag{5}$$

which is straightforward to calculate numerically. The prediction for output $\mathbf{p}$ is calculated as $\mathbf{p}^T = \mathbf{d}^T\mathbf{W}$. In this study, we use the kernel function of type $1/r$

$$\phi(\mathbf{x}_1, \mathbf{x}_2, a) = \frac{1}{a|\mathbf{x}_2 - \mathbf{x}_1| + 1} \tag{6}$$

for KRR. The parameter a was decided during the cross-validation. EMLM uses the Euclidean distance as a kernel function.

2.4. Learning Kernel Ridge Regression. When using multiple kernels in a kernel-based ML method, there are many choices for a base learner.[68] In this study, we used the KRR-based method developed by Cortes et al.[38] This, so-called learning kernel ridge regression (LKRR), can be derived

https://doi.org/10.1021/acs.jpcc.3c02539
J. Phys. Chem. C 2023, 127, 14211−14221

similarly to a normal KRR. We include a detailed derivation in the Supporting Information Section 2.1. For the sake of understanding the parameters of the method, it is enough to consider the following equation

$$\left(\sum_{i=1}^{N_k} \mu_i \mathbf{G}_i + \beta \mathbf{I}\right)\alpha = \mathbf{Y}, \text{ where } \mu_i = \mu_{0,i} + \Lambda \frac{v_i}{|\mathbf{v}|} \tag{7}$$

which is used iteratively to solve Lagrangian multipliers in the vector $\alpha \in \mathrm{R}^{N \times n_y}$. N is the number of training data points, and N_k is the number of kernel functions. $\mathbf{Y}$ contains the expected outputs, as before. $\mathbf{G}_i \in \mathrm{R}^{N \times N}$ is a Gram matrix, which contains all of the dot products between training data points projected into the ith kernel space. The elements of vector $\mathbf{v} \in \mathrm{R}^{N_k}$ are defined as $v_i = \alpha^T \mathbf{G}_i \alpha$. The μ_0 vector is a base combination vector, the values of which are parameters, β is the model regularization, and Λ is a sensitivity parameter for kernel weighting. The kernel weights μ_i are learned during the training process. The final weight matrix is calculated as

$$\mathbf{W} = \sum_{i=1}^{N} \left(\alpha_i \sum_{j=1}^{N_k} \mu_j \phi_j(\mathbf{x}_i)\right) \tag{8}$$

The output is predicted as $p = \left[\sum_{j=1}^{N_k} \mu_j \phi_j(\mathbf{x})\right]^T \mathbf{W}$. We used the same $1/r$-type kernel for the LKRR as shown in eq 6.

During testing, we observed that the LKRR method has a tendency to contain a linear displacement. Hence, we added a linear correction at the end of the training. When the model weight matrix $\mathbf{W}$ and kernel weighting μ are solved, we predict the outputs for the training data and make a linear correction of form $y = s \cdot p + c$, where y is the expected result, p is a model prediction, and s and c are the slope and the constant, respectively. Everything is done based on the training data. The linear correction is discussed in the Supporting Information Section 2.2.

2.5. Minimal Learning Machine. The minimal learning machine (MLM), presented originally by de Souza et al.[32] and thoroughly formalized in reference,[33] differs from EMLM, KRR, and LKRR by its fundamental way of forming predictions. The MLM contains reference points from both input and output spaces, and it forms a regression between input and output space distances (similarities) with respect to reference points. The starting point is exactly the same as for EMLM and KRR but one has also the output space reference points $\mathbf{T} = \{\mathbf{t}_j\}_{j=1}^{M} \in \mathbb{R}^{M \times n_y}$. The aim is to form a regression between distance (similarity) spaces as

$$\mathbf{D}_{\text{out}} = \mathbf{D}_{\text{in}}\mathbf{B} + \epsilon \tag{9}$$

Here, $\mathbf{D}_{\text{in}} \in \mathbb{R}^{N \times M}$ contains the kernel values between training data and reference points in the input space. $\mathbf{D}_{\text{out}} \in \mathbb{R}^{N \times M}$ contains similarity measures in the output space. We used the Euclidean distance as a kernel function the same way as in an original MLM. The residue ϵ is assumed to be small. According to de Souza et al., the weight matrix $\mathbf{B} \in \mathbb{R}^{M \times M}$ is solved as[32]

$$\mathbf{B} = (\mathbf{D}_{\text{in}}^T \mathbf{D}_{\text{in}})^{-1} \mathbf{D}_{\text{in}}^T \mathbf{D}_{\text{out}} \tag{10}$$

The output prediction is done in two parts. First, the output space distances are predicted from the input space distances in the same way as in eq 9. The output is solved from a multilateration problem discussed in the Supporting Information Section 3.

3. RESULTS AND DISCUSSION

In this section, we present how data was processed and the main findings for our ML methods. The performance was determined via a 5-fold cross-validation and separate validation.

3.1. Data Preparation. We formed graph structures as described above and applied the WL updates five times. All input data features were minmax-scaled in [0, 1], and constant variables were excluded. The interaction energies were minmax-scaled in [−1, 1]. Before forming the 5-fold cross-validation sets, 414 data points were separated for validation. Validation and cross-validation sets were sampled randomly, but stratification with respect to the relative amount of different charge states was performed.

In order to utilize the weighting of the multikernel LKRR, we split the input data in five different ways. Initially, all input features are kept in a single vector, which then contains the charge and the GGT features from the WL updates up to the fifth iteration. This is considered as a splitting scheme 0. Scheme 0 is used for all ML methods but the four following ones only with the LKRR. In the splitting scheme 1, charge is represented with its own set of kernels. The charge of the system has a significant effect on the energies; hence, we wanted to see whether the LKRR can address this. The splitting scheme 2 separates the charge and different WL-0-j updates, so that the inputs contain the whole WL update history up to the given level. The splitting scheme 3 is similar to scheme 2 but it uses the WL-i updates, therefore considering only the current WL updates without the whole update history. The splitting scheme 4 has the most separation. There charge, the WL-i updates, and within the WL updates, the three GGT feature sets are separated. In schemes 1−4, the charge is handled as a categorical variable; i.e., it is represented as a unit vector. For example, the charge state −1 would be [01000] and the neutral system would be [00100]. This increases the dimensionality of the charge input. Catalytic HER reactions are performed under a voltage; thus, MPCs possess energetically favorable charge states. In simulations, the voltage effect is imitated by setting up a certain charge state, which makes it a crucial input for the ML methods.

In the LKRR, every data segment gets its own set of $1/r$ kernel functions shown in eq 6, where the parameter a spans 20 values from 0.5 to 10.0 in increments of 0.5. Hence, every data segment has 20 kernels and the number of kernels in different splitting schemes is 0:20, 1:40, 2:140, 3:140, and 4:360. For the conventional KRR, we used the same type of kernel, but parameter a was given 10 values from 1.0 to 10.0 in increments of 1.0, from which the best models were selected for further analysis.

3.2. Cross-Validation with Single Vector Features. The different levels of the concatenated WL updates were studied with the MLM, EMLM, and KRR. There were two hyperparameters tested: the number of reference points for all methods and kernel parameter a for the KRR. The learning capabilities of the kernel-based ML methods are determined by the reference points sampled from the training data. During the prediction, kernel function(s) measure the similarities between an input and reference points forming a kernel vector/matrix, which is used to perform regression. The regularization

https://doi.org/10.1021/acs.jpcc.3c02539
J. Phys. Chem. C 2023, 127, 14211−14221

coefficient for the EMLM and KRR was set equal to $\sqrt{\epsilon_{machine}}$.

The average test RMSEs for the KRR are shown in Supporting Information Figures S6−S11 as a function of kernel parameter a and the relative number of reference points with respect to the WL-0-i updates. Setting a = 1.0 produced the most accurate models, and this value is used for the KRR from here on. The average cross-validation test RMSEs are visualized in Figure 2 and in Supporting Information Figure S12. As illustrated in Figure 2, the first WL update is almost enough to reach the highest accuracy. There are only minor improvements visible after adding further updates. The method type also affects how much the method benefits from the WL updates. In Figure 2b,c, we see that the EMLM is able to lower the RMSE even if the KRR could not. In Supporting Information Table S3, the maximum, minimum, and average RMSEs are tabulated for WL-0-5, which shows that the MLM is overall the most accurate method by a small margin.

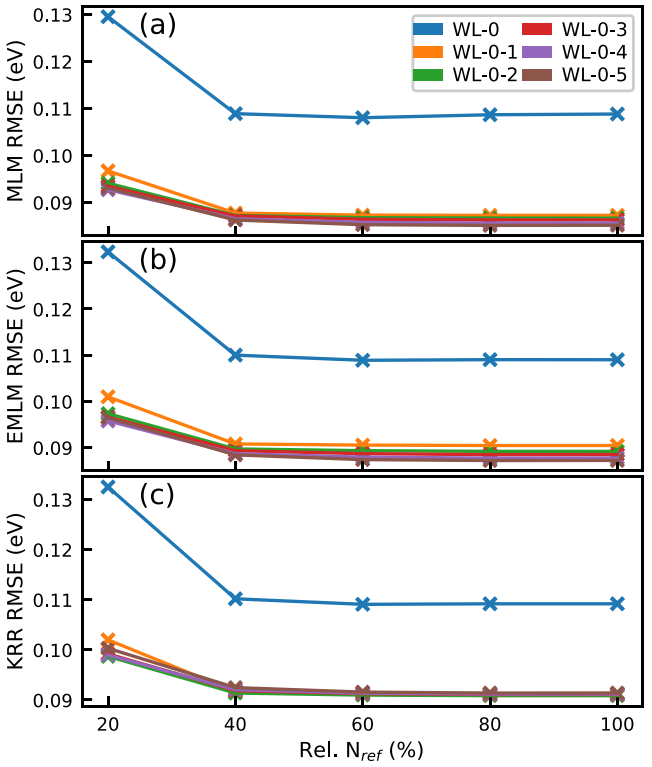


Figure 2. Average cross-validation RMSEs with different WL updates. (a) MLM, (b) EMLM, and (c) KRR models with kernel parameter a = 1.0 as a function of the relative number of reference points. In every model, charge and GGT data up to the designated WL level are used as a single input.

The effect of the WL updates supports chemical intuition about the locality of the H adsorption. Knowing only the environment of the hydrogen atom is not enough, but when information about the environments of the nearest neighbors is included, the model performance is enhanced significantly. However, information about the second nearest or further atoms is not equally vital. This highlights the locality of hydrogen chemisorption on the studied nanoparticles. Furthermore, the effect of WL updates agrees with the observations of Xu et al., who showed that kernel using WL updates and Wasserstein metrics on graphs outperformed conventional radial basis function kernels in Gaussian processes.[44]

Next, we tested the LKRR similarly with splitting scheme 0. The LKRR has three main hyperparameters: number of reference points, model regularization β, and kernel sensitivity Λ. The initial combination vector μ_0, which affects the weighting of the kernels, was initialized with $\sqrt{\epsilon_{machine}}$. We ran the cross-validation as before with the relative numbers of reference points [20%, 40%, 60%, 80%, 100%], $\beta \in \{0.001, 0.01, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5\}$, and $\Lambda \in \{0.5, 1.0, 5.0, 10.0, 15.0, 20.0\}$. The cross-validation test RMSEs are shown in Supporting Information Figures S14−S18. It has to be noted that not all of the LKRR models managed to converge with every set of parameters. We picked three representative models with (β, Λ) parameters numbered 1: (0.5, 1.0), 2: (1.5, 0.5), and 3: (0.75, 20.0).

The trained and tested RMSEs for these three LKRR models are visualized in Figure 3. The LKRR has a reasonable performance with about 0.15 eV RMSE, which is higher than with the single kernel methods. Interestingly, contrary to the previous methods, the LKRR has a clear optimal amount of reference points. This is caused by the two sets of weights: model weights and kernel weights. The relative importance of the kernels, shown in Supporting Information Figure S13, supports this. When the number of reference points increases, the kernel weighting becomes more linear and less flexible. The number of reference points determines the learning capabilities of a kernel method; hence, it can be thought that with a low number of reference points, the model compensates for the performance with the kernel weighting. Furthermore, the kernel weighting has only a minor variation across the cross-validation sets proving the LKRR to be stable.

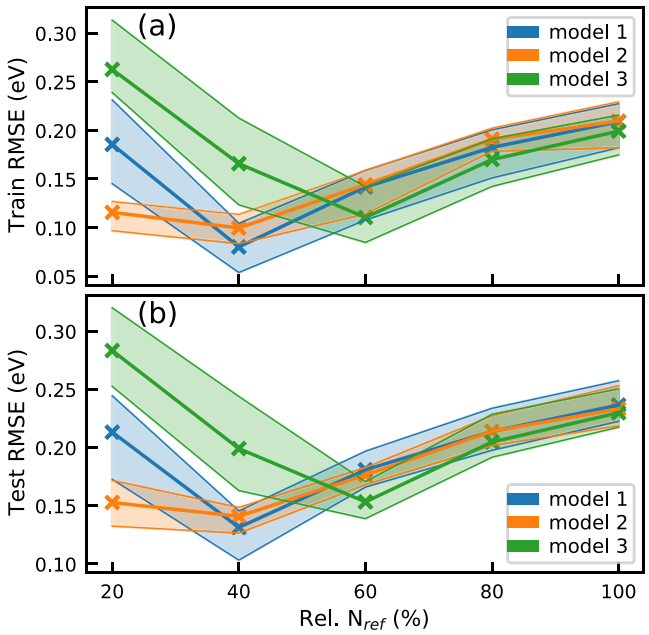


Figure 3. Performance of three LKRR models. (a) Training RMSEs and (b) test RMSEs are visualized for three LKRR models as a function of the relative number of reference points. The model parameters (β, Λ) are as follows: model 1: (0.5, 1.0), model 2: (1.5, 0.5), and model 3: (0.75, 20.0). Shaded areas highlight the maximum and minimum values.

https://doi.org/10.1021/acs.jpcc.3c02539
J. Phys. Chem. C 2023, 127, 14211−14221

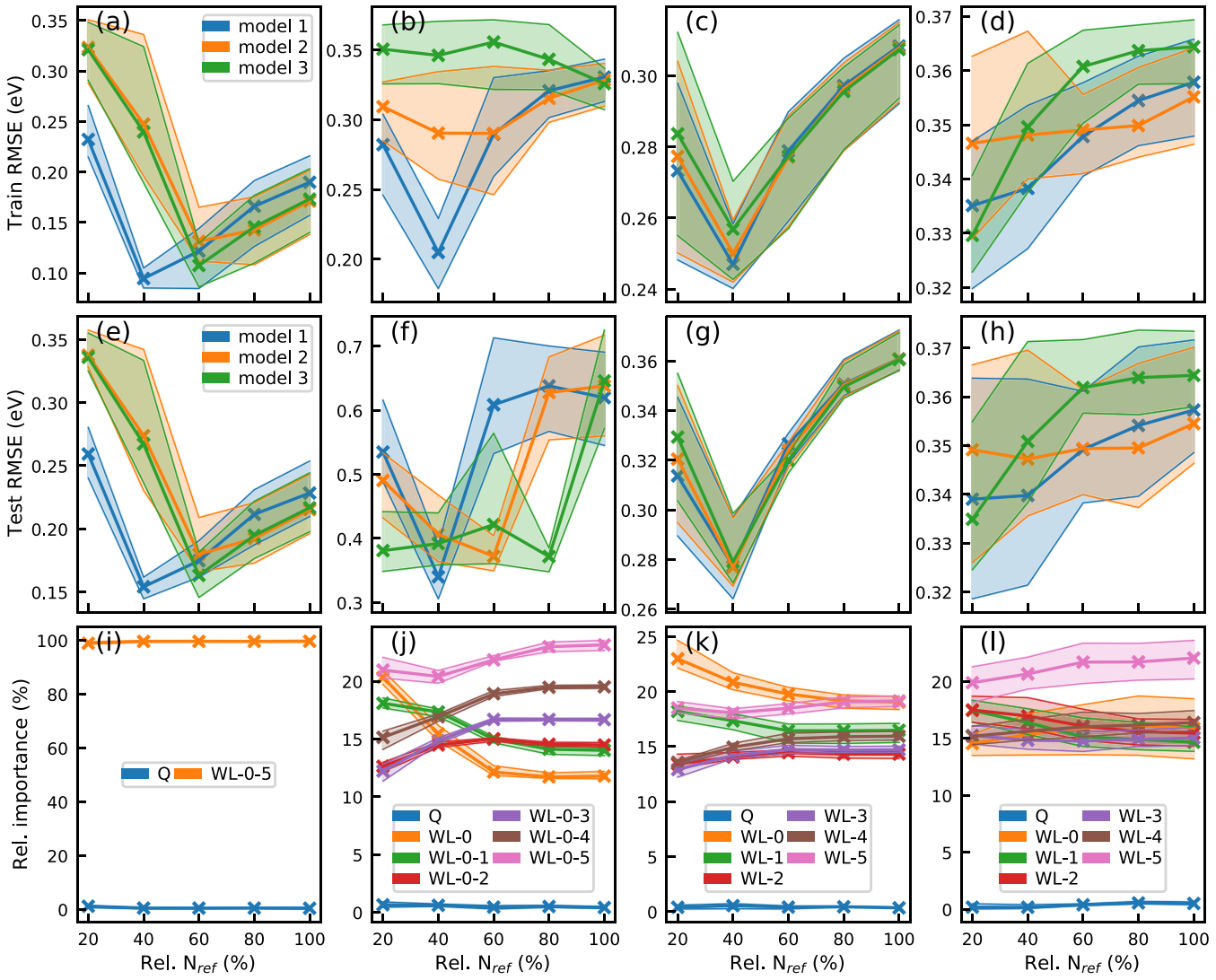


Figure 4. Effect of data splitting on the LKRR performance. For splitting scheme 1, train RMSE is shown in panel (a), test RMSE in panel (e), and relative importances for different kernels (i). Panels (b), (f), and (j) are the corresponding visualizations for splitting 2. Panels (c), (g), and (k) are for splitting 3. Panels (d), (h), and (l) are for splitting 4. Relative importances are always shown for the models labeled as "model 1". The label Q refers to the charge kernels. The β and Λ parameters of the models are listed in Supporting Information Table S4. Shaded areas highlight the maximum and minimum values.

3.3. Data Splitting with LKRR. Next, the input data were split into segments. The LKRR hyperparameters were tested as before with every splitting scheme, and the results are shown in Supporting Information Figures S19–S38. The cross-validation training and test RMSEs are visualized in Figure 4a–h for representative models, whose parameters are listed in Supporting Information Table S4. The splitting scheme shows a compelling effect on the model performance in Figure 4, visualizing minimum, maximum, and average RMSEs. Corresponding test RMSE values are listed for "model 1"s in Supporting Information Table S5. The best-performing models are the ones with the simplest data splitting (scheme 1: charge and WL-0-5) in Figure 4a,e. In the best cases, the average test RMSE reached 0.15 eV.

The initial impression about the relative importances in Figure 4i–k is positive because the differences between the highest and the lowest values are small, implying a systematic behavior. The splitting scheme 4 in Figure 4l caused more variation than the others, which is expected, as it has the most kernels. The common characteristic is that the relative importance of the charge is low. This is anticipated because it is just a single discrete property and should not dictate too strongly the E_{int}. Figure 4 demonstrates clearly how the LKRR emphasizes different features depending on the reference points. The WL-0-*i* contains all updates up to the *i*th level; hence, it is expected that in Figure 4j the WL-0-5 gets always the highest priority. The WL-0 is interesting in the same figure because it initially gets high importance but is then reduced to the least-weighted WL feature when the reference points are added. This again highlights the locality of the hydrogen adsorption because WL-0 contains only information about the local geometric environment of the hydrogen atom. The locality is also supported by the similar high weighting in Figure 4k. Another plausible origin is the dimensionality of the kernel space where the regression is performed. In order to extract the necessary information from WL-0-5 and WL-5, the LKRR needs enough reference points.

The data splitting 4 could also be analyzed in terms of the GGT features. In Figure 5, the relative importances of these data types are visualized for the LKRR model 1 from Figure

https://doi.org/10.1021/acs.jpcc.3c02539
J. Phys. Chem. C 2023, 127, 14211−14221

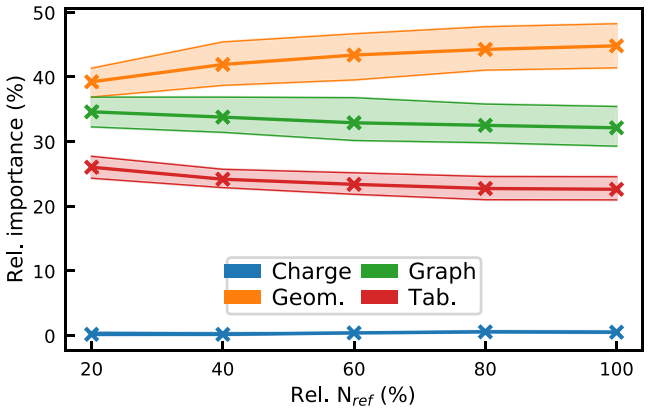


Figure 5. Relative importances of the GGT features. The curves show the relative importances of the charge and GGT features of the LKRR model 1 ($\beta = 0.001$, $\Lambda = 0.5$) in Figure 4d,h,i using splitting scheme 4. Shaded areas highlight the maximum and minimum values.

4d,h,l. We observed that the geometric features get weighted the most, then the graph theoretical ones, and finally the tabulated features. This agrees with the findings of Cha et al. who also reported similar conclusions for the protein−protein interaction.[46] However, one has to keep in mind that tabulated features have 20 kernels less than the other GGT features due to the deletion of the constants.

3.4. Validation Results. The validations were run with representative parameters acquired from the tests above. For MLM, EMLM, and KRR, we used the WL-0-5 updates and 60% of the training data as reference points. For the LKRR models, only 40% of the training data was used as reference points and the other parameters correspond to the ones listed as "model 1"s in Figures 3 and 4. All of the validation predictions were averaged over the sets of five models trained with the respective cross-validation sets.

The validation results in Figure 6 show that MLM and EMLM are easily the best-performing models with the KRR in close pursuit. The LKRR with splitting schemes 0 and 1 shows similar performance. However, the splitting proves to be a "double-edged sword". Even if it enables the analysis of different data sources, it also causes prediction errors. Changing from splitting scheme 0 to 1 induced a slightly larger RMSE. Comparison of Figure 6f,g demonstrates how differently the LKRR behaves with WL-0-*i* and WL-*i* inputs. Using only current WL-*i* updates, the lower RMSE is reached, but we observe clear shifts depending on the charge. This refers to the fact that the charge state and the structure play an intertwined role in adsorption. Handling charge separately is able to highlight its property as a global feature but it hides some underlying connections.

It is not totally straightforward to compare our model performance if the studied catalytic systems are different, but we could get some qualitative ideas from the literature. Chen et al. estimated the CO adsorption on bare gold nanoparticles with 168,419 gold atoms using ANNs.[42] They used 1104 data points for training and 140 data points for testing and validation, respectively. At best, they reached RMSE between 0.05 and 0.06 eV. Here, we have to keep in mind that their nanoparticle did not have any protecting ligands in contrast to our study. Fung et al. predicted binding energies of H atom on nitrogen-doped graphene single atom catalysts using just 108 data points with various ML methods.[45] Their data set was small, and test RMSE values varied from 0.218 to 0.366 eV. Xu et al. predicted binding energies of molecules on different metal surfaces using around 1700 data points.[44] Their best Gaussian process regressor using WL updates and Wasserstein

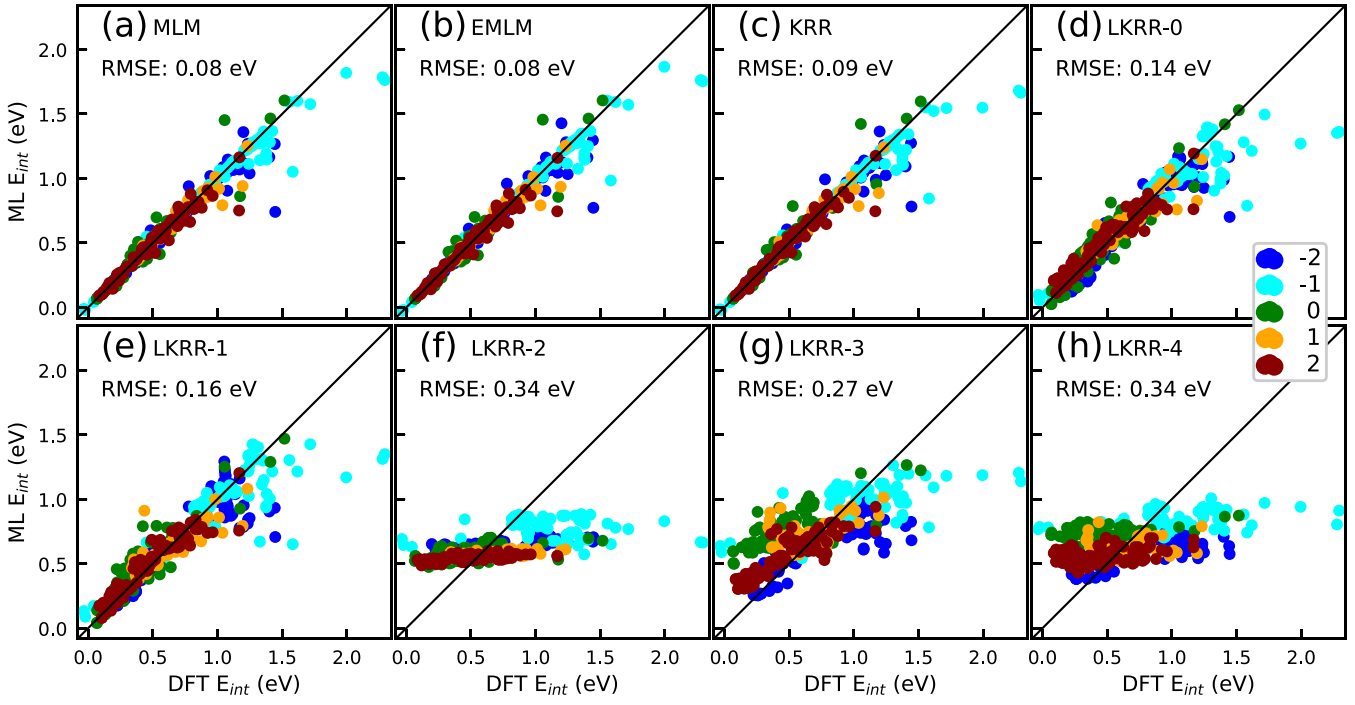


Figure 6. Averaged validation predictions for all ML methods. (a) MLM, (b) EMLM, and (c) KRR ($a = 1.0$) use WL-0-5 and 60% of the training data as reference points. LKRR-0 in panel (d) has not gone through data splitting. In panels (e−h), LKRR models utilize corresponding data splitting schemes. All LKRR models had 40% of the training data as reference points, and the hyperparameters corresponded to the models labeled as "model 1"s in previous analysis cases. The data points are visualized by the system charge per the color labels.

https://doi.org/10.1021/acs.jpcc.3c02539
J. Phys. Chem. C 2023, 127, 14211−14221

metrics as a kernel managed to reach an RMSE of 0.18 eV. Chowdhury et al. used various representations and ML methods to predict binding energies of small molecules with a maximum of four carbon atom backbones on the Pt(111) surface.[41] Their data set was also small, less than 300 data points. Their test mean absolute error (MAE) varied between 0.14 and 0.40 eV when the test set contained similar-sized molecules as in the training set, i.e., models were interpolating. With smaller molecules, i.e., models were at least partially extrapolating, MAEs varied between 0.20 and 3.50 eV. In light of these results, one can conclude that our data set and model performance are on par with recent studies.

4. CONCLUSIONS

In this study, we created a simple graph representation about $[M_xAu_{25-x}(SCH_3)_{18}]^q$ (M ∈ { Pd, Cu }, $x \in \{0, 1\}$, and $q \in [-2, 2]$) structures and utilized the continuous WL scheme to update node attributes, which were used to predict interaction energies with various kernel-based ML methods. Graph representation encoded geometric, graph theoretical, and tabulated features about the atomic system, but the methodology was also able to address the charge state of the system. Catalytic reactions are regularly studied on surfaces, substrates, or large metal particles; thus, the charge of the system is ambiguous. However, electrocatalysis is performed under a voltage and nanoparticles will opt for a charge state accordingly. Hence, in the simulations, the charge is used to imitate the effect of the voltage. Being able to implement physically and chemically meaningful properties into the representation is highly useful because it enables the analysis of model behavior and feature effects on the level, which can be directly mapped back into the real chemical setting.

From the machine learning point of view, our data sets were limited in size, which justified the kernel-based approach. Moreover, a similar size of the training data as here, only thousands of observations, has been used with multilabel classification problems.[69−71] Both shallow and deep neural networks (NNs) are popular tools in catalysis.[6,40−43] However, NNs are known to be "data hungry", often requiring at least a few tens of thousands of training data points to reach their optimal performance, thus further justifying the method choice. Furthermore, several non-NN-based methods have also shown to be able to predict binding behavior accurately for molecules and hydrogen on various surfaces and doped graphene.[44,45,72−75]

From our kernel-based methods, the MLM, EMLM, and KRR reached the highest accuracy, and the analysis showed that applying WL updates only once was enough for accurate predictions. This demonstrated that the imminent surroundings of the hydrogen atom dictate adsorption. We also explored the multikernel method LKRR and how it could be utilized to split the features into separate kernels. The importance of the data features was evaluated based on the kernel weighting learned by the LKRR. We found out that the geometric and graph theoretical features proved to be more meaningful than tabular information about atom properties. Geometric properties describe how accessible/exposed an atom is, and graph theoretical features encode the connectivity of the node, i.e., how much an atom interacts with its neighbors; hence, this highlights the local nature of the hydrogen−nanoparticle interaction. We also observed that the charge state is closely linked to the structural features. Separating the charge enabled us to address it as a global feature, but it lost some key connections. In conclusion, we demonstrated that the combination of graphs and kernel ML offers powerful tools to find relevant information for a high-dimensional catalysis problem with a limited number of data. This study also promotes simple ML methods as analysis tools for complex problems in nanoscience and attempts to advance interpretable ML.

■ ASSOCIATED CONTENT

Data Availability Statement

The MLM, EMLM, KRR, and LKRR have been programmed using Python3 with Numpy,[76] Scipy,[77] Scikit-Learn,[78] Networkx,[79] Atomic Simulation Environment (ASE),[80] and MPI4Py[81−84] for parallelization of the LKRR cross-validation. The code used in this study is available at 10.17011/jyx/dataset/87521. More information and access to the up-to-date repository are found at 10.17011/jyx/dataset/87525. The training data (from DFT calculations in ref 17) is available in ref 39.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jpcc.3c02539.

Detailed description of graph attributes and construction, theory and the parameters of the LKRR method, multilateration problem of the MLM, parameter test results for KRR and LKRR, and additional cross-validation results (PDF)

Special Issue Paper

Published as part of *The Journal of Physical Chemistry C* virtual special issue "Machine Learning in Physical Chemistry Volume 2".

■ AUTHOR INFORMATION

Corresponding Author

Hannu Häkkinen − *Department of Physics, Nanoscience Center, University of Jyväskylä, FI-40014 Jyväskylä, Finland; Department of Chemistry, Nanoscience Center, University of Jyväskylä, FI-40014 Jyväskylä, Finland;* orcid.org/0000-0002-8558-5436; Email: hannu.j.hakkinen@jyu.fi

Authors

Antti Pihlajamäki − *Department of Physics, Nanoscience Center, University of Jyväskylä, FI-40014 Jyväskylä, Finland*

Sami Malola − *Department of Physics, Nanoscience Center, University of Jyväskylä, FI-40014 Jyväskylä, Finland*

Tommi Kärkkäinen − *Faculty of Information Technology, University of Jyväskylä, FI-40014 Jyväskylä, Finland*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jpcc.3c02539

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by the Academy of Finland through grants 351582 and 351579 in the EuroHPC Research Programme. Computations were done at the FCCI node in the University of Jyväskylä (persistent identifier: urn:nbn:fi:research-infras-2016072533). The authors acknowledge J. Linja for discussions on methodology and O. López-Estrada, N. Mammen, and L. Laverdure for providing the DFT data.

REFERENCES

(1) Zhao, G.; Rui, K.; Dou, S. X.; Sun, W. Heterostructures for Electrochemical Hydrogen Evolution Reaction: A Review. *Adv. Funct. Mater.* **2018**, *28*, No. 1803291.

(2) Hu, C.; Zhang, L.; Gong, J. Recent progress made in the mechanism comprehension and design of electrocatalysts for alkaline water splitting. *Energy Environ. Sci.* **2019**, *12*, 2620−2645.

(3) Sun, F.; Tang, Q.; Jiang, D.-e. Theoretical Advances in Understanding and Designing the Active Sites for Hydrogen Evolution Reaction. *ACS Catal.* **2022**, *12*, 8404−8433.

(4) Toyao, T.; Maeno, Z.; Takakusagi, S.; Kamachi, T.; Takigawa, I.; Shimizu, K.-i. Machine Learning for Catalysis Informatics: Recent Applications and Prospects. *ACS Catal.* **2020**, *10*, 2260−2297.

(5) Ma, S.; Liu, Z.-P. Machine Learning for Atomic Simulation and Activity Prediction in Heterogeneous Catalysis: Current Status and Future. *ACS Catal.* **2020**, *10*, 13213−13226.

(6) Guan, Y.; Chaffart, D.; Liu, G.; Tan, Z.; Zhang, D.; Wang, Y.; Li, J.; Ricardez-Sandoval, L. Machine learning in solid heterogeneous catalysis: Recent developments, challenges and perspectives. *Chem. Eng. Sci.* **2022**, *248*, No. 117224.

(7) Sulley, G. A.; Montemore, M. M. Recent progress towards a universal machine learning model for reaction energetics in heterogeneous catalysis. *Curr. Opin. Chem. Eng.* **2022**, *36*, No. 100821.

(8) Liu, C.-Y.; Senftle, T. P. Finding physical insights in catalysis with machine learning. *Curr. Opin. Chem. Eng.* **2022**, *37*, No. 100832.

(9) Schlexer Lamoureux, P.; Winther, K. T.; Garrido Torres, J. A.; Streibel, V.; Zhao, M.; Bajdich, M.; Abild-Pedersen, F.; Bligaard, T. Machine learning for computational heterogeneous catalysis. *ChemCatChem* **2019**, *11*, 3581−3601.

(10) Back, S.; Tran, K.; Ulissi, Z. W. Toward a design of active oxygen evolution catalysts: insights from automated density functional theory calculations and machine learning. *ACS Catal.* **2019**, *9*, 7651−7659.

(11) Musa, E.; Doherty, F.; Goldsmith, B. R. Accelerating the structure search of catalysts with machine learning. *Curr. Opin. Chem. Eng.* **2022**, *35*, No. 100771.

(12) Hu, G.; Wu, Z.; Jiang, D.-e. Stronger-than-Pt hydrogen adsorption in a Au22 nanocluster for the hydrogen evolution reaction. *J. Mater. Chem. A* **2018**, *6*, 7532−7537.

(13) Deng, C.; Li, F.; Tang, Q. Electrocatalytic Oxygen Reduction Reaction over the Au22(L8)6 Nanocluster with Promising Activity: A DFT Study. *J. Phys. Chem. C* **2019**, *123*, 27116−27123.

(14) Kumar, B.; Kawawaki, T.; Shimizu, N.; Imai, Y.; Suzuki, D.; Hossain, S.; Nair, L. V.; Negishi, Y. Gold nanoclusters as electrocatalysts: size, ligands, heteroatom doping, and charge dependences. *Nanoscale* **2020**, *12*, 9969−9979.

(15) Kawawaki, T.; Kataoka, Y.; Hirata, M.; Iwamatsu, Y.; Hossain, S.; Negishi, Y. Toward the creation of high-performance heterogeneous catalysts by controlled ligand desorption from atomically precise metal nanoclusters. *Nanoscale Horiz.* **2021**, *6*, 409−448.

(16) Jin, R.; Li, G.; Sharma, S.; Li, Y.; Du, X. Toward Active-Site Tailoring in Heterogeneous Catalysis by Atomically Precise Metal Nanoclusters with Crystallographic Structures. *Chem. Rev.* **2021**, *121*, 567−648.

(17) López-Estrada, O.; Mammen, N.; Laverdure, L.; Melander, M. M.; Häkkinen, H.; Honkala, K. *Computational Criteria for Hydrogen Evolution Activity on Ligand-protected Au_{25}-based Nanoclusters*. chemRxiv: DOI: 10.26434/chemrxiv-2023-6zm7w (submitted 2023−01−03, accessed 2023−05−26).

(18) Tsukuda, T.; Häkkinen, H. *Protected Metal Clusters: From Fundamentals to Applications*; Elsevier: Amsterdam, Netherlands, 2015.

(19) Shibu, E. S.; Muhammed, M. A. H.; Tsukuda, T.; Pradeep, T. Ligand Exchange of Au25SG18 Leading to Functionalized Gold Clusters: Spectroscopy, Kinetics, and Luminescence. *J. Phys. Chem. C* **2008**, *112*, 12168−12176.

(20) Srisombat, L.-o.; Park, J.-S.; Zhang, S.; Lee, T. R. Preparation, Characterization, and Chemical Stability of Gold Nanoparticles Coated with Mono-, Bis-, and Tris-Chelating Alkanethiols. *Langmuir* **2008**, *24*, 7750−7754.

(21) Maity, P.; Xie, S.; Yamauchi, M.; Tsukuda, T. Stabilized gold clusters: from isolation toward controlled synthesis. *Nanoscale* **2012**, *4*, 4027−4037.

(22) Sokołowska, K.; Malola, S.; Lahtinen, M.; Saarnio, V.; Permi, P.; Koskinen, K.; Jalasvuori, M.; Häkkinen, H.; Lehtovaara, L.; Lahtinen, T. Towards Controlled Synthesis of Water-Soluble Gold Nanoclusters: Synthesis and Analysis. *J. Phys. Chem. C* **2019**, *123*, 2602−2612.

(23) Malola, S.; Nieminen, P.; Pihlajamäki, A.; Hämäläinen, J.; Kärkkäinen, T.; Häkkinen, H. A method for structure prediction of metal-ligand interfaces of hybrid nanoparticles. *Nat. Commun.* **2019**, *10*, No. 3973.

(24) Wang, S.; Wu, Z.; Dai, S.; Jiang, D.-e. Deep Learning Accelerated Determination of Hydride Locations in Metal Nanoclusters. *Angew. Chem., Int. Ed.* **2021**, *60*, 12289−12292.

(25) Liu, C.-Y.; Yuan, S.-F.; Wang, S.; Guan, Z.-J.; Jiang, D.-e.; Wang, Q.-M. Structural transformation and catalytic hydrogenation activity of amidinate-protected copper hydride clusters. *Nat. Commun.* **2022**, *13*, No. 2082.

(26) Li, J.; Chen, T.; Lim, K.; Chen, L.; Khan, S. A.; Xie, J.; Wang, X. Deep learning accelerated gold nanocluster synthesis. *Adv. Intell. Syst.* **2019**, *1*, No. 1900029.

(27) Pihlajamäki, A.; Hämäläinen, J.; Linja, J.; Nieminen, P.; Malola, S.; Kärkkäinen, T.; Häkkinen, H. Monte Carlo Simulations of $Au_{38}(SCH_3)_{24}$ Nanocluster Using Distance-Based Machine Learning Methods. *J. Phys. Chem. A* **2020**, *124*, 4827−4836.

(28) Pihlajamäki, A.; Malola, S.; Kärkkäinen, T.; Häkkinen, H. Orientation Adaptive Minimal Learning Machine: Application to Thiolate-Protected Gold Nanoclusters and Gold-Thiolate Rings. 2022, arXiv:2203.09788v2 [physics.comp-ph]. arXiv.org e-Printarchive. https://arxiv.org/abs/2203.09788 (submitted March 18, 2022, accessed May 26, 2023).

(29) Copp, S. M.; Swasey, S. M.; Gorovits, A.; Bogdanov, P.; Gwinn, E. G. General approach for machine learning-aided design of DNA-stabilized silver clusters. *Chem. Mater.* **2020**, *32*, 430−437.

(30) Mastracco, P.; González-Rosell, A.; Evans, J.; Bogdanov, P.; Copp, S. M. Chemistry-Informed Machine Learning Enables Discovery of DNA-Stabilized Silver Nanoclusters with Near-Infrared Fluorescence. *ACS Nano* **2022**, *16*, 16322−16331.

(31) Chen, T.; Li, J.; Cai, P.; Yao, Q.; Ren, Z.; Zhu, Y.; Khan, S.; Xie, J.; Wang, X.; et al. Identification of chemical compositions from "featureless" optical absorption spectra: Machine learning predictions and experimental validations. *Nano Res.* **2022**, *15*, 4188−4196.

(32) de Souza, A. H.; Corona, F.; Barreto, G. A.; Miche, Y.; Lendasse, A. Minimal Learning Machine: A novel supervised distance-based approach for regression and classification. *Neurocomputing* **2015**, *164*, 34−44.

(33) Hämäläinen, J.; Alencar, A. S. C.; Kärkkäinen, T.; Mattos, C. L. C.; Souza Júnior, A. H.; Gomes, J. P. P. Minimal Learning Machine: Theoretical results and clustering-based reference point selection. *J. Mach. Learn. Res.* **2020**, *21*, 1−29.

(34) Kärkkäinen, T. Extreme minimal learning machine: Ridge regression with distance-based basis. *Neurocomputing* **2019**, *342*, 33−48.

(35) Linja, J.; Hämäläinen, J.; Nieminen, P.; Kärkkäinen, T. Feature selection for distance-based regression: An umbrella review and a one-shot wrapper. *Neurocomputing* **2023**, *518*, 344−359.

(36) Hoerl, A. E.; Kennard, R. W. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* **1970**, *12*, 55−67.

(37) Hoerl, R. W.Ridge Analysis 25 Years Later. In *The American Statistician*; Taylor & Francis, 1985; Vol. *39*, pp 186−192.

(38) Cortes, C.; Mohri, M.; Rostamizadeh, A. In *L2 Regularization for Learning Kernels*, the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009), Montreal, Canada, 2009; pp 109−116.

(39) López-Estrada, O.; et al. Computational Criteria for Hydrogen Evolution Activity on Ligand-Protected Au25-Based Nanoclusters. *ACS Catal.* **2023**, 8997−9006, DOI: 10.1021/acscatal.3c01065.
(40) Fung, V.; Hu, G.; Ganesh, P.; Sumpter, B. G. Machine learned features from density of states for accurate adsorption energy prediction. *Nat. Commun.* **2021**, *12*, No. 88.
(41) Chowdhury, A. J.; Yang, W.; Abdelfatah, K. E.; Zare, M.; Heyden, A.; Terejanu, G. A. A Multiple Filter Based Neural Network Approach to the Extrapolation of Adsorption Energies on Metal Surfaces for Catalysis Applications. *J. Chem. Theory Comput.* **2020**, *16*, 1105−1114.
(42) Chen, Y.; Huang, Y.; Cheng, T.; Goddard, W. A. I. Identifying Active Sites for CO2 Reduction on Dealloyed Gold Surfaces by Combining Machine Learning with Multiscale Simulations. *J. Am. Chem. Soc.* **2019**, *141*, 11651−11657.
(43) Back, S.; Tran, K.; Ulissi, Z. W. Toward a Design of Active Oxygen Evolution Catalysts: Insights from Automated Density Functional Theory Calculations and Machine Learning. *ACS Catal.* **2019**, *9*, 7651−7659.
(44) Xu, W.; Reuter, K.; Andersen, M. Predicting binding motifs of complex adsorbates using machine learning with a physics-inspired graph representation. *Nat. Comput. Sci.* **2022**, *2*, 443−450.
(45) Fung, V.; Hu, G.; Wu, Z.; Jiang, D.-e. Descriptors for Hydrogen Evolution on Single Atom Catalysts in Nitrogen-Doped Graphene. *J. Phys. Chem. C* **2020**, *124*, 19571−19578.
(46) Cha, M.; Emre, E. S. T.; Xiao, X.; Kim, J.-Y.; Bogdan, P.; VanEpps, J. S.; Violi, A.; Kotov, N. A. Unifying structural descriptors for biological and bioinspired nanoscale complexes. *Nat. Comput. Sci.* **2022**, *2*, 243−252.
(47) Shervashidze, N.; Schweitzer, P.; Van Leeuwen, E. J.; Mehlhorn, K.; Borgwardt, K. M. Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.* **2011**, *12*, 2539−2561.
(48) Togninalli, M.; Ghisu, E.; Llinares-López, F.; Rieck, B.; Borgwardt, K. Wasserstein Weisfeiler-Lehman Graph Kernels. *Adv. Neural Inf. Process. Syst.* **2019**, 6439−6449.
(49) Heaven, M. W.; Dass, A.; White, P. S.; Holt, K. M.; Murray, R. W. Crystal structure of the gold nanoparticle $[N(C_8H_{17})_4][Au_{25}(SCH_2CH_2Ph)_{18}]$. *J. Am. Chem. Soc.* **2008**, *130*, 3754−3755.
(50) Zhu, M.; Aikens, C. M.; Hollander, F. J.; Schatz, G. C.; Jin, R. Correlating the crystal structure of a thiol-protected Au_{25} cluster and optical properties. *J. Am. Chem. Soc.* **2008**, *130*, 5883−5885.
(51) Enkovaara, J.; Rostgaard, C.; Mortensen, J. J.; Chen, J.; Dułak, J.; Ferrighi, J.; Gavnholt, J.; Glinsvad, J.; Haikola, J.; Hansen, J.; et al. Electronic structure calculations with GPAW: a real-space implementation of the projector augmented-wave method. *J. Phys.: Condens. Matter* **2010**, *22*, No. 253202.
(52) Mortensen, J. J.; Hansen, L. B.; Jacobsen, K. W. Real-space grid implementation of the projector augmented wave method. *Phys. Rev. B* **2005**, *71*, No. 035109.
(53) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **1996**, *77*, No. 3865.
(54) Held, A.; Walter, M. Simplified continuum solvent model with a smooth cavity based on volumetric data. *J. Chem. Phys.* **2014**, *141*, No. 174108.
(55) Kawabata, T. Detection of multiscale pockets on protein surfaces using mathematical morphology. *Proteins: Struct., Funct., Bioinf.* **2010**, *78*, 1195−1211.
(56) Osipov, M.; Pickup, B.; Dunmur, D. A new twist to molecular chirality: intrinsic chirality indices. *Mol. Phys.* **1995**, *84*, 1193−1206.
(57) Ni, C.-C.; Lin, Y.-Y.; Gao, J.; David Gu, X.; Saucan, E. In *Ricci Curvature of the Internet Topology*, IEEE Conference on Computer Communications (INFOCOM), 2015; pp 2758−2766.
(58) Sreejith, R. P.; Mohanraj, K.; Jost, J.; Saucan, E.; Samal, A. Forman curvature for complex networks. *J. Stat. Mech.: Theory Exp.* **2016**, *2016*, No. 063206.
(59) Samal, A.; Sreejith, R.; Gu, J.; Liu, S.; Saucan, E.; Jost, J. Comparative analysis of two discretizations of Ricci curvature for complex networks. *Sci. Rep.* **2018**, *8*, No. 8650.
(60) Ni, C.-C.; Lin, Y.-Y.; Luo, F.; Gao, J. Community detection on networks with ricci flow. *Sci. Rep.* **2019**, *9*, No. 9984.
(61) Falconer, K. *Fractal Geometry: Mathematical Foundations and Applications*; John Wiley & Sons, 2004.
(62) Bahar, I.; Atilgan, A. R.; Erman, B. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding Des.* **1997**, *2*, 173−181.
(63) Haliloglu, T.; Bahar, I.; Erman, B. Gaussian Dynamics of Folded Proteins. *Phys. Rev. Lett.* **1997**, *79*, 3090−3093.
(64) Kriege, N. M.; Johansson, F. D.; Morris, C. A survey on graph kernels. *Appl. Network Sci.* **2020**, *5*, 1−42.
(65) Morris, C.; Ritzert, M.; Fey, M.; Hamilton, W. L.; Lenssen, J. E.; Rattan, G.; Grohe, M. In *Weisfeiler and Leman Go Neural: Higher-Order Graph Neural Networks*, Proceedings of the AAAI conference on artificial intelligence, 2019; pp 4602−4609.
(66) Jäger, M. O. J.; Morooka, E. V.; Federici Canova, F.; Himanen, L.; Foster, A. S. Machine learning hydrogen adsorption on nanoclusters through structural descriptors. *npj Comput. Mater.* **2018**, *4*, No. 37.
(67) Gonzalez, T. F. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.* **1985**, *38*, 293−306.
(68) Gönen, M.; Alpaydın, E. Multiple kernel learning algorithms. *J. Mach. Learn. Res.* **2011**, *12*, 2211−2268.
(69) He, Z.-F.; Yang, M.; Gao, Y.; Liu, H.-D.; Yin, Y. Joint multi-label classification and label correlations with missing labels and feature selection. *Knowl.-Based Syst.* **2019**, *163*, 145−158.
(70) Wu, G.; Zheng, R.; Tian, Y.; Liu, D. Joint ranking SVM and binary relevance with robust low-rank learning for multi-label classification. *Neural Networks* **2020**, *122*, 24−39.
(71) Sun, L.; Wang, T.; Ding, W.; Xu, J.; Lin, Y. Feature selection using Fisher score and multilabel neighborhood rough sets for multilabel classification. *Inf. Sci.* **2021**, *578*, 887−912.
(72) Andersen, M.; Levchenko, S. V.; Scheffler, M.; Reuter, K. Beyond Scaling Relations for the Description of Catalytic Materials. *ACS Catal.* **2019**, *9*, 2752−2759.
(73) Andersen, M.; Reuter, K. Adsorption Enthalpies for Catalysis Modeling through Machine-Learned Descriptors. *Acc. Chem. Res.* **2021**, *54*, 2741−2749.
(74) Villadsen, T.; Ligterink, N. F. W.; Andersen, M. Predicting binding energies of astrochemically relevant molecules via machine learning. *Astron. Astrophys.* **2022**, *666*, A45.
(75) Vandermause, J.; Xie, Y.; Lim, J. S.; Owen, C. J.; Kozinsky, B. Active learning of reactive Bayesian force fields applied to heterogeneous catalysis dynamics of H/Pt. *Nat. Commun.* **2022**, *13*, No. 5183.
(76) Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357−362.
(77) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17*, 261−272.
(78) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.
(79) Hagberg, A. A.; Schult, D. A.; Swart, P. J. In *Exploring Network Structure, Dynamics, and Function using NetworkX*, Proceedings of the 7th Python in Science Conference, Pasadena, CA USA, 2008; pp 11−5.
(80) Hjorth Larsen, A.; Mortensen, J. J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dułak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C.; et al. The atomic simulation environment—a Python library for working with atoms. *J. Phys.: Condens. Matter* **2017**, *29*, No. 273002.
(81) Dalcín, L.; Paz, R.; Storti, M. MPI for Python. *J. Parallel Distr. Comput.* **2005**, *65*, 1108−1115.

(82) Dalcín, L.; Paz, R.; Storti, M.; D'Elía, J. MPI for Python: Performance improvements and MPI-2 extensions. *J. Parallel Distr. Comput.* **2008**, *68*, 655−662.
(83) Dalcín, L. D.; Paz, R. R.; Kler, P. A.; Cosimo, A. Parallel distributed computing using Python. *Adv. Water Resour.* **2011**, *34*, 1124−1139. New Computational Methods and Software Tools.
(84) Dalcín, L.; Fang, Y.-L. L. mpi4py: Status Update After 12 Years of Development. *Comput. Sci. Eng.* **2021**, *23*, 47−54.

https://doi.org/10.1021/acs.jpcc.3c02539
J. Phys. Chem. C 2023, 127, 14211−14221