



A novel method for atomization energy prediction based on natural-parameter network

Chaoqin Chu^a, Qinkun Xiao^{a,b,*}, Chaozheng He^{c,*}, Chen Chen^b, Lu Li^a, Junyan Zhao^b, Jinzhou Zheng^c, Yinhuan Zhang^a

^a School of Mechanical and Electrical Engineering, Xi'an Technological University, Xi'an 710021, China

^b School of Electrical and Information Engineering, Xi'an Technological University, Xi'an 710021, China

^c School of Materials Science and Chemical Engineering, Xi'an Technological University, Xi'an 710021, China

ARTICLE INFO

Article history:

Received 31 August 2023

Revised 7 October 2023

Accepted 9 October 2023

Available online 17 October 2023

Keywords:

Structure prediction

Atomization energy

Deep learning

Coulomb matrix

NPN

End-to-end

ABSTRACT

Atomization energy (AE) is an important indicator for measuring material stability and reactivity, which refers to the energy change when a polyatomic molecule decomposes into its constituent atoms. Predicting AE based on the structural information of molecules has been a focus of researchers, but existing methods have limitations such as being time-consuming or requiring complex preprocessing and large amounts of training data. Deep learning (DL), a new branch of machine learning (ML), has shown promise in learning internal rules and hierarchical representations of sample data, making it a potential solution for AE prediction. To address this problem, we propose a natural-parameter network (NPN) approach for AE prediction. This method establishes a clearer statistical interpretation of the relationship between the network's output and the given data. We use the Coulomb matrix (CM) method to represent each compound as a structural information matrix. Furthermore, we also designed an end-to-end predictive model. Experimental results demonstrate that our method achieves excellent performance on the QM7 and BC2P datasets, and the mean absolute error (MAE) obtained on the QM7 test set ranges from 0.2 kcal/mol to 3 kcal/mol. The optimal result of our method is approximately an order of magnitude higher than the accuracy of 3 kcal/mol in published works. Additionally, our approach significantly accelerates the prediction time. Overall, this study presents a promising approach to accelerate the process of predicting structures using DL, and provides a valuable contribution to the field of chemical energy prediction.

© 2023 Published by Elsevier B.V. on behalf of Chinese Chemical Society and Institute of Materia Medica, Chinese Academy of Medical Sciences.

Accurate prediction of structures in chemical compound space (CCS) is crucial for the design and synthesis of novel materials [1]. Although theoretical structure prediction methods have been widely adopted in material research [2,3], the computational cost associated with large-scale structural systems remains a challenge. A CALYPSO software based on swarm intelligence has been developed to accelerate structure prediction [4,5]. The existing first-principles-based (DFT) methods are time-consuming and economically expensive when applied to predict large-scale structures. Consequently, there is an urgent need to accelerate the process of material structure prediction [6].

Typically, structure prediction methods involve several steps [6]: (1) Randomly generating an initial population of structures un-

der symmetry constraints; (2) Assessing the similarity among generated structures and eliminating duplicates; (3) Performing local optimization and energy calculations using the DFT on generated structures; (4) Constructing the next generation of structures using swarm intelligence algorithms, where low-energy structures evolve into new structures while high-energy structures are replaced by randomly generated ones. However, for large-scale structures containing hundreds or thousands of atoms, the computational time required for energy calculations is significant. Furthermore, performing energy calculations and optimizations for multiple randomly generated structures further compounds the time cost. This limitation significantly impedes the prediction of large-scale material structures in most structure prediction software. Deep learning (DL) methods have gained substantial success in computer vision, pattern recognition, and other fields [7,8]. However, their application in predicting and generating new structures in computational chemistry remains limited [9].

In conclusion, the acceleration of material structure prediction is a critical problem that needs to be addressed. While traditional

* Corresponding author.

** Corresponding author at: School of Mechanical and Electrical Engineering, Xi'an Technological University, Xi'an 710021, China.

E-mail addresses: xiaoqinkun10000@163.com (Q. Xiao), hec2019@xatu.edu.cn (C. He).

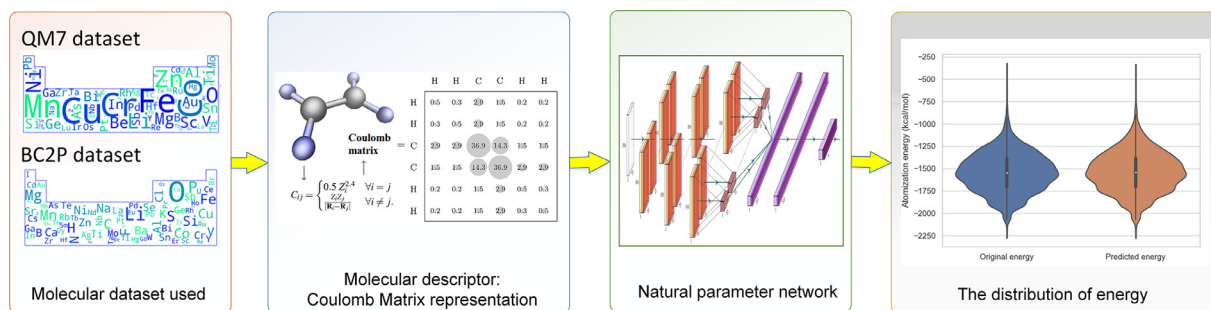


Fig. 1. Schematic diagram of an end-to-end chemical energy prediction process.

methods face challenges in dealing with large-scale structures, DL methods offer a potential solution. By leveraging DL techniques, it may be possible to overcome the time constraints associated with energy calculations and optimize the prediction process. This application of DL in computational chemistry holds promise for advancing the field of structure prediction and generating new structures.

In recent years, the rapid development of artificial intelligence has made ML a new method that researchers are eager to explore. ML has become popular in various application scenarios that require prediction, classification, and decision-making [4,10]. With the availability of large-scale quantum mechanical data [11–13], researchers have established ML models and used them to predict material properties such as formation energies, defect energies, elasticity, and other mechanical properties [4]. For instance, Hansen *et al.* [14] employed a linear regression (LR) algorithm to learn the relationship between structural information and cluster energy and predicted the energy of new clusters based on atomic Cartesian coordinates and nuclear charge. Meanwhile, neural network (NN) methods [15,16] have been leveraged to accelerate the construction of potential energy surfaces. Rupp *et al.* [17] introduced a machine learning model that predicts the AEs of different organic molecules. Gupta *et al.* [18,19] utilized the DL method to establish the correlation between molecular topological features and accurate AE predictions, achieving an impressive MAE.

In addition, some DL methods were used to replace the DFT calculation process to reduce the computational complexity of the system and accelerate structure prediction in an end-to-end way [20]. Due to the strong feature extraction and feature learning abilities of the DL methods, we have introduced a new NPN [21] model that predicts the AE of different molecules in an end-to-end manner based solely on nuclear charge and atomic position. To address this task, an end-to-end energy prediction model was constructed, and its schematic diagram is shown in Fig. 1.

Let Φ be an AE prediction model, and the model Φ_{NPN} utilizes a neural network architecture for probabilistic modeling. The model leverages the properties of exponential family distributions to transform the neural network output into the natural parameters of the exponential family distribution. This allows NPN to conveniently parameterize the probabilistic model and maximize the log-likelihood function using gradient-based optimization algorithms during training. NPN transforms the conditional probability $p(Y/X)$ between the input variable and the output variable into the form of an exponential family distribution. According to all the molecular structure information, we extract the coordinate information of all atoms of each molecule as the input of Φ_{NPN} , which is represented by X . The AE of each molecule is represented by Y . Here we have:

$$Y = \Phi(X; \theta_{\text{NPN}}) \quad (1)$$

where X is the coordinate information of all atoms of each molecule, and θ_{NPN} is the parameters of the NPN model. NPN con-

sists of both linear and nonlinear transformation layers. Here, we first introduce the linear form of NPN. For simplicity, let's assume a Gaussian distribution $\xi = (q, d)^T$ with two natural parameters. We decompose the distribution over a weight matrix $p(W^l | W_q^l, W_d^l) = \prod_{i,j} p(W_{ij}^l | W_{q,ij}^l, W_{d,ij}^l)$, where $(W_{q,ij}^l, W_{d,ij}^l)$ is the corresponding natural parameter. We assume similar decomposed distributions for b^l , z^l , and x^l in NPN, which are all exponential family distributions. For computational convenience, we wish to approximate z^l using another exponential family distribution by matching its mean and variance. To compute $(W_m^l, W_s^l) = f(W_q^l, W_d^l)$ and $(b_m^l, b_s^l) = f(b_q^l, b_d^l)$, we can obtain z_m^l and z_s^l from the mean z_m^l and variance z_s^l of z^l as follows:

$$(x_m^{l-1}, x_s^{l-1}) = f(x_q^{l-1}, x_d^{l-1}), z_m^l = x_m^{l-1} W_m^l + b_m^l \quad (2)$$

$$z_s^l = x_s^{l-1} W_s^l + x_s^{l-1} (W_m^l \odot W_m^l) + (x_m^{l-1} \odot x_m^{l-1}) W_s^l + b_s^l \quad (3)$$

$$(z_q^l, z_d^l) = f^{-1}(z_m^l, z_s^l) \quad (4)$$

where, the symbol \odot represents the product of the elements in a distribution, while the bijective function $f(\cdot, \cdot)$ maps its natural parameters to the distribution's mean and variance. Similarly, symbol $f^{-1}(\cdot, \cdot)$ denotes the inverse transformation. The means and variances of W^l and b^l , which are obtained from the natural parameters, are represented by W_m^l, W_s^l, b_m^l and b_s^l , respectively. The values of z_m^l and z_s^l that are computed can be used to recover z_q^l and z_d^l , thereby streamlining the feed-forward calculation of the nonlinear transformation.

Once we have the distribution over z^l , defined by natural parameters z_q^l and z_d^l , that has been linearly transformed, we then apply an element-wise, nonlinear transformation $v(\cdot)$ to it. The resulting distribution of activations is denoted as $p_x(x^l) = p_z(v^{-1}(x^l))|v^{-1'}(x^l)|$, where p_z is the factorized distribution over z^l defined by (z_q^l, z_d^l) . Even if $p_x(x^l)$ is not an exponential-family distribution, we can still approximate it using $p(x^l | x_q^l, x_d^l)$ by matching the first two moments. Once we obtain the mean x_m^l and variance x_s^l of $p_x(x^l)$, we can compute the corresponding natural parameters using $f^{-1}(\cdot, \cdot)$. The feed-forward computation can be expressed as:

$$x_m = \int p_z(z | z_q, z_d) v(z) dz \quad (5)$$

$$x_s = \int p_z(z | z_q, z_d) v(z)^2 dz - x_m^2, (x_q, x_d) = f^{-1}(x_m, x_s) \quad (6)$$

To model distributions over weights and neurons, the natural parameters must be learned. The NPN is designed to take a vector random distribution as input, such as a multivariate Gaussian distribution. It then multiplies this input by a matrix random distribution and applies a nonlinear transformation before outputting another distribution. Since all three distributions in this process can

be characterized by their natural parameters, learning and prediction of the network can be performed in the space of natural parameters. During back-propagation, for distributions characterized by two natural parameters, the gradient is composed of two terms. For instance, in the equation $\partial E/\partial z_q = (\partial E/\partial x_m) \odot (\partial x_m/\partial z_c) + (\partial E/\partial x_s) \odot (\partial x_s/\partial z_q)$, where E represents the error term of the network. Naturally, nonlinear NPN layers can be stacked to form deep NPN, as shown in Algorithm 1. NPN does not need expensive reasoning algorithms, such as variational reasoning or Markov chain Monte Carlo (MCMC). Moreover, in terms of flexibility, it can choose different types of exponential family distributions for weights and neurons.

We know that χ can be converted using a previous study [22] with Coulomb matrices (CM). The CM are matrices that contain information about the atomic charges and atomic coordinates. The CM can be calculated using the following formula:

$$C_{ij} = \begin{cases} Z_i^{2.4}/2 & \text{if } i = j \\ Z_i Z_j / |\mathbf{R}_i - \mathbf{R}_j| & \text{if } i \neq j \end{cases} \quad (7)$$

among them, $1 \leq i, j \leq 23$, Z_i is the nuclear charge of atom i , and \mathbf{R}_i is its Cartesian coordinate.

Specifically, QM7 is a dataset consisting of 7165 molecules taken from the larger GDB-13 database, which comprises nearly one billion synthetic organic molecules. QM7 has information about the molecule that consists of the atoms H, C, N, O, and S. The maximum number of atoms in a molecule is 23. The molecules in QM7 exhibit a diverse range of chemical structures, such as double and triple bonds, cycles, carboxylic acids, cyanides, amides, alcohols, and epoxides [23]. The CM representation is used to describe the molecular structures, and this approach is invariant to translations and rotations of the molecule. In addition to the CM, the dataset includes AEs ranging from -800 kcal/mol to -2000 kcal/mol [14].

The main focus of this article is on the χ and \mathcal{L} components in the QM7 dataset. Due to the varying number of atoms in each molecule, the sizes of their true CMs are also different. To make all CMs with the same size of 23×23 , we extended the size of the original CM and filled the expanded space in the CM with 0.

For NPN, GCN, CNN, and LSTM models, we use the rectified linear unit (ReLU) activation function in the hidden layer. We used the ADAM optimizer [24] with a learning rate of 0.0001 and a decay rate of 0.00001, and based on our experience and knowledge, we have selected different combinations of network hyper-parameters for model training and fine-tuning. We set the learning rate to [0.01 0.001 0.0001], the number of training rounds [100 200 300], the batch size [32 64 128], and the hidden layer nodes [100 200 300]. Then, we combined the values of different network hyper-parameters. The experimental results showed that when the learning rate was 0.0001, the batch size was 64, the hidden layer nodes were 200, and the number of training rounds was

200, the model performance was optimal. The proposed models have been implemented in the Pytorch library. As for the hardware system configuration, the processor is an Intel (R) Xeon (R) Silver 4110 CPU@2.10 GHz, the RAM has a capacity of 32.0 GB, and the graphics card is an NVIDIA RTX 2080 Ti [25]. We randomly divided the dataset into a training dataset and a test dataset in the ratio of 8:2.

We used the proposed method to train the AE prediction models on training data and validated the performance of the models using testing data, and the losses of the proposed models during the training are shown in Fig. 2.

By training on a large set of different molecules in QM7, we use MAE and root mean square error (RMSE) as evaluation indicators to evaluate the prediction performance of various algorithms under different molecule presentation methods. The comparison results between our method and other methods [22] are shown in Table 1. Our results indicate that the proposed method exhibits a lower prediction error, indicating excellent prediction performance. Based on this, we trained and tested different DL models we designed on the QM7 dataset [26], and obtained some information that reflects the performance of the models, such as model parameters, testing time, and testing errors. Experiments on real-world datasets [26] show that NPN can achieve state-of-the-art performance on regression tasks. The performance of different methods are compared in Table 2.

In previous experiments [14], researchers often merged feature values/vectors and feature centers with flattened CM to form a 7165-sample dataset with N -dimensional features. However, the integration of new molecular property information has brought about a new problem: the concatenation of old and new features may lead to unwanted "heterogeneity" in the feature vectors, and including more features may actually increase noise in the dataset

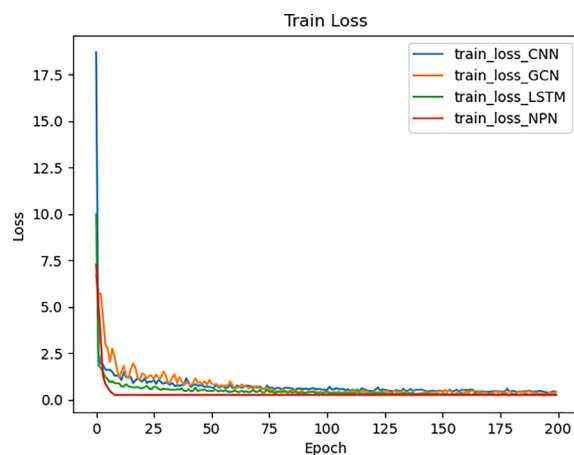


Fig. 2. The results of training losses of different DL models.

Algorithm 1

Deep NPN's algorithm.

- 1: Input: $D = \{(\chi_i, \mathcal{L}_i)\}_{i=1}^N$, number of iterations T , learning rate ρ_t , number of layers L .
- 2: for $t = 1: T$ do
- 3: for $l = 1: L$ do
- 4: Apply Eqs. 2–6 to compute the linear and nonlinear transformation in different layers.
- 5: end for
- 6: Compute the error E from (z_q^l, z_d^l) or (x_q^l, x_d^l) .
- 7: for $l = L: 1$ do
- 8: Compute $\partial E/\partial W_m^l, \partial E/\partial W_s^l, \partial E/\partial b_m^l$, and $\partial E/\partial b_s^l$.
- 9: end for
- 10: Update W_q^l, W_d^l, b_q^l , and b_d^l in all layers.
- 11: end for

Table 1

Comparison of prediction errors across multiple algorithms under different representation types using MAE and RMSE metrics.

Method	Representation	MAE	RMSE
K-nearest neighbors	Eigen-spectrum	70.72 ± 2.12	92.49 ± 2.70
	Sorted CM	71.54 ± 0.97	95.97 ± 1.45
Linear regression	Eigen-spectrum	29.17 ± 0.352	38.01 ± 1.11
	Sorted CM	20.72 ± 0.32	27.22 ± 1.45
Mixed effects	Eigen-spectrum	10.50 ± 0.48	20.38 ± 9.29
	Sorted CM	8.5 ± 0.45	12.16 ± 0.95
Neural network	Eigen-spectrum	14.08 ± 0.29	20.29 ± 0.73
	Sorted CM	11.82 ± 0.45	16.01 ± 0.81
Ours (NPN)	Sorted CM	5.47 ± 0.36	3.69 ± 0.51
	Extended CM	1.31 ± 0.011	1.28 ± 0.024

Table 2
Comparison results of performance evaluation of different networks.

Method	Network performance			Data
	Parameter	Testing time (ms)	Test loss	
CNN	93,449	0.174	0.277 (kcal/mol)	QM7
LSTM	78,465	0.196	0.135 (kcal/mol)	
GCN	16,861	0.547	2.564 (kcal/mol)	
NPN	212,402	0.118	0.268 (kcal/mol)	
ML [27]	×	×	1.7 (kcal/mol)	QM7
DL [28]	×	×	1.45 (kcal/mol)	GDB-10
BRNN [29]	×	×	3 (kcal/mol)	QM7
NPN	212,402	0.391	0.236 (eV/atom)	BC2P

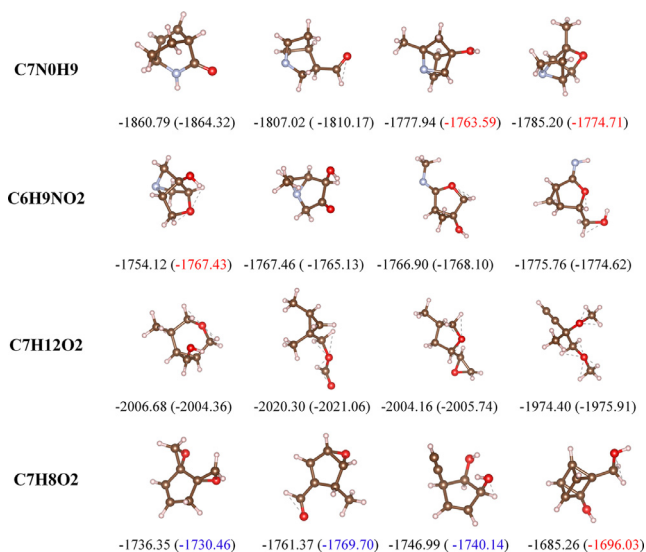


Fig. 3. Visualization of different conformations of four molecules and comparison of their AE prediction results. Corresponding predicted AEs by our model are shown in parentheses, all AEs are in kcal/mol. Our model has a prediction error of about 0.2–3 kcal/mol for AE. The data marked in blue indicates an error between 5 and 10 kcal/mol, while the data marked in red indicates an error over 10 kcal/mol.

[30]. It should be noted that some traditional ML techniques may not be able to identify meaningful patterns from these newly added features, so merging new features may result in poorer results [31].

We present the results analysis of AEs prediction for four different molecules and their isomers, it can be shown in Fig. 3. Based on the structural information, our proposed method achieved good AE prediction results by using the extended CM method to characterize molecules. However, we also found that the model has a significant prediction error for molecules [32] with fewer atoms, which may be due to the interference of additional information supplemented in the extended CM to the model's learning of structural information features. For some molecules with compact structures relative to sparse spatial structures, the prediction errors were relatively larger. We analyzed that this may be due to the high density distribution of a large number of identical atoms in the molecule, which led to the model's unsatisfactory extraction of the CM features.

To validate the robustness and effectiveness of our proposed method, we randomly divided the BC2P data from Fu et al. [33] into a train dataset and a test dataset with an 8:2 ratio. We use our proposed method to train a model on train data, and the trained model accurately predicts the corresponding energy [33]. The training loss curve can be seen in Fig. 4. Moreover, we tested the model on test data, and the results showed that our model had an average prediction time of 0.391 ms for the energy of each molecular conformation and an average test loss of 0.236 eV/atom.

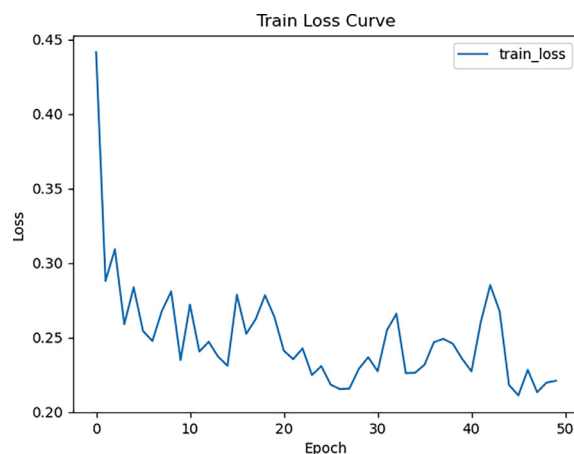


Fig. 4. The results of training loss curve of our model on BC2P data.

The results of performance evaluation of NPN model on BC2P is shown in Table 2.

The study aims to propose and validate a method for predicting atomic energy. Initially, we introduce the extended CM representation of molecular structure along with its inherent characteristics. Subsequently, we compare the performance of various ML and DL models for predicting AEs. These models not only enable fast AE prediction but also expedite the structure prediction process. Furthermore, to overcome the limitations of our current method, we plan to incorporate the three-dimensional spatial pattern of material structure information in future research. We intend to employ the graph convolutional network (GCN) method to mathematically represent atoms in chemical molecules. In addition to model development, we informally discuss the results of numerical experiments conducted in this study. Through extensive comparative experiments and analyses, we gain valuable insights into the performance and potential applications of the proposed method. Some potential applications include catalyst design, materials discovery, optimization of energy storage and conversion, as well as material performance prediction. In the future, we envision conducting further in-depth research on material performance prediction and material discovery. By leveraging the power of machine learning and data mining techniques on large datasets, we can uncover hidden correlations within complex data structures. This will allow us to predict new material structures and their corresponding properties. Ultimately, this research contributes to the advancement of materials science and opens up new possibilities for designing innovative materials with tailored properties.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the Nature Science Foundation of China (Nos. 61671362 and 62071366).

References

- [1] G. Montavon, M. Rupp, V. Gobre, et al., *New J. Phys.* 15 (2013) 095003.
- [2] E. Tuca, G. DiLabio, A. Otero-de-la-Roza, *J. Chem. Inf. Model.* 62 (2022) 4107–4121.
- [3] D. Yang, L. Wang, P. Yuan, et al., *Chin. Chem. Lett.* 34 (2023) 107964.
- [4] Q. Tong, L. Xue, J. Lv, et al., *Faraday Discuss* 211 (2018) 31–43.
- [5] P. Gao, S. Wang, J. Lv, et al., *RSC Adv.* 7 (2017) 39869–39876.
- [6] X. Wei, C. Zhou, X. Shen, et al., *J. Jilin Univ.* 51 (2021) 667–676.

- [7] L. Torres, J.P. Arrais, B. Ribeiro, *Neural Comput. Appl.* 35 (2023) 13167–13185.
- [8] M. Sumanto, M.A. Martoprawiro, A.L. Ivansyah, *J. Phys. Conf. Ser.* 2072 (2021) 012005.
- [9] G.A. Pinheiro, J. Mucelini, M.D. Soares, et al., *J. Phys. Chem. A* 124 (2020) 9854–9866.
- [10] M. Tsubaki, T. Mizoguchi, *Phys. Rev. Lett.* 125 (2020) 206401.
- [11] P. Reddy, A.B. Bhattacharjee, *Mach. Learn. Sci. Technol.* (2) (2021) 025019.
- [12] N. Dandu, L. Ward, R.S. Assary, et al., *J. Phys. Chem. A* 124 (2020) 5804–5811.
- [13] L. Ward, B. Blaiszik, I. Foster, et al., *MRS Commun.* 9 (2019) 891–899.
- [14] Hansen K, G. Montavon, F. Biegler, et al., *J. Chem. Theory Comput.* 9 (2013) 3404–3419.
- [15] A. Jussupow, V.R.I. Kaila, *J. Chem. Theory Comput.* 19 (2023) 1965–1975.
- [16] B. Selvaratnam, R.T. Koodali, P. Miró, *J. Phys. Chem.* 60 (2020) 1928–1935.
- [17] M. Rupp, A. Tkatchenko, K.R. Müller, et al., *J. Cheminf.* 4 (2012) 1.
- [18] A.K. Gupta, K. Raghavachari, *J. Chem. Theory Comput.* 18 (2022) 2132–2143.
- [19] E.M. Collins, K. Raghavachari, *J. Comput. Chem.* 19 (2023) 2804–2810.
- [20] M. Kilgour, J. Rogal, M. Tuckerman, *J. Chem. Theory Comput.* (2023). arXiv: 2303.10140
- [21] H. Wang, X. Shi, D.Y. Yeung, *Adv. Neural Inf. Process. Syst.* (2016) 118–126.
- [22] G. Montavon, K. Hansen, S. Fazli, et al., *Adv. Neural Inf. Process. Syst.* 1 (2012) 449–457.
- [23] J. Hoja, L. Medrano Sandonas, B.G. Ernst, et al., *Sci. Data* 8 (2021) 43.
- [24] D.P. Kingma, J. Ba, *Learning* (2014) arXiv:1412.6980 [cs.LG].
- [25] C. Chu, Q. Xiao, Y. Zhang, et al., *Int. J. Pattern Recognit. Artif. Intell.* 36 (2022) 2250036.
- [26] M. Rupp, A. Tkatchenko, K.R. Müller, et al., *Phys. Rev. Lett.* 108 (2012) 058301.
- [27] M. Krykunov, T.K. Woo, *J. Chem. Theory Comput.* 14 (2018) 5229–5237.
- [28] S. Laghuvarapu, Y. Pathak, U.D. Priyakumar, *J. Comput. Chem.* 41 (2020) 790–799.
- [29] A.B. Tchagang, J.J. Valdés, Prediction of the Atomization Energy of Molecules Using Coulomb Matrix and Atomic Composition in a Bayesian Regularized Neural Networks, ICANN, 2019, pp. 793–803.
- [30] S. Wengert, G. Csányi, K. Reuter, et al., *J. Chem. Theory Comput.* 18 (7) (2022) 4586–4593.
- [31] Y.H. Tang, W.A. de Jong, *J. Chem. Phys.* 150 (2019) 044107.
- [32] M. Tsubaki, T. Mizoguchi, *J. Chem. Theory Comput.* 17 (2021) 7814–7821.
- [33] X. Fu, X. Cheng, C. He, et al., *Phys. Chem. Chem. Phys.* 25 (2023) 2430–2438.