Supplementary Information

High-Accuracy Machine-Learning Water Model for Exploring Water Nanocluster Structures

Hao Zhou[1], Ya-Juan Feng[*1], Chao Wang[2], Teng Huang[3], Yi-Rong Liu[1], Shuai Jiang[1], Chun-Yu Wang[1], Wei Huang[*,1,3,4]

[1]School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230026, China

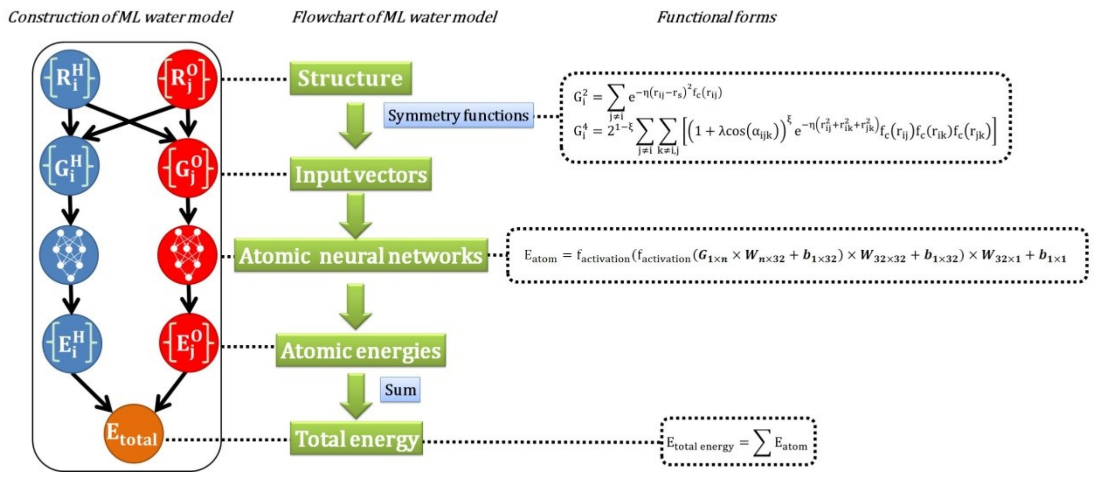
[2]National Synchrotron Radiation Laboratory, University of Science and Technology of China, Hefei, Anhui 230026, China

[3]Laboratory of Atmospheric Physico-Chemistry, Anhui Institute of Optics & Fine Mechanics, Chinese Academy of Sciences, Hefei, Anhui 230031, China

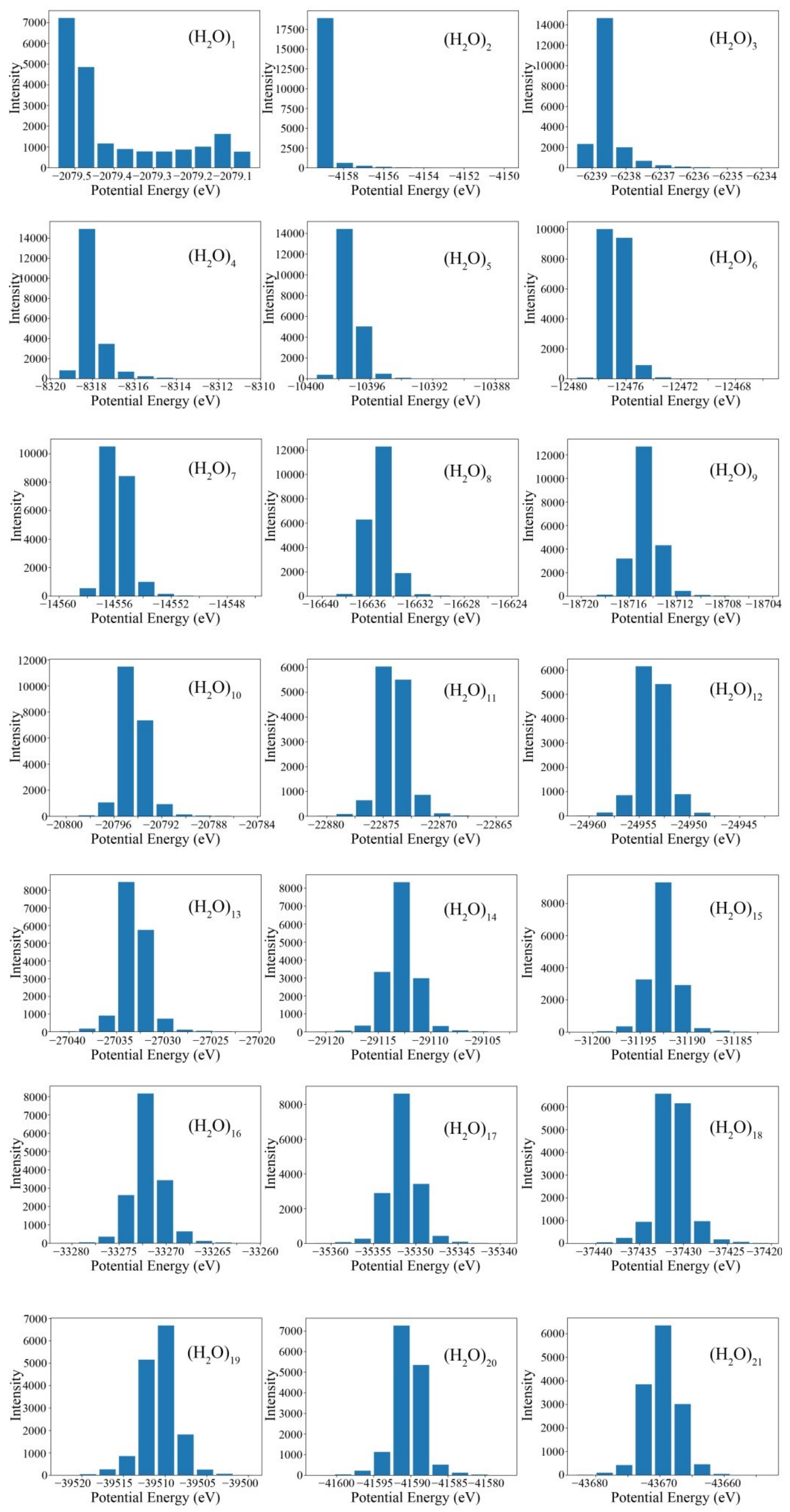
[4]CAS Center for Excellent in Urban Atmospheric Environment, Xiamen, Fujian 361021, China

Table of Contents **Page**

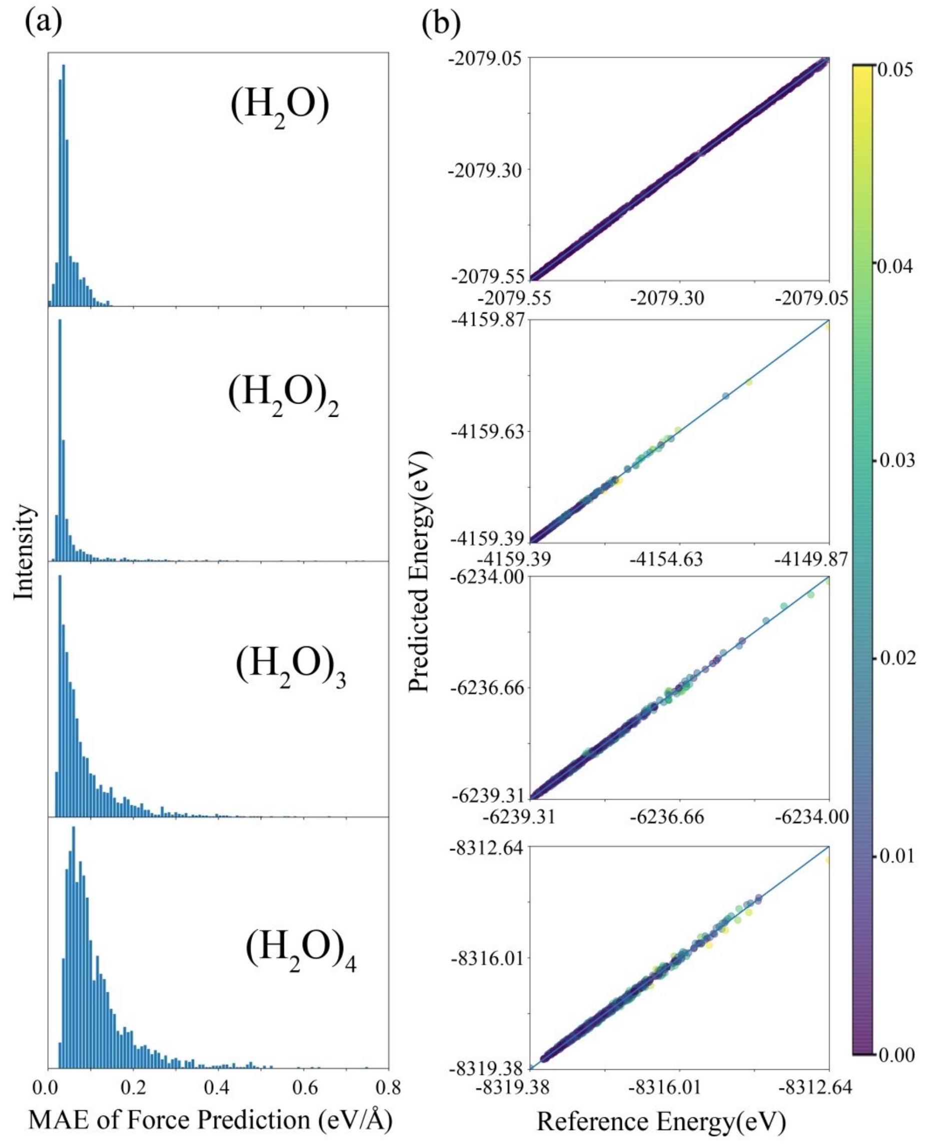
1. Supplementary Figures


Construction of ML water model
Flowchart of ML water model
Functional forms
$\{R_i^H\}$
$\{R_j^O\}$
$\{G_i^H\}$
$\{G_j^O\}$
$\{E_i^H\}$
$\{E_j^O\}$
E_{total}
Structure
Symmetry functions
Input vectors
Atomic neural networks
Atomic energies
Sum
Total energy
$G_i^2 = \sum_{j\neq i} e^{-\eta(r_{ij}-r_s)^2} f_c(r_{ij})$
$G_i^4 = 2^{1-\xi} \sum_{j\neq i}\sum_{k\neq i,j} \left[\left(1+\lambda\cos(\alpha_{ijk})\right)^{\xi} e^{-\eta(r_{ij}^2+r_{ik}^2+r_{jk}^2)} f_c(r_{ij}) f_c(r_{ik}) f_c(r_{jk})\right]$
$E_{atom} = f_{activation}(f_{activation}(\boldsymbol{G}_{1\times n} \times \boldsymbol{W}_{n\times 32} + \boldsymbol{b}_{1\times 32}) \times \boldsymbol{W}_{32\times 32} + \boldsymbol{b}_{1\times 32}) \times \boldsymbol{W}_{32\times 1} + \boldsymbol{b}_{1\times 1}$
$E_{total\ energy} = \sum E_{atom}$

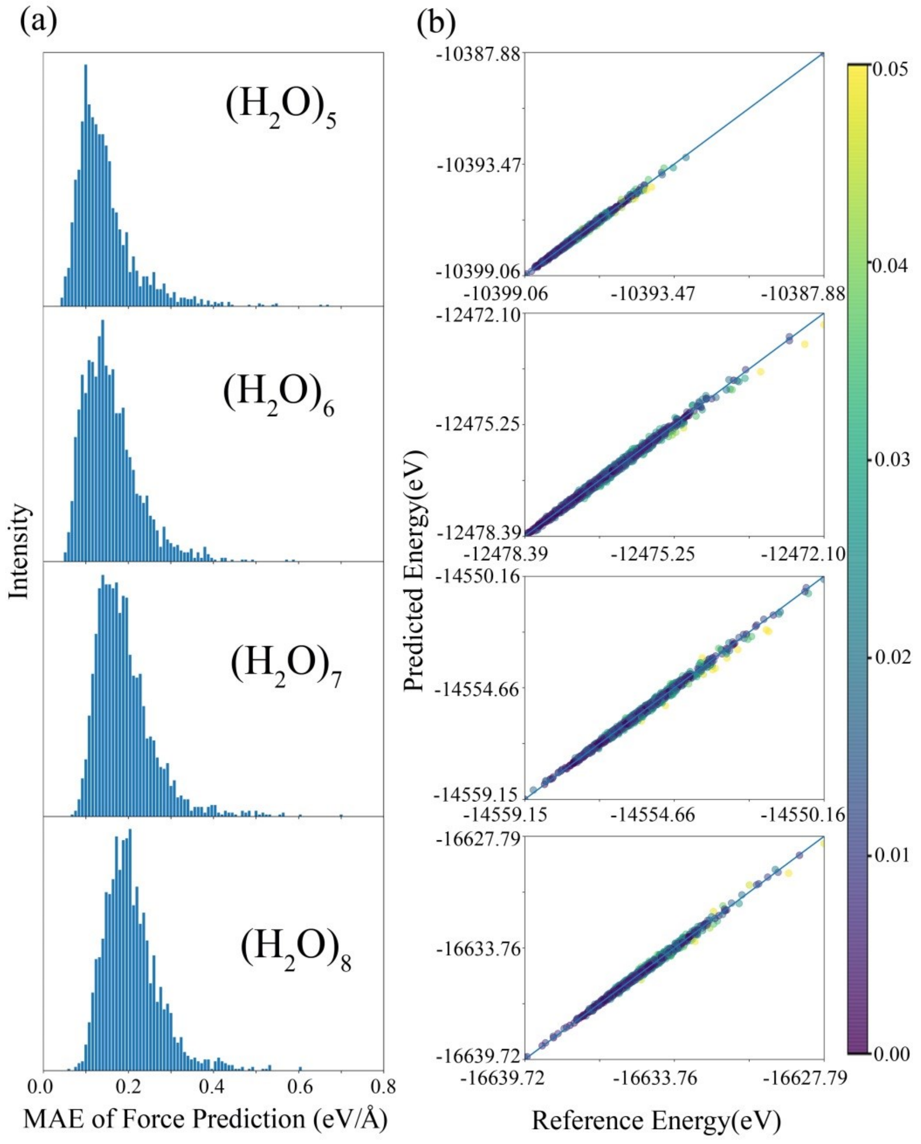
Supplementary Figure 1. Schematic representation of the construction and flowchart of ML water model and functional forms of symmetry functions, neural networks and total energy.



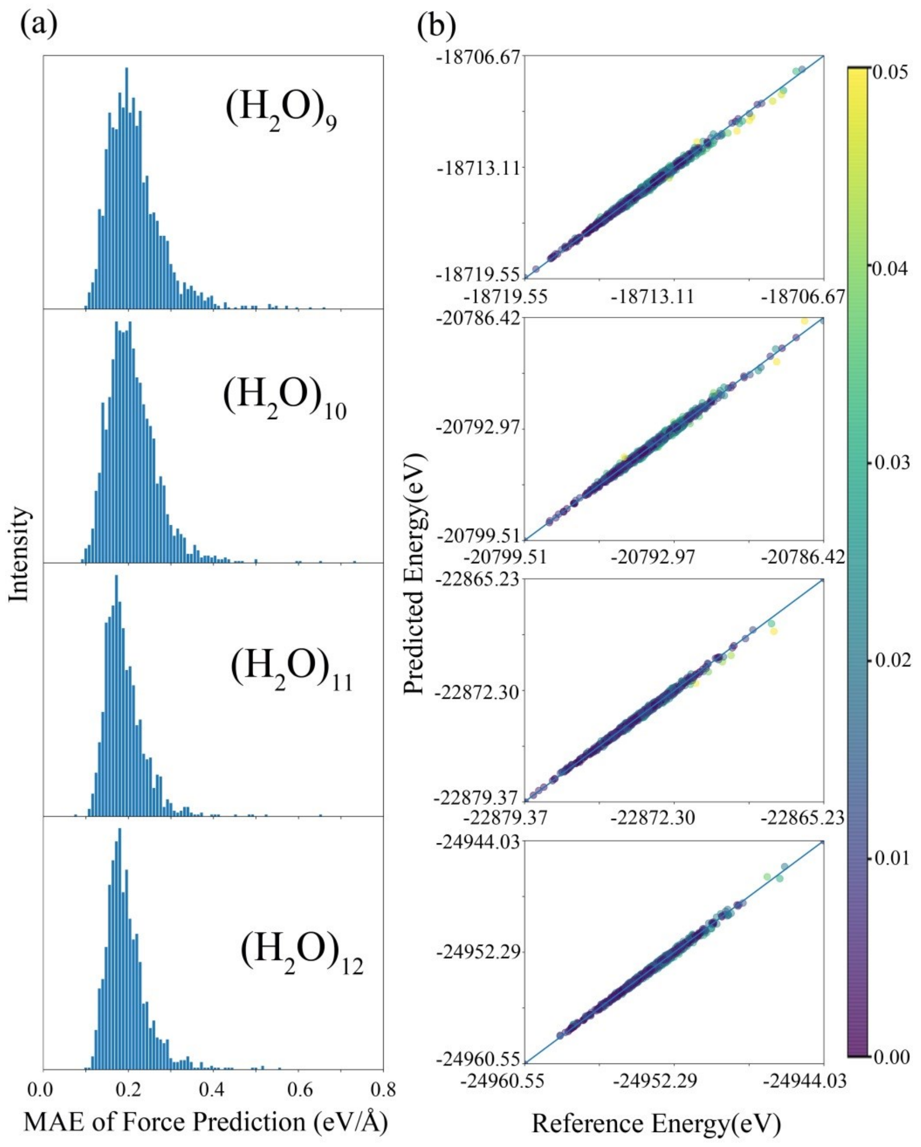
Supplementary Figure 2. Distributions of the potential energies of $(H_2O)_n$ (n = 1~21) clusters in the data set.



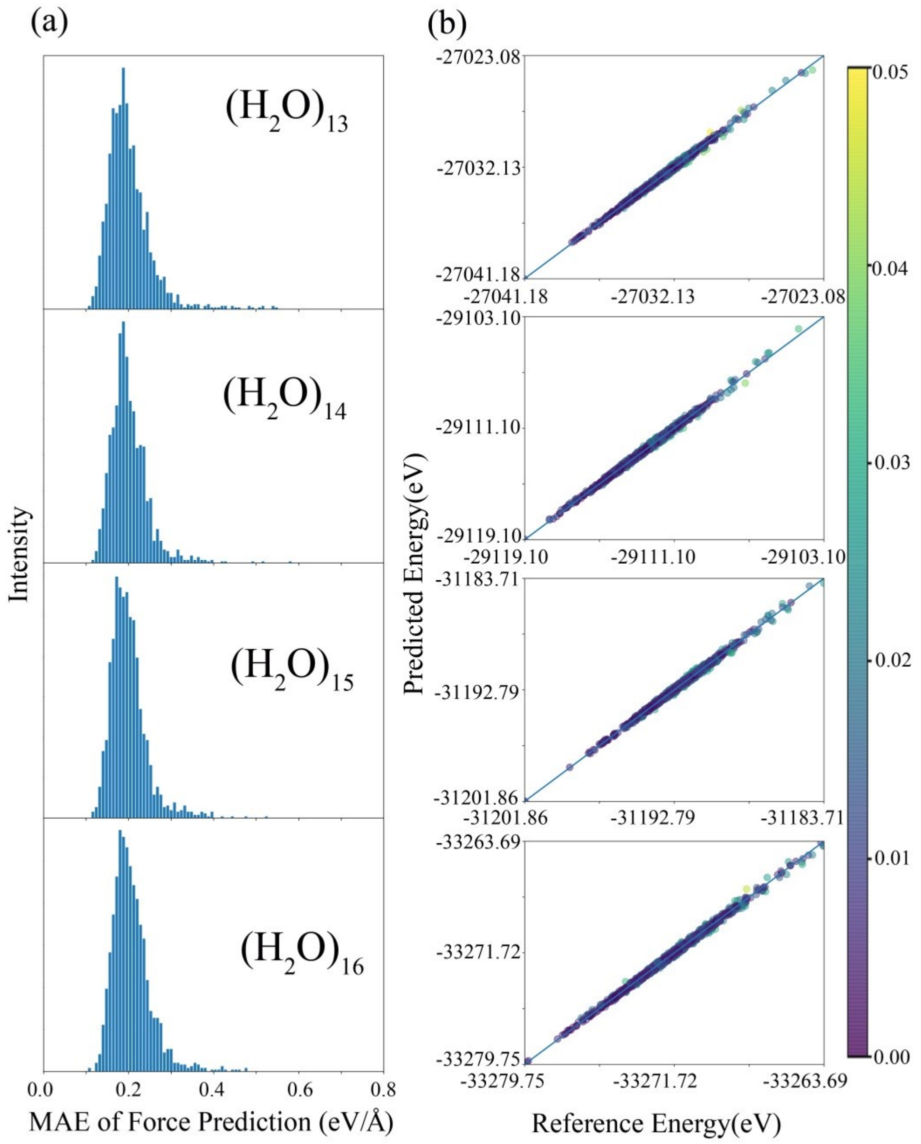
Supplementary Figure 3. (a) The MAE of force errors of $(H_2O)_n$ (n = 1 - 4), and (b) the comparisons of energies of $(H_2O)_n$ (n = 1 - 4) clusters calculated by the ML water model and *ab initio* calculations.



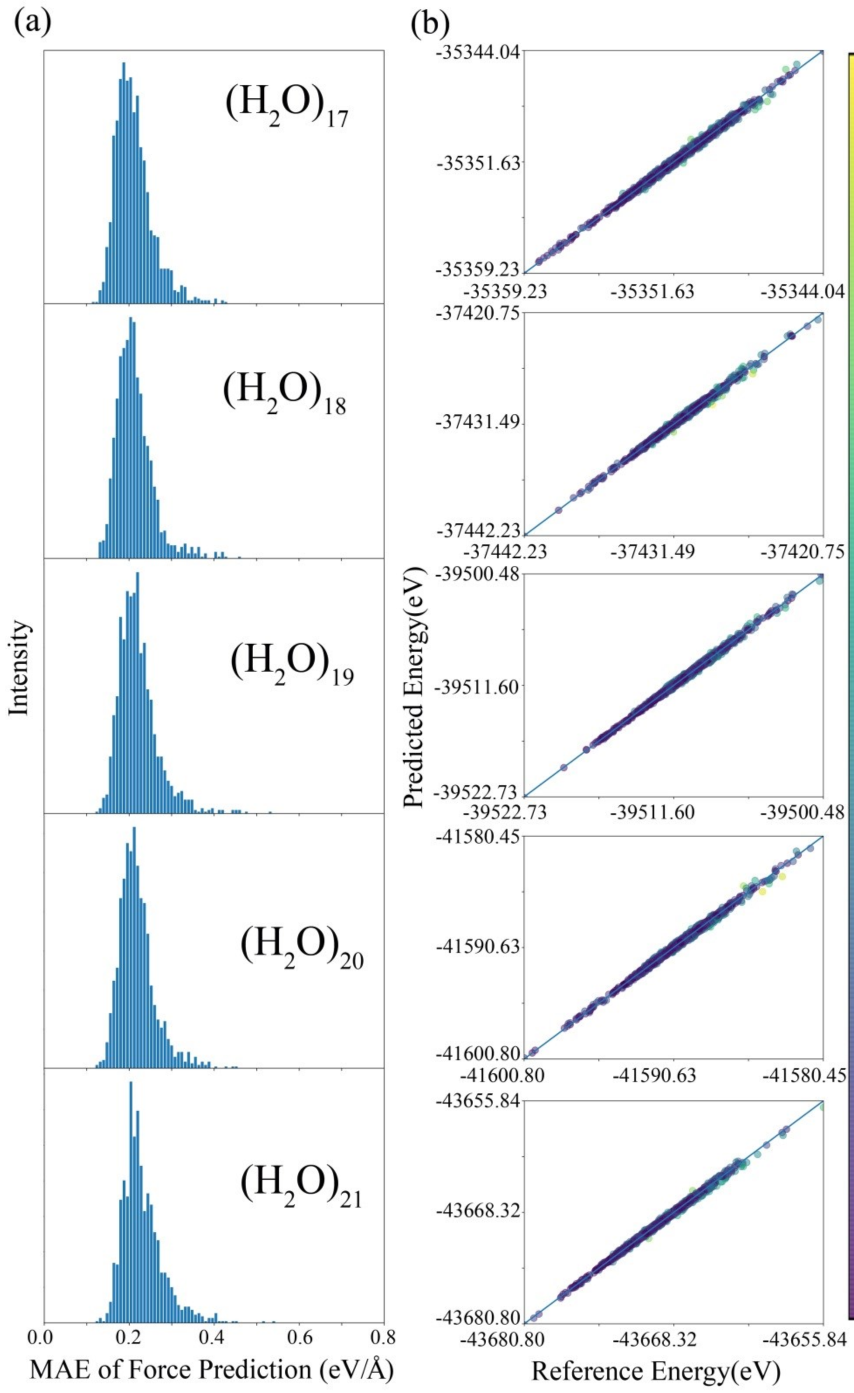
Supplementary Figure 4. (a) The MAE of force errors of $(H_2O)_n$ (n = 5 - 8), and (b) the comparisons of energies of $(H_2O)_n$ (n = 5 - 8) clusters calculated by the ML water model and *ab initio* calculations.



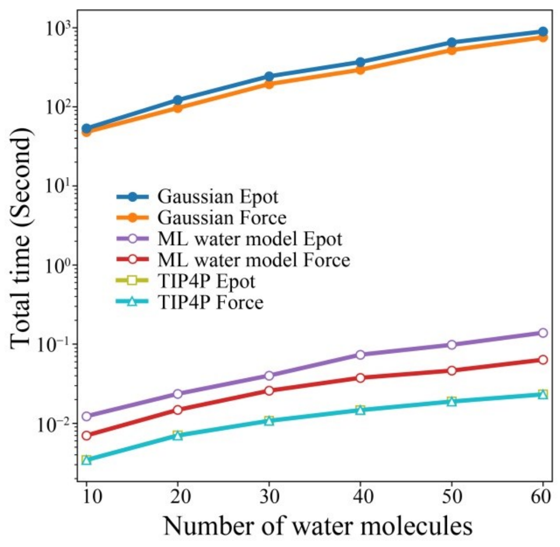
Supplementary Figure 5. (a) The MAE of force errors of $(H_2O)_n$ (n = 9 - 12), and (b) the comparisons of energies of $(H_2O)_n$ (n = 9 - 12) clusters calculated by the ML water model and *ab initio* calculations.



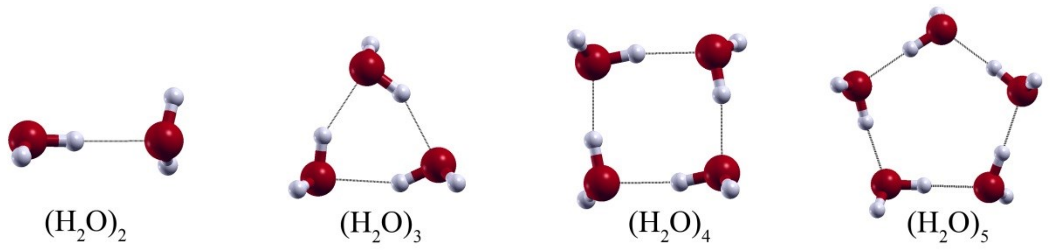
Supplementary Figure 6. (a) The MAE of force errors of $(H_2O)_n$ (n = 13 - 16), and (b) the comparisons of energies of $(H_2O)_n$ (n = 13 - 16) clusters calculated by the ML water model and *ab initio* calculations.



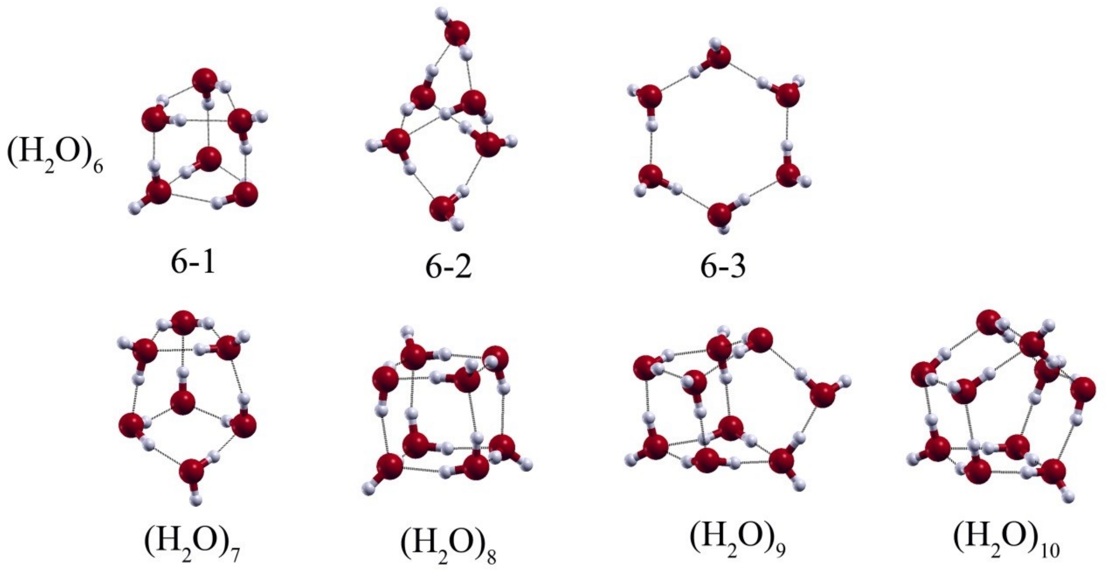
Supplementary Figure 7. (a) The MAE of force errors of $(H_2O)_n$ (n = 17 - 21), and (b) the comparisons of energies of $(H_2O)_n$ (n = 17 - 21) clusters calculated by the ML water model and *ab initio* calculations.



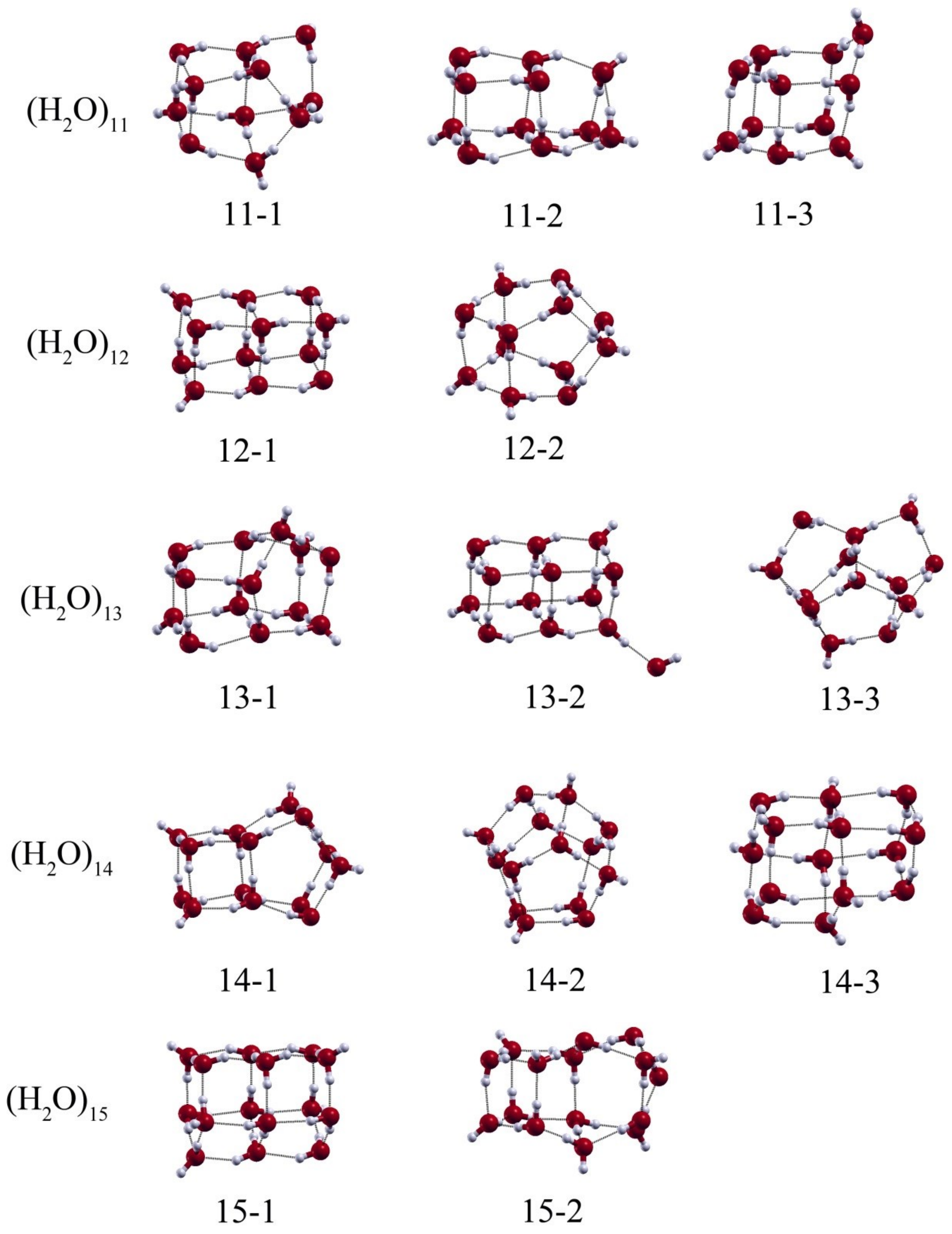
Supplementary Figure 8. Total time to calculate the energy and force of water clusters via the ML water model, the Gaussian 09 program package and the empirical water models such as TIP4P. The TIP4P simulations use the ASE codes[1].



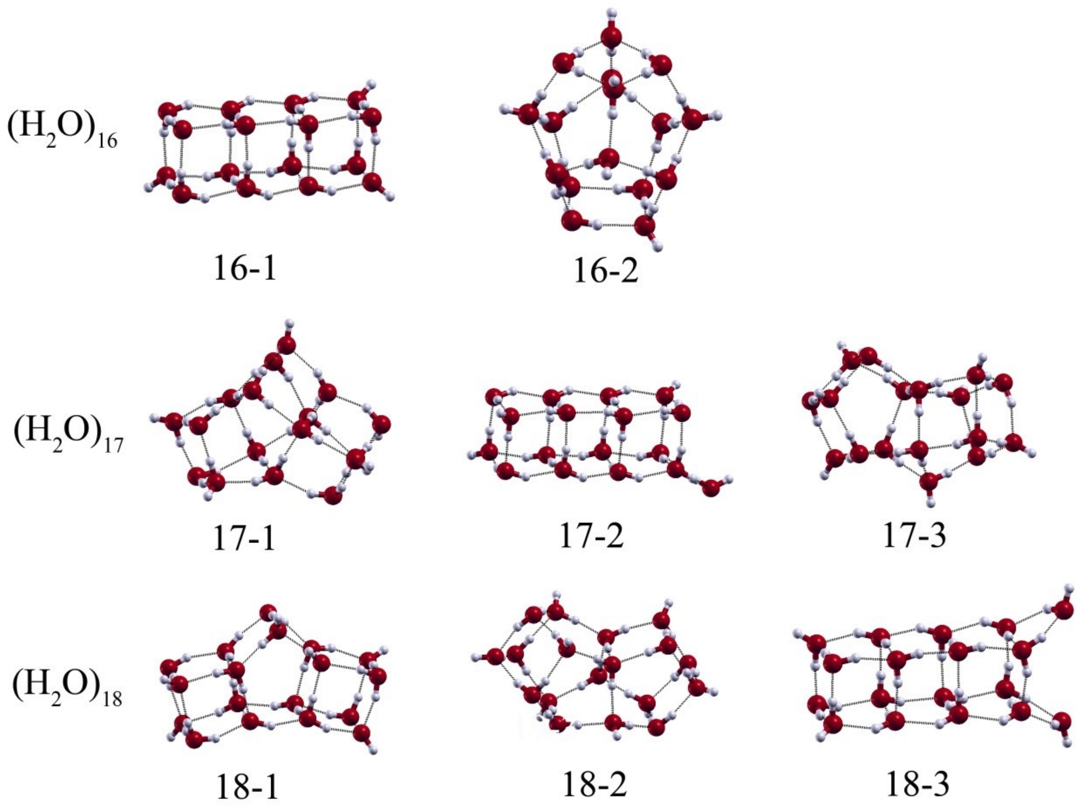
Supplementary Figure 9. The ML optimized low-lying geometrical configurations of $(H_2O)_n$ (n=2 - 5).



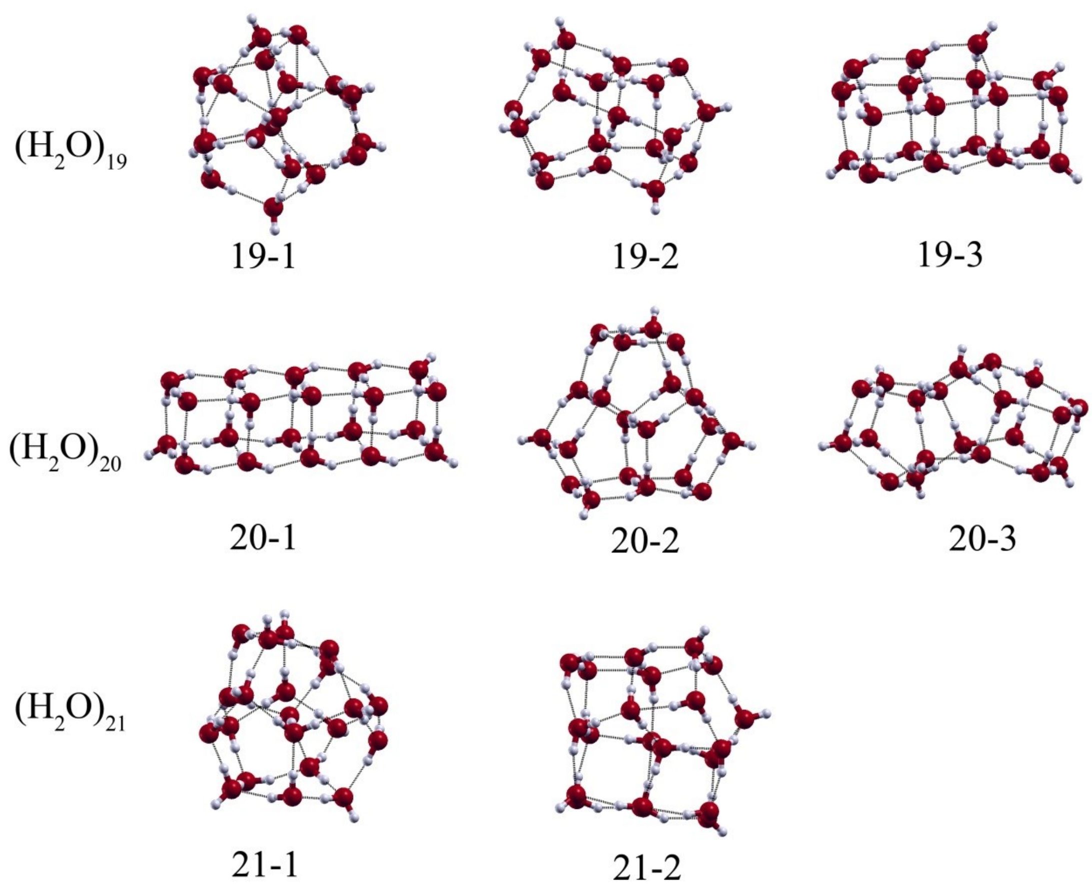
Supplementary Figure 10. The ML optimized low-lying geometrical configurations of $(H_2O)_n$ (n=6 - 10).



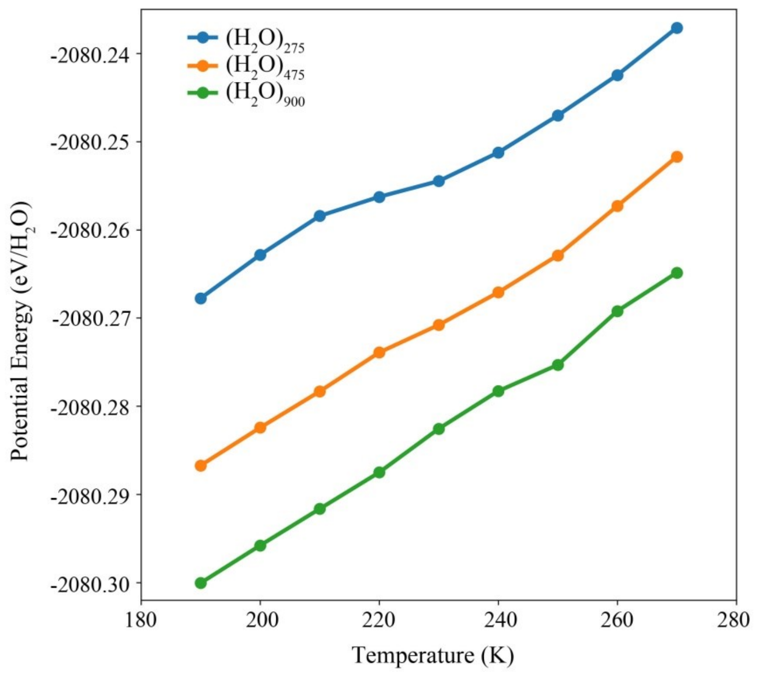
Supplementary Figure 11. The ML optimized low-lying geometrical configurations of $(H_2O)_n$ (n=11 - 15).



Supplementary Figure 12. The ML optimized low-lying geometrical configurations of $(H_2O)_n$ (n=16 - 18).



Supplementary Figure 13. The ML optimized low-lying geometrical configurations of $(H_2O)_n$ (n=19 - 21).



Supplementary Figure 14. The temperature dependences of the potential energy of $(H_2O)_{275}$, $(H_2O)_{475}$ and $(H_2O)_{900}$ nanoclusters.

2. Trained ML Water Model

The trained ML water model is recorded in the Supplementary File, *ML_water_model.json*, in the JSON format. The keys and their description for the model JSON file are listed in the Table S1.

Table S1 The keys and their description for the model JSON file

Key	Data Type	Description
r_cutoff	float	the cutoff distance for the calculation of symmetry function
G2_X	Python dictionary object	the parameters for the calculation of G2 symmetry function of X (H/O) element
G4_X	Python dictionary object	the parameters for the calculation of G4 symmetry function of X (H/O) element
G_X_max	1×27 float array (X=H) 1×30 float array (X=O)	the maximum values for the symmetry functions of X (H/O) element
G_X_min	1×27 float array (X=H) 1×30 float array (X=O)	the minimum values for the symmetry functions of X (H/O) element
G_X_mean	1×27 float array (X=H) 1×30 float array (X=O)	the mean values for the symmetry functions of X (H/O) element
weight_X_0	27×32 float array (X=H) 30×32 float array (X=O)	the values of the weights of the 1st hidden layer of the network for X (H/O) element
weight_X_1	32×32 float array	the values of the weights of the 2nd hidden layer of the network for X (H/O) element
weight_X_2	32×1 float array	the values of the weights of the 3rd hidden layer of the network for X (H/O) element
bias_X_0	1×32 float array	the values of the biases of the 1st hidden layer of the network for X (H/O) element
bias_X_1	1×32 float array	the values of the biases of the 2nd hidden layer of the network for X (H/O) element
bias_X_2	1×1 float array	the value of the bias of the 3rd hidden layer of the network for X (H/O) element

The atomic energy of a H/O atom can be calculated via the following procedure:
1. calculate the G2 and G4 symmetry function descriptor

$$G_i^2 = \sum_{j \neq i} e^{-\eta(r_{ij} - r_s)^2 f_c(r_{ij})};$$

$$G_i^4 = 2^{1-\xi} \sum_{j \neq i} \sum_{k \neq i,j} \left[(1 + \lambda cos(\alpha_{ijk}))^{\xi} e^{-\eta(r_{ij}^2 + r_{ik}^2 + r_{jk}^2)} f_c(r_{ij}) f_c(r_{ik}) f_c(r_{jk}) \right];$$

$$f_c(r_{ij}) = [\tanh(1 - \frac{r_{ij}}{r_{cutoff}})]^3;$$

$$G_i = [G_i^2, G_i^4];$$

where the parameters are all included in the *r_cutoff*, *G2_X*, and *G4_X*.
2. normalize the symmetry function descriptor

$$G_{in} = (G_i - G_{mean})/(G_{max} - G_{min});$$

3. calculate the atomic energy

$$L1 = tanh(G_{in} \times weight_0 + bias_0);$$

$$L2 = tanh(L1 \times weight_1 + bias_1);$$

$$E_i = L2 \times weight_2 + bias_2;$$

4. calculate the potential energy

$$E = \sum E_i.$$

The atomic forces are calculated analytically by $F = \frac{dE}{dR} = \frac{dE}{dG} \times \frac{dG}{dR}$. Both items $\frac{dE}{dG}$ and $\frac{dG}{dR}$ are calculated using the automatic differentiation function of TensorFlow.

References

1. Larsen, A. H. et al. The atomic simulation environment-a Python library for working with atoms. Journal of Physics-Condensed Matter **29**, 273002 (2017).