ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

MACHINE LEARNING OF THE DISPERSION INTERACTION IN PHOSPHORUS

Supervisors: Prof. Michele Ceriotti
Prof. Giuseppe Carleo
Kevin Kazuki Huguenin-Dumittan

Master Thesis

Haoran Ni

24th June 2022

Contents

1 Abstract **2**

2 Research background **3**

3 Representation of atomic systems **4**
3.1 The Dirac notation formalism . . . 4
3.2 Invariance under symmetry operations . . . 4
3.2.1 Translational invariance . . . 4
3.2.2 Rotational invariance . . . 6
3.3 The LODE representation . . . 8

4 Machine learning models **11**
4.1 Linear regression model . . . 11
4.2 The kernel method . . . 12

5 Learning of the dispersion interaction **13**
5.1 The exfoliation data set . . . 13
5.1.1 Manual extraction of the dispersion interaction potential and the SOAP model . . . 13
5.1.2 the LODE model and the SOAP + αLODE model . . . 15
5.2 The phosphorus allotropes data set . . . 16
5.2.1 The 2-body+SOAP+R6 model . . . 16
5.2.2 The SOAP+αLODE model . . . 17

6 Summary **21**

7 Acknowledgements **22**

8 Appendix **23**
8.1 Summary of training parameters and plots . . . 23
8.1.1 SOAP model trained on all 100 exfoliation frames . . . 23
8.1.2 SOAP+R6 model trained on all 100 exfoliation frames . . . 24
8.1.3 LODE model trained on all 100 exfoliation frames . . . 25
8.1.4 LODE model trained on the first 40 + the last exfoliation frames . . . 26
8.1.5 SOAP+αLODE model trained on the first 40 + the last exfoliation frames . . . 27
8.1.6 2bd+SOAP+R6 model trained on the total data set . . . 28
8.1.7 SOAP model trained on 1/10 of the total data set . . . 30
8.1.8 SOAP + αLODE model trained on 1/10 of the total data set . . . 31
8.2 Visualized representative frames from the phosphorus allotropes data set . . . 32

References **34**

1 Abstract

The integration of dispersion interactions in atomic machine learning (ML) models has been a challenging topic. People have used methods such as baselining[1] and local parametrization[2] to approximate explicitly the dispersion behavior at long distances. The long-distance equivariant (LODE) framework[3] was recently proposed as a data-driven ML method to learn the long-range interactions in atomic systems, but more examinations of the capability of this method are yet to be performed, especially how well it can capture the long-range interactions from a general data set. In this project, we will study and compare three different types of ML models, namely a pure short-range model using SOAP (Smooth Overlap of Atomic Positions)[4] features, a short-range model using SOAP features combined with an explicit r^{-6} (R6) model, a multiscale model using combined features of SOAP and LODE. All three types of models are trained and compared upon two data sets, the exfoliation of black phosphorus data set and the phosphorus allotropes data set[5]. We will show explicitly that localized short-range models are not able to learn the dispersion interaction from the training set. In contrast, the multiscale model combining SOAP and LODE features is able to accurately capture the dispersion interaction from a general phosphorus data set, and can correctly reproduce the binding curve of black phosphorus.

2 Research background

In modern day research, computational methods which rely on large-scale supercomputers, have gained the same importance as experimental and theoretical studies. With the current computational modeling methods such as density functional theory (DFT), researchers are able to perform *ab initio* calculations on realistic physical systems and predict novel properties without the reliance on experimental data, which has yielded lots of significant research papers in the field of physics, chemistry and materials science [6, 7, 8].

However, the comparable-to-experiment accuracy of DFT methods results from the fact that they are first principles calculations, and the computational costs of *ab initio* methods scale with the cube of the number of atoms $\mathcal{O}(N^3)$[9]. Thus the calculations immediately become formidable for systems with hundreds of atoms. On the other hand, force field methods have computational costs several orders smaller than DFT because they only depend on the internal positional and elementary parameters of the systems[10], and simulations over millions of atoms and millisecond time scales have been performed[11, 12]. As a consequence of the parametrized nature, force field methods are not as accurate as DFT methods. For simulations with high accuracy requirements, people still prefer DFT despite its high computational cost.

This situation is changed by the introduction of ML methods in atomic modeling, making the construction of accurate and less computationally expensive models possible. One key characteristic of the ML methods is that they are able to extract/fit a pattern from a large amount of data, which makes it an appropriate candidate for atomic modeling, where people can train potential energy surface (PES) models of certain elements on DFT data. With finely trained ML PES models, highly accurate and fast calculations can be performed, and the simulations of large scale systems and long time processes become possible[13, 14, 15].

To reduce the computational costs when training the ML model, people apply a smooth cutoff function to restrict each atomic neighborhood to a finite sphere, so that distant atoms with minor contributions are neglected. But for systems where the long-range interactions are important, such as water or graphene, a localized model can lead to significant deviations from the real values. As a workaround, explicit physics-based methods to capture the long-range interactions have been proposed[1, 2]. But these methods only apply in certain circumstances, the same process of extracting the long-range behavior has to be repeated every time we have a different system. A data-driven method to capture the long-range interactions in atomic systems, known as the long-distance equivariant (LODE) framework,was proposed by A. Grisafi and M. Ceriotti[3], where an atomic density potential representation is constructed to fit the long-range interactions of different kinds. They have verified the capability of LODE in capturing the long-range interactions in small-scale data sets.

As a further step to [3], in this project we aim at applying LODE to a more general data set where different allotropes of phosphorus are included, and test the transferrability and capability of capturing the dispersion interaction of the model on the exfoliation data set. We first compare the learning capability of the SOAP model, the SOAP + R6 model, the LODE model and the combined SOAP + LODE model in terms of the long-range dispersion interaction on the exfoliation of black phosphorus data set. Then we move on to a much larger data set with 4798 training frames to test the transferrability and the learning capability of the dispersion interaction of the commonly used 2-body + SOAP + R6 model, which is also compared with a localized 2-body + SOAP model. In the last section, we take 400 training frames from the phosphorus data set and train a SOAP + LODE model. The transferrability and long-range learning capability of this model are tested on the exfoliation frames. From the contrast against the pure SOAP model, we show that a transferrable ML atomic potential model with correct long-range dispersion behavior can be constructed using a combined SOAP + LODE model.

3 Representation of atomic systems

The definitions and derivations used here mostly come from the references [16, 17].

3.1 The Dirac notation formalism

The representations of atomic systems can be better formalized under **the Dirac notation**. We represent an atomic configuration by a ket $|A\rangle$, which in essence contains the information about the position of each atom and the elementary composition of this system. And since position and chemical composition are independent information, we can express $|A\rangle$ in terms of the outer product between these two:

$$|A\rangle = \sum_i |\boldsymbol{r}_i\rangle \otimes |\alpha_i\rangle \,, \tag{1}$$

where $|\boldsymbol{r}\rangle$ stands only for the position information of atomic configuration $|A\rangle$, and $|\alpha\rangle$ only for the composition information. The sum index i stands for each atom in this configuration.

Furthermore, if we consider a Gaussian distribution of atomic density $g(\boldsymbol{r})$ of each atom, then in position

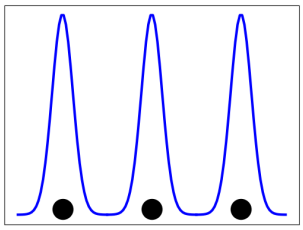


Figure 1: Demonstration of Gaussian atomic density

space we have a more explicit expression of Eq.(1):

$$\langle \boldsymbol{r}|A\rangle = \sum_i g(\boldsymbol{r} - \boldsymbol{r}_i)\, |\alpha_i\rangle \,, \tag{2}$$

where we have omitted the outer product symbol.

3.2 Invariance under symmetry operations

The representation given above is not ideal since it does not possess the symmetries of real physical systems, such as translational, rotational and permutational invariance. To do this, we will integrate $|A\rangle$ over a certain symmetry groups, equivalently, we are averaging over a symmetry group. This procedure is also known as the **Haar integration**[18]:

$$|A\rangle_{\hat{S}} = \int d\hat{S}\, \hat{S}\, |A\rangle \,, \tag{3}$$

where $\hat{S}$ stands for the symmetry group such as the translation group $\hat{t}$ or the rotation group $\hat{R}$.

3.2.1 Translational invariance

Let us first consider the translational invariance. Based on Eq.(3), we integrate $|A\rangle$ over the translation group, which is equivalent to the integration over $\mathbb{R}^3$, under the real space basis $\langle \boldsymbol{r}|$:

$$\langle \boldsymbol{r}|A\rangle_{\hat{t}} = \int d\hat{t}\ \langle \boldsymbol{r} \,|\, \hat{t} \,|\, A\rangle \tag{4}$$

$$= \sum_i |\alpha_i\rangle \int d\hat{t}\, \langle \boldsymbol{r} \,|\, \hat{t} \,|\, \boldsymbol{r}_i\rangle \tag{5}$$

$$= \sum_i |\alpha_i\rangle \int d\hat{t} \, \langle \boldsymbol{r} + \hat{t} \,|\, \boldsymbol{r}_i \rangle \tag{6}$$

$$= \sum_i |\alpha_i\rangle \int d\boldsymbol{t} \, g(\boldsymbol{r} + \boldsymbol{t} - \boldsymbol{r}_i) \tag{7}$$

$$= \sum_\alpha N_\alpha \, |\alpha\rangle \,, \tag{8}$$

where we have used Eq.(1), Eq.(2), and N_α stands for the number of atoms of species $|\alpha\rangle$.

We see from above that the position information is completely lost during the integration. To avoid this, we define the tensor product of $|A\rangle$ and the corresponding Haar integration:

$$\left|A^{(\nu)}\right\rangle = \underbrace{|A\rangle \otimes |A\rangle \otimes ... \otimes |A\rangle}_{\nu}, \tag{9}$$

$$\left|A^{(\nu)}\right\rangle_{\hat{t}} = \int d\hat{t} \, \underbrace{\hat{t}\,|A\rangle \otimes \hat{t}\,|A\rangle \otimes ... \otimes \hat{t}\,|A\rangle}_{\nu} . \tag{10}$$

Then we consider the case where $\nu = 2$. We perform the same integration over the translation group but in this case under two independent real space bases $\langle \boldsymbol{r}|$ and $\langle \boldsymbol{r'}|$:

$$\langle \boldsymbol{r}\boldsymbol{r'}|A^{(2)}\rangle_{\hat{t}} = \int d\hat{t} \, \langle \boldsymbol{r} \,|\, \hat{t} \,|\, A\rangle \, \langle \boldsymbol{r'} \,|\, \hat{t} \,|\, A\rangle \tag{11}$$

$$= \sum_{ij} |\alpha_i \alpha_j\rangle \int d\hat{t} \, \langle \boldsymbol{r} + \hat{t} \,|\, \boldsymbol{r}_i \rangle \, \langle \boldsymbol{r'} + \hat{t} \,|\, \boldsymbol{r}_j \rangle \tag{12}$$

$$= \sum_{ij} |\alpha_i \alpha_j\rangle \int d\boldsymbol{t} \, g(\boldsymbol{r} + \boldsymbol{t} - \boldsymbol{r}_i) \, g(\boldsymbol{r'} + \boldsymbol{t} - \boldsymbol{r}_j) \tag{13}$$

$$= \sum_{ij} |\alpha_i \alpha_j\rangle \, h(\boldsymbol{r} - \boldsymbol{r'} - \boldsymbol{r}_{ij}), \tag{14}$$

where in the last step we have used the convolution property of the Gaussian function, and h is a Gaussian with twice the variance of g.

We notice that there is a redundancy of using two independent real space bases $\langle \boldsymbol{r}\boldsymbol{r'}|$ because Eq.(14) depends only on $\boldsymbol{r} - \boldsymbol{r'}$, so we simply write:

$$\langle \boldsymbol{r}|A^{(2)}\rangle_{\hat{t}} = \sum_{ij} |\alpha_i \alpha_j\rangle \, h(\boldsymbol{r} - \boldsymbol{r}_{ij}), \tag{15}$$

where $\boldsymbol{r}$ here is the substitution for $\boldsymbol{r} - \boldsymbol{r'}$ in Eq.(14).

Atom-centered description Furthermore, from Eq.(15) we can group together the terms only related to atom i:

$$\langle \boldsymbol{r}|A^{(2)}\rangle_{\hat{t}} = \sum_i |\alpha_i\rangle \sum_j |\alpha_j\rangle \, h(\boldsymbol{r} - \boldsymbol{r}_{ij}) \tag{16}$$

$$\equiv \sum_i |\alpha_i\rangle \, \langle \boldsymbol{r} \,|\, \rho_i \rangle \,, \tag{17}$$

thus

$$|A^{(2)}\rangle_{\hat{t}} = \sum_i |\alpha_i\rangle \, |\rho_i\rangle \,, \tag{18}$$

where $|\rho_i\rangle$ is the local environment centered on atom i. Hence, we see that an atom-centered description naturally results from the integration over translation group.

To clarify, the L.H.S. and R.H.S. of Eq.(17) are still kets, not scalars, because we haven't dealt with the basis related to chemical compositions. A more complete expression of Eq.(17) will be

$$\langle \alpha \boldsymbol{r}|A^{(2)}\rangle_{\hat{t}} = \sum_i \delta_{\alpha\alpha_i} \, \langle \boldsymbol{r} \,|\, \rho_i \rangle \,. \tag{19}$$

Locality In real implementations, a smooth cutoff function $f_c(r_{ij})$ is added to the representation to confine the environment of each atom in a finite neighbourhood to lower the computational costs, we have:

$$\langle \boldsymbol{r} \,|\, \rho_i \rangle \equiv \sum_j |\alpha_j\rangle \, h(\boldsymbol{r} - \boldsymbol{r}_{ij}) f_c(r_{ij}). \tag{20}$$

If atom j is close to atom i, namely r_{ij} small, then $f_c(r_{ij}) \to 1$. If they are far apart, namely r_{ij} larger than the cutoff radius, then $f_c(r_{ij}) = 0$.

Though adding the cutoff function has reduced the computational cost, a clear drawback of it is that we can no longer consider the contributions from the atoms outside the cutoff radius, which makes us incapable of capturing the long-range interactions. Further more, even without the cutoff function, the quick decay of the Gaussian-like atomic density prevents us from learning the long-range interaction which decays much slower than the Gaussian, such as the dispersion interaction. We will introduce *the long-distance equivarant (LODE)* framework[3] to see how we can learn the correct behavior of the long-range interactions in later chapters.

3.2.2 Rotational invariance

For the rotational invariance, we perform the same steps as in section 3.2.1 to have

$$\left|A^{(\nu)}\right\rangle_{\hat{R}} = \int d\hat{R} \, \underbrace{\hat{R}\,|A\rangle \otimes \hat{R}\,|A\rangle \otimes \ldots \otimes \hat{R}\,|A\rangle}_{\nu}, \tag{21}$$

where $\hat{R}$ stands for the rotation group. We also see notations like $|\overline{A^{\otimes\nu}}\rangle$, which has the same meaning as Eq.(21), and the line above means the average over the rotation group. From now on we will use the later notation.

A convenient basis to evaluate the rotationally-invariant representations is the radial function $R_n(r) = \langle r \,|\, n\rangle$ and the spherical harmonic $Y_l^m(\hat{\boldsymbol{r}}) = \langle \hat{\boldsymbol{r}} \,|\, lm\rangle$, because we can derive explicit expansion coefficients using them. Notice that $\boldsymbol{r} = (r, \hat{\boldsymbol{r}})$, where $\boldsymbol{r}$ is the 3-dimensional position vector, r is the length of $\boldsymbol{r}$, and $\hat{\boldsymbol{r}}$ is a unit vector which only contains the angular (directional) information of $\boldsymbol{r}$.

To clarify,

$$\begin{aligned}
\langle \alpha n l m \,|\, A\rangle &= \int d\boldsymbol{r} \, \langle \alpha n l m \,|\, \boldsymbol{r}\rangle \, \langle \boldsymbol{r} \,|\, A\rangle && (22)\\
&= \sum_i \delta_{\alpha,\alpha_i} \int d\boldsymbol{r} \, \langle \alpha n l m \,|\, \boldsymbol{r}\rangle \, \langle \boldsymbol{r} \,|\, \boldsymbol{r}_i\rangle && (23)\\
&= \sum_i \delta_{\alpha,\alpha_i} \int d\boldsymbol{r} \, \langle n \,|\, r\rangle \, \langle l m \,|\, \hat{\boldsymbol{r}}\rangle \, \langle r, \hat{\boldsymbol{r}} \,|\, \boldsymbol{r}_i\rangle \,, && (24)
\end{aligned}$$

where we have used Eq.(1).

Now we can derive the explicit expressions for the symmetrized field representations of order $\nu = 1, 2$:

$$\begin{aligned}
\langle \alpha_1 n_1 l_1 m_1 | \overline{\rho_i^{\otimes 1}} \rangle &= \int d\hat{R} \, \langle \alpha_1 n_1 l_1 m_1 | \hat{R} | \rho_i \rangle && (25)\\
&= \sum_{lm} \int d\hat{R} \, \langle \alpha_1 n_1 l_1 m_1 | \hat{R} | \alpha_1 n_1 l m \rangle \, \langle \alpha_1 n_1 l m | \rho_i \rangle && (26)\\
&= \sum_{lm} \int d\hat{R} \, \langle l_1 m_1 | \hat{R} | l m \rangle \, \langle \alpha_1 n_1 l m | \rho_i \rangle && (27)\\
&= \sum_{lm} \int d\hat{R} \, \boldsymbol{D}^{l_1}_{m_1 m} \delta_{l_1 l} \, \langle \alpha_1 n_1 l m | \rho_i \rangle && (28)\\
&= \sum_{m} \langle \alpha_1 n_1 l_1 m | \rho_i \rangle \int d\hat{R} \, \boldsymbol{D}^{l_1}_{m_1 m} && (29)
\end{aligned}$$

$$= \sum_m \langle \alpha_1 n_1 l_1 m | \rho_i \rangle \int d\hat{R}\, \boldsymbol{D}^{l_1}_{m_1 m} \boldsymbol{D}^{0}_{00} \tag{30}$$

$$= \frac{8\pi^2}{2l_1+1} \delta_{m_1,0}\delta_{l_1,0} \langle \alpha_1 n_1 l_1 m_1 \,|\, \rho_i \rangle \,, \tag{31}$$

where $|\rho_i\rangle$ is the atom-centered representation obtained from the second order integration over the translational group. From the second line to third line we have used the fact that $\hat{R}$ does not operate on $|\alpha_1 n_1\rangle$ and that $\langle \alpha_1 n_1 \,|\, \alpha_1 n_1 \rangle = 1$. In the last step we have used the orthogonality property of the Wigner matrix.

$$\langle \alpha_1 n_1 l_1 m_1; \alpha_2 n_2 l_2 m_2 | \overline{\rho_i^{\otimes 2}} \rangle = \int d\hat{R} \langle \alpha_1 n_1 l_1 m_1 | \hat{R} | \rho_i \rangle \langle \alpha_2 n_2 l_2 m_2 | \hat{R} | \rho_i \rangle \tag{32}$$

$$= \sum_{lm} \sum_{l'm'} \int d\hat{R} \langle \alpha_1 n_1 l_1 m_1 | \hat{R} | \alpha_1 n_1 l m \rangle \langle \alpha_1 n_1 l m | \rho_i \rangle \tag{33}$$

$$\langle \alpha_2 n_2 l_2 m_2 | \hat{R} | \alpha_2 n_2 l' m' \rangle \langle \alpha_2 n_2 l' m' | \rho_i \rangle$$

$$= \sum_{lm} \sum_{l'm'} \int d\hat{R} \boldsymbol{D}^{l_1}_{m_1 m} \delta_{l_1 l} \boldsymbol{D}^{l_2}_{m_2 m'} \delta_{l_2 l'} \langle \alpha_1 n_1 l m | \rho_i \rangle \langle \alpha_2 n_2 l' m' | \rho_i \rangle \tag{34}$$

$$= \sum_{mm'} \int d\hat{R} \boldsymbol{D}^{l_1}_{m_1 m} \boldsymbol{D}^{l_2}_{m_2 m'} \langle \alpha_1 n_1 l_1 m | \rho_i \rangle \langle \alpha_2 n_2 l_2 m' | \rho_i \rangle \tag{35}$$

$$= \sum_{mm'} (-1)^{m_1 - m} \int d\hat{R} \boldsymbol{D}^{l_1\,*}_{-m_1,-m} \boldsymbol{D}^{l_2}_{m_2 m'} \langle \alpha_1 n_1 l_1 m | \rho_i \rangle \langle \alpha_2 n_2 l_2 m' | \rho_i \rangle \tag{36}$$

$$= \sum_{mm'} (-1)^{m - m_1} \int d\hat{R} \boldsymbol{D}^{l_1\,*}_{m_1,m} \boldsymbol{D}^{l_2}_{m_2 m'} \langle \alpha_1 n_1 l_1 (-m) | \rho_i \rangle \langle \alpha_2 n_2 l_2 m' | \rho_i \rangle \tag{37}$$

$$= \sum_{mm'} (-1)^{m - m_1} \langle \alpha_1 n_1 l_1 (-m) | \rho_i \rangle \langle \alpha_2 n_2 l_2 m' | \rho_i \rangle \int d\hat{R} \boldsymbol{D}^{l_1\,*}_{m_1,m} \boldsymbol{D}^{l_2}_{m_2 m'} \tag{38}$$

$$= \sum_{mm'} (-1)^{m - m_1} \frac{8\pi^2}{2l_1+1} \langle \alpha_1 n_1 l_1 (-m) | \rho_i \rangle \langle \alpha_2 n_2 l_2 m' | \rho_i \rangle \delta_{l_1 l_2} \delta_{m_1 m_2} \delta_{m m'} \tag{39}$$

$$= \sum_{m} (-1)^{m - m_1} \frac{8\pi^2}{2l_1+1} \langle \alpha_1 n_1 l_1 (-m) | \rho_i \rangle \langle \alpha_2 n_2 l_2 m | \rho_i \rangle \delta_{l_1 l_2} \delta_{m_1 m_2}, \tag{40}$$

where we have used the orthogonality property and complex conjugate property of the Wigner matrix.

Similar to the case of integration over the translation group, we have redundancy in indices after symmetrization for the rotation group. For example, for $\nu = 1$, the expansion coefficients can be rewritten as

$$\langle \alpha n l m | \overline{\rho_i^{\otimes 1}} \rangle = \frac{8\pi^2}{2l+1} \delta_{m,0}\delta_{l,0} \langle \alpha n l m \,|\, \rho_i \rangle \tag{41}$$

$$= \frac{8\pi^2}{2l+1} \langle \alpha n 0 0 \,|\, \rho_i \rangle \tag{42}$$

$$= \frac{8\pi^2}{2l+1} \langle \alpha n \,|\, \rho_i \rangle \,, \tag{43}$$

which is in nature a pair correlation function

$$\langle \alpha n | \overline{\rho_i^{\otimes 1}} \rangle = \int d\hat{R} \langle \alpha n | \hat{R} | \rho_i \rangle \tag{44}$$

$$= \int d\hat{R} \langle \alpha n \,|\, \rho_i \rangle \tag{45}$$

$$\propto \langle \alpha n \,|\, \rho_i \rangle \tag{46}$$

$$= \int d\boldsymbol{x}\, \langle n|x\rangle\, \langle 00|\hat{x}\rangle\, \langle \alpha \boldsymbol{x}|\rho_i\rangle \tag{47}$$

$$\propto \int dx\, x^2\, \langle n|x\rangle \int d\hat{x}\, \langle \alpha \boldsymbol{x}|\rho_i\rangle \tag{48}$$

$$\propto \int dr\, r^2 R_n^*(r) g_\alpha(r), \tag{49}$$

where we have used $\hat{R}\,|n\rangle = |n\rangle$, and $g_\alpha(r)$ stands for the radial distribution of α atoms of the i-atom-centered environment, which is called a *pair correlation function*.

For $\nu = 2$, the expansion coefficients can be rewritten as

$$\langle \alpha_1 n_1; \alpha_2 n_2; l|\overline{\rho_i^{\otimes 2}}\rangle = \frac{(-1)^l}{\sqrt{2l+1}} \sum_m (-1)^m \langle \alpha_1 n_1 lm|\rho_i\rangle \langle \alpha_2 l_2 l(-m)|\rho_i\rangle\,, \tag{50}$$

which corresponds to the *SOAP* features that are used as the short-range part of our model. The $\nu = 2$ case can also be expressed in real space basis $\langle \alpha_1 r_1; \alpha_2 r_2; \omega|\overline{\rho_i^{\otimes 2}}\rangle$, which shows its nature as a three-body correlation function:

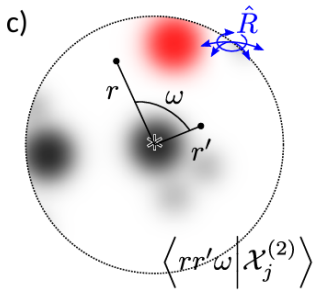


Figure 2: Demonstration of the three-body correlation representation

3.3 The LODE representation

As mentioned before, using the Gaussian atomic density and applying the cutoff function in the representation will disable the model from learning the long-range interactions. Here, we introduce *the long-distance equivarant (LODE)* framework[3] to see how we can construct the representation that is capable of capturing the long-range interactions.

Based on our previous definition of the Dirac notation Eq.(2), the *atomic-density potential representation* is defined as

$$\langle \boldsymbol{r} \,|\, \mathcal{V}^p \rangle = \sum_i |\alpha_i\rangle \int d\boldsymbol{r}' \frac{g(\boldsymbol{r}' - \boldsymbol{r}_i)}{|\boldsymbol{r}' - \boldsymbol{r}|^p}, \tag{51}$$

where p characterizes the long-range behavior of this representation, e.g. $p = 1$ for the electrostatic interaction, $p = 6$ for the dispersion interaction.

Compared with Eq.(2), Eq.(51) introduces a non-localized behavior that is determined by the value of p, therefore the central atom can still "feel" the existences of the atoms far away whose density decay asymptotically the same at long distances as the long-range interactions. In the following derivation, we will consider the special case where $p = 1$ to derive the explicit function that we use in the representation.

Consider the Gaussian atomic densities of species α,

$$\rho_\alpha(\boldsymbol{r}) = \frac{1}{(2\pi\sigma^2)^{3/2}} \sum_{i\in\alpha} e^{-\frac{(\boldsymbol{r}-\boldsymbol{r}_i)^2}{2\sigma^2}}, \tag{52}$$

which satisfies $\int_\Omega \rho_\alpha(\boldsymbol{r}) = N_\alpha$, where Ω is the total volume and N_α is the number of atoms of species α. σ is called the *smearing*. The Fourier transform of this Gaussian density is,

$$\rho_\alpha(\boldsymbol{k}) = \frac{1}{\Omega}\int_\Omega \rho_\alpha(\boldsymbol{r}) e^{-i\boldsymbol{k}\boldsymbol{r}} d\boldsymbol{r} \tag{53}$$

$$= \frac{1}{\Omega}\sum_{i\in\alpha}\int_\Omega \frac{1}{(2\pi\sigma^2)^{3/2}} e^{-\frac{(\boldsymbol{r}-\boldsymbol{r}_i)^2}{2\sigma^2}} e^{-i\boldsymbol{k}\boldsymbol{r}} d\boldsymbol{r} \tag{54}$$

$$= \frac{1}{\Omega}\sum_{i\in\alpha}\int_\Omega \frac{1}{(2\pi\sigma^2)^{3/2}} e^{-\frac{(\boldsymbol{r}-\boldsymbol{r}_i)^2}{2\sigma^2}} e^{-i\boldsymbol{k}(\boldsymbol{r}-\boldsymbol{r}_i)} d(\boldsymbol{r}-\boldsymbol{r}_i) e^{-i\boldsymbol{k}\boldsymbol{r}_i} \tag{55}$$

$$= \frac{1}{\Omega}\left(\sum_{i\in\alpha} e^{-i\boldsymbol{k}\boldsymbol{r}_i}\right) e^{-\frac{k^2\sigma^2}{2}}. \tag{56}$$

From poisson equation $\nabla^2 V(\boldsymbol{r}) = -4\pi\rho(\boldsymbol{r})$, we have

$$\langle \alpha\boldsymbol{r}|\mathcal{V}\rangle = \int\int -4\pi\rho_\alpha(\boldsymbol{r}) d\boldsymbol{r} d\boldsymbol{r} \tag{57}$$

$$= \sum_{\boldsymbol{k}\neq 0}\frac{4\pi}{k^2}\rho_\alpha(\boldsymbol{k}) e^{i\boldsymbol{k}\boldsymbol{r}} \tag{58}$$

$$= \frac{1}{\Omega}\sum_{\boldsymbol{k}\neq 0}\left(\sum_{i\in\alpha} e^{-i\boldsymbol{k}\boldsymbol{r}_i}\right)\frac{4\pi}{k^2} e^{-\frac{k^2\sigma^2}{2}} e^{i\boldsymbol{k}\boldsymbol{r}}. \tag{59}$$

Using the plane wave expansion

$$e^{i\boldsymbol{k}\boldsymbol{r}} = 4\pi\sum_{l=0}^{\infty}\sum_{|m|\leq l} i^l j_l(kr) Y^*_{lm}(\hat{\boldsymbol{k}}) Y_{lm}(\hat{\boldsymbol{r}}), \tag{60}$$

where $\hat{\boldsymbol{k}}$ and $\hat{\boldsymbol{r}}$ stand for the angular part of $\boldsymbol{k}$ and $\boldsymbol{r}$, and j_l is a spherical Bessel function. We have

$$\langle \alpha r l m|\mathcal{V}_j\rangle = \int \langle rlm|\boldsymbol{r}\rangle \langle \alpha\boldsymbol{r}|V_j\rangle \, d\boldsymbol{r} \tag{61}$$

$$= \int \langle r|\,\mathfrak{r}\rangle \langle lm|\hat{\boldsymbol{r}}\rangle \langle \alpha\boldsymbol{r}|V_j\rangle \, d\boldsymbol{r} \tag{62}$$

$$= \int \langle r|\,\mathfrak{r}\rangle \, Y^*_{lm}(\hat{\boldsymbol{r}})\frac{1}{\Omega}\sum_{\boldsymbol{k}\neq 0}\left(\sum_{i\in\alpha} e^{-i\boldsymbol{k}\boldsymbol{r}_i}\right)\frac{4\pi}{k^2} e^{-\frac{k^2\sigma^2}{2}} e^{i\boldsymbol{k}\boldsymbol{r}} d\boldsymbol{r} \tag{63}$$

$$= \int \langle r|\,\mathfrak{r}\rangle \, Y^*_{lm}(\hat{\boldsymbol{r}})\frac{1}{\Omega}\sum_{\boldsymbol{k}\neq 0}\left(\sum_{i\in\alpha} e^{-i\boldsymbol{k}\boldsymbol{r}_i}\right)\frac{16\pi^2}{k^2} e^{-\frac{k^2\sigma^2}{2}} \tag{64}$$

$$\sum_{l'=0}^{\infty}\sum_{|m'|\leq l'} i^{l'} j_{l'}(kr) Y^*_{l'm'}(\hat{\boldsymbol{k}}) Y_{l'm'}(\hat{\boldsymbol{r}}) d\boldsymbol{r} \tag{65}$$

$$= \int_{radial}\langle r|\,\mathfrak{r}\rangle \, d\mathfrak{r}\,\frac{1}{\Omega}\sum_{\boldsymbol{k}\neq 0}\left(\sum_{i\in\alpha} e^{-i\boldsymbol{k}\boldsymbol{r}_i}\right)\frac{16\pi^2}{k^2} e^{-\frac{k^2\sigma^2}{2}} \tag{66}$$

$$\sum_{l'=0}^{\infty}\sum_{|m'|\leq l'} i^{l'} j_{l'}(kr) Y^*_{l'm'}(\hat{\boldsymbol{k}}) \int_{angular} Y_{l'm'}(\hat{\boldsymbol{r}}) Y^*_{lm}(\hat{\boldsymbol{r}}) d\hat{\boldsymbol{r}} \tag{67}$$

$$= \int_{radial}\langle r|\,\mathfrak{r}\rangle \, d\mathfrak{r}\,\frac{1}{\Omega}\sum_{\boldsymbol{k}\neq 0}\left(\sum_{i\in\alpha} e^{-i\boldsymbol{k}\boldsymbol{r}_i}\right)\frac{16\pi^2}{k^2} e^{-\frac{k^2\sigma^2}{2}} \tag{68}$$

$$\sum_{l'=0}^{\infty}\sum_{|m'|\leq l'} i^{l'} j_{l'}(kr) Y^*_{l'm'}(\hat{\boldsymbol{k}}) \delta_{ll'}\delta_{mm'} \tag{69}$$

$$= \int_{radial} \langle r|\, \mathfrak{r}\rangle \, d\mathfrak{r} \, \frac{1}{\Omega} \sum_{\boldsymbol{k}\neq 0} \left(\sum_{i\in\alpha} e^{-i\boldsymbol{k}\boldsymbol{r}_i} \right) \frac{16\pi^2}{k^2} e^{-\frac{k^2\sigma^2}{2}} i^l j_l(kr) Y_{lm}^*(\hat{\boldsymbol{k}}) \tag{70}$$

$$= \frac{16\pi^2}{\Omega} \sum_{\boldsymbol{k}\neq 0} \left(\sum_{i\in\alpha} e^{-i\boldsymbol{k}\boldsymbol{r}_i} \right) \frac{e^{-\frac{k^2\sigma^2}{2}}}{k^2} i^l j_l(kr) Y_{lm}^*(\hat{\boldsymbol{k}}), \tag{71}$$

where $\mathfrak{r}$ only stands for the radial part of $\boldsymbol{r}$ and we have considered $\int_{radial} \langle r|\mathfrak{r}\rangle \, d\mathfrak{r} = 1$.

This representation does not yet possess the rotational invariance (if the number of angular functions is not set to zero), which is implemented by calculating the power spectrum of the above function.

4 Machine learning models

The definitions and derivations used here mostly come from the references [19, 20].

4.1 Linear regression model

The simplest machine learning model should be the linear regression model. It is *linear* in terms of the linearity of the *weights* of the fitted function. We define the linear model to be

$$y(\vec{x}, \vec{\omega}) = \sum_{i=1}^{M} \omega_i \phi_i(\vec{x}), \tag{72}$$

where $\vec{x}$ is the input data, ω is the weight, $\phi(\vec{x})$ is called the basis function, and M is the total number of basis functions. The above form can be reformulated as

$$\vec{y}(\vec{\omega}) = \hat{\Phi}\vec{\omega}, \tag{73}$$

where

$$\vec{\omega}^T = \begin{pmatrix} \omega_1 & \omega_2 & ... & \omega_M \end{pmatrix}, \tag{74}$$

and

$$\hat{\Phi} = \begin{pmatrix} \phi_1(\vec{x}_1) & \phi_2(\vec{x}_1) & ... & \phi_M(\vec{x}_1) \\ \phi_1(\vec{x}_2) & \phi_2(\vec{x}_2) & ... & \phi_M(\vec{x}_2) \\ ... & ... & ... & ... \\ \phi_1(\vec{x}_N) & \phi_2(\vec{x}_N) & ... & \phi_M(\vec{x}_N) \end{pmatrix} \tag{75}$$

is called the *design matrix*, where N is the total number of input points.

The training of the linear model is equivalent to finding the weights that minimize the *loss function*, which is defined as

$$\mathcal{L} = \frac{1}{2} \sum_{i=1}^{N} (t_i - y_i)^2 \tag{76}$$

$$= \frac{1}{2} \|\vec{t} - \vec{y}\|^2 \tag{77}$$

$$= \frac{1}{2} (\vec{t} - \vec{y})^T (\vec{t} - \vec{y}), \tag{78}$$

where $\vec{t}^T = \begin{pmatrix} t_1 & t_2 & ... & t_N \end{pmatrix}$ are the target values that we are trying to learn/fit.

Minimizing the loss function with respect to the weights gives

$$\nabla_{\vec{\omega}^T} \mathcal{L} = -\frac{1}{2} \hat{\Phi}^T (\vec{t} - \hat{\Phi}\vec{\omega}) = 0, \tag{79}$$

which yields

$$\vec{\omega} = (\hat{\Phi}^T \hat{\Phi})^{-1} \hat{\Phi}^T \vec{t}. \tag{80}$$

However, defining the loss function in this way will lead to *overfitting*, which means that the model is greatly tuned and distorted by the noises in the target values, thus the for such a model the transferrability will be weak even though it gives almost perfect predictions on the training data. To solve this problem, we usually add a *regularization* term in the loss function

$$\mathcal{L} = \frac{1}{2} \|\vec{t} - \vec{y}\|^2 + \frac{\lambda}{2} \|\vec{\omega}\|^2, \tag{81}$$

where λ is the regularization coefficient that controls the importance of the regularization term, in the context of this project we will call λ the *regularizer*.

The expression of calculating the weights that minimize the loss function after adding the regularization term is

$$\vec{\omega} = (\hat{\Phi}^T \hat{\Phi} + \lambda \mathbb{I})^{-1} \hat{\Phi}^T \vec{t}. \tag{82}$$

4.2 The kernel method

The idea behind the kernel method is that we can make predictions based on the *distance* between the new input data and the selected sparse training data. The *kernel* $k(\vec{x}, \vec{x}_m)$ is a function that measures the *similarity* of two points $\vec{x}$ and $\vec{x}_m$. Now the linear model is written as

$$y(\vec{x}, \vec{\omega}) = \sum_{m=1}^{M} \omega_m k(\vec{x}, \vec{x}_m), \tag{83}$$

$$\vec{y}(\vec{\omega}) = \hat{K}_{NM}\vec{\omega}, \tag{84}$$

where

$$\hat{K}_{NM} = \begin{pmatrix} k(\vec{x}_1, \vec{x}_{m_1}) & k(\vec{x}_1, \vec{x}_{m_2}) & ... & k(\vec{x}_1, \vec{x}_{m_M}) \\ k(\vec{x}_2, \vec{x}_{m_1}) & k(\vec{x}_2, \vec{x}_{m_2}) & ... & k(\vec{x}_2, \vec{x}_{m_M}) \\ ... & ... & ... & ... \\ k(\vec{x}_N, \vec{x}_{m_1}) & k(\vec{x}_N, \vec{x}_{m_2}) & ... & k(\vec{x}_N, \vec{x}_{m_M}) \end{pmatrix} \tag{85}$$

and $\vec{x}_m$ are the selected representative data points from the training set, which are called the *sparse points* or *representative points*, and M is the total number of sparse points.

There are many functions that can be used as the kernel to measure the similarity of two points. In our project, we use the dot product as the kernel function

$$k(\vec{x}, \vec{x}_m) = |\vec{x} \cdot \vec{x}_m|^z, \tag{86}$$

where z is referred as *zeta* during the training, which affects the performance of the model.

To avoid overfitting, the regularization term is still needed. For the kernel method, in most cases we choose $\frac{\lambda}{2}\|\omega\|^2_{K_{MM}} = \frac{\lambda}{2}\sum_{n,m}\omega_n k(\vec{x}_n, \vec{x}_m)\omega_m$ as the regularization term, thus the loss function can be written as

$$\mathcal{L} = \frac{1}{2}\|\vec{t} - \vec{y}\|^2 + \frac{\lambda}{2}\|\omega\|^2_{K_{MM}}. \tag{87}$$

With the above loss function, the optimal weights that minimize the loss function is obtained from

$$\vec{\omega} = (\hat{K}^T_{NM}\hat{K}_{NM} + \lambda\hat{K}_{MM})^{-1}\hat{K}^T_{NM}\vec{t}. \tag{88}$$

where

$$\hat{K}_{MM} = \begin{pmatrix} k(\vec{x}_{m_1}, \vec{x}_{m_1}) & k(\vec{x}_{m_1}, \vec{x}_{m_2}) & ... & k(\vec{x}_{m_1}, \vec{x}_{m_M}) \\ k(\vec{x}_{m_2}, \vec{x}_{m_1}) & k(\vec{x}_{m_2}, \vec{x}_{m_2}) & ... & k(\vec{x}_{m_2}, \vec{x}_{m_M}) \\ ... & ... & ... & ... \\ k(\vec{x}_{m_M}, \vec{x}_{m_1}) & k(\vec{x}_{m_M}, \vec{x}_{m_2}) & ... & k(\vec{x}_{m_M}, \vec{x}_{m_M}) \end{pmatrix} \tag{89}$$

Though using non-linear functions as the basis functions, the kernel model is still a linear model because of the linearity in the weights. If we choose the same regularization term as the linear regression model, the weights can be calculated using exactly the same expression as Eq.(82), and thus it is possible of combining these two models by expanding the basis functions, or equivalently, by stacking the design matrices.

5 Learning of the dispersion interaction

In this section, we will start with a simple yet representative training set, the exfoliation of black phosphorus, to compare the performances of three different models we mentioned at the beginning, namely a pure short-range model using SOAP features, a short-range model using SOAP features combined with an R6 model, a multiscale model using combined features of SOAP and LODE. Their long-range learning capability are especially mentioned. Then we move on to a larger and general data set, the phosphorus allotropes data set, to train the three models and compare their performances on reproducing the binding curve. The detailed training parameters of all the models shown here can be found in the Appendix 8.1.

5.1 The exfoliation data set

There are in total 100 frames in this training set[21], with each frame containing 8 atoms as shown below.

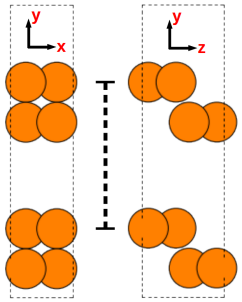


Figure 3: A representative training frame from the exfoliation data set. The length of the dotted line is defined as the interlayer distance.

We define the interlayer distance as the length of the dotted line in Fig.(3). In this training set, the interlayer distance changes from 3.84 Å to 13.74 Å evenly. The reference DFT plus many-body dispersion (DFT+MBD) energies and forces of these frames are calculated without doing relaxation, so the only difference between different frames is the interlayer distance. Below we show the DFT+MBD binding curve of this data set.

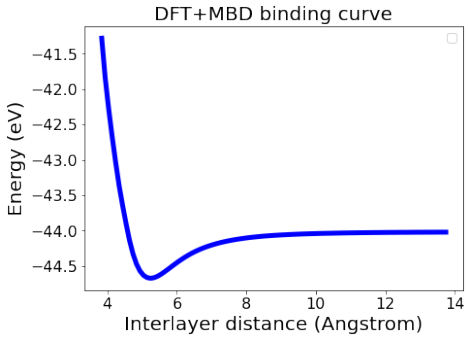


Figure 4: The DFT+MBD energy curve of the exfoliation data set.

5.1.1 Manual extraction of the dispersion interaction potential and the SOAP model

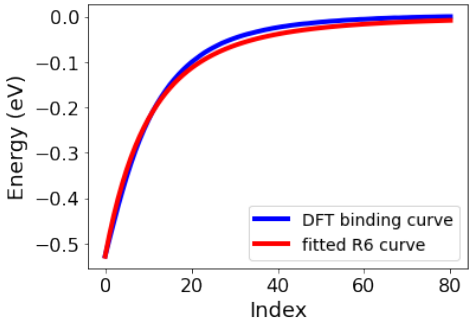
In this section we will train a model on the exfoliation data set using SOAP. However, SOAP itself is not capable of capturing the long-range interaction, so for the long-range part we have to extract the potential manually.

The dispersion potential (or R6 potential) is extracted from the exfoliation data set in the following steps:

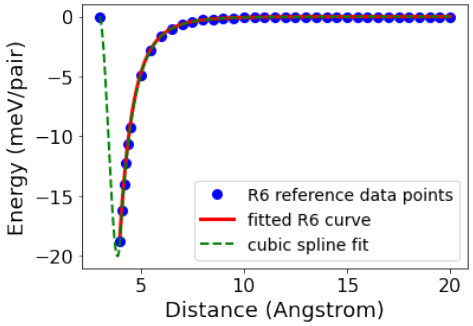
- Extend the unit cell, except for the exfoliation axis, such that all atoms within the neighbourhood, which is determined by the cutoff radius, of the atoms from the original cell are included

- Classify the atoms into the upper layer and the lower layer, then obtain all interlayer atom pairs and their distances
- Fit the dispersion potential, which is defined as $-\sum_{ij} C/r_{ij}^6$, where i and j denote the atoms from the upper and lower layers respectively

The fitted curve is shown in Fig.(5a). Here we use the energy with the largest interlayer distance as the reference energy, and it is subtracted from the energies of all other exfoliation frames. We see a slight mismatch of these two curves, which is expected because the long-range behavior of the exfoliation curve is not exactly $\sim 1/r^6$.



(a) Fitting of the long-range tail of the exfoliation curve.



(b) Comparison among the fitted curve, the reference data points and the cublic spline fit.

Figure 5: Manual extraction of the dispersion potential.

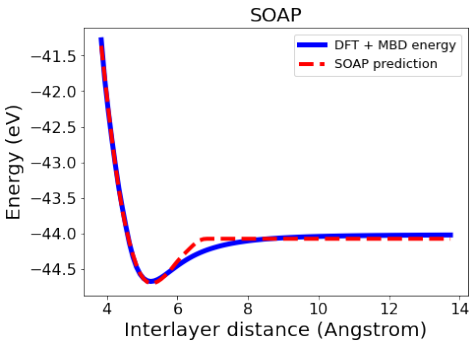
As a confirmation, we compare the fitted R6 curve with the data points of the R6 potential provided in [5], as shown in Fig.(5b). The eventual R6 potential is obtained by fitting a cubic spline function to the reference data points shown in Fig.(5b), where a zero point is added manually at 3Å.

Having obtained the R6 potential, we are prepared to train a SOAP + R6 model where the SOAP model is trained on

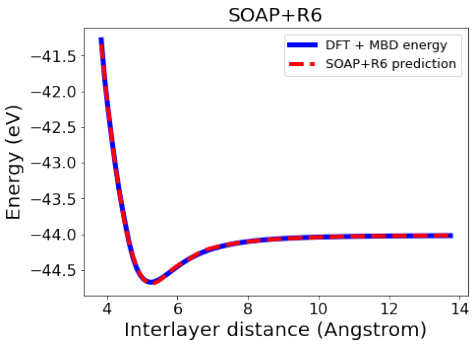
$$E_{train} = E_{DFT+MBD} - \sum_{i>j} V_{R6}(r_{ij}), \tag{90}$$

$$\boldsymbol{F}_{train} = \boldsymbol{F}_{DFT+MBD} + \boldsymbol{\nabla} \sum_{i>j} V_{R6}(r_{ij}). \tag{91}$$

The training results are shown below.



(a) Predicted binding curve of a pure SOAP model.



(b) Predicted binding curve of a SOAP+R6 model.

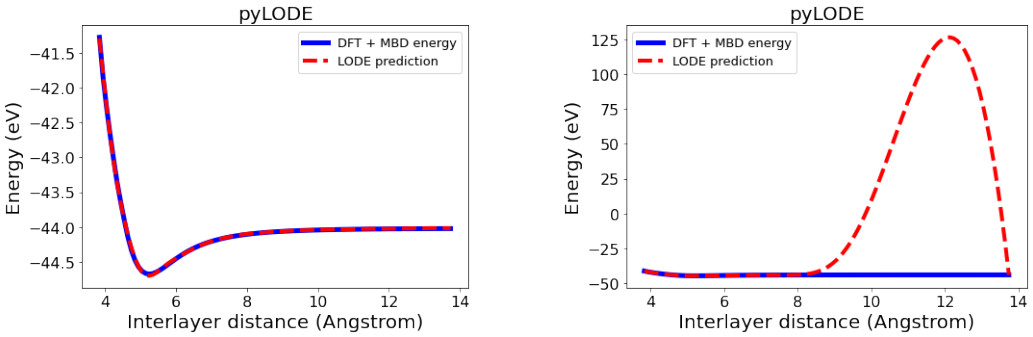
Figure 6: Comparison of the binding curve predictions between the SOAP model and the SOAP+R6 model.

In Fig.(6a), we see that the energy predictions are all flat for interlayer distances larger than the cutoff distance, which demonstrates the problem we mentioned before about the locality of the SOAP model that

the environmental changes outside the cutoff are ignored. When combined with the R6 model, we see in Fig.(6b) that the long-range tail of the binding curve is perfectly captured. As a matter of fact, the training energies after subtracting the R6 potential are flat for the long-range part, therefore in Fig.(6b) the long-range behavior is completely determined by the manually extracted R6 potential, while the SOAP model only determines the baseline.

5.1.2 the LODE model and the SOAP + αLODE model

The powerful aspect of LODE is that one can still capture the long-range interaction without the need to manually extract it, which makes it possible to learn the long-range interactions in systems where the layered structures, from which the long-range interaction is fitted, are not existent. Furthermore, the training of the model is much simpler than training a combined model. Below we show the predictions of the binding curve using a LODE model.



(a) Predicted binding curve of the LODE model trained on all **100** exfoliation frames.

(b) Predicted binding curve of the LODE model trained on the first **40** exfoliation frames plus the largest-separation frame.

Figure 7: Training of the LODE model on the exfoliation data set.

We see from Fig.(7a) that the binding curve is perfectly learned using LODE. However, using a pure LODE model does not guarantee a good long-range model because it is not able to extrapolate (or more strictly speaking, interpolate within the range of the exfoliation) the long-range behavior once the corresponding training frames are removed, as shown in Fig.(7b). As a solution to this problem, we use a new feature matrix which is the combination of the SOAP features and the LODE features, where in this case we use radial spectrum invariants for the LODE part to capture the dispersion interaction, such that the short-range part and long-range part can be trained and learned at the same time. For a given atom i, its energy reads

$$\epsilon(i) = \sum_{\boldsymbol{q}_1} \epsilon_i^{SOAP}(\boldsymbol{q}_1) + \alpha \sum_{\boldsymbol{q}_2} \epsilon_i^{LODE}(\boldsymbol{q}_2), \tag{92}$$

where $\boldsymbol{q}_1$ denotes the SOAP vector, and $\boldsymbol{q}_2$ denotes the LODE vector. We temporarily refer to this one as *the SOAP + αLODE model*.

The training results using the SOAP + αLODE model are shown in Fig.(8). Here we are only using the first 40 frames in the data set, where the interlayer distances are smaller than 7.8Å, plus the last frame where the interlayer distance is 13.74Å as the reference point. Wee see that this model successfully extrapolates for frames with large interlayer distance. It should be stated that for the SOAP + αLODE model, the parameter α needs to be finely tuned if one wants to reach high accuracy.

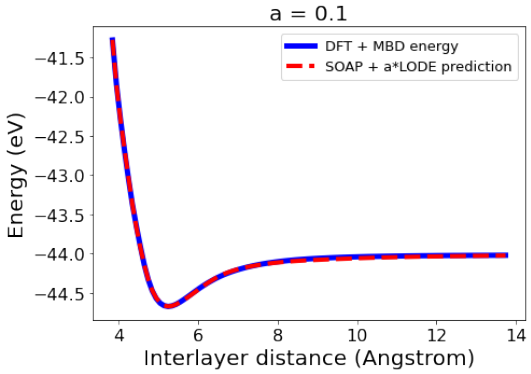


Figure 8: Predicted binding curve of the SOAP + αLODE model trained on the first **40** exfoliation frames plus the largest-separation frame.

5.2 The phosphorus allotropes data set

Due to the unparametric nature of the machine-learning method, the quality of the model depends highly on the training set. For an atomic potential that is transferrable, the data set should be as much diverse and representative as possible. The training set we use here is the same as the one in [5], where they combined the structures generated from two disjoint methods, manual construction and GAP+RSS (Gaussian approximation potential + random structure search), to increase the diversity and representativeness of the data set. The detailed composition of this data set is shown in Table.(1).

			Cells	**Atoms**	**Standard deviation (eV/atom)**
GAP-RSS	Initial (random)		199	1920	0.802550
	Intermediates		995	9320	0.268517
	Relaxed		596	5706	0.223722
	3-coordinated		400	7412	0.102984
Manually Constructed Structures	Liquid	Network	164	40672	0.111348
		Molecular (P4)	88	21824	0.009957
	2D	Ribbons	40	4216	0.058685
		Large sheets	87	11535	0.040544
		Exfoliation	1234	14172	0.129646
	Bulk crystals		959	24292	0.151315
	P2/P4 molecules		35	110	1.083762
	Free atom		1	1	0.0
Total			**4798**	**141180**	

Table 1: Detailed composition of the phosphorus allotropes data set

A visualized demonstration of this data set is given in the Appendix (section 8.2).

5.2.1 The 2-body+SOAP+R6 model

As the first trial, we train a model where the long-range part is captured by the manually extracted R6 potential as in section 5.1.1. We also add a 2-body model here in the training to increase the accuracy. The 2-body model is trained first, then the SOAP model is trained on the differences between the 2-body model predictions and the references. Due to the diversity of the training set, we apply different regularization values for different subsets to further increase the accuracy. The detailed training parameters can be found in section 8.1.6.

In Fig.(9) we show the parity plots of the training set of the 2-body + SOAP + R6 model. We separate the structures from different methods into two columns, and the energies and forces into two different rows. Structures from different subsets are marked in different colors.

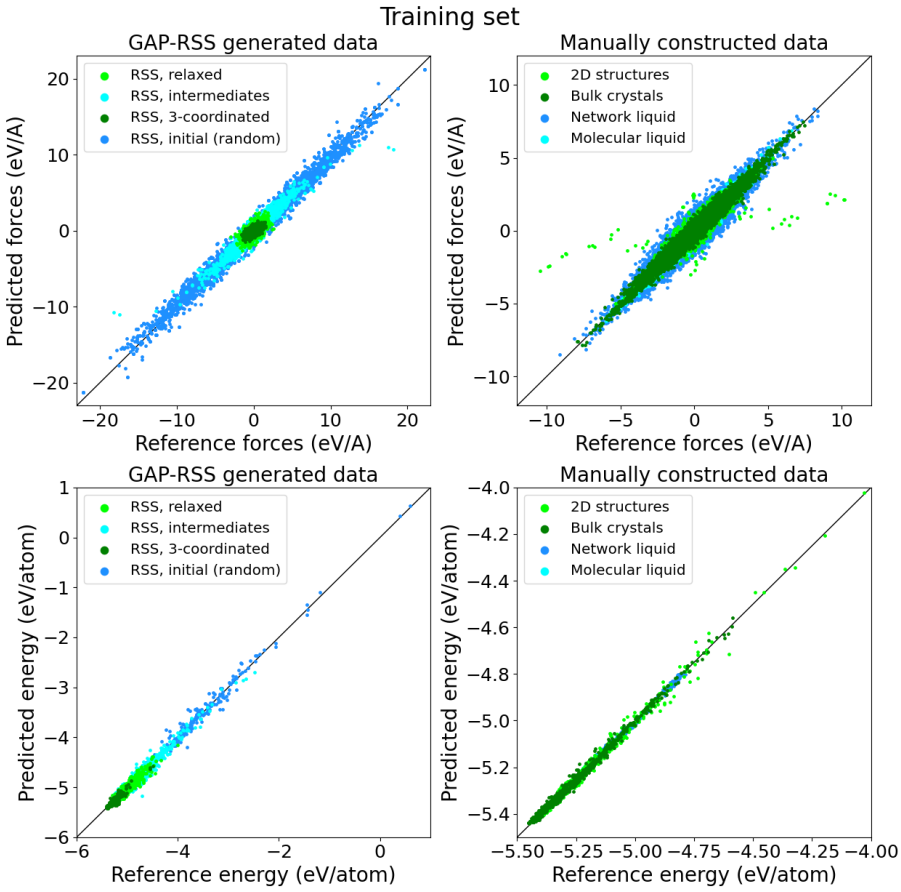


Figure 9: Parity plots of the training set of the 2-body + SOAP + R6 model.

In this model, we have achieved comparable RMSEs with respect to the published work [5]. A more visualized comparison is given in Fig.(10). Due to the different implementations of the calculation packages we use, such as the 2-body model,we do not expect the results to be exactly the same. In this plot we see the forces predictions are very close, yet there are some overfittings for the energies, such as P4 molecular liquid and random structures. The overfitting is not solved even if we increase the customized regularization values for them, which suggests this problem may come from the selection of the sparse points. Besides, we have tuned the customized regularization values to achieve better results. As can be seen from the plot, our model gives better predictions than the reference model on 2D structures.

What we care about the most about this model is its capability on capturing the dispersion interaction, which is examined using the exfoliation data set we introduced before. The predicted binding curve is shown in Fig.(11). For better illustration, we also include the same predictions of the 2-body+SOAP model, which is trained using exactly the same parameters without the R6 model. For the short-range part, these two models give similar predictions, but the contrast near the cutoff distance clearly shows that without the R6 model, a SOAP model is not able to learn the long-range interactions, as has been confirmed before.

5.2.2 The SOAP+αLODE model

In this section we will show the SOAP+αLODE model trained on the phosphorus allotropes data set and test its capability to correctly learn the dispersion interaction. Here we are only using part of the total phosphorus data set because the primitive version of pyLODE is computationally expensive. Calculating the LODE feature matrix without parallelization of the total data set could easily take days even with small training parameters. What's more, it will generate two matrices of dimension 140910×252 for the energy and $19577410 \times 3 \times 252$ for the forces (using `nmax = lmax = 6` for example), which will crash the regression model due to the memory limit when we do matrix multiplication. Therefore using a smaller yet

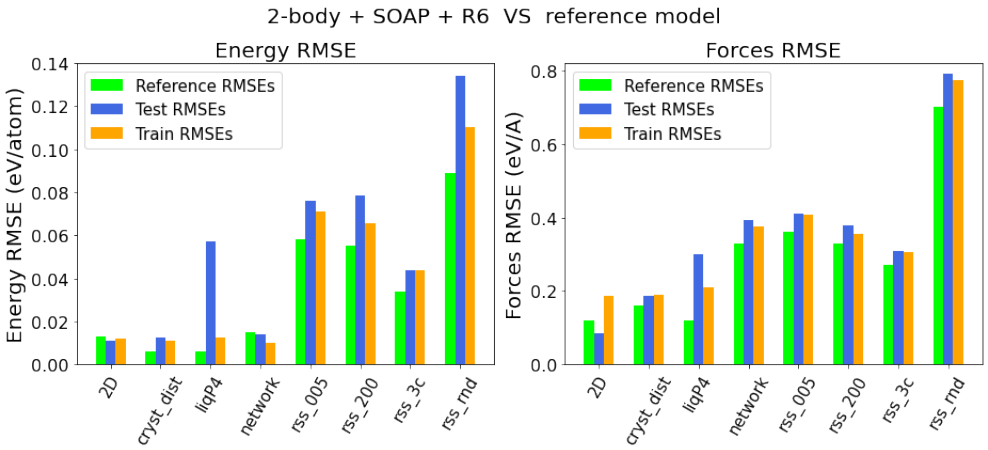


Figure 10: Histogram comparison of the RMSEs of our model and the reference model[5].

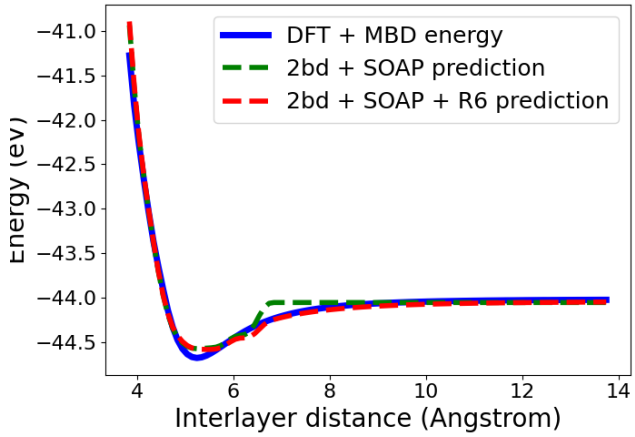


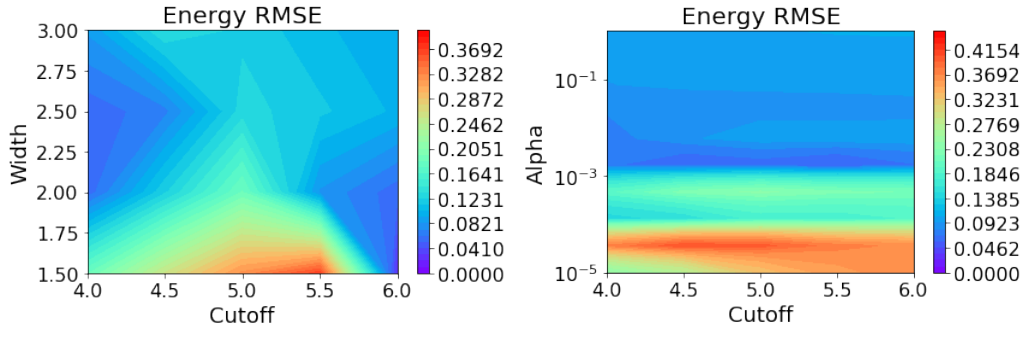
Figure 11: Predicted binding curves of the 2-body+SOAP+R6 model and the 2-body+SOAP model.

representative data set will not only save the computational time and memory, but also allow us to tune the training parameters quickly and efficiently. Most importantly, the key statement that LODE is able to capture the dispersion interaction in a general training set can still be demonstrated.

The training set we use here is obtained by taking every 10 frames from the phosphorus allotropes data set, such that various structures can be included. We further exclude the first 80 frames in the selected data set to speed up the training. A detailed composition of the training set used here can be found in the Appendix 8.1.7. The test set is the exfoliation data set we introduced before.

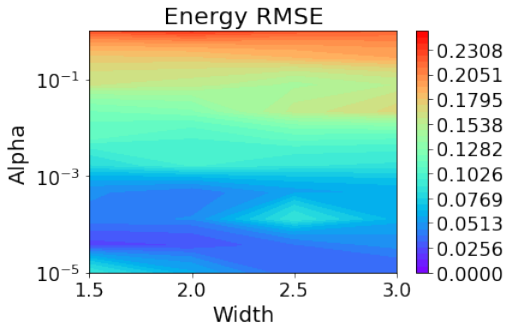
The tuning of the SOAP+αLODE model is very important, since improper parameters will completely mislead the model that the dispersion interaction can not be learned. In our training, the main three parameters we tuned are the cutoff radius, the cutoff width of the SOAP model and the value of α which determines the relative importance of LODE.

In Fig.(12) we show the change of energy RMSEs of the test set (the exfoliation data set) using different model parameters. We see from Fig.(12b) that the optimal value of α falls in a small region. As long as α is determined, the performance of the model does not rely too much on the cutoff radius or the cutoff width of the SOAP model, while the relative values of those two significantly affects the performance of the short-range part. It should be noted that the quality of the short-range model affects significantly whether LODE can learn the long-range interactions. From the previous experience, we suggest to firstly construct an accurate short-range model, then adjust the value of α to learn the long-range part.



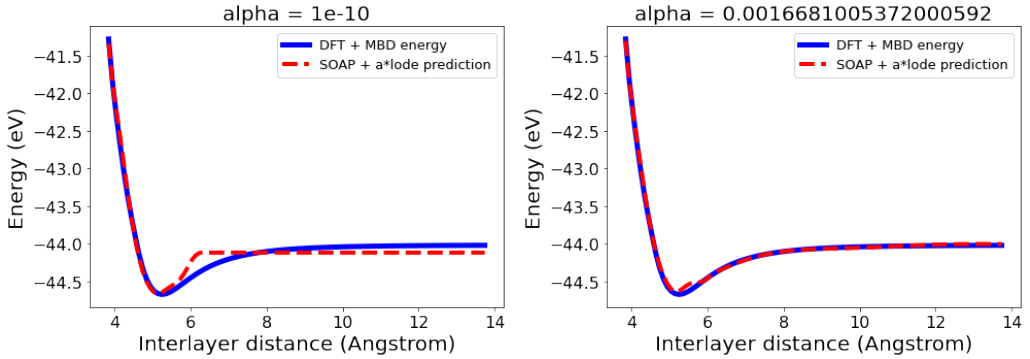
(a) Cutoff width vs cutoff radius

(b) Alpha vs cutoff radius



(c) Alpha vs cutoff width

Figure 12: Energy RMSE plots of the exfoliation frames for different model parameters



(a) the SOAP model

(b) the SOAP + αLODE model

Figure 13: Binding curve predictions of the SOAP model and the SOAP + αLODE model trained on part of the phosphorus data set.

The binding curve predictions of the SOAP + αLODE model trained on part of the phosphorus data set is shown in Fig.(13). In Fig.(13a), we show the predictions of a pure SOAP model. The short-range energy predictions of this model are very accurate, and the flat predictions outside the cutoff radius are expected. In Fig.(13b) we increase the contribution of the LODE feature matrix to learn the dispersion interaction. The clear contrast here shows that we successfully capture the dispersion interaction in the phosphorus data set by using a SOAP + αLODE model.

Another very important criteria of ML models is extrapolation capability. Here we are using a data set in which 68 black phosphorus frames are included. Even though there are only 7 frames whose interlayer distance is larger than 10Å, the model is still performing an interpolative task on the exfoliation frames. In our trials we have tried excluding all black phosphorus frames in the training set, however, this results in a deteriorated short-range model and thus affects the accuracy of the long-range part. Therefore, the test of the

extrapolation capability of LODE still remains an open question, and an extrapolative short-range model is crucial, which should be trained on a data set with acceptable size so that the LODE feature matrix will not crash the computer.

6 Summary

In this project, we have trained a general multi-scale SOAP + αLODE model to learn the dispersion interaction in phosphorus, and it is successfully examined on the exfoliation process of black phosphorus, where the dispersion interaction dominates the tail of the binding curve. We expect the SOAP + αLODE model to be capable of learning the long-range interactions in a general data set given the following three conditions:

- an accurate short-range model
- an optimal value of α
- a correct potential exponent in the LODE representation

As a comparison with the physics-based approach where the dispersion interaction is approximated using the fitted R6 potential, the SOAP + αLODE model has a higher computational cost, and for very large data set the training of the model is likely to crash due to the memory limit. On the other hand, R6 potential can only be extracted from systems where layered structures are available, so the SOAP + αLODE model is certainly more general and applicable to arbitrary physical systems with long-range interactions.

Besides the successful test in this project, we still notice the high computational resource requirement when we include LODE features. On the implementation side, my colleagues are working on a new implementation of LODE in *Rust* to speed up the running of codes. On the fundamental side, though the construction of LODE feature matrix can be reduced to several hours if we do parallelization, the mathematical manipulation of those huge matices are inevitable in the training process. In the future, it is worth trying to parallelize the training process as well, that the calculation of the optimal weights vector can be subdivided into small calculations and the final vector is obtained by stacking all of them. In that case, the training of a SOAP + αLODE model on thousands of frames is possible, so does the extrapolation test.

7 Acknowledgements

It has been a really wonderful year doing my specialization project and master project in COSMO, not only for the academic aspect that COSMO is one of the best groups in the world in the field of atomic modeling, but also for the great atmosphere among the students and our colleagues.

I want to first thank Prof. Michele Ceriotti for being the supervisor of my projects. He is the kind of the best professor I could imagine, who has a sharp mind and at the same time humorous, who is willing to spend time with students and offer helpful advice. I have learned a lot from him and appreciate the time he spent for keeping my projects on the right track. I also want to thank Prof. Giuseppe Carleo for being my co-supervisor. Kevin Kazuki Huguenin-Dumittan is whom I spent the most time with, and he is the best TA and mentor I have ever met. I really appreciate his detailed explanations of the questions I asked, and will miss the time we spent together discussing where the story is going in One-piece. By the way, the Rengoku figure he brought from Japan is really nice! I also want to thank Dr. Guillaume Fraux for helping me with the coding and the implementations of the utility functions. I am super impressed by how fast he can solve a problem and finish his coding. Last but not least, I want to thank Jigyasa Nigam for helping me with the theory part, thank Dr. Philip Loche for the suggestions about pyLODE, thank Dr. Max Veit for giving advice for my first group meeting presentation, and thank Alex for solving my account problems when using the clusters.

The academic year 2021-2022 was not ordinary. Besides the fact that this is the last year of my master degree, we also witnessed that the world has changed a lot. I have thought a lot about my position in the world, and how I can be a better world citizen. It has been a year of growth of my inside world. Besides, I hope I would have the opportunity to reunite with my family in the next Chinese new year, seeing their faces gives me the greatest motivation. I will head for the next journey in my life soon, wish me and wish all of my friends the best luck!

8 Appendix

8.1 Summary of training parameters and plots

8.1.1 SOAP model trained on all 100 exfoliation frames

training set	invariants type	model type	E regularizer	F regularizer
all 100 exfoliation frames	power-spectrum	linear model do_normalize = False	1.25e-4	1.25e-4

The hyper parameters are

```
hypers_rascaline = {
    "cutoff": 5.,
    "atomic_gaussian_width": 0.3,
    "max_radial": 10,
    "max_angular": 10,
    "radial_basis": {"Gto": {}},
    "cutoff_function": {"ShiftedCosine": {"width": 1.}},
    "gradients": True
}
```

The energy RMSE of this model is **0.00759165** eV/atom.

The forces RMSE of this model is **0.008069327104658661** eV/Å.

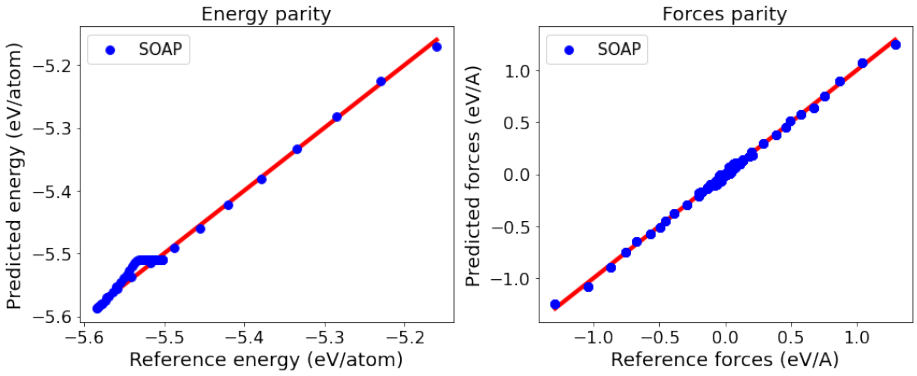


Figure 14: Parity plots of the SOAP model trained on all 100 exfoliation frames

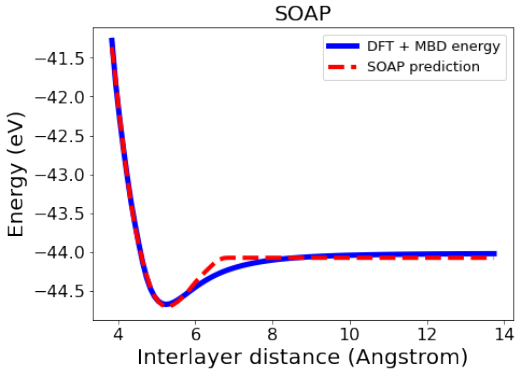


Figure 15: Predicted binding curve of the SOAP model trained on all 100 exfoliation frames

8.1.2 SOAP+R6 model trained on all 100 exfoliation frames

training set	invariants type	model type	E regularizer	F regularizer
all 100 exfoliation frames	power-spectrum	linear model do_normalize = False	1.25e-4	1.25e-4

The hyper parameters are

```
hypers_rascaline = {
    "cutoff": 5.,
    "atomic_gaussian_width": 0.3,
    "max_radial": 10,
    "max_angular": 10,
    "radial_basis": {"Gto": {}},
    "cutoff_function": {"ShiftedCosine": {"width": 1.}},
    "gradients": True
}
```

The energy RMSE of this model is **0.00131** eV/atom.

The forces RMSE of this model is **0.006396407970904036** eV/Å.

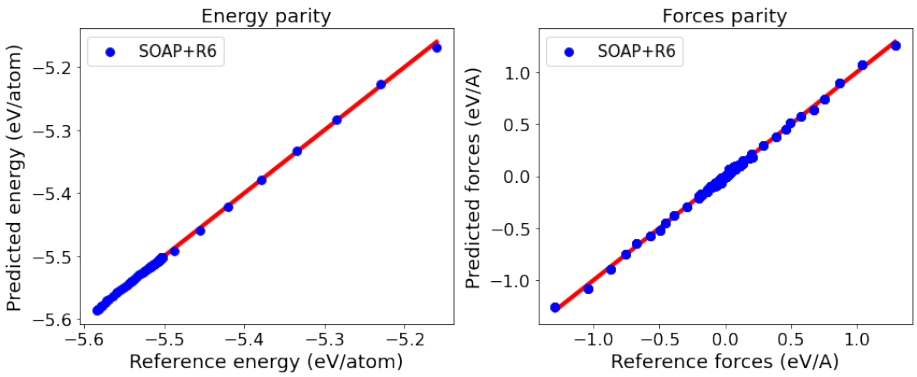


Figure 16: Parity plots of the SOAP+R6 model trained on all 100 exfoliation frames

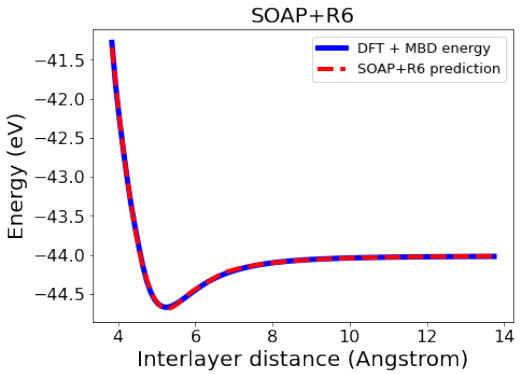


Figure 17: Predicted binding curve of the SOAP+R6 model trained on all 100 exfoliation frames

8.1.3 LODE model trained on all 100 exfoliation frames

training set	invariants type	model type	E regularizer	F regularizer
all 100 exfoliation frames	power-spectrum	linear model do_normalize = False	1.25e-4	1.25e-4

The hyper parameters are

```
hypers_lode = {
    'smearing':0.7,
    'max_angular':6,
    'max_radial':6,
    'cutoff_radius':5.,
    'potential_exponent':1,
    'radial_basis': 'gto',
    'compute_gradients':True
}
```

The energy RMSE of this model is **0.0013711** eV/atom.

The forces RMSE of this model is **0.00543551537509253** eV/Å.

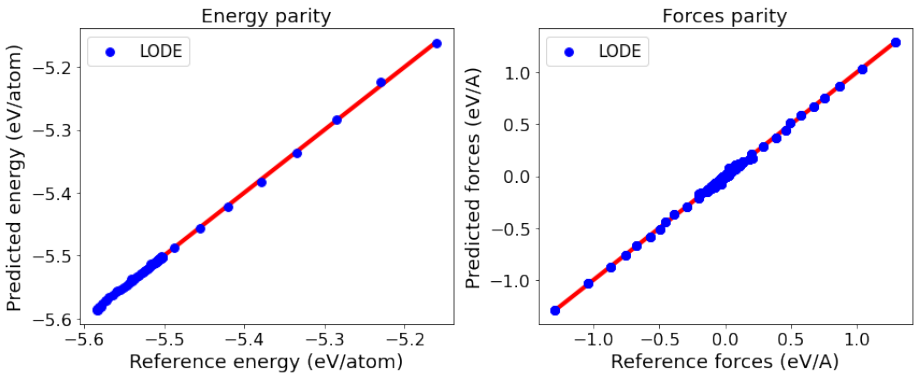


Figure 18: Parity plots of the LODE model trained on all 100 exfoliation frames

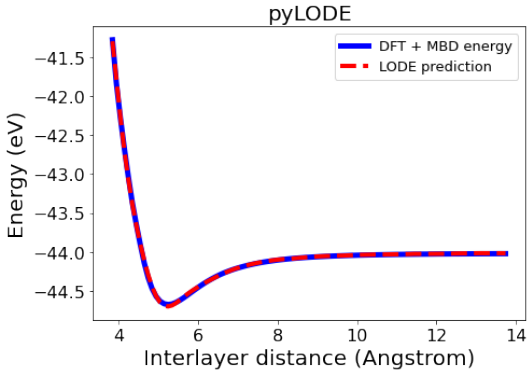


Figure 19: Predicted binding curve of the LODE model trained on all 100 exfoliation frames

8.1.4 LODE model trained on the first 40 + the last exfoliation frames

training set	invariants type	model type	E regularizer	F regularizer
the first 40 exfoliation frames + the largest-separation frame	power-spectrum	linear model do_normalize = False	1.25e-4	1.25e-4

The hyper parameters are

```
hypers_lode = {
    'smearing':0.7,
    'max_angular':6,
    'max_radial':6,
    'cutoff_radius':5.,
    'potential_exponent':1,
    'radial_basis': 'gto',
    'compute_gradients':True
}
```

The energy RMSE of this model is **9.85302012** eV/atom.

The forces RMSE of this model is **8.55970805560699** eV/Å.

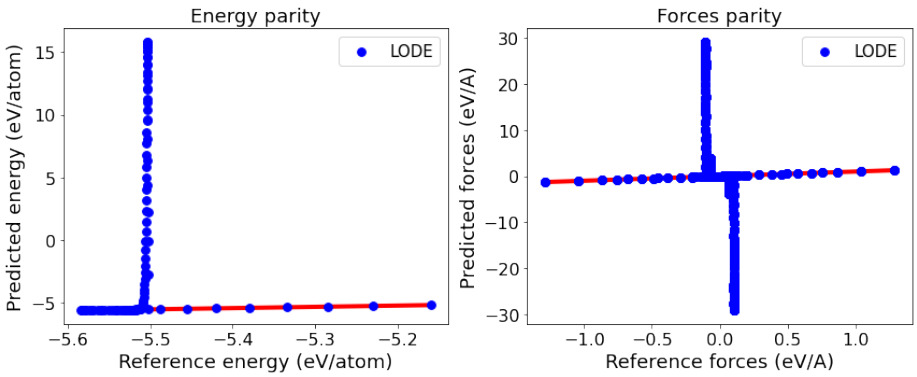


Figure 20: Parity plots of the LODE model trained on the first 40 + the last exfoliation frames

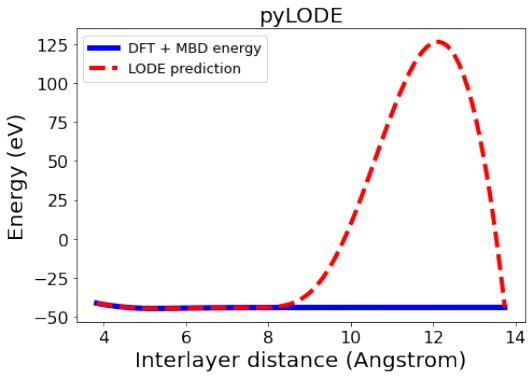


Figure 21: Predicted binding curve of the LODE model trained on the first 40 + the last exfoliation frames

8.1.5 SOAP+αLODE model trained on the first 40 + the last exfoliation frames

training set	invariants type	model type	E regularizer	F regularizer
the first 40 exfoliation frames + the largest-separation frame	SOAP: power-spectrum LODE: radial-spectrum	linear model do_normalize = False	1.25e-4	1.25e-4

The hyper parameters are

```
hypers_lode = {
    'smearing':0.7,
    'max_angular':0,
    'max_radial':10,
    'cutoff_radius':5.,
    'potential_exponent':6,
    'radial_basis': 'gto',
    'compute_gradients':True
}
hypers_rascaline = {
    "cutoff": 4.5,
    "atomic_gaussian_width": 0.7,
    "max_radial": 6,
    "max_angular": 6,
    "radial_basis": {"Gto": {}},
    "cutoff_function": {"ShiftedCosine": {"width": 1.}},
    "gradients": True
}
```

The energy RMSE of this model is **0.00143552** eV/atom.

The forces RMSE of this model is **0.004444833876084638** eV/Å.

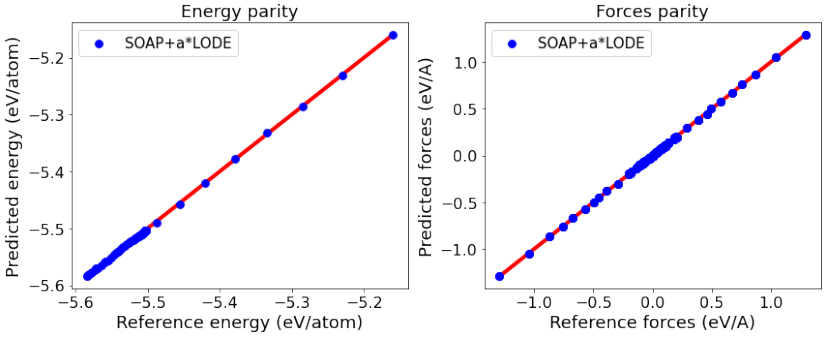


Figure 22: Parity plots of the SOAP + αLODE model trained on the first 40 + the last exfoliation frames

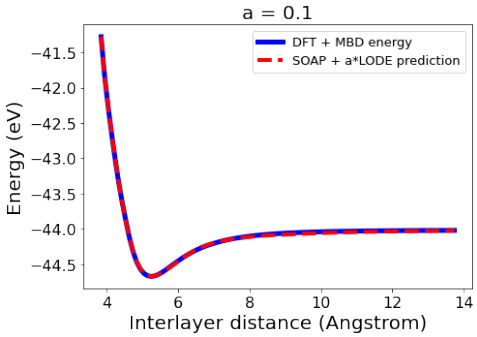


Figure 23: Predicted binding curve of the SOAP + αLODE model trained on the first 40 + the last exfoliation frames

8.1.6 2bd+SOAP+R6 model trained on the total data set

Here are the training parameters of this model.

	2-body model	SOAP model
training set	all 4798 phosphorus frames	all 4798 phosphorus frames
cutoff	5.	5.
nmax	12	12
lmax	1	6
normalization	False	True
number of sparse points	15	8000
zeta	1	4
energy regularization	0.003	customized
forces regularization	0.003	customized
gaussian constant	0.5	0.5
cutoff width	1.0	1.0

The customized regularization values for each subset are shown in Table.(2).

	Energy regularization	Forces regularization
P2/P4	0.03	0.4
rss_200	0.035	0.4
rss_005	0.035	0.4
2D	0.01	0.07
rss_3c	0.025	0.4
cryst_dist	0.03	0.3
liq_12_03_02_network	0.03	0.4
rss_rnd	0.05	0.35
liq_12_03_01_liqP4	0.4	0.5
phosphorene	0.03	0.4
phosphorus_ribbons	0.03	0.4
isolated_atom	0.04	0.6

Table 2: Customized regularization values for different subsets

The parity plots of the training set have been given in Fig.(9). Here we show the parity plots of the test set (ref. [5]) from the same model. In general, the predictions agree very well with the reference values, but we see a noticeable mismatch for the energy parity for a subset labeled with 'hp', which stands for high-pressure structures. This comes from the fact that high-pressure structures are not included in the training set, therefore resulting in larger errors.

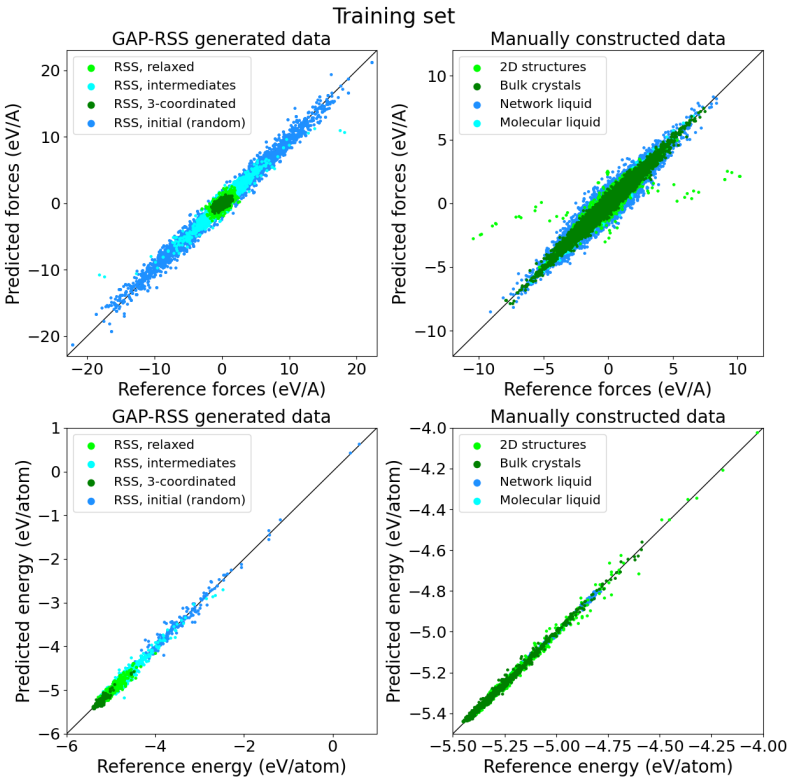


Figure 24: Parity plots of the training set of the 2-body + SOAP + R6 model.

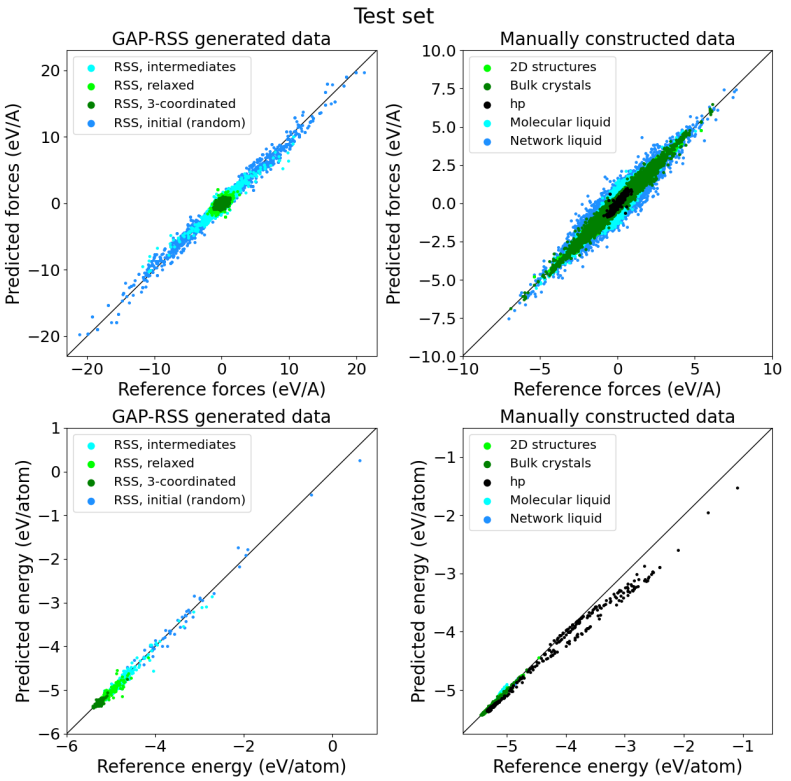


Figure 25: Parity plots of the test set of the 2-body + SOAP + R6 model.

8.1.7 SOAP model trained on 1/10 of the total data set

The composition of the training set is shown in Table (3).

2D	cryst_dist	rss_200	rss_3c	rss_005	rss_rnd	total
120	66	60	32	104	18	400

Table 3: Composition of one tenth of the phosphorus data set

The training parameters are:

training set	invariants type	model type	E regularizer	F regularizer
shown in Table (3)	power spectrum	linear model do_normalize = False	1.25e-4	1.25e-2

The hyper parameters are:

```
hypers_rascaline = {
    "cutoff": 4.5,
    "atomic_gaussian_width": 0.2,
    "max_radial": 10,
    "max_angular": 10,
    "radial_basis": {"Gto": {}},
    "cutoff_function": {"ShiftedCosine": {"width": 1.5}},
    "gradients": True
}
```

The energy RMSE of the training set is **0.05484343** eV/atom.

The forces RMSE of the training set is **0.48150174328302936** eV/Å.

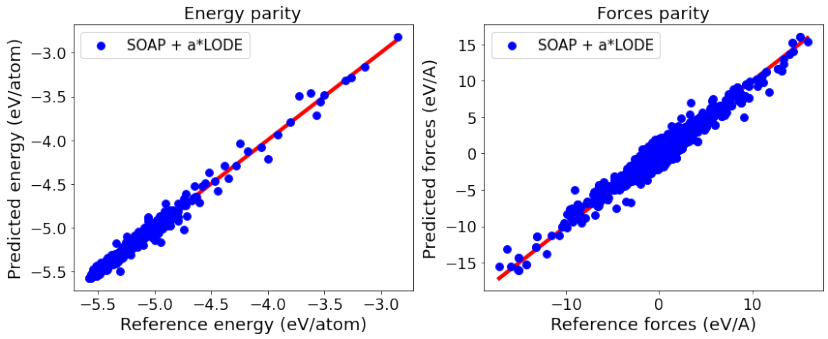


Figure 26: Parity plots of the SOAP model trained on 1/10 of the phosphorus data set

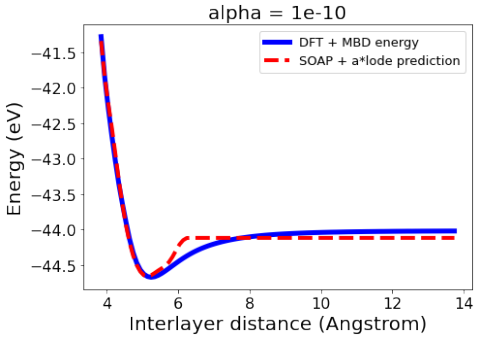


Figure 27: Predicted binding curve of the SOAP model trained on 1/10 of the phosphorus data set

8.1.8 SOAP + αLODE model trained on 1/10 of the total data set

training set	invariants type	model type	E regularizer	F regularizer
shown in Table (3)	SOAP: power spectrum LODE: radial spectrum	linear model do_normalize = False	1.25e-4	1.25e-2

The hyper parameters are:

```
hypers_lode = {
    'smearing':1.4,
    'max_angular':0,
    'max_radial':10,
    'cutoff_radius':5.,
    'potential_exponent':6,
    'radial_basis': 'gto',
    'compute_gradients':True
}
hypers_rascaline = {
    "cutoff": 4.5,
    "atomic_gaussian_width": 0.2,
    "max_radial": 10,
    "max_angular": 10,
    "radial_basis": {"Gto": {}},
    "cutoff_function": {"ShiftedCosine": {"width": 1.5}},
    "gradients": True
}
```

The energy RMSE of the training set is **0.05419398** eV/atom.

The forces RMSE of the training set is **0.482292372706692** eV/Å.

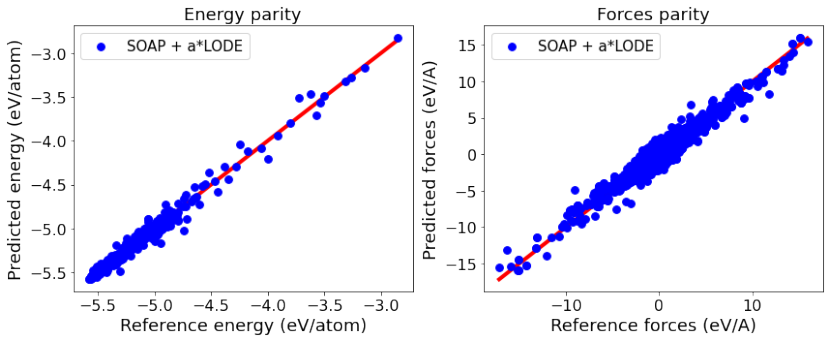


Figure 28: Parity plots of the SOAP + αLODE model trained on 1/10 of the phosphorus data set

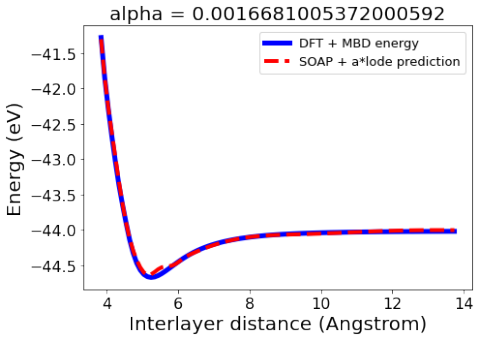
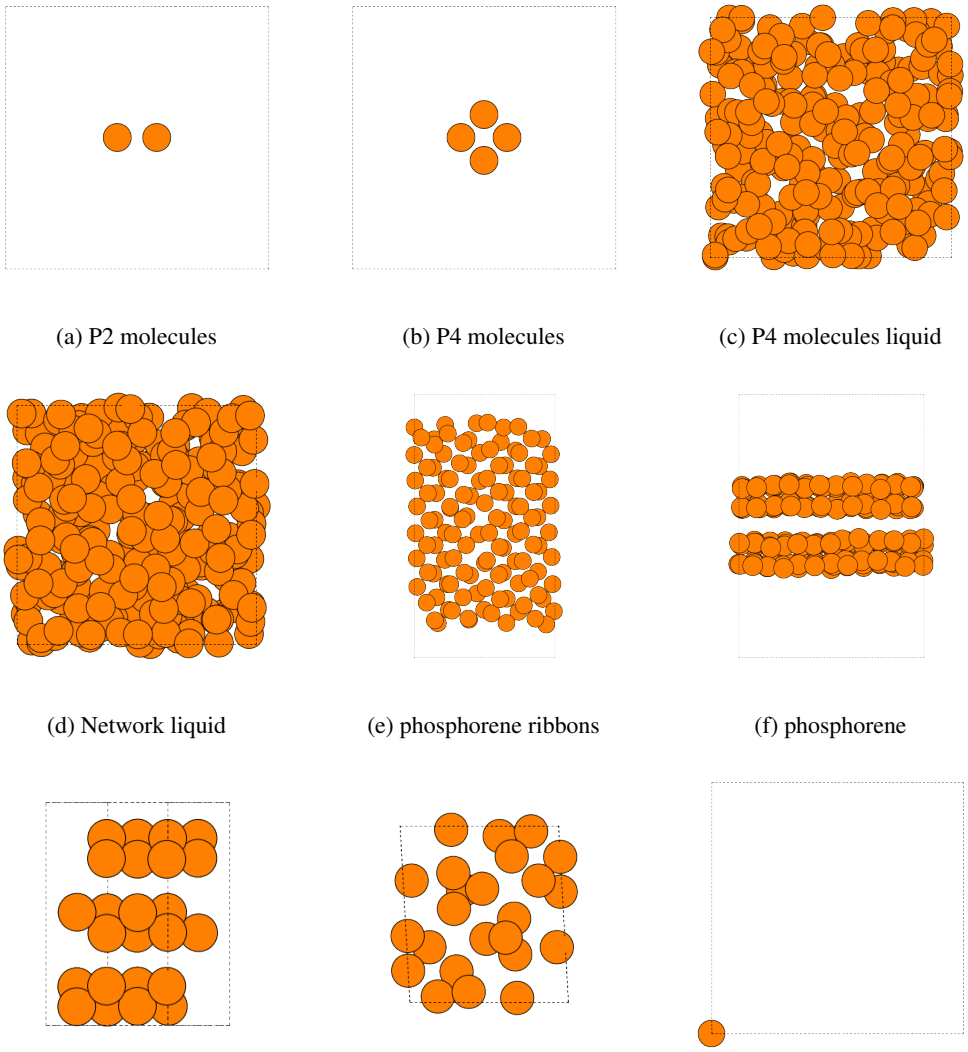


Figure 29: Predicted binding curve of the SOAP + αLODE model trained on 1/10 of the phosphorus data set

8.2 Visualized representative frames from the phosphorus allotropes data set

In this section we select representative frames from each subset in the phosphorus allotropes data set. Frames from GAP-RSS and manually constructed structures are shown separately.

Manually constructed frames:



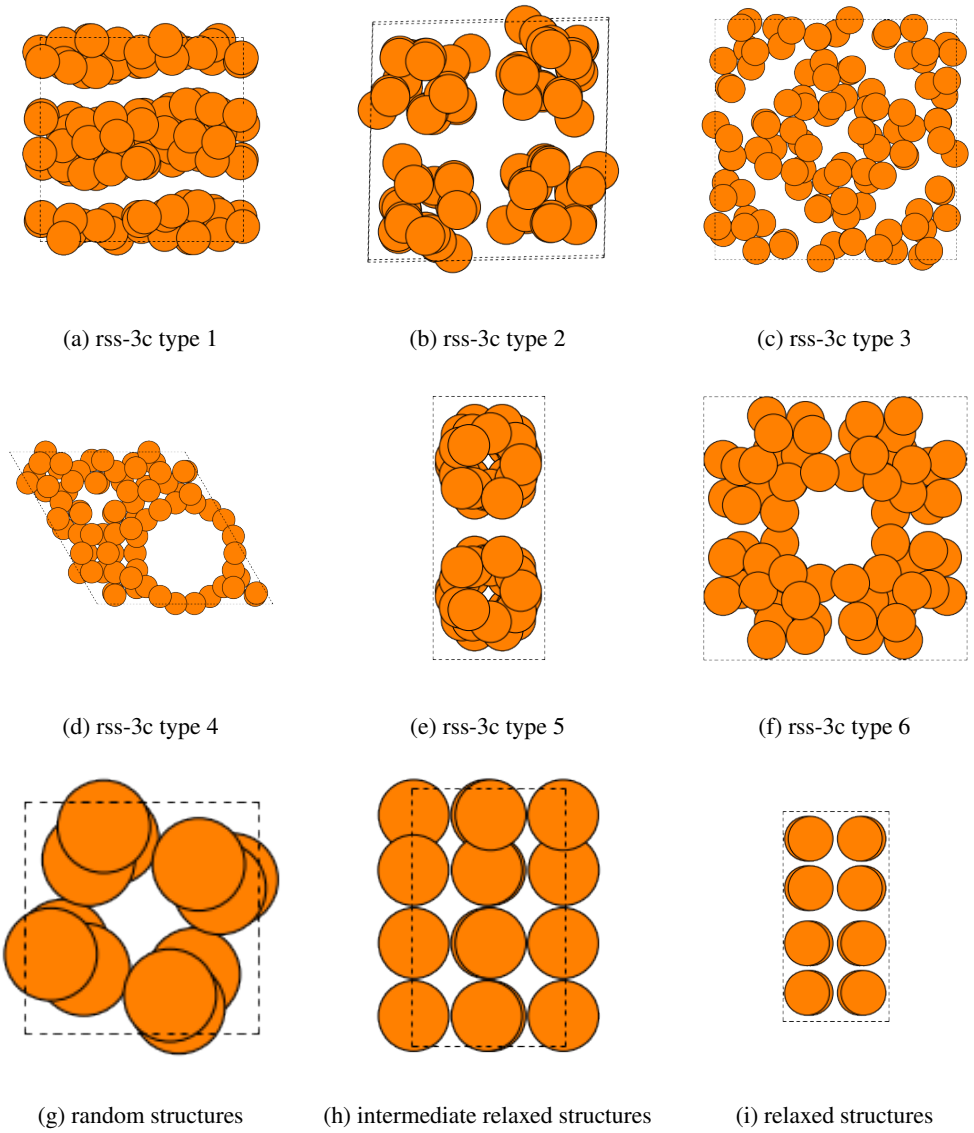
(a) P2 molecules (b) P4 molecules (c) P4 molecules liquid

(d) Network liquid (e) phosphorene ribbons (f) phosphorene

(g) 2D structures (h) crystal structures (i) isolated atom

Figure 30: Representative **manually constructed structures** from the phosphorus allotropes data set.

GAP-RSS frames:



(a) rss-3c type 1 (b) rss-3c type 2 (c) rss-3c type 3

(d) rss-3c type 4 (e) rss-3c type 5 (f) rss-3c type 6

(g) random structures (h) intermediate relaxed structures (i) relaxed structures

Figure 31: Representative **GAP-RSS structures** from the phosphorus allotropes data set.

References

[1] Patrick Rowe et al. "An accurate and transferable machine learning potential for carbon". In: *The Journal of Chemical Physics* 153.3 (2020), p. 034702.

[2] Heikki Muhli et al. "Machine learning force fields based on local parametrization of dispersion interactions: Application to the phase diagram of C 60". In: *Physical Review B* 104.5 (2021), p. 054106.

[3] Andrea Grisafi and Michele Ceriotti. "Incorporating long-range physics in atomic-scale machine learning". In: *The Journal of chemical physics* 151.20 (2019), p. 204105.

[4] Albert P Bartók, Risi Kondor and Gábor Csányi. "On representing chemical environments". In: *Physical Review B* 87.18 (2013), p. 184115.

[5] Volker L Deringer, Miguel A Caro and Gábor Csányi. "A general-purpose machine-learning force field for bulk and nanostructured phosphorus". In: *Nature communications* 11.1 (2020), pp. 1–11.

[6] Stefan Grimme et al. "A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu". In: *The Journal of chemical physics* 132.15 (2010), p. 154104.

[7] Mohammad K Nazeeruddin et al. "Combined experimental and DFT-TDDFT computational study of photoelectrochemical cell ruthenium sensitizers". In: *Journal of the American Chemical Society* 127.48 (2005), pp. 16835–16847.

[8] IB Obot, DD Macdonald and ZM Gasem. "Density functional theory (DFT) as a powerful tool for designing new organic corrosion inhibitors. Part 1: an overview". In: *Corrosion Science* 99 (2015), pp. 1–30.

[9] Norbert Schuch and Frank Verstraete. "Computational complexity of interacting electrons and fundamental limitations of density functional theory". In: *Nature physics* 5.10 (2009), pp. 732–735.

[10] MA González. "Force fields and molecular dynamics simulations". In: *École thématique de la Société Française de la Neutronique* 12 (2011), pp. 169–200.

[11] Roland Schulz et al. "Scaling of multimillion-atom biological molecular dynamics simulation on a petascale supercomputer". In: *Journal of Chemical Theory and Computation* 5.10 (2009), pp. 2798–2808.

[12] John L Klepeis et al. "Long-timescale molecular dynamics simulations of protein structure and function". In: *Current opinion in structural biology* 19.2 (2009), pp. 120–127.

[13] Matthias Rupp et al. "Fast and accurate modeling of molecular atomization energies with machine learning". In: *Physical review letters* 108.5 (2012), p. 058301.

[14] Stefan Chmiela et al. "Towards exact molecular dynamics simulations with machine-learned force fields". In: *Nature communications* 9.1 (2018), pp. 1–10.

[15] Justin S Smith, Olexandr Isayev and Adrian E Roitberg. "ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost". In: *Chemical science* 8.4 (2017), pp. 3192–3203.

[16] Michael J Willatt, Félix Musil and Michele Ceriotti. "Atom-density representations for machine learning". In: *The Journal of chemical physics* 150.15 (2019), p. 154110.

[17] Felix Musil et al. "Physics-inspired structural representations for molecules and materials". In: *Chemical Reviews* 121.16 (2021), pp. 9759–9815.

[18] Leopoldo Nachbin. *The haar integral*. RE Krieger Publishing Company, 1976.

[19] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[20] Volker L Deringer et al. "Gaussian process regression for materials and molecules". In: *Chemical Reviews* 121.16 (2021), pp. 10073–10141.

[21] Gábor Csányi. https://github.com/libAtoms/testing-framework/blob/public/tests/P/black_exfoliation/exfoliation_mbd_reference.xyz. 2020.