

# FIRST LINE OF TITLE

## SECOND LINE OF TITLE

THIS IS A TEMPORARY TITLE PAGE  
It will be replaced for the final print by a version  
provided by the registrar's office.

Thèse n. 1234 2023  
présentée le November 1, 2023  
à la Faculté des sciences de base  
laboratoire SuperScience  
programme doctoral en SuperScience  
École polytechnique fédérale de Lausanne  
pour l'obtention du grade de Docteur ès Sciences  
par

Paolino Paperino

acceptée sur proposition du jury:

Prof Name Surname, président du jury  
Prof Name Surname, directeur de thèse  
Prof Name Surname, rapporteur  
Prof Name Surname, rapporteur  
Prof Name Surname, rapporteur

Lausanne, EPFL, 2023

**EPFL**



Wings are a constraint that makes  
it possible to fly.  
— Robert Bringhurst

To my parents...



# Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

*Lausanne, November 1, 2023*

D. K.



# Abstract

In high-throughput material design, large databases of materials are searched for candidates with desirable characteristics. So far, searches based on experimental data have been limited in scope, due to the vast combinatorial space of materials, the heterogeneous quality of available data, and the difficulty in separating the intrinsic properties of a material from those that are contingent on the processing or the synthesis conditions. A viable alternative is to calculate material properties using computer simulations, that make it possible to exploit advances in parallel computing to construct databases with millions of entries, and to obtain results that are internally consistent. The quantitative accuracy of these predictions, however, is dependent on the quality of the reference electronic structure calculations, increasing the computational effort and reducing the breadth of the searches. Data-driven approaches have been applied to reduce the cost of accurate computational studies, by using only a small number of reference calculations for a representative subset of materials space, and using them to train surrogate models that predict inexpensively the outcome of such calculation on new materials. The way materials structures are processed into a numerical description as input of machine learning algorithms is crucial to obtain efficient and computationally inexpensive models. Recent advancements in the design of information-efficient representations have embedded novel types of information, such as neighborhood environments or pair descriptions. Despite the rapid development in offloading calculations to more dedicated hardware, these enhancements nevertheless substantially increase the cost of the representation that remains a crucial factor in simulations. It is therefore vital to delve deeper into the design space of representations to understand the type of information they encapsulate. Insights from such analyses aid in making more informed decisions regarding the trade-off between accuracy and performance. While a substantial amount of work has been undertaken to compare representations concerning their structure-property relationship, a thorough exploration into understanding the inherent nature of the information capacity of these representations remains mostly uncharted. This thesis introduces a set of measures that facilitate quantitative analysis concerning the relationship between features and datasets, thereby assisting in such decision-making processes and providing valuable insights to the academic community. Additionally, a considerable amount of effort has been dedicated to optimize the basis set involved in all representations, typically driven by heuristic considerations on the behavior of the regression target. This thesis showcases a scheme that utilizes splines to approximate the basis expansion coefficients, paving the way for expansive optimization methods to create more effective basis functions at no additional cost during simulation time. This is pivotal in

## **Abstract**

---

simulations targeting materials encompassing a high variety of chemical species or relying on qualitative collective variables.

# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Abstract (English)</b>	<b>iii</b>
<b>1 Theory of atomistic representations</b>	<b>1</b>
1.1 Atomic-centered density . . . . .	1
1.2 Hierarchy of symmetrized representations . . . . .	3
1.2.1 Ordered basis set . . . . .	3
1.2.2 Fixed basis set . . . . .	4
<b>2 Measures of information capacity</b>	<b>9</b>
2.1 Reconstruction error . . . . .	9
2.2 Formulation as optimization problem . . . . .	12
2.3 Linear decodable information . . . . .	14
2.3.1 Efficient cross-validation ridge for high-dimensional feature space . . .	15
2.3.2 Example: Wasserstein distance . . . . .	16
2.3.3 Example: Comparison of bispectrum and message-passing . . . . .	19
2.4 Distortion in linear transformations . . . . .	19
2.4.1 Example: Radial scaling . . . . .	20
2.5 Nonlinearity as local linearity . . . . .	22
2.5.1 Local linear embedding . . . . .	22
2.5.2 Jacobian . . . . .	23
2.5.3 Example: Message-passing and connectivity . . . . .	27
2.5.4 Local stiffness [optional] . . . . .	29
2.5.5 Future directions - Restricting nonlinearities . . . . .	31
<b>3 Splining</b>	<b>33</b>
3.1 Grid . . . . .	33
3.1.1 Adaptive grid . . . . .	33
3.1.2 Equispaced grid . . . . .	33
3.2 Spline function . . . . .	34
3.2.1 Cubic spline . . . . .	34
<b>4 Symmetry-preserving basis optimization</b>	<b>39</b>

## Contents

---

4.1	Closed-form solutions . . . . .	39
4.1.1	Unsupervised optimization . . . . .	41
4.1.2	Supervised optimization . . . . .	42
4.2	Higher-order information . . . . .	43
4.3	Future directions . . . . .	43
4.3.1	Hierarchical optimization . . . . .	43
4.3.2	Radial dependent smoothness . . . . .	44
<b>5</b>	<b>Implementation of machine learning interatomic potentials</b>	<b>45</b>
5.1	Implementation of gradients in kernel models . . . . .	46
5.2	Interfacing with molecular dynamics packages . . . . .	47
5.3	A metadynamic software framework with LAMMPS, PLUMED and i-PI . . . . .	48
5.4	Serialization of MLIP . . . . .	50
<b>Bibliography</b>		<b>53</b>
<b>Bibliography</b>		<b>60</b>

# 1 Theory of atomistic representations

This chapter covers the theory and the computation of symmetrized features on the atomic-scale. A similar approach as in Ref. [1] is taken into the theory of atomistic representations utilizing concepts that exists in representation theory. We therefore first introduce different functional forms  $f : V \rightarrow \mathbb{R}$  representing the atomic structure  $A$ . On these functional forms a set of functions  $\{b_k : V \rightarrow \mathbb{R}\}_{k=1}^M$ , the *basis set*, is expanded

$$\int_V d\mathbf{q} f(\mathbf{q}) b_k(\mathbf{q}) = c_k \quad (1.1)$$

to obtain a set of *expansion coefficients*  $\{c_k \in \mathbb{R}\}_{k=1}^M$  that can be used to build a model mapping the coefficients to physical properties like energy and forces. The choice of the functional form as well as the basis set is essential for an effective description, i.e. a description that captures information with the least amount of number coefficients. Prior knowledge about physical properties and distributions of atomic structures is used to bias the construction to more effective descriptions in the field. A family of representations based on the higher orders of the *atomic density* function is introduced and it is shown how invariances are embedded efficiently. Furthermore, general characteristics of basis functions used in the field of atomic-scale modelling and how to efficiently embed invariances into these are presented.

## 1.1 Atomic-centered density

A majority of developed atomistic descriptors can be seen as different approaches in describing the same function representing the atomic structure in different ways, the atomic density[1]

$$\rho(\mathbf{r}) = \sum_{i \in A} \rho(\mathbf{r} - \mathbf{r}_i), \quad \rho : \mathbb{R}^3 \rightarrow \mathbb{R} \quad (1.2)$$

where  $\mathbf{r}_i \in \mathbb{R}^3$  is the position of the  $i$ th atom in the atomic structure  $A$  and  $\rho$  is an arbitrary function decaying from its origin. Commonly, the Gaussian  $g$  or Dirac  $\delta$  function are used for the density  $\rho$ [4]. A widely adapted approach to impose translational invariance, i.e. inde-

## Chapter 1. Theory of atomistic representations

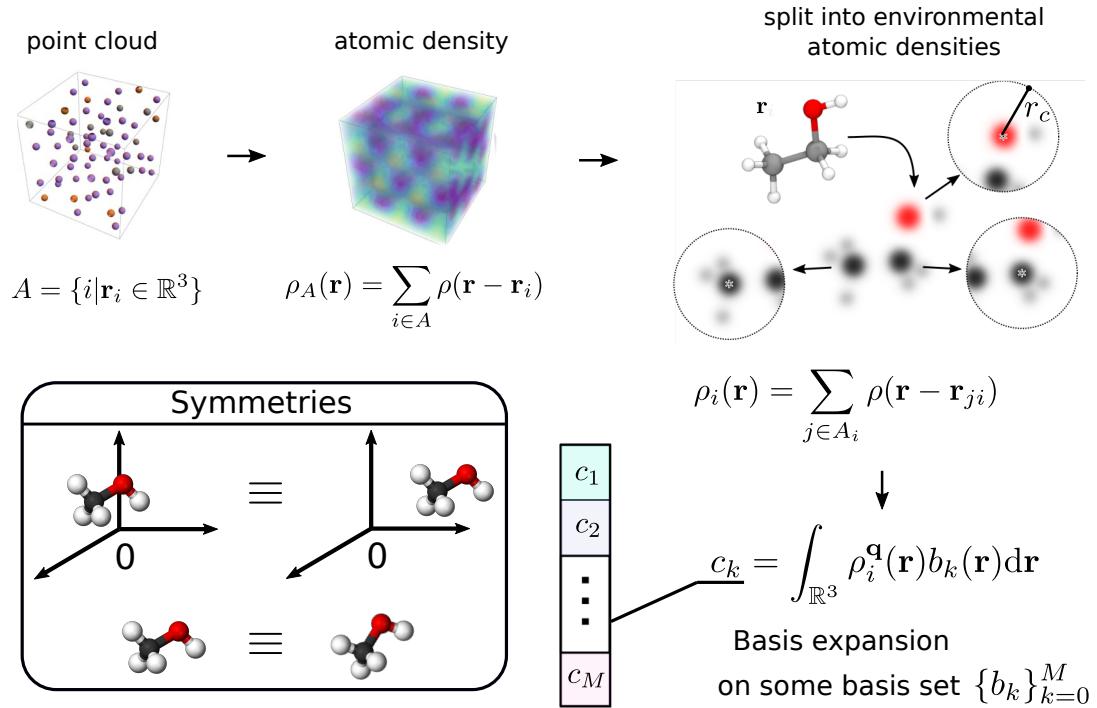


Figure 1.1: A schematic showing the construction of features as input for a statistical inference model based on the atomic-centered density correlations function. The figure of the atoms in the box are retrieved from Ref. [2]. The figure of the atomic environments is retrieved from Ref. [1]. The methanol molecule is retrieved from Ref. [3].

## 1.2 Hierarchy of symmetrized representations

---

pendence of the center of the structure, is to describe the atomic structure as a sum of local atomic environments

$$\sum_{i \in A} \rho_i(\mathbf{r}) = \sum_{i \in A} \sum_{j \in A_i} \rho(\mathbf{r} - \mathbf{r}_{ji}), \quad (1.3)$$

where  $A_i$  is the set of all atoms that are in the *environment* of atom  $i$ , in most applications defined as the set of atoms within a certain distance, the *cutoff*. Further,  $\mathbf{r}_{ji}$  is the direction vector  $\mathbf{r}_j - \mathbf{r}_i$ . We call  $\rho_i$  *neighbor density*. This approach aligns with the partition of the target property  $y$  into atomic contributions

$$\sum_{i \in A} y_i = y. \quad (1.4)$$

This decomposition into local properties is motivated by the heuristical observation that atomic properties decay with their distance to the center, coined *nearsightedness*[5].

### 1.2 Hierarchy of symmetrized representations

As a large family of physical quantities are invariant under rotations of the atomic structure, it is needed to account for rotational invariance in the process of relating atomic structure to such quantities. Rotational invariance can be embedded into the functional form by integrating over the rotation group  $SO(3)$

$$f^1(\mathbf{r}) = \int_{SO(3)} d\hat{R} \rho_i(\hat{R}\mathbf{r}) \quad (1.5)$$

and can be further extended to higher orders of the density

$$f^v(\mathbf{r}, \dots, \mathbf{r}_v) = \int_{SO(3)} d\hat{R} \rho_i(\hat{R}\mathbf{r}_1) \rho_i(\hat{R}\mathbf{r}_2) \dots \rho_i(\hat{R}\mathbf{r}_v) \quad (1.6)$$

named *atom-centered density correlation* function[6]. While in principle this integral can be numerically evaluated, such an integration is not suitable to efficiently evaluate the expansion coefficients. Hence, it is essential to chose suitable candidates for the density  $\rho$  and the basis set  $\{b_k\}_{k=1}^M$  that yield an analytical solution for the integral. Note that post-symmetrization methods have recently been developed that also allow an efficient evaluation of the integral by averaging over all possible reference frames within a order  $v$  tuple[7], these are however not in the scope of the thesis.

#### 1.2.1 Ordered basis set

An explicit solution of the integral over  $SO(3)$  can be expressed for the Dirac  $\delta$  densities. Here we present solutions for order 1 and 2 that should make the idea clear

$$\int_{SO(3)} d\hat{R} \delta(\hat{R}\mathbf{r} - \mathbf{r}_{ji}) \propto_r \delta(r - r_{ji}), \quad (1.7a)$$

$$\int_{SO(3)} d\hat{R} \delta(\hat{R}\mathbf{r} - \mathbf{r}_{ji}) \delta(\hat{R}\mathbf{r}' - \mathbf{r}_{ki}) \propto_r \delta(r - r_{ji}) \delta(r' - r_{ki}) \delta(\theta - \theta_{jki}). \quad (1.7b)$$

## Chapter 1. Theory of atomistic representations

---

We use  $\propto_r$  to omit radial terms  $r$  that appear due to the integration in 3-dimensional space and are not essential as typically a radial scaling term is added to the functional form to control the general scaling. The correlation function of order 2 naturally results in a description of distance to atom  $i$  and the order 3 function in the two distances and an angle wrt. atom  $i$ . Here we use  $\propto_r$  to ignore any radial factors that appear. One approach to obtain a numerical description from this function is based on the concatenation of discrete information of the nonzero values in Eqs. 1.7 to an ordered  $n$ -tensor e.g. pairwise distances[8, 9, 10, 11], 3-body angles, 4-body torsions[12]. To achieve permutational invariance the sorted eigenvalues[8, 11, 13] or sorted tensor entries[12, 14, 15] have been used. For the pairwise distances it has been shown that both approaches for permutational invariance suffer from degeneracies, mapping different atomic structure to the same descriptor[16]. A solution to avoid these degeneracies has been to sum over random permutations of the matrix[9, 10, 11]. While this approach works for small molecules, it does not scale well with the number of atoms. A still computational very costly but feasible solution is to find a canonical permutation for all environments in the data set by constructing a transitive closure of all bi-partite permutations[17]. The bi-partite permutations are determined by solving the matching problem between the principal eigenvectors of two matrices with the Hungarian algorithm[18], the transitive closure is constructed with a minimum spanning tree. Although it has been shown that for regression tasks other descriptors lead to more accurate results[19, 20, 21], still comparable results have been reached with a local atomic environment version of the sorted pairwise distances matrix[15] and a sorted 4-body torsion tensor[12].

### 1.2.2 Fixed basis set

As the concatenation of the coefficient results in noncontinuous descriptions, a problem that arises from keeping varying the basis set for each sample, a natural solution is to expand on the function with a fixed basis set that is used for all environments[22, 23, 24]. By expanding with a basis function for different orders one obtains coefficients of the form

$$b_k(r_{ji}) \propto_r \int_{\mathbb{R}^3} d\mathbf{r} b_k(r) \delta(r - r_{ji}) \quad (1.8a)$$

$$b_k(r, r', \theta_{jki}) \propto_r \int_{\mathbb{R}^3 \times \mathbb{R}^3} d\mathbf{r} d\mathbf{r}' b_k(r, r', \theta_{ijk}) \delta(r' - r_{ji}) \delta(r - r_{ki}) \delta(\theta(\mathbf{r}\mathbf{r}') - \theta_{jki}) \quad (1.8b)$$

For order 1 an analytical solution of the integral in Eq. 1.5 can be solved exploiting properties of the Gaussian function[25]

$$\int_{SO(3)} d\hat{\mathbf{R}} g(\hat{\mathbf{R}}\mathbf{r} - \mathbf{r}_{ji}) \quad (1.9)$$

Solving the integral for higher orders or other densities requires more complex mathematical results. As spherical harmonics  $Y_\mu^\lambda(\hat{\mathbf{r}})$  have been studied extensively in invariant theory[26] it is a suitable candidate to solve the integral in Eq. 1.6. To take advantage of the results in angular momentum theory[27], the density needs to be reexpressed as a combination spherical harmonics. Consequently, for a complete basis of the density a radial basis is needed

## 1.2 Hierarchy of symmetrized representations

---

$\{R_n(r) : \mathbb{R} \rightarrow \mathbb{R}\}$  to cover the radial part of the density. Then density can thus be reformulated as

$$\rho_i(\mathbf{r}) = \sum_{n\lambda\mu} c_{n\lambda\mu} R_n(r) Y_\mu^\lambda(\hat{\mathbf{r}}). \quad (1.10)$$

We will solve Eq. 1.6 for order 2 omitting the radial part for simplicity

$$\sum_{\lambda\lambda'\mu\mu'} \int d\hat{R} Y_\mu^\lambda(\hat{R}\hat{\mathbf{r}}) Y_{\mu'}^{\lambda'}(\hat{R}\hat{\mathbf{r}}') \quad (1.11a)$$

$$= \sum_{\lambda\mu} \int d\hat{R} Y_\mu^\lambda(\hat{R}\hat{\mathbf{r}}) Y_\mu^\lambda(\hat{R}\hat{\mathbf{r}}') \quad (\text{orthogonality spherical harmonics}) \quad (1.11b)$$

$$= \sum_{\lambda\mu} \int d\hat{R} \sum_k D_{\mu k}^\lambda(\hat{R}) Y_k^\lambda(\hat{\mathbf{r}}) \sum_{k'} D_{\mu k'}^\lambda(\hat{R}) Y_{k'}^\lambda(\hat{\mathbf{r}}') \quad (\mathbf{D}(\hat{R}) \text{ is the Wigner D-matrix}) \quad (1.11c)$$

$$= \sum_\lambda Y_k^\lambda(\hat{\mathbf{r}}) Y_{k'}^\lambda(\hat{\mathbf{r}}') \int d\hat{R} \sum_{\mu k k'} D_{\mu k}^\lambda(\hat{R}) D_{\mu k'}^\lambda(\hat{R}) \quad (1.11d)$$

$$= \sum_\lambda \sum_{kk'} \delta_{kk'} Y_k^\lambda(\hat{\mathbf{r}}) Y_{k'}^\lambda(\hat{\mathbf{r}}') \quad (\text{orthogonality Wigner D-matrix}) \quad (1.11e)$$

$$= \sum_{\lambda k} Y_k^\lambda(\hat{\mathbf{r}}) Y_k^\lambda(\hat{\mathbf{r}}') \quad (1.11f)$$

Incorporating the radial part into the above derivation we obtain

$$\sum_{nn'} R_n(r) R_{n'}(r') \sum_{\lambda\mu} Y_\mu^\lambda(\hat{\mathbf{r}}) Y_\mu^\lambda(\hat{\mathbf{r}}') \quad (1.12)$$

This retrieves the well known representation named *smooth overlap of atomic positions*[23]. A solution for the order 3 can be further retrieved using the fact that the product of Wigner D-matrices can be decomposed into a linear combination of Wigner D-matrices

$$D_{m_1 m'_1}^{l_1}(\hat{R}) D_{m_2 m'_2}^{l_2}(\hat{R}) = \sum_{\lambda m m'} D_{\mu m'}^\lambda(\hat{R}) (C_{\mu m_1 m_2}^{\lambda l_1 l_2}) C_{m m'_1 m'_2}^{\lambda l_1 l_2} \quad (1.13)$$

where  $C_{\mu m_1 m_2}^{\lambda l_1 l_2}$  are the Clebsch-Gordan coefficients[27]. This relationship was first used in Ref. [23] to produce order 3 functions, often referred to as *bispectrum* and has later been put into a recursive formula to create expressions for higher-order functions with different approaches for the compression of the basis set in the high-dimensional space[28, 29, 30]

$$f^{v+1}(\mathbf{r}^{v+1}) = \sum_k c_k f_k^v(\mathbf{r}^v) f_k^1(\mathbf{r}) \quad (1.14)$$

Another reason for the choice of spherical harmonics as bases is the fact that they form an irreducible representation of  $SO(3)$ , thus they cannot further compressed without some information loss when expanded on a general function on a surface of a sphere. While a reducible basis as *moment tensor potential*[31] results in lower accuracies, it also leads to a faster evaluation time[32, 33].

### Basis expansion

As the derivation in Eq. 1.11 shows that the evaluation of the integral results in a product sum of radial and angular components. A natural choice for a basis function is choosing  $B_{nn'\lambda}(\mathbf{r}, \mathbf{r}') = R_n(r)R_{n'}(r')\sum_{\mu} Y_{\mu}^{\lambda}(\hat{\mathbf{r}})Y_{\mu}^{\lambda}(\hat{\mathbf{r}}')$ . The order 2 expansion coefficients then decomposes into the spherical expansion coefficients

$$c_{n\lambda\mu} = \int d\mathbf{r} R_n(r) Y_{\mu}^{\lambda}(\hat{\mathbf{r}}) \rho(\mathbf{r}), \quad \text{spherical expansion coefficients} \quad (1.15a)$$

$$\sum_{\mu} c_{n\lambda\mu} c_{n'\lambda\mu} = \int d\mathbf{r} d\mathbf{r}' B_{nn'\lambda}(\mathbf{r}, \mathbf{r}') R_n(r) R_{n'}(r') \sum_{\lambda k} Y_k^{\lambda}(\hat{\mathbf{r}}) Y_k^{\lambda}(\hat{\mathbf{r}}') \quad (1.15b)$$

From the choice of the basis set for the neighbor density in Eq. 1.10 a candidate for order 2 naturally derives that can be constructed from the initial spherical expansion coefficients. Including the recursion in Eq. 1.14 into the consideration it becomes apparent that all higher order correlations can be constructed from one expansion. This fact has been coined *density trick*. It moves the computation from evaluating the basis expansions for  $(v + 1)$ -tuples as it shown in Eqs. 1.8b for oder 2 to computing the tensor products between the expansion coefficients of order  $v$  as indicated the derivation in Eq. 1.15b. Increasing the body-order from the expansion coefficients of order  $v$  to  $v + 1$  scales as  $O(NM + M^v)$  while a direct expansion of the higher-order function scales as  $O(M^{v+1}N^{v+1})$ [34]. As example the costs for order 2 features in Eq. 1.15b without using the density trick the computation of the features for one atomic environem requires the expansion of  $O(\binom{N}{3}) = O(N^3)$  triplets for each of the  $M$  basis functions resulting in a total time complexity of  $O(M^2N^3)$ . With the density trick one has to compute the expansion coefficients for the density  $O(MN)$  to then to increase the order by a contracted tensor product scaling with  $O(M^2)$  resulting a total time complexity of  $O(MN + M^2)$ . Even though the scaling favors the use of the density trick, when embedding splines into the computation of the features, the slower iteration over all  $(v + 1)$ -body parts can be balanced out by omitting the computational cost of tensor product and the model as shown in Ref .[33]. This point will be discussed in more detail in Chapter [? ].

### Radial basis set

The radial basis function is a one dimensional function on a compatact domain  $R_n : [0, r_c] \rightarrow \mathbb{R}$  determined by the cutoff  $r_c$  forming one hyperparameter. In the literature several radial basis functions have been proposed as shifted-Gaussians[23], Chebyshev polynomials[24, 31] or Gaussian type orbitals[35] that all share certain characteristics to shown having positive effects on learning performance. One of these characteristics is the decay of the density with respect to the radial distance together with an increase in the spread of the density as it deemphasizes the importance of information far from the center motivated by the beforementioned principle of nearsightedness[5]. To decrease the redundancy of the basis, one imposes orthogonality of the basis As the dot product is a natural choice for a similarity measure between basis functions it can be seen that orthogonality is a desired property reducing the redundancy

## 1.2 Hierarchy of symmetrized representations

---

between the basis functions

$$\int d\mathbf{r} b_k(\mathbf{r}) b_{k'}(\mathbf{r}) = 0, \text{ if orthogonal.} \quad (1.16)$$

If the basis choice does not provide orthogonality by definition, it is often achieved after the computation of the basis expansion coefficients using Löwden orthogonalization[36]. Another common decision is that the basis functions are spread to uniformly cover the interval  $[0, r_c]$  as first unbiased guess on representing the radial space[37, 38]. This can be proceeded by a subsequent optimization step of the basis exactly for the radial distribution of the dataset.

### Radial expansion

An important result of the expansion in Eq. 1.15a is that it is decomposed into radial and angular expansion term

$$c_{n\lambda\mu} R_n(r) Y_\mu^\lambda(\hat{r}) = \underbrace{c_{n\lambda} R_n^{lambda}(r)}_{\text{radial expansion}} c_{\lambda\mu} \tilde{Y}_\mu^\lambda(\hat{r}). \quad (1.17)$$

The radial basis function couples with the angular basis function which makes an analytic expression nontrivial. One approach to avoid the cost of an numerical integration is to approximate the neighbor density as a decoupled radial and angular contribution[39]

$$\rho_i(\mathbf{r}) \approx \rho_i, r(r) \rho_{i,\perp r}(\hat{\mathbf{r}}) \quad (1.18)$$

Another approach is to chose a radial basis that allows an analytical expression as Gaussian type orbitals[35]. The analytical expression is nowadays further replaced by a spline of for each of the radial expansion coefficents  $c_{n\lambda}$  allowing an evaluation at constant time. Since the splining the radial expansion coefficients has been shown to be cost-efficient[35], costly optimizations of the radial expansion coefficients have entered the design space[40, 41, 42]. These points are discussed in more detail in Chapter 3 about splining.



## 2 Measures of information capacity

In the last chapter we learned about the process of choosing a set of basis functions and the target function to obtain a numerical description, a process referred as *featurization*. As we discussed common characteristics of widely-applied radial basis in Sec. 1.2.2, the design of these characteristics is driven by physical and chemical intuition and is then tested on a variety of datasets to quantify the effectiveness of the design choice[35]. This makes the quantification dependent on a target property and thereby limiting the kind of insights that can be obtained.. In this chapter information measures are presented that extend the ways how we evaluate the quality of features. It is shown how they can be used to give different forms of insights helping with design decision in model construction and how to efficiently compute them.

### 2.1 Reconstruction error

An approach independent of a target property is to compare the reconstruction error of the basis expansion to the original functional form on which the features are expanded Given a representation  $f$  of the atomic structure  $A$  and a basis expansion as in Eq. 1.2 with an orthonormal basis, the approximation error of a basis  $\{b_k\}_{k=1}^M$  to  $f$  can be expressed

$$\ell(\{c_k b_k\}_{k=1}^M, f) = \int_V d\mathbf{q} \left\| \sum_{k=1}^M c_k b_k(\mathbf{q}) - f(\mathbf{q}) \right\|^2. \quad (2.1a)$$

$$\min_{\{c_k b_k\}_{k=1}^M} \ell(\{c_k b_k\}_{k=1}^M, f) \quad (2.1b)$$

Assuming orthonormality of  $b_k$  and normalization of  $f$  then the error can be expressed as

$$\ell(\{c_k b_k\}_{k=1}^M, f) = 2 \left( 1 - \sum_k c_k \int_V d\mathbf{q} b_k(\mathbf{q}) f(\mathbf{q}) \right). \quad (2.2)$$

A weighted sum of dot products between the basis functions and the target functions with expansion coefficients as weights. For example, a Gaussian as basis function centered at  $r_k$

## Chapter 2. Measures of information capacity

---

the similarity the integral results in

$$\int_V d\mathbf{q} b_k(\mathbf{q}) f(\mathbf{q}) = g(|r_k - r_{ij}|). \quad (2.3)$$

The dot product is higher when the Gaussian is close to the target function. In comparison if we can chose basis functions that can never evaluate to 1 ...TODO try `cos(...)`

Such purely analytical treatment only works for simple functional forms. More complex forms can be evaluated by a numerical integration 2.1a as the basis functions are defined on a limited interval restricted by the cutoff  $r_c$ . As example we can compare the effect of a radial scaling term in the basis function applied as global factor  $1/r$  and on the spreading  $\sigma$  into the Gaussian basis functions.

$$R_n^{1/r}(r) = \frac{1}{r} \exp\left(-0.5\left(\frac{r - r_n}{\sigma}\right)^2\right) \text{ (global factor)}, \quad (2.4a)$$

$$R_n^{r\sigma}(r) = \exp\left(-0.5\left(\frac{r - r_n}{r\sigma}\right)^2\right) \text{ (smearing)}, \quad (2.4b)$$

$$R_n^{1/r,r\sigma}(r) = \frac{1}{r} \exp\left(-0.5\left(\frac{r - r_n}{r\sigma}\right)^2\right) \text{ (both)}, \quad (2.4c)$$

$$r_n = r_c \frac{n}{n_{\max}}, \text{ for } n_{\max} > 1. \quad (2.4d)$$

We compare the convergence behavior of these bases on a Lennard-Jones (LJ) potential as domain-specific target function with parameters  $\epsilon_{\text{LJ}} = 1.0$ ,  $\sigma_{\text{LJ}} = 1.1$  on the interval [1.05, 3.0].

$$f_{\text{LJ}}(r) = \epsilon_{\text{LJ}} (\sigma_{\text{LJ}}^{-12} - \sigma_{\text{LJ}}^{-6}). \quad (2.5)$$

From the plots in Fig. 2.1 we can clearly see that the radial scaling term on the smearing reduces the reconstruction error significantly more than the global factor. This observation goes along the lines with the competitiveness of the GTO basis functions that includes a radial scaling term in the smearing shown in Ref. ???. In fact the global radial factor first increases the error up to  $n_{\max} = 4$  till it has a reducing effect.

Due to the duality expressed in Eq. ?? the reconstruction error is a reformulation of the mean-squared linear regression error of the energy determined by the defined LJ potential assuming uniform sampling. Numerous cases of interest however do not have a clear functional form as the LJ potential (e.g. neural network representations) such that this approach becomes unapplicable. Furthermore, in the case of the LJ potential we purposefully did not include the range close to 0 as the target function explodes and a reconstruction error would be dominated by that range providing not much insight. For higher-body orders the control of the intervals of interest becomes more complicated to embed in the analysis. In the next section we show an approach to tackle these problems.

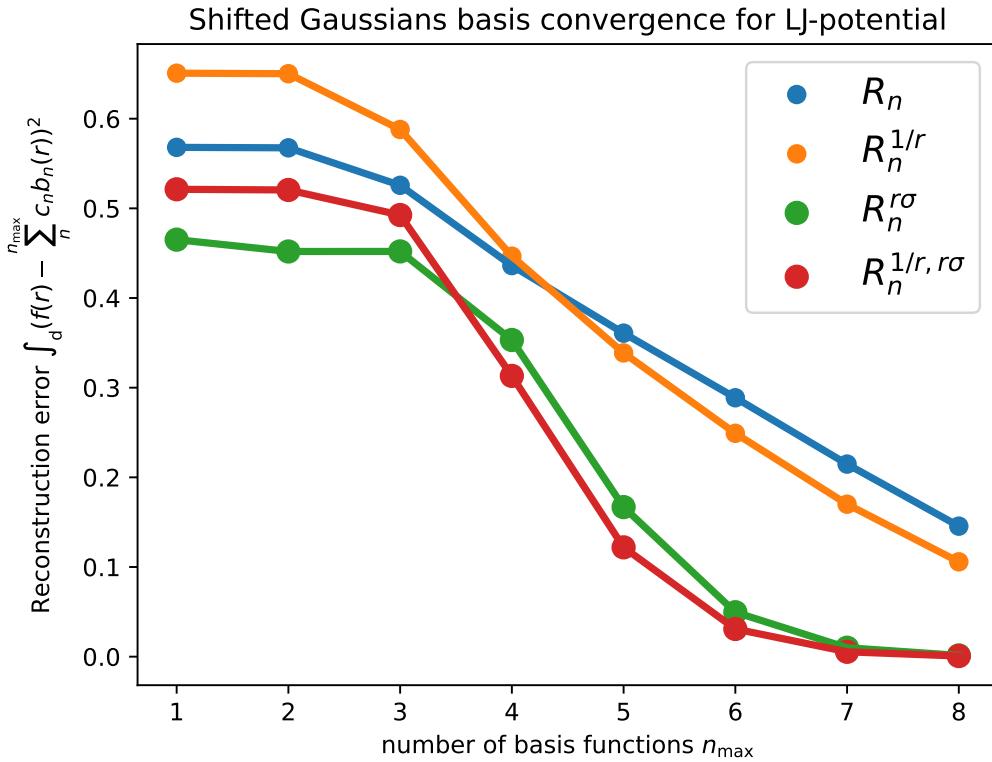


Figure 2.1: Comparison of the effect of radial scaling terms on the basis reconstruction for the LJ potential with the parameters  $\epsilon_{\text{LJ}} = 1.0$ ,  $\sigma_{\text{LJ}} = 1.1$  on the interval  $[1.05, 3.0]$ . A radial scaling in form of a global factor and in the smearing of the Gaussian  $\sigma$  are compared.

## 2.2 Formulation as optimization problem

To address the problems discussed in the last section we reformulate the integration problem into an optimization problem that can be solved more efficiently.

$$\ell(\{c_n b_n\}_{n=1}^M, \{c'_k b'_k\}_{n=1}^M) = \int_D d\mathbf{q} \|\mathbf{T}(\mathbf{c}_\mathbf{q} \odot \mathbf{b}(\mathbf{q})) - \mathbf{c}'_\mathbf{q} \odot \mathbf{b}'(\mathbf{q})\|^2, \quad \mathbf{c}, \mathbf{b} \in \mathbb{R}^{n_{\max}}, \mathbf{T} \in \mathbb{R}^{n_{\max} \times n'_{\max}} \quad (2.6)$$

Assuming an orthonormal basis we obtain

$$\ell(\{c_n b_n\}_{n=1}^M, \{c'_k b'_k\}_{n=1}^M) = \int_D d\mathbf{q} \|\mathbf{T}\mathbf{c}_\mathbf{q} - \mathbf{c}'_\mathbf{q}\|^2 \quad (2.7)$$

Assuming orthogonormality can be a restrictive assumption. We can see the dependency on the orthonormalization matrix  $\mathbf{S}^{-\frac{1}{2}}$  on the target features

$$\min_{\mathbf{T}} \|\mathbf{T}\mathbf{c}_\mathbf{q} - \mathbf{S}^{-\frac{1}{2}}\mathbf{c}'_\mathbf{q}\|^2 \leq \|\mathbf{S}^{-\frac{1}{2}}\|^2 \min_{\mathbf{T}} \|\mathbf{T}\mathbf{c}_\mathbf{q} - \mathbf{c}'_\mathbf{q}\|^2. \quad (2.8a)$$

This expresses just the general dependency of the error on linear transformations of the target function when no full reconstruction is possible. Assume a reconstruction of  $\{(1, 0)_1, (0, 1)_2\}$  by coefficients  $\{(1, 0)_1, (1, 0)_2\}$ . By introducing a linear transformation to the targeted coefficients, the error can be made arbitrary large. This is crucial since it limits the insights we can get about the feature relationship, since a full rank linear transformation of the features does not reduce regression performance, but vice-versa the reconstruction error. For analysis that still can be related the meaning of Eq. 2.6 we therefore have to use an orthonormal basis for the target features. In this case a full rank orthonormality matrix can be absorbed by the linear transformation so we have equivalence to the orthonormal coefficients

$$\min_{\mathbf{T}} \|\mathbf{T}\mathbf{c}_\mathbf{q} - \mathbf{c}'_\mathbf{q}\|^2. = \min_{\mathbf{T}} \|\mathbf{T}\mathbf{S}^{-\frac{1}{2}}\mathbf{c}_\mathbf{q} - \mathbf{c}'_\mathbf{q}\|^2, \quad (2.9a)$$

so we only constrain the targeted feature space to be orthonormal if we want to retrieve Eq. 2.6. On the other hand reconstruction of a nonorthonormal basis can still be justified by the fact that until the limit of convergence is not reached even a full rank linear transformation affects the regularization and thus also effects the regression performance.

In the following sections we show error measures based on the loss in Eq. 2.7 that constraint the transformation  $\mathbf{T}$  to provide different insights to the feature-relationship. In Figure 2.2 a schematic of the different types of transformations  $T$  that are covered in the upcoming sections. These methods are of particular use in assessing the hyperparameters of ML descriptors[34] and have been employed to compare the efficiency of different basis sets in encoding geometrical information[4, 40].

## 2.2 Formulation as optimization problem

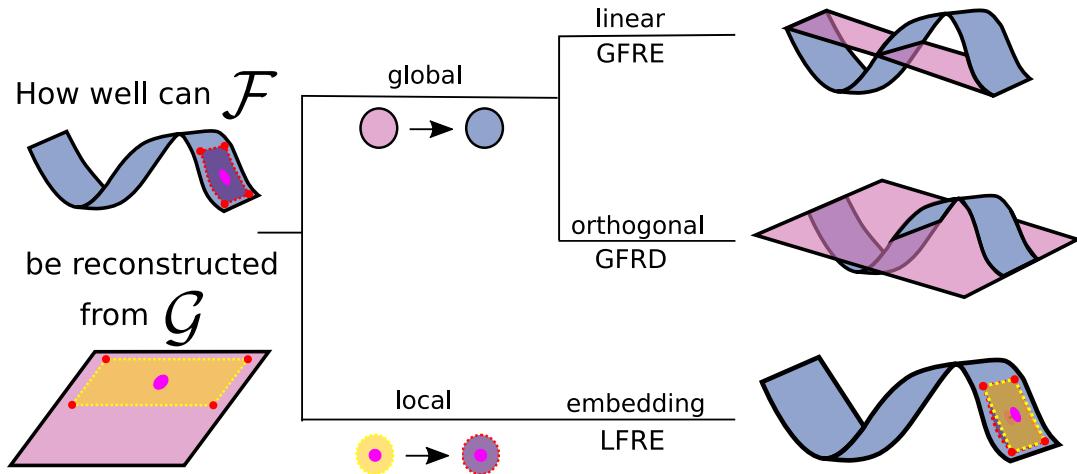


Figure 2.2: **The different forms of feature reconstructions** to assess two feature spaces (blue and pink) describing the same dataset. Here, we are reconstructing the curved manifold (blue) using the planar manifold (pink), as it is often the case to approximate a complex manifold with a simpler alternative. The area framed by the dotted line is an example of a local neighbourhood of one sample (the pink dot) that enables the reconstruction of nonlinearities. (Top) The linear transformation is used in the global feature reconstruction error (GFRE). (Middle) The orthogonal transformation is used in the global feature reconstruction distortion (GFRD). (Bottom) A local linear transformation of a neighbourhood is used in local feature reconstruction error (LFRE). On the right, the reconstructions of the manifold are drawn in pink together with the curved manifold in blue. The measures correspond to the root-mean-square difference between the reconstructed and curved manifold.

## 2.3 Linear decodable information

The FRMs differ in two aspects: the locality of the reconstruction (global or local) and the constraints of the regression. In each FRM, the two feature sets are partitioned into training and testing sets. We standardise the features of  $\mathcal{F}$  and  $\mathcal{G}$  individually, then we regress the features of  $\mathcal{F}$  onto  $\mathcal{G}$  to compute the errors. In the global measures, we use the entire dataset for the reconstruction, whereas in the local measure, we perform a regression for each sample on the set of the  $k$ -nearest points within  $\mathcal{F}$ . The number  $k$  is given by the user with the parameter `n_local_points`. The *reconstruction error* is by default computed using a 2-fold cross-validation (CV) and ridge regression as estimator.

As most interesting applications use large feature vectors, we implemented a custom 2-fold ridge to improve computational efficiency. In `slearn`, an implementation of the leave-one-out CV with a similar purpose of speeding up the CV exists. Put here some results on the comparison contained in the examples of `skmatter`

As a simple, easily-interpretable measure of the relative expressive power of  $\mathcal{F}$  and  $\mathcal{F}'$ , we introduce the global feature space reconstruction error  $\text{GFRE}^{\mathcal{D}}(\mathcal{F}, \mathcal{F}')$ , defined as the mean-square error that one incurs when using the feature matrix  $\mathbf{X}_{\mathcal{F}}$  to linearly regress  $\mathbf{X}_{\mathcal{F}'}$ . In this work we compute the GFRE by a 2-fold split of the dataset, i.e. compute the regression weights  $\mathbf{P}_{\mathcal{F}\mathcal{F}'}$  over a train set  $\mathcal{D}_{\text{train}}$  composed of half the entries in  $\mathcal{D}$ ,

$$\begin{aligned} \mathbf{P}_{\mathcal{F}\mathcal{F}'} &= \underset{\mathbf{P} \in \mathbb{R}^{m_{\mathcal{F}} \times m_{\mathcal{F}'}}}{\operatorname{argmin}} \left\| \mathbf{X}_{\mathcal{F}'}^{\mathcal{D}_{\text{train}}} - \mathbf{X}_{\mathcal{F}}^{\mathcal{D}_{\text{train}}} \mathbf{P} \right\| \\ &= \left( \mathbf{X}_{\mathcal{F}}^{\mathcal{D}_{\text{train}}} {}^T \mathbf{X}_{\mathcal{F}}^{\mathcal{D}_{\text{train}}} \right)^{-1} (\mathbf{X}_{\mathcal{F}}^{\mathcal{D}_{\text{train}}} {}^T \mathbf{X}_{\mathcal{F}'}^{\mathcal{D}_{\text{train}}} \mathbf{P}) \end{aligned} \quad (2.10)$$

and then compute the error over the remaining test set  $\mathcal{D}_{\text{test}}$

$$\text{GFRE}^{\mathcal{D}}(\mathcal{F}, \mathcal{F}') = \sqrt{\left\| \mathbf{X}_{\mathcal{F}'}^{\mathcal{D}_{\text{test}}} - \mathbf{X}_{\mathcal{F}}^{\mathcal{D}_{\text{test}}} \mathbf{P}_{\mathcal{F}\mathcal{F}'} \right\|^2 / n_{\text{test}}}, \quad (2.11)$$

averaging, if needed, over multiple random splits. The GFRE is a positive quantity, which is equal to zero when there is no error in the reconstruction, and that is usually bound by one<sup>1</sup>.

Notice that by expressing the basis and target functions on a radial grid basis we retrieve the numerical integration loss in Eq. ???. The minimization problem is moved from an optimization of  $\mathbf{c}$  to an optimization of  $\mathbf{W}$ . While the former requires to solve the loss for each datapoint the latter only needs one the whole dataset. As solving the minimization problem has worst scaling behavior in the computation of the loss, the later approach is typically more efficient to compute.

For numbers of features larger than  $n_{\text{train}}$ , the covariance matrix is not full rank, and one needs to compute a pseudoinverse. Without loss of generality, one can regularize the regression to stabilize the calculation. In this paper, we computed the pseudoinverse by means of a

---

<sup>1</sup>This is due to the fact that feature matrices are standardized, and so  $\left\| \mathbf{X}_{\mathcal{F}'}^{\mathcal{D}_{\text{test}}} \right\| / n_{\text{test}}$  is of the order of one

singular value decomposition, and we determined the optimal regularization in terms of the truncation of the singular value spectrum, using 2-fold cross-validation over the training set to determine the optimal truncation threshold. Often, it is also useful to observe the behavior of the GFRE in the absence of any regularization: overfitting is in itself a signal of the instability of the mapping between feature spaces. In general,  $\text{GFRE}^{\mathcal{D}}(\mathcal{F}, \mathcal{F}')$  is not symmetric. If  $\text{GFRE}^{\mathcal{D}}(\mathcal{F}, \mathcal{F}') \approx \text{GFRE}^{\mathcal{D}}(\mathcal{F}', \mathcal{F}) \approx 0$ ,  $\mathcal{F}$  and  $\mathcal{F}'$  contain similar types of information; if  $\text{GFRE}^{\mathcal{D}}(\mathcal{F}, \mathcal{F}') \approx 0$ , while  $\text{GFRE}^{\mathcal{D}}(\mathcal{F}', \mathcal{F}) > 0$ , one can say that  $\mathcal{F}$  is more descriptive than  $\mathcal{F}'$ : this is the case, for instance, one would observe if  $\mathcal{F}'$  consists of a sparse version of  $\mathcal{F}$ , with some important and linearly-independent features removed; finally, if  $\text{GFRE}^{\mathcal{D}}(\mathcal{F}, \mathcal{F}') \approx \text{GFRE}^{\mathcal{D}}(\mathcal{F}', \mathcal{F}) > 0$ , the two feature spaces contain different, and complementary, kinds of information and it may be beneficial to combine them to achieve a more thorough description of the problem.

### 2.3.1 Efficient cross-validation ridge for high-dimensional feature space

In linear regression, the complexity of computing the weight matrix is theoretically bounded by the inversion of the covariance matrix. This is more costly when conducting regularized regression, wherein we need to optimise the regularization parameter in a cross-validation (CV) scheme, thereby recomputing the inverse for each parameter. scikit-learn offers an efficient leave-one-out CV (LOO CV) for its ridge regression which avoids these repeated computations [43]. Because we needed an efficient ridge that works in predicting for the reconstruction measures in metric we implemented in `skmatter` an efficient 2-fold CV ridge regression that uses a singular value decomposition (SVD) to reuse it for all regularization parameters  $\lambda$ . Assuming we have the standard regression problem optimizing the weight matrix in

$$\|\mathbf{X}\mathbf{W} - \mathbf{Y}\| \tag{2.12}$$

Here  $\mathbf{Y}$  can be seen also a matrix as it is in the case of multi target learning. Then in 2-fold cross validation we would predict first the targets of fold 2 using fold 1 to estimate the weight matrix and vice versa. Using SVD the scheme estimation on fold 1 looks like this.

$$\mathbf{X}_1 = \mathbf{U}_1 \mathbf{S}_1 \mathbf{V}_1^T, \quad \text{feature matrix } \mathbf{X} \text{ for fold 1} \tag{2.13}$$

$$\mathbf{W}_1(\lambda) = \mathbf{V}_1 \tilde{\mathbf{S}}_1(\lambda)^{-1} \mathbf{U}_1^T \mathbf{Y}_1, \quad \text{weight matrix fitted on fold 1} \tag{2.14}$$

$$\tilde{\mathbf{Y}}_2 = \mathbf{X}_2 \mathbf{W}_1, \quad \text{prediction of } \mathbf{Y} \text{ for fold 2} \tag{2.15}$$

The efficient 2-fold scheme in ‘Ridge2FoldCV’ reuses the matrices

$$\mathbf{A}_1 = \mathbf{X}_2 \mathbf{V}_1, \quad \mathbf{B}_1 = \mathbf{U}_1^T \mathbf{Y}_1. \quad (2.16)$$

for each fold to not recompute the SVD. The computational complexity after the initial SVD is thereby reduced to that of matrix multiplications.

We can see that Leave-one-out CV is estimating the error wrong for low alpha values. That seems to be a numerical instability of the method. If we would have limit our alphas to 1E-5, then LOO CV would have reach similar accuracies as the 2-fold method.

We note that this is not an fully encompassing comparison covering sufficient enough the parameter space. We just want to note that in cases with high feature size and low effective rank the ridge solvers in skmatter can be numerical more stable and act on a comparable speed.

### Cutoff and Tikhonov regularization

When using a hard threshold as regularization (using parameter “cutoff”), the singular values below  $\lambda$  are cut off, the size of the matrices  $\mathbf{A}_1$  and  $\mathbf{B}_1$  can then be reduced, resulting in further computation time savings. This performance advantage of *cutoff* over the *Tikhonov* is visible if we to predict multiple targets and use a regularization range that cuts off a lot of singular values. For that we increase the feature size and use as regression task the prediction of a shuffled version of  $\mathbf{X}$ .

We can see that a regularization value of 1e-8 cuts off a lot of singular values. This is crucial for the computational speed up of the *cutoff regularization method*

#### 2.3.2 Example: Wasserstein distance

As an example of the transformation induced by a non-Euclidean metric we consider the effect of using a Wasserstein distance to compare  $v = 1$  density correlation features. The Wasserstein distance (also known as the Earth Mover Distance, EMD) is defined as the minimum “work” that is needed to transform one probability distribution into another – with the work defined as the amount of probability density multiplied by the extent of the displacement [44, 45, 46]. The EMD has been used to define a “regularized entropy match” kernel to combine local features into a comparison between structures [47], to obtain permutation-invariant kernels based on Coulomb matrices[48], and has been shown to be equivalent to the Euclidean distance between vectors of sorted distances [49]. Here we use the Wasserstein distance to compare two-body ( $v = 1$ ) features, that can be expressed on a real-space basis and take the form of one-dimensional probability distributions.

The formal definition of the Wasserstein distance of order 2 between two probability distribu-

tions  $p(r)$  and  $p'(r)$  defined on a domain  $M$  reads

$$W(p, p')^2 = \inf_{\gamma \in \Gamma(p, p')} \int_{M \times M} d(r, r')^2 d\gamma(r, r'), \quad (2.17)$$

where  $\Gamma(p, p')$  is the set of all joint distributions with marginals  $p$  and  $p'$ . For 1-dimensional distributions,  $W(p, p')$  can be expressed as the 2-norm of the difference between the associated inverse cumulative distribution function (ICDF)  $P^{-1}$  of two environments,  $W(p, p')^2 = \int_0^1 |P^{-1}(s) - P'^{-1}(s)|^2 ds$ , with  $P(r) = \int_0^r p(r) dr$

In order to express the symmetrized 2-body correlation function as a probability density, we first write it on a real-space basis  $\langle r |$ , and evaluate it on 200 Gaussian quadrature points, that we also use to evaluate the CDF and its inverse. We then proceed to normalize it, so that it can be interpreted as a probability density. We estimate the integral of the distribution (that effectively counts the number of atoms within the cutoff distance)

$$Z_i = \int_0^{r_c} \langle r | A; \overline{\rho_i^{\otimes 1}} \rangle dr, \quad (2.18)$$

and the maximum value of the integral over the entire dataset  $Z_{\mathcal{D}}$ . A simple scaling of the correlation function

$$p_i^s(r) = \frac{1}{Z_i} \langle r | A; \overline{\rho_i^{\otimes 1}} \rangle \quad (2.19)$$

distorts the comparison between environments with different numbers of atoms. To see how, we use the *displaced methane* dataset, in which three atoms in a CH<sub>4</sub> molecule are held fixed in the ideal tetrahedral geometry, at a distance of 1Å from the carbon centre. The fourth atom, aligned along the  $z$  axis, is displaced along it, so that each configuration is parameterised by a single coordinate  $z_H$ . Figure ??(a) shows the distance computed between pairs of configurations with different  $z_H$ , demonstrating the problem with the renormalized probability (2.19):  $p^s$  loses information on the total number of atoms within the cutoff, and so once the tagged atom moves beyond  $r_c$  the remaining CH<sub>3</sub> environment becomes indistinguishable from an ideal CH<sub>4</sub> geometry.

One can obtain a more physical behavior when atoms enter and leave the cutoff by introducing a  $\delta$ -like “sink” at the cutoff distance, defining

$$p_i^\delta(r) = \frac{1}{Z_{\mathcal{D}}} \left[ \langle r | \mathcal{A}; \overline{\rho_i^{\otimes 1}} \rangle + (Z_{\mathcal{D}} - Z_i) \delta(r - r_c) \right]. \quad (2.20)$$

Fig. ??b shows that with this choice the Wasserstein metric between  $p_i^\delta(r)$  reflects the distance between the moving atoms. With this normalization, in fact, the Wasserstein metric corresponds to a smooth version of the Euclidean metric computed between vectors of sorted interatomic distances [49], shown in Fig. ??c. The distortions that can be seen in the comparison between Fig. ??b,c are a consequence of the Gaussian smearing, the smooth cutoff function, and the SO(3) integration that modulates the contribution to  $\langle r | \mathcal{A}; \overline{\rho_i^{\otimes 1}} \rangle$  coming from atoms at different distances.

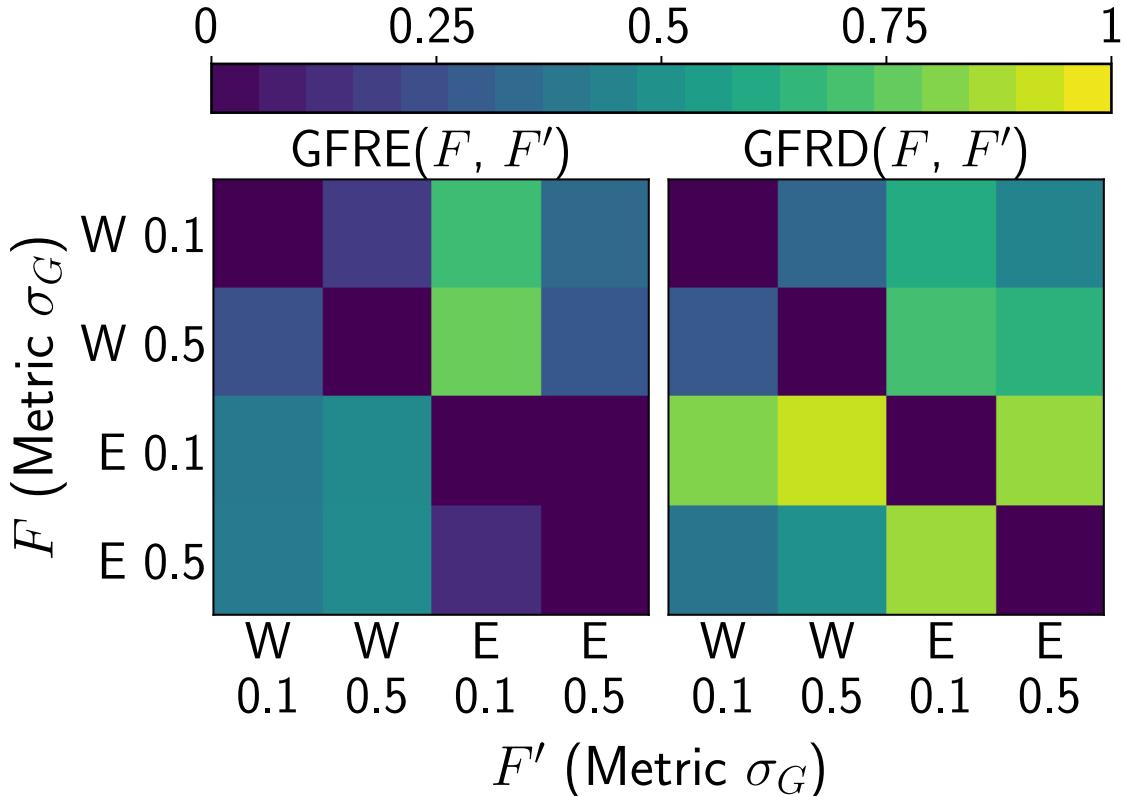


Figure 2.3: Comparison of GFRE and GFRD for the *carbon* dataset, using sharp ( $\sigma_G = 0.1\text{\AA}$ ) and smooth ( $\sigma_G = 0.5\text{\AA}$ ) radial SOAP features, as well as Euclidean (E) and Wasserstein (W) metrics.

Having defined a meaningful normalization and a probabilistic interpretation of the radial density correlation features, we can investigate how the feature space induced by a Wasserstein metric relates to that induced by an Euclidean distance. Figure ?? shows the error in the reconstruction of  $z_H$  for the *displaced methane* dataset when restricting the training set to  $0.05\text{\AA}$  and  $1.0\text{\AA}$  spaced grids. Using a Euclidean distance with a sharp  $\sigma_G$  leads to a highly non-linear mapping between the displacement coordinate and feature space, and a linear model cannot interpolate accurately between the points of a sparse grid. A Wasserstein metric, on the other hand, measures the minimal distortion needed to transform one structure into another, and so provides a much more natural interpolation along  $z_H$ , which is robust even with a sharp density and large spacing between training samples. It is worth stressing that the sorted distance metric – which effectively corresponds to the  $\delta$  density limit of the Wasserstein metric – performs rather poorly, and cannot even reproduce the training points. This is because the mapping between feature space and  $z_H$  is not exactly linear, changing slope when  $z_H$  crosses  $1\text{\AA}$  (because the sorting of the vector changes) and  $4\text{\AA}$  (because one atom exits the cutoff). The sorted-distances feature space does not have sufficient flexibility to regress this piecewise linear map, as opposed to its smooth Wasserstein counterpart.

Having rationalized the behavior of the Wasserstein metric for a toy model, we can test

how it compares to the conventional Euclidean metric on a more realistic data set. We consider in particular the AIRSS *carbon* data set, and compare different levels of density smearing as well as Euclidean and Wasserstein metrics. Figure 2.3 paints a rather nuanced picture of the relationship between the linear and the Wasserstein-induced feature spaces. The GFRE is non-zero in both directions, meaning that (in a linear sense) Wasserstein and Euclidean features provide complementary types of information. Smearing of the density has a small effect on the Wasserstein metric, so that both  $\text{GFRE}(W(\sigma_G = 0.1\text{\AA}), W(\sigma_G = 0.5\text{\AA}))$  and  $\text{GFRD}(W(\sigma_G = 0.1\text{\AA}), W(\sigma_G = 0.5\text{\AA}))$  are small, whereas for Euclidean features – as observed in Section ?? – changing  $\sigma_G$  induces small information loss, but a large distortion of feature space. Overall, there is no sign of the pathological behavior seen in Fig. ??, which is an indication that (at least for 2-body features) the *carbon* dataset is sufficiently dense, and that the better interpolative behavior of the EMD does not lead to a more informative feature space.

### 2.3.3 Example: Comparison of bispectrum and message-passing

Put in Jigyasa's work where the GFRE was used to compare MP with 3-body.

## 2.4 Distortion in linear transformations

The feature space reconstruction error gives insights into whether a feature space can be inferred by knowledge of a second one. However, having both a small  $\text{GFRE}^{\mathcal{D}}(\mathcal{F}, \mathcal{F}')$  and  $\text{GFRE}^{\mathcal{D}}(\mathcal{F}', \mathcal{F})$  does not imply two feature spaces are identical. Even though they contain similar amounts of information, one feature space could give more emphasis to some features compared to the other, which can eventually result in different performance when building a model. To assess the amount of distortion of  $\mathcal{F}'$  relative to  $\mathcal{F}$ , we introduce the global feature space reconstruction distortion  $\text{GFRD}^{\mathcal{D}}(\mathcal{F}, \mathcal{F}')$ . To evaluate it, we first compute the singular value decomposition of the projector Eq. (2.10),  $\mathbf{P}_{\mathcal{F}\mathcal{F}'} \approx \mathbf{U}\Sigma\mathbf{V}^T$ , and then use it to reduce the two feature spaces to a common basis, in which the reconstruction error is zero, because the residual has been discarded

$$\tilde{\mathbf{X}}_{\mathcal{F}} = \mathbf{X}_{\mathcal{F}} \mathbf{U} \quad \tilde{\mathbf{X}}_{\mathcal{F}'} = \tilde{\mathbf{X}}_{\mathcal{F}} \Sigma. \quad (2.21)$$

When the second feature space  $\mathcal{F}'$  has a lower dimensionality than  $\mathcal{F}$ , some combinations of the starting features are not used to compute  $\tilde{\mathcal{F}'}$ . In this case, we pad  $\Sigma$  with zeros, so that  $\tilde{\mathcal{F}'}$  has the same dimensionality  $m_{\mathcal{F}'}$  as the starting space. This choice ensures that the GFRD takes the same value it would have in the case  $\mathcal{F}'$  had the same dimensionality as  $\mathcal{F}$ , but lower rank. In the opposite case, with  $m_{\mathcal{F}} < m'_{\mathcal{F}'}$ , padding  $\Sigma$  and  $\mathbf{U}$  with zeros, or truncating  $\mathbf{V}$ , yields the same GFRD.

We can then address the question of whether  $\tilde{\mathbf{X}}_{\mathcal{F}}$  and  $\tilde{\mathbf{X}}_{\mathcal{F}'}$  are linked by a unitary transformation (in which case the GFRD should be zero), or there is a distortion involved. A possible answer involves solving the orthogonal Procrustes problem [50] – i.e. finding the orthogonal

transformation that “aligns” as well as possible  $\tilde{\mathbf{X}}_{\mathcal{F}}$  to  $\tilde{\mathbf{X}}_{\mathcal{F}'}$ :

$$\begin{aligned} \mathbf{Q}_{\mathcal{F}\mathcal{F}'} &= \operatorname{argmin}_{\mathbf{Q} \in \mathbb{U}^{m \times m}} \left\| \tilde{\mathbf{X}}_{\mathcal{F}'}^{\mathcal{D}_{\text{train}}} - \tilde{\mathbf{X}}_{\mathcal{F}}^{\mathcal{D}_{\text{train}}} \mathbf{Q} \right\| \\ &= \tilde{\mathbf{U}} \tilde{\mathbf{V}}^T, \end{aligned} \quad (2.22)$$

where  $\tilde{\mathbf{U}} \tilde{\Sigma} \tilde{\mathbf{V}}^T = (\tilde{\mathbf{X}}_{\mathcal{F}}^{\mathcal{D}_{\text{train}}})^T \tilde{\mathbf{X}}_{\mathcal{F}'}^{\mathcal{D}_{\text{train}}}$ . The amount of distortion can then be computed by assessing the residual on the test set,

$$\text{GFRD}^{\mathcal{D}}(\mathcal{F}, \mathcal{F}') = \sqrt{\left\| \tilde{\mathbf{X}}_{\mathcal{F}'}^{\mathcal{D}_{\text{test}}} - \tilde{\mathbf{X}}_{\mathcal{F}}^{\mathcal{D}_{\text{test}}} \mathbf{Q}_{\mathcal{F}\mathcal{F}'} \right\|^2 / n_{\text{test}}}. \quad (2.23)$$

If desired, the error can be averaged over multiple random splits of the reference data set  $\mathcal{D}$ .

#### 2.4.1 Example: Radial scaling

One of the most important hyperparameters when defining an atom-centred representation is the cutoff distance, which restricts the contributions to the density to the atoms with  $r_{ji} < r_c$ . Fig. 2.4(c,d) shows that the GFRE captures the loss of information associated with an aggressive truncation of the environment, with very similar behavior between GTO and DVR bases. The figure also reflects specific features of the different data sets: for instance,  $\text{GFRE}(r_c = 4\text{\AA}, r_c = 6\text{\AA})$  is close to zero for the random methane data set, because there are no structures where atoms are farther than  $4\text{\AA}$  from the centre of the environment.  $\text{GFRE} > 0$  also when mapping long-cutoff features to short-range features, although the reconstruction error is much smaller than in the opposite direction. This indicates the need for an increase in  $n_{\max}$  to fully describe the structure of an environment when using a large value of  $r_c$ , which is consistent with the greater amount of information encoded within a larger environment. The GFRD plot also underscores the strong impact of the choice of  $r_c$  on the emphasis that is given to different parts of the atom-density correlations. This effect explains the strong dependency of regression performance on  $r_c$ , and the success of multi-scale models that combine features built on different lengthscales [51]. A similar modulation of the contributions from different radial distances can be achieved by scaling the neighbour contribution to the atom-centred density by a decaying function, e.g.  $1/(1 + (r_{ji}/r_0)^s)$ . This approach has proven to be very effective in fine-tuning the performance of regression models using density-based features [52, 53, 54]. As shown in Fig. 2.4(e,f), this is an example of a transformation of the feature space that entails essentially no information loss – resulting in a very small GFRE between different values of the scaling exponent  $s$ . However, it does result in substantial GFRD, providing additional evidence of how the emphasis given by a set of features to different interatomic correlations can affect regression performance even if it does not remove altogether pieces of structural information.

## 2.4 Distortion in linear transformations

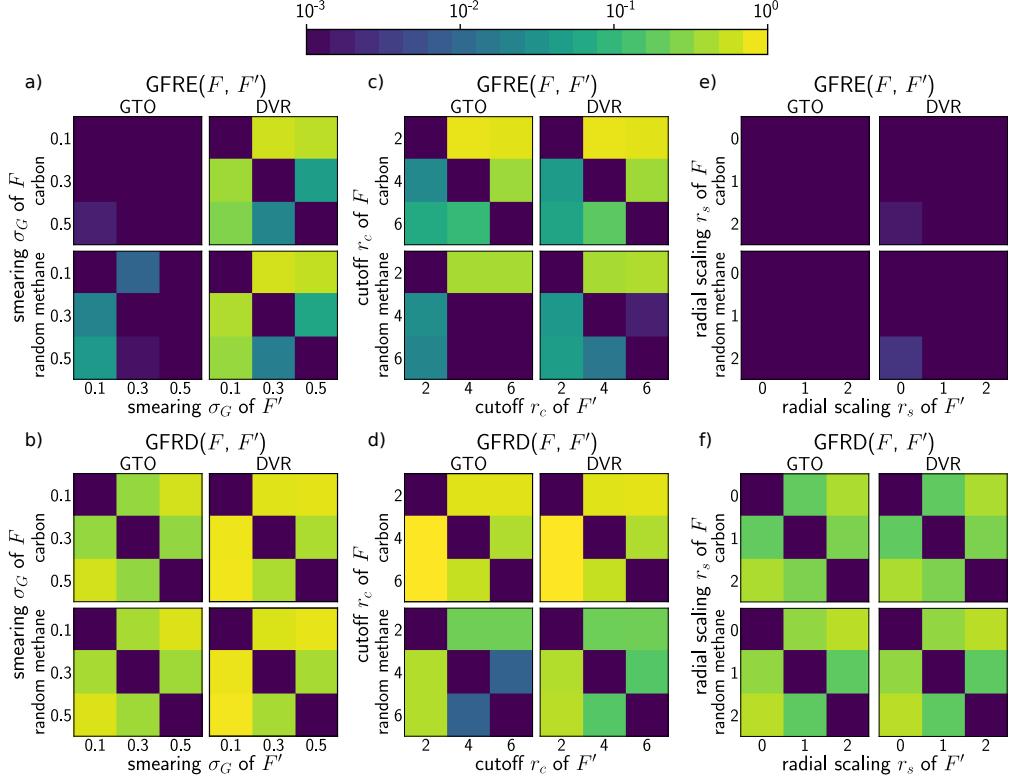


Figure 2.4: Comparison of the GFRE (top) and GFRD (bottom) a),b) for different smearing  $\sigma_G$  ( $r_c = 4\text{\AA}$ ) c),d) for different cutoff values ( $\sigma_G = 0.5\text{\AA}$ ), and e),f) for different radial scaling exponents ( $r_c = 4\text{\AA}$ ,  $\sigma_G = 0.5\text{\AA}$ ). For all comparisons  $(n_{\max}, l_{\max}) = (10, 6)$  were used. The feature specified by the row is used to reconstruct the feature specified by the column.

## 2.5 Nonlinearity as local linearity

The limitation to linear transformations allows us to efficiently compute the loss over a dataset, but also restricts us from describing nonlinear relationships. A branch of models are based on NNs and apply nonlinearities to an initial representation to obtain their final one. As these models are also often opaque in their learned representation, it is essential to develop a measure that can also work for nonlinearities.

In principle for the transformation  $T$  in Eq. ?? a neural network could be used that model nonlinearities. Neural networks are however usually only efficient in the prediction of a few target properties and not large number of features where convergence might be unfeasible. In addition NNs have a lot model parameters that need to be chosen (e.g. activation function, number of layers) making the loss highly dependent on the exact architecture which makes it hard to make general statements.

In this Section we discuss ways to model nonlinearities the linearities and its limitations. We propose for future directions to restrict the type of nonlinearities carefully which still allows to answer certain research questions.

### 2.5.1 Local linear embedding

An alternative approach is to compute a local version of the feature space reconstruction error,  $\text{LFRE}^{\mathcal{D}}(\mathcal{F}, \mathcal{F}')$ , loosely inspired by locally-linear embedding [55]. To compute the LFRE, a local regression is set up, computed in the  $k$ -neighbourhood  $\mathcal{D}_{k-\text{neigh}}^{(i)}$  around sample  $i$  – the set of  $k$  nearest neighbours of sample  $i$ , based on the Euclidean distance between the samples in  $\mathcal{F}$  – to reproduce the  $\mathcal{F}'$  features using  $\mathcal{F}$  as input features, centred around their mean values  $\bar{\mathbf{x}}_{\mathcal{F}'}$  and  $\bar{\mathbf{x}}_{\mathcal{F}}$ .

A local embedding of  $\mathbf{x}_i$  is determined as

$$\tilde{\mathbf{x}}'_i = \bar{\mathbf{x}}_{\mathcal{F}'} + (\mathbf{x}_i - \bar{\mathbf{x}}_{\mathcal{F}}) \mathbf{P}_{\mathcal{F}\mathcal{F}'}^{(i)}, \quad (2.24)$$

where  $\mathbf{P}_{\mathcal{F}\mathcal{F}'}^{(i)}$  contains the regression weights computed from  $\mathcal{D}_{k-\text{neigh}}^{(i)}$ . The local feature space reconstruction error is given by the residual discrepancy between the  $\mathcal{F}'$  counterpart of the  $i$ -th point and its local embedding (2.24):

$$\text{LFRE}^{\mathcal{D}}(\mathcal{F}, \mathcal{F}') = \sqrt{\sum_i \|\mathbf{x}'_i - \tilde{\mathbf{x}}'_i\|^2 / n_{\text{test}}}. \quad (2.25)$$

Inspecting the error associated with the reconstruction of individual points can reveal regions of feature space for which the mapping between  $\mathcal{F}$  and  $\mathcal{F}'$  is particularly problematic. Similarly, one can compute a local version of GFRD, that could be useful to detect strong local distortions that might indicate the presence of a singularity in the mapping between two feature spaces.

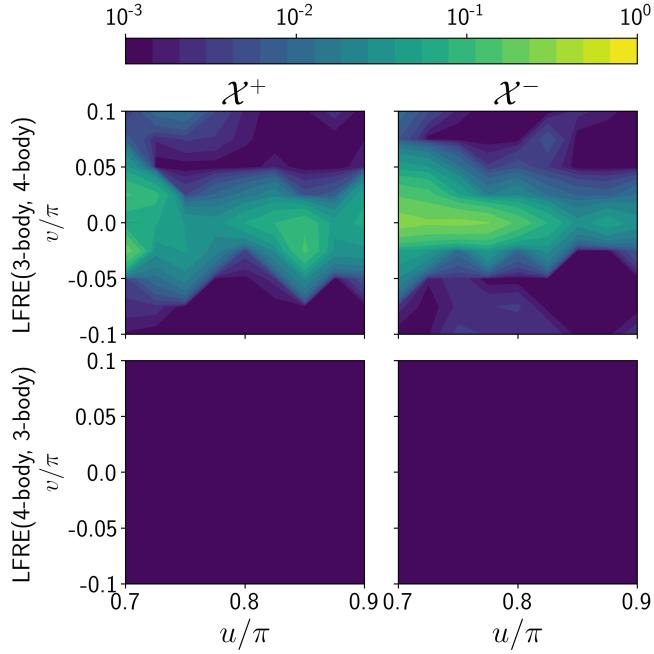


Figure 2.5: Pointwise LFRE for the structures from the degenerate methane dataset as a function of the structural coordinates  $(u, v)$  for  $(n_{\max}, l_{\max}) = (6, 4)$  and  $k = 15$  neighbours.

### Example - Degeneracies in 3-body features

The LFRE also makes it possible to identify regions of phase space for which the construction of a mapping between feature spaces is difficult or impossible. Consider the case of the degenerate manifold discussed in Ref. 56. The dataset includes two sets of CH<sub>4</sub> environments, and those parameterised by  $v = 0$  cannot be distinguished from each other using 3-body ( $v = 2$ ) features. Fig. 2.5 shows the LFRE for each point along the two manifolds. When trying to reconstruct 3-body features using as inputs 4-body features (that take different values for the two manifolds) the LFRE is essentially zero. When using the 3-body features as inputs, instead, one observes a very large error for points along the degenerate line, while points that are farther along the manifold can be reconstructed well. This example demonstrates the use of the LFRE to identify regions of feature space for which a simple, low-body-order representation is insufficient to fully characterize the structure of an environment, and can be used as a more stringent, explicit test of the presence of degeneracies than the comparison of pointwise distances discussed in Ref. 56.

### 2.5.2 Jacobian

For an atomic environment the rows of a Jacobian matrix correspond to the spatial directions of the atoms in the environment. A row describes the change of the features with respect to a change in the corresponding spatial direction of the atom. If we treat the Jacobian reconstruction as a classical linear learning problem, then these rows represent the sample

$$\mathbf{J}_1 \in \mathbb{R}^{3n_{\text{atoms}}, n_{\text{feat}}}$$

$$\mathbf{J}_1 = \begin{bmatrix} \frac{\partial \langle q_1 | A_1 \rangle}{\partial \mathbf{r}_{11}} & \dots & \frac{\partial \langle q_{n_{\text{feat}}} | A_1 \rangle}{\partial \mathbf{r}_{11}} \\ \frac{\partial \langle q_1 | A_1 \rangle}{\partial \mathbf{r}_{21}} & \dots & \frac{\partial \langle q_{n_{\text{feat}}} | A_1 \rangle}{\partial \mathbf{r}_{21}} \\ \frac{\partial \langle q_1 | A_1 \rangle}{\partial \mathbf{r}_{31}} & \dots & \frac{\partial \langle q_{n_{\text{feat}}} | A_1 \rangle}{\partial \mathbf{r}_{31}} \end{bmatrix}$$

Figure 2.6: Sketch of the Jacobian matrix of ACDC features for atomic environment  $A_1$ . The dashed circle represents the cutoff of the environment.

dimension. The number of atoms within an environment is in reasonable settings very small. Due to this very small number of samples, the generalization error cannot be well approximated using cross-validation. We therefore need a different method to determine a regularization term that can be seen as an estimation of the noise in the features. Rank deficiencies in the Jacobian matrix correspond to linearly dependent directions with the same change in features. Each symmetry embedded in the features results in a deficiency in the Jacobian matrix. In our setting there are three deficiencies due to translational symmetries and three due to rotational. Additional deficiencies exist due to structure-specific symmetries (see Fig. 2.7), or due to directions that the features cannot represent. In the latter case it is the intersection of degenerate manifolds, manifolds of pointwise different environments but with the same features, that create these singular points with deficiencies[57]. To express a Jacobian reconstruction error, we worked on a method for finding a reasonable threshold cutting off singular values corresponding to such deficient directions.

### Machine precision for rank estimation

Since we are in control of the whole computation of the features up to the singular values, errors are introduced even due to the machine precision of the number format or to mathematical numerical approximations in the features computation. One approach for the estimation of the error due to machine precision would require an error estimation of the computation starting from the featurization of the atom's positions up to the singular value decomposition of the Jacobian matrix. The error estimations of each mathematical operation have to be chained together to one final estimation. Analytical bounds are however too loose and require probabilistic considerations to be reasonable tight for our application[58]. Least-square solvers in SciPy and NumPy use a heuristic for the threshold dependent on the machine precision of number format (for float  $\approx 1.19\text{e-}7$ , for double  $\approx 2.22\text{e-}16$ ) and the maximal dimension of the matrix  $\mathbf{J} \in \mathbb{R}^{m,n}$

$$\tau_{\text{numpy}} = \eta \cdot \max(n, m), \text{ where } \eta \text{ is number format dependent unit roundoff.} \quad (2.26)$$

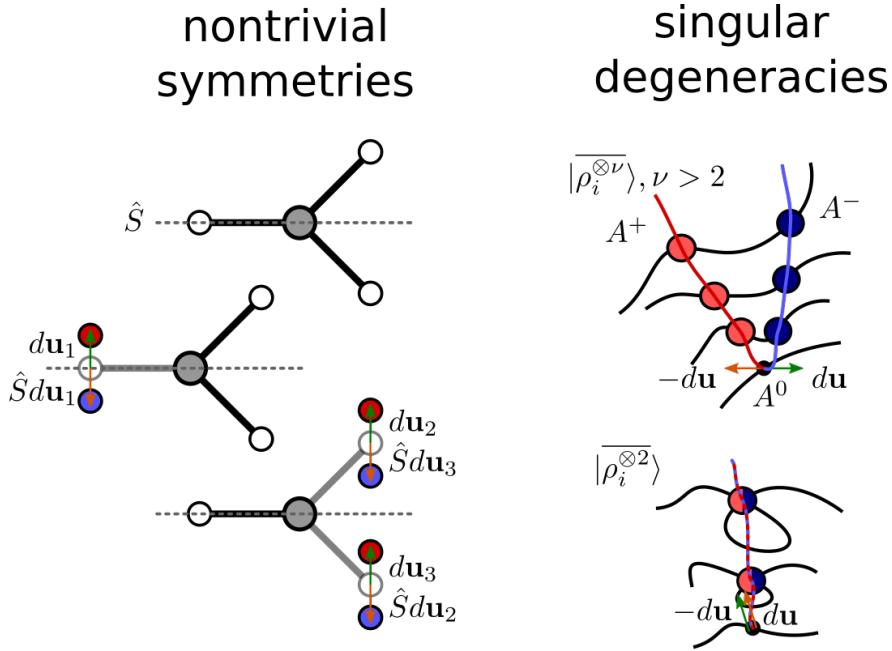


Figure 2.7: Sketch of nontrivial rank deficiencies in the Jacobian matrix in case of nontrivial symmetries or singular degeneracies. Adapted from Ref. [57].

For most cases of interest, the heuristic gives reasonable thresholds, see Fig. 2.8a). But in cases it produces disputable thresholds, where the singular values are close to the threshold, it is hard to justify that the term  $\max(n, m)$  is the correct prefactor for features with arbitrary hyperparameters, as for example in the case of low number of basis functions in Fig. 2.8b).

### Signal-noise separation for rank estimation

Another approach assumes that the singular values corresponding to signal and to noise are well-separable. This characteristic of well-separability can be expressed into different conditions depending on the context. In the case of independent component analysis (ICA) it is that the noise is statistically independent from the signal. For rank estimation the approach of Ref. [59] assumes that the singular values have different orders of magnitude. The rank  $\hat{d}$  maximizes the likelihood of two Gaussians  $\mathcal{N}(0, \sigma_{\text{signal}})$  and  $\mathcal{N}(0, \sigma_{\text{noise}})$  describing the singular value probability density function. By optimizing their standard deviation  $\sigma_{\text{signal}}$  and  $\sigma_{\text{noise}}$  the estimated rank is

$$\hat{d} = \operatorname{argmin}_{1 \leq d \leq n} \left( \frac{d}{n} \log(\sigma_{\text{signal}}^2) + \frac{n-d}{n} \log(\sigma_{\text{noise}}^2) \right). \quad (2.27)$$

In cases of sufficient number of basis function, no radial scaling and reasonable smearing sigma as in Fig. 2.8a) a clear separation of the singular values can be achieved. Such constraints are not for all cases of interest fulfilled and subsequently this method starts to fail in such cases Fig. 2.8b) producing arbitrary rank estimations.

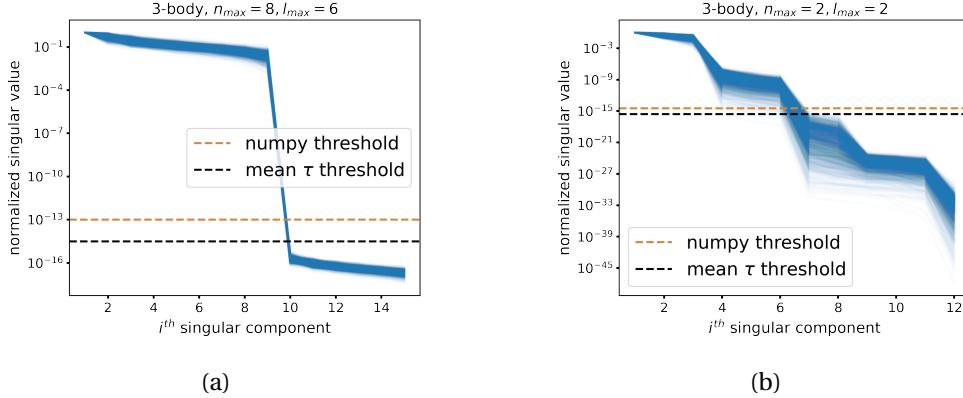


Figure 2.8: Singular value spectrum of the Jacobian of the carbon environments for several methane molecules. Different number of basis functions have been used to demonstrate the separability of the singular value spectrum for rank estimation. The singular spectra are plotted with a high transparent value so that opaque regions correspond to numerous spectra going through. We can see a) for a reasonably number of basis functions that the signal and the noise are well-separable while b) for a too small number this is note the case.

### Exploiting symmetries for rank estimation

From the last approach we can identify the core problem that the rank estimation becomes arbitrary in cases where no single clear gap exists. The question becomes how to deal with a certain amount of arbitrariness for these cases in a reasonable way while being mostly precise in cases where a single clear gap exist. To this end we exploit the null space spanned by the symmetries embedded in the features to retrieve the numerical error. Assuming that the numerical noise is isotropical in all directions  $\sigma$  we can split our Jacobian matrix in information  $\mathbf{X}$  and noise  $\sigma\mathbf{Z}$ , where  $\mathbf{Z}$  are random entries from Gaussian distribution  $[\mathbf{Z}]_{ij} \sim \mathcal{N}(0, 1)$

$$\mathbf{J} = \mathbf{X} + \mathbf{Z}\sigma. \quad (2.28)$$

By projecting with directions corresponding to symmetries  $\hat{\mathbf{u}}$  with unit length onto  $\mathbf{J}$  we obtain

$$\hat{\mathbf{u}}\mathbf{J} = \hat{\mathbf{u}}\mathbf{X} + \hat{\mathbf{u}}\mathbf{Z}\sigma = \hat{\mathbf{u}}\mathbf{Z}\sigma, \quad (2.29)$$

where the entries  $[\hat{\mathbf{u}}\mathbf{Z}]_{ij} \sim \mathcal{N}(0, 1)$  due to the unit length of  $\hat{\mathbf{u}}$ . By sampling from different directions  $\hat{\mathbf{u}}$  we can sample from the entries in  $\sigma\mathbf{Z}$ , and thereby retrieve  $\sigma$ . Given the noise estimation we can use results from random matrix theory[60] to determine the optimal threshold value

$$\tau = \lambda(\beta)\sqrt{n}\sigma, \quad \mathbf{J} \in \mathbb{R}^{m,n}, m \leq n, \beta = m/n,$$

$$\text{where } \lambda(\beta) = \sqrt{2(\beta+1) + \frac{8\beta}{(\beta+1) + \sqrt{\beta^2 + 14\beta + 1}}}, \quad (2.30)$$

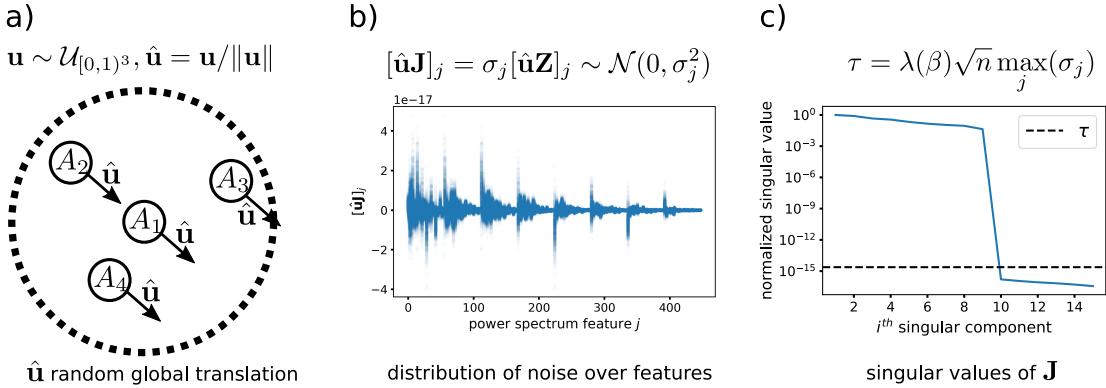


Figure 2.9: The procedure for estimating the threshold for a Jacobian matrix  $\mathbf{J}$ . a) We sample directions corresponding to global translations which are invariant to features. b) By mapping these directions on the Jacobian matrix corresponding, we sample from a distribution of the numerical error over the features. c) Then we use the maximal numerical error over the features to estimate the threshold using Eq. 2.30.

with respect to the asymptotic mean squared error between the truncated Jacobian matrix  $\mathbf{J}_\tau$  and the actual information matrix  $\mathbf{X}$ . In this asymptotic setting the matrix dimension of  $\mathbf{J}$  goes to infinity while the rank and ratio  $\beta$  stays constant[61]. While the asymptotic threshold is not necessary optimal for the finite setting, it nevertheless gives decent results as discussed in Ref. [60]. When we applied this threshold method on 3-body features we could see that the quality of the rank predictions depends highly on our noise estimation. We observed that the rank of the Jacobian matrix for 3-body features was overestimated for almost  $\approx 10\%$  of the carbon environments in the random methane dataset. The reason for this rank overestimation is that the noise is not isotropically distributed over the different features seen in Fig. 2.9b). To prevent these underestimations, we use the maximal noise over all features for the threshold computation as described in Fig. 2.9.

### 2.5.3 Example: Message-passing and connectivity

A lot of development in the community has been targeted on message-passing models due to highly accurate predictions in the low data regime[? ? ]. Recent results have established that message-passing is essentially an efficient way of decomposing the space for higher resolution[6]. It is however not clear what the effect of higher number of interactions in message-passing has on the capacity of environment features. Nonlinear effects due to the activation functions as well as due to the message-passing are usually entangled with each other in NN architectures and have not been fully understood yet. Considering domain decomposition parallelization as it is typically used in molecular dynamics software packages[62] message-passing prohibits an embarrassingly parallel problem over domains and even enforces a nontrivial communication between the CPUs/GPUs corresponding to the domains or an increase in the cutoff of the potential resulting in less room for parallelization. Especially, regarding recent results that achieve similar accurate predictions without message-passing[?]

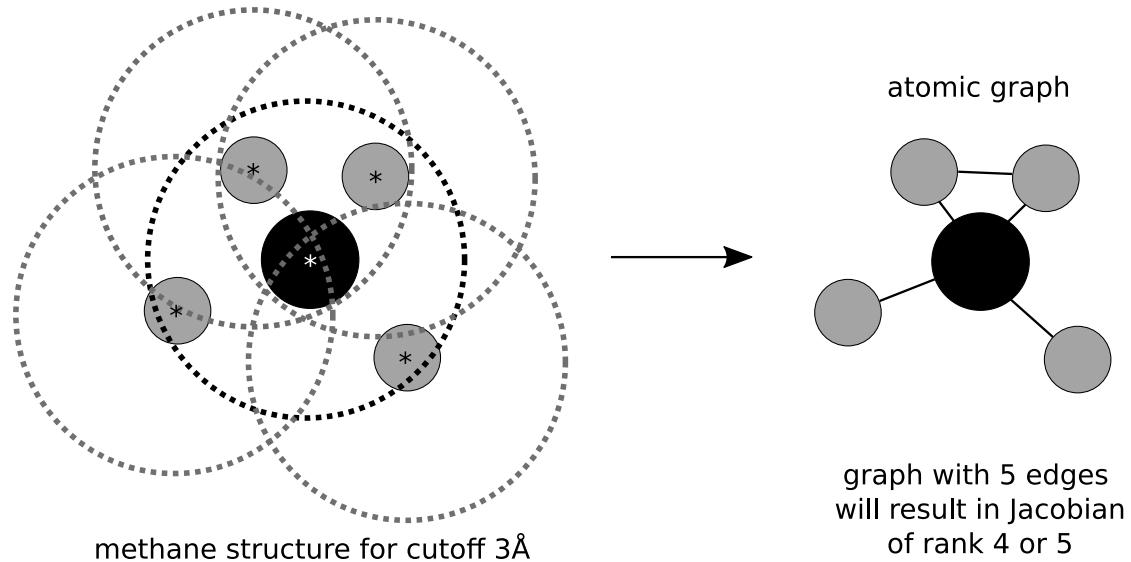


Figure 2.10: Illustration of the atomic graph for one methane structure with a cutoff of 3Å.

] or with a minimal amount of message interactions[? ], it is a valid question how much the more efficient decomposition of the geometric space due to message-passing contributes to the learning performance for learned short-range force fields. We have run preliminary experiments on a dataset with randomly displaced methane structures[?] using 2-body message-passing features[37] analysing the rank of the Jacobian matrix with respect to the number of interactions  $m - 1$

$$\langle n00 | \rho_i^{\otimes([1 \leftarrow 1])^m} \rangle, \text{ 2-body message-passing features as in Ref. [6].} \quad (2.31)$$

We analysed the ranks of the Jacobians corresponding to carbon environments with the method presented in Section ???. We consider environments with a cutoff of 6 Å where all atoms are connected with each other, and 3 Å where all hydrogen atoms are connected with the carbon atom, but not necessary with all other hydrogen atoms. It can be observed in Fig. 2.9 that the rank does not increase after one interaction for any significant amount of the carbon environments. The maximum rank correlates well with the total number of edges in the graph with respect to the cutoff seen in Fig. 2.9. A simple explanation for this observation is that each edge can be reached two moves from the carbon environment. The correlation of the number of edges with the rank can be explained by the fact that we only deal with a finite system of 5 atoms in total. The number of edges grows as  $n$  over 2 while the largest possible rank grows as  $3n - 6$  due to symmetries. Thus for the methane dataset with a maximal number of edges 10 and a maximal rank of 9 there exist always one edge that adds linearly dependent information. Therefore the Jacobian rank must be within  $[n_{\text{edges}} - 1, n_{\text{edges}}]$  for all points in this dataset. We expect that for larger or periodic structures the rank will increase also after one interaction, as in this case every passed message can reach the carbon atom within two steps

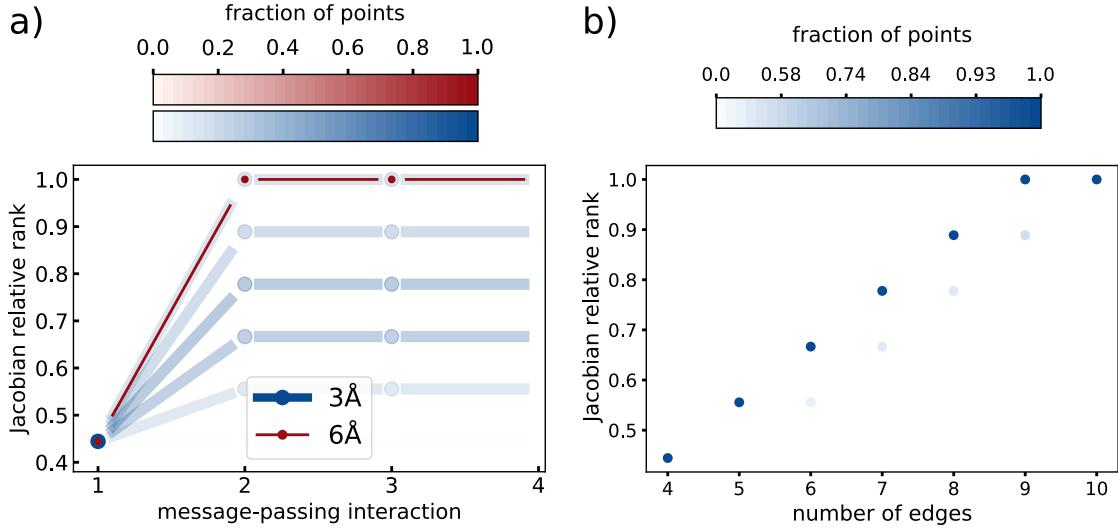


Figure 2.11: Rank estimations for the Jacobians of 2-body message-passing features for the carbon environments: a) the ranks with respect to increasing number of interactions, b) the ranks with respect to the number edges in the atomic graph for a cutoff of 3 Å. The ranks are scaled such that the maximum possible rank  $5 \cdot 3 - 6 = 9$  is normalized to 1.0.

#### 2.5.4 Local stiffness [optional]

Even in the case of no degeneracies we want to express how stable a mapping from one feature space to another is in general. We want to differ the error between a mapping which changes in each point dramatically and a mapping which is nearly constant over the whole dataset. One way to approach this problem is to use higher-order derivatives extending the range with each order. However, already computing second-order derivatives is a computationally demanding task, higher-orders are not even feasible. We therefore developed a similar approach as the Tikhonov regularization. Instead of penalizing the norm of the weight matrix, we penalize the change of the weight matrix between two points.

$$\begin{aligned} \ell(X_1, \dots, X_n, Y_1, \dots, Y_n) &= \sum_i \|X_i W_i - Y_i\|^2 \\ &+ \beta \sum_{i \neq j} \frac{1}{d(\mathbf{x}_i, \mathbf{x}_j) + 1} \|W_i - W_j\|^2 \end{aligned} \quad (2.32)$$

An unstable map with significant changes in the weight matrices between points will be penalized and thus be pushed to more stable mapping with a larger error in the optimization. In contrast, an identity map can be kept constant over the whole dataset and is thereby not affected by the penalization term. This term is closely related to a stretching energy term in the optimization of elastic maps[63]. One can see for the toy example in Figure 2.12c) mapping  $x$  to 1 that by increasing  $\beta$  the mapping starts to break at the degenerate point at  $x = 0$ . In Figure 2.12b) the 1-features and  $0.5 \sin(x)$  are compared, both reconstructing the  $x$ -features by optimizing the loss function described in Eq. 2.32. We can see that initially 1 to  $x$

## Chapter 2. Measures of information capacity

---

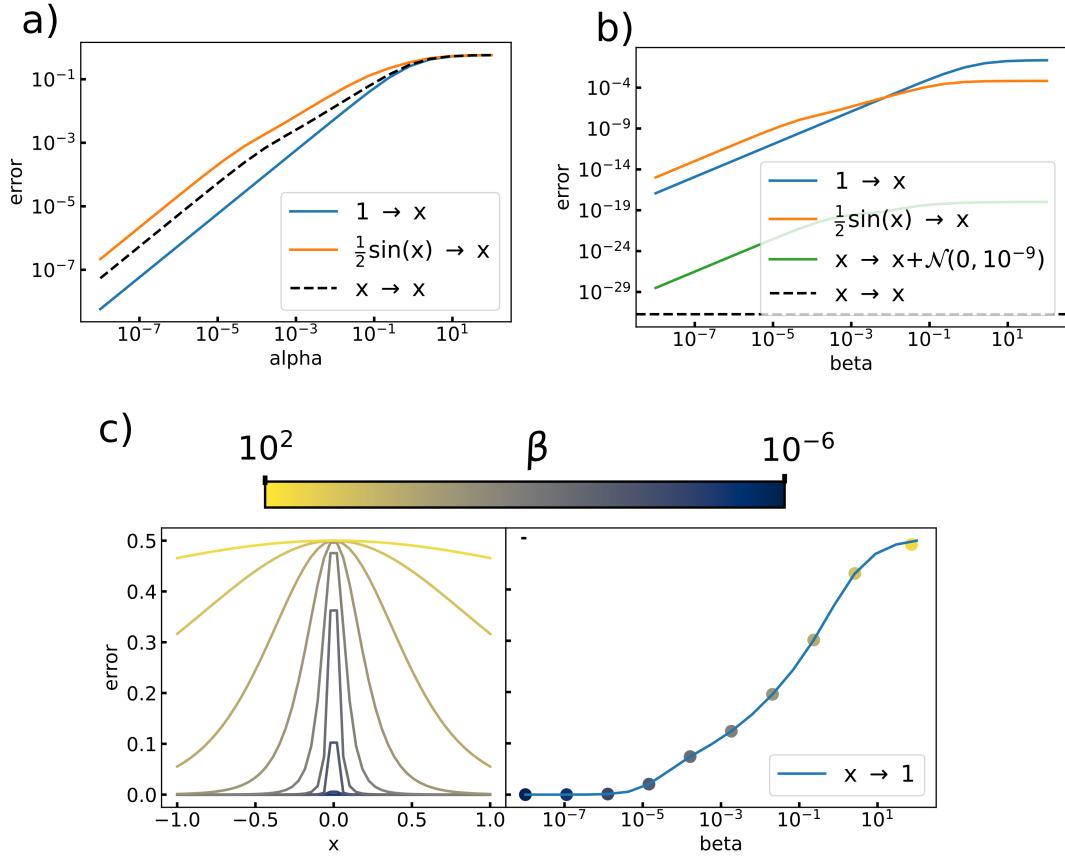


Figure 2.12: The regression error for the toy dataset  $\{x \in [-1, 1]\}$  for different features over the spectrum of a) ridge regularization and b) stretch regularization as described in Eq. 2.32. c) The error reconstructing the 1-features from  $x$ -features for increasing  $\beta$  (left) samplewise (right) and averaged over the whole dataset. It can be seen that the error starts to spread starting from the degeneracy at  $x = 0$ .

produces a smaller errors, since its eigenvalue spectrum is more uniform than  $0.5\sin(x)$ , while the mapping  $0.5\sin(x)$  to  $x$  gives smaller error for large  $\beta$  values, because the feature mapping  $0.5\sin(x)$  to  $x$  requires fewer changes of the weights over the dataset than 1 to  $x$  and is thus more stable. Increasing  $\beta$  even more results in the mapping constant over the whole dataset which recovers the global mapping.

While the loss in Eq. 2.32 is not jointly convex over all  $W_i$ , it is convex in  $W_i$  separately keeping all  $W_j$  for  $i \neq j$  fixed[64]. We can therefore solve it iteratively until convergence by optimizing in every iteration each weight matrix separately

$$W_i = \left( X_i^T X_i + \beta \sum_{j \neq i} \frac{1}{d(\mathbf{x}_i, \mathbf{x}_j) + 1} I \right)^{-1} \left( X_i^T Y_i + \beta \sum_{j \neq i} \frac{1}{d(\mathbf{x}_i, \mathbf{x}_j) + 1} W_j \right) \quad (2.33)$$

With the distance metric  $d$ , we can induce different degrees of elasticity on the penalty. For

example by using a Gaussian kernel distances  $d_\sigma$  we can control with  $\sigma$ , how local the penalty should be.

### 2.5.5 Future directions - Restricting nonlinearities

My interpretation is that  $\ell_1$  is the effect of the nonlinearities plus incompleteness and that  $\ell_\infty$  is the effect of only incompleteness. So  $\ell_\infty - \ell_1$  should give us the effect due to nonlinearities. In my mind nonlinearities are just Taylor expansions.

Theoretically speaking there are nonlinearities that cannot be modelled by this approach, but we argue that these are negligible as potentials used in MD are always approximatable.

#### Future experiment: Effect of incompleteness

Information capacity separate the effect of nonlinearities and incompleteness

$$\ell_{v'} = \sum_{v'} \overline{\rho^1}^{v'} \rightarrow \overline{\rho^2}.$$

Then compare  $\ell_1$  with  $\ell_\infty$  (sum over  $v'$  until convergence of error). Important: Normalize each body-order separately, and check that the regularizer is very small (like in the range of a jitter to argue that the constant regularization and therefore does not constraint the transformation). Repeat result for 3-body potential  $f^3$  (like LJ for 3-body) that is well defined over the whole interval

$$\ell_{v'} = \sum_{v'} \overline{\rho^1}^{v'} \rightarrow f^3$$

To see if there is any difference the results using the GFRE.

#### Future experiment: Characterizing body-order information in CG-Nets

it is super unclear what body orders are actually learning in CG-Net. So after each iteration  $p$  in the network  $f^{(p)}$  we test regression of

$$\overline{\rho^v} \rightarrow f^{(p)}$$

to see how the body order changes. We compare CG-Nets that increase the body order by itself and CG nets that multiply by themselves.



# 3 Splining

Splining has been a popular approach in the community for optimizing the speed of lower-dimensional potentials[?] . The idea is to express a complex function piecewise with "less complex" spline functions. One can think of as describing a higher-order function with a lot of lower-order functions. The splining procedure can be split into two subproblems. Given an input to evaluate the function we need to find the corresponding constrained domain and then evaluate the spline function.

There exist different approaches in splining techniques in atomistic learning.

- splining the radial part  $R_{nl}$  or  $R_n$  (Caro)
- splining potential in invariant space

## 3.1 Grid

The choice of grid determines

### 3.1.1 Adaptive grid

-  $O(\log N)$  binary search lookup time but reduced error for same number of points (consider that a family of potentials don't have any important information at the center)

### 3.1.2 Equispaced grid

- $O(1)$  constant lookup time but not optimal error
- Uniform parallelization guarantees (not breaking out of for loop)

## 3.2 Spline function

- In principle tradeoff between higher-orders Easier fun But usually we do not have access to higher derivatives due to their complexity -

...

### 3.2.1 Cubic spline

As part my work in the first year, I participated in the development of a library designed to allow efficient computation of atomic density based descriptors, named librascal. My main contribution in the last year has been the development of efficient methods for the computation of the radial basis expansion for the SOAP descriptor. In contrast to the approaches discussed in Section ??, for librascal Gaussian typed orbital (GTO) functions are used as radial basis functions

$$R_n^{GTO}(r) = r^{n+2} \exp(-b_n r^2) N_n, \quad (3.1a)$$

$$\text{with } b_n = 1/(2\sigma_n), \quad \sigma_n = r_c \max(\sqrt{n}, 1)/n_{\max}, \quad (3.1b)$$

that allow computing analytically the integral for the radial expansion in Eq. ???. However, due to the modified spherical Bessel function  $i_l$  an analytical expansion of radial term requires the evaluation of the computationally expensive confluent hypergeometric function  ${}_1F_1$

$$\begin{aligned} \int_0^\infty R_n^{GTO}(r) \exp\left[-a(r^2 + r_{ij}^2)\right] i_l(2ar r_{ij}) dr &= N_n \pi^{\frac{3}{2}} \frac{\Gamma(\frac{n+l+3}{2})}{\Gamma(l+\frac{3}{2})} \\ (a+b)^{-\frac{n+l+3}{2}} (ar_{ij})^l \exp\left[-ar_{ij}^2\right] {}_1F_1\left(\frac{n+l+3}{2}, l+\frac{3}{2}, \frac{a^2 r_{ij}^2}{a+b}\right) &= f^{nl}(r_{ij}). \end{aligned} \quad (3.2)$$

The contribution of this function to the overall cost of the radial expansion can be seen in Fig. 3.1 ranging from 55% to 75% of the total time. The term in Eq. 3.2 can be expressed as a one dimensional function  $f^{nl}$  of the distance  $r_{ij}$  only dependent on the radial and angular numbers  $n$  and  $l$ . The computation of this term can be bypassed by interpolating the function for each  $l, n$  term on the grid  $[0, r_c]$ . An implementation of cubic spline interpolation for the term has been part of my work for the first year. A spline interpolation splits the targeted range  $[0, r_c]$  into intervals, each interval being interpolated by a polynomial with same boundary conditions. Let  $\{r_k\}_{k=1}^{M+1}$  be the set of  $M+1$  boundary points in the interval  $[0, r_c]$ , then cubic spline defines polynomials  $p_k(r) = A_k + B_k r + C_k r^2 + D_k r^3$  on the interval  $[0, 1]$  with the

### 3.2 Spline function

---

boundary conditions

$$p_k^{nl}(0) = f^{nl}(r_k) \text{ and } p_k^{nl}(1) = f^{nl}(r_{k+1}) \text{ for } k = 1, \dots, M, \quad (3.3a)$$

$$p_k^{nl}(0) = p_{k-1}^{nl}(1) \text{ and } p_k^{nl'}(0) = p_{k-1}^{nl'}(1) \text{ for } k = 2, \dots, M, \quad (3.3b)$$

$$p_k^{nl}(1) = p_{k+1}^{nl}(0) \text{ and } p_k^{nl'}(1) = p_{k+1}^{nl'}(0) \text{ for } k = 1, \dots, M-1, \quad (3.3c)$$

$$p_1^{nl''}(0) = 0 \text{ and } p_M^{nl''}(1) = 0. \quad (3.3d)$$

The boundary conditions ensure the smoothness at the boundary points. A linear system of equations can be formed from the  $4M$  conditions in Eqs. 3.3b, 3.3c, 3.3d and their evaluation in Eq. 3.3a by rearranging them[65] resulting in

$$A_k = f(r_k), \quad C_k = 3(f^{nl}(r_{k+1}) - f^{nl}(r_k)) - 2B_k - B_{k+1}, \quad (3.4a)$$

$$D_k = 2(f^{nl}(r_k) - f^{nl}(r_{k+1})) + B_k + B_{k+1}, \quad (3.4b)$$

$$\begin{pmatrix} 2 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 2 \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_{M-1} \\ B_M \end{pmatrix} = \begin{pmatrix} 3(f^{nl}(r_2) - f^{nl}(r_1)) \\ 3(f^{nl}(r_3) - f^{nl}(r_1)) \\ \vdots \\ 3(f^{nl}(r_M) - f^{nl}(r_{M-2})) \\ 3(f^{nl}(r_M) - f^{nl}(r_{M-1})) \end{pmatrix}. \quad (3.4c)$$

A linear systems forming a tridiagonal matrix can be solved in linear time with time complexity  $O(2M)$  by iterating two times through the matrix following a Gaussian elimination scheme.

Commonly, for interpolators the grid size is adaptively changed until it is below an error bound given by the user as input. The error can be estimated by sampling points in the intervals  $[r_k, r_{k+1}]$  and comparing the results of  $f^{nl}$  with  $p^{nl}$ . During implementation it has been shown that conditioning the error on a relative and absolute error has been most robust in terms of interpolation accuracy and convergence of the grid size. This means that the boundary points are extended until one of the two estimated errors lies under the given error.

A point  $r$  can be evaluated by first searching the corresponding boundary point  $r_k$  for which  $r \in [r_k, r_{k+1}]$ , and then mapping  $[r_k, r_{k+1}]$  to the polynomial interval  $[0, 1]$  with  $p_k((r - r_k)/(r_{k+1} - r_k))$ . The design choice mainly affecting the efficiency of the cubic spline interpolator was the choice of the grid. For an arbitrary grid with  $M+1$  points, the optimal complexity for searching the boundary point  $r_k$  is  $O(\log(M+1))$  with a binary search. On the other hand for uniform grids the search time can be reduced to  $O(1)$  by

$$k = \min(M, \lfloor (r - r_1)\Delta \rfloor + 1), \quad \Delta = (M+1)/(r_{M+1} - r_1). \quad (3.5)$$

For each  $(n, l)$  pair a cubic spline interpolator is computed. The effect of the speed-up for a reasonable range of  $(n, l)$  pairs for  $10k$  evaluations ranges from 6 up to 18 times as seen in

### Chapter 3. Splining

---

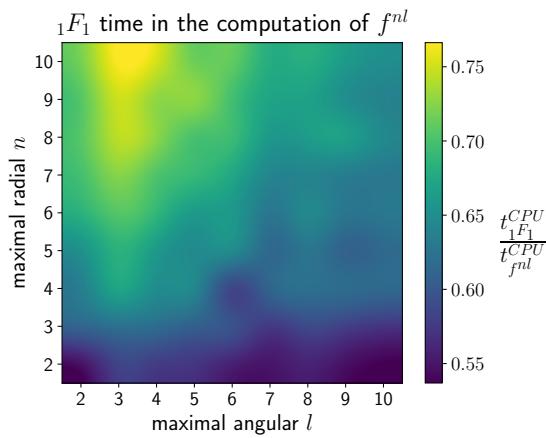


Figure 3.1

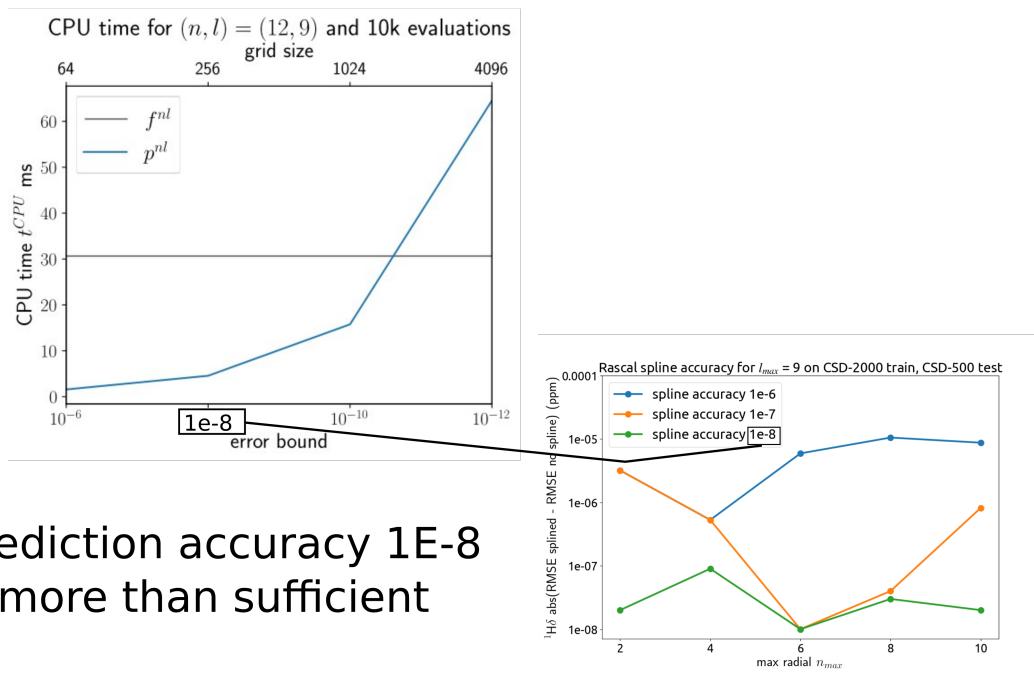
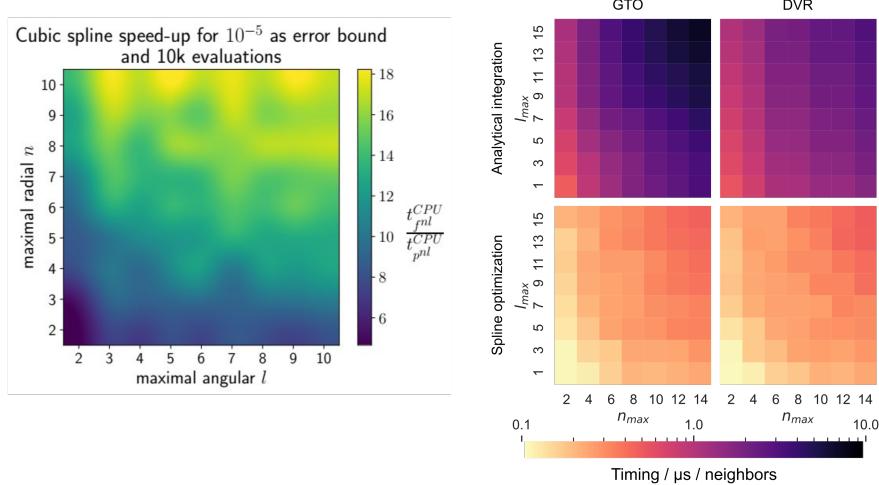


Figure 3.2: Effect of accuracy on prediction error

### 3.2 Spline function



We were able to reach a speed of up 6-18

Musil, F., Veit, M., Goscinski, A., Fraux, G., Willatt, M. J., Stricker, M., ... & Ceriotti, M. (2021). Efficient implementation of atom-density representations. *The Journal of Chemical Physics*, 154(11), 114109.

Figure 3.3: Speedups

Fig. ???. The exact speed up depends on the exact error bound as seen in Fig ???. By using the same grid for each  $(n, l)$  pair, the matrix solver for tridiagonal matrices and the interpolation of a point  $r$  for all  $(n, l)$  can be vectorized. Vectorization allows the parallel execution of the same instruction on multiple  $(n, l)$  pairs. These instructions are commonly known as single instruction, multiple data (SIMD) and their exact effect on the efficiency depend on the supported instruction set of the CPU architecture and the compiler. The exact effect of the SIMD instructions for the implemented vectorized operations is a tedious task, since other parts of the code also rely on SIMD instructions[66]. Therefore we limit our analysis to the effect of the SIMD optimization induced by the linear algebra library Eigen[67] used for the vectorized operations. The optimization can be deactivated with the compiler option EIGEN\_DONT\_VECTORIZE. A deactivation did not effect the performance significantly as seen in Fig. ???. A more exact analysis of the effect of the vectorization as in Ref. [66] is part of future work.

In summary, due to the polynomial evaluation of cubic spline on a equispaced grid we gain speed-ups for the evaluation of the radial contribution ranging from 5.5 to 18 for commonly used radial and angular numbers.

Boundary conditions: Three boundary conditions for Cubic spline, check with Hermite - natural boundary



# 4 Symmetry-preserving basis optimization

The splining techniques presented in the last chapter open the door for optimizing the basis function without any additional cost at evaluation time (e.g. when running molecular dynamics). We present in this chapter several optimization methods for the basis preserving symmetries that can be used in combination with splining that can be summarized by

$$\sum_n^{n_{\max}} U_{qn}^{\lambda}(\mathcal{D}) c_{n\lambda} = c_{q\lambda}. \quad (4.1)$$

The linear transformation  $\mathbf{U}$  computed from dataset  $\mathcal{D}$  recombines the radial expansion coefficients for a fixed basis  $\{b_n^{\lambda}\}_{n=1}^M$  resulting in optimized coefficients  $c_{q\lambda}$ . The idea is that the initial basis is chosen to cover a large hypothesis space  $\mathcal{H}$ , the optimized basis is optimal wrt. to a dataset  $\mathcal{D} \subset \mathcal{H}$ . The optimized coefficients can be then truncated at  $M'$  with the hope that  $M' \ll M$ . Such that the featurization is much lower dimensional space reducing the number of computational steps when predicting a property. Constraining  $\mathbf{U}$  to be linear allows the use of the splining as the targeted function is not changed

$$\sum_n U_{qn}^{\lambda} c_{n\lambda} R_{n\lambda}(r) = c_{q\lambda} R_{q\lambda}(r) \quad (4.2)$$

Then the splining as described in Eq. ?? is constructed for the radial coefficients  $c_{q\lambda}$  such that the transformation  $\mathbf{U}$  can be bypassed during evaluation.

TODO do we need orthogonality?

## 4.1 Closed-form solutions

We dicuss here optimizations that have closed-form solution for the computation of  $U$ . Such solutions are typically faster to compute but also less accurate.

Closed-solution are usually more efficient in computation time, but often only improve the basis by linear transformation or fixed subset of nonlinear transformation.

We do unsupervised optimization with PCA

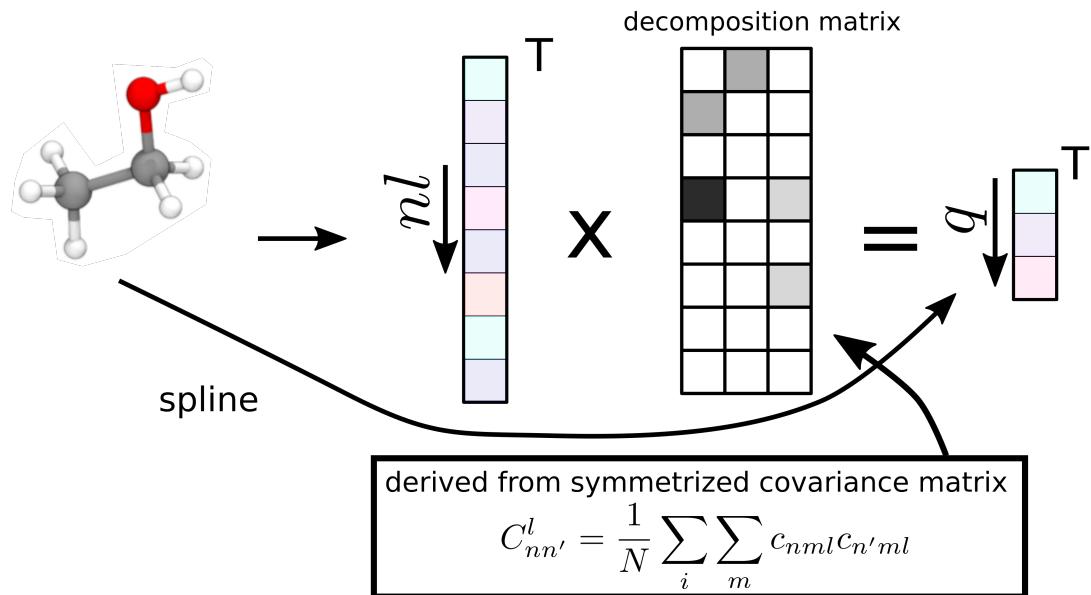


Figure 4.1: A schematic showing the spline trick. TODO use poster spline trick as basis maybe add more description about the optimization.

### 4.1.1 Unsupervised optimization

Principal component analysis has been proposed to compute the data-driven contractions of equivariant features that represent in the most informative way the variability of a dataset as part of the N-body iterative contraction of equivariant (NICE) frameworks. [68] We propose to apply this procedure on the first-order equivariants – that correspond to the density coefficients – as a mean to determine a data-driven radial basis. Keeping different chemical species separate, this amounts to computing the rotationally invariant covariance matrix (see SI)

$$C_{nn'}^{al} = \frac{1}{N} \sum_i \sum_m \langle anlm | \rho_i \rangle \langle \rho_i | an'lm \rangle, \quad (4.3)$$

where the summation over  $m$  ensures that the covariance is independent of the orientation of structures in the dataset. For each  $(a, l)$  channel, one diagonalizes  $\mathbf{C}^{al} = \mathbf{U}^{al} \mathbf{\Lambda}^{al} (\mathbf{U}^{al})^T$ , and computes the optimal coefficients

$$(4.4)$$

Note that we compute  $\mathbf{C}^{al}$  without centering the density coefficients. For  $l > 0$ , the mean ought to be zero by symmetry (although it might not be for a finite dataset), and even for the totally symmetric,  $l = 0$  terms, density correlation features are usually computed in a way that is more consistent with the use of non-centered features.

The number of contracted numerical coefficients  $q_{\max}$  can be chosen inspecting the eigenvalues  $\Lambda_q^{al}$ . At first, it might appear that in order to evaluate the contracted basis one has to compute the full set of  $n_{\max}$  coefficients, and this is how the idea was applied in Ref. 68. When combining Eq. (??) with Eq. (??), however, one sees that the contracted coefficients can be evaluated directly

$$\langle aqlm; \text{opt} | \rho_i \rangle = \sum_j \delta_{aa_j} \langle aql; \text{opt} | r_{ji}; g \rangle \langle lm | \hat{\mathbf{r}}_{ji} \rangle, \quad (4.5)$$

using the contracted radial integrals

$$\langle aql; \text{opt} | r; g \rangle = \sum_n U_{qn}^{al} \langle nl | r; g \rangle, \quad (4.6)$$

, and then evaluated at exactly the same cost as for a spline approximation of the radial integrals of a primitive basis of size  $q_{\max}$ . Splining does not affect the equivariant behavior of the atom-density features, and introduces minute discrepancies relative to the analytical basis that do not affect the quality of the resulting models.

Even though Eq. (4.3) is defined separately for different species  $a$ , it is also possible to compute cross-correlations between different elemental channels, defining

$$C_{an;a'n'}^l = \frac{1}{N} \sum_i \sum_m \langle anlm | \rho_i \rangle \langle \rho_i | a'n'l'm \rangle, \quad (4.7)$$

## Chapter 4. Symmetry-preserving basis optimization

---

as done in the NICE framework[68] following ideas proposed in Ref. 49, resulting in coefficients that combine information on multiple species

$$\langle qlm; \text{opt} | \rho_i \rangle = \sum_n U_{q;an}^l \langle anlm | \rho_i \rangle, \quad (4.8)$$

similar in spirit to the alchemical contraction discussed in Ref. 53. It is worth noting that although the NICE code[?] contains the infrastructure to compute these contractions *as a post-processing of the primitive basis*, the implementation we propose in LIBRASCAL computes the contracted coefficients directly. However, it only implements the less information-efficient separate  $(a, n)$ -PCA strategy. An implementation that evaluates directly the combined contraction would incur an overhead because every neighbor would contribute to every  $q$  channel irrespective of their species:

(4.9)

Given however that the cost of evaluating the density coefficients is usually a small part of the calculation of density-correlation features[? ?], we expect that this approach should be in general preferable compared to the calculation of a large primitive basis, and to a two-step procedure in which element-wise optimal functions are further contracted into mixed-element coefficients.

### Example: Collective variable optimization

We can see that we can retrieve similar quality CV as in the BaTiO<sub>3</sub> paper from a snapshot. Applying PCA is not a new thing, as it is a chicken-egg problem.

#### 4.1.2 Supervised optimization

For a given number of radial functions, and a target data set, the data-driven contracted basis (??) provides the most efficient description of the atom-centred density in terms of the fraction of the retained variance. The most effective variance-preserving compression however does not guarantee that the features are the most effective to predict a given target property. In fact, it has already been shown that SOAP features tend to emphasize correlations between atoms that are far from the atomic center, which can lead to a counter-intuitive degradation of the model accuracy with increasing cutoff radius[51, 53]. This effect can be contrasted by introducing a radial scaling[53, 69] that de-emphasizes the magnitude of the atom density in the region far from the central atom. By applying this scaling – or other analogous tweaks[?] – to the atom density before it is expanded in the primitive basis, one ensures that the optimal basis is also built with a similar focus on the structural features that contribute more strongly to the target property. In other terms, the information-optimal basis set we introduce here can be combined with a heuristic or data-driven optimization of the underlying density representation, to reflect the scale and resolution of the target property.

Another possibility is to extend the scheme to incorporate a supervised target  $y_i$  in the selection of the optimal basis using principal covariates regression (PCovR) [70, 71]. PCovR is a simple linear scheme that can be tuned to provide a projection of features to a low-dimensional latent space that combines an optimal variance compression target with that of providing an accurate linear approximation of the desired target property. Since  $l > 0$  contributions of the features have zero mean, the optimization problem can be combined with a supervised component only for  $l = 0$ , and yields an optimal basis

$$\langle r || aq0; \text{opt}; \gamma \rangle = \sum_n U_{qn}^{a0;\gamma} \langle r || n0 \rangle, \quad (4.10)$$

which is a special case of Eq. (??) for  $l = 0$ , where  $U_{qn}^{a0;\gamma}$  is obtained as the orthogonalized PCovR projector, as discussed in Refs. 70, 71, using a mixing parameter  $\gamma$ , that determines how strong the emphasis of the optimization should be on minimizing the residual variance or the error in regressing the target.

$$\begin{aligned} \langle Q; nlk | \overline{\rho_i^{\otimes(v+1)}; \sigma; \lambda\mu} \rangle \propto \sum_m \langle n | \overline{\rho_i^{\otimes 1}; lm} \rangle \times \\ \langle Q | \overline{\rho_i^{\otimes v}; \sigma((-1)^{l+k+\lambda}); k(\mu-m)} \rangle \langle lm; k(\mu-m) | \lambda\mu \rangle, \end{aligned} \quad (4.11)$$

using Clebsch-Gordan coefficients  $\langle lm; l'm' | l''m'' \rangle$  in an expression analogous to the sum of angular momenta. The  $v = 1$  equivariants are nothing but the density coefficients

$$\langle n | \overline{\rho_i^{\otimes 1}; \sigma; lm} \rangle = \delta_{\sigma 1} \langle nlm | \rho_i \rangle^*, \quad (4.12)$$

and one can compute invariant descriptors by retaining only the  $\langle Q | \overline{\rho_i^{\otimes v}; 1; 00} \rangle$  terms, using the other components only as computational intermediates.

## 4.2 Higher-order information

Most NN structures use the approach to find a radial basis that is simple to evaluate to then let the NN optimize the basis. Due to the flexible optimization space of NN, they can achieve quite good accuracies which would not have been able to achieve with shallow methods[37].

## 4.3 Future directions

### 4.3.1 Hierarchical optimization

Optimizing first the chemical part and then the radial part, because optimizing both seems not to work well

In the work of Natasha & Guillaume the optimization of radial and chemical due to the huge

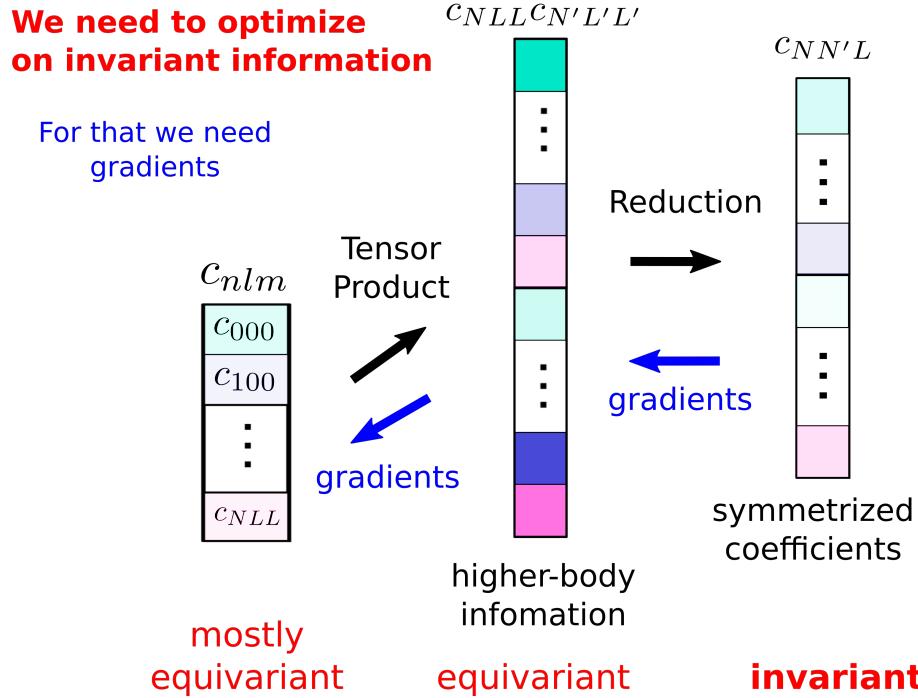


Figure 4.2: Nonlinear the equivariant optimization by gradient descent

optimization space. It is well known that deep NN have an equivalent functional form with one layer, but the learning performance is dramatically better in the former one. Both the radial and species channel increase the feature size by a multiplicative factor and thereby also the weigh matrix size.

### 4.3.2 Radial dependent smoothness

make sigma dependent on r, because we see that it works well for GTOs

# 5 Implementation of machine learning interatomic potentials

Interatomic potential have been long time used to approximate the potential energy in classical MD simulation by a separation into separate body-order terms.

$$H(\mathbf{p}, \mathbf{q}) = \frac{\mathbf{p}^2}{2m} + V(\mathbf{q}), \quad (5.1a)$$

$$V(\mathbf{q}) = \sum_{i=1}^N V_1(\mathbf{q}_i) + \sum_{i,j=1}^N V_2(\mathbf{q}_i, \mathbf{q}_j) + \sum_{i,j,k=1}^N V_3(\mathbf{q}_i, \mathbf{q}_j, \mathbf{q}_k) + \dots \quad (5.1b)$$

Due to limitation of hardware the first succesful approaches to interatomic potentials have been carefully handcrafted models designed for a specific materials in a specific phases.[72, 73]. With increased hardware power the use of data-driven models emerged allowing a more automatized model construction. These fall under the name machine learning interatomic potentials (MLIP)[74, 75].

The current ecosystem of ML packages is extremely abundant, that one could question why there is a need for new ML packages for the domain of material science at all. But most of the developed packages are targeted for users that do not share the same requirements as in the domain of material science and therefore cannot be directly embedded for the development of MLIP. This is due to the fact that the symmetrization of features based on point clouds in 3D and the inference of gradients are not requirements that are shared with many other domains that deploy ML models. To reduce the development and maintainance costs it is necessary to find points in ones method where already established software packages can be embedded in. For that a mathematical and algorithmic understanding of the landscape of methods and proficient skills in software development are needed which makes it such challenging problem.

In this chapter I discuss my contributions to the software ecosystem that facilitates the development of MLIP. This covers my contributions to the package `librascal` for the computation of the featurization of atomic structures and building a model, `scikit-matter` the ML method packages, implementation of an interface to LAMMPS that enabled study of

ferroelectric phase transitions in barium titanate[76] as well as the transport properties of lithium ortho-thiophosphate[77]. Furthermore, this includes contributions to `equisolve` and `metatensor` that allow a more modular approach to build MLIP.

## 5.1 Implementation of gradients in kernel models

We do not cover an introduction to kernel methods, but only discuss the specificities that arise when extending them for gradients. For a comprehensive introduction to kernel models please refer to Ref. [78].

For a given kernel  $k$  from a set of training samples  $\{c_n \in \mathbb{R}^d\}_{t=1}^N$  and targets  $\{c_n \in \mathbb{R}\}_{t=1}^N$  a set of weights  $\{\alpha_n \in \mathbb{R}\}_{t=1}^N$  can be retrieved as solution of the minimization problem

$$\min_{\alpha} \sum_{n'}^N \|y_{n'} - \sum_n \alpha_n k(c_n, c_{n'})\|^2 \quad (5.2a)$$

that subsequently can be used to evaluate an arbitrary point  $i$  by the relationship

$$y_n = \sum_n \alpha_n k(c_n, c_i). \quad (5.3)$$

The solution of the problem in matrix form can be expressed as

$$\alpha = (\mathbf{K} + \Lambda)^{-1} \mathbf{y} \quad (5.4)$$

Now to include the gradients wrt. to the atomic position  $\partial \mathbf{r}_k$  of atom  $k$  into the picture. We note that the training points used to construct the kernel are independent with respect to the gradients, therefore we use the notation  $k_n(c_i)$  to denote  $k(c_n, c_i)$  for easier readability of the derivatives

$$\frac{\partial E_i}{\partial \mathbf{r}_k} = \sum_n^N \frac{\partial \alpha_n k_n(\mathbf{c}_i)}{\partial \mathbf{r}_k} \quad (5.5a)$$

$$= \sum_n^N \alpha_n \frac{k_n(\mathbf{c}_i)}{\partial \mathbf{c}_i} \frac{\partial \mathbf{c}_i}{\partial \mathbf{r}_k} \quad (5.5b)$$

$$= \frac{\partial \mathbf{c}_i}{\partial \mathbf{r}_k} \sum_n^N \alpha_n \frac{k_n(\mathbf{c}_i)}{\partial \mathbf{c}_i}. \quad (5.5c)$$

Note that the last step is essential to reduce the number of iterations from  $O(dN)$  to  $O(d+N)$ . Since gradients make the evaluation of the kernel matrix computationally costly, low-rank approximation techniques have become essential early on to reduce the memory intensive usage during training. The subset of regressor method[79] has been a popular low-rank estimation used in the MLIP packages QUIP[80]. The core idea is to project the data points on

## 5.2 Interfacing with molecular dynamics packages

---

a subset of the  $M$  pseudo points.

$$\mathbf{K} = \mathbf{K}_{MM} + \mathbf{K}_{MN}\Lambda^{-2}\mathbf{K}_{MN}^T \quad (5.6a)$$

Note that to compute one kernel entry for two structures with each  $N_i$  atoms it requires  $N_i^2$  evaluations

$$k^A(A_i, A_{i'}) = \sum_{j \in A_i}^{N_i} \sum_{j' \in A_{i'}}^{N_{i'}} k(\mathbf{c}_j, \mathbf{c}_{j'}). \quad (5.7a)$$

The low-rank approximation contributes greatly in the reduction of the computation of these kernel entries as it projects the structural features on single  $M$  environmental features resulting in a linear scaling of one kernel entry. Since the number of atoms in structures is a substantial quantity and the fact that structures typically contain redundant environments, the rank can be reduced quite significantly making this approach indispensable.

One disadvantage with `librascal` was the entanglement of the featurization and the model building to the library. This was required as the construction of the kernel matrix requires understanding by the model building tool of the decomposition of the target property into local contributions Eq. 1.4 as well as what its gradients are, domain-agnostic packages as `scikit-learn` are not suitable for a direct application in this case. The software package `metatensor` allows to attribute these structural characteristics to the object itself as metadata and offers numpy-like data manipulations that take advantage of the metadata. This allowed a disentanglement of the featurization that has reimplemented in a new package `rascaline` and model construction that has been moved to `equisolve`. Part of my contribution was to participate in the development of `metatensor` as well as heavily developing on the initial design of `equisolve` contributing module of shallow methods including standardizer, linear and the above-presented kernel model as well as the initial designs for neural network models based on the data formats created in `metatensor`.

## 5.2 Interfacing with molecular dynamics packages

In molecular dynamics the separation of the computation of the potential energy into a separate module has been established approach[62, 81, 82, 83]. As computation of the potential energy only requires the atomic positions and species and as result returns the energy with its gradients, it is a logical point to separate. An advantage for such a separation is to avoid a reimplementation of established thermo- and barostats as well as time integrator. While the implementation of these are in principle simple, a robust implementation that covers important corner cases still requires a longer consideration which makes it a time consuming task. Another advantage is the fact that long developed MD software support different parallelizations of the potential energy for interatomic potentials. The fact that interatomic short-range potentials separate the total energy into contributions of local environments, allows a decomposition of the cell into smaller domains. The computation of the potential

energy for each of these domains can be distributed among multiple processors or machines and then be using message passing interface (MPI). In the study of barium titanite in Ref. [76] this parallelization allowed us to study the effect of long-range dielectric correlations on the predictions of the Curie temperature. The domain decomposition requires particular considerations in the implementation of the gradients. The force computation for an interatomic potential can be computed as

$$\frac{\partial E_A}{\partial \mathbf{r}_k} = \sum_{i \in A} \frac{\partial E_i}{\mathbf{r}_k} = \sum_{i \in A} \sum_{j \in A_i} \frac{\partial E_i}{\mathbf{r}_{ji}} \frac{\mathbf{r}_{ji}}{\mathbf{r}_k}, \text{ where } \frac{\mathbf{r}_{ji}}{\mathbf{r}_k} = \begin{cases} 1, & k = i \\ -1, & k = j \\ 0, & \text{else.} \end{cases} \quad (5.8)$$

It can be seen that the forces at position  $i$  depend on the partial forces wrt. to energy  $i$  and all its neighbors  $j \in A_i$ . Depending on memory storage of partial forces this case needs to be handled with care. In particular considering the fact that MD software as LAMMPS offers the option to allow MPI communication between the domains with the *newton\_pair* option. Keeping the option on prevents redundant computations of the forces but requires more MPI communication as the partial forces need to be communicated. In *librascal* if atom  $j$  is a periodic neighbor, the access on atom  $j$  remapped to corresponding atom in the box. The MPI communication requires to differ between periodic neighbors and ghost atoms that are part of another domain. One can simplify this complexity storing the partial forces  $\partial E_i / \partial \mathbf{r}_{ij}$  and  $\partial E_j / \partial \mathbf{r}_{ij}$  for redundantly for atom  $i$  and  $j$ . This is redundant due to Newton's third law  $\partial E_j / \partial \mathbf{r}_{ij} = -\partial E_i / \partial \mathbf{r}_{ji}$ . While being redundant it does not change asymptotic scaling of the memory requirements for the forces.

### 5.3 A metadynamic software framework with LAMMPS, PLUMED and i-PI

To study phase transitions of barium titanite in Ref. [76] we needed to accelerate the sampling of the transition. One common technique that we use is metadynamics that adds a bias potential to the simulation that is later normalized out in the calculation of the free energy. While the forces of the MLIP were computed with LAMMPS and the forces of the bias potential were computed with PLUMED[84]. To consolidate both forces for the metadynamics the software-package i-PI was used, that implemented a custom protocol to each of these MD packages to allow communication of the forces to a python interface. A schematic of the interwork between the software packages can be seen in Fig. 5.1.

As CV for the metadynamics  $l = 1$  components of the spherical expansion coefficients computed with *librascal* were used. My implementation of the cubic spline helped to speed up the computation of the bias term. Since only the expansion coefficients centered for the oxygen neighbors centered on the titanite was sufficient for the CV, I further implemented an option to selectively compute the partial gradients for certain species. Note that a selectional

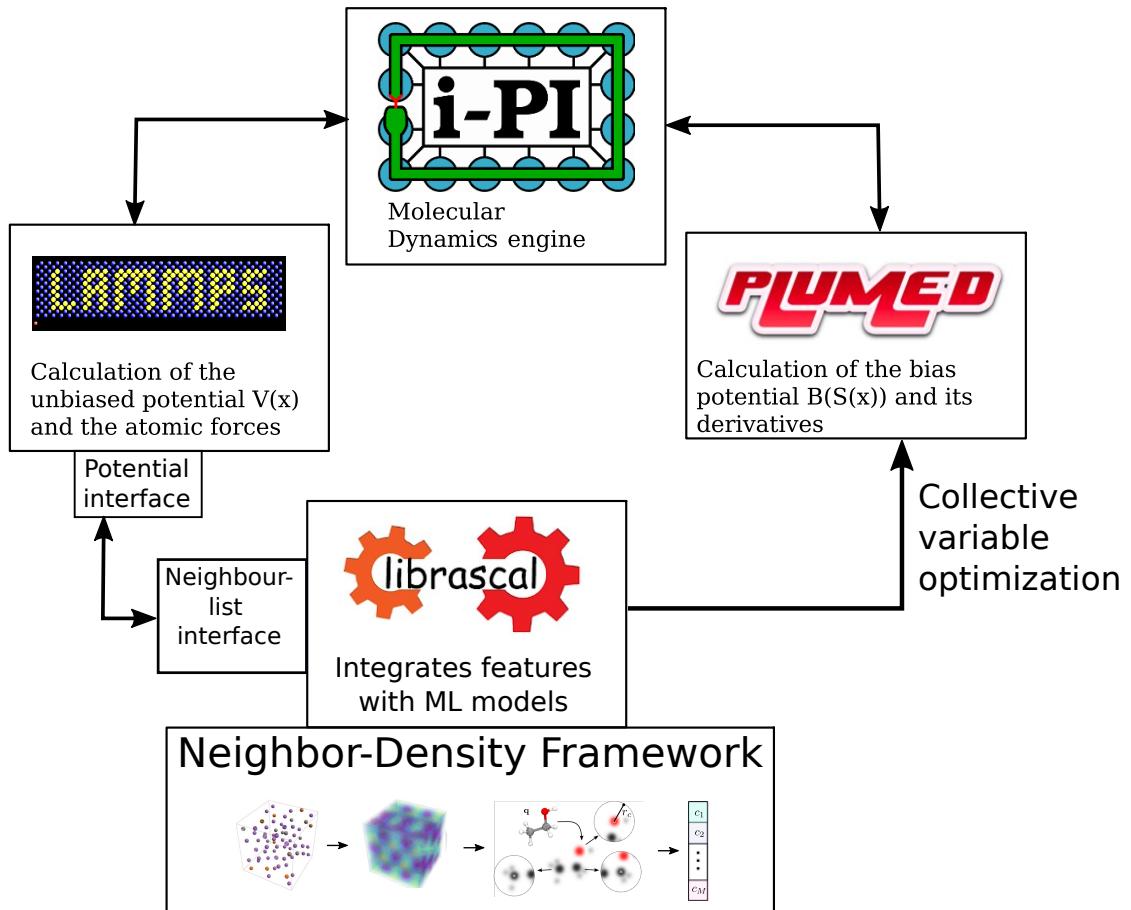


Figure 5.1: A schematic showing the interwork of the software pieces to run metadynamic simulations to study interfacial effects of barium titanite.

computation of partial gradients also needs to consider dependencies of the gradient on the central energies of its neighbors as pointed out in Eq. 5.8.

## 5.4 Serialization of MLIP

In last decade, the rapid development of machine learning models has resulted in numerous interfaces for LAMMPS[85]. While the model accuracy is a good indicator for the usability of a model, it is still not reliable enough to be a guarantee that the model will work for MD software as undersampling of the phase space can always be an issue for any kind of model. One is therefore forced to retrain the model for a dataset covering a larger part of the phase pace or switch to a more accurate model during the analysis of an experiment. As all ML models show different trade-offs between accuracy, evaluation time, trainings cost and hyperparameter optimization the set of suitable models depend on the system of interest, the dataset size and given computational resources. As the dataset size changes over the course of analysis different models become more suitable candidates and thus a different model is more suitable for the analysis. The current software infrastructure of ML models for MD software causes a lot of friction when changing the model type as the MD software needs to be recompiled for a different interface. Even more crucial is the fact that the model development is often conducted in higher-level languages like Python or Julia due to their flexibility. This however restricts the usage of the model also to MD packages in the same higher-level language or requires the implementation of a serialization for the model plus interface for an MD package. This work restricts the usage of a lot of developed ML models in low-level MD packages which are often required to conduct insightful research. For nearly five years following its initial publication, the widely-used SchNet model[86] lacked an interface with a low-level MD package. Similar problems exist in industry where models trained with different ML packages need to be shipped to devices with different hardware architectures and different software stacks making it hard to reliably provide the same version of the package on each device. The industry therefore developed an open standard for machine learning models open neural network exchange (ONNX). This standard can however not be used for MLIPs as they lack the support of the inference for gradients.

The Open Knowledgebase of Interatomic Models (OpenKIM)[87] tries to address the problem. They developed abstract representations of the data and processing directives necessary to perform a molecular simulation thereby unifying the interfaces of several MD packages to one interface, namely the KIM API[88]. While it reduces the cost of the number interfaces that are needed, they are far from covering comprehensively all relevant MD packages, as packages like GROMACS[89] and C2PK[82] are missing. Most of the supported MD packages are implemented in higher-level languages for which a custom MLIP interface can be easily implemented.

A solution we target with the software ecosystem developed in our lab is to use TorchScript as it supports the use of gradients and offers a usage of the model in C++ surpassing the high-level

## **5.4 Serialization of MLIP**

---

language barrier. It further supports advances model optimization utilities that can be used optimize complex models by kernel fusioning of the operation graph. Considering all that it seems like a promising candidate to standardize the landscape of MLIPs.



# Bibliography

- [1] Michael J Willatt, Félix Musil, and Michele Ceriotti. Atom-density representations for machine learning. *The Journal of chemical physics*, 150(15):154110, 2019.
- [2] Juhwan Noh, Jaehoon Kim, Helge S Stein, Benjamin Sanchez-Lengeling, John M Gregoire, Alan Aspuru-Guzik, and Yousung Jung. Inverse design of solid-state materials via a continuous representation. *Matter*, 1(5):1370–1384, 2019.
- [3] Wikipedia. Alcohol (chemistry) — Wikipedia, the free encyclopedia. [http://en.wikipedia.org/w/index.php?title=Alcohol%20\(chemistry\)&oldid=1177472846](http://en.wikipedia.org/w/index.php?title=Alcohol%20(chemistry)&oldid=1177472846), 2023. [Online; accessed 15-October-2023].
- [4] Felix Musil, Andrea Grisafi, Albert P Bartók, Christoph Ortner, Gábor Csányi, and Michele Ceriotti. Physics-inspired structural representations for molecules and materials. *Chemical Reviews*, 121(16):9759–9815, 2021.
- [5] Emil Prodan and Walter Kohn. Nearsightedness of electronic matter. *Proceedings of the National Academy of Sciences*, 102(33):11635–11638, 2005.
- [6] Jigyasa Nigam, Sergey Pozdnyakov, Guillaume Fraux, and Michele Ceriotti. Unified theory of atom-centered representations and message-passing machine-learning schemes. *The Journal of Chemical Physics*, 156(20), 2022.
- [7] Sergey N Pozdnyakov and Michele Ceriotti. Smooth, exact rotational symmetrization for deep learning on point clouds. *arXiv preprint arXiv:2305.19302*, 2023.
- [8] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5):058301, 2012.
- [9] Grégoire Montavon, Katja Hansen, Siamac Fazli, Matthias Rupp, Franziska Biegler, Andreas Ziehe, Alexandre Tkatchenko, Anatole V Lilienfeld, and Klaus-Robert Müller. Learning invariant representations of molecules for atomization energy prediction. In *Advances in neural information processing systems*, pages 440–448, 2012.
- [10] Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole Von Lilienfeld.

## Bibliography

---

- Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics*, 15(9):095003, 2013.
- [11] Ali Sadeghi, S Alireza Ghasemi, Bastian Schaefer, Stephan Mohr, Markus A Lill, and Stefan Goedecker. Metrics for measuring distances in configuration spaces. *The Journal of chemical physics*, 139(18):184118, 2013.
  - [12] Bing Huang and O. Anatole von Lilienfeld. Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity. *The Journal of Chemical Physics*, 145.
  - [13] Li Zhu, Maximilian Amsler, Tobias Fuhrer, Bastian Schaefer, Somayeh Faraji, Samare Rostami, S Alireza Ghasemi, Ali Sadeghi, Migle Grauzintytė, Chris Wolverton, et al. A fingerprint based metric for measuring similarities of crystalline structures. *The Journal of chemical physics*, 144(3):034203, 2016.
  - [14] Katja Hansen, Franziska Biegler, Raghunathan Ramakrishnan, Wiktor Pronobis, O Anatole Von Lilienfeld, Klaus-Robert Müller, and Alexandre Tkatchenko. Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space. *The journal of physical chemistry letters*, 6(12):2326–2331, 2015.
  - [15] James Barker, Johannes Bulin, Jan Hamaekers, and Sonja Mathias. Localized coulomb descriptors for the gaussian approximation potential. *arXiv preprint arXiv:1611.05126*, 2016.
  - [16] Jonathan E Moussa. Comment on “fast and accurate modeling of molecular atomization energies with machine learning”. *Physical review letters*, 109(5):059801, 2012.
  - [17] Stefan Chmiela, Huziel E Sauceda, Klaus-Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature communications*, 9(1):1–10, 2018.
  - [18] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
  - [19] Sandip De, Albert P Bartók, Gábor Csányi, and Michele Ceriotti. Comparing molecules and solids across structural and alchemical space. *Physical Chemistry Chemical Physics*, 18(20):13754–13769, 2016.
  - [20] Albert P Bartók, Sandip De, Carl Poelking, Noam Bernstein, James R Kermode, Gábor Csányi, and Michele Ceriotti. Machine learning unifies the modeling of materials and molecules. *Science advances*, 3(12):e1701816, 2017.
  - [21] Marc OJ Jäger, Eiaki V Morooka, Filippo Federici Canova, Lauri Himanen, and Adam S Foster. Machine learning hydrogen adsorption on nanoclusters through structural descriptors. *npj Computational Materials*, 4(1):1–8, 2018.

## Bibliography

---

- [22] Jörg Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of chemical physics*, 134(7):074106, 2011.
- [23] Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18):184115, 2013.
- [24] Ralf Drautz. Atomic cluster expansion for accurate and transferable interatomic potentials. *Physical Review B*, 99(1):014104, 2019.
- [25] Félix Musil and Michele Ceriotti. Machine learning at the atomic scale. *CHIMIA International Journal for Chemistry*, 73(12):972–982, 2019.
- [26] JS Dowker. Spherical harmonics, invariant theory and maxwell’s poles. *arXiv preprint arXiv:0805.1904*, 2008.
- [27] AP Yutsis and AA Bandzaitis. Theory of angular momentum in quantum mechanics. *Vil’nyus*, 1965.
- [28] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. *Advances in Neural Information Processing Systems*, 31, 2018.
- [29] Tao Yan, Jiamin Wu, Tiankuang Zhou, Hao Xie, Feng Xu, Jingtao Fan, Lu Fang, Xing Lin, and Qionghai Dai. Fourier-space diffractive deep neural network. *Physical review letters*, 123(2):023901, 2019.
- [30] Jigyasa Nigam, Sergey Pozdnyakov, and Michele Ceriotti. Recursive evaluation and iterative contraction of n-body equivariant features. *The Journal of Chemical Physics*, 153(12), 2020.
- [31] Alexander V Shapeev. Moment tensor potentials: A class of systematically improvable interatomic potentials. *Multiscale Modeling & Simulation*, 14(3):1153–1173, 2016.
- [32] Yunxing Zuo, Chi Chen, Xiangguo Li, Zhi Deng, Yiming Chen, Jörg Behler, Gábor Csányi, Alexander V Shapeev, Aidan P Thompson, Mitchell A Wood, et al. Performance and cost assessment of machine learning interatomic potentials. *The Journal of Physical Chemistry A*, 124(4):731–745, 2020.
- [33] Stephen R Xie, Matthias Rupp, and Richard G Hennig. Ultra-fast interpretable machine-learning potentials. *npj Computational Materials*, 9(1):162, 2023.
- [34] Alexander Goscinski, Guillaume Fraux, Giulio Imbalzano, and Michele Ceriotti. The role of feature space in atomistic learning. *Machine Learning: Science and Technology*, 2(2):025028, 2021.
- [35] Félix Musil, Max Veit, Alexander Goscinski, Guillaume Fraux, Michael J Willatt, Markus Stricker, Till Junge, and Michele Ceriotti. Efficient implementation of atom-density representations. *The Journal of Chemical Physics*, 154(11), 2021.

## Bibliography

---

- [36] Lucjan Piela. Appendix j - orthogonalization. In Lucjan Piela, editor, *Ideas of Quantum Chemistry (Second Edition)*, pages e99–e103. Elsevier, Oxford, second edition edition, 2014.
- [37] Kristof T Schütt, Huziel E Saucedo, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet—a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.
- [38] Genevieve Dusson, Markus Bachmayr, Gábor Csányi, Ralf Drautz, Simon Etter, Cas van der Oord, and Christoph Ortner. Atomic cluster expansion: Completeness, efficiency and stability. *Journal of Computational Physics*, 454:110946, 2022.
- [39] Miguel A Caro. Optimizing many-body atomic descriptors for enhanced computational performance of machine learning based interatomic potentials. *Physical Review B*, 100(2):024112, 2019.
- [40] Alexander Goscinski, Félix Musil, Sergey Pozdnyakov, Jigyasa Nigam, and Michele Ceriotti. Optimal radial basis for density-based atomic representations. *The Journal of Chemical Physics*, 155(10), 2021.
- [41] Filippo Bigi, Kevin K Huguenin-Dumittan, Michele Ceriotti, and David E Manolopoulos. A smooth basis for atomistic machine learning. *The Journal of Chemical Physics*, 157(23), 2022.
- [42] Nataliya Lopanitsyna, Guillaume Fraux, Maximilian A Springer, Sandip De, and Michele Ceriotti. Modeling high-entropy transition metal alloys with alchemical compression. *Physical Review Materials*, 7(4):045802, 2023.
- [43] Rifkin. Regularized least squares. [Online; accessed 20-October-2023].
- [44] SS Vallender. Calculation of the Wasserstein distance between probability distributions on the line. *Theory Probab. Its Appl.*, 18(4):784–786, 1974.
- [45] Scott D. Cohen and Leonidas Guibas. The earth mover’s distance: Lower bounds and invariance under translation. pages 1–44, 1997.
- [46] Marco Cuturi. Permanents, transportation polytopes and positive definite kernels on histograms. *Int. Jt. Conf. Artif. Intell. IJCAI*, pages 732–737, 2007.
- [47] Sandip De, Albert P. Bartók, Gábor Csányi, and Michele Ceriotti. Comparing molecules and solids across structural and alchemical space. *Phys. Chem. Chem. Phys.*, 18(20):13754–13769, 2016.
- [48] Onur Çaylak, O Anatole von Lilienfeld, and Björn Baumeier. Wasserstein metric for improved qml with adjacency matrix representations. *arXiv preprint arXiv:2001.11005*, 2020.

## Bibliography

---

- [49] Michael J. Willatt, Félix Musil, and Michele Ceriotti. Atom-density representations for machine learning. *J. Chem. Phys.*, 150(15):154110, April 2019.
- [50] Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, March 1966.
- [51] Albert P. Bartók, Sandip De, Carl Poelking, Noam Bernstein, James R. Kermode, Gábor Csányi, and Michele Ceriotti. Machine learning unifies the modeling of materials and molecules. *Sci. Adv.*, 3(12):e1701816, December 2017.
- [52] Felix A. Faber, Luke Hutchison, Bing Huang, Justin Gilmer, Samuel S. Schoenholz, George E. Dahl, Oriol Vinyals, Steven Kearnes, Patrick F. Riley, and O. Anatole von Lilienfeld. Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error. *J. Chem. Theory Comput.*, 13(11):5255–5264, November 2017.
- [53] Michael J. Willatt, Félix Musil, and Michele Ceriotti. Feature optimization for atomistic machine learning yields a data-driven construction of the periodic table of the elements. *Phys. Chem. Chem. Phys.*, 20(47):29661–29668, 2018.
- [54] Federico M. Paruzzo, Albert Hofstetter, Félix Musil, Sandip De, Michele Ceriotti, and Lyndon Emsley. Chemical shifts in molecular solids by machine learning. *Nat. Commun.*, 9(1):4501, December 2018.
- [55] S T Roweis and L K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–6, December 2000.
- [56] Sergey N Pozdnyakov, Michael J Willatt, Albert P Bartók, Christoph Ortner, Gábor Csányi, and Michele Ceriotti. Incompleteness of Atomic Structure Representations. *Phys. Rev. Lett.*, 125:166001, 2020.
- [57] SN Pozdnyakov, L Zhang, C Ortner, G Csányi, and M Ceriotti. Local invertibility and sensitivity of atomic structure-feature mappings [version 1; peer review: 2 approved]. *Open Research Europe*, 1(126), 2021.
- [58] Thang Viet Huynh and Manfred Mücke. Error analysis and precision estimation for floating-point dot-products using affine arithmetic. In *The 2011 International Conference on Advanced Technologies for Communications (ATC 2011)*, pages 319–322. IEEE, 2011.
- [59] Mikio L. Braun, Joachim M. Buhmann, and Klaus-Robert Müller. On relevant dimensions in kernel feature spaces. *Journal of Machine Learning Research*, 9(62):1875–1908, 2008.
- [60] Matan Gavish and David L. Donoho. The optimal hard threshold for singular values is  $4/\sqrt{3}$ . *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.
- [61] Andrey A Shabalin and Andrew B Nobel. Reconstruction of a low-rank matrix in the presence of gaussian noise. *Journal of Multivariate Analysis*, 118:67–76, 2013.

## Bibliography

---

- [62] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comp. Phys. Comm.*, 271:108171, 2022.
- [63] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.
- [64] Benjamin Yee Shing Li, Lam Fat Yeung, and King Tim Ko. Indefinite kernel ridge regression and its application on qsar modelling. *Neurocomputing*, 158:127–133, 2015.
- [65] Richard H Bartels, John C Beatty, and Brian A Barsky. *An introduction to splines for use in computer graphics and geometric modeling*. Morgan Kaufmann, 1995.
- [66] Amrita Mathuriya, Ye Luo, Anouar Benali, Luke Shulenburger, and Jeongnim Kim. Optimization and parallelization of b-spline based orbital evaluations in qmc on multi/many-core shared memory processors. In *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 213–223. IEEE, 2017.
- [67] Gael Guennebaud, Benoit Jacob, et al. Eigen: a c++ linear algebra library. URL <http://eigen.tuxfamily.org>, Accessed, 22, 2014.
- [68] Jigyasa Nigam, Sergey Pozdnyakov, and Michele Ceriotti. Recursive evaluation and iterative contraction of  $N$ -body equivariant features. *J. Chem. Phys.*, 153(12):121101, September 2020.
- [69] Bing Huang and O. Anatole Von Lilienfeld. Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity. *J. Chem. Phys.*, 145(16), 2016.
- [70] Sijmen de Jong and Henk A.L. Kiers. Principal covariates regression. *Chemometrics and Intelligent Laboratory Systems*, 14(1-3):155–164, April 1992.
- [71] Benjamin Helfrecht, Rose K Cersonsky, Guillaume Fraux, and Michele Ceriotti. Structure-property maps with Kernel Principal Covariates Regression. *Mach. Learn.: Sci. Technol.*, July 2020.
- [72] Frank H Stillinger and Thomas A Weber. Computer simulation of local order in condensed phases of silicon. *Physical review B*, 31(8):5262, 1985.
- [73] Jerry Tersoff. Empirical interatomic potential for silicon with improved elastic properties. *Physical Review B*, 38(14):9902, 1988.
- [74] Albert P Bartók, Mike C Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical review letters*, 104(13):136403, 2010.

## Bibliography

---

- [75] Albert P Bartók, James Kermode, Noam Bernstein, and Gábor Csányi. Machine learning a general-purpose interatomic potential for silicon. *Physical Review X*, 8(4):041048, 2018.
- [76] Modeling the ferroelectric phase transition in barium titanate with dft accuracy and converged sampling.
- [77] Lorenzo Gigli, Davide Tisi, Federico Grasselli, and Michele Ceriotti. Mechanism of charge transport in lithium thiophosphate. *arXiv preprint arXiv:2310.15679*, 2023.
- [78] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- [79] Joaquin Quinonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.
- [80] Gábor Csányi, Steven Winfield, J R Kermode, A De Vita, Alessio Comisso, Noam Bernstein, and Michael C Payne. Expressive programming for computational physics in fortran 95+. *IoP Comput. Phys. Newsletter*, page Spring 2007, 2007.
- [81] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Pál, Jeremy C Smith, Berk Hess, and Erik Lindahl. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25, 2015.
- [82] Thomas D Kühne, Marcella Iannuzzi, Mauro Del Ben, Vladimir V Rybkin, Patrick Seewald, Frederick Stein, Teodoro Laino, Rustam Z Khaliullin, Ole Schütt, Florian Schiffmann, et al. Cp2k: An electronic structure and molecular dynamics software package-quickstep: Efficient and accurate electronic structure calculations. *The Journal of Chemical Physics*, 152(19), 2020.
- [83] Venkat Kapil, Mariana Rossi, Ondrej Marsalek, Riccardo Petraglia, Yair Litman, Thomas Spura, Bingqing Cheng, Alice Cuzzocrea, Robert H Meißner, David M Wilkins, et al. i-pi 2.0: A universal force engine for advanced molecular simulations. *Computer Physics Communications*, 236:214–223, 2019.
- [84] Massimiliano Bonomi, Davide Branduardi, Giovanni Bussi, Carlo Camilloni, Davide Provasi, Paolo Raiteri, Davide Donadio, Fabrizio Marinelli, Fabio Pietrucci, Ricardo A Broglia, and Michele Parrinello. PLUMED: A portable plugin for free-energy calculations with molecular dynamics. *Comput. Phys. Commun.*, 180(10):1961–1972, October 2009.
- [85] James R. Kermode, Albert P. Bartók, and Gábor Csányi. QUIP.
- [86] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. SchNet – A deep learning architecture for molecules and materials. *J. Chem. Phys.*, 148(24):241722, June 2018.

## Bibliography

---

- [87] Daniel S Karls, Matthew Bierbaum, Alexander A Alemi, Ryan S Elliott, James P Sethna, and Ellad B Tadmor. The openkim processing pipeline: A cloud-based automatic material property computation engine. *The Journal of Chemical Physics*, 153(6), 2020.
- [88] Ryan S. Elliott and Ellad B. Tadmor. Knowledgebase of interatomic models (kim) application programming interface (api), 2011.
- [89] B Hess, C Kutzner, D van der Spoel, and E Lindahl. {GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation}. *J Chem Theory Comput*, 4(3):435–447, 2008.