Beyond the Born-Oppenheimer static surface model for molecule-surface reactions

Spiering, P.

Author: Spiering, P.
Title: Beyond the Born-Oppenheimer static surface model for molecule-surface reactions
Issue Date: 2019-12-16

Chapter 6

Machine-Learning based Continuous Representations of Electronic Friction Tensors

Abstract

Electronic friction theory allows to account for effects of electron-hole pair excitation on the dynamics of molecules on metal surfaces in a computationally efficient manner, given that continuous representations of the molecular-coordinate-dependent electronic friction tensors are available. In general, those tensors can be anisotropic and non-diagonal, which makes continuous representations more challenging than for scalar quantities, like e.g. potential energy surfaces. In this chapter neural-network-based machine learning models are developed and compared that account for symmetry properties of molecule-surface systems to meet this challenge. First, a formalism is presented to describe symmetry properties of electronic friction tensors for molecules on frozen metal surfaces. Two strategies result that allow these properties to be enforced within machine learning algorithms: an "*a priori*" and an "*a posteriori*" method. For the "*a priori*" method, different symmetry adapted input coordinates can affect the fitting accuracy and computational efficiency. The elements of the tensor can either be obtained by assigning a unique neural network to each element, or by constructing a neural network with multiple outputs and subsequently arrange these outputs as a tensor. Positive definite tensors can be enforced by generating a lower triangular tensor that is subsequently squared. Comparing different methods shows that the best results are obtained for an "*a priori*" method when including "asymmetric" contributions for the symmetry adapted input coordinates using a single neural network for all tensor elements and enforcing positive definiteness.

6.1 Introduction

Machine learning has become an ubiquitous technique in computational chemistry for the construction of continuous representations of quantities that depend on a large number of atomic coordinates based on first-principles calculations [1–14]. In particular potential energy surfaces (PESs) have received a lot of attention, usually in order to perform molecular dynamics (MD) simulations at considerably reduced computational cost [1, 5, 10, 12–14]. One essential ingredient for all machine learning models is the representation of atomic coordinates, since it is the only way to rigorously enforce (spatial) symmetries of the studied systems (molecules, solids and/or interfaces) in the model [1, 5, 7, 10, 12, 15]. When PESs are the target quantities, it is “only” the invariance of these scalar quantities under the corresponding symmetry transformation that needs to be directly accounted for. Forces that are calculated as the partial derivatives with respect to the atomic coordinates from the machine learning model for the PES automatically conform to the proper symmetries [1, 15], but need more care to be represented accurately [2, 3, 5, 16]. On the other hand, making atomic forces the primary target quantities requires constructing and incorporating the symmetry-related invariance of vectors into a machine learning model, which is a much more difficult challenge, that Chmiela et al. [10, 16] have mastered only recently. For tensors the situation is even more challenging. In their recent work Grisafi et al. [6] have developed a machine learning model that includes symmetries of an atomistically described system for its “global” (intensive) property tensors (e.g. the 3×3 polarizability tensor).

The focus here is on the construction of machine learning models for electronic friction tensors. Given its computational efficiency, electronic friction theory [17] is the current workhorse for modeling the effect of electron-hole pair excitations in a metal surface on molecules moving close to the latter [18, 19]. These so-called non-adiabatic effects can considerably affect the vibrationally inelastic scattering of diatomic molecules [20–23] as well as their dissociation probabilities [24], when the electronic friction calculations are based on electronic structure theory that in principle allows them to be non-isotropic and non-diagonal (in their Cartesian representation) [25].

In that case, these so-called orbital-dependent friction (ODF) tensors are atom-specific (extensive) quantities, i.e. the amount of elements and coordinate dependence scales with the amount of atoms for which non-adiabatic effects are to be considered, which is a significant difference compared to the aforementioned work of [6]. At present, the computational burden of evaluating electronic friction tensors practically forbids including surface atom motion [20], which is currently considered to not significantly affect the non-adiabatic effects described by electronic friction.[26] Machine learning models that capture the combined symmetry of diatomic molecules on highly-symmetric low-index metal surfaces are therefore currently the only way to calculate sufficiently many MD trajectories (based on a Langevin equation) as required to obtain observables that can be compared with experimental data [21–24].

The goal of this chapter is to formalize symmetry constraints for atom-specific tensors and subsequently obtain a method for constructing continuous representations via machine learning that incorporate the aforementioned symmetry - with a particular focus on ODF tensors for molecules interacting with frozen surfaces. Three key strategies of the latter are considered here.

The first strategy is the most straightforward one, where the neural network (NN) is expected to learn the symmetries from the presented data set. This approach can easily be enhanced by using the relevant symmetry operators to extend the training and test data sets. The subsequent fit will, however, not strictly obey the correct symmetry behavior because of the fitting error which is arbitrarily different also for symmetrically equivalent configurations. A further downside is that the training time increases substantially since a much larger data set is required.

The second strategy is to perform *a posteriori* correction to a non-symmetric NN fit in order to correct symmetry violations using an explicit symmetrization method. This approach results in fitting errors that are also symmetrized. Although the overall result is guaranteed to be exactly symmetric, the NN has to be evaluated multiple times at symmetrically equivalent configurations. This results in both increased training and evaluation times.

Finally, the third strategy, which is the most desirable, is to present an *a priori* symmetry-adapted model to the machine learning algorithm. Non-scalar quantities are the same at symmetrically equivalent configurations only after the appropriate well-defined mathematical transformation. Here a pragmatic approach is provided to construct the appropriate tensor transformations, which are validated by applying the second approach to the postulated model and verifying that that no symmetry violations were observed.

This chapter is structured as follows. First, symmetry properties of tensors are introduced using an abstract formalism, which allows to selectively incorporate the aforementioned key strategies. Afterwards, this formalism is applied to construct NN-based continuous representations of electronic friction tensors that have recently been studied [21, 24]. Using the same reference data for training and test sets, the resulting fits thus allow one to compare practical performance of these machine learning models.

6.2 Tensor Field Properties of Systems with Symmetry

Here a systematic approach is provided for obtaining symmetry-constrained representations of tensor fields. These symmetric representations are obtainable by considering their behavior under coordinate transforms and impose this behavior on analytical expressions, such as NN fits. Specifically, the focus is on the imposed behaviour on tensor fields that transform in the same way as the electronic friction tensor, which has previously been obtained NN fits [21, 24]. In order to obtain these coordinate transformations, it is necessary to define the configuration $\mathbf{p}$ that describes the location of all atoms in the system. This $\mathbf{p}$, while in principle an abstract object, can be expressed in different coordinate systems. Two examples, which are commonly used in chemistry, are a Cartesian coordinate system, where each atom is described by its three Cartesian coordinates, and an internal coordinate system, where the center of mass (COM) of a group of atoms is used in combination with some measure of the relative positioning of the atoms in that group together with its orientation. Now let $\mathbf{R}$ and $\mathbf{R}'$ denote two such different coordinate systems where the atomic configuration $\mathbf{p}$ is expressed in the respective coordinates as $\mathbf{R}_\mathbf{p}$ and $\mathbf{R}'_\mathbf{p}$. Here the elements $R^i_\mathbf{p}$ and $R'^i_\mathbf{p}$ of $\mathbf{R}$ and $\mathbf{R}'$ then simply describe the same physical object in their respective different coordinate systems. Moreover, assume that a forward mapping $\hat{M}_{\mathbf{R}'}\mathbf{R}_\mathbf{p} = \mathbf{R}'_\mathbf{p}$ and a backward mapping $\hat{M}_\mathbf{R}\mathbf{R}'_\mathbf{p} = \mathbf{R}_\mathbf{p}$ exist, noting that these mappings would usually be non-linear.

Tensor field properties are then simply properties that are associated with a configuration $\mathbf{p}$ and that transform in a specific way when a different coordinate system is chosen. Some such properties, like the PES $\boldsymbol{V}(\mathbf{p})$, do not have a transformation rule and can thus also be considered scalar fields $(V(\mathbf{p}))$. Other properties of the system, which are at least partially defined through a change in the configuration $\mathbf{p}$, do have associated transformation rules. Examples are velocities $\left[v^{i}\left(\mathbf{p}\right)\right]^{\mathbf{R}} = \frac{dR^{i}_{\mathbf{p}}}{dt}$ and forces $\left[F_{i}\left(\mathbf{p}\right)\right]_{\mathbf{R}} = -\frac{dV}{dR^{j}_{\mathbf{p}}}$. The elements of these properties are related to a change in the configuration $\mathbf{p}$ expressed in a specific coordinate system $\mathbf{R}_{\mathbf{p}}$. It is convenient to be able to transform the elements of these properties to allow expressing them in a different coordinate system. Even though $\hat{M}_{\mathbf{R}'}$ (and $\hat{M}_{\mathbf{R}}$) in general do not describe linear transformations in the context of chemical systems (e.g. typically internal coordinates are given by distances and angles), the transformation of both the aforementioned differential properties is given by a point-wise linear transformation i.e.

$$\left[v^{i}\left(\mathbf{p}\right)\right]^{\mathbf{R}'} = \frac{dR'^{i}_{\mathbf{p}}}{dt} = \frac{\partial R'^{i}_{\mathbf{p}}}{\partial R^{j}_{\mathbf{p}}}\frac{dR^{j}_{\mathbf{p}}}{dt} = \frac{\partial R'^{i}_{\mathbf{p}}}{\partial R^{j}_{\mathbf{p}}}\left[v^{j}\left(\mathbf{p}\right)\right]^{\mathbf{R}} \tag{6.1}$$

for velocities and

$$\left[F_{i}\left(\mathbf{p}\right)\right]_{\mathbf{R}'} = \frac{dV}{dR'^{i}_{\mathbf{p}}} = \frac{\partial R^{j}_{\mathbf{p}}}{\partial R'^{i}_{\mathbf{p}}}\frac{dV}{dR^{j}_{\mathbf{p}}} = \frac{\partial R^{j}_{\mathbf{p}}}{\partial R'^{i}_{\mathbf{p}}}\left[F_{j}\left(\mathbf{p}\right)\right]_{\mathbf{R}} \tag{6.2}$$

for forces. Here Einstein's summation convention for implicitly summing over repeated indices is used. A tensor field such as the velocity $\boldsymbol{v}\left(\mathbf{p}\right)$ or force $\boldsymbol{F}\left(\mathbf{p}\right)$ can thus be defined invariant of the chosen coordinate system. Elements of these invariant velocity $\left[v^{i}\left(\mathbf{p}\right)\right]^{\mathbf{R}}$ and force $\left[F_{i}\left(\mathbf{p}\right)\right]_{\mathbf{R}}$ tensors can then be expressed for a specific coordinate system, this is indicated here explicitly using a bracket notation, as it will later be necessary to keep track of the coordinate system for introducing symmetry properties. Moreover, the elements of the force have a lower index to indicate that they transform covariantly, i.e. in the same way as the tangent vector basis (linearized) of the corresponding coordinate system, while the elements of the velocity have an upper index to indicate that they transform contravariantly, i.e. in the same way as the dual of the tangent vector basis.

Tensor fields can also describe properties that transform as a combination of co- and contravariant elements. An example of this is when the partial derivatives in equation 6.2 are written as the Jacobian tensor

$$\left[J^{j}_{i}\left(\mathbf{p}\right)\right]^{\mathbf{R}}_{\mathbf{R}'} = \frac{\partial R^{j}_{\mathbf{p}}}{\partial R'^{i}_{\mathbf{p}}} = \frac{\partial\left(\hat{M}_{\mathbf{R}}\mathbf{R}'_{\mathbf{p}}\right)^{j}}{\partial R'^{i}_{\mathbf{p}}}. \tag{6.3}$$

Here we focus on tensor fields that transform in the same way as the electronic friction tensor: the doubly covariant tensor field $\left[\eta_{ij}\left(\mathbf{p}\right)\right]_{\mathbf{RR}}$. The transformation rule for the elements of this type of tensor field from coordinate system $\mathbf{R}$ to $\mathbf{R}'$ is written as

$$\left[\eta_{i'j'}\left(\mathbf{p}\right)\right]_{\mathbf{R}'\mathbf{R}'} = \left[J^{i}_{i'}\left(\mathbf{p}\right)\right]^{\mathbf{R}}_{\mathbf{R}'} \left[\eta_{ij}\left(\mathbf{p}\right)\right]_{\mathbf{RR}} \left[J^{j}_{j'}\left(\mathbf{p}\right)\right]^{\mathbf{R}}_{\mathbf{R}'}, \tag{6.4}$$

with indices i, j and i', j' denoting tensor elements in the R and R' coordinate systems respectively. Here and in the following, the use of the Einstein summation convention for these tensor elements is limited to indices that appear exactly once covariantly and once contravariantly, while they are expressed in the same basis.

Some systems also have symmetries, which means that there exist symmetrically equivalent configurations that have equivalent tensors. Equivalent for tensor fields however, does not mean that all the elements of the tensor are the same. For example, if a mirror symmetry exists then the symmetrically equivalent configurations have the same forces, except that the forces are additionally mirrored themselves, or equivalently, the coordinate basis is mirrored and the same force is now found in this symmetry coordinate system as illustrated in Figure 6.1. This can be elegantly expressed using the above tensor field transformations.

Let $\hat{s}$ be a symmetry operator that generates the symmetrically equivalent configuration $\mathbf{p}_{\hat{s}}$ of $\mathbf{p}$ and $\hat{M}_{\hat{s}\mathbf{R}}$ be the mapping which obtains the concomitant symmetry coordinate system of $\hat{s}$. Following the above reasoning, the elements of a symmetric tensor field then must be the same at configurations $\mathbf{p}$ and $\mathbf{p}_{\hat{s}}$ if the latter is expressed in the mapped coordinate system according to

$$\hat{s}\left[\eta_{ij}\left(\mathbf{p}\right)\right]_{\mathbf{RR}} = \left[\eta_{ij}\left(\mathbf{p}_{\hat{s}}\right)\right]_{\hat{s}\mathbf{R}\hat{s}\mathbf{R}} = \left[\eta_{ij}\left(\mathbf{p}\right)\right]_{\mathbf{RR}} \; \forall\ \hat{s}\ \in\ G. \tag{6.5}$$

This equivalence holds for all symmetry operators $\hat{s}$ in the group G that together describe the symmetry of the system. By making use of equation 6.4 it is possible to write $\hat{s}\boldsymbol{\eta}\left(\mathbf{p}\right)$ expressed in the original coordinate system R according to

$$\hat{s}\left[\eta_{ij}\left(\mathbf{p}\right)\right]_{\mathbf{RR}} = \left[J^{i'}_{i}\left(\mathbf{p}_{\hat{s}}\right)\right]^{\mathbf{R}}_{\hat{s}\mathbf{R}} \left[\eta_{i'j'}\left(\mathbf{p}_{\hat{s}}\right)\right]_{\mathbf{RR}} \left[J^{j'}_{j}\left(\mathbf{p}_{\hat{s}}\right)\right]^{\mathbf{R}}_{\hat{s}\mathbf{R}}, \tag{6.6}$$

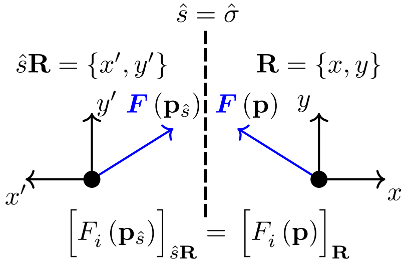
$\hat{s} = \hat{\sigma}$
$\hat{s}\mathbf{R} = \{x', y'\}$
$\mathbf{R} = \{x, y\}$
y'
$\boldsymbol{F}(\mathbf{p}_{\hat{s}})$
$\boldsymbol{F}(\mathbf{p})$
y
x'
x
$\left[F_i(\mathbf{p}_{\hat{s}})\right]_{\hat{s}\mathbf{R}} = \left[F_i(\mathbf{p})\right]_{\mathbf{R}}$

Figure 6.1: Schematic explanation of the behavior of tensors (force $\boldsymbol{F}(\mathbf{p})$) if a mirror symmetry (indicated by the dashed black line) $\hat{\sigma}$ can be identified. On the right hand side of the mirror, the original configuration $\mathbf{p}$ is indicated together with the force $\boldsymbol{F}(\mathbf{p})$ and Cartesian coordinate system $\mathbf{R} = \{x, y\}$. The left hand side shows the symmetrically equivalent configuration $\mathbf{p}_{\hat{s}}$ which has the mirrored force $\boldsymbol{F}(\mathbf{p}_{\hat{s}})$ that has the same elements as $\boldsymbol{F}(\mathbf{p})$ if they are expressed in the symmetry coordinate system $\hat{s}\mathbf{R} = \{x', y'\}$ and the Cartesian coordinate system $\mathbf{R}$ respectively.

where if $\mathbf{R}$ is taken in cartesian coordinates then the resulting Jacobian tensor $\boldsymbol{J}(\mathbf{p}_{\hat{s}})$ is independent of $\mathbf{p}$ and equal to the appropriate transformation matrix describing the symmetry operator $\hat{s}$.

A continuous representation of a tensor field $\boldsymbol{\eta}^{\mathsf{cont}}(\mathbf{p})$ can thus be symmetrized *a posteriori* by taking the average over the h symmetrically equivalent configurations under the symmetry operators in G, represented in the appropriately rotated basis, according to

$$\left[\eta_{ij}^{\mathsf{post}}(\mathbf{p})\right]_{\mathbf{RR}} = \frac{1}{h}\sum_{\hat{s}\in G} \hat{s}\left[\eta_{ij}^{\mathsf{cont}}(\mathbf{p})\right]_{\mathbf{RR}} = \frac{1}{h}\sum_{\hat{s}\in G}\left[\eta_{ij}^{\mathsf{cont}}(\mathbf{p}_{\hat{s}})\right]_{\hat{s}\mathbf{R}\hat{s}\mathbf{R}}. \qquad (6.7)$$

The other way round, equation 6.7 also provides a test for a candidate tensor construction $\boldsymbol{\eta}^{\mathsf{cand}}(\mathbf{p})$ which is intended to already have the correct symmetry behavior. If it can be shown that this test is passed based on analytical arguments then this is considered an *a priori* symmetrized tensor $\boldsymbol{\eta}^{\mathsf{priori}}(\mathbf{p})$. In practice, this can also be verified by a numerical test on a sufficiently large grid of configurations, or be used as an estimate of how large the violation of symmetry is (e.g. in comparison with the fitting error of $\boldsymbol{\eta}^{\mathsf{cand}}(\mathbf{p})$).

A special case exists when a configuration exists that is invariant under one of the symmetry operators $\hat{s}$ such that independent of the chosen coordinate system it is found that $\mathbf{R}_{\mathbf{p}} = \hat{s}\mathbf{R}_{\mathbf{p}} = \mathbf{R}_{\mathbf{p}^{\mathsf{seam}}}$. Example of such configurations are the fixed points (as present on the dashed line in Figure 6.1) on a mirror plane. These configurations consist along what is defined here as the symmetry seam, which would in the case of a mirror plane be the entire mirror plane, and have additional constraints on their elements. The constraints on the elements can be found by using the relevant $\hat{s}$ in equation 6.5 as

$$\left[\eta_{ij}\left(\mathbf{p}^{\mathsf{seam}}\right)\right]_{\mathbf{RR}} = \left[\eta_{ij}\left(\mathbf{p}^{\mathsf{seam}}\right)\right]_{\hat{s}\mathbf{R}\hat{s}\mathbf{R}}, \tag{6.8}$$

and realize that the constraints are now imposed due to the equivalence being required for the exact same configuration, i.e. the same tensor is required to be the same in two (or more) coordinate systems. These symmetry constraints are automatically satisfied for both *a priori* and *a posteriori* tensor constructions.

6.2.1 *a priori* Symmetric Coordinate Systems and Mappings

There are several methods to obtain an *a priori* symmetric tensor field $\boldsymbol{\eta}^{\mathsf{priori}}(\mathbf{p})$ by choosing an appropriate coordinate system and concomitant mapping from and to Cartesian coordinates. The general idea is to obtain the tensor elements in a coordinate system $\mathbf{R}^{\mathsf{sym}}$ which for symmetry equivalent configurations $\mathbf{p}_{\hat{s}}$ yields the same elements in the representation of that coordinate system i.e. $(R^{\mathsf{sym}})^{i}_{\mathbf{p}_{\hat{s}}} = (R^{\mathsf{sym}})^{i}_{\mathbf{p}}$. After this representation is obtained (e.g. through fitting a NN) it can then be transformed to the required coordinate system $\mathbf{R}$ using Eq. 6.4. For fitting it is not strictly necessary to have a well defined backwards mapping because forward mapping in combination with back-propagation (as implemented automatically in TensorFlow) can circumvent the need to transform the fitting data into the $\mathbf{R}^{\mathsf{sym}}$ representation.

6

Reduced Dimensional Mapping

By choosing a reduced dimensional dependence on the configuration it is possible to avoid some of the symmetry considerations, although at the cost of having a lower fitting accuracy. This can be done by defining a reference configuration $\mathbf{p}^{\mathsf{ref}}$ to be the same as the actual configuration $\mathbf{p}$ except for one (or possible multiple) direction(s) of a certain coordinate system $\mathbf{R}^{\mathsf{ref}}$ being a reference constant. A tensor can then be expressed in a different coordinate system $\mathbf{R}$ but including the imposed reduction in dependence in the reference coordinate system according to

$$\left[\eta^{\mathsf{ref}}_{ij}\left(\mathbf{p}\right)\right]_{\mathbf{RR}} = \underbrace{\left[J_i^k\left(\mathbf{p}\right)\right]_{\mathbf{R}}^{\mathbf{R}^{\mathsf{ref}}} \left[J_k^m\left(\mathbf{p}^{\mathsf{ref}}\right)\right]_{\mathbf{R}^{\mathsf{ref}}}^{\mathbf{R}}}_{\left[T_i^m\left(\mathbf{p},\mathbf{p}^{\mathsf{ref}}\right)\right]_{\mathbf{R}}^{\mathbf{R}}} \left[\eta^{\mathsf{ref}}_{mn}\left(\mathbf{p}^{\mathsf{ref}}\right)\right]_{\mathbf{RR}} \left[J_l^n\left(\mathbf{p}^{\mathsf{ref}}\right)\right]_{\mathbf{R}^{\mathsf{ref}}}^{\mathbf{R}} \left[J_j^l\left(\mathbf{p}\right)\right]_{\mathbf{R}}^{\mathbf{R}^{\mathsf{ref}}} . \tag{6.9}$$

For this approximation to work, the dependence of the tensor on the coordinate direction that is kept constant must be sufficiently small.

Piece-Wise Mapping

A piece-wise mapping is obtained by defining a region of the configuration space, the irreducible wedge, that has the property that starting from that region it is possible to access the entire configuration space by only applying consecutive symmetry operations. The piece-wise mapping is then constructed to map any configuration $\mathbf{p}$ to the irreducible wedge $\mathbf{p}^{\mathsf{irr}}$. In practice this can be done by applying consecutive symmetry operations until a configuration within the irreducible wedge is found. The tensor only needs to be fitted within this irreducible wedge and if the queried configuration is outside the irreducible wedge then the tensor can be unfolded using Equation 6.4.

While this method does obtain the correct symmetry of the system, there is no *a priori* reason for the behavior at the symmetry seam (which coincides with the boundary of such a irreducible wedge) to be correct. This becomes immediately clear by using the example of mirror symmetry in Figure 6.1. A piece-wise mapping procedure would assign an identity rotation when the configuration is still in the irreducible wedge and an infinitesimally small distance from the mirror plane while it would assign a mirror if it is an infinitesimally small distance on the other side of the mirror plane. This imposes, in this particular case, that the x component of the force tensor is exactly zero. If this is not the case then a discontinuity occurs, as in, the x component of the force tensor would be positive on one side of the mirror plane and negative on the other side. Doubly covariant tensors would e.g. have such discontinuities in the off-diagonal xy and yx components in this case.

The symmetry seam is only a small region and the violation can be kept under control by appropriately generating the data for a fitting procedure (i.e. provide enough data at the symmetry seam) and as such this method can still be effectively used in practice when proper care is taken.

Symmetry Adapted Coordinate Mapping

It is also possible to obtain a continuous mapping that can take into account the correct symmetry behavior. Let us assume a coordinate system $\mathbf{Q}$ exists that is invariant to any choice of symmetrically equivalent configuration $\mathbf{p}_{\hat{s}}$ such that $Q^i_{\mathbf{p}_{\hat{s}}} = Q^i_{\mathbf{p}}$ for all $\mathbf{p}$ and $\hat{s}$. Examples of such coordinates are symmetry adapted coordinates[27], permutation invariant polynomials [12] and the atom centered symmetry functions of Behler and Parinello [15].

If the tensor $\boldsymbol{\eta}^{\mathsf{sym}}(\mathbf{p})$ is expressed in this $\mathbf{Q}$ and its dependence on $\mathbf{p}$ is additionally constrained to yield the same tensor at configurations that are expressed the same in $\mathbf{Q}$ (i.e. $\boldsymbol{\eta}^{\mathsf{sym}}(\mathbf{Q}_{\mathbf{p}})$) it automatically becomes an *a priori* symmetric tensor in any coordinate system $\mathbf{R}$. This is verified using Equation 6.4 to express $\boldsymbol{\eta}^{\mathsf{sym}}(\mathbf{Q}_{\mathbf{p}})$ as the candidate tensor in Equation 6.7 according to

$$\left[{\eta^{\mathsf{sym}}}_{ij}(\mathbf{p})\right]_{\mathbf{RR}} = \frac{1}{h}\sum_{\hat{s}\in G} \underbrace{\left[J_i^{i'}(\mathbf{p}_{\hat{s}})\right]_{\hat{s}\mathbf{R}}^{\mathbf{Q}} \left[\eta_{i'j'}(\mathbf{Q}_{\mathbf{p}})\right]_{\mathbf{QQ}} \left[J_j^{j'}(\mathbf{p}_{\hat{s}})\right]_{\hat{s}\mathbf{R}}^{\mathbf{Q}}}_{\left[{\eta^{\mathsf{sym}}}_{ij}(\mathbf{p}_{\hat{s}})\right]_{\hat{s}\mathbf{R}\hat{s}\mathbf{R}}}, \tag{6.10}$$

where $\hat{s}\mathbf{Q}$ has been substituted with $\mathbf{Q}$ since they are the same by definition ($\mathbf{Q}_{\mathbf{p}_{\hat{s}}} = \mathbf{Q}_{\mathbf{p}}$). Since the elements of $\left[J_i^{i'}(\mathbf{p}_{\hat{s}})\right]_{\hat{s}\mathbf{R}}^{\mathbf{Q}}$ are the same for any $\hat{s}$ due to substitution (see Equation 6.3), all terms in the sum are equivalent and $\boldsymbol{\eta}^{\mathsf{sym}}(\mathbf{p})$ passes the symmetry test.

Here at symmetry equivalent configurations $\mathbf{p}$ and $\mathbf{p}_{\hat{s}}$, the evaluation of $\boldsymbol{\eta}$ in the $\mathbf{Q}$ basis is the same, however, while $Q^i_{\mathbf{p}_{\hat{s}}} = Q^i_{\mathbf{p}}$, the Jacobian tensor (and thus the partial derivatives) are not. In fact, this exactly accounts for the transformation that is expected to occur on the tensor after a symmetry operation.

The gain of using this method is that the tensor needs to be evaluated only once and not for every symmetry operator as would be the case for the *a posteriori* method, speeding up the evaluation by a factor equal to the number of available symmetries.

Naively, one may expect to be able to fit any tensorfield using this method, however this is not true for tensors of order two or higher. In those cases, the tensorfields transform as products of lower order tensorfields and thus also allow for asymmetric contributions when the product is again symmetric. Thus using only a symmetric mapping yields to the following problem. If the partial derivative $\frac{\partial Q_i(\mathbf{R})}{\partial \mathbf{R}_j} = 0$ for any element, which always occurs exactly where the space is symmetric since moving from such a symmetry seam yields the same value for $\mathbf{Q}$ in either direction (yielding zero derivatives), a cross like shape of zeros is formed in a second order tensor due to multiplying from both sides with a Jacobian which has a row of zeros. However, non-zero values are allowed at the crossings of these rows and columns of zeros if one also includes the product of asymmetric contributions. As a consequence, this method is not able to fit all possible tensorfields. It is important to note that these crosses of zeros only appear if some index of the tensor is along the same direction as the partial derivative, otherwise, they are hidden as a linear dependence of two (or more) such directions. In those cases, they still prevent this method from fitting all possible tensors. This issue can be solved using asymmetric transformations.

Symmetry Adapted Coordinate Mapping with Asymmetric Transformations

To solve the issue for tensors of order two or higher, following solution is proposed.

Assume that a set of n tensor fields $\left\{ \boldsymbol{J}^{\mathsf{A},\mathsf{B},\mathsf{C},..}(\mathbf{p}) \right\}$ exists such that any symmetry operator $\hat{s}$ only creates a permutation of the set and that at any $\mathbf{p}$, for any element ij, at least one tensor element of the tensors in this group is non zero. Moreover, the sum of these tensor fields are required to yield a tensor field that imposes the same constraints as the Jacobian tensor field in the symmetry adapted coordinate mapping does.

We can now define a tensor, for which we will check whether it is invariant under the symmetrisation procedure

$$\left[\eta^{\mathsf{test}}{}_{ij}\left(\mathbf{p}\right)\right]_{\mathbf{RR}} \quad = \quad \frac{1}{n}\sum_{\mathsf{a}}^{\mathsf{A,B,C,..}} \left[J^{\mathsf{a}}{}^{i'}_{i}\left(\mathbf{p}\right)\right]^{\mathbf{Q}}_{\mathbf{R}} \left[\eta_{i'j'}\left(\mathbf{Q_p}\right)\right]_{\mathbf{QQ}} \left[J^{\mathsf{a}}{}^{j'}_{j}\left(\mathbf{p}\right)\right]^{\mathbf{Q}}_{\mathbf{R}}. \tag{6.11}$$

Now since the symmetry operator $\hat{s}$ permutes only the terms of the sum in eq. 6.11, the test tensor is invariant under the symmetrisation and is thus already symmetric. The asymmetric tensor model thus consists of asymmetric terms that after addition form a symmetric model even when symmetry operators are applied. It should be noted here that the individual assymetric terms are technically not tensors themselves as only the sum of all terms transforms as a tensor.

6.3 Machine Learning Models for Orbital-Dependent Friction Tensors

This section describes the details on how to implement the *a priori* and *a posteriori* symmetric tensor fields in a machine-learning framework.

Neural Network architectures for tensor fields that reproduce the correct symmetry behaviour result in complex designs. First the hidden layer, which contains all the fitting parameters is explained. Then two different ways to obtain a order two tensor from such a hidden layer are shown. Next, a method is given to obtain a symmetric order two tensor (this is not the same as a tensor field with the correct symmetry behaviour), i.e. the two indices permute $\eta_{ij} = \eta_{ji}$. Finally, the correct symmetry behaviour is obtained by either using *a priori* or *a posteriori* architectures.

In this chapter, neural networks are a part of the building blocks used to design the algorithms for continuous representations of tensors. As such, these building blocks are henceforth refered to as hidden networks. The hidden networks used in this chapter (see Figure 6.2c) are constructed from L hidden layers (see Figure 6.2b) L^{l} labeled with l together with an output layer $\mathsf{L}^{\mathsf{out}}$. Each layer consists of N_l nodes (see Figure 6.2a) $\mathsf{N}^{l}_{n_l}$ where the output of the n_lth node defines the n_lth output of the lth layer. The number of inputs (M_l) in each layer is equal to the number of outputs from the

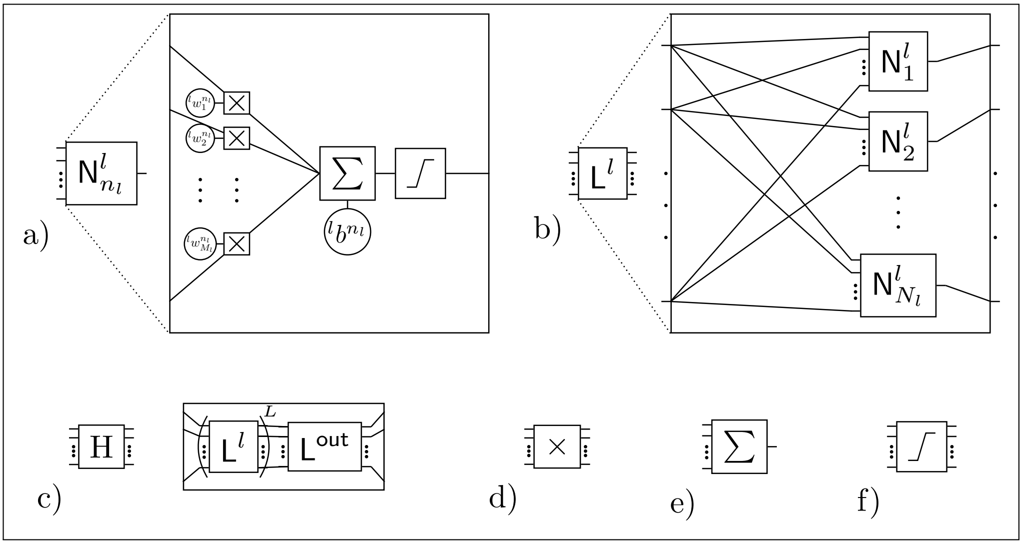

Figure 6.2: Schematic representation of the structure of a hidden network. a) Shows the inner structure of a node. b) Shows how nodes make up a layer. c) Shows how consequetive hidden and output layers make up a hidden network. d) Multiplication operator: multiplies two numbers. e) Sum operator that takes any amount of inputs and yields the sum. f) Activation function $F^(l)$.

previous layer $M_l = M_{l-1}$ except for the first layer which uses the N_0 inputs to the hidden network. Inputs of layer l are used for each node N_l in that layer and are first multiplied (see Figure 6.2d) with the weights ${}^lw^{n_l}_{m_l}$ after which the result is summed (see Figure 6.2e) and added to the bias ${}^lb^{n_l}$. Finally this sum is used as the argument of an activation function (see Figure 6.2f) F^l according to

$$y_{n_l} = F^l\left({}^lb^{n_l} + \sum_{m_l}^{M_l} {}^lw^{n_l}_{m_l} x_{m_l}\right). \tag{6.12}$$

The activation function $F^l(x)$ is chosen as the hyperbolic tangent $\tanh(x)$ for all hidden layers and as the linear function $f(x) = x$ for the output layer. Only the weights and biases are optimized during the training routine and not the architecture itself.

In order to obtain the 21 independent elements of an electronic friction tensor (in 6D), it is necessary to either fit one hidden network with 21 outputs or 21 hidden networks with 1 output as is shown by the choice between option 1a and 1b in Figure 6.3. Once the 21 independent elements are obtained, they have to be arranged either as a symmetric tensor directly or as a lower triangular tensor and subsequently multiplied with its transpose. The latter method yields a positive-definite tensor which ensures that for any velocity (direction and magnitude) a net energy loss is obtained.

6.3.1 Reduced Dimensional Mapping

A reduced dimensional mapping has been shown to work [21, 24] previously for H$_2$ on Cu(111) and N$_2$ on Ru(0001). Here the Cartesian coordinates of the two atoms A and B with coordinates X_A, Y_A, Z_A and X_B, Y_B, Z_B are transformed first to the molecular coordinate system $\mathbf{R}^{\mathsf{sph}} = \{X, Y, Z, d, \theta, \phi\}$, where

$$X = \frac{m_A}{M} X_A + \frac{m_B}{M} X_B, \tag{6.13a}$$

$$Y = \frac{m_A}{M} Y_A + \frac{m_B}{M} Y_B, \tag{6.13b}$$

$$Z = \frac{m_A}{M} Z_A + \frac{m_B}{M} Z_B, \tag{6.13c}$$

$$d = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2 + (Z_B - Z_A)^2}, \tag{6.13d}$$

$$\theta = \arccos\left(\frac{Z_B - Z_A}{d}\right), \tag{6.13e}$$

$$\phi = 2\pi\mathcal{H}\left(Y_A - Y_B\right) + \operatorname{sign}\left(Y_B - Y_A\right) \arccos\left(\frac{X_B - X_A}{\sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2}}\right), \tag{6.13f}$$

where $\mathcal{H}(x)$ is the heaviside step function and $\operatorname{sign}(x)$ yields the sign of x. Note that for H$_2$, N$_2$ and other homo-nuclear diatomic molecules X$_2$ consisting of identical isotopes the mass fractions are always one half since the mass of atom A and B are equal ($m_A = m_B = m$) and the total mass M is thus equal to $2m$. Figure 6.4 shows how the Cartesian coordinates in X are transformed to the molecular coordinates in Q. Next the reference molecular coordinate is computed by setting both angles to $\theta_{\mathsf{ref}} = \phi_{\mathsf{ref}} = 90°$. The Jacobian is then computed for both the reference and original coordinate and the matrix product of these two Jacobians yields a transformation

6

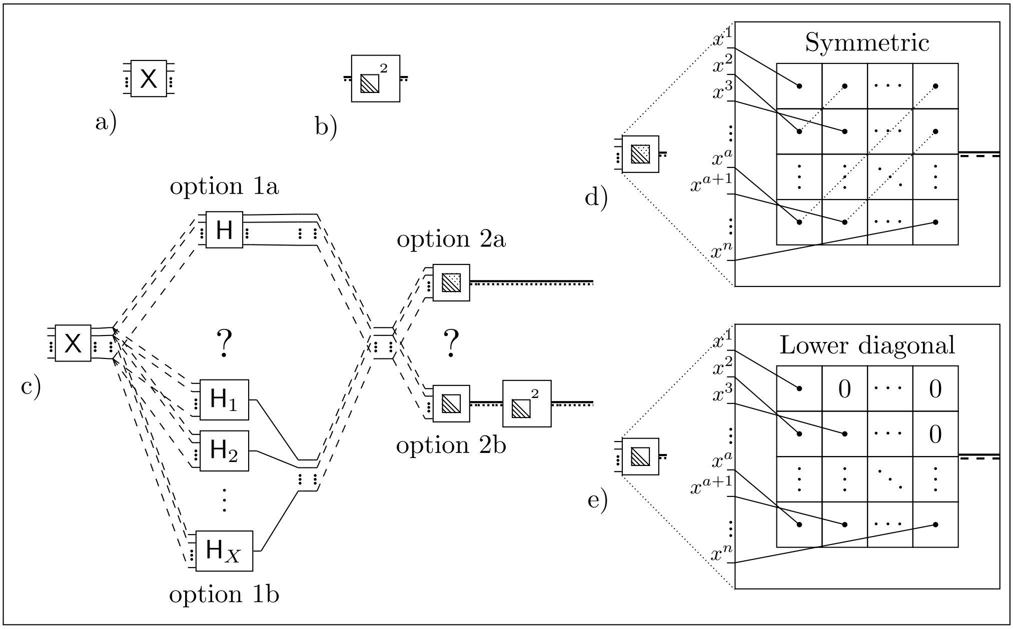

Figure 6.3: Schematic representation of how to use a hidden network for fitting a tensor field. a) Input vector to the NN(e.g. Cartesian coordinates). b) Multiplies a lower triangular matrix with its transpose. c) A tensor field is obtained by making two choices. First, the independent elements can be obtained using (1a) one hidden network with as many outputs as required elements or (1b) an hidden network with one output is used for every independent element. Second, the elements can be arranged directly in a symmetric fashion (2a) or a lower triangular can be obtained which is subsequently multiplied with its transpose to obtain a symmetric positive-definite tensor (2b). d) Shows the direct symmetric arrangement. e) Shows the lower triangular arrangement.

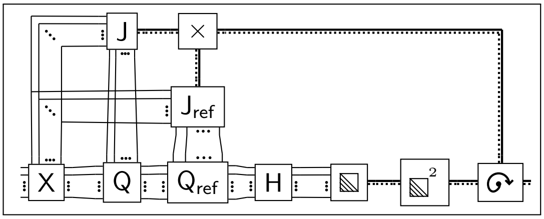

Figure 6.4: Schematic representation of the reduced dimensional mapping model used in Chap. 4 and Chap. 5 [21, 24]

.

tensor $\boldsymbol{T}(\mathbf{p}, \mathbf{p}_{\mathsf{ref}})$. Finally $\boldsymbol{T}(\mathbf{p}, \mathbf{p}_{\mathsf{ref}})$ is used to transform the output of the hidden network and subsequent positive definite arrangement. This transformation in this particular case is the same as taking the friction elements along the bond (d) of the molecule to be constant regardless of the chosen angular orientation. The coordinates that are given to the hidden network are $Q_{5\ldots8}$ from Eq. 6.15 (*vide infra*) to also account for periodicity.

6.3.2 Piece-wise Mapping

Jiang and co-workers [22, 23] have shown that it is possible to obtain an accurate fit using a piece-wise mapping model. They have described this as applying subsequent symmetry operations until the mapping to the irreducible wedge is found. Here a slightly different approach is taken that yields the same result for a homonuclear-diatomic molecule X_2 consisting of identical isotopes.

To perform the mapping procedure, we first find the surface atom closest to the X_2 COM position and translate X_2 such that all molecules and up near the same surface atom. Secondly, the angle α_{map} is defined as the angle between the x-axis and the axis going through the formar surface atom and the X_2 COM projected on the surface plane. The X_2 molecule and concomitant friction tensor is then transformed according to the symmetry operator $\hat{s}_{?}$ defined by the following case statement:

$$\hat{s}_{?} = \begin{cases} \hat{E} & \text{for } -\frac{1}{6}\pi < \alpha_{\mathsf{map}} \leq \frac{1}{6}\pi, \\ \hat{\sigma}_4 & \text{for } \frac{1}{6}\pi < \alpha_{\mathsf{map}} \leq \frac{3}{6}\pi, \\ \hat{C}_3^1 & \text{for } \frac{3}{6}\pi < \alpha_{\mathsf{map}} \leq \frac{5}{6}\pi, \\ \hat{\sigma}_2 & \text{for } \frac{5}{6}\pi < \alpha_{\mathsf{map}} \leq -\frac{5}{6}\pi, \\ \hat{C}_3^2 & \text{for } -\frac{3}{6}\pi < \alpha_{\mathsf{map}} \leq \frac{1}{6}\pi, \\ \hat{\sigma}_3 & \text{for } -\frac{5}{6}\pi < \alpha_{\mathsf{map}} \leq -\frac{3}{6}\pi. \end{cases} \tag{6.14}$$

These boundaries assign a different symmetry operator $\hat{s}_{?}$ depending at which of the 6 irreducable wedges (of the hexaganol surface) the X_2 COM is located. Applying the symmetry operator $\hat{s}_{?}$ then always maps the X_2 COM to the same irreducable wedge, namely the irreducable wedge that gets assigned the identy operator $\hat{E}$.

Jiang and co-workers also apply a permutation to the friction tensor of the coordinates associated with atom A and B if $Z_A > Z_B$. This method in principle takes care of the permutation symmetry, however, not correctly when the molecule is perpendicular to the surface. In practice, this is not a big issue for performing dynamics since the probability of being exactly perpendicular is very small. For the training set however, a significant amount of configurations where the molecule was exactly perpendicular was used thus amplifying this issue. The strategy of Jiang and co-workers has been adapted in this thesis by finding the Cartesian direction along which the separation between the two atoms is largest, and apply the permutation selection to that Cartesian direction instead of always along Z. This method is denoted as IW (irreducible wedge) henceforth. Additionally, a more simplified piece-wise mapping scheme has been used where the X_2 molecule is always mapped to a single unit cell (SUC). Using this method, the *a posteriori* method only requires a sum over all C_{3v} and permutation symmetries and not the translational symmetries (as those are included in the mapping). Note that a purely translational mapping does not introduce a transformation of the tensor, tensors elements are invariant under this operation.

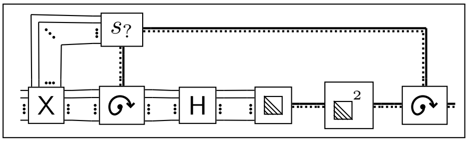

Figure 6.5: Schematic representation of the piece-wise mapping method.

6.3.3 Symmetry Adapted Coordinate Mapping

For the symmetry adapted coordinate mapping used in this thesis, the following symmetry adapted coordinates [27] are used

$$Q_1 = \frac{1}{2}\left[\exp\left(\frac{Z_A}{2}\right)\cdot g_1\left(X_A, Y_A\right) + \exp\left(\frac{Z_B}{2}\right)\cdot g_1\left(X_B, Y_B\right)\right] \tag{6.15a}$$

$$Q_2 = \exp\left(\frac{Z_A}{2}\right)\cdot g_1\left(X_A, Y_A\right)\cdot \exp\left(\frac{Z_B}{2}\right)\cdot g_1\left(X_B, Y_B\right) \tag{6.15b}$$

$$Q_3 = \frac{1}{2}\left[\exp\left(\frac{Z_A}{2}\right)\cdot g_2\left(X_A, Y_A\right) + \exp\left(\frac{Z_B}{2}\right)\cdot g_2\left(X_B, Y_B\right)\right] \tag{6.15c}$$

$$Q_4 = \exp\left(\frac{Z_A}{2}\right)\cdot g_2\left(X_A, Y_A\right)\cdot \exp\left(\frac{Z_B}{2}\right)\cdot g_2\left(X_B, Y_B\right) \tag{6.15d}$$

$$Q_5 = \exp\left(\frac{Z}{2}\right)\cdot g_1\left(X, Y\right) \tag{6.15e}$$

$$Q_6 = \exp\left(\frac{Z}{2}\right)\cdot g_2\left(X, Y\right) \tag{6.15f}$$

$$Q_7 = \exp\left(\frac{Z}{2}\right) \tag{6.15g}$$

$$Q_8 = d \tag{6.15h}$$

$$Q_9 = \cos\left(\theta\right)^2, \tag{6.15i}$$

which are based on the Cartesian and molecular coordinates.

The g_1 and g_2 generate a unique coordinate for every symmetry equivalent x and

y along a C_{3v} surface (e.g. Ru(0001) and Cu(111)) according to

$$g_1(x,y) = g\left(x - \frac{a}{4}, y - \frac{a}{4\sqrt{3}}\right) \tag{6.16a}$$

$$g_2(x,y) = g\left(x + \frac{a}{4}, y + \frac{a}{4\sqrt{3}}\right) \tag{6.16b}$$

$$\begin{aligned} g(x,y) = \frac{1}{3\sqrt{3}}\Bigg[&\sin\left(\frac{2\pi}{a}\left(x - \frac{1}{\sqrt{3}}y\right)\right) + \\ &\sin\left(\frac{2\pi}{a}\left(\frac{2}{\sqrt{3}}y\right)\right) + \sin\left(\frac{2\pi}{a}\left(x + \frac{1}{\sqrt{3}}y\right)\right)\Bigg] + \frac{1}{2}. \end{aligned} \tag{6.16c}$$

Using this coordinate system, a tensor field with the correct symmetry behavior is obtained (see Figure 6.6a) by transforming to the $\mathbf{Q}$ coordinate system and transforming the result of the hidden network from $\mathbf{Q}$ to $\mathbf{R}$.

6.3.4 Symmetry Adapted Coordinate Mapping with Asymmetric Transformations

To include asymmetric contributions, the following alternative g functions, based on seperating the different terms in g from Eq. 6.16c, are defined

$$g^{\text{one}}(x,y) = \frac{1}{3\sqrt{3}}\left[\sin\left(\frac{2\pi}{a}\left(x - \frac{1}{\sqrt{3}}y\right)\right)\right] \tag{6.17a}$$

$$g^{\text{two}}(x,y) = \frac{1}{3\sqrt{3}}\left[\sin\left(\frac{2\pi}{a}\left(\frac{1}{\sqrt{3}}y\right)\right)\right] \tag{6.17b}$$

$$g^{\text{three}}(x,y) = \frac{1}{3\sqrt{3}}\left[\sin\left(\frac{2\pi}{a}\left(x + \frac{1}{\sqrt{3}}y\right)\right)\right]. \tag{6.17c}$$

These g functions define corresponding coordinate systems Q and Jacobian tensors $\boldsymbol{J}(\mathbf{p})$ that are used to transform the same output of the hidden network in Figure 6.6b and an average of these transformed results is taken. Note that the hidden network is evaluated only once, which results in a significant performance increase compared to the *a posteriori* method.

Any of the symmetry operators in the C_{3v} group either cause the swapping of two of these Jacobian tensors J, which changes the permutation order, or reorder them by pulling the last one to the front, which keeps the permutation order. In both cases, the averaging of the transformed result remains the same and thus this representation is symmetry invariant.

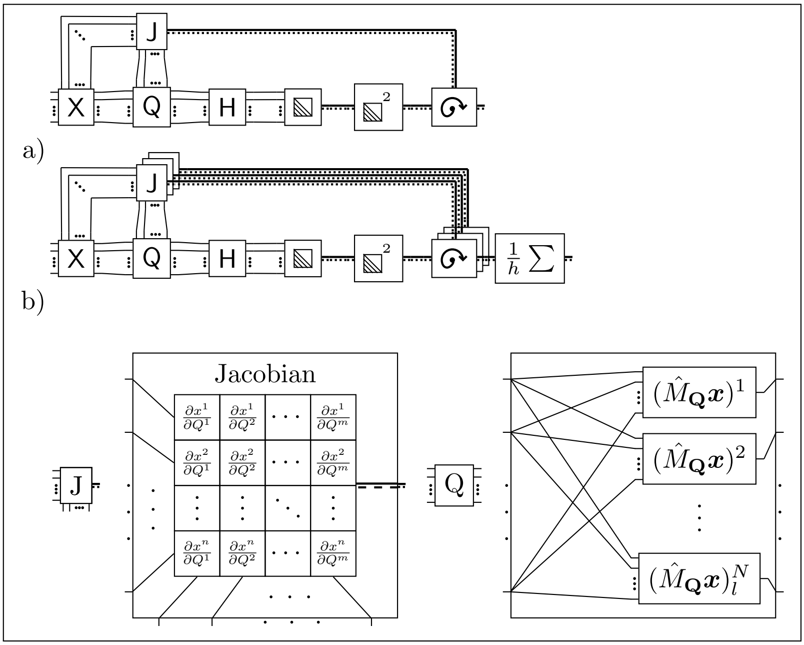
a)
b)
J
X
Q
H
$\frac{1}{h}\sum$
Jacobian
$\frac{\partial x^1}{\partial Q^1}$
$\frac{\partial x^1}{\partial Q^2}$
$\frac{\partial x^1}{\partial Q^m}$
$\frac{\partial x^2}{\partial Q^1}$
$\frac{\partial x^2}{\partial Q^2}$
$\frac{\partial x^2}{\partial Q^m}$
$\frac{\partial x^n}{\partial Q^1}$
$\frac{\partial x^n}{\partial Q^2}$
$\frac{\partial x^n}{\partial Q^m}$
$(\hat{M}_{\mathbf{Q}}\boldsymbol{x})^1$
$(\hat{M}_{\mathbf{Q}}\boldsymbol{x})^2$
$(\hat{M}_{\mathbf{Q}}\boldsymbol{x})_l^N$

Figure 6.6: Schematic representation of the symmetry adapted coordinate mapping either without (a) or with (b) asymmetric transformations.

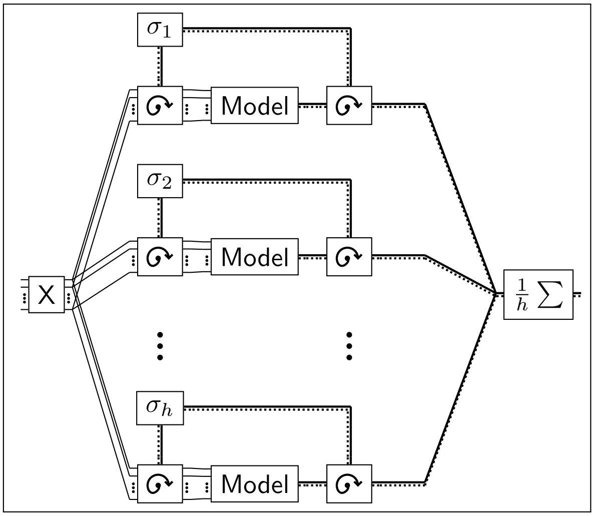

Figure 6.7: A schematic representation of the *a posteriori* architecture.

6.3.5 *a posteriori* Imposed Symmetry Behavior

The *a posteriori* method is implemented as shown schematically in Figure 6.7. For each symmetry operator in the C_{3v} group, the input coordinates are transformed and given to any of the previously discussed models. The result of these models are subsequently transformed and averaged.

6.4 Computational Details

TensorFlow [28] has been used for fitting the electronic friction tensors. The fitting parameters of the models were initialized using a normal distribution with $\mu = 0$ and $\sigma = \frac{1}{N_l}$ (N_l is the number of nodes in a specific layer) for the weights and $\sigma = 0.1$ for the biases. Next the RMSProp algorithm [29] is used in three consecutive runs of 100.000 epochs using a decay of 0.997 and a learning rate of 10^{-3}, 10^{-4}, and 10^{-5} for the first, second, and third run respectively. A training and test set was obtained by randomly shuffling the order of electronic friction tensors in the data set and assigning the first 10% of the data entries to the test set and the remaining data entries to the training set. Only the training set was used during the learning stage. A so-called committee

[30, 31] with 5 committee members was created by repeating all fitting procedures for each member with the same settings, but with different randomly initialized weights and biases. The training and test set shuffling was different for different committee member It has been kept the same for all machine learning models such that the respective committees are all presented the same training and test sets for the same member.

6.5 Results

Here a comparison is made between the different methods of obtaining continuous representations of friction tensors as discussed previously based on the root-mean-square errors (RMSEs). For each data set, both the RMSE of the training set and test set has been computed (see Sec. 6.4). The argument here is that the RMSE of the test set is a measure of how well the method is at continuously representing. The RMSE of the training set on the other hand shows the flexibility of the method. In particular, if the symmetry of the coordinate dependence is not correctly taken into account *a priori*, the training set may still yield a low RMSE if the data points in the training set are not sampled from different irreducible wedges. In contrast, the test set, which for this chapter is always replicated to other irreducible wedges, does yield a high RMSE if this symmetry is not taken into account properly.

All friction tensors discussed here are for N_2 on Ru(0001), where only the N_2 degrees of freedom are considered and the Ru(0001) surface is kept frozen in the hexagonal symmetry (including surface expansion in the lateral directions and interlayer relaxation that is representative for $T_s = 575$K). Ru(0001) is kept at its ideal configuration (but includes surface expansion and layer relaxation for 575K). Since the electronic friction tensor is symmetric, it has 21 independent elements for the 6 degrees of freedom of the N_2 molecule.

6.5.1 Reduced-Dimensional Models: Machine Learning Parameters

For this comparison, the data set of electronic friction tensors from Ref. [24] has been used, which directly accounts for the dependence of four of the six molecular coordinates. This data set has subsequently been fitted using the reduced dimensional method. This allows one to focus on how the friction tensor itself is represented rather than how the symmetry of the tensor field (i.e., the symmetry of the coordinate dependence) is included. Here the result of using one NN that produces the 21 independent elements as its output is compared with using 21 independent NNs (options 1a and 1b from Fig. 6.3, respectively). Moreover, a comparison is made between the procedure of obtaining 21 independent elements and arranging them in a symmetric tensor directly, and of obtaining 21 independent elements in a lower triangular tensor and subsequently squaring this tensor to obtain a positive-definite tensor (options 2a and 2b in Fig. 6.3, respectively). Finally, results obtained with different numbers of layers and nodes are compared with reference to the number of free parameters needed to construct the corresponding NN topologies.

Figure 6.8 shows the RMSEs for NN fits for directly obtaining a symmetric tensor (option 2a). When using a different NN for each element (option 1a), there is no substantial difference for the RMSE between using two or three NN layers over the wide range of NN topologies considered here. On the other hand, if a single NN to fit the entire tensor is used (option 1b) the RMSEs are significantly larger for two NN layers (not reported here). Apparently, in this case two layers are not enough and the additional NN layer is required to describe the difference between the different elements, in contrast to option 1a where using different NNs can provide this flexibility. The additional third layer under option 1a simply provides more fitting parameters, which is supported by the similar behavior of the RMSE as a function of the number of fitting parameters for both two and three layers as plotted in Fig. 6.8. Using option 1b with three layers performs significantly better than both two and three layers with option 1a with a much smaller number of fitting parameters. The minimum error is found with 15 nodes per layer (resulting in 891 fitting parameters in total). Option 1a with both two and three hidden layers only performs similar to option 1b f the number of fitting parameters has the same order of magnitude as the number of data points.

Figure 6.9 shows the RMSEs for obtaining a lower triangular tensor and subsequently squaring it (option 2b). The RMSEs of option 1a with both two and three layers are again very similar in their behavior as a function of the number of fitting parameters. However, the range of the RMSE is slightly smaller as it is between 0.21 and 0.255 meVpsÅ^{-2} instead of 0.21 and 0.285 meVpsÅ^{-2} when comparing to the corresponding curves in Fig. 6.8. Option 1b on the other hand yields a higher RMSE if more than 2000 fitting parameters are used when combined with option 2b (Fig. 6.9) instead of option 2a (Fig. 6.8). For 10 and 15 nodes the results are essentially the same, in particular when considering the scattering over the respective neural network committees, while 21 and 25 nodes now give a lower RMSE with option 2b, with 21 nodes being the best fit.

The need for increasing the number of fitting parameters from 891 to 1971 in order to obtain the best fit when using option 2b over 2a is not surprising. When using option 2b the friction tensor is more constrained by construction and thus a larger number of parameters is required to obtain the same flexibility.

For the data set used here it is clear that using option 1b instead of 1a is clearly better. The same NN used to obtain the entire friction tensor yields better results for substantially fewer fitting parameters. Moreover, option 2b instead of 2a provides additional advantages: although the improvement to the RMSE is only 0.02 meVpsÅ^{-2} and comes at the cost of needing more fitting parameters, it ensures a positive definite and thus physically meaningful friction tensor by construction. This property is important for using friction tensors in the generalized Langevin equation (GLE) to perform molecular dynamics with electronic friction. Option 2a does not ensure that the result of the friction tensor is a dissipation of energy, it is possible that for some configurations the NN will predict an energy gain of the sy stem.

It is not clear whether these results generalize to other data sets. It is suggested here to use option 2b over option 2a as long as the RMSE is not influenced significantly. It is suspected here that option 1b will generally perform better than option 1a for the same number of fitting parameters, since the different friction tensor elements depend

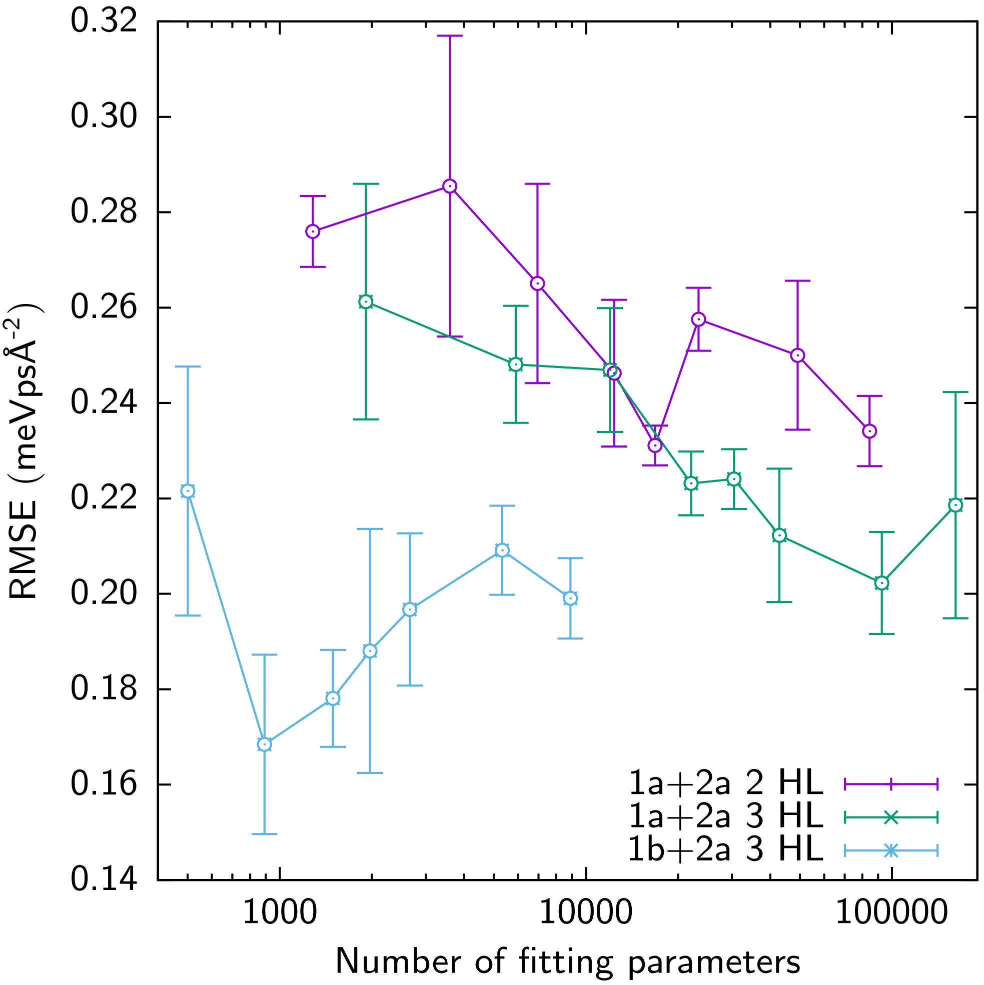

Figure 6.8: RMSE of the test set for the combination of options 1a+2a (details see text and Fig. 6.3) with 2 hidden layers (purple), options 1a+2a with 3 hidden layers (green), and options 1b+2a with 3 hidden layers (blue) as a function of the number of parameters (neural network weights and biases, on a logarithmic scale). Data points correspond to 5, 10, 15, 21, 25, 30, 45, and 60 nodes for each hidden layer in this order for option 1a while 5 nodes is ommited for option 1b due to too large errors. Error bars show the spread of RMSEs for different committee members (expressed as a RMSE itself). Note that since only a single NN is employed for the same amount of hidden nodes the blue curve corresponds to a factor of 20 fewer fitting parameters.

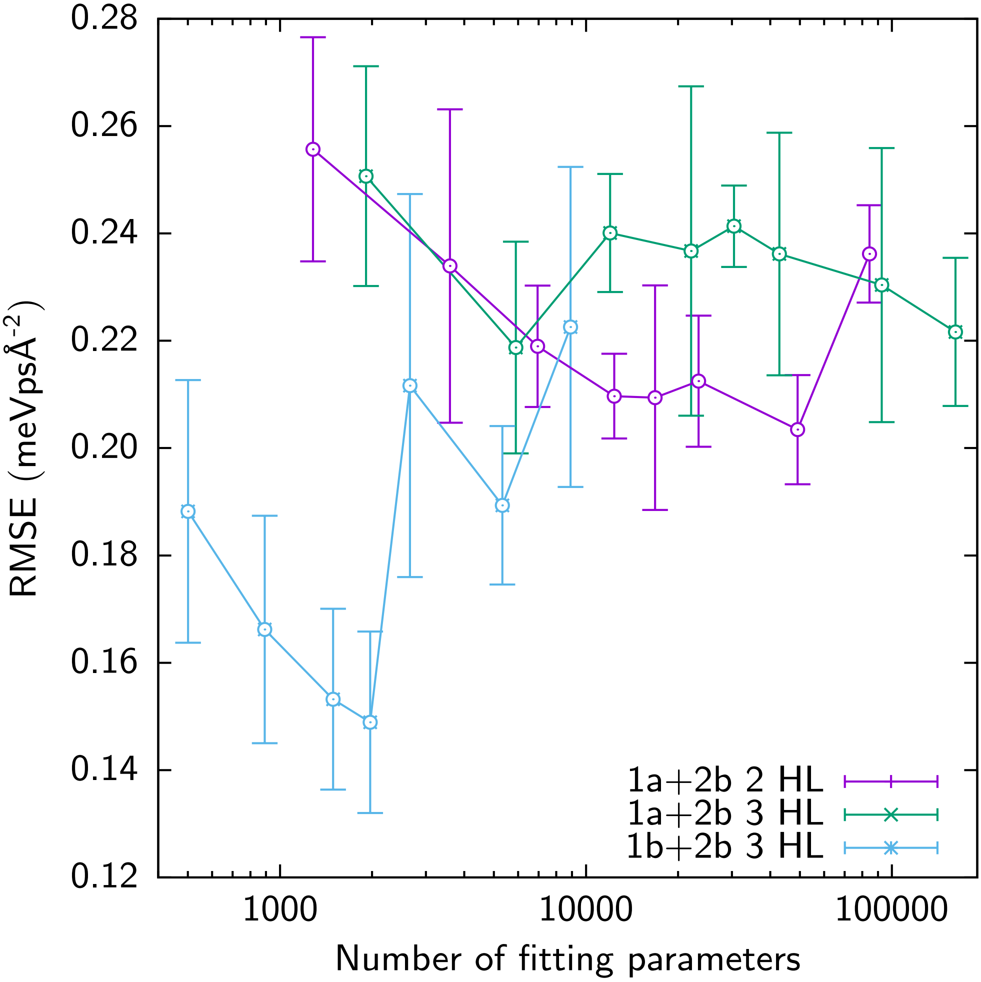

Figure 6.9: Same as Fig. 6.8, but for the combination of options 1a+2b with 2 hidden layers (purple), options 1a+2b with 3 hidden layers (green), and options 1b+2b with 3 hidden layers (blue).

similarly on the input coordinates. For example, all elements will be small when the molecule is far away from the surface. Using option 1b this information needs to be 'understood' by the NN only once while for option 1a this needs to be learned by all independent NNs. For results henceforth, option 1b is used in combination with 2b, together with three NN layers each with 21 nodes (using 25 nodes yields a comparatively small increase in fitting accuracy for a large increase in fitting parameters).

6.5.2 Six-Dimensional Symmetry Adapted Models

In order to assess the performance of the different symmetry adapted models, a 6D data set has been fitted with each method.

Fig. 6.10 shows the RMSEs of both the training and test set. Additionally, the amount of time needed for fitting is provided in hours per processor core. The single unit cell (SUC) piece-wise mapping (see Sec. 6.2.1) has been fitted in three ways: without any additional considerations (SUC in Fig. 6.10), using the *a posteriori* method (SUC-P) and by replicating the training set using the available symmetry operations (SUC-D). The SUC test RMSEs are the worst at 0.44 meVpsÅ^{-2}even though the training error is satisfactory. This is not a result of over-fitting, instead this is because the test set contains data from other irreducible wedges not included in the training set. Since there is no information available to the NN on how the friction tensors in different irreducible wedges are related and because there is not necessarily training data available in all regions, it is not possible for the NN to make good predictions. Using the SUC-P method substantially improves the description of the test RMSE at a slight setback of the training RMSE. Similar to previous arguments, this can be explained by the *a posteriori* method constraining the NN prediction which reduces flexibility. Inspection of the elements as a function of movement of N_2 along the Ru(0001) surface shows no jumps at the symmetry seams for SUC-P as expected. Replicating the training set for different irreducible wedges improves the test RMSE in the SUC-D method only slightly over the SUC method. While the SUC-D does have symmetry information available in the form of symmetrically replicated data, it misses the exact relation between and constraints of the individual elements, which is enforced with the SUC-P.

The test and training RMSEs for all piece-wise mapping schemes based on the irreducible wedge (IW, see Sec. 6.3.2) are essentially the same (see Fig. 6.10). This is not surprising for IW and IW-D as the IW method already correctly takes into account symmetry. However, it was expected that IW-P, using the *a posteriori* method, no longer has the discontinuities at the symmetry seams present in IW. It turns out that they are still present since computational rounding errors (due to finite floating piont precision of the coordinates) can assign the N_2 molecule to the wrong irreducible wedges. This happens only when a data point is extremely close to the symmetry seam. No simple solution was found to circumvent these errors due to the large number of computations needed to obtain the mapping between irreducible wedges.

For the sym and asym methods (see Sec. 6.3.3 and Sec. 6.3.4 respectively) no results for the *a posteriori* and training data replication are shown in Fig. 6.10 because, as expected, the results are identical (only an increased computational effort was found). The sym method performs substantially worse due to the additional (nonphysical) constraint introduced along some directions at the symmetry seam which are not present in the asym method. Asym yields the smallest test RMSE (0.28 meVpsÅ^{-2}) with a similar training RMSE (0.20 meVpsÅ^{-2}) indicating that an accurate fit was obtained. There are no discontinuities at the symmetry seam for either method as the functional forms of these methods are completely continuous.

Generally the piece-wise models (SUC and IW) are computationally more expensive due to conditional statements present in the algorithm for mapping back to surface unit cell or irreducible wedge. The *a posteriori* method and training data replication method both increase the computational effort. Due to the additional input coordinate transformations and output tensor transformations necessary in the *a posteriori* method it is the most expensive. The *a priori* methods based on symmetry-adapted coordinates (sym and asym) are the computationally cheapest methods.

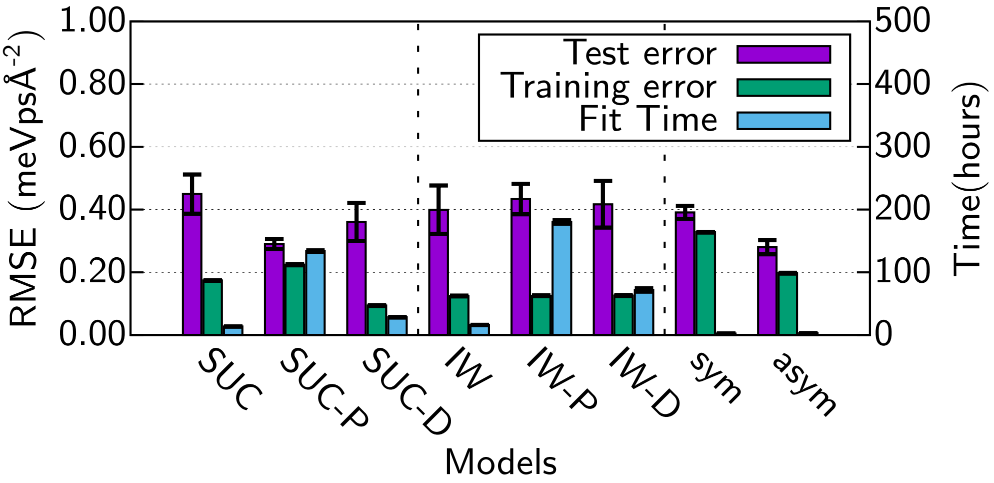

Figure 6.10: RMSE of test (purple) and training (green) data sets for several fitting models (see explanations in text) as well as concomitant fitting times (blue) in hours per processor core.

6.6 Conclusions

This chapter has introduced several strategies for obtaining continuous representations of friction tensors based on the properties of tensors under coordinate transformations. Obtaining all friction tensor elements using a single neural network with multiple outputs results in more accurate fits and additionally requires a substantially smaller number of fitting parameters. Positive-definite friction tensors were obtained by generating a lower triangular tensor and subsequently squaring it, which did not introduce additional difficulties for fitting the friction tensors compared to obtaining symmetric tensors only.

The piece-wise mapping procedure targetting the irreducible wedge, which is based on the work of Jiang and coworkers [22], yields accurate fits of the electronic friction tensors, albeit without exact compliance with the exact tensorial symmetry. Future dynamical studies are likely to confirm that the small concomittant RMSEs do not (significantly) affect calculated observables. From an analytical point of view these discontinuities should be removed if the *a posteriori* method is used, this turned out to not work in practice due to computational rounding errors. As expected, the symmetry adapted machine learning model developed here properly captures the symmetry of the friction tensor. However, additional constraints are imposed which prevent an accurate fit in some regions. Most notably, constraints at the symmetry seams that should only be applicable for off-diagonal elements are also present for diagonal elements. The asymmetric tensor model has none of the above problems and yields the most accurate fit. Surprisingly, taking a simplified piece-wise mapping model targetting the entire surface unit cell yields almost the same accuracy when combined with the *a posteriori* method.

References

[1] J. Behler and M. Parrinello. “Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces”. In: *Phys. Rev. Lett.* 98 (2007), p. 146401. DOI: 10.1103/PhysRevLett.98.146401.

[2] J. Behler. “Representing Potential Energy Surfaces by High-Dimensional Neural Network Potentials”. In: *J. Phys. Condens. Matter* 26 (2014), p. 183001. DOI: 10.1088/0953-8984/26/18/183001.

[3] J. Behler. “Constructing High-Dimensional Neural Network Potentials: A Tutorial Review”. In: *Int. J. Quantum Chem.* 115 (2015), pp. 1032–1050. DOI: 10.1002/qua.24890.

[4] A. Singraber, J. Behler, and C. Dellago. “Library-Based LAMMPS Implementation of High-Dimensional Neural Network Potentials”. In: *J. Chem. Theory Comput.* (2019). DOI: 10.1021/acs.jctc.8b00770.

[5] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi. “Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons”. In: *Phys. Rev. Lett.* 104 (2010), p. 136403. DOI: 10.1103/PhysRevLett.104.136403.

[6] A. Grisafi, D. M. Wilkins, G. Csányi, and M. Ceriotti. “Symmetry-Adapted Machine Learning for Tensorial Properties of Atomistic Systems”. In: *Phys. Rev. Lett.* 120 (2018), p. 036002. DOI: 10.1103/PhysRevLett.120.036002.

[7] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld. “Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning”. In: *Phys. Rev. Lett.* 108 (2012), p. 058301. DOI: 10.1103/PhysRevLett.108.058301.

[8] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko. "Quantum-Chemical Insights from Deep Tensor Neural Networks". In: *Nat. Commun.* 8 (2017), p. 13890. DOI: 10.1038/ncomms13890.

[9] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. "SchNet – A Deep Learning Architecture for Molecules and Materials". In: *J. Chem. Phys.* 148 (2018), p. 241722. DOI: 10.1063/1.5019779.

[10] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, et al. "Machine Learning of Accurate Energy-Conserving Molecular Force Fields". In: *Sci. Adv.* 3 (2017), e1603015. DOI: 10.1126/sciadv.1603015.

[11] S. Chmiela, H. E. Sauceda, I. Poltavsky, K.-R. Müller, and A. Tkatchenko. "sGDML: Constructing Accurate and Data Efficient Molecular Force Fields Using Machine Learning". In: *Comp. Phys. Comm.* 240 (2019), pp. 38–45. DOI: 10.1016/j.cpc.2019.02.007.

[12] B. Jiang and H. Guo. "Permutation Invariant Polynomial Neural Network Approach to Fitting Potential Energy Surfaces. III. Molecule-Surface Interactions". In: *J. Chem. Phys.* 141 (2014), p. 034109. DOI: 10.1063/1.4887363.

[13] B. Kolb, X. Luo, X. Zhou, B. Jiang, and H. Guo. "High-Dimensional Atomistic Neural Network Potentials for Molecule–Surface Interactions: HCl Scattering from Au(111)". In: *J. Phys. Chem. Lett.* 8 (2017), pp. 666–672. DOI: 10.1021/acs.jpclett.6b02994.

[14] K. Shakouri, J. Behler, J. Meyer, and G.-J. Kroes. "Accurate Neural Network Description of Surface Phonons in Reactive Gas–Surface Dynamics: N_2 + Ru(0001)". In: *J. Phys. Chem. Lett.* 8 (2017), pp. 2131–2136. DOI: 10.1021/acs.jpclett.7b00784.

[15] J. Behler, S. Lorenz, and K. Reuter. "Representing Molecule-Surface Interactions with Symmetry-Adapted Neural Networks". In: *J. Chem. Phys.* 127 (2007), p. 014705. DOI: 10.1063/1.2746232.

[16] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko. "Towards Exact Molecular Dynamics Simulations with Machine-Learned Force Fields". In: *Nat. Commun.* 9 (2018), pp. 1–10. DOI: 10.1038/s41467-018-06169-2.

[17] M. Head-Gordon and J. C. Tully. "Molecular Dynamics with Electronic Frictions". In: *J Chem Phys* 103 (1995), pp. 10137–10145. DOI: 10.1063/1.469915.

[18] M. Alducin, R. Díez Muiño, and J. I. Juaristi. "Non-Adiabatic Effects in Elementary Reaction Processes at Metal Surfaces". In: *Prog. Surf. Sci.* 92 (2017), pp. 317–340. DOI: 10.1016/j.progsurf.2017.09.002.

[19] W. Dou and J. E. Subotnik. "Perspective: How to Understand Electronic Friction". In: *J. Chem. Phys.* 148 (2018), p. 230901. DOI: 10.1063/1.5035412.

[20] R. J. Maurer, B. Jiang, H. Guo, and J. C. Tully. “Mode Specific Electronic Friction in Dissociative Chemisorption on Metal Surfaces: H_2 on Ag(111)”. In: *Phys. Rev. Lett.* 118 (2017), p. 256001. DOI: 10.1103/PhysRevLett.118.256001. arXiv: 1705.09753 [cond-mat.mtrl-sci].

[21] P. Spiering and J. Meyer. “Testing Electronic Friction Models: Vibrational De-Excitation in Scattering of H_2 and D_2 from Cu(111)”. In: *J. Phys. Chem. Lett.* 9 (2018), pp. 1803–1808. DOI: 10.1021/acs.jpclett.7b03182.

[22] R. J. Maurer, Y. Zhang, H. Guo, and B. Jiang. “Hot Electron Effects during Reactive Scattering of H_2 from Ag(111): Assessing the Sensitivity to Initial Conditions, Coupling Magnitude, and Electronic Temperature”. In: *Faraday Discuss.* 214 (2019), pp. 105–121. DOI: 10.1039/C8FD00140E.

[23] Y. Zhang, R. J. Maurer, H. Guo, and B. Jiang. “Hot-Electron Effects during Reactive Scattering of H_2 from Ag(111): The Interplay between Mode-Specific Electronic Friction and the Potential Energy Landscape”. In: *Chem. Sci.* 10 (2019), pp. 1089–1097. DOI: 10.1039/C8SC03955K.

[24] P. Spiering, K. Shakouri, J. Behler, G.-J. Kroes, and J. Meyer. “Orbital-Dependent Electronic Friction Significantly Affects the Description of Reactive Scattering of N_2 from Ru(0001)”. In: *J. Phys. Chem. Lett.* 10 (2019), pp. 2957–2962. DOI: 10.1021/acs.jpclett.9b00523.

[25] M. Askerka, R. J. Maurer, V. S. Batista, and J. C. Tully. “Role of Tensorial Electronic Friction in Energy Transfer at Metal Surfaces”. In: *Phys. Rev. Lett.* 116 (2016), p. 217601. DOI: 10.1103/PhysRevLett.116.217601.

[26] D. Novko, M. Blanco-Rey, M. Alducin, and J. I. Juaristi. “Surface Electron Density Models for Accurate *Ab Initio* Molecular Dynamics with Electronic Friction”. In: *Phys. Rev. B* 93 (2016), p. 245435. DOI: 10.1103/PhysRevB.93.245435.

[27] J. Meyer. “Ab Initio Modeling of Energy Dissipation during Chemical Reactions at Transition Metal Surfaces”. PhD thesis. Freie Universität Berlin, Freie Universität Berlin, Germany, 2012.

[28] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, et al. “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems”. In: (2015). Software available from tensorflow.org.

[29] T. Tieleman and G. Hinton. “Lecture 6.5-Rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude”. In: *COURSERA Neural Netw. Mach. Learn.* 4.2 (2012), pp. 26–31.

[30] S. Manzhos and T. Carrington. “A Random-Sampling High Dimensional Model Representation Neural Network for Building Potential Energy Surfaces”. In: *J. Chem. Phys.* 125 (2006), p. 084109. DOI: 10.1063/1.2336223.

[31] S. Manzhos, R. Dawes, and T. Carrington. “Neural Network-Based Approaches for Building High Dimensional and Quantum Dynamics-Friendly Potential Energy Surfaces”. In: *Int. J. Quantum Chem.* 115 (2015), pp. 1012–1020. DOI: 10.1002/qua.24795.