Efficient nonparametric n-body force fields from machine learning

Aldo Glielmo, Claudio Zeni, and Alessandro De Vita

Efficient non-parametric n-body force fields from machine learning

Aldo Glielmo,[1, *] Claudio Zeni,[1, †] and Alessandro De Vita[1, 2]

[1]*Department of Physics, King's College London, Strand, London WC2R 2LS, United Kingdom*
[2]*Dipartimento di Ingegneria e Architettura, Università di Trieste, via A. Valerio 2, I-34127 Trieste, Italy*

We provide a definition and explicit expressions for n-body Gaussian Process (GP) kernels which can learn any interatomic interaction occurring in a physical system, up to n-body contributions, for any value of n. The series is complete, as it can be shown that the "universal approximator" squared exponential kernel can be written as a sum of n-body kernels. These recipes enable the choice of optimally efficient force models for each target system, as confirmed by extensive testing on various materials. We furthermore describe how the n-body kernels can be "mapped" on equivalent representations that provide database-size-independent predictions and are thus crucially more efficient. We explicitly carry out this mapping procedure for the first non-trivial (3-body) kernel of the series, and show that this reproduces the GP-predicted forces with meV/Å accuracy while being orders of magnitude faster. These results open the way to using novel force models ("M-FFs") that are computationally as fast as their corresponding standard parametrised n-body force fields, while retaining the non-parametric character, the ease of training and validation, and the accuracy of the best recently proposed machine learning potentials.

I. INTRODUCTION

Since their conception, first-principles molecular dynamics (MD) simulations [1] based on density functional theory (DFT) [2, 3] have proven extremely useful to investigate complex physical processes that require quantum accuracy. These simulations are computationally expensive, and thus still typically limited to hundreds of atoms and the picosecond timescale. For larger systems that are non-uniform and thus intractable using periodic boundary conditions, multiscale embedding ("QM/MM") approaches can sometimes be used successfully. This is possible if full quantum accuracy is only needed in a limited (QM) zone of the system, while a simpler molecular mechanics (MM) description suffices everywhere else. Very often, however, target problems require minimal model system sizes and simulation times so large that the calculations must be exclusively based on classical force fields i.e., force models that use the position of atoms as the only explicitly represented degrees of freedom.

In the remainder of this introductory section we briefly review the relative strengths and weaknesses of standard parametrized (P-FFs) and machine learning force fields (ML-FFs). We then consider how accurate P-FFs are hard to develop but eventually fully exploit useful knowledge on the systems, while GP-based ML-FFs offer a general mathematical framework for handling training and validation, but are significantly slower (Section I A). These shortcomings motivate an analysis of how prior knowledge such as symmetry has been so far incorporated in GP kernels (Section I B) and points to features still missing in ML kernels, which are commonplace in the more standard, highly efficient P-FFs based on truncated n-body expansions (Section I C). This suggests the possibility of defining a series of n-body GP kernels (Section II B), providing a scheme to construct them (Section II C and D) and, after the best value of n for the given target system has been identified with appropriate testing (Section II E), exploiting their dimensionally-reduced feature spaces to massively boost the execution speed of force prediction (Section III).

A. Parametrized and machine learning force fields

Producing accurate and fully transferable force fields is a remarkably difficult task. The traditional way to do this involves adjusting the parameters of carefully chosen analytic functions in the hope to match extended reference data sets obtained from experiments or quantum calculations [4, 5]. The descriptive restrictiveness of the parametric functions used is both a drawback and a strength of this methodology. The main difficulty is that developing a good parametric function requires a great deal of chemical intuition and patient effort, guided by trial and error steps with no guarantee of success [6]. However, for systems and processes where the approach is fruitful, the development effort is amply rewarded by the opportunity to provide extremely fast and accurate force models [7–10]. The identified functional forms will in these cases contain valuable knowledge on the target system, encoded in a compact formulation that still accurately captures the relevant physics. Such knowledge is furthermore often transferable to novel (while similar) systems as a "prior" piece of information, i.e., constitutes a good working hypothesis on how these systems will behave. When QM data on the novel system become available, this can be simply used to fine-tune the parameters of the functional form to a new set of best-fit values that maximise prediction accuracy.

Following a different approach, "non-parametric" ML force fields can be constructed, whose dependence on the atomic position is not constrained to a particular ana-

---

lytic form. An implementation and tests exploring the feasibility of ML to describe atomic interactions can be found e.g., in pioneer work by Skinner and Broughton [11] proposing to use ML models to reproduce first principles potential energy surfaces. More recent works implementing this general idea have been based on Neural Networks [12], Gaussian Process (GP) regression [13] or linear regression on properly defined bases [14]. Current work aims at making these learning algorithms both faster and more accurate [15–20].

As processing power and data communication bandwidth increase, and the cost of data storage decreases, modeling based on ML and direct inference promises to become an increasingly attractive option, compared with more traditional classical force field approaches. However, although ML schemes are general and have shown to be remarkably accurate interpolators in specific systems, so far they have not become as widespread as it might have been expected. This is mainly because "standard" classical potentials are still orders of magnitude faster than their ML counterpart [21]. Moreover, ML-FFs also involve a more complex mathematical and algorithmic machinery than the traditional compact functional forms of P-FFs, whose arguments are physically descriptive features that remain easier to visualize and interpret.

B. Prior knowledge and GP kernels

These shortcomings provide motivation for the present work. The high computational cost of many ML models is a direct consequence of the general inverse relation between the sought flexibility and the measured speed of any algorithm capable of learning. Highly flexible ML algorithms by definition assume very little or no prior knowledge of the target systems. In a Bayesian context, this involves using a general prior kernel, typically aspiring to preserve the full universal approximator properties of e.g., the square exponential kernel [22, 23]. The price of such a kernel choice is that the ML algorithm will need large training databases [24], slowing down computations as the prediction time grows linearly with the database size.

Large database sizes are not, however, unavoidable, and any data-intensive and fully flexible scheme to potential energy fitting is suboptimal by definition, as it exploits no prior knowledge of the system. This completely "agnostic" approach is at odds with the general lesson from classical potential development, indicating that it is essential for efficiency to incorporate in the force prediction model as much prior knowledge of the target system as can be made available. In this respect, GP kernels *can* be tailored to bring some form of prior knowledge to the algorithm.

It is for example possible to include any symmetry information of the system. This can be done by using descriptors that are independent of rotations, translations and permutations [15, 25–27]. Alternatively, one

can construct scalar-valued GP kernels that are made invariant under rotation (see e.g., [16, 28]) or matrix-valued GP kernels made covariant under rotation ([16], an idea that can be extended to higher-order tensors [29, 30]). Invariance or covariance are in these cases obtained starting from a non-invariant representation by appropriate integration over the $SO(3)$ rotation group [16, 28].

Symmetry aside, progress can be made by attempting to use kernels based on simple, descriptive features corresponding to low-dimensional feature spaces. Taking inspiration from parametrized force fields, these descriptors could e.g., be chosen to be interatomic distances taken singularly or in triplets, yielding kernels based on 2- or 3-body interactions [16, 31, 32]. Since low-dimensional feature spaces allow efficient learning (convergence is reached using small databases), to the extent that simple descriptors capture the correct physics, the GP process will be a relatively fast, while still very accurate, interpolator.

C. Scope of the present work

There are, however, two important aspects that have not as yet been fully explored while trying to develop efficient kernels based on dimensionally reduced feature spaces. Both aspects will be addressed in the present work.

First, a systematic classification of rotationally invariant (or covariant, if matrix valued) kernels, representative of the feature spaces corresponding to n-body interactions is to date still missing. Namely, no definition or general recipe has been proposed for constructing n-body kernels, or for identifying the actual value (or effective interval of values) of n associated with already available kernels. This would be clearly useful, however, as the discussion above strongly suggests that the kernel corresponding to the lowest value of n compatible with the physics of the target system will be the most informationally efficient one for carrying out predictions: striking the right balance between speed and accuracy.

Second, for any ML approach based on a GP kernel and a fixed database, the GP predictions for any target configuration are also fixed once and for all. For an n-body kernel, these predictions do not need, however, to be explicitly carried out as sums over the training dataset, as they could be approximated with arbitrary precision by "mapping" the GP prediction on a new representation based on the underlying n-body feature space. We note that this approximation step would make the final prediction algorithm independent of the database size, and thus in principle as fast as any classical n-body potential based on functional forms, while still parameter free. The remainder of this work explores these two issues, and it is structured as follows.

In the next Section II, after introducing the terminology and the notation (II A), we provide a definition of n-body kernel (II B) and propose a systematic way

of constructing n-body kernels of any order n, showing how previously proposed approaches can be reinterpreted within this scheme (II C and D). We furthermore show, by extensive testing on a range realistic materials, how the optimal interaction order can be chosen as the lowest n compatible with the required accuracy and the available computational power (II E). In the following Section III we describe how the predictions of n-body GP kernels can be recast (mapped) with arbitrary accuracy into very fast non-parametric force fields based on machine learning (M-FFs) which fully retain the n-body character of the GP process they were derived from. The procedure is carried out explicitly for a 3-body kernel, and we find that evaluating atomic forces is orders of magnitude faster than the corresponding GP calculation.

II. N-BODY EXPANSIONS WITH N-BODY KERNELS

A. Notation and terminology

GP-based potentials are usually constructed by assigning an energy ε to a given atomic configuration ρ, typically including a central atom and all its neighbors up to a suitable cutoff radius. The existence of a corresponding local energy function $\varepsilon(\rho)$ is generally assumed, in order to provide a total energy expression and guarantee a linear scaling of the predictions with the total number of atoms in the system. Within GP regression this function is calculated from a database $\mathcal{D} = \{\rho_d, \varepsilon_d, \mathbf{f}_d\}_{d=1}^N$ of reference data, typically obtained by quantum mechanical simulations, and usually consisting of a set of N atomic configurations $\{\rho_d\}$ together with their relative energies $\{\varepsilon_d\}$ and/or forces $\{\mathbf{f}_d\}$.

It is worth noting here that although there is no well defined local atomic energy in a reference quantum simulation, one can always use gradient information (atomic forces, which are well defined local physical quantities) to machine-learn a potential energy function. This can be done straightforwardly using derivative kernels (cf. e.g., Ref. [22], Section 9.4) to learn and predict forces. Alternatively, one can learn forces directly without an intermediate energy expression, as done in Refs. [15, 33] or more recently in Ref. [16]. A necessary condition for any of these approaches to produce energy-conserving force fields (i.e., fields that make zero work on any closed trajectory loop) is that the database is constructed once and for all, and never successively updated. After training on the given fixed database, the GP prediction on a target configuration ρ consists of a linear combination of the kernel function values measuring the similarity of the target configuration with each database entry:

$$\varepsilon(\rho) = \sum_{d=1}^N k(\rho, \rho_d)\alpha_d, \qquad (1)$$

where the coefficients α_d are obtained by means of in-

version of the *covariance matrix* [22] and can be shown to minimise the regularised quadratic error between GP predictions and reference calculations.

B. Definition of an n-body kernel

Classical interatomic potentials are often characterized by the number of atoms ("bodies") they let interact simultaneously (cf. e.g., Refs. [9, 10]). To translate this concept into the realm of GP regression, we assume that the target configuration $\rho(\{\mathbf{r}_i\})$ represents the local atomic environment of an atom fixed at the origin of a Cartesian reference frame, expressed in terms of the relative positions $\mathbf{r}_i$ of the surrounding atoms. We define the order of a kernel $k_n(\rho, \rho')$ as the smallest integer n for which the following property holds true:

$$\frac{\partial^n k_n(\rho, \rho')}{\partial \mathbf{r}_{i_1} \cdots \partial \mathbf{r}_{i_n}} = 0 \qquad \forall \mathbf{r}_{i_1} \neq \mathbf{r}_{i_2} \neq \cdots \neq \mathbf{r}_{i_n}, \qquad (2)$$

where $\mathbf{r}_{i_1}, \ldots, \mathbf{r}_{i_n}$ are the positions of any choice of a set of n different surrounding atoms. By virtue of linearity, the local energy in Eq. (1) will also satisfy the same property if k_n does. Thus, Eq. (2) implies that the central atom in a local configuration interacts with up to $n - 1$ other atoms simultaneously, making the interaction energy term n-body. For instance, using a 2-body kernel, the force on the central atom due to atom $\mathbf{r}_j$ will not depend on the position of any other atom $\mathbf{r}_{l \neq j}$ belonging to the target configuration $\rho(\{\mathbf{r}_i\})$. Eq. (2) can be used directly to check through either numeric or symbolic differentiation if a given kernel is of order n, a fact which might be far from obvious from its analytic form, depending on how the kernel is built.

C. Building n-body kernels I: SO(3) integration

Following a standard route [16, 28, 34], we begin by representing each local atomic configuration as a sum of Gaussian functions $\mathcal{N}$ with a given variance σ^2, centered on the M atoms of the configuration:

$$\rho(\mathbf{r}, \{\mathbf{r}_i\}) = \sum_{i=1}^M \mathcal{N}(\mathbf{r} \mid \mathbf{r}_i, \sigma^2), \qquad (3)$$

where $\mathbf{r}$ and $\{\mathbf{r}_i\}_{i=1}^M$ are position vectors relative to the central atom of the configuration. This representation guarantees by construction invariance with respect to translations and permutations of atoms (here assumed to be of a single chemical species). As described in [16], a covariant 2-body force kernel can be constructed from the non-invariant scalar ("base") kernel obtained as a dot

product overlap integral of the two configurations:

$$k_2(\rho, \rho') = \int d\mathbf{r} \, \rho(\mathbf{r})\rho'(\mathbf{r})$$

$$= L \sum_{i \in \rho, j \in \rho'} e^{-(\mathbf{r}_i - \mathbf{r}'_j)^2/4\sigma^2}, \qquad (4)$$

where L is an unessential constant factor, omitted for convenience from now on. That (4) is a 2-body kernel consistent with the definition of Eq. (2) can be checked straightforwardly by explicit differentiation (see Appendix A). Its 2-body structure is also readable from the fact that k_2 is a sum of contributions comparing pairs of atoms in the two configurations, the first pair located at the two ends of vector $\mathbf{r}_i$ in the target configuration ρ, and consisting of the central atom and atom i, and the second pair similarly represented by the vector $\mathbf{r}'_j$ in the database configuration ρ'. A rotation-covariant matrix-valued force kernel can at this point be constructed by Haar integration [35, 36] as an integral over the $SO(3)$ manifold [16]:

$$\mathbf{K}_2^s(\rho, \rho') = \int_{SO(3)} d\mathcal{R} \, \mathbf{R} \, k_2(\rho, \mathcal{R}\rho'). \qquad (5)$$

This kernel can be used to infer forces on atoms using a GP regression vector formula analogous to Eq. (1) (see Ref. [16]). These forces belong to a 2-body force field purely as a consequence of the base kernel property in Eq. (2). It is interesting to notice that there is no use or need for an intermediate energy expression to construct this 2-body force field, which is automatically energy-conserving.

Higher order n-body base kernels can be constructed as finite powers of the 2-body base kernel (4):

$$k_n(\rho, \rho') = k_2(\rho, \rho')^{n-1} \qquad (6)$$

where the n-body property (Eq. (2)) can once more be checked by explicit differentiation (see Appendix A). Furthermore, taking the exponential of the kernel in Eq. (4) gives rise to a fully many-body base kernel, as all powers of k_2 are contained in the exponential formal series expansion:

$$k_{MB}(\rho, \rho') = e^{k_2(\rho, \rho')/\theta^2}$$

$$= 1 + \frac{1}{\theta^2}k_2 + \frac{1}{2!\theta^4}k_3 + \dots . \qquad (7)$$

One can furthermore check that the simple exponential many-body kernel k_{MB} defined above is, up to normalisation, equivalent to the squared exponential kernel [22] on the natural distance induced by the dot product kernel $k_2(\rho, \rho')$:

$$e^{-d^2(\rho, \rho')/2\theta^2} = N(\rho)N(\rho')k_{MB}(\rho, \rho') \qquad (8)$$

$$d^2(\rho, \rho') = k_2(\rho, \rho) + k_2(\rho', \rho') - 2k_2(\rho, \rho'). \qquad (9)$$

To check on these ideas, we next test the accuracy of these kernels in learning the interactions occurring in a
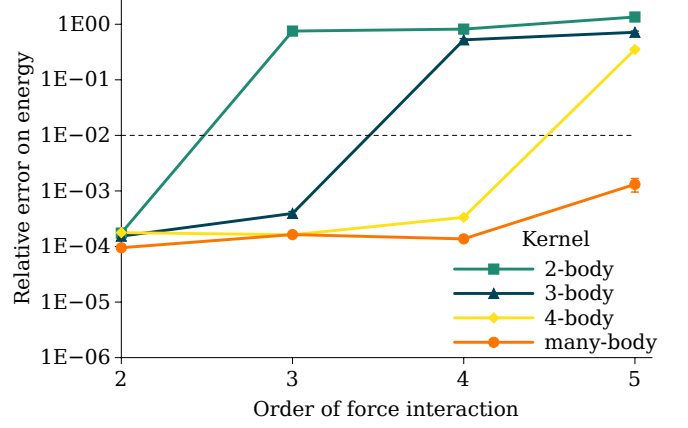

Figure 1. GP relative error as a function of the force interaction order (2- to 5-body), using n-body kernels with increasing n. Learning forces within baseline precision (black dashed line) requires an n-body kernel with n at least as high as the order of the force interaction.

simple 1D model consisting of n' particles interacting via an *ad-hoc* n'-body potential (see Appendix B.). We first let the particles interact to generate a configuration database, and then attempt to machine-learn these interactions using the kernels just described. Figure 1 illustrates the average prediction errors on the local energies of this system incurred by the GP regression based on four different kernels as a function of the interaction order n'. It is clear from the graph that a force field that lets interact the n' particles simultaneously can only be learned accurately with a $(n \geq n')$-body kernel (6), or with the many-body exponential kernel (7) which contains all interaction orders.

To construct n-body kernels useful for applications to real 3D systems we need to include rotational symmetry by averaging over the rotation group. For our present scopes, it is sufficient to discuss the case of rotation-invariant n-body scalar energy kernels, for which the integral (formally a *transformation integration* [37]) is readily obtained from Eq. (5) by simply dropping the $\mathbf{R}$ matrix in the integrand:

$$k_n^s(\rho, \rho') = \int_{SO(3)} d\mathcal{R} \, k_n(\rho, \mathcal{R}\rho'). \qquad (10)$$

The use of this integral in the context of potential energy learning was originally proposed in [28], where it was carried out using appropriate functional expansions. Alternatively, one can exploit the Gaussian nature of the configuration expansion (3) to obtain an analytically exact formula, as done further below. The resulting symmetrized n-body kernel k_n^s will learn faster than its non-symmetrized counterpart k_n, as the rotational degrees of freedom have been integrated out. This is because a non-symmetrized n-body kernel (k_n) must learn functions of $3n - 3$ variables (translations are taken into account by the local representation based on relative position in Eq. (3)). After integration, the new kernel k_n^s

defines a smaller and more physically-based space of functions of $3n - 6$ variables, which is the rotation-invariant functional domain of n interacting particles.

The symmetrization integral in Eq. (10) can be written down for the many-body base kernel k_{MB} (Eq. (7)), to define a new many-body kernel k_{MB}^s invariant under all physical symmetries:

$$k_{MB}^s(\rho, \rho') = \int_{SO(3)} d\mathcal{R}\, k_{MB}(\rho, \mathcal{R}\rho'). \qquad (11)$$

By virtue of the universal approximation theorem [22, 38] this kernel would be able to learn arbitrary physical interactions with arbitrary accuracy, if provided with sufficient data.

Unfortunately, the exponential kernel (7) has to date resisted all attempts to carry out the analytic integration over rotations (11), leaving as the only open options numerical integration, or discrete summation over a relevant point group of the system [16]. On the other hand, the analytic integration of 2- or 3-body kernels to give symmetric n-body kernels can be carried out in different ways.

For example, one could use an intermediate step that was introduced during the construction of the widely used SOAP kernel [28, 31, 39–41]. This kernel has a full many-body character [28], ensured by the prescribed normalisation step defined by Eqs. (31-36) of the standard Ref. [28]), which made it possible to use it e.g., to augment to full many-body the descriptive power of a 2- and 3-body explicit kernel expansion [42]. However, the Haar integral over rotations introduced in [28] as an intermediate kernel construction step could also be seen, if taken on its own, as a transformation integration procedure [37] yielding a symmetrized n-body kernel as defined in Eq. (10) above, which would in turn become a higher finite-order kernel if raised to integer powers $\zeta \geq 2$ (see next subsection).

Carrying out Haar integrals is not, in general, an easy task. In the example above, computing a general rotation invariant n-body kernel via the exact, suitably truncated spherical harmonics expansion procedure of Ref. [28] becomes challenging for $n > 3$. Significant difficulties likewise arise if attempting a "covariant" integration over rotations, for which we found an exact analytic expression only for 2- and 3-body matrix-valued kernels [16], with a technique that becomes unviable for $n > 3$. Fortunately, the Haar integration can be avoided altogether, following the simple route of constructing symmetric n-kernels directly using symmetry-invariant descriptors, as we will see in the next section. The problem of obtaining an analytic Haar integral expression for the general n case remains, however, an interesting one, which we tackle in the reminder of this section following a novel analytic route.

We first write the n-body base kernel of Eq. (6) as an explicit product of $(n - 1)$ 2-body kernels. The Haar
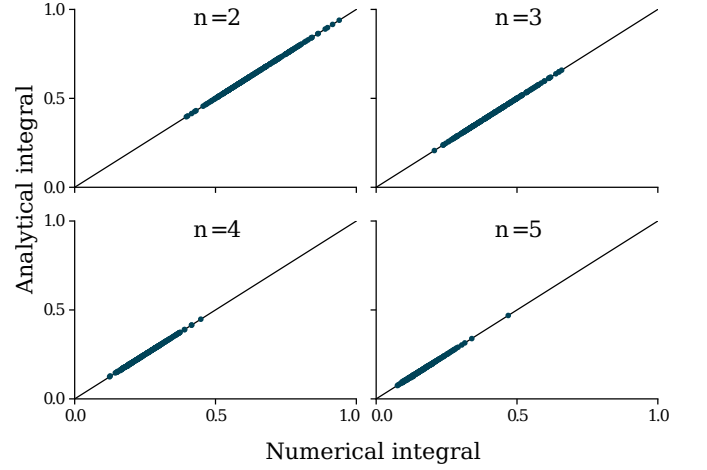

Figure 2. Scatter plots showing the values of the integral in (13) (on a random selection of configurations) computed either by numerical integration or via the analytic expression (Eqs. (14, 15, 17)). Interaction orders from $n = 2$ to $n = 5$ are considered.

integral (10) can then be written as

$$k_n^s(\rho, \rho') = \sum_{\substack{\mathbf{i}=(i_1,\ldots,i_{n-1})\in\rho \\ \mathbf{j}=(j_1,\ldots,j_{n-1})\in\rho'}} \tilde{k}_{\mathbf{i},\mathbf{j}} \qquad (12)$$

$$\tilde{k}_{\mathbf{i},\mathbf{j}} = \int d\mathcal{R}\, e^{-\frac{\|\mathbf{r}_{i_1} - \mathbf{R}\mathbf{r}'_{j_1}\|^2}{4\sigma^2}} \ldots e^{-\frac{\|\mathbf{r}_{i_{n-1}} - \mathbf{R}\mathbf{r}'_{j_{n-1}}\|^2}{4\sigma^2}} \qquad (13)$$

where now for each of the two configurations ρ, ρ', the sum runs over all n-plets of atoms that include the central atom (whose indices i_0 and j_0 are thus omitted). Expanding the exponents as $(\mathbf{r}_i - \mathbf{R}\mathbf{r}'_j)^2 = r_i^2 + r_j'^2 - 2\mathrm{Tr}(\mathbf{R}\mathbf{r}'_j\mathbf{r}_i^T)$ allows extracting from the integral (13) a rotation independent constant $\mathcal{C}_{\mathbf{i},\mathbf{j}}$, and to express the rotation-dependent scalar products sum as a trace of a matrix product:

$$\tilde{k}_{\mathbf{i},\mathbf{j}} = \mathcal{C}_{\mathbf{i},\mathbf{j}}\mathcal{I}_{\mathbf{i},\mathbf{j}} \qquad (14)$$

$$\mathcal{C}_{\mathbf{i},\mathbf{j}} = e^{-(r_{i_1}^2 + r_{j_1}'^2 + \ldots + r_{i_{n-1}}^2 + r_{j_{n-1}}'^2)/4\sigma^2} \qquad (15)$$

$$\mathcal{I}_{\mathbf{i},\mathbf{j}} = \int d\mathcal{R}\, e^{\mathrm{Tr}(\mathbf{R}\mathbf{M}_{\mathbf{i},\mathbf{j}})} \qquad (16)$$

where the matrix $\mathbf{M}_{\mathbf{i},\mathbf{j}}$ is the sum of the outer products of the ordered vector couples in the two configurations: $\mathbf{M}_{\mathbf{i},\mathbf{j}} = (\mathbf{r}'_{j_1}\mathbf{r}_{i_1}^T + \cdots + \mathbf{r}'_{j_{n-1}}\mathbf{r}_{i_{n-1}}^T)/2\sigma^2$. The integral (16) occurs in the context of multivariate statistics as the generating function of the non-central Wishart distribution [43]. As shown in [44], it can be expressed as a power series in the symmetric polynomials ($\alpha_1 = \sum_i^3 \mu_i, \alpha_2 = \sum_{i<j}^3 \mu_i\mu_j, \alpha_3 = \mu_1\mu_2\mu_3$) of the eigenvalues $\{\mu_i\}_{i=1}^3$ of

the symmetric matrix $\mathbf{M}_{\mathbf{i},\mathbf{j}}^{\mathrm{T}}\mathbf{M}_{\mathbf{i},\mathbf{j}}$:

$$\mathcal{I}_{\mathbf{i},\mathbf{j}} = \sum_{p_1,p_2,p_3} A_{p_1 p_2 p_3} \alpha_1^{p_1} \alpha_2^{p_2} \alpha_3^{p_3} \tag{17}$$

$$A_{p_1 p_2 p_3} = \frac{\pi\, 2^{-(1+2p_1+4p_2+6p_3)}(p_1+2p_2+4p_3)!}{p_1! p_2! p_3! \Gamma(\frac{3}{2}+p_1+2p_2+3p_3)\Gamma(1+p_2+2p_3)}$$
$$\times \frac{1}{\Gamma(\frac{1}{2}+p_3)(p_1+2p_2+3p_3)!}. \tag{18}$$

Remarkably, in this result (whose exactness is checked numerically in Figure 2) the integral over rotations does not depend on the order n of the base kernel, once the matrix $\mathbf{M}_{\mathbf{i},\mathbf{j}}$ is computed. This is not the case for previous approaches to integrating over rotations [16, 28] that need to be reformulated with increasing and eventually prohibitive difficulty each time the order n needs to be increased.

However, the final expression given by Eqs. (14-18) is still a relatively complex and computationally heavy function of the atomic positions. To alleviate its evaluation cost, it would be interesting to see whether it is possible to recast it as an explicit scalar product in a given feature space. This would allow e.g., to transfer most of the computational burden to the pre-computation of the corresponding basis functions. Fortunately such complexity can be largely avoided altogether if equally accurate kernels can be built by physical intuition at least for the most relevant lowest n orders, as discussed in the next section.

D. Building n-body kernels II: n-body feature spaces and uniqueness issues

The practical effect of the Haar integration (10) is the elimination of the three spurious rotational degrees of freedom. The same result can always be achieved by selecting a group of symmetry- invariant degrees of freedom for the system, typically including the distances and/or bond angles found in local atomic environments, or simple functions of these. Appropriate symmetrized kernels can then simply be obtained by defining a similarity measure *directly* on these invariant quantities [15, 27, 42, 45]. To construct symmetry invariant n-body kernels with $n = 2$ and $n = 3$ we can choose these degrees of freedom to be just interparticle distances:

$$k_2^s(\rho,\rho') = \sum_{\substack{i\in\rho \\ j\in\rho'}} \tilde{k}_2(r_i, r_j) \tag{19}$$

$$k_3^s(\rho,\rho') = \sum_{\substack{i_1,i_2\in\rho \\ j_1,j_2\in\rho'}} \tilde{k}_3((r_{i_1}, r_{i_2}, r_{i_1 i_2}), (r'_{j_1}, r'_{j_2}, r'_{j_1 j_2})) \tag{20}$$

where the $\tilde{k}$ are kernel functions that directly specify the correlation of distances, or triplets of distances, found
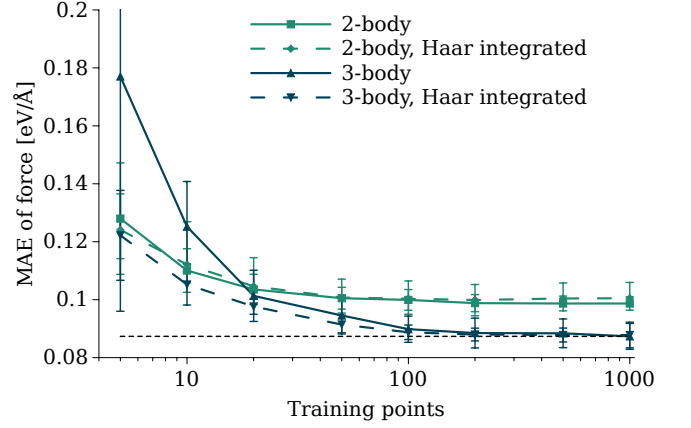

Figure 3. Learning curves for 2 and 3-body kernels obtained either via a Haar integration (Eqs. (12-18)), or directly specifying a similarity kernel function of the effective degrees of freedom (Eqs. (19, 20)).

within the two configurations. Since these kernels learn functions of low-dimensional spaces, their exact analytic form is not essential for performance, as any fully non-linear function $\tilde{k}$ will give equivalent converged results in the rapidly reached large-database limit. This equivalence can be neatly observed in Figure 3, which reports the performance of 2- and 3-body kernels built either directly over the set of distances (Eqs. (19) and (20)) or via the exact Haar integral (Eqs. (12-18)). As the test system is crystalline Silicon, 3-body kernels are better performing. However, since convergence of the 2- and 3-body feature space is quickly achieved (at about $N = 50$ and $N = 100$ respectively), there is no significant performance difference between $SO(3)$-integrated n-body kernels and physically motivated ones. Consequently, for low interaction orders, simple and computationally fast kernels like the ones in Eqs. (19, 20) are always preferable to more complex (and heavier) alternatives obtained via integration over rotations (e.g., the one defined by Eqs. (12-18) or those found in Refs. [16, 28].

We note at this point that Eq. (19) can be generalized to construct a symmetric n-body kernel

$$k_n^s(\rho,\rho') = \sum_{\substack{i_1,...,i_{n-1}\in\rho \\ j_1,...,j_{n-1}\in\rho'}} \tilde{k}_n(\mathbf{q}_{i_1,...,i_{n-1}}, \mathbf{q}'_{j_1,...,j_{n-1}}), \tag{21}$$

where the components of the feature vectors $\mathbf{q}$ are the chosen symmetry-invariant degrees of freedom describing the n-plet of atoms.

The $\mathbf{q}$ feature vectors are required to be $(3n-6)$ dimensional for all n, except for $n = 2$, where they become scalars. In practice, for $n > 3$ selecting a suitable set of invariant degrees of freedom is not trivial. For instance, for $n = 4$ the set of six unordered distances between four particles do not specify their relative positions unambiguously, while for $n > 4$ the number of distances associated with n atoms exceeds the target feature space dimension
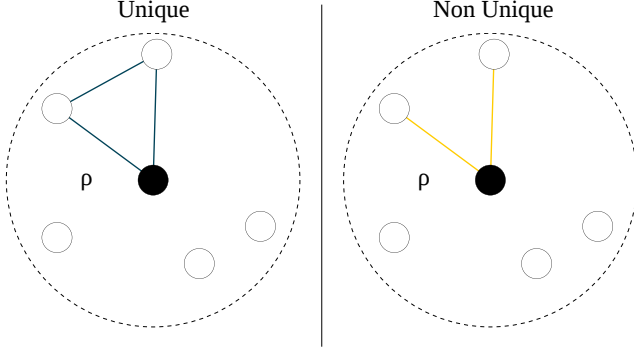
Figure 4. Unique interaction (left panel) associated with the 3-body kernel k_3^s (20) compared with the non-unique 3-body interaction (right panel) associated with the kernel $k_3^{\neg u} = (k_2^s)^2$ (22), which is a function of two distances only (see text).

kernel	order	symm.	unique	name
k_2^s	2	✓	✓	2-body
$k_3^{\neg u}$	3	✓	✗	3-body, non-unique
k_3^s	3	✓	✓	3-body
$k_5^{\neg u}$	5	✓	✗	5-body, non-unique
k_{MB}^{ds}	∞	$\sim$	✓	many-body, approx. symmetric

Table I. Some of the kernels presented and their properties.

$3n - 6$. Meanwhile, the computational cost of evaluating the full sum in Eq. (21) very quickly becomes prohibitively large as the number of elements in the sum grows exponentially with n.

The order of an *already symmetric* n-body kernel can however be augmented with no computational overhead by generating a derived kernel through simple exponentiation to an integer power, at the cost of losing the *uniqueness* [28, 46, 47] of the representation. This can be easily understood by means of an example (graphically illustrated in Figure 4). Let us consider the 2-body symmetric kernel k_2^s (Eq. (19)) which learns a function of just a single distance, and therefore treats the r_i distances between the central atom and its neighbors independently. Its square is the kernel

$$k_3^{\neg u}(\rho, \rho') = \sum_{\substack{i_1, i_2 \in \rho \\ j_1, j_2 \in \rho'}} \tilde{k}_2(r_{i_1}, r'_{j_1}) \tilde{k}_2(r_{i_2}, r'_{j_2}) \qquad (22)$$

which will be able to learn functions of two distances r_{i_1}, r_{i_2} from the central atom of the target configuration ρ (see Figure 4) and thus will be a 3-body kernel in the sense of Eq. (2). However, this kernel cannot resolve angular information, as rotating the atoms in ρ around the origin by independent, arbitrary angles will yield identical predictions.

Extending this line of reasoning, it is easy to show that squaring a symmetric 3-body kernel yields a kernel that can capture interactions up to 5-body, although again non-uniquely. This has often been done in practice by squaring the SOAP integral [41, 42]. In general, raising a 3-body "input" kernel to an arbitrary integer power $\zeta \geq 2$ yields an n-body output kernel of order $2\zeta + 1$, $k_{n=2\zeta+1}^{\neg u} = k_3^s(\rho, \rho')^\zeta$. This kernel is also non-unique as it will learn a function of only 3ζ variables, while the total number of relevant n-body degrees of freedom $(3n - 6 = 6\zeta - 3)$ is always larger than this. Substituting 3 with any n' order of the symmetrized input kernel will similarly generate a $k_n^{\neg u} = k_{n'}^s(\rho, \rho')^\zeta$ kernel of order $n = (n' - 1)\zeta + 1$. A

simple calculation reveals that, also in the general case, the number of variables on which $k_n^{\neg u}$ is implicitly built is $(3n' - 6)\zeta$, always smaller than the full dimension of n-body feature space $(3n' - 3)\zeta - 3$ (as expected, the two become equal only for the trivial exponent $\zeta = 1$).

None of the kernels obtained as finite powers of some symmetric lower-order kernels is a many-body one (they will all satisfy Eq. (2) for some finite n). However, an attractive immediate generalization consists of substituting any squaring or cubing with full exponentiation. For instance, exponentiating a symmetrized 3-body kernel we obtain the many-body kernel $k_{MB} = \exp[k_3^s(\rho, \rho')]$. It is clear from the infinite expansion in Eq. (7) that this kernel is a many-body one in the sense of Eq. (2), and is also fully symmetric [1]. As is also the case for all finite-power kernels, the computational cost of this many body kernel will depend on the order n' of the input kernel (3 in the present example) as the sum in Eq. (21) only runs on the atomic n'-plets (here, triplets) in ρ and ρ'. This new kernel is not a priori known to neglect any order of interaction that might occur in a physical system and thus be encoded in a reference QM training database.

To summarise, we provided a definition for an n-body kernel, and proposed a general formalism for building n-body kernels by exact Haar integration over rotations. We then defined a class of simpler kernels based on rotation invariant features that are also n-body according to the previous definition. As both approaches become computationally expensive for high values of n, we pointed out that n-body kernels can be built as powers of lower-order input n'-body kernels, with no additional computational overhead. While such a procedure in our case comes at the cost of sacrificing the unicity property of the descriptor, it also suggests how to build, by full exponentiation, a many-body symmetric kernel. For many applications, however, using a finite-order kernel will provide the best option.

---

[1] One could also inexpensively obtain a many body kernel by normalisation of an explicit finite order one, for instance, as $k_{MB} = k_3^s(\rho, \rho')/\sqrt{k_3^s(\rho, \rho)k_3^s(\rho', \rho')}$. The denominator makes this many-body in the sense of Eq. (2) (as is also the case for the SOAP kernel, see discussion in Section II C, while no Haar integration is needed here).
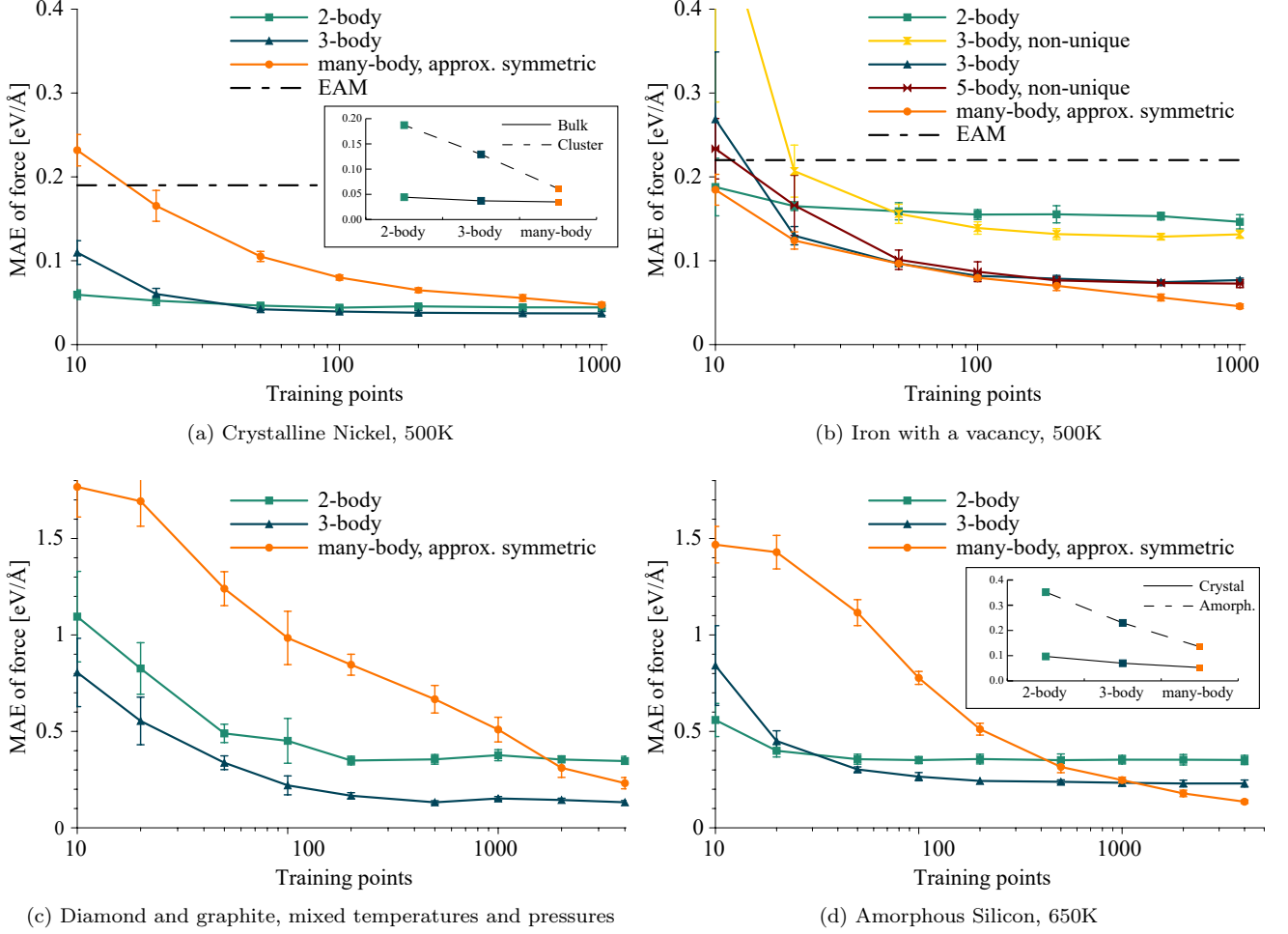
Figure 5. Learning curves reporting the mean generalization error (measured as the modulus of the difference between target and predicted force vectors) as a function of the training set size, for different materials and kernels of increasing order. For extra detail on the data sets and kernels used, and on the experimental methodology, see Appendices C and D.

E. Optimal n-kernel choice

In general, choosing a higher order n-body kernel will improve accuracy at the expense of speed. The optimal kernel choice for a given application will correspond to the best tradeoff between computational cost and representation power, which will depend on the physical system investigated. The properties of some of the kernels discussed above are summarized in Table I, while their performance is tested on a range of materials in Figure 5.

The figure reveals some general trends. 2-body kernels can be trained very quickly, as good convergence can be attained already with ∼100 training configurations. The 2-body representation is a very good descriptor for a few materials under specific conditions, while their overall accuracy is ultimately limited. This will yield e.g., excellent force accuracy for a close-packed bulk system like crystalline Nickel (inset (a)), and reasonable accuracy for a defected α-Fe system whose bcc structure is however metastable if just pair potentials are used (inset (b)). Ac-

curacy improves dramatically once angular information is acquired by training 3-body kernels. These can accurately describe forces acting on iron atoms in the bulk α-Fe system containing a vacancy (inset (b)) and those acting on carbon atoms in both diamond and graphite (inset (c)). However, 3-body GPs need larger training databases. Also, atoms participate in many more triplets than simple bonds in their standard environments contained in the database, which will make 3-body kernels slower than 2-body ones for making predictions by GP regression. Both problems would extend, getting worse, to higher values of n, as summing over all database configurations and all feature n-plets in each database configuration will make GP predictions progressively slower. However, complex materials where high-order interactions presumably play a significant role should be expected to be well described by ML-FF based on a many-body kernel. This is verified here in the case of amorphous Silicon (inset (d)).

Identifying the n value best suited for the description

of a given material system can also be done in practice by monitoring how the converged error varies as a function of the kernel order. Plots illustrating this behaviour are provided in insets (a) and (d) for nickel and silicon systems, respectively. In each plot the more complex system (a Ni cluster and an amorphous Si system, respectively) display a high accuracy gain (larger negative slope) when the kernel order is increased, while the relatively simpler cristalline Ni and Si systems show a practically constant trend on the same scale.

Figure 5 (b) also shows the performance of some non-unique kernels. As discussed above, these are options to increase the order of an input kernel avoiding the need to sum over the correspondingly higher order n-plets. Our tests indicate that the ML-FFs generated by non-unique kernels sometimes improve appreciably on the input kernels' performance: e.g., the error incurred by the 2-body kernel of Eq. (19) in the Fe-vacancy system is higher than that associated with its square, the non-unique 3-body kernel of Eq. (22). Unfortunately, but not surprisingly, the improvement can be in other cases modest or nearly absent, as exemplified by comparing the errors associated with the 3-body kernel and its square -the non-unique 5-body kernel-, in the same system.

Overall, the analysis of Figure 5 suggests that an optimal kernel can be chosen by comparing the learning curves of the various n-body kernels and the many-body kernel over the available QM database: the comparison will reveal the simplest (most informative, lowest n) description that is still compatible with the error level deemed acceptable in the simulation.

Trading transferability for accuracy by training the kernels on a QM database appropriately tailored for the target system (e.g., restricted to just bulk or simply-defected system configurations sampled at the relevant temperatures as done in the Ni and Fe-systems of Figure 5) will enable surprisingly good accuracy even for low n values. This should be expected to systematically improve on the accuracy performance of classical potentials involving non-linear parameter fitting, as exemplified by comparing the errors associated with n-body kernel models and the average errors of state-of-the-art embedded atom model (EAM) P-FFs [7, 48] (insets (a) and (b)). The next section further explores the performance of GP-based force prediction, to address the final issue of what execution speed can be expected for ML-based force fields, once the optimally accurate choice of kernel has been made.

III. MAPPED FORCE FIELDS (M-FFS)

Once a GP kernel is recognized as being n-body, it automatically defines an n-body force field corresponding to it, for any given choice of training set. This will be an n-body function of atomic positions satisfying Eq. (2), whose values *can* be computed by GP regression sums over the training set as done by standard ML-FF imple-
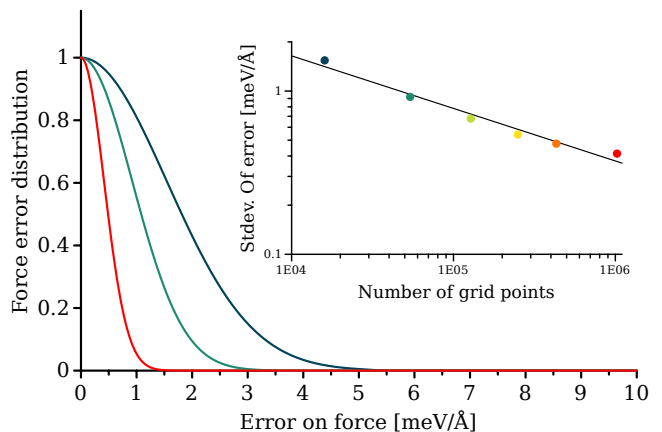

Figure 6. Error distributions of the 3D spline interpolation upon approximating the atomic forces predicted by the 3-body GP kernel of Eq. (20), for a 650K a-Si sytem. The distribution is narrower for denser grids. Inset: standard deviation of the distributions as a function of the number of interpolation grid points, on a log-log scale (each distribution in the main panel corresponds to a dot of same color in the insert).

mentations, but *do not have to* be computed this way. In particular, the execution speed of a machine learning-derived n-body force field might be expected to depend on its order n (e.g., it will involve sums over all atomic triplets, like any 3-body P-FF, if n=3), but should otherwise be independent of the training set size. It should therefore be possible to construct a mapping procedure yielding a machine learning-derived, non-parametric force field (an efficient "M-FF") that allows a very significant speed-up over calculating forces by direct GP regression. We note that non-unique kernels obtained as powers of n'-body input kernels exceed their reference n'-body feature space and thus could not be similarly sped up by mapping their predictions onto an M-FF of equal order n', while mapping onto an M-FF of the higher output order n would still be feasible.

For convenience, we will analyze a 3-body kernel case, show that a 3-body GP exactly corresponds to a classical 3-body M-FF, and show how the mapping yielding the M-FF can be carried out in this case, using a 3D-spline approximator. The generalization to any order n is straightforward provided that a good approximator can be identified and implemented. We begin by inserting the general form of a 3-body kernel (Eq. (21)) into the GP prediction expression (Eq. (1)), to obtain

$$\varepsilon(\rho) = \sum_{d=1}^{N} \sum_{\substack{i_1, i_2 \in \rho \\ j_1, j_2 \in \rho_d}} \tilde{k}_3(\mathbf{q}_{i_1, i_2}, \mathbf{q}_{j_1, j_2}^d)\alpha_d. \qquad (23)$$

Inverting the order of the sums over the database and atoms in the target configurations yields a general ex-
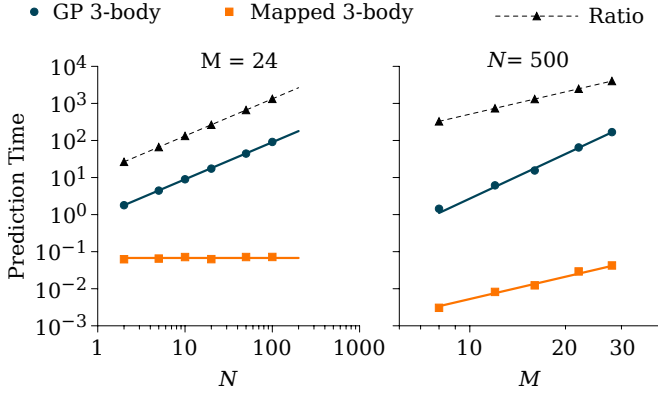
Figure 7. Computational cost of evaluating the 3-body energy (Eq. (24)) as a function of the database size N and the number of atoms M located within the cutoff radius. Left: time taken (s) for a single energy prediction using the GP (blue dots and solid line) and the mapped potential (orange dots and solid line), as a function of N, for $M=24$. The speed-up ratio is also provided as a dotted line. Right: scaling of the same quantities as a function of M for a training set of $N = 500$ configurations.

pression for the 3-body potential:

$$\varepsilon(\rho) = \sum_{i_1, i_2 \in \rho} \tilde{\varepsilon}(\mathbf{q}_{i_1, i_2}) \tag{24}$$

$$\tilde{\varepsilon}(\mathbf{q}_{i_1, i_2}) = \sum_{d=1}^{N} \sum_{j_1, j_2 \in \rho_d} \tilde{k}_3(\mathbf{q}_{i_1, i_2}, \mathbf{q}_{j_1, j_2}^d) \alpha_d. \tag{25}$$

Eq. (24) reveals that the GP implicitly defines the local energy of a configuration as a sum over all triplets containing the central atom, where the function $\tilde{\varepsilon}$ represents the energy associated with each triplet in the physical system. The triplet energy is calculated by three nested sums, one over the N database entries and two running over the M atoms of each database configuration (M may slightly vary over configurations, but can be assumed to be constant for the present purpose). The computational cost of a single evaluation of the triplet energy (25) scales consequently as $\mathcal{O}(NM^2)$. Clearly, improving the GP prediction accuracy by increasing N and M will make the prediction slower. However, such computational burden can be avoided, bringing the complexity of the triplet energy calculation (25) to $\mathcal{O}(1)$.

Since the triplet energy $\tilde{\varepsilon}$ is a function of just three variables (the effective symmetry-invariant degrees of freedom associated with three particles in three dimensions), we can calculate and store its values on an appropriately distributed grid of points within its domain. This procedure effectively maps the GP predictions on the relevant 3-body feature space: once completed, the value of the triplet energy at any new target point can be calculated via a local interpolation, using just a subset of nearest tabulated grid points. If the number of grid points N_g is made sufficiently high, the mapped function will be essentially identical to the original one but, by virtue of
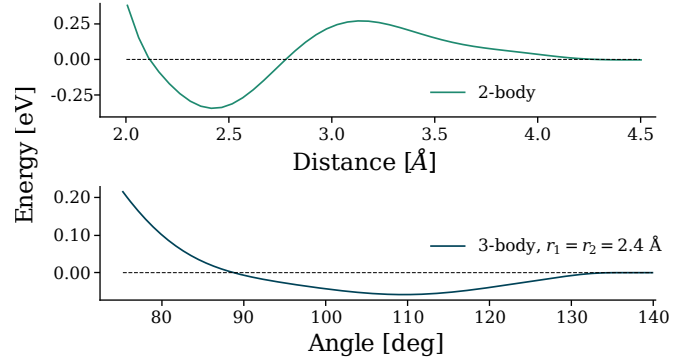

Figure 8. Energy profiles of the 3-body M-FF, trained for the a-Si system at 650K. Upper panel: 2-body interaction term. Lower panel: 3-body interaction energy for an atomic triplet, angular dependence when the two distances from the central atoms are both equal to 2.4Å.

the locality of the interpolation, the cost of evaluating it will not depend on N_g.

The 3-body configuration energy of Eq. (24) also includes 2-body contributions coming from the terms in the sum for which the indices i_1 and i_2 are equal. When $i_1 = i_2 = i$ the term $\varepsilon(\mathbf{q}_{i,i})$ can be interpreted as the pairwise energy associated with the central atom and atom i. The term can consequently be mapped onto a 1D 2-body feature space whose coordinate is the single independent component of the $\mathbf{q}_{i,i}$ feature vector, typically the distance between atom i and the central atom. In the same way, an n-body kernel naturally defines a set of n-body energy terms of order comprised between 2 and n, depending on the number of repeated indices.

Figure 6 shows the convergence of the mapped forces derived from the 3-body kernel in Eq. (20) for a database of DFTB atomic forces for the a-Si system. The interpolation is carried out using a 3D cubic spline for different 3D mesh sizes. Comparison with the reference forces produced by the GP allows to generate, for each mesh size, the distribution of the absolute deviation of the force components from their GP-predicted values. The standard deviation of the interpolation error distribution is shown in the insert on a log-log scale, as a function of N_g. Depending on the specific reference implementation, the speed-up in calculating the local energy (Eq. (24)) provided by the mapping procedure can vary widely, while it will always grow linearly with N and quadratically with M (see Figure 7), and it will be always substantial: in typical testing scenarios we found this to be of the order of $10^3 - 10^4$.

An M-FF example, obtained for a-Si with $n = 3$ is shown in Figure 8. As the energy profile is not prescribed by any particular functional form, it is free to optimally adapt to the information contained in the QM training set, to best reproduce the quantum interactions that produced it. Figure 8 contains some expected features e.g., a radial minimum at about $r \simeq 2.4$Å in the 2-body section (upper panel), the corresponding angular minimum at

$\theta_0 \simeq 110°$ (lower panel), which is approximately equal to the sp^3 hybridization angle of $109.47°$, and rapid growth for small radii (upper panel) and angles (lower panel). Less intuitive features are also visible, which however contribute to the best representation of the bulk system's interactions that a 3-body expansion can achieve for the given database. An example is the shallow maximum in the 2-body section at $r \simeq 3.1$Å, which would of course disappear if we fitted our model on QM forces calculated for a Si dimer, that do not contain a hump. The resulting Si force field, appropriate for a Si dimer, would however inevitably reproduce the QM bulk interactions less accurately. More generally, training on the aggregate dataset could be a sensible compromise, producing a more transferable, but locally less accurate force field.

IV. CONCLUDING REMARKS

The results presented in this work exemplify how physical priors built in the GP kernels restrict their descriptiveness, while improving their convergence speed as a function of training dataset size. This provides a framework to optimise efficiency. Comparing the performance of n-body kernels allows us to identify the lowest order n that is compatible with the required accuracy as a consequence of the physical properties of the target system. As a result, accuracy can in each case be achieved using the most efficient kernel e.g., a 2-body kernel for bulk Ni, or a 3-body for carbon and graphite, for a ~ 0.1 eV/Å target force accuracy, see Figure 5. As should be reasonably expected, relying on low-dimensional feature spaces will limit the maximum achievable accuracy if higher-order interactions, missing in the representation, occur in the target system.

On the other hand, we find that once the optimal order n has been identified and the n-kernel has been trained (whichever its form e.g., whether defined as a function of invariant descriptors as in (21), or constructed as a power of such a function, or derived as an analytic integral over rotations using Eqs. (12-18)), it becomes possible to map its prediction on the appropriate n-dimensional domain, and thus generate a n-body M-FF machine-learned atomic force field that predicts atomic forces at essentially the same computational cost of a classical n-body P-FF parametric force field. The GP predictions allow for a natural intrinsic measure of uncertainty -the GP predicted variance-, and the same mapping procedure used for the former can also be applied to the latter. Thus, like their ML-FF counterparts, and unlike P-FFs, M-FFs offer a tool which could be used to monitor whether any extrapolation is taking place that might involve large prediction errors.

In general, our results suggest a possible three-step procedure to build fast non-parametric M-FFs whenever a series of kernels k_n can be defined with progressively higher complexity/descriptive power and well-defined feature spaces with n-dependent dimensionality. However the series is constructed (and whether or not it converges to a universal descriptor) this will involve (i) GP training of n-kernels for different values of n, in each case using as many database entries relevant to the target system as needed to achieve convergence of the n-dependent prediction error; (ii) identification of the optimal order n, yielding the simplest viable description of the system's interactions - this could be e.g., the minimal value of n compatible with the target accuracy or, within a GP statistical learning framework, the result of a Bayesian model selection procedure [22]; (iii) mapping of the resulting (GP-predicted) ML-FF onto an efficient M-FF, using a suitably fast approximator function defined over the relevant feature space.

A major limitation of the M-FFs obtained this way is that, similar to P-FFs, each of them can be used only in "interpolation mode", that is when the target configurations are all well represented in the fixed database used. This is not the case in molecular dynamics simulations potentially revealing new chemical reaction paths, or whenever online learning or the use of dynamically-adjusted database subsets are necessary to avoid unvalidated extrapolations and maximise efficiency. In such cases, "learn on the fly" (LOTF) algorithms can be deployed, which have the ability to incorporate novel QM data into the database used for force prediction. In such schemes, the new data are either developed at runtime by new QM calculations, or are adaptably retrieved as the most relevant subset of a much larger available QM database [15]. The availability of an array of n-body kernels is very useful for this class of algorithms, which provides further motivation for their development. In particular, distributing the use of n-body kernels non-uniformly in both space and time along the system's trajectory has the potential to provide an optimally efficient approach to accurate MD simulations using the LOTF scheme. Finally, while the complication of continuously mapping the GP predictions to reflect a dynamically updated training database makes on the fly M-FF generation a less attractive option, a strategy to produce the MD trajectory with classical force field efficiency might involve using (concurrently across the system, and at any given time) a locally optimal choice built from a comprehensive set of pre-computed low-order M-FFs.

ACKNOWLEDGEMENTS

[1] R. Car and M. Parrinello, Physical Review Letters **55**, 2471 (1985).
[2] P. Hohenberg and W. Kohn, Physical review (1964).
[3] W. Kohn and L. J. Sham, Physical review **140**, A1133 (1965).
[4] F. H. Stillinger and T. A. Weber, Physical review **B31**, 5262 (1985).
[5] J. Tersoff, Physical Review B **37**, 6991 (1988).
[6] D. W. Brenner, physica status solidi(b) (2000).
[7] Y. Mishin, Acta Materialia **52**, 1451 (2004).
[8] A. C. T. van Duin, S. Dasgupta, F. Lorant, and W. A. Goddard, The Journal of Physical Chemistry A **105**, 9396 (2001).
[9] G. A. Cisneros, K. T. Wikfeldt, L. Ojamäe, J. Lu, Y. Xu, H. Torabifard, A. P. Bartók, G. Csányi, V. Molinero, and F. Paesani, Chemical Reviews **116**, 7501 (2016).
[10] S. K. Reddy, S. C. Straight, P. Bajaj, C. Huy Pham, M. Riera, D. R. Moberg, M. A. Morales, C. Knight, A. W. Götz, and F. Paesani, The Journal of Chemical Physics **145**, 194504 (2016).
[11] A. J. Skinner and J. Q. Broughton, Modelling and Simulation in Materials Science and Engineering (1995).
[12] J. Behler and M. Parrinello, Physical Review Letters **98**, 146401 (2007).
[13] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, Physical Review Letters **104**, 136403 (2010).
[14] A. V. Shapeev, Multiscale Modeling & Simulation **14**, 1153 (2016).
[15] Z. Li, J. R. Kermode, and A. De Vita, Physical Review Letters **114**, 096405 (2015).
[16] A. Glielmo, P. Sollich, and A. De Vita, Physical Review B **95**, 214302 (2017).
[17] V. Botu and R. Ramprasad, Physical Review B **92**, 094306 (2015).
[18] G. Ferré, T. Haut, and K. Barros, (2016), 1612.00193v1.
[19] E. V. Podryabinkin and A. V. Shapeev, Computational Materials Science **140**, 171 (2017).
[20] A. Takahashi, A. Seko, and I. Tanaka, (2017), 1710.05677.
[21] J. R. Boes, M. C. Groenenboom, J. A. Keith, and J. R. Kitchin, International Journal of Quantum Chemistry **116**, 979 (2016).
[22] C. K. I. Williams and C. E. Rasmussen, the MIT Press (2006).
[23] C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics (Springer, New York, NY, 2006).
[24] M. J. Kearns and U. V. Vazirani, *An Introduction to Computational Learning Theory* (MIT Press, 1994).
[25] M. Rupp, R. Ramakrishnan, and O. A. von Lilienfeld, The Journal of Physical Chemistry Letters **6**, 3309 (2015).
[26] V. L. Deringer and G. Csányi, arXiv.org (2016), 1611.03277v1.

[27] F. A. Faber, A. S. Christensen, B. Huang, and O. A. von Lilienfeld, The Journal of Chemical Physics **148**, 241717 (2018).
[28] A. P. Bartók, R. Kondor, and G. Csányi, Physical Review B **87**, 184115 (2013).
[29] T. Bereau, R. A. DiStasio Jr, A. Tkatchenko, and O. A. von Lilienfeld, arXiv.org (2017), 1710.05871v1.
[30] A. Grisafi, D. M. Wilkins, G. Csányi, and M. Ceriotti, arXiv.org (2017), 1709.06757v1.
[31] W. J. Szlachta, A. P. Bartók, and G. Csányi, Physical Review B **90**, 104108 (2014).
[32] H. Huo and M. Rupp, arXiv.org (2017), 1704.06439v1.
[33] V. Botu and R. Ramprasad, International Journal of Quantum Chemistry (2014).
[34] G. Ferré, J. B. Maillet, and G. Stoltz, The Journal of Chemical Physics **143**, 104114 (2015).
[35] L. Nachbin, *The Haar integral* (Van Nostrand, Princeton, 1965).
[36] H. Schulz-Mirbach, in *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3 - Conference C: Signal Processing (Cat. No.94CH3440-5* (IEEE, 1994) pp. 387–390 vol.2.
[37] B. Haasdonk and H. Burkhardt, Machine Learning **68**, 35 (2007).
[38] K. Hornik, Neural networks **6**, 1069 (1993).
[39] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker, Journal of Computational Physics **285**, 316 (2015).
[40] S. De, A. P. Bartók, G. Csányi, and M. Ceriotti, Physical Chemistry Chemical Physics **18**, 13754 (2016).
[41] P. Rowe, G. Csányi, D. Alfè, and A. Michaelides, arXiv.org (2017), 1710.04187v2.
[42] V. L. Deringer and G. Csányi, Physical Review B **95**, 094203 (2017).
[43] T. W. Anderson, The Annals of Mathematical Statistics **17**, 409 (1946).
[44] A. T. James, Proc. R. Soc. Lond. A **229**, 367 (1955).
[45] M. Rupp, A. Tkatchenko, K. R. Müller, and O. A. von Lilienfeld, Physical Review Letters **108**, 058301 (2012).
[46] B. Huang and O. A. von Lilienfeld, The Journal of Chemical Physics **145**, 161102 (2016).
[47] O. A. von Lilienfeld, R. Ramakrishnan, M. Rupp, and A. Knoll, International Journal of Quantum Chemistry **115**, 1084 (2015).
[48] M. I. Mendelev, S. Han, D. J. Srolovitz, G. J. Ackland, D. Y. Sun, and M. Asta, Philosophical Magazine **83**, 3977 (2003).
[49] J. P. Perdew, K. Burke, and M. Ernzerhof, Physical Review Letters **77**, 3865 (1996).
[50] C. Zeni, K. Rossi, A. Glielmo, N. Gaston, F. Baletto, and A. De Vita, arXiv.org (2018), 1802.01417v1.

APPENDIX

A. Kernel order by explicit differentiation

We first prove that the kernel given in Eq. (4) is 2-body in the sense of Eq. (2). For this it is sufficient to show that its second derivative with respect to the relative position of two different atoms of the target configuration ρ always vanishes. The first derivative is

$$\frac{\partial k_2(\rho, \rho')}{\partial \mathbf{r}_{i_1}} = \sum_{ij} \frac{\partial}{\partial \mathbf{r}_{i_1}} e^{-\|\mathbf{r}_i - \mathbf{r}'_j\|^2/4\sigma^2}$$

$$= \sum_{ij} e^{-\|\mathbf{r}_i - \mathbf{r}'_j\|^2/4\sigma^2} \frac{(\mathbf{r}_i - \mathbf{r}'_j)}{2\sigma^2} \delta_{ii_1}$$

$$= \sum_{j} e^{-\|\mathbf{r}_{i_1} - \mathbf{r}'_j\|^2/4\sigma^2} \frac{(\mathbf{r}_{i_1} - \mathbf{r}'_j)}{2\sigma^2}.$$

This depends only the atom located at $\mathbf{r}_{i_1}$ of the configuration ρ. Thus, differentiating with respect to the relative position $\mathbf{r}_{i_2}$ of any other atom of the configuration gives the relation in Eq. (2) for 2-body kernels:

$$\frac{\partial^2 k_2(\rho, \rho')}{\partial \mathbf{r}_{i_1} \partial \mathbf{r}_{i_2}} = 0.$$

We next show that the kernel defined in Eq. (6) is an n-body in the sense of Eq. (2). This follows naturally from the result above, given that k_n is defined as $k_n = k_2^{n-1}$. We can thus write down its first derivative as

$$\frac{\partial k_n}{\partial \mathbf{r}_{i_1}} = (n-1)k_2^{n-2} \frac{\partial k_2}{\partial \mathbf{r}_{i_1}}.$$

Since the second derivative of k_2 is null, the second derivative of k_n is simply

$$\frac{\partial^2 k_n}{\partial \mathbf{r}_{i_1} \partial \mathbf{r}_{i_2}} = (n-2)(n-1)k_2^{n-3} \frac{\partial k_2}{\partial \mathbf{r}_{i_1}} \frac{\partial k_2}{\partial \mathbf{r}_{i_2}}$$

and after $n-1$ derivations we similarly obtain

$$\frac{\partial^2 k_2^{n-1}}{\partial \mathbf{r}_{i_1} \cdots \partial \mathbf{r}_{i_n}} = (n-1)! \, k_2^0 \, \frac{\partial k_2}{\partial \mathbf{r}_{i_1}} \cdots \frac{\partial k_2}{\partial \mathbf{r}_{i_{n-1}}}.$$

Since $k_2^0 = 1$, the final derivative with respect to the n_{th} particle position $\mathbf{r}_{i_n}$ is zero as required by Eq. (2).

B. 1D n'-body model

To test the ideas behind the n-body kernels, we used a 1D n'-particle model reference system where a ("central") particle is kept fixed at the coordinate axis origin (consistent with the local configuration convention of Eq. (3)). The energy of the central particle is defined as

$$f = \sum_{i_1 \ldots i_{n'-1}} J \, x_{i_1} \ldots x_{i_{n'-1}}$$

where $\{x_{i_p}\}_{p=1}^{n'-1}$ are the relative positions of $n'-1$ particles, and J is an interaction constant.

To generate Figure 1 a large set of configurations was generated by uniformly and independently sampling each relative position x_{i_p} within the range $(-0.5, 0.5)$. The energy of the central particle of each configuration was then given by the above equation, with the interaction constant J set to 0.5. In order to analyse the converged properties of the n-body kernels presented, large training sets ($N = 1000$) were used.

C. Databases details

The bulk Ni and Fe databases were obtained from simulations using a $4 \times 4 \times 4$ periodically repeated unit cell, modelling the electronic exchange and correlation interactions via the PBE/GGA approximation [49], and controlling the temperature (set at 500K) by means of a weakly-coupled Langevin thermostat (the DFT trajectories are available from the KCL research data management system at the link http://doi.org/10.18742/RDM01-92). The C database comprises bulk diamond and AB- and ABC-stacked graphene layer structures. These structures were obtained from DFT simulations at varying temperatures and pressures, using a fixed $3 \times 3 \times 2$ periodic cell geometry for graphite, and simulation cells ranging from $1 \times 1 \times 1$ to $2 \times 2 \times 2$ unitary cells for diamond, the relative DFT trajectories can be found in the "libAtoms" data repository via the following link http://www.libatoms.org/Home/DataRepository. Crystalline and amorphous Si database was obtained from a microcanonical DFTB 64-atom simulation carried out in periodic boundaries, with average kinetic energy corresponding to a temperature of $T = 650K$. The results presented for the Ni cluster are reported from Ref. [50] and correspond to constant temperature DFT MD runs ($T = 300K$) of a particular Ni_{19} geometry (named "4HCP" in the article). The radial cutoffs used to create the local environments for the four elements considered are: 4.0 Å (Ni), 4.45 Å (Fe), 3.7 Å (C) and 4.5 Å (Si).

D. Details on the kernels used and on the experimental methodology

All energy kernels presented in the work can be used to learn/predict forces after generating a standard derivative kernel (Ref. [22], section 9.4, also cf. Section II A of the main text.) In particular, for each scalar energy kernel k a matrix-valued force kernel $\mathbf{K}$ can be readily obtained by double differentiation with respect to the positions ($\mathbf{r}_0$ and $\mathbf{r}'_0$) of the central atoms in the target and database configurations ρ and ρ':

$$\mathbf{K}(\rho, \rho') = \frac{\partial^2 k(\rho, \rho')}{\partial \mathbf{r}_0 \partial \mathbf{r}_0'^{\mathrm{T}}}.$$

The kernels $\tilde{k}_2$ and $\tilde{k}_3$ (Eqs. (19,20)) were chosen as simple squared exponentials in the tests shown. Noting as $\mathbf{q}$ (or q) the vector (or scalar) containing the effective degrees of freedom of the atomic n-plet considered (see Eq. (21)), the two kernels read:

$$\tilde{k}_2(q_i, q_j') = \mathrm{e}^{-(q_i-q_j')^2/2\sigma^2}$$

$$\tilde{k}_3(\mathbf{q}_{i_1,i_2}, \mathbf{q}'_{j_1,j_2}) = \sum_{\mathbf{P} \in \mathcal{P}_c} \mathrm{e}^{-\|\mathbf{q}_{i_1,i_2} - \mathbf{P}\mathbf{q}'_{j_1,j_2}\|^2/2\sigma^2},$$

where $\mathcal{P}_c$ ($|\mathcal{P}_c| = 3$) is the set of cyclic permutations of three elements. Summing over the permutation group is needed to guarantee permutation symmetry of the energy. As discussed in the main text, the exact form of these low-order kernels is not essential as the large database limit is quickly reached.

The many-body force kernel referred to in Fig. 5 was built as a covariant discrete summation of the many-body energy base-kernel (7) over the O_{48} crystallographic point group, using the procedure of Ref. [16]. This procedure yields an approximation to the full covariant integral of the many-body kernel (7) given in Eq. (5).

In order to obtain Figure 5, repeated (randomised) realisations of the same learning curves were performed. The points (and error bars) plotted are the means (and standard deviations) of the generated data . The kernel hyperparameters were independently optimised by cross validation for each dataset.