Article

Kernel-Based Machine Learning for Efficient Simulations of Molecular Liquids

Christoph Scherer, René Scheid, Denis Andrienko,* and Tristan Bereau*

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Current machine learning (ML) models aimed at learning force fields are plagued by their high computational cost at every integration time step. We describe a number of practical and computationally efficient strategies to parametrize traditional force fields for molecular liquids from ML: the particle decomposition ansatz to two- and three-body force fields, the use of kernel-based ML models that incorporate physical symmetries, the incorporation of switching functions close to the cutoff, and the use of covariant meshing to boost the training set size. Results are presented for model molecular liquids: pairwise Lennard-Jones, three-body Stillinger−Weber, and bottom-up coarse-graining of water. Here, covariant meshing proves to be an efficient strategy to learn canonically averaged instantaneous forces. We show that molecular dynamics simulations with tabulated two- and three-body ML potentials are computationally efficient and recover two- and three-body distribution functions. Many-body representations, decomposition, and kernel regression schemes are all implemented in the open-source software package VOTCA.
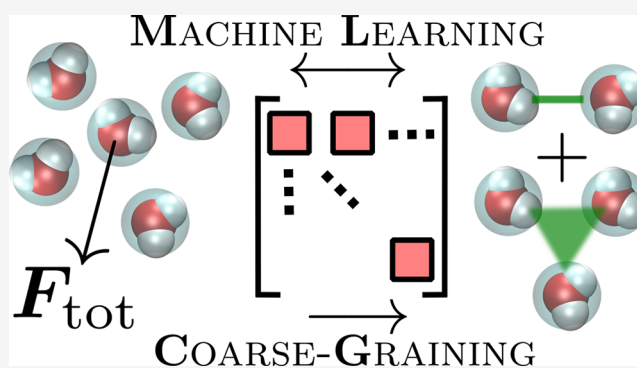
1. INTRODUCTION

Machine learning (ML) techniques have a long history of interpolating the high-dimensional potential energy surface (PES) of molecular systems.[1−4] Access to an accurate PES enables the simulation of the system's dynamics. In comparison to standard molecular dynamics (MD),[5] ML does not explicitly rely on fixed force field functions. This has recently resulted in impressive MD studies for various materials and molecular systems.[6−12]

To get there, it does not quite suffice to gather reference configurations and train a Gaussian process (GP) or neural network—the interpolation space is simply too large for all but the simplest of systems. Instead, incorporating physical symmetries in the ML model effectively reduces the interpolation space, thereby helping convergence. An early realization of this aspect was demonstrated by Behler and Parrinello, who relied on symmetry functions to incorporate translational and rotational invariance:[13] If the physics of the problem does not depend on arbitrary translations, encode particle geometries by means of relative *distances*. A number of molecular representations have since built on these ideas to offer more efficient learning performance.[11,14−16] Other aspects that aim at incorporating physics include local dynamical symmetries found in molecules[17] and the learning of tensorial properties.[18−20]

Incorporating symmetries does not have to be done at the level of the representation: one can also build it in the ML *architecture*. When building kernel-based ML models for force fields, the GP aims at inferring a vector-valued function.[21] The underlying vector field ought to remain curl-free—directly related to energy conservation in simple Hamiltonians—which can be implemented by learning the Hessian of the kernel.[22,23] Several variants of this strategy have by now been implemented to learn energy-conserving force fields.[6,10,24,25] In parallel, the learning of forces has sparked interest in building so-called *covariant* kernels, meaning that the prediction rotates with the sample in three-dimensional space. Glielmo et al. have derived an analytic expression for a covariant Gaussian kernel.[26] It effectively builds on the integration over all actions of the rotation group by means of a Haar measure.[27] Covariant ML models have more recently been developed for neural networks.[28,29]

In addition to symmetries, physical intuition and approaches can be further implemented to help build better ML models. Rather than learning the full PES at once, it can prove easier to converge a series of ML models that follow a body decomposition of the interactions, akin to a cluster expansion or virial equation for a real gas.[6,11,30−32] Parametrizing ML

models for each body term is not necessarily straightforward, because reference calculations typically contain *all* contributions at once. The contribution of each body is instead often inferred by means of a decomposition ansatz, which assumes the target property to be decomposable in terms of a linear combination of function values.[24,33]

Once trained, an ML model is ready to be used as a force predictor coupled to an MD integrator. In the context of ab initio MD simulations based on quantum-mechanical calculations, an ML model leads to significant performance improvements: While the former scales with the number of electrons raised to some power (≥ 3), the ML model will typically scale almost linearly with the number of *nuclei*, albeit with a large prefactor. John and Csányi[30] pointed out that the situation is very different when running coarse-grained (CG) simulations[34,35]—lumping together atoms into superparticles or beads. Since they are typically parametrized from classical atomistic simulations, the scaling will be similar, apart from a small reduction due to the reduced number of particles.[30] To remedy this situation and recover some computational efficiency, one can resolve to project the ML model onto standard tabulated potentials, as suggested by Glielmo et al.[31] This is the objective of the present study.

The scope of this paper is to summarize key aspects of applying GP regression to parametrize force fields of molecular liquids in a computationally efficient setting. A decomposition of the PES to two- and three-body interactions is presented, ultimately projected onto tabulated potentials to avoid making ML predictions at each integration time step. We compare several kernels that encode various levels of physics. Further technical refinements are incorporated, such as a smooth decay close to the interaction cutoff.[6] We also discuss performance during learning and structural accuracy in the condensed phase of a liquid. Results are presented for two example liquids: a two-body Lennard-Jones system and a three-body Stillinger–Weber system.

Finally, we extend our implementation to the coarse-graining of molecular liquids. CG is an appealing resolution in soft matter due to its ability to reach longer length and time scales.[35,36] We specifically focus on bottom-up strategies that systematically derive a CG model from higher-level microscopic information.[37] Consistency between the equilibrium probability densities of the two models leads to the many-body potential of mean force (MB-PMF), which effectively replaces the coveted PES.[38] One practical approach to building a CG model is force matching (FM) or multiscale coarse-graining (MS-CG),[38−40] which projects the MB-PMF into the space of force fields defined by the CG basis set. Both kernel-based and neural-network approaches have recently been applied to solving the MS-CG problem.[30,41,42] Akin to the study of John and Csányi,[30] we solve the MS-CG problem using kernels on both two- and three-body interactions, but we project the resulting decompositions onto computationally efficient tabulated potentials. We discuss both parametrization strategies and structural accuracy of the resulting condensed phase of liquid water.

2. KERNEL-BASED REGRESSION FOR FORCE FIELDS

We first recapitulate the terminology of a Gaussian process, keeping in mind physical quantities relevant to liquids. A sample i in our data set will represent particle i and its environment, which we will specify later. A particle can be an atom or a molecule.

We propose to learn a mapping $Q \rightarrow O$ between the representation of sample i, Q_i, and an observable, $O_i = O(Q_i)$. The observable may correspond to the sample's energy or force, for instance. Training a kernel model is then equivalent to solving the set of linear equations $O = \hat{K}\alpha$, where the kernel function $\hat{K}_{ij} = K(Q_i, Q_j) = \mathrm{Cov}(O_i, O_j)$ measures the similarity between samples Q_i and Q_j, and α is the vector of weight coefficients. The latter is optimized by inverting the regularized problem

$$\alpha = (\hat{K} + \lambda \mathbb{I})^{-1} O \qquad (1)$$

where λ implements Tikhonov regularization. Prediction of the observable for a configuration Q^* is then given by an expansion of the kernel evaluations on the training set

$$O(Q^*) = \sum_{i=1}^{N} \alpha_i K(Q_i, Q^*) \qquad (2)$$

summing over the N training points.

2.1. Particle-Decomposition Ansatz. The above-mentioned formulation can, in principle, be applied to learn local properties of a liquid, such as forces, F. Reference high-level simulations provide the force of a particle embedded in its environment, represented by Q. A kernel machine can then be set out to learn the mapping $Q \rightarrow F$. Learning this mapping directly is in general challenging, due to the interpolation size of Q. Instead, we aim at breaking down the problem to ease convergence of the kernel and avoid extrapolation.

We first limit the size of our particle environments or configurations, Q, by setting a cutoff, r_{cut}. In other words, for a sample i only the neighbors j with $r_{ij} \leq r_{\mathrm{cut}}$ are added to the environment Q_i. Note that this decomposition assumes that r_{cut} is large enough to account for the physical interaction range, which is not always possible, e.g., for unscreened long-range Coulomb interactions.

Q are further decomposed in *body* terms, q, which consist of interparticle interactions: from pairwise, to triplets, all the way to fully many-body. In the following we limit our approach to the first two terms, though the method generalizes in a straightforward manner.

The potential benefit of the body expansion is its reasonable convergence, which helps to reduce the size of Q. We denote the body decomposition using a superscript: $q^{(2)}$ and $q^{(3)}$ for vertices of pair and triplet interactions. These interparticle interactions can be associated with a set of local forces, $f^{(2)}$ and $f^{(3)}$, which are a priori unknown. The present framework thus motivates several mappings

$$Q \xrightarrow{\hat{K}} F \quad \text{and} \quad q^{(b)} \xrightarrow{\hat{k}^{(b)}} f^{(b)} \qquad (3)$$

where $\hat{K}$ and $\hat{k}^{(b)}$ denote the global and local kernels and b runs over pairwise and triplet models.

While learning a local model is a more attractive option, we do not have direct access to the local forces $f^{(2)}$ and $f^{(3)}$. The particle-decomposition ansatz assumes that the force acting on a particle due to its environment can be approximated by a set of constitutive interparticle interactions

$$F(Q_i) = \sum_{p=1}^{P_i} f_o^{(2)}(q_{i_p}^{(2)}) + \sum_{t=1}^{T_i} f_o^{(3)}(q_{i_t}^{(3)}) \qquad (4)$$

where p and t run over P_i pair and T_i triplet interactions. Each pair interaction is between a pair of particles, $p = (ab)$, and

each triplet is between three particles, $t = (abc)$, where a, b, and c are the indices of particles in a pair or triplet; o is the index of the central particle of configuration i, in a pair p or triplet t. The body expansion and the indexing are illustrated in Figure 1. The total number of interparticle interactions for sample Q_i is thus $M_i = P_i + T_i$.
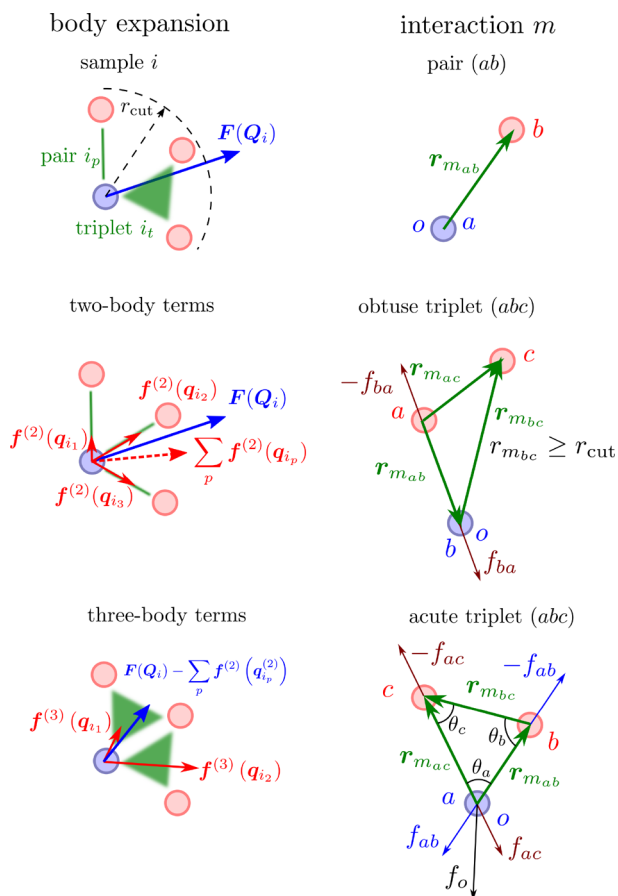

Figure 1. (left) Illustration of the body expansion of a configuration i on pairs and triplets, where the reference force $F(Q_i)$ acts on the central particle. The two-body terms are shown by interparticle vectors (green lines) and result in total pairwise force $\sum_p f^{(2)}(q_{i_p}^{(2)})$. A similar decomposition for triplets is used to fit the residual, i.e., total minus two-body, force. (right) Representations used for pairs and triplets. a, b, and c are the particles of the pair or triplet, and o is the central particle of the sample, which can be either a, b, or c. For triplets, two situations are possible: For obtuse triplets, with one edge larger than cutoff, the force on the central particle $o = b$ is due to the reaction force on particle a. For acute triplets with all edges below the cutoff, the force on the central particle $o = a$, f_o, is a sum of two forces, f_{ac} and f_{ab}, which depend on all three angles, θ_a, θ_b, and θ_c. For vector differences, we use the standard convention, $r_{m_{ab}} = r_{m_b} - r_{m_a}$.

Equation 4 formally links the kernels $\hat{K}$, $\hat{k}^{(2)}$, and $\hat{k}^{(3)}$. We will use the target properties available for the global kernel, $\hat{K}$, in order to infer an ML model for the local interactions. To this end, we concatenate a set of N particle environments together with the corresponding set of M local interactions

$$F = \hat{L}f \tag{5}$$

where we introduce the mapping matrix $\hat{L} = (\hat{\mathcal{L}}_{i_m})$. Its component, $\hat{\mathcal{L}}_{i_m}$, is a 3×3 identity matrix, $\mathbb{I}_{3\times3}$, if particle

environment i contains the pair or triplet interaction m, and a 3×3 zero matrix, $\mathbb{O}_{3\times3}$, otherwise. $\hat{L}$ is the bookkeeping matrix that connects the local interactions with the right particle environments. It has dimension $3N \times 3M$ due to the three spatial dimensions.

We can now relate the particle-environment kernel $\hat{K}$ to its local counterpart, $\hat{k}$, by

$$\hat{K}(F_i, F_j) = \text{Cov}\left(\sum_{m=1}^{M_i} \hat{\mathcal{L}}_{i_m} f_m, \sum_{l=1}^{M_j} \hat{\mathcal{L}}_{j_l} f_l\right) = (\hat{L}\hat{k}\hat{L}^T)_{ij} \tag{6}$$

leading to the relationship $\hat{K} = \hat{L}\hat{k}\hat{L}^T$. Similar to $\hat{L}$, both $\hat{K}$ and $\hat{k}$ have a block matrix structure, $\hat{K} = (\hat{\mathcal{K}}_{ij})$ and $\hat{k} = (\hat{\kappa}_{ml})$, where elements $\hat{\mathcal{K}}_{ij}$ and $\hat{\kappa}_{ml}$ are 3×3 matrices.

Training a model then corresponds to solving the equation $F = \hat{K}\alpha$ and taking advantage of the relationship between global and local kernels

$$\alpha = (\hat{K} + \lambda\mathbb{I})^{-1}F = (\hat{L}\hat{k}\hat{L}^T + \lambda\mathbb{I})^{-1}F \tag{7}$$

with a regularization term λ and a $3N$-dimensional vector of weight coefficients α.

Once trained, the ML model can be used for predictions of the local interactions, f^*. An analogous link between the covariance matrices of the global and local kernels for prediction only yields

$$f^* = (\hat{L}\hat{k}^*)^T\alpha \tag{8}$$

where $\hat{k}^*$ denotes the local kernel between training and test data.

Note that the prediction for entire particle environments yields a new mapping matrix $\hat{L}^*$ for the test data,

$$F^* = \hat{L}^*(\hat{L}\hat{k}^*)^T\alpha \tag{9}$$

2.2. Representations. It is useful to describe local interactions in terms of internal coordinates as they are invariant with respect to translations. For pairwise interactions we therefore use the interparticle vector of a pair m, $r_{m_{ab}} = r_{m_b} - r_{m_a}$ to represent a pair of particles a and b. For triplet interactions we use the three interparticle vectors, $r_{m_{ab}}$, $r_{m_{ac}}$, and $r_{m_{bc}}$.

When predicting a rotationally invariant property, such as energy, or when building a covariant kernel from a rotationally invariant one, it is sufficient to use a rotationally invariant representation. For a pair m with particles (a, b) we use the interparticle distance

$$q_m^{(2)} = r_{m_{ab}} \tag{10}$$

while for a triplet m with particles (a, b, c) we use a vector of three interparticle distances

$$q_m^{(3)} = (r_{m_{ab}}, r_{m_{ac}}, r_{m_{bc}})^T \tag{11}$$

3. COVARIANT, CONSERVATIVE FIELDS

If the target property is a vector, such as a force, we expect the kernel to behave like a second rank tensor upon rotations of coordinates. Formally, this can be expressed as[26]

$$\hat{\kappa}_c(\hat{\mathcal{R}}(q_m), \hat{\mathcal{R}}'(q_l)) = \hat{\mathcal{R}}\hat{\kappa}_c(q_m, q_l)\hat{\mathcal{R}}'^T \tag{12}$$

for any two rotation matrices $\hat{\mathcal{R}}$ and $\hat{\mathcal{R}}'$. This condition imposes restrictions on the form of the kernel $\hat{\kappa}_c$, forcing it to become covariant.

Apart from covariance, we are interested in *conservative* force fields, for which the total energy of a particle is conserved. Mathematically, the force field has to be curl-free or be a gradient of a scalar function (energy).

We now discuss three different approaches to construct a matrix-valued kernel, for which the target property rotates with the sample and is curl-free.

3.1. Explicit Rotations. We first propose to encode covariance in explicit rotations of individual local representations. To do this, we use the covariance of local kernels

$$\hat{\kappa}(\hat{\mathcal{R}}_m(\boldsymbol{q}_m), \hat{\mathcal{R}}_l(\boldsymbol{q}_l)) = \hat{\mathcal{R}}_m \hat{\kappa}(\boldsymbol{q}_m, \boldsymbol{q}_l)\hat{\mathcal{R}}_l^{\mathrm{T}} \tag{13}$$

where for every local interaction m we introduced a rotation matrix $\hat{\mathcal{R}}_m$.

To relate all M pairs or triplets with their local counterparts, we introduce a block-diagonal matrix of rotation matrices

$$\hat{R} = \mathrm{blkdiag}(\hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2, ..., \hat{\mathcal{R}}_M) \tag{14}$$

We can then link the $3N \times 3N$ dimensional global kernel, $\hat{K}$, with the $3M \times 3M$ dimensional local kernels, $\hat{k}^{(2)}$ and $\hat{k}^{(3)}$

$$\hat{K} = \hat{L}\hat{R}\hat{k}(\hat{R}(\boldsymbol{q}), \hat{R}(\boldsymbol{q}))\hat{R}^{\mathrm{T}}\hat{L}^{\mathrm{T}} \tag{15}$$

which is a variation of eq 6 that includes rotations of individual local interactions.

As such, training a kernel model amounts to a modification of eq 7

$$\boldsymbol{\alpha} = (\hat{L}\hat{R}\hat{k}\hat{R}^{\mathrm{T}}\hat{L}^{\mathrm{T}} + \lambda\mathbb{1})^{-1}\boldsymbol{F} \tag{16}$$

and prediction becomes

$$\boldsymbol{F}^* = \hat{L}^*\hat{R}^*(\hat{L}\hat{R}\hat{k}^*)^{\mathrm{T}}\boldsymbol{\alpha} \tag{17}$$

We now chose rotation matrices such that $\hat{\mathcal{R}}_m(\boldsymbol{q}_m)$ becomes invariant with respect to sample rotations, by rotating the individual pairs or triplets into a *fixed* axis system. In practice, we align pairs along the z-axis, and triplets are aligned in the zy plane, as explained in the Supporting Information.

In the fixed frame interaction representations become rotationally invariant and we can use representations given by eqs 10 and 11 and a standard Gaussian kernel to compare them:

$$k_{ml}(\boldsymbol{q}_m, \boldsymbol{q}_l) = \exp\left(-\sum_\alpha \frac{(q_{m_\alpha} - q_{l_\alpha})^2}{2\sigma_\alpha^2}\right) \tag{18}$$

The covariance in this scheme is encoded into the explicit rotation matrix $\hat{R}$. In practice, $\hat{R}$ and $\hat{L}$ can be combined by replacing unit 3×3 matrices in $\hat{L}$ by corresponding rotation matrices.

Note that each local kernel element $\hat{\kappa}_{ml}$ is matrix-valued. We can assume that all three components are independent. However, since such kernel does not yield a conservative force, the ML model will have to learn it empirically. In the Supporting Information we derive an explicit expression for $\hat{\kappa}_{ml}$ that results in a conservative three-body kernel.

3.2. Integration over SO(3). A covariant matrix-valued kernel can also be constructed by integrating a scalar kernel over all rotation matrices, effectively summing over all actions of the rotation group:[26,31]

$$\hat{\kappa}_c(\boldsymbol{q}_m, \boldsymbol{q}_l) = \int d\hat{\mathcal{R}}\,\hat{\mathcal{R}}k_b(\boldsymbol{q}_m, \hat{\mathcal{R}}\boldsymbol{q}_l) \tag{19}$$

where $\hat{\kappa}_c$ is the covariant 3×3 matrix-valued kernel, while k_b represents a base scalar-valued kernel. Glielmo et al. derived an analytical solution for pairs, starting from a Gaussian kernel, $k_b^{(2)}(\boldsymbol{q}_m, \boldsymbol{q}_l) = \exp(-(r_{m_{ab}} - r_{l_{ab}})^2/2\sigma^2)$, yielding[26]

$$\hat{\kappa}_c^{(2)}(\boldsymbol{q}_m, \boldsymbol{q}_l) = e^{-\alpha_{ml}}\left(\frac{\cosh\gamma_{ml}}{\gamma_{ml}} - \frac{\sinh\gamma_{ml}}{\gamma_{ml}^2}\right)\hat{\boldsymbol{r}}_{m_{ab}}\hat{\boldsymbol{r}}_{l_{ab}}^{\mathrm{T}} \tag{20}$$

where $\alpha_{ml} = r_{m_{ab}}^2 + r_{l_{ab}}^2/4\sigma^2$, $\gamma_{ml} = r_{m_{ab}}r_{l_{ab}}/2\sigma^2$, and $\hat{\boldsymbol{r}}_{m_{ab}} = r_{m_{ab}}/r_{m_{ab}}$. Note that covariance of $\hat{\kappa}_c^{(2)}(\boldsymbol{q}_m, \boldsymbol{q}_l)$ is encoded in the tensor product between the two interparticle vectors, $\hat{\boldsymbol{r}}_{m_{ab}}\hat{\boldsymbol{r}}_{l_{ab}}^{\mathrm{T}}$.

3.3. Hessian of the Scalar Kernel. An alternative statement consists in recognizing that the local forces are spatial derivatives of the potential energy of a respective pair or triplet, $\boldsymbol{f} = -\nabla E_m$. As such, a kernel between two forces can be expressed through the energies involved, and that the spatial derivatives can be taken out of the scalar kernel function

$$\hat{\kappa}_h(\boldsymbol{q}_m, \boldsymbol{q}_l) = \mathrm{Cov}\left(\frac{\partial E(\boldsymbol{q}_m)}{\partial \boldsymbol{r}_{m_o}}, \frac{\partial E(\boldsymbol{q}_l)}{\partial \boldsymbol{r}_{l_o}}\right)$$

$$= \sum_{\alpha,\beta} \frac{\partial^2 k(\boldsymbol{q}_m, \boldsymbol{q}_l)}{\partial q_{m,\alpha}\partial q_{l,\beta}}\left(\frac{\partial q_{m,\alpha}}{\partial \boldsymbol{r}_{m_o}}\right)\left(\frac{\partial q_{l,\beta}}{\partial \boldsymbol{r}_{l_o}}\right)^{\mathrm{T}} \tag{21}$$

where we neglected the second term that contains a first-order partial derivative of the kernel function, because they will be zero around training points. Here, α and β run over the different components of the representation vector; o is the central particle of the samples, as illustrated in Figure 1.

The Hessian-based kernel is proportional to the tensor product

$$\left(\frac{\partial q_{m,\alpha}}{\partial \boldsymbol{r}_{m_o}}\right)\left(\frac{\partial q_{l,\beta}}{\partial \boldsymbol{r}_{l_o}}\right)^{\mathrm{T}}$$

and is therefore covariant, similar to the covariant kernel of section 3.2. In addition, it is energy conserving, since it ensures a curl-free force field.

We now choose a Gaussian kernel, eq 18, as scalar-valued kernel. For pairwise interactions, the representation (10) results in an energy-conserving force kernel:

$$\hat{\kappa}_h^{(2)}(\boldsymbol{q}_m, \boldsymbol{q}_l) = k^{(2)}(r_{m_{ab}} - r_{l_{ab}})\hat{\boldsymbol{r}}_{m_{ab}}\hat{\boldsymbol{r}}_{l_{ab}}^{\mathrm{T}} \tag{22}$$

where

$$k^{(2)}(r) = \left[\frac{1}{\sigma^2} - \frac{r^2}{\sigma^4}\right]\exp\left(-\frac{r^2}{2\sigma^2}\right)$$

$\hat{\boldsymbol{r}}_{m_{ab}} = (\delta_{m_o,m_b} - \delta_{m_o,m_a})r_{m_{ab}}/r_{m_{ab}}$; the expression for $\hat{\boldsymbol{r}}_{l_{ab}}$ can be obtained by exchanging m with l in $\hat{\boldsymbol{r}}_{m_{ab}}$. Note that the Kronecker deltas only change the sign of the kernel element, depending on whether the central particle of the sample, o, coincides with particle a or b. The tensor product, $\hat{\boldsymbol{r}}_{m_{ab}}\hat{\boldsymbol{r}}_{l_{ab}}^{\mathrm{T}}$, is responsible for the covariance of the kernel, similar to the

kernel obtained by integrating over SO(3). The scalar prefactor is, however, different.

For a three-body kernel, the representation (11) has three components, and the Hessian-based kernel becomes a sum over the three pair distances in a triplet:

$$\hat{k}_h^{(3)}(\boldsymbol{q}_m, \boldsymbol{q}_l) = \sum_{p \in ab, ac, bc} k_{ml}^{(3)}(r_{m_p} - r_{l_p})\hat{r}_{m_p}\hat{r}_{l_p}^{\mathrm{T}} \tag{23}$$

with

$$k_{ml}^{(3)}(r) = \left[\frac{1}{\sigma_p^2} - \frac{r^2}{\sigma_p^4}\right]\exp\left(-\sum_p \frac{(r_{m_p} - r_{l_p})^2}{2\sigma_p^2}\right)$$

Its covariance and conservativeness are now explicit: it is a sum of three tensor products and three radial fields. Generalization to any descriptor and scalar kernel is given in the Supporting Information.

We now compare the performance of three ML models for Lennard-Jones (two-body) and Stillinger–Weber (three-body) fluids.

4. TWO-BODY LENNARD-JONES FLUID

Simulation details of the Lennard-Jones (LJ) fluid are given in the Supporting Information. In Figure 2 we compare the
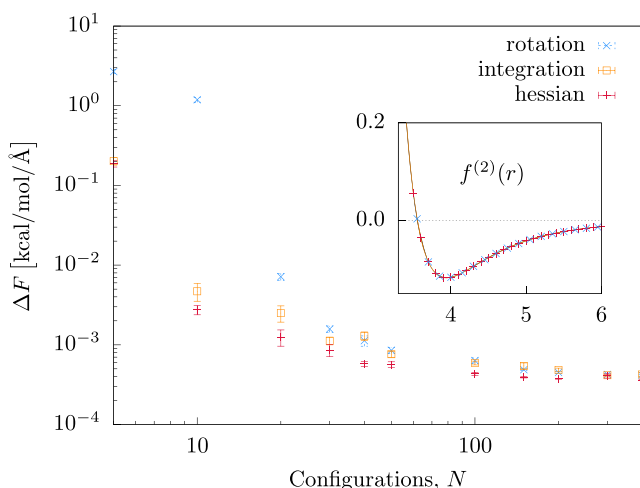
Figure 2. Learning curves of the pair-force kernels applied to the LJ fluid. ΔF is the force error per configuration; N is the number of training configurations. The average number of pairs per configuration is 29. (inset) Predicted force fields, $f^{(2)}(r)$, are virtually identical to the original LJ reference. Each ML prediction is based on 400 training configurations.

learning curves of the integration, Hessian, and rotation kernels. The LJ force field is learned accurately and efficiently irrespectively of the kernel. The ML model saturates at around 100 particle environments or configurations in the training set, corresponding to roughly 2900 reference pairwise interactions, with an error of $\Delta F \approx 4 \times 10^{-4}$ kcal/mol/Å per configuration. For $r_{\mathrm{cut}} = 6.0$ Å the average number of pairs per particle environment is 29, implying a mean error per pair of $\Delta f^{(2)} = \Delta F / \sqrt{29} \approx 8 \times 10^{-5}$ kcal/mol/Å. Here we assume that the M_i individual pair forces of each configuration i are independent and their errors have the same magnitude. A per-pair error is close to the output precision of the reference forces in the simulation trajectory, 10^{-5} kcal/mol/Å. There is

virtually no difference between the performances of the three different kernels, suggesting that they all encode the same amount of physics relevant to this system—they are covariant and curl-free—which is evident for radial fields.

The prediction of the force field projected on the pairwise distance r leads to a quantitative recovery of the reference potential, as shown in the inset of Figure 2 for $N = 400$ configurations.

5. THREE-BODY STILLINGER–WEBER FLUID

Next, we compare the learning of the three-body forces of the Stillinger–Weber (SW) fluid. Simulation details of the SW fluid are given in the Supporting Information. Note that we only compare the Hessian and the rotation kernels, since an analytic expression for the integration kernel exists only for the scalar case.[31] In Figure 3 we compare the learning curves for
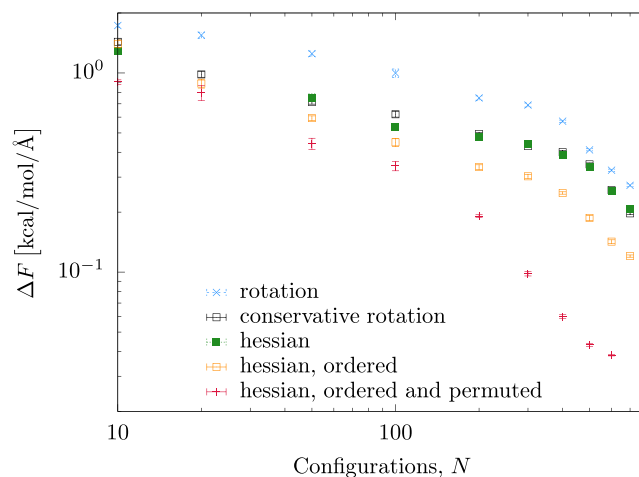
Figure 3. Learning curves of the three-body force kernels applied to the SW fluid. ΔF is the force error per configuration; N is the number of training configurations. The average number of triplets per configuration is 43 in all cases except the last curve. The Hessian kernel with ordered representation and all permutations has on average 52 triplets per particle, due to a different decomposition of the configuration into three-body interactions.

the Hessian (green), rotation (blue), and conservative rotation (black) kernels. One can clearly see that the nonconservative kernel performs significantly worse than the Hessian kernel, but once we make it conservative, its performance becomes comparable to that of the Hessian kernel.

In spite of the fact that the conservative rotation and Hessian kernels encode all physical symmetries of the force field, one can, however, further improve the ML model. To do this, we recall that the efficiency of the body decomposition relies, in part, on the interaction cutoffs: only those triangles which have two edges smaller than the cutoff contribute to the body expansion. If the central vertex of the triangle, o, is linked to an edge longer that the cutoff (e.g., bc in the obtuse triplet shown in Figure 1), the force on this particle is only due to the shorter edge. In other words, a representation that emphasizes the shorter edge would lead to a faster learning. One can enforce this asymmetry by sorting the triplet representation vector, eq 11, with respect to the edge lengths. Indeed, the corresponding learning curve, Figure 3, orange, learns faster than the model with a simple descriptor.

Similarly, for triangles with all three edges below the cutoff, the force on every vertex is a sum of two forces originating from the neighboring edges, as depicted for an acute triplet in Figure 1. Each force depends on the three angles of the triangle, and their "individual" representations differ only by permutations of elements of the representation vector, eq 11. Hence, the optimal representation for the total force on a particle should include three permutations of this representation. In fact, adding these permutations, in addition to ordering the descriptor, leads to an even better ML model, with the best learning curve shown in Figure 3, red symbols. In other words, this representation captures well the internal decomposition of the three-body force on three vertices of the triangle. This is the model we will be using in the remainder of this work, referring to it as a Hessian kernel model. For this model, 600 training configurations result in the average error of 0.04 kcal/mol/Å per configuration or of 0.006 kcal/mol/Å per triplet, given an average of 52 triplets per configuration.

6. CUTOFFS AND SWITCHING FUNCTIONS

To avoid numerical instabilities and ensure energy conservation in an MD simulation, the short-range forces ought to smoothly decay to zero at the cutoff distance, r_{cut}. This is indeed the case for the LJ force field learned above; see the inset of Figure 2. If we, however, use the ML model for the SW fluid and predict forces for a triangle with fixed angles, the force shows some fluctuations at r_{cut}, as shown Figure 4a. This is due to the limited training set size.

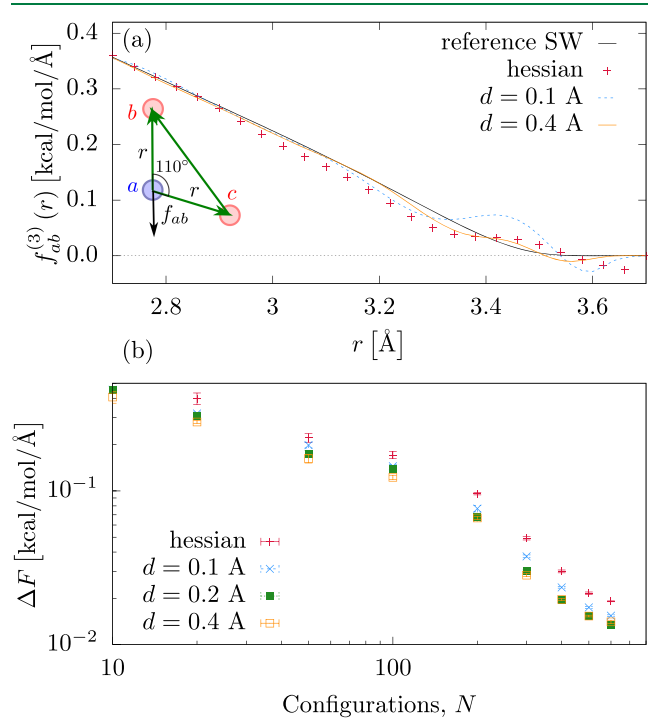One way to correct for these artifacts is to enforce smoothness via a switching function[6]

Figure 4. Comparison of the three-body Hessian kernel applied to the SW fluid, without and with switching function for $d = 0.1$ and $d = 0.4$ Å. In each case the ML models were based on 600 configurations. (a) Force scan of particle a along $r_{m_{ab}}^{(3)}$. (b) Learning curves for the different ML models.

$$f_{cut}(r)$$
$$= \begin{cases} 1, & r \leq r_{cut} - d \\ \frac{1}{2}\cos\frac{\pi(r - r_{cut} + d)}{d} + \frac{1}{2}, & r_{cut} - d < r < r_{cut} \\ 0, & r \geq r_{cut} \end{cases}$$

(24)

which goes smoothly to zero in the transition region of $r_{cut} - d < r < r_{cut}$.

In case of a standard force field one would multiply the interparticle force with the cutoff function $f_{cut}(r)$ with an appropriate transition region d. Applying $f_{cut}(r)$ to the ML model means that each entry of the mapping matrix $\hat{\mathcal{L}}_{i_m}$ is multiplied by $f_{cut}(r_{m_{ab}})$ for pairs and $f_{cut}(r_{m_{ab}})\,f_{cut}(r_{m_{ac}})$ for triplets. The same distance-dependent weight has to be applied to the mapping matrix of the training, $\hat{L}$, and prediction, $\hat{L}^*$. As such, pairs and triplets that are close to the cutoff distance contribute *less* to the GP regression, guaranteeing a smooth decay of the prediction to zero at r_{cut}.

In Figure 4a we show force cross sections of triplets trained on $N = 600$ configurations of the three-body SW fluid using the Hessian kernel for different smoothing functions. One can see that incorporating f_{cut} suppresses fluctuations around the cutoff. This is also reflected in the learning curves, shown in Figure 4b. The lowest test error per particle environment is about $\Delta F = 0.03$ kcal/mol/Å for $M = 600$ configurations, obtained with $d = 0.4$ Å, corresponding to a mean test error per triplet of about $\Delta f^{(3)} = \Delta F_{conf}/\sqrt{52} \approx 0.004$ kcal/mol/Å. Though the accuracy is not significantly improved for the SW liquid, using the switching function is essential for coarse-grained force fields, discussed in section 9, to ensure that the mean force decays to zero at the cutoff.

7. COVARIANT MESHING

The error of the three-body force kernel predictions could be further decreased by increasing the training set size. This, however, is not possible due to the size of the kernel matrix, which reaches the computer memory limits: While the maximum number of configurations N is relatively small (hundreds), they involve tens of thousands of triplet interactions M. To bound the size of the kernel, we propose to mesh the representation. To make meshing efficient, we make use of the kernel covariance: All samples that are only rotated with respect to each other are assigned to the same bin. In other words, we mesh the rotationally invariant representation and supplement it with the rotation matrices. Apart from reducing the size of the kernel, meshing averages over forces, which as we will see in section 9 results in smoother coarse-grained force fields.

In practice, we precompute the pair or triplet kernel $\hat{k}_{bins}$ on a fixed grid, limiting the number of kernel elements to $M_{bins} \times M_{bins}$ for training. Due to the covariance of the force kernels, one can choose an arbitrary reference orientation of the binned pair and triplet representations. For the two-body kernel, we use a linear binning with M_{bins} between r_{min} and r_{max} and an orientation of the pair vector along the z-axis. For the three-body kernel, we choose a uniform binning on a three-dimensional grid, as described in the Supporting Information.

The binned pairs and triplets are rotated to the pairs and triplets of each sample with the help of the mapping matrix

$\hat{L}_{\text{bins}} = (\hat{\mathcal{L}}_{i_m}^{\text{bins}})$. The index i denotes the N configurations and m denotes the M_{bins} pairs or triplets in the set of binned representations. Each matrix element $\hat{\mathcal{L}}_{i_m}^{\text{bins}}$ is a 3×3 matrix:

$$\hat{\mathcal{L}}_{i_m}^{\text{bins}} = \sum_{i_l} w_{i_l m} \hat{\mathcal{R}}_{i_l m} \tag{25}$$

where i_l denotes pairs or triplets of configuration i; $\hat{\mathcal{R}}_{i_l m}$ is the rotation matrix that rotates pair or triplet i_l to the fixed orientation of the binned representation m.

We apply a Gaussian smearing, $w_{i_l m}$, so that a pair or triplet i_l is related to several binned representations. For pairs

$$w_{i_l m}(r_{ab}) = \exp\left[-\frac{(r_{i_{l,ab}} - r_{m_{ab}}^{\text{bins}})^2}{2\sigma_r^2}\right] f_{\text{cut}}(r_{i_{l,ab}}) \tag{26}$$

where $r_{i_{l,ab}}$ and $r_{m_{ab}}^{\text{bins}}$ are pair distances of the actual and binned pairs. For triplets the weights are a product of three Gaussians:

$$w_{i_l m} = w_{i_l m}(r_{ab})\, w_{i_l m}(r_{ac})\, w_{i_l m}(\theta) \tag{27}$$

The learning and predicting can be readily derived from eqs 7−9 and are given in the Supporting Information.

In Figure 5 we compare the meshed MLs for the LJ (two-body) and SW (three-body) liquids. As expected, reducing the number of bins systematically saturates the error to higher values, since it lowers the resolution of the predicted force field. Covariant meshing does allow us to include another 2 orders of magnitude in training data, but the high-resolution
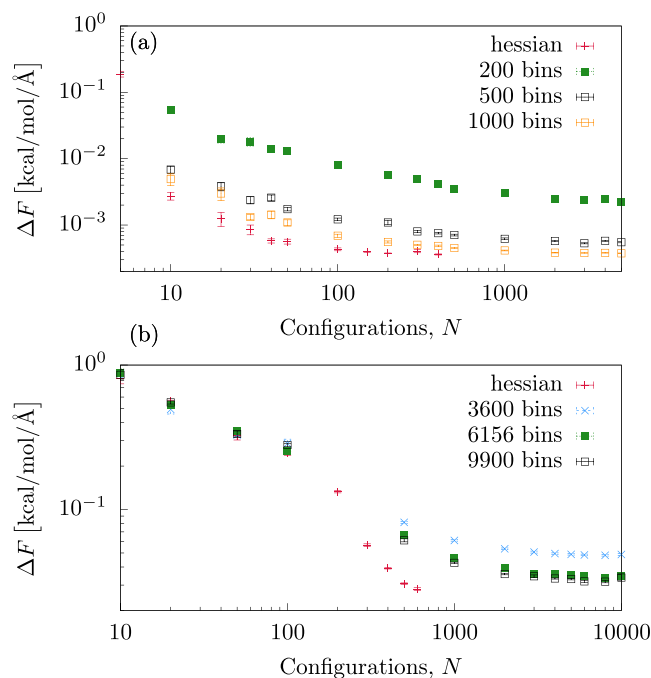
1000-bin (LJ) or 9900-bin kernels (SW) do not result in smaller errors than the original kernel. We also do not observe a faster *rate* of learning. We therefore conclude that the covariant meshing is not needed to reduce the error for ML models of simple systems. However, as stated above, machine learning of *coarse-grained* force fields requires much larger training set sizes. We will therefore see that meshing is unavoidable and is, in fact, the key for building an accurate coarse-graining model.

8. TABULATED FORCE FIELDS

We now use the ML force field to run MD simulations. To balance the flexibility of ML with the necessity for quick force evaluations,[31] we project the ML model on tabulated two- and three-body potentials, as described in the Supporting Information.

Figure 6 compares the distribution functions of the original SW model with the tabulated kernel predictions of the three-
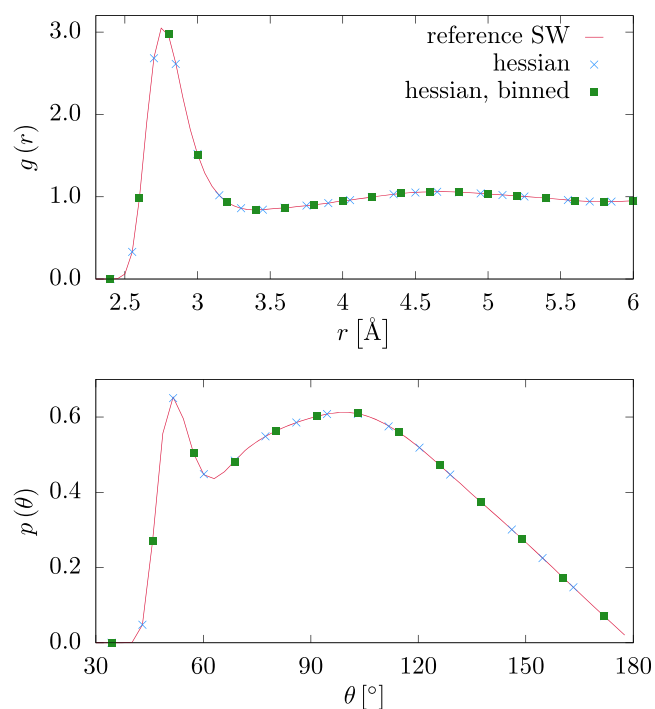

Figure 6. Comparison of MD simulation of tabulated three-body kernel predictions with the reference SW simulation. $g(r)$ is the pair; $p(\theta)$ is the angular distribution functions. $p(\theta)$ is calculated using a cutoff of 3.7 Å. We use a Hessian kernel with a switching function and $d = 0.4$ Å, 600 configurations without binning, and 10 000 configurations with 9900 bins.

Figure 5. (a) Learning curves of pair force kernels for the LJ fluid. Comparison of test errors for Hessian kernel with $\sigma = 0.5$ Å without and with binning with different number of bins from 200 to 1000. (b) Learning curves of three-body force kernels for the SW fluid. Comparison of test errors for Hessian kernel with switching function (eq 24) with $d = 0.4$ Å without and with binning with different number of bins from 3600 to 9900. The optimized hyperparameters are $\sigma_1 = 0.7$ Å, $\sigma_2 = 0.7$ Å, and $\sigma_3 = 1.4$ Å.

body forces. All curves are virtually identical, confirming that the kernel predictions projected on a tabulated potential lead to the correct sampling of the canonical ensemble. Note that the computational cost of these simulations is comparable to the original three-body SW potential, but there is no restriction on the functional form of the three-body potential, which in this case exactly matches the SW potential. The performance on a 20 core node with MPI paralellization is about 130 ns/day for the SW potential, about 140 ns/day for the tabulated kernel prediction with binning, and about 144 ns/day for the tabulated kernel prediction without binning. Evaluating the kernel prediction at each simulation time step would be more computationally expensive by orders of magnitude.[43] While a

kernel prediction at each MD time step of an all-atom simulation is competitive with quantum-mechanical calculations, the benefit of a coarse-grained simulation with respect to its atomistic counterpart is not necessarily straightforward.

9. COARSE-GRAINING

We move on to the challenging task of learning a coarse-grained (CG) force field for molecular liquids. The idea of coarse-graining is to reduce the number of degrees of freedom by combining several atoms into one CG particle. We assume that the position of the CG particle i, $\boldsymbol{R}_i$, is its center of mass:

$$\boldsymbol{R}_i = \frac{1}{\sum_{j \in i} m_j} \sum_{j \in i} m_j \boldsymbol{r}_j \tag{28}$$

We focus on a one-bead CG model of liquid SPC/E water.[44] Water is a particularly interesting liquid, as it has a strong local orientational order and therefore has attracted significant attention also regarding CG modeling. It is well-known that especially for CG water models pair potentials alone fail to fully capture the structural correlations of the atomistic system. To do so, at least three-body interaction terms are required.[45−48]

Mapping the atomistic degrees of freedom to a CG representation leads to a degeneracy: many fine-grained configurations map to a single point in the conformational space of the CG model.[35] We will refer to the reference coarse-grained atomistic forces as instantaneous collective forces (ICFs):

$$\boldsymbol{F}_i(\boldsymbol{r}) = \sum_{j \in i} \boldsymbol{f}_j(\boldsymbol{r}) \tag{29}$$

where $\boldsymbol{f}_j$ are the atomistic forces of each atom j assigned to bead i.

We now introduce the $3N$-dimensional block vector of the N instantaneous collective forces of the N CG particles, $\boldsymbol{F}(\boldsymbol{r}) = (\boldsymbol{F}_1, \boldsymbol{F}_2, ..., \boldsymbol{F}_N)^{\mathrm{T}}$. Noid and co-workers showed that the *mean* force, $\langle \boldsymbol{F} \rangle_{\boldsymbol{R}}$, gives the required canonical distribution of coarse-grained variables.[38] The subscript $\boldsymbol{R}$ denotes the canonical average over all fine-grained configurations $\boldsymbol{r}$ that map to a specific CG configuration $\boldsymbol{R} = (\boldsymbol{R}_1, \boldsymbol{R}_2, ..., \boldsymbol{R}_N)^{\mathrm{T}}$. Hence, one wants to learn the mapping between $\boldsymbol{R}$ and the *mean* force (MF), not the ICF. In force matching, the ensemble averaging is performed implicitly when solving the linear regression problem.[39,49,50] Alternatively, the mean force can be obtained explicitly by performing additional atomistic simulations with constrained CG coordinates.[30]

Here, we pursue a different approach, and average ICFs using covariant meshing, introduced in section 7. In fact, a representation which depends exclusively on the coarse-grained variables will automatically average the ICF into the MF. In practice, for the two-body kernel, we bin the center-of-mass distances. The resulting ML model should be identical to the two-body force matching (FM) results.[48,50]

Figure 7 illustrates the CG two-body force kernel predictions for 10^4 configurations with 1000 bins and a switching function, eq 24, with $d = 0.4$ Å. The use of switching is essential here: the MD simulations of ML models with and without switching are compared in the Supporting Information. We also show an average over 10 independent training sets, as it is often done in minibatch training in deep learning.[51] In all cases the binned kernel predictions are close to the FM results. Averaging over
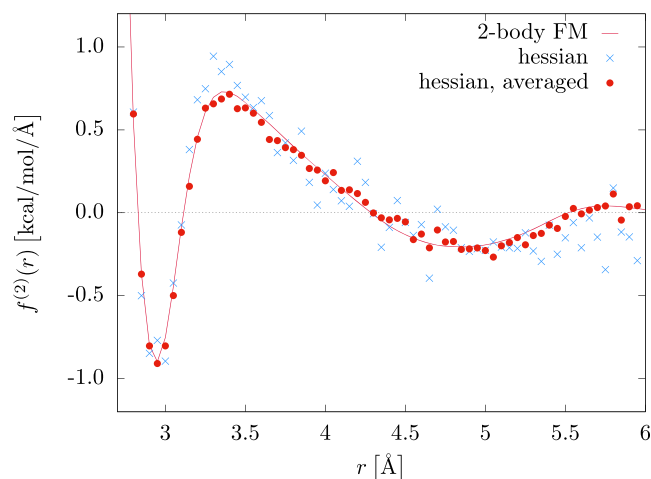

Figure 7. CG SPC/E water. Two-body force matching is compared to a kernel prediction based on 10^4 configurations with 1000 bins and an average over 10 independent kernel predictions. Optimized hyperparameter $\sigma = 0.5$ Å; output on a grid up to $r_{\mathrm{cut}} = 6$ Å.

the resulting output tables helps to reduce the error by smoothing the force curves, but is not critical for MD simulations, where stochastic integrators of thermostats compensate for small force fluctuations.

We now apply this averaging scheme to the three-body forces and train ML on the *residual* force between target and pairwise model:[48]

$$\Delta \boldsymbol{F}_i = \boldsymbol{F}_i - \boldsymbol{F}_i^{2\text{-body}} \tag{30}$$

where $\boldsymbol{F}_i$ is the ICF on CG particle i as given by eq 29 and $\boldsymbol{F}_i^{2\text{-body}}$ is the CG pairwise force.[48,50] We should make clear that $\boldsymbol{F}_i^{2\text{-body}}$ is the pairwise force from force matching as we want to compare learning the same residual forces with that in our previous FM parametrization. To learn these residual forces, we use the binned three-body Hessian kernel including all permutations and a switching function, eq 24, with $d = 0.4$ Å. We use 9900 bins and hyperparameters of $\sigma_1 = 0.7$ Å, $\sigma_2 = 0.7$ Å, and $\sigma_3 = 1.4$ Å. We then project the ML models onto tabulated force fields as described in section 8 with a total $N_{\mathrm{out}} = 27\,900$ table entries. Finally, we average over the 10 independent table predictions.

In Figure 8, we show the liquid water distribution functions for three different simulations: the original atomistic simulations, the force matching, and the Hessian kernel model. All schemes use a cutoff for the three-body forces of $r_{\mathrm{cut}} = 3.7$ Å and employ the same FM two-body forces with a cutoff of $r_{\mathrm{cut}} = 12$ Å.

The ML model is very close to the atomistic curve and reproduces the second peak of the RDF better than the FM parametrization of the SW potential. The angular distribution is also more accurate than the FM result. This stems from the additional flexibility of the fitting procedure: the FM fit imposes a fixed functional form on the radial dependence of the three-body force, while the ML model is fully flexible. The average pressure at the CG level is slightly too high at the atomistic density for both the ML and FM models, 4 and 3 kbar, respectively. We showed that one can correct for this by adding a small linear perturbation to the pair force without altering the structural properties.[48]
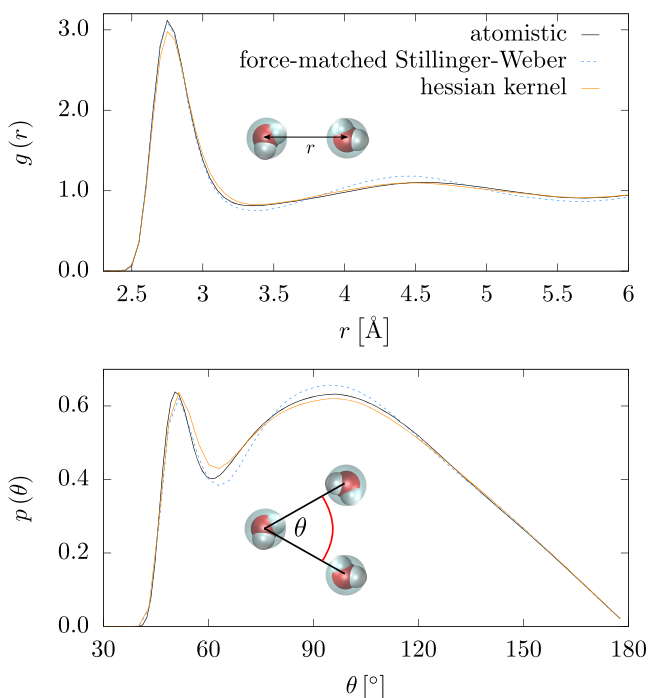
Figure 8. CG SPC/E water system. Comparison of MD simulations for the atomistic simulations, coarse-grained model using force matching to parametrize the SW model, and Hessian kernel averaged over 10 predictions with 9900 bins. Both pair, $g(r)$, and angular, $p(\theta)$, distribution functions are shown.

10. SOFTWARE

All simulations are performed with the LAMMPS package.[52] A custom implementation of the features presented here can be found online.[53] It includes the pair styles sw/table and 3b/table as user packages. The coarse-graining procedure relies on the open-source software package VOTCA,[50] also available online.[54]

11. CONCLUSIONS

We have described several practical ingredients of the kernel-based regression for classical simulations of liquids. Our approach is to first train a machine learning model by using body decomposition and then project this model onto tabulated potentials.

First, we explain in detail the general idea of the decomposition ansatz and use it to link the accessible particle-environment kernel with the local interaction kernels. We then concentrate on covariant and conservative kernels for pair and triplet interactions and compare three approaches: an analytic integration over the rotation group, the Hessian of a scalar kernel, and a kernel that makes predictions in a fixed axis system. The three kernels perform equally well on a two-body Lennard-Jones fluid, quickly converging to the precision of the reference data. Their limitations become evident when learning the three-body Stillinger—Weber (SW) fluid, where the lack of energy conservation is detrimental to the learning performance. Overall, we find that the best performing Hessian kernel is also the easiest to adapt and implement.

Several technical points help us to further improve the learning performance. The use of a switching function close to the interaction cutoff reduces noisy features and is a useful addition for coarse-grained interactions. A postaveraging

procedure over multiple ML models smoothens the resulting force fields. Covariant meshing of training configurations helps to increase the training set size. We present and implement these linear models in the general framework provided by the decomposition ansatz, which also offers the potential to build hybrid models that would learn only parts of the local interaction.

Last, the extension to coarse-grained liquids has shown to be the most challenging. By switching from the potential energy surface to the many-body potential of mean force, coarse-graining introduces degeneracy in the reference configurations that map to a single point at the coarse-grained level. This introduces noise in the learning procedure. Rather than relying on extra simulations with constrained coarse-grained degrees of freedom, we propose to use covariant meshing to average the instantaneous force into a mean force. Using this approach we recover accurate two- and three-body structural distribution functions of the atomistic liquid water. The covariant meshing in combination with tabulation is a promising technique to construct computationally cheap and flexible two- and three-body models for a range of CG applications.

All machine learning models and coarse-graining procedures are integrated in the VOTCA[54] and LAMMPS[53] packages which are publicly available.

■ ASSOCIATED CONTENT

ⓢ Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.9b01256.

> Derivation of the Hessian of the scalar kernel, explicit form of the conservative rotation kernel, details of two-body Lennard-Jones fluid and three-body Stillinger—Weber fluid, technical details for the cutoffs, covariant meshing, tabulated force field, and coarse-graining sections (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Denis Andrienko − *Max Planck Institute for Polymer Research, 55128 Mainz, Germany;* ⓞ orcid.org/0000-0002-1541-1377; Email: denis.andrienko@mpip-mainz.mpg.de

Tristan Bereau − *Max Planck Institute for Polymer Research, 55128 Mainz, Germany;* ⓞ orcid.org/0000-0001-9945-1271; Email: bereau@mpip-mainz.mpg.de

Authors

Christoph Scherer − *Max Planck Institute for Polymer Research, 55128 Mainz, Germany;* ⓞ orcid.org/0000-0003-2587-4567

René Scheid − *Max Planck Institute for Polymer Research, 55128 Mainz, Germany*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jctc.9b01256

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

Noether program of the Deutsche Forschungsgemeinschaft (DFG) and the long program Machine Learning for Physics and the Physics of Learning at the Institute for Pure and Applied Mathematics (IPAM).

■ REFERENCES

(1) Sumpter, B. G.; Noid, D. W. Potential energy surfaces for macromolecules. a neural network technique. *Chem. Phys. Lett.* **1992**, *192*, 455−462.

(2) Ho, T.-S.; Rabitz, H. A general method for constructing multidimensional molecular potential energy surfaces from ab initio calculations. *J. Chem. Phys.* **1996**, *104*, 2584−2597.

(3) Csányi, G.; Albaret, T.; Payne, M.; De Vita, A. Learn on the fly": a hybrid classical and quantum-mechanical molecular dynamics simulation. *Phys. Rev. Lett.* **2004**, *93*, 175503.

(4) Lorenz, S.; Groß, A.; Scheffler, M. Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks. *Chem. Phys. Lett.* **2004**, *395*, 210−215.

(5) Frenkel, D.; Smit, B. *Understanding Molecular Simulation: From Algorithms to Applications*; Elsevier: 2001; Vol. *1*.

(6) Bartók, A. P.; Csányi, G. Gaussian approximation potentials: A brief tutorial introduction. *Int. J. Quantum Chem.* **2015**, *115*, 1051−1057.

(7) Mones, L.; Bernstein, N.; Csányi, G. Exploration, Sampling, And Reconstruction of Free Energy Surfaces with Gaussian Process Regression. *J. Chem. Theory Comput.* **2016**, *12*, 5100−5110.

(8) Deringer, V. L.; Csányi, G. Machine learning based interatomic potential for amorphous carbon. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2017**, *95*, 094203.

(9) Deringer, V. L.; Pickard, C. J.; Csányi, G. Data-Driven Learning of Total and Local Energies in Elemental Boron. *Phys. Rev. Lett.* **2018**, *120*, 156001.

(10) Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R. Machine learning of accurate energy-conserving molecular force fields. *Science Advances* **2017**, *3*, e1603015.

(11) Nguyen, T. T.; Székely, E.; Imbalzano, G.; Behler, J.; Csányi, G.; Ceriotti, M.; Götz, A. W.; Paesani, F. Comparison of permutationally invariant polynomials, neural networks, and Gaussian approximation potentials in representing water interactions through many-body expansions. *J. Chem. Phys.* **2018**, *148*, 241725.

(12) Zhang, L.; Han, J.; Wang, H.; Car, R.; E, W. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.* **2018**, *120*, 143001.

(13) Behler, J.; Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **2007**, *98*, 146401.

(14) Bartók, A. P.; Kondor, R.; Csányi, G. On representing chemical environments. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2013**, *87*, 184115.

(15) Shapeev, A. V. Moment tensor potentials: A class of systematically improvable interatomic potentials. *Multiscale Model. Simul.* **2016**, *14*, 1153−1173.

(16) von Lilienfeld, O. A. Quantum machine learning in chemical compound space. *Angew. Chem., Int. Ed.* **2018**, *57*, 4164−4169.

(17) Chmiela, S.; Sauceda, H. E.; Müller, K.-R.; Tkatchenko, A. Towards exact molecular dynamics simulations with machine-learned force fields. *Nat. Commun.* **2018**, *9*, 3887.

(18) Bereau, T.; Andrienko, D.; von Lilienfeld, O. A. Transferable Atomic Multipole Machine Learning Models for Small Organic Molecules. *J. Chem. Theory Comput.* **2015**, *11*, 3225−3233.

(19) Grisafi, A.; Wilkins, D. M.; Csányi, G.; Ceriotti, M. Symmetry-Adapted Machine Learning for Tensorial Properties of Atomistic Systems. *Phys. Rev. Lett.* **2018**, *120*, 036002.

(20) Bereau, T.; DiStasio, R. A., Jr; Tkatchenko, A.; Von Lilienfeld, O. A. Non-covalent interactions across organic and biological subsets of chemical space: Physics-based potentials parametrized from machine learning. *J. Chem. Phys.* **2018**, *148*, 241706.

(21) Alvarez, M. A.; Rosasco, L.; Lawrence, N. D.; et al. Kernels for vector-valued functions: A review. *Found. Trends Mach. Learn.* **2012**, *4*, 195−266.

(22) Fuselier, E. J., Jr. Refined error estimates for matrix-valued radial basis functions. Ph.D. Thesis, Texas A&M University, 2007.

(23) Macêdo, I.; Castro, R. *Learning Divergence-Free and Curl-Free Vector Fields with Matrix-Valued Kernels*; IMPA: 2010.

(24) Mathias, S. A Kernel-based Learning Method for an Efficient Approximation of the High-Dimensional Born-Oppenheimer Potential Energy Hypersurface. M.Sc. Thesis, Rheinischen Friedrich-Wilhelms-Universität Bonn, 2015.

(25) Christensen, A. S.; Faber, F. A.; von Lilienfeld, O. A. Operators in quantum machine learning: Response properties in chemical space. *J. Chem. Phys.* **2019**, *150*, 064105.

(26) Glielmo, A.; Sollich, P.; De Vita, A. Accurate interatomic force fields via machine learning with covariant kernels. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2017**, *95*, 214302.

(27) Mehta, M. L. *Random Matrices*; Elsevier: 2004; Vol. *142*.

(28) Kondor, R. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials. *arXiv*, 2018, 1803.01588. https://arxiv.org/abs/1803.01588.

(29) Thomas, N.; Smidt, T.; Kearnes, S.; Yang, L.; Li, L.; Kohlhoff, K.; Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3D point clouds. *arXiv*, 2018, 1802.08219. https://arxiv.org/abs/1802.08219.

(30) John, S. T.; Csányi, G. Many-Body Coarse-Grained Interactions Using Gaussian Approximation Potentials. *J. Phys. Chem. B* **2017**, *121*, 10934−10949.

(31) Glielmo, A.; Zeni, C.; De Vita, A. Efficient nonparametric n-body force fields from machine learning. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2018**, *97*, 184307.

(32) Glielmo, A.; Zeni, C.; Fekete, A.; De Vita, A. Building nonparametric *n*-body force fields using Gaussian process regression. *arXiv (Physics)*, 2019, 1905.07626. https://arxiv.org/abs/1905.07626.

(33) Bartók-Pírtay, A. *The Gaussian Approximation Potential: An Interatomic Potential Derived from First Principles Quantum Mechanics*; Springer Science & Business Media: 2010.

(34) Voth, G. A. *Coarse-Graining of Condensed Phase and Biomolecular Systems*; CRC Press: 2008.

(35) Noid, W. G. Perspective: Coarse-grained models for biomolecular systems. *J. Chem. Phys.* **2013**, *139*, 090901.

(36) Peter, C.; Kremer, K. Multiscale simulation of soft matter systems. *Faraday Discuss.* **2010**, *144*, 9−24.

(37) Tschöp, W.; Kremer, K.; Batoulis, J.; Bürger, T.; Hahn, O. Simulation of polymer melts. I. Coarse-graining procedure for polycarbonates. *Acta Polym.* **1998**, *49*, 61−74.

(38) Noid, W. G.; Liu, P.; Wang, Y.; Chu, J.-W.; Ayton, G. S.; Izvekov, S.; Andersen, H. C.; Voth, G. A. The multiscale coarse-graining method. II. Numerical implementation for coarse-grained molecular models. *J. Chem. Phys.* **2008**, *128*, 244115.

(39) Izvekov, S.; Parrinello, M.; Burnham, C. J.; Voth, G. A. Effective force fields for condensed phase systems from ab initio molecular dynamics simulation: A new method for force-matching. *J. Chem. Phys.* **2004**, *120*, 10896.

(40) Tóth, G. Effective potentials from complex simulations: a potential-matching algorithm and remarks on coarse-grained potentials. *J. Phys.: Condens. Matter* **2007**, *19*, 335222.

(41) Zhang, L.; Han, J.; Wang, H.; Car, R.; E, W. DeePCG: Constructing coarse-grained models via deep neural networks. *J. Chem. Phys.* **2018**, *149*, 034101.

(42) Wang, J.; Olsson, S.; Wehmeyer, C.; Pérez, A.; Charron, N. E.; de Fabritiis, G.; Noé, F.; Clementi, C. Machine Learning of Coarse-Grained Molecular Dynamics Force Fields. *ACS Cent. Sci.* **2019**, *5*, 755−767.

(43) Zuo, Y.; Chen, C.; Li, X.; Deng, Z.; Chen, Y.; Behler, J.; Csányi, G.; Shapeev, A. V.; Thompson, A. P.; Wood, M. A.; Ong, S. P. Performance and Cost Assessment of Machine Learning Interatomic Potentials. *J. Phys. Chem. A* **2020**, *124*, 731−745.

(44) Berendsen, H. J. C.; Grigera, J. R.; Straatsma, T. P. The missing term in effective pair potentials. *J. Phys. Chem.* **1987**, *91*, 6269−6271.

(45) Molinero, V.; Moore, E. B. Water Modeled As an Intermediate Element between Carbon and Silicon [†]. *J. Phys. Chem. B* **2009**, *113*, 4008−4016.

(46) Larini, L.; Lu, L.; Voth, G. A. The multiscale coarse-graining method. VI. Implementation of three-body coarse-grained potentials. *J. Chem. Phys.* **2010**, *132*, 164107.

(47) Das, A.; Andersen, H. C. The multiscale coarse-graining method. IX. A general method for construction of three body coarse-grained force fields. *J. Chem. Phys.* **2012**, *136*, 194114.

(48) Scherer, C.; Andrienko, D. Understanding three-body contributions to coarse-grained force fields. *Phys. Chem. Chem. Phys.* **2018**, *20*, 22387−22394.

(49) Noid, W. G.; Chu, J.-W.; Ayton, G. S.; Krishna, V.; Izvekov, S.; Voth, G. A.; Das, A.; Andersen, H. C. The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *J. Chem. Phys.* **2008**, *128*, 244114.

(50) Rühle, V.; Junghans, C.; Lukyanov, A.; Kremer, K.; Andrienko, D. Versatile Object-Oriented Toolkit for Coarse-Graining Applications. *J. Chem. Theory Comput.* **2009**, *5*, 3211−3223.

(51) Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv*, 2015, 1502.03167. https://arxiv.org/abs/1502.03167.

(52) Plimpton, S. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.* **1995**, *117*, 1−19.

(53) *LAMMPS Custom Implementation*. https://gitlab.mpcdf.mpg.de/votca/lammps (accessed 2019-10-15).

(54) *VOTCA Repository*. https://gitlab.mpcdf.mpg.de/votca (accessed 2019-10-15).