Expanding Density-Correlation Machine Learning Representations for Anisotropic Coarse-Grained Particles

Arthur Lin,[1] Kevin K. Huguenin-Dumittan,[2] Yong-Cheol Cho,[1,3] Jigyasa Nigam,[2] and Rose K. Cersonsky[1]
[1)]*Department of Chemical and Biological Engineering, University of Wisconsin, Madison, WI, USA*
[2)]*Laboratory of Computational Science and Modeling, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland*
[3)]*Department of Computer Science and Engineering, University of Wisconsin, Madison, WI, USA*

(*Author to whom correspondence should be addressed: rose.cersonsky@wisc.edu)

(Dated: 24 June 2024)

Physics-based, atom-centered machine learning (ML) representations have been instrumental to the effective integration of ML within the atomistic simulation community. Many of these representations build off the idea of atoms as having spherical, or isotropic, interactions. In many communities, there is often a need to represent groups of atoms, either to increase the computational efficiency of simulation via coarse-graining or to understand molecular influences on system behavior. In such cases, atom-centered representations will have limited utility, as groups of atoms may not be well-approximated as spheres. In this work, we extend the popular Smooth Overlap of Atomic Positions (SOAP) ML representation for systems consisting of non-spherical anisotropic particles or clusters of atoms. We show the power of this anisotropic extension of SOAP, which we deem AniSOAP , in accurately characterizing liquid crystal systems and predicting the energetics of Gay-Berne ellipsoids and coarse-grained benzene crystals. With our study of these prototypical anisotropic systems, we derive fundamental insights on how molecular shape influences mesoscale behavior and explain how to reincorporate important atom-atom interactions typically not captured by coarse-grained models. Moving forward, we propose AniSOAP as a flexible, unified framework for coarse-graining in complex, multiscale simulation.

I. INTRODUCTION

In understanding molecular interactions and modeling their resultant behavior, it is very often a worthwhile endeavor to group (i.e. coarse-grain) one or more atoms as a theoretical "unit" or particle. This choice can be practical; often, the time- and length-scales necessary to simulate molecular processes limit our ability to simulate with all-atom resolution. Conversely, choosing particle-based, rather than atom-based computational approaches can also be a scientific choice; when we selectively limit the degrees of freedom within our systems, we can identify factors that are causal to molecular behavior or phenomena. Thus, these simplified coarse-grained simulation approaches serve as both a tool and a lens with which to study chemical systems.

Similarly, machine learning (ML) methods have emerged as a powerful tool for scientific inquiry, with the ability to elucidate new patterns within or relationships between chemical spaces and observed properties, often in order to predict the properties of unseen systems. So, how do we incorporate the idea of atom grouping in the context of machine learning? Many approaches consider our configurational space as a manifold and use a variety of machine learning architectures (variational[1,2] and hierarchical[3] auto-encoders, different forms of neural networks[4–8], ensemble learning[9]) to determine a latent space in which this grouping is implicitly embedded that minimizes a chosen fitness function. Despite high performance and generalizability, these end-to-end models are often limited by data requirements or in fundamental analyses by their lack of intrinsic interpretability.

Another approach within the machine-learning community is so-called *feature-forward* modeling, wherein one explicitly transforms the raw chemical data into numerical "features" that reflect the underlying physics or chemistry of interest prior to applying ML methods. While both end-to-end and feature-forward approaches have merits and overlaps, the latter method is often noted for its interpretability and comparable performance using shallower ML architectures[10–12]. Within this umbrella of approaches, there are many ways to encode the raw chemical data into features, and the suitable choice depends entirely on the scientific context. For cheminformatics, wherein we are often looking to compare different chemistries or identify the role of specific functional groups, string-based featurizations such as SMILES[13] or SELFIES[14] are popular, as they encode important parameters such as present functional groups and connectivity and can be parsed using natural language processing (NLP) models and other text-based technologies. However, in thermodynamic contexts where the chemistry and connectivity remain largely unchanged, such as in molecular simulation, it is more typical to use configuration-dependent features[12,15], including Behler-Parinello symmetry functions[16], smooth overlap of atomic positions (SOAP)[17], and molecular graphs[18].

So, for coarse-grained systems, how do we apply a feature-forward approach? A large challenge is that groups of atoms, hereon referred to as "particles", are non-isotropic, and so methods based on atomistic ML will fail to capture the anisotropy of interparticle interactions. Frameworks based on density expansions (e.g., SOAP[17], NICE[19]) present a compelling avenue for expansion, given that they can putatively be made flexible to *any* density expansion, even anisotropic density fields or hard particle volumes. Furthermore, by ex-

tending these frameworks to isolate molecule-level interactions within atom-atom site potentials, we gain the ability to combine or compare representations across multiple scales.

Here, we propose and demonstrate the first such anisotropic expansion of symmetrized density-based frameworks for ML representations by taking the popular SOAP (Smooth Overlap of Atomic Positions) formalism and demonstrating its expansion to simple anisotropic bodies, in a representation we deem AniSOAP , which can be read as either "The Smooth Overlap of Anisotropic Particles" or "Anisotropic-SOAP". While here we demonstrate the expansion for multivariate Gaussian densities, similar expansions can be made for arbitrary anisotropic density fields.

In Section II, we provide the underlying mathematical theory behind this expansion, including how deliberate selection of basis sets enables analytical evaluation of the expansion coefficients. In Section III, we demonstrate three performance-defining case studies for its usage across multiple simulation length scales and materials systems. We start by analyzing two classic cases of explicitly ellipsoidal particles: liquid crystalline configurations and those governed by the Gay-Berne interaction potential. In doing so, we show the conceptual overlap and divergence of AniSOAP from traditional mesoscale order parameters, as well as its ability to generalize to supervised tasks. Then, we analyze a set of systems that is only *implicitly* ellipsoidal – benzene molecules arranged in stable and unstable crystalline configurations, and show that AniSOAP can be used to provide molecule-level approximations to first-principles energetics. We then couple AniSOAP with the traditional SOAP formalism to demonstrate how to combine representations across multiple scales. The corresponding open-source code, AniSOAP , is available at `github.com/cersonsky-lab/anisoap`.

General Variables	
$\boldsymbol{r}$	a vector in Cartesian space
r	magnitude of vector $\boldsymbol{r}$
$\hat{r}$	direction of vector $\boldsymbol{r}$
$\boldsymbol{r}_{ij}$	the vector between point i and point j
$\mathscr{R}$	a 3x3 rotation matrix
$\boldsymbol{x}$	a feature vector for one configuration
$\boldsymbol{X}$	a matrix containing, as rows, feature vectors for multiple configurations
Density Expansions	
$g(\boldsymbol{r})$	a potentially anisotropic density function centered at $\boldsymbol{r}$
ρ_{ij}	the contribution of the density located at the position of point j to the density expansion of point i
n,l,m	indices for radial bases and spherical harmonics
R_{nl}	a basis function indexed by n and l
σ_{GTO}	the width of a Gaussian-type orbital (GTO) basis
Y_l^m	the m^{th} component spherical harmonic of order l
$\mathscr{D}^l_{mm'}$	the Wigner matrix used to rotate spherical harmonics
Multivariate Gaussians (MVG)	
σ_1, σ_2, σ_3	the three principal components of an MVG
$\boldsymbol{D}$	the principal axis decomposition of an MVG, where $\boldsymbol{D} \equiv \text{diag}\left(\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \frac{1}{\sigma_3^2}\right)$
$\mathbf{A}$	the rotated principal axis decomposition of an MVG, where $\mathbf{A} \equiv \mathscr{R}\boldsymbol{D}\mathscr{R}^T$
$\boldsymbol{S}$	the Gay-Berne analog to $\boldsymbol{D}$, where $\boldsymbol{S} \equiv \boldsymbol{D}^{-2}$

TABLE I. **Notation Guide.** Throughout the text, we adopt the notation typical of the atom-centered symmetrized representation community as detailed in the table above.

II. THEORY

In the traditional formalism for SOAP and related representations, we treat each atom as a localized isotropic field in three dimensions and construct an "atom-density" by summing over all neighboring atoms within a spherical shell. The atom-density is usually written on a basis of radial and angular functions, the latter of which are chosen to be spherical harmonics to make the atom-density more easily amenable to rotational symmetrization.. To represent the many-body nature of these atoms, we can then compute n-body correlations of these density expansions or introduce message-passing into the featurization formalism[20,21].

A. The Effect of Symmetry Breaking in Density Frameworks

Using the braket notation prescribed in Ref. 22, the contribution of an atom j to center atom i's expansion coefficients are given by

$$\langle nlm|\rho_{ij}\rangle = \int_{\mathbb{R}^3} g(\boldsymbol{r}-\boldsymbol{r}_{ij})R_{nl}(r)Y_l^m(\hat{r})d^3\boldsymbol{r}. \qquad (1)$$

where $g(\boldsymbol{r})$ is a localized function (usually a Gaussian), $R_{nl}(r)$ is a radial basis function, and $Y_l^m(\hat{r})$ are spherical harmonics. For a general $g(\boldsymbol{r})$ and $R_{nl}(r)$, there is no hope to evaluate this analytically. There is not even a general way to evaluate a general one-dimensional integral $\int f(x)\mathrm{d}x$, which is why, after all, there are so many books on integral tables.

If we use any arbitrary density $g(\boldsymbol{r})$, e.g. to model more closely the shape of rigid molecules, nanoparticles, or arbitrary bodies, we are forced to use numerical integration, which could be done using Lebedev grids. A fully numerical implementation would also provide us with complete freedom regarding the choice of the radial basis function R_{nl}, which would allow us to choose the basis based on nice mathematical properties like the Laplacian eigenstate (LE) basis[23]. The main downside of the numerical approach, at least for sufficiently general densities $g(\boldsymbol{r})$, is the inevitable and potentially severe numerical cost-accuracy trade-off. The necessary inaccuracies introduced by numerical integration may negate the fidelity of the anisotropic density field, thus, we would therefore like to examine the possibility of a fully analytical approach for the evaluation of these coefficients.

For isotropic representations, it is traditional to do an implicit transformation to align $\boldsymbol{r}_{ij}$ onto the $z-axis$ of our coordinate system, eliminating any dependence of $g(\boldsymbol{r}-\boldsymbol{r}_{ij})$ on the angular integrands. This simplifies (1), even for complicated bases, to

$$\langle nlm|\rho_{ij}\rangle = F(r_{ij})Y_l^m(\hat{r}_{ij}), \qquad (2)$$

where $F(r_{ij})$ is some function that contains the complete dependence on the distance between the two atoms. The angular dependence is fully captured in the spherical harmonic factor, allowing us to precompute $F(\boldsymbol{r}_{ij})$ on the full relevant interval for efficient spline evaluations[22].

For non-isotropic densities, however, this is no longer the case. Even taking the relatively simple case of a multivariate Gaussian, we cannot decouple our radial (e.g., distance) and angular components, as the Gaussian requires a transformation into the three principal components $\sigma_1, \sigma_2, \sigma_3$ and corresponding axes. In order to maintain consistency with (1), we replace all instances of $\boldsymbol{r}$ with $\mathscr{R}\mathscr{R}^T\boldsymbol{r}$, where $\mathscr{R}$ is our rotation matrix into this component axis. We then simplify by changing our axes of integration from $\boldsymbol{r}$ to $\boldsymbol{r}' \equiv \mathscr{R}^T\boldsymbol{r}$ and noting that $r = \mathscr{R}\mathscr{R}^T r$ to obtain

$$\langle nlm|\rho_{ij}\rangle = \int_{\mathbb{R}^3} \mathrm{d}^3(\boldsymbol{r}')\; g(\mathscr{R}\boldsymbol{r}' - \mathscr{R}\boldsymbol{r}'_{ij}) R_{nl}(r) Y_l^m(\mathscr{R}\hat{r}'). \quad (3)$$

To perform the rotation of our spherical harmonics, one commonly sums over Wigner-D matrices[24] to obtain

$$\langle nlm|\rho_{ij}\rangle = \quad (4)$$
$$\sum_{m'=-l}^{l} \mathscr{D}^l_{mm'}(\mathscr{R}) \int_{\mathbb{R}^3} \mathrm{d}^3(\boldsymbol{r}')\; g(\mathscr{R}\left(\boldsymbol{r}' - \boldsymbol{r}'_{ij}\right)) R_{nl}(r') Y_l^{m'}(\hat{r}')$$

With this step, we can reduce our calculation in the special case in which the matrix $\mathscr{D}^l_{mm'}(\mathscr{R})$ is diagonal, which proves useful for analytical evaluation techniques. We thus focus on the factor

$$\int_{\mathbb{R}^3} \mathrm{d}^3\boldsymbol{r}'\; g(\mathscr{R}\left(\boldsymbol{r}' - \boldsymbol{r}'_{ij}\right)) R_{nl}(r') Y_l^{m'}(\hat{r}'). \quad (5)$$

which will be the focus of our discussion from hereon.

B. Multivariate Gaussian Densities

The simplest anisotropic function that can be put in place of $g(r)$ in (5) is the multivariate Gaussian density (hereon MVG). The goal of this subsection is to show that by choosing $g(\boldsymbol{r})$ to be an MVG and $R_{nl}(r)$ to be of the monomial, GTO, or STO form, we can reduce the evaluation of the expansion coefficient to the evaluation of integrals of the form

$$\langle nlm|\rho_{ij}\rangle = \int_{\mathbb{R}^3} e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{r}_{ij})^T \boldsymbol{D}(\boldsymbol{r}-\boldsymbol{r}_{ij})} \mathrm{p}(\boldsymbol{r}), \quad (6)$$

where $\mathrm{p}(\boldsymbol{r})$ is some polynomial expression in $\boldsymbol{r} \equiv (x,y,z)$ and $\boldsymbol{D}$ is some 3×3 diagonal matrix.

Consider the three-dimensional Gaussian defined by

$$g : \mathbb{R}^3 \to \mathbb{R} \quad (7)$$
$$\boldsymbol{r} \to g(\boldsymbol{r}) = \exp\left(-\frac{1}{2}\boldsymbol{r}^T\mathbf{A}\boldsymbol{r}\right), \quad (8)$$

where A is a symmetric and (strictly) positive definite 3×3 matrix. Any matrix satisfying these two conditions can be orthogonally diagonalized, also called the principal axis decomposition, allowing us to write

$$\mathbf{A} = \mathscr{R}\boldsymbol{D}\mathscr{R}^T, \quad (9)$$

where $\mathscr{R} \in \mathrm{SO}(3)$ is a rotation matrix that specifies the three principal axes and

$$\boldsymbol{D} \equiv \mathrm{diag}\left(\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \frac{1}{\sigma_3^2}\right) = \begin{pmatrix} \frac{1}{\sigma_1^2} & & \\ & \frac{1}{\sigma_2^2} & \\ & & \frac{1}{\sigma_3^2} \end{pmatrix} \quad (10)$$

is a diagonal matrix containing the widths σ_j of the Gaussian along the three principal directions. With this decomposition, we can write

$$g(\mathscr{R}\boldsymbol{r}') = \exp\left(-\frac{1}{2}\boldsymbol{r}'^T\mathscr{R}^T\mathscr{R}D\mathscr{R}^T\mathscr{R}\boldsymbol{r}'\right) \quad (11)$$
$$=: \exp\left(-\frac{1}{2}\boldsymbol{r}'^T\boldsymbol{D}\boldsymbol{r}'\right), \quad (12)$$

where $\boldsymbol{r}' = \mathscr{R}^T\boldsymbol{r}$ are the coordinates with respect to the principal axes.

Putting the equation for our MVG into (5), we get

$$\int_{\mathbb{R}^3} \mathrm{d}^3\boldsymbol{r}' \exp\left(-\frac{1}{2}(\boldsymbol{r}' - \boldsymbol{r}'_{ij})^T\boldsymbol{D}(\boldsymbol{r}' - \boldsymbol{r}'_{ij})\right) R_{nl}(r') Y_l^{m\prime}(\hat{r}'). \quad (13)$$

We have now set the stage for the evaluation of the general coefficient. Our goal is to choose a suitable radial basis that would allow us to evaluate the coefficients analytically. One possibility is to use certain polynomial bases, which is motivated by the fact that for any $n = 0,1,2,\ldots$, we have

$$\int_{\mathbb{R}} \mathrm{d}x x^n e^{-\frac{x^2}{2\sigma^2}} = \begin{cases} \left(2\sigma^2\right)^{\frac{n+1}{2}} \Gamma\left(\frac{n+1}{2}\right) & n \quad \text{even} \\ 0 & n \quad \text{odd} \end{cases} \quad (14)$$

where Γ is the gamma function. If we could reduce the evaluation of the integral (6) to such (multivariate) polynomials, this might provide us with analytical expressions for the expansion coefficients. Thus, following (13), we require that $R_{nl}(r)Y_l^m(\hat{r})$ is a polynomial.

Integrability of Monomial Basis We will first show that when $R_{nl}(r) = r^{l+2n}$, the expansion coefficients can be evaluated analytically. First, we separate $R_{nl}(r)$ into two factors, where

$$r^{l+2n} = (r^2)^n \cdot r^l \quad (15)$$

Given that $r^l Y_l^m(\hat{r})$ is a polynomial in (x,y,z), as is $r^2 = x^2+y^2+z^2$ and thus, so is $(r^2)^n$, the monomial basis can be analytically integrated.

Integrability of GTO Basis With one extra step, we can also show that a suitable modification of the GTO basis, $R_{nl}(r) = r^{l+2n}e^{-\frac{r^2}{2\sigma^2}}$, provides an equally well-suited basis. .

By separating out all the exponential and polynomial factors, we can write the integrand as

$$\begin{aligned} & e^{-\frac{1}{2}(\boldsymbol{r}'-\boldsymbol{r}'_{ij})^T D(\boldsymbol{r}'-\boldsymbol{r}'_{ij})} R_{nl}(r') Y_l^m(\hat{\boldsymbol{r}'}) \\ =& e^{-\frac{1}{2}(\boldsymbol{r}'-\boldsymbol{r}'_{ij})^T D(\boldsymbol{r}'-\boldsymbol{r}'_{ij})} \boldsymbol{r}^{l+2n} e^{-\frac{r^2}{2\sigma^2}} Y_l^m(\hat{r}) \\ =& e^{-\frac{1}{2}(\boldsymbol{r}'-\boldsymbol{r}'_{ij})^T D(\boldsymbol{r}'-\boldsymbol{r}'_{ij}) - \frac{r^2}{2\sigma^2}} r^{l+2n} Y_l^m(\hat{r}) \\ =& e^{-\frac{1}{2}(\boldsymbol{r}'-\boldsymbol{r}'_{ij})^T D(\boldsymbol{r}'-\boldsymbol{r}'_{ij}) - \frac{r^2}{2\sigma^2}} \mathrm{p}(x,y,z) \\ =& e^{-\mathrm{quadratic}(\boldsymbol{r})} \mathrm{p}(x,y,z) \end{aligned} \tag{16}$$

We can see that this is again an exponential of a (now different) quadratic form multiplied by a polynomial in (x,y,z), which can be integrated analytically.

Concluding this section, we can see that for both choices of the basis, the final expression we need to evaluate is of the form

$$\int_{\mathbb{R}^3} \mathrm{d}^3\boldsymbol{r} e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{r}_{ij})^T \boldsymbol{D}(\boldsymbol{r}-\boldsymbol{r}_{ij})} r^{2n} R_l^m(\boldsymbol{r}), \tag{17}$$

with $R_l^m(\boldsymbol{r}) = r^l Y_l^m(\hat{r})$ and where the diagonal matrix $\boldsymbol{D}$ is either the one directly obtained from the principal components of the Gaussian density (monomial basis), or the modified version (GTO basis). Similar arguments can be made for Slater-type orbitals (STO), which follow a similar form to the GTO basis. From this expansion, we can then calculate n-body correlations for a given particle, which for the 3-body term is

$$\langle nl; n'l'|\rho_i^{\nu=2}\rangle = \sum_m \langle nlm|\rho_i\rangle \langle n'l'm|\rho_i\rangle \tag{18}$$

where $\langle nlm|\rho_i\rangle = \sum_j \langle nlm|\rho_{ij}\rangle$. We go through the details on analytically computing eq. 17 in Appendix A 2-A 7. We first detail how to convert $r^{2n}R_l^m$ into a cartesian polynomial (A 2, A 5), then detail how to evaluate the integrals (i.e. moments of the MVG) in A 3 and A 6. A 4 gives an overview of how these steps are combined together to result in evaluating a weighted sum of high-order moments of the MVG. Finally, we discuss imposing orthonormality within the basis set in Appendix A 7 via Löwdin symmetric orthogonalization[25].

III. RESULTS AND DISCUSSION

Even for general point clouds, determining the optimal anisotropic analog is non-trivial, and the focus of ongoing work[26]. Thus, we choose to demonstrate the efficacy of AniSOAP on systems where the choice of an ellipsoidal proxy is trivial, being both historically founded and well-defined. We ground our expansion in the rich history of identifying causal mechanisms through anisotropic proxy particles[27], as many complex behaviors within molecular systems can be explained by analyzing the steric interactions of their molecular volumes[28–32]. Future studies will focus on optimizing AniSOAP for less trivial cases and further varieties of molecular anisotropy.

A. Ellipsoids in Liquid Crystals

In Sections III A-III B, we start by analyzing two archetypal ellipsoidal systems: particles in different liquid crystalline (LC) phases, and particles governed by the classical Gay-Berne potential. These case studies demonstrate the similarities and differences between AniSOAP and traditional LC order parameters, namely the orientational (or nematic) order parameter and pair-wise Steinhardt order parameters, and examine the ability of AniSOAP to perform continuous supervised tasks. In Section III C, we then move to the opposite end of the molecular spectrum and analyze benzene crystals – molecules that are ellipsoidal in shape but whose atomic interactions are complex and fundamental to their interaction landscape. Of the systems that ellipsoidal AniSOAP is already well-suited to describe, these cases provide a relative upper- and lower-bound to model performance from a system where interactions are, by definition, entirely defined by the ellipsoidal correlations to those where the energetics are minimally determined by the shape of the molecule alone.

Unless otherwise specified, when discussing $X_{AniSOAP}$, we are referring to the power spectrum (the *3-body* term) given in Eq. 18, but note the simplicity of continuing onto higher body-order terms for greater accuracy and transferability[19].

Liquid crystals (LC) are mesoscopic phases that form from rigid ellipsoid-like particles or molecules and can be controlled to direct the flow of light for a variety of applications.[33] Typically, LCs are characterized as phases with orientational order but limited positional or translational order. Molecules with orientational order but no translational order are generally considered "nematic", whereas those exhibiting one direction of translational order (e.g. the particles appear as stacking planes, where within the planes, there is little translational order) are deemed *smectic*.

In this analysis, we aim to show how our new AniSOAP representation is similar to previously established descriptors and how it enables functionality and analyses beyond the current capabilities. In characterizing LC phases, scientists have relied on a library of different order parameters, including different orientational order parameters[34] to classify the orientational alignment of particles, Steinhardt order parameters (SOPs[35]) to characterize the neighborhood of different particles. We note the similarity of SOPs to Eq. (2), wherein an SOP for a given particle involves the integration of spherical harmonics over that particle's neighbors:

$$\langle lm|q_i\rangle = \frac{1}{N_j}\sum_j w_{ij} Y_l^m(r_{ij}) \tag{19}$$

where j is a neighbor of i, and weights w_{ij} can be introduced based on neighbor distances or Voronoi tesselations[36]. The l^{th} SOP is computed by combining these terms such that

$$X_{SOP} = \langle l|q_i\rangle = \sqrt{\frac{4\pi}{2l+1}\sum_{m=-l}^{l} |\langle lm|q_i\rangle|^2}. \tag{20}$$

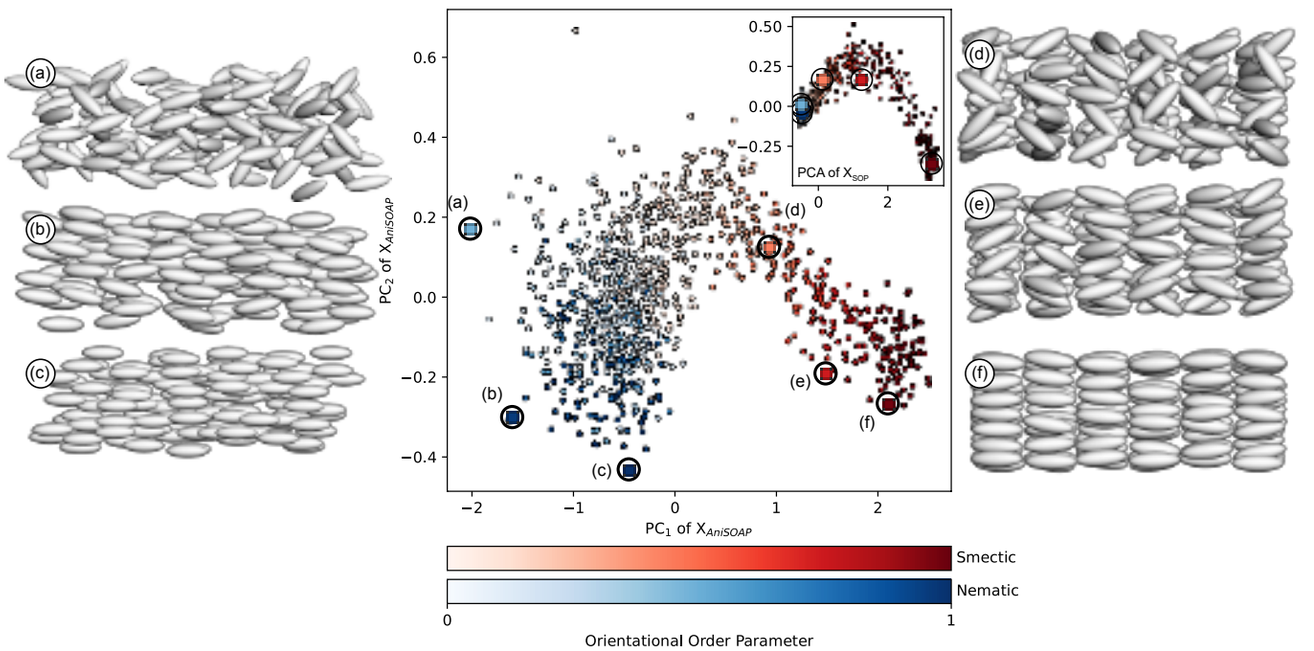


FIG. 1. **The feature space of AniSOAP in the context of LC phases.** The middle plot shows the first two components of the structure-average AniSOAP vectors for 2000 generated liquid crystal configurations showing either no translational order (nematic phases, blue), planar stacking (smectic phases, red), and varying degrees of orientation order (color saturation, with white being little to no order). The inset shows the first two principal components of the popular Steinhardt Order parameters, which clearly delineate the smectic (red) phases, but in these and all following components, do not separate the different nematic (blue) phases. Representative snapshots are shown to the left and right corresponding to labeled points on the plots.

Thus, for comparison, for all configurations, we also compute the orientational order parameter $\boldsymbol{X}_{OOP}$ (traditionally called the nematic order parameter) and distance-weighted Steinhardt order parameters $\boldsymbol{X}_{SOP}$ with `freud`[34], using similar $l_{\max}$ and cutoff radius to ensure comparability. All representations are properly scaled and centered – that is, each matrix of feature vectors is centered to have zero mean with unit variance, where $\boldsymbol{X}_{SOP}$ and $\boldsymbol{X}_{AniSOAP}$ are scaled non-column-wise in order to retain important relative variance information.

We generate 1000 liquid crystal configurations, 500 exhibiting nematic order, and 500 exhibiting smectic order. Phases were generated to correspond with a range of orientational order parameters, from 0 (particles are randomly oriented) to 1 (particles are all ordered along the same director). We populate each of these configurations with prolate ellipsoids with a length-to-diameter ratio (L/D) equal to 3. For each of these configurations, we compute the AniSOAP radial and power spectrum $\sigma_{GTO} = 3.0$, $n_{\max} = 6$, $l_{\max} = 6$ (for the power spectrum), and a cutoff sufficient to include the surrounding neighbors of each ellipsoid, including in neighboring smectic planes.

We first look at the principal components (PCs) of the AniSOAP vectors for these phases (Fig. 1). The PCA reflects that the AniSOAP features can be used to delineate translational order (blue versus red coloring), and orientational order (saturated versus unsaturated color). These mappings highlight qualitatively the information contained in the AniSOAP representation and show that it simultaneously and smoothly represents the translational and orientational order of the configurations. In Fig. 1 of the supplementary material, we demonstrate that the AniSOAP features also smoothly delineate different particle shapes, with a similar plot provided with data for both L/D=2 and 3 ellipsoids. By combining this information into one smooth feature space, we are able to better recreate the "nearsightedness" of molecular interactions[37], as these aspects (shape, mutual orientation, and translational order) are often interrelated and correlated in the ways in which they influence molecular behavior.

Quantitatively, we can compare the information density of AniSOAP with other order parameters for our configurations using the global feature reconstruction error (GFRE[38], computed using `scikit-matter`[39]). For two representations $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ of the same dataset, $\mathrm{GFRE}(X_1, X_2)$, determines how much information $\boldsymbol{X}_1$ contains relative to $\boldsymbol{X}_2$, where 0.0 infers $\boldsymbol{X}_1$ can perfectly reconstruct $\boldsymbol{X}_2$, and 1.0 corresponds to poor reconstruction. As shown in Fig. 2, we see that AniSOAP representations (both the 2-body radial spectrum $X^{(\nu=1)}_{AniSOAP}$ and 3-body power spectrum $X^{(\nu=2)}_{AniSOAP}$) are able to re-

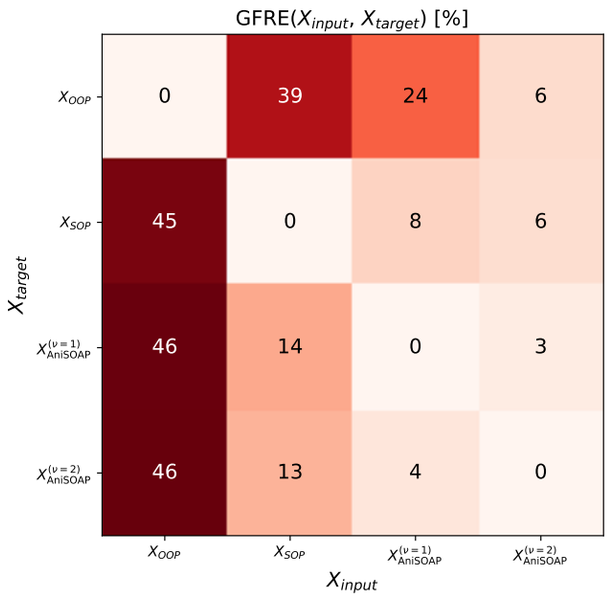


FIG. 2. **Comparison of Different Featurizations for Liquid Crystal Configurations.** The Global Feature Reconstruction Error (GFRE) encodes the error using one feature representation $\boldsymbol{X}_{\text{input}}$ to reconstruct another $\boldsymbol{X}_{\text{target}}$, and can be used to infer the information density of one featurization compared to another. Higher values indicate poor reconstruction, whereas lower indicates better reconstruction. Here we see that AniSOAP features better reconstruct traditional Steinhardt order parameters (q_{0-6}) and orientational order parameters than vice versa. While Steinhardt OPs carry large mutual information with the AniSOAP vectors, they are unable to reconstruct the Orientational Order Parameter.

construct the analogous $\boldsymbol{X}_{SOP}$ with lower error than vice versa (6-8% versus 12-14% reconstruction error). Furthermore, the AniSOAP representations are able to reconstruct the orientational order parameter with low error (6% for the 3-body, and 24% for the 2-body, compared to 39% for X_{SOP}). The improvement in information density can be attributed to the smooth radial bases that underlie the AniSOAP construction. It is worth noting that the Steinhardt order parameter is not well-suited for heterogenous datasets, as shown in Fig. 2 of the supplementary material, and its ability to reconstruct the AniSOAP representation will decrease when a dataset contains multiple particle types.

B. Gay-Berne Ellipsoids

The Gay-Berne (GB) potential is a Lennard Jones-type potential that contains additional terms to account for ellipsoidal anisotropies. We use the generalized formulation of the Gay-Berne potential outlined by Everaers and Ejtehadi [40] and first introduced by Berardi, Fava, and Zannoni [41]. This formulation calculates the pairwise potential between (potentially dissimilar) ellipsoids, i.e., ellipsoids with three unequal semi-axes: a_i, b_i, c_i. These three semi-axes define a diagonal structure matrix for particle i:

$$\boldsymbol{S}_i = \begin{pmatrix} a_i & 0 & 0 \\ 0 & b_i & 0 \\ 0 & 0 & c_i \end{pmatrix} \tag{21}$$

where $\boldsymbol{S}$ is analogous to $\boldsymbol{D}^{-2}$ of Eq. (12), although (a,b,c) are discrete semiaxes, rather than Gaussian widths, like σ. In our case-study, we generate 25,000 dimers of $a_i = 1, b_i = 1.5, c_i = 2$ ellipsoids at random offsets and orientations.

The center position and orientation of ellipsoid i is given by position vector $\boldsymbol{r}_i$ and a 3×3 rotation matrix $\mathscr{R}_i$, respectively. The relative position is $\boldsymbol{r}_{12} = \boldsymbol{r}_2 - \boldsymbol{r}_1$. The Gay-Berne potential takes into account interparticle distance dependence and orientation dependence of the ellipsoid pairs. The potential is given as a product of three terms:

$$U(\mathscr{R}_1, \boldsymbol{S}_1, \mathscr{R}_2, \boldsymbol{S}_2, \vec{r_{12}}) = \underbrace{U_r(\mathscr{R}_1, \boldsymbol{S}_1, \mathscr{R}_2, \boldsymbol{S}_2, \vec{r_{12}})}_{\text{LJ-like term}} \tag{22}$$

$$\cdot \underbrace{\eta_{12}(\mathscr{R}_1, \boldsymbol{S}_1, \mathscr{R}_2, \boldsymbol{S}_2) \cdot \chi_{12}(\mathscr{R}_1, \mathscr{R}_2, \hat{r_{12}})}_{\text{anisotropy corrections}} \tag{23}$$

The first term U_r is given as follows:

$$U_r = 4\varepsilon_{GB}\left[\left(\frac{\sigma_{GB}}{h_{12} + \gamma\sigma_{GB}}\right)^{12} - \left(\frac{\sigma_{GB}}{h_{12} + \gamma\sigma_{GB}}\right)^{6}\right] \tag{24}$$

where the distance h_{12} between two ellipsoids is estimated as the Perram distance of closest approach[42]

$$h_{12}(\mathscr{R}_1, \mathscr{R}_2, \boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{r}_{12}) = r_{12} - s_{12}(\mathscr{R}_1, \mathscr{R}_2, \boldsymbol{S}_1, \boldsymbol{S}_2, \hat{r}_{12}) \tag{25}$$

where

$$s_{12}(\mathscr{R}_1, \mathscr{R}_2, \boldsymbol{S}_1, \boldsymbol{S}_2, \hat{r}_{12}) = \left[\frac{1}{2}\hat{r}_{12}^T \boldsymbol{\Xi}_{12}^{-1}(\mathscr{R}_1, \mathscr{R}_2, \boldsymbol{S}_1, \boldsymbol{S}_2)\hat{r}_{12}\right]^{-1/2} \tag{26}$$

and

$$\boldsymbol{\Xi}_{12}(\mathscr{R}_1, \mathscr{R}_2, \boldsymbol{S}_1, \boldsymbol{S}_2) = \mathscr{R}_1^T \boldsymbol{S}_1^2 \mathscr{R}_1 + \mathscr{R}_2^T \boldsymbol{S}_2^2 \mathscr{R}_2 \tag{27}$$

The terms ε_{GB} and σ_{GB} are the potential well depth and location analogous to LJ. γ is a parameter that shifts the potential well and is typically set to 1. Since the interparticle distance is defined to be the "distance of closest approach", h_{12}, rather than center-center distance, so U_r must also contain dependence on the geometry and orientation of each ellipsoid to calculate h_{12}. The remaining pieces of Eq. (23) account for the ellipsoidal geometry (η_{12}) and misorientation (χ_{12}); further explanation of these terms are given in Sec. B 1.

One of the less trivial aspects of the Gay-Berne implementation is calculating the "distance of closest approach" h_{12}. The approximation given by Perram and Wertheim [42] is computationally efficient, but open to failure in some cases, like in the case of two bodies with very unequal radii[40]. It is worth noting that it is the nature of density expansions to implicitly contain the distance of closest approach, provided that the

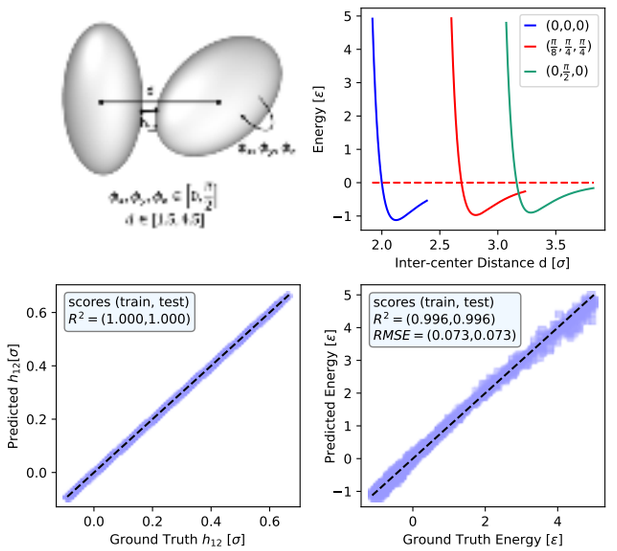


FIG. 3. **Predictions of ellipsoid interactions based on the Gay-Berne Potential**. The ellipsoid dimer dataset consists of two ellipsoids with varying inter-center distance d and orientations based on rotations along the x (ϕ_x), y (ϕ_y), and z (ϕ_z) axes. These varying orientations and distances lead to different characteristic energy wells, shown in the top right. Parity plots detail the performance in predicting the distance of closest approach h_{12} (bottom-left, units of characteristic length scale σ) and energies (bottom-right, units of potential well depth ε). For clarity, only the test points are shown; the train points show very similar trends.

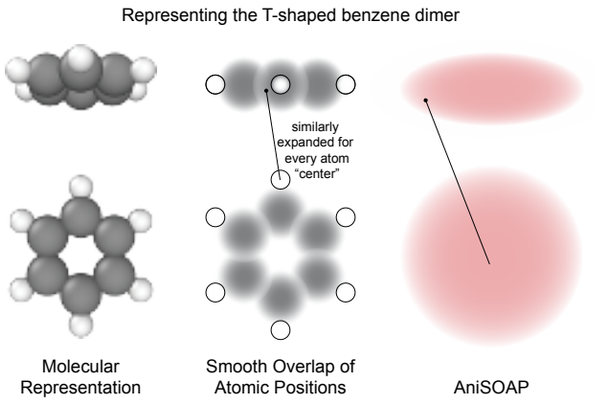


FIG. 4. Representations of the T-Shaped Benzene Dimer. From left to right. A molecular image of the dimer at approximately 5.5Å separation, generated using Ovito[44]. The underlying representation to the SOAP formalism, wherein each atom is represented by a Gaussian field, and we expand n-body terms of each atomic neighborhood. The underlying representation for the AniSOAP formalism, wherein each *molecule* is represented by an anisotropic Gaussian field, and we expand n-body terms of each *molecular* neighborhood.

chosen cutoff distance and radial resolution are sufficiently large to include any necessary neighbors, which is evidenced by the perfect agreement in the computed and predicted h_{12} distances given in Fig. 3[43]. And thus, in AniSOAP , we alleviate the requirement to explicitly compute h_{12} or rely on its approximations.

From here, learning the Gay-Berne potential is straightforward, similar to how traditional SOAP vectors can learn the LJ potential. With the AniSOAP power spectrum vectors, it is possible to interpolate, using solely regularized linear regression, the interaction potential of ellipsoids of arbitrary distances from one another, as shown in Fig. 3. Performance decreases for repulsive, heavily overlapped configurations (right end of the parity plot in the lower right of Fig. 3), likely due to the fact that, in the repulsive regime, small feature differences correspond to disproportionately large energy differences. Our learning exercise converges with as few as 1,000 training points, as evidenced by the learning curve given in Fig. 3 of the supplementary material.

C. Benzene Molecules

For the following analysis, we intentionally construct a set of benzene configurations to demonstrate the possibilities and, simultaneously, the limitations for AniSOAP .

We start with a set of symmetry-constrained crystals comprised of planar benzene molecules with the software `PyXtal`[45] across all 230 space groups. We then augmented this dataset by randomly rotating or translating the molecules, where many configurations correspond to the same *positions* or *orientations* of the molecules. The resultant dataset contains roughly 7,000 benzene crystals. We then compute energetic quantities using QuantumEspresso v7.0[46] using Perdew–Burke-Ernzerhof (PBE) pseudopotentials and cutoff parameters reported by Prandini *et al.*[47], a Grimme D3-dispersion correction[48], and a 3x3 Monkhorst-Pack k-point grid[49]. Computations were managed with the `signac` and `signac-flow` packages[50,51], and computed using the `ASE` computational front-end[52].

Hyperparameter Tuning

With any new featurization, there is always valid concern about hyperparameter optimization and the sensitivity of the featurization to changes in these values[53]. In many ways, this is the appeal of the SOAP formalism – the hyperparameters, namely the widths of the Gaussian densities, basis sets, and cutoff parameters, all have roots in chemical physics and can be chosen from a large range of "reasonable" values with minimal impact on model interpretability and performance.

As the primary axes of planar benzenes are well-defined, the only new hyperparameters to consider are the semiaxes lengths of the MVG. For simplicity, we will only consider $\sigma_1 = \sigma_2$, and prove that, in line with conventional knowledge on benzene geometry, best results are obtained with $\sigma_1 > \sigma_3$ (an oblate ellipsoid).

To tune these parameters, we again use the GFRE of a given AniSOAP representation and an analogous SOAP rep-

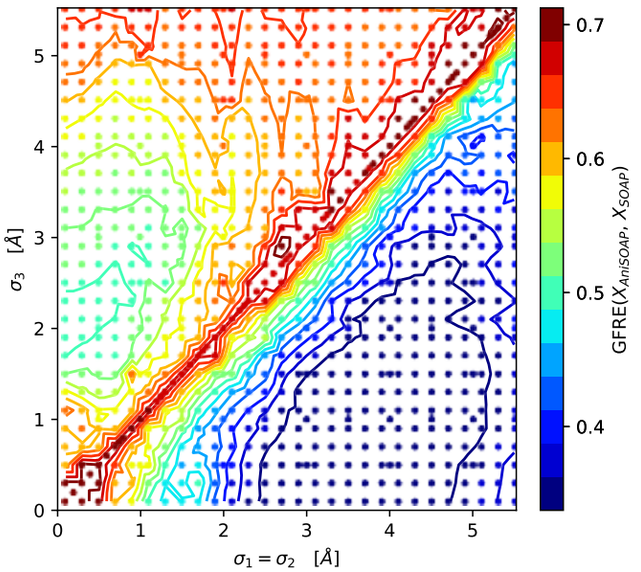


FIG. 5. **Results of tuning σ_1 and σ_3 for benzene crystals.** The color of the scatter points and contour levels denote $\mathrm{GFRE}(X_{\mathrm{AniSOAP}}, X_{\mathrm{SOAP}})$, where lower values indicate a greater fidelity of the AniSOAP representation to the SOAP analog, and higher values indicate a greater information loss. Our MVG is oblate where $\sigma_1 > \sigma_3$, prolate where $\sigma_1 < \sigma_3$, and spherical where $\sigma_1 = \sigma_3$. As expected, oblate representations carry more mutual information to the atomistic representation than prolate or spherical ones. The spherical representation $\sigma_1 = \sigma_2 = \sigma_3$ is consistent with constructing a traditional SOAP representation from the molecule centers.

resentation of our benzene crystal configurations. By taking $\mathrm{GFRE}(X_{\mathrm{AniSOAP}}, X_{\mathrm{SOAP}})$, we determine how much information is lost by moving from the atomistic to coarse-grained AniSOAP representation, again where 0.0 infers perfect reconstruction of the atomic correlations, and higher numbers correspond to poor reconstruction. Note that this hyperparameter tuning occurs independently of any traditional supervised learning task.

For our SOAP representation, we compute the structure-averaged representation using `rascaline`[54], with a cutoff radius of 7.0Å, Gaussian density width of 0.5Å, $l_{\mathrm{max}} = 10$, and $n_{\mathrm{max}} = 4$. This results in a length 1,188 vector for each configuration. We compare against AniSOAP vectors using a similar cutoff radius, number of angular and radial channels, varying $\sigma_1 = \sigma_2$, σ_3, and σ_{GTO}. Each AniSOAP vector has length 146.

From the results in Fig. 5, we see how the oblate MVG with $\sigma_1 = \sigma_2 = 4.0$ and $\sigma_3 = 0.5$ minimizes the information loss compared to the atomistic SOAP representation with a $\sigma_{GTO} = 1.5$. Prolate ($\sigma_1 < \sigma_3$) and spherical ($\sigma_1 = \sigma_3$) Gaussian densities perform notably worse. It is interesting, however, that there is a wide range of σ_1, σ_3 values that obtain similar results, signifying that the AniSOAP hyperparameters are robust with respect to small changes, provided the MVG remain oblate, and the semiaxis lengths are within reasonable proportion. This indicates that fitting on AniSOAP vectors is robust to small changes in hyperparameters; without this robustness, models are prone to overfit. We include in the SI similar analyses for $\sigma_{\mathrm{GTO}} = \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$, demonstrating that $\sigma_{\mathrm{GTO}} \in [1.0\text{Å}, 3.0\text{Å}]$ obtains similar results to $\sigma_{\mathrm{GTO}} = 1.5\text{Å}$.

Learning of Benzene Energetics

Taking the optimized hyperparameters from the previous section, we now show how the AniSOAP representation performs in simple supervised tasks, demonstrating where AniSOAP performs best and where it is incomplete for atomistic systems. It is worth noting that the intention of AniSOAP is to provide a molecule-level analog to SOAP, and is, by design, incomplete[55] with respect to many atom-atom correlations.

To highlight the importance of molecular representation, we will use regularized ridge models, employing a 90/10 training/test split and five-fold cross-validation. We first perform three regression tasks – learning the baselined, per-atom energy (equivalent to learning the per-molecule energy) using the all-atom SOAP representation, a SOAP representation built solely from the molecule centers (coinciding with the ridge line at $\sigma_1 = \sigma_2 = \sigma_3$ in Fig. 5), and our optimized AniSOAP representation. We have enforced that all three representations have similar ranks, as higher-rank features will outperform lower-rank ones based on size alone. To do so, we perform dimensionality reduction via principal components analysis[56].

The results, as shown in Fig. 6, demonstrate how incorporating the anisotropy of intermolecular correlations can greatly improve our ability to learn energetic quantities, even in contexts where atom-atom interactions are important. A SOAP representation on the molecule centers (center panels)[57], has heavily limited regression performance, with a typical RMSE on the order of $45 - 50$meV/atom, and an $R^2 \leq 0.4$. By introducing intermolecular anisotropy, our performance jumps $R^2 \approx 0.9, \mathrm{RMSE} \approx 15 - 20$meV/atom, much closer to the all-atom regression in the left panel of Fig. 6.

While AniSOAP can achieve reasonable performance, the performance of AniSOAP cannot match that of all-atom SOAP due to the nature of the underlying dataset. In this dataset, we specifically include configurations that will always be challenging to represent via molecular[58] coarse-graining, such as those with unstable hydrogen-hydrogen repulsion, or stabilizing hydrogen-hole interactions. As shown in the dynamic chemiscope[59] visualization in the corresponding Materials Cloud entry[60,61], these configurations are those with the largest error when regressed using the AniSOAP representation. This shows that the limits of AniSOAP correspond to the limits of coarse-graining itself.

IV. CONCLUSIONS

The tools developed under the umbrella of "machine-learning" are powerful, and revolutionizing society and the scientific community at an aggressive pace. One important advancement for chemical sciences is the concept of treating

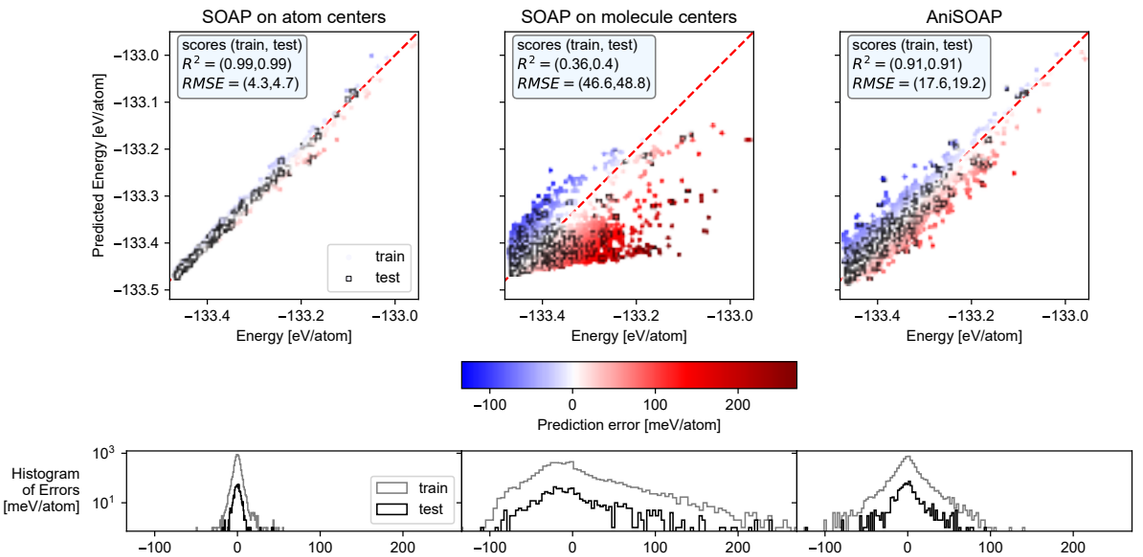


FIG. 6. **Parity plots comparing the performance of SOAP (built from the atom centers), optimized SOAP (built on the molecule centers), and AniSOAP** for the dataset of benzene crystalline configurations. All regressions were conducted using 5-fold cross-validated regularized linear regression, and errors are reported for the same 90/10 training/test split. The SOAP representations was first reduced via PCA to be of similar rank to AniSOAP . In the parity plots, color denotes the error in meV/atom, with the corresponding distribution of errors shown in the lower panel.

chemistry as data – choosing specifically the numerical representations with which to cast chemical questions into statistical and analytical models. These representations are strongest when grounded in the established physical principles that we are trying to emulate or predict.

Here, we have demonstrated one such physically-driven approach to machine-learning representations, aimed at leveraging machine learning for coarse-grained and mesoscale entities, demonstrating such an approach for particles and molecules that are well-represented by ellipsoidal bodies. Our results show that, for both classical and quantum mechanical datasets, AniSOAP provides a suitable representation for ellipsoidal bodies, as these representations are able to accurately and *linearly* map onto complex energetics without the need for deeper ML infrastructures. We have constructed AniSOAP in such a way as to retain compatibility with the popular atomistic SOAP formalism – in practice, we hope that these two technologies are used hand-in-hand to simplify energetic landscapes and explicitly incorporate many-body effects. Furthermore, this consistency between formalisms can enable multiscale machine learning approaches, wherein one can use coarse-grained representations for long-scale molecular motion and atom-atom representations for nearsighted interactions.

By feeding AniSOAP descriptors into various functional forms to learn molecular energy, we envision these anisotropic descriptors to be foundational for building general coarse-grained potentials of anisotropic molecules, similar to how many machine-learned interatomic potentials are built off atomistic descriptors[16,17,21] . Future work will focus on the efficient implementation of these anisotropic potentials to explore complex, multiscale molecular systems and compare the performance of AniSOAP-based potentials with traditional methods focused on reproducing the potential of mean force. For full integration into molecular dynamics, further derivations will also be necessary to calculate spatial derivatives of these descriptors for use in predicting the forces and torques on each particle.

V. SUPPLEMENTARY MATERIAL

The supplementary material contains additional analysis on our case studies. For the liquid crystals, we show how AniSOAP represents differently-shaped spheroids, and that it can delineate differently-shaped particles (unlike existing order parameters). For the Gay-Berne and Benzene case studies, we show learning curves and further results of our GFRE-based hyperparameter optimization.

VI. DATA AVAILABILITY

Data for this paper will be made available via MaterialsCloud[60] upon acceptance, including all raw input files and analysis scripts.

VII. AUTHOR CONTRIBUTIONS

KKHD derived the expansion of multi-variate Gaussian and wrote the corresponding proof-of-principle code. AL, JN, Y-CC and RKC refined the code, implementing orthonormalization and additional basis sets. AL and RKC wrote the manuscript and designed and executed the case studies.

VIII. CONFLICTS OF INTEREST

There are no conflicts to declare.

IX. ACKNOWLEDGEMENTS

This project was funded by the Wisconsin Alumni Research Fund (RKC), by NSF through the University of Wisconsin Materials Research Science and Engineering Center (DMR-2309000, AL), and the European Research Council (ERC) under the research and innovation program (Grant Agreement No. 101001890-FIAMMA, JN, KKHD).

We extend our un-ending gratitude to Guillaume Fraux and the developers of `rascaline` for fielding our many questions during the implementation and validation of AniSOAP .

X. REFERENCES

[1] J. I. Monroe and V. K. Shen, "Learning Efficient, Collective Monte Carlo Moves with Variational Autoencoders," Journal of Chemical Theory and Computation **18**, 3622–3636 (2022).

[2] J. I. Monroe and V. K. Shen, "Systematic control of collective variables learned from variational autoencoders," The Journal of Chemical Physics **157**, 094116 (2022).

[3] A. d. S. Costa, I. Mitnikov, M. Geiger, M. Ponnapati, T. Smidt, and J. Jacobson, "Ophiuchus: Scalable Modeling of Protein Structures through Hierarchical Coarse-graining SO(3)-Equivariant Autoencoders," (2023), arXiv:2310.02508 [cs].

[4] M. Majewski, A. Pérez, P. Thölke, S. Doerr, N. E. Charron, T. Giorgino, B. E. Husic, C. Clementi, F. Noé, and G. De Fabritiis, "Machine learning coarse-grained potentials of protein thermodynamics," Nature Communications **14**, 5739 (2023), number: 1 Publisher: Nature Publishing Group.

[5] J. Ruza, W. Wang, D. Schwalbe-Koda, S. Axelrod, W. H. Harris, and R. Gómez-Bombarelli, "Temperature-transferable coarse-graining of ionic liquids with dual graph convolutional neural networks," The Journal of Chemical Physics **153**, 164501 (2020).

[6] B. E. Husic, N. E. Charron, D. Lemm, J. Wang, A. Pérez, M. Majewski, A. Krämer, Y. Chen, S. Olsson, G. de Fabritiis, F. Noé, and C. Clementi, "Coarse graining molecular dynamics with graph neural networks," The Journal of Chemical Physics **153**, 194101 (2020).

[7] J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. de Fabritiis, F. Noé, and C. Clementi, "Machine Learning of Coarse-Grained Molecular Dynamics Force Fields," ACS Central Science **5**, 755–767 (2019), publisher: American Chemical Society.

[8] L. Zhang, J. Han, H. Wang, R. Car, and W. E, "DeePCG: Constructing coarse-grained models via deep neural networks," The Journal of Chemical Physics **149**, 034101 (2018).

[9] J. Wang, S. Chmiela, K.-R. Müller, F. Noé, and C. Clementi, "Ensemble learning of coarse-grained molecular dynamics force fields with a kernel approach," The Journal of Chemical Physics **152**, 194106 (2020).

[10] S. De, A. P. Bartók, G. Csányi, and M. Ceriotti, "Comparing molecules and solids across structural and alchemical space," Physical Chemistry Chemical Physics **18**, 13754–13769 (2016).

[11] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, and G. Csányi, "Gaussian Process Regression for Materials and Molecules," Chemical Reviews **121**, 10073–10141 (2021).

[12] F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi, and M. Ceriotti, "Physics-Inspired Structural Representations for Molecules and Materials," Chemical Reviews **121**, 9759–9815 (2021), publisher: American Chemical Society.

[13] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," Journal of Chemical Information and Computer Sciences **28**, 31–36 (1988), publisher: American Chemical Society.

[14] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik, "Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation," Machine Learning: Science and Technology **1**, 045024 (2020), publisher: IOP Publishing.

[15] D. S. Wigh, J. M. Goodman, and A. A. Lapkin, "A review of molecular representation in the age of machine learning," WIREs Computational Molecular Science **12**, e1603 (2022), _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1603.

[16] J. Behler, "Atom-centered symmetry functions for constructing high-dimensional neural network potentials," The Journal of Chemical Physics **134**, 074106 (2011).

[17] A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," Physical Review B **87**, 184115 (2013).

[18] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "MoleculeNet: a benchmark for molecular machine learning," Chemical Science **9**, 513–530 (2018), publisher: Royal Society of Chemistry.

[19] J. Nigam, S. Pozdnyakov, and M. Ceriotti, "Recursive evaluation and iterative contraction of N-body equivariant features," The Journal of Chemical Physics **153** (2020), publisher: AIP Publishing.

[20] J. Nigam, S. Pozdnyakov, G. Fraux, and M. Ceriotti, "Unified theory of atom-centered representations and message-passing machine-learning schemes," The Journal of Chemical Physics **156**, 204115 (2022).

[21] I. Batatia, D. P. Kovacs, G. Simm, C. Ortner, and G. Csanyi, "MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields," Advances in Neural Information Processing Systems **35**, 11423–11436 (2022).

[22] F. Musil, M. Veit, A. Goscinski, G. Fraux, M. J. Willatt, M. Stricker, T. Junge, and M. Ceriotti, "Efficient implementation of atom-density representations," The Journal of Chemical Physics **154**, 114109 (2021).

[23] F. Bigi, K. K. Huguenin-Dumittan, M. Ceriotti, and D. E. Manolopoulos, "A smooth basis for atomistic machine learning," The Journal of Chemical Physics **157**, 234101 (2022).

[24] E. Wigner, *Gruppentheorie und ihre Anwendung auf die Quantenmechanik der Atomspektren* (Vieweg+Teubner Verlag, Wiesbaden, 1931).

[25] P.-O. Löwdin, "On the Nonorthogonality Problem**The work reported in this paper has been sponsored in part by the Swedish Natural Science Research Council, in part by the Air Force Office of Scientific Research (OSR) through the European Office of Aerospace Research (OAR), U.S. Air Force under Grant AF-EOAR 67-50 with Uppsala University, and in part by the National Science Foundation under Grant GP-5419 with the University of Florida." in *Advances in Quantum Chemistry*, Vol. 5, edited by P.-O. Löwdin (Academic Press, 1970) pp. 185–199.

[26] V. G. Satorras, E. Hoogeboom, and M. Welling, "E(n) Equivariant Graph Neural Networks," (2022), arXiv:2102.09844 [cs, stat].

[27] Zhang, A. S. Keys, T. Chen, and S. C. Glotzer, "Self-Assembly of Patchy Particles into Diamond Structures through Molecular Mimicry," Langmuir **21**, 11547–11551 (2005).

[28] S. C. Glotzer and M. J. Solomon, "Anisotropy of building blocks and their assembly into complex structures," Nature Materials **6**, 557–562 (2007).

[29] R. K. Cersonsky, J. Dshemuchadse, J. Antonaglia, G. Van Anders, and S. C. Glotzer, "Pressure-tunable photonic band gaps in an entropic colloidal crystal," Physical Review Materials **2**, 125201 (2018).

[30] R. K. Cersonsky, G. Van Anders, P. M. Dodd, and S. C. Glotzer, "Relevance of packing to colloidal self-assembly," Proceedings of the National Academy of Sciences **115**, 1439–1444 (2018).

[31]A. S. Keys, C. R. Iacovella, and S. C. Glotzer, "Characterizing complex particle morphologies through shape matching: Descriptors, applications, and algorithms," Journal of Computational Physics **230**, 6438–6463 (2011).
[32]G. Van Anders, D. Klotsa, N. K. Ahmed, M. Engel, and S. C. Glotzer, "Understanding shape entropy through local dense packing," Proceedings of the National Academy of Sciences **111** (2014), 10.1073/pnas.1418159111.
[33]I. W. Hamley, *Introduction to Soft Matter– Revised Edition* (John Wiley & Sons, Ltd, Chichester, UK, 2007).
[34]V. Ramasubramani, B. D. Dice, E. S. Harper, M. P. Spellings, J. A. Anderson, and S. C. Glotzer, "freud: A software suite for high throughput analysis of particle simulation data," Computer Physics Communications **254**, 107275 (2020).
[35]P. J. Steinhardt, D. R. Nelson, and M. Ronchetti, "Bond-orientational order in liquids and glasses," Physical Review B **28**, 784–805 (1983), publisher: American Physical Society.
[36]W. Lechner and C. Dellago, "Accurate determination of crystal structures based on averaged local bond order parameters," The Journal of Chemical Physics **129**, 114707 (2008).
[37]E. Prodan and W. Kohn, "Nearsightedness of electronic matter," Proceedings of the National Academy of Sciences **102**, 11635–11638 (2005), publisher: Proceedings of the National Academy of Sciences.
[38]A. Goscinski, G. Fraux, G. Imbalzano, and M. Ceriotti, "The role of feature space in atomistic learning," Machine Learning: Science and Technology **2**, 025028 (2021), publisher: IOP Publishing.
[39]A. Goscinski, V. P. Principe, G. Fraux, S. Kliavinek, B. A. Helfrecht, P. Loche, M. Ceriotti, and R. K. Cersonsky, "scikit-matter : A Suite of Generalisable Machine Learning Methods Born out of Chemistry and Materials Science," Open Research Europe **3**, 81 (2023).
[40]R. Everaers and M. R. Ejtehadi, "Interaction potentials for soft and hard ellipsoids," Physical Review E **67**, 041710 (2003), arXiv:cond-mat/0306096.
[41]R. Berardi, C. Fava, and C. Zannoni, "A generalized Gay-Berne intermolecular potential for biaxial particles," Chemical Physics Letters **236**, 462–468 (1995).
[42]J. W. Perram and M. S. Wertheim, "Statistical mechanics of hard ellipsoids. I. Overlap algorithm and the contact function," Journal of Computational Physics **58**, 409–416 (1985).
[43]The h_{12} distances here fall within the confidence region of the Perram and Wertheim [42] algorithm, so they can be taken as exact.
[44]A. Stukowski, "Visualization and analysis of atomistic simulation data with OVITO–the Open Visualization Tool," Modelling and Simulation in Materials Science and Engineering **18**, 015012 (2009).
[45]S. Fredericks, K. Parrish, D. Sayre, and Q. Zhu, "PyXtal: A Python library for crystal structure generation and symmetry analysis," Computer Physics Communications **261**, 107810 (2021).
[46]P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. D. Corso, S. d. Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch, "QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials," Journal of Physics: Condensed Matter **21**, 395502 (2009).
[47]G. Prandini, A. Marrazzo, I. E. Castelli, N. Mounet, and N. Marzari, "Precision and efficiency in solid-state pseudopotential calculations," npj Computational Materials **4**, 1–13 (2018), number: 1 Publisher: Nature Publishing Group.
[48]S. Grimme, A. Hansen, J. G. Brandenburg, and C. Bannwarth, "Dispersion-Corrected Mean-Field Electronic Structure Methods," Chemical Reviews **116**, 5105–5154 (2016), publisher: American Chemical Society.
[49]H. J. Monkhorst and J. D. Pack, "Special points for Brillouin-zone integrations," Physical Review B **13**, 5188–5192 (1976), publisher: American Physical Society.
[50]V. Ramasubramani, C. Adorf, P. Dodd, B. Dice, and S. Glotzer, "signac: A Python framework for data and workflow management," (Austin, Texas, 2018) pp. 152–159.
[51]C. S. Adorf, P. M. Dodd, V. Ramasubramani, and S. C. Glotzer, "Simple data and workflow management with the signac framework," Computational Materials Science **146**, 220–229 (2018).
[52]A. Hjorth Larsen, J. Jørgen Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Bjerre Jensen, J. Kermode, J. R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, "The atomic simulation environment—a Python library for working with atoms," Journal of Physics: Condensed Matter **29**, 273002 (2017).
[53]B. Cheng, R.-R. Griffiths, S. Wengert, C. Kunkel, T. Stenczel, B. Zhu, V. L. Deringer, N. Bernstein, J. T. Margraf, K. Reuter, and G. Csanyi, "Mapping Materials and Molecules," Accounts of Chemical Research **53**, 1981–1991 (2020), publisher: American Chemical Society.
[54]G. Fraux, P. Loche, S. Kliavinek, F. Bigi, K. Hugeunin-Dumittan, A. Grisafi, R. K. Cersonsky, M. Kellner, D. Tisi, S. Shah, A. Goscinski, and M. Ceriotti, "rascaline," (2023).
[55]S. N. Pozdnyakov, M. J. Willatt, A. P. Bartók, C. Ortner, G. Csányi, and M. Ceriotti, "On the Completeness of Atomic Structure Representations," Physical Review Letters **125**, 166001 (2020), arXiv:2001.11696 [cond-mat, physics:physics].
[56]Principal covariates regression[62], while ideal for regression-related dimensionality reductions, would again limit our ability to compare the *relevant information content* of these representations.
[57]Using the same major semiaxis length as the width of the isotropic Gaussian field, and keeping all other hyperparameters consistent.
[58]One bead per molecule.
[59]G. Fraux, R. K. Cersonsky, and M. Ceriotti, "Chemiscope: interactive structure-property explorer for materials and molecules," Journal of Open Source Software **5**, 2117 (2020).
[60]L. Talirz, S. Kumbhar, E. Passaro, A. V. Yakutovich, V. Granata, F. Gargiulo, M. Borelli, M. Uhrin, S. P. Huber, S. Zoupanos, C. S. Adorf, C. W. Andersen, O. Schütt, C. A. Pignedoli, D. Passerone, J. VandeVondele, T. C. Schulthess, B. Smit, G. Pizzi, and N. Marzari, "Materials Cloud, a platform for open computational science," Scientific Data **7**, 299 (2020), number: 1 Publisher: Nature Publishing Group.
[61]Upon submission, we will upload our datasets and chemiscopes to Materials Cloud[60].
[62]B. A. Helfrecht, R. K. Cersonsky, G. Fraux, and M. Ceriotti, "Structure-property maps with Kernel principal covariates regression," Machine Learning: Science and Technology **1**, 045021 (2020), publisher: IOP Publishing.
[63]F. Bigi, G. Fraux, N. J. Browning, and M. Ceriotti, "Fast evaluation of spherical harmonics with sphericart," The Journal of Chemical Physics **159**, 064802 (2023).
[64]A. Goscinski, F. Musil, S. Pozdnyakov, J. Nigam, and M. Ceriotti, "Optimal radial basis for density-based atomic representations," The Journal of Chemical Physics **155**, 104106 (2021).
[65]F. Corbato, "A Paging Experiment with the Multics System," in *MIT Press* (1969).
[66]N. D. Matsakis and F. S. Klock, "The rust language," ACM SIGAda Ada Letters **34**, 103–104 (2014).

Appendix A: Evaluating the Coefficients

We have shown that, by choosing a MVG density and suitable basis functions, we can analytically compute the coefficients

$$\langle nlm|\rho_i\rangle, \tag{A1}$$

by evaluating of integrals of the form

$$I_{nlm} = \int_{\mathbb{R}^3} \mathrm{d}^3 r e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{a})^T A(\boldsymbol{r}-\boldsymbol{a})} r^{2n} R_l^m(\boldsymbol{r}). \tag{A2}$$

where $R_l^m = r^l Y_l^m(\hat{\mathbf{r}})$, a polynomial of degree l, and $r^{2n}R_l^m(\boldsymbol{r})$ is a polynomial of degree $l+2n$.

In this section, we will present an explicit algorithm to obtain all I_{nlm}. The algorithm will require two key ingredients, namely converting the spherical formulation of $r^{2n}R_l^m(\boldsymbol{r})$ into Cartesian coordinates (Sec. A 2), then computing the corresponding moments across the subject to our MVG (Sec. A 3). Section A 4 gives an overview of how to combine these two components to recapitulate I_{nlm}. Details on these transformations are given in Sec. A 5, and A 6.

1. Computing the effective quadratic form in the exponential

If we use the monomial basis, the integrand in

$$I_{nlm} = \int_{\mathbb{R}^3} \mathrm{d}^3 r e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{r}_{ij})^T A(\boldsymbol{r}-\boldsymbol{r}_{ij})} r^{2n} R_l^m(\boldsymbol{r}) \tag{A3}$$

is already in a convenient form, since the Gaussian part is completely specified by the center $\boldsymbol{r}_{ij}$ and the precision matrix A. If, on the other hand, we use the GTO basis, we get an extra exponential factor

$$I_{nlm} = \int_{\mathbb{R}^3} \mathrm{d}^3 r e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{r}_{ij})^T A(\boldsymbol{r}-\boldsymbol{r}_{ij})} r^{2n} R_l^m(\boldsymbol{r}) e^{-\frac{r^2}{2\sigma^2}} \tag{A4}$$

$$= \int_{\mathbb{R}^3} \mathrm{d}^3 r e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{r}_{ij})^T A(\boldsymbol{r}-\boldsymbol{r}_{ij}) - \frac{r^2}{2\sigma^2}} r^{2n} R_l^m(\boldsymbol{r}) \tag{A5}$$

meaning that the Gaussian part

$$e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{r}_{ij})^T A(\boldsymbol{r}-\boldsymbol{r}_{ij}) - \frac{r^2}{2\sigma^2}} \tag{A6}$$

as a whole no longer has the convenient form. By completing the square, ignoring the global factor of $-\frac{1}{2}$ in the exponent, we can rewrite

$$(\boldsymbol{r}-\boldsymbol{r}_{ij})^T A(\boldsymbol{r}-\boldsymbol{r}_{ij}) + \frac{1}{\sigma^2}\boldsymbol{r}^2 = (\boldsymbol{r}-\boldsymbol{r}_0)^T \tilde{A}(\boldsymbol{r}-\boldsymbol{r}_0) + c, \tag{A7}$$

with

$$\tilde{A} = A + \frac{1}{\sigma^2} \tag{A8}$$

$$\boldsymbol{r}_0 = \tilde{A}^{-1} A \boldsymbol{r}_{ij} = \boldsymbol{r}_{ij} - \frac{1}{\sigma^2}\tilde{A}^{-1}\boldsymbol{r}_{ij} \tag{A9}$$

$$c = \frac{1}{\sigma^2}\boldsymbol{r}_{ij}^T \tilde{A}^{-1} A \boldsymbol{r}_{ij} \tag{A10}$$

The second form of $\boldsymbol{r}_0$ is more convenient to obtain a qualitative picture: the second term represents the deviation of the center due to the addition of the second Gaussian. Depending on how sharp this Gaussian is, the relative importance of this term will change.

Firstly, note that using the first representation of $\boldsymbol{r}_0$ and the fact that both A and $\tilde{A}$ are symmetric, we get

$$\boldsymbol{r}_0^T \tilde{A} \boldsymbol{r}_0 = \boldsymbol{r}_{ij}^T A \tilde{A}^{-1} \tilde{A} \tilde{A}^{-1} A \boldsymbol{r}_{ij} \tag{A11}$$

$$= \boldsymbol{r}_{ij}^T A \tilde{A}^{-1} A \boldsymbol{r}_{ij} \tag{A12}$$

$$= \boldsymbol{r}_{ij}^T A \tilde{A}^{-1} \left(\tilde{A} - \frac{1}{\sigma^2}\right) \boldsymbol{r}_{ij} \tag{A13}$$

$$= \boldsymbol{r}_{ij}^T A \boldsymbol{r}_{ij} - \frac{1}{\sigma^2}\boldsymbol{r}_{ij}^T \tilde{A}^{-1} A \boldsymbol{r}_{ij} \tag{A14}$$

Expanding the quadratic terms, we get

$$(\boldsymbol{r}-\boldsymbol{r}_0)^T A(\boldsymbol{r}-\boldsymbol{r}_0) = \boldsymbol{r}^T\tilde{A}\boldsymbol{r} - 2\boldsymbol{r}^T\tilde{A}\boldsymbol{r}_0 + \boldsymbol{r}_0^T\tilde{A}\boldsymbol{r}_0 \tag{A15}$$

$$= \boldsymbol{r}^T A\boldsymbol{r} + \frac{1}{\sigma^2}\boldsymbol{r}^2 - 2\boldsymbol{r}^T\tilde{A}\left(\boldsymbol{r}_{ij} - \frac{1}{\sigma^2}\tilde{A}^{-1}\boldsymbol{r}_{ij}\right) + \boldsymbol{r}_0^T\tilde{A}\boldsymbol{r}_0 \tag{A16}$$

$$= \boldsymbol{r}^T A\boldsymbol{r} + \frac{1}{\sigma^2}\boldsymbol{r}^2 - 2\boldsymbol{r}^T\tilde{A}\boldsymbol{r}_{ij} + 2\frac{1}{\sigma^2}\boldsymbol{r}^T\boldsymbol{r}_{ij} + \boldsymbol{r}_0^T\tilde{A}\boldsymbol{r}_0 \tag{A17}$$

$$= \boldsymbol{r}^T A\boldsymbol{r} + \frac{1}{\sigma^2}\boldsymbol{r}^2 - 2\boldsymbol{r}^T A\boldsymbol{r}_{ij} + \boldsymbol{r}_0^T\tilde{A}\boldsymbol{r}_0 \tag{A18}$$

$$= \boldsymbol{r}^T A\boldsymbol{r} + \frac{1}{\sigma^2}\boldsymbol{r}^2 - 2\boldsymbol{r}^T A\boldsymbol{r}_{ij} + \boldsymbol{r}_{ij}^T A\boldsymbol{r}_{ij} - \frac{1}{\sigma^2}\boldsymbol{r}_{ij}^T\tilde{A}^{-1}A\boldsymbol{r}_{ij} \tag{A19}$$

$$= (\boldsymbol{r}-\boldsymbol{r}_{ij})^T A(\boldsymbol{r}-\boldsymbol{r}_{ij}) + \frac{1}{\sigma^2}\boldsymbol{r}^2 - \frac{1}{\sigma^2}\boldsymbol{r}_{ij}^T\tilde{A}^{-1}A\boldsymbol{r}_{ij} \tag{A20}$$

Thus, we do see that indeed,

$$(\boldsymbol{r}-\boldsymbol{r}_{ij})^T A(\boldsymbol{r}-\boldsymbol{r}_{ij}) + \frac{1}{\sigma^2}\boldsymbol{r}^2 = (\boldsymbol{r}-\boldsymbol{r}_0)^T\tilde{A}(\boldsymbol{r}-\boldsymbol{r}_0) + \frac{1}{\sigma^2}\boldsymbol{r}_{ij}^T\tilde{A}^{-1}A\boldsymbol{r}_{ij}. \tag{A21}$$

2. Spherical to Cartesian Transformation

Recall that for both the monomial and GTO bases, our integrand is a product of a Gaussian and an expression of the form

$$r^{2n}R_l^m(\boldsymbol{r}) = \mathrm{poly}(x,y,z), \tag{A22}$$

where we emphasize that $r^{2n}R_l^m(\boldsymbol{r})$ is simply a polynomial in the variables (x,y,z).
For practical evaluations, however, it does not suffice to know that it is equal to "some" polynomial. We will need to explicitly express the solid harmonics $R_l^m(\boldsymbol{r})$ in monomial terms. In general, there will exist a decomposition

$$R_l^m(\boldsymbol{r}) = \sum_{n_0+n_1+n_2=l} T^{lm}_{n_0,n_1,n_2} x^{n_0}y^{n_1}z^{n_2}, \tag{A23}$$

where $T^{lm}_{n_0,n_1,n_2}$ are some coefficients. These coefficients depend only on l,m and can thus be cached. For the full basis function, there will then exist coefficients $T^{n,lm}_{n_0,n_1n_2}$ such that

$$r^{2n}R_l^m(\boldsymbol{r}) = \sum_{n_0+n_1+n_2=2n+l} T^{n,lm}_{n_0,n_1,n_2} x^{n_0}y^{n_1}z^{n_2}. \tag{A24}$$

As soon as we are provided with a complete list of the (l,n) pair we need, we can precompute these coefficients.

3. Evaluation of Moments

Once we have decomposed $r^{2n}R_l^m(\boldsymbol{r})$ into a sum of monomial terms, we have reduced the evaluation problem to integrals of the form

$$\langle x^{n_0}y^{n_1}z^{n_2}\rangle = \int_{\mathbb{R}^3} \mathrm{d}^3 r e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{a})^T A(\boldsymbol{r}-\boldsymbol{a})} x^{n_0}y^{n_1}z^{n_2}. \tag{A25}$$

We shall call this expression the (n_0,n_1,n_2)-th moment of the Gaussian, or just moment for short.

To provide some context for the terminology, "moment" is a term used in probability theory. If we are given some probability density (in three variables) $p(\boldsymbol{r}) = p(x,y,z)$, its (n_0,n_1,n_2)-th moment is defined as

$$\int_{\mathbb{R}^3} \mathrm{d}^3 r p(\boldsymbol{r}) x^{n_0}y^{n_1}z^{n_2}. \tag{A26}$$

The most common notation for this in the mathematical literature is to write it as $\mathbb{E}[x^{n_0}y^{n_1}z^{n_2}]$. In the quantum field theory / statistical mechanics literature, on the other hand, the notation $\langle x^{n_0}y^{n_1}z^{n_2}\rangle$ is used more commonly instead.

Connecting this back to our integrals, we can see that the Gaussian part is almost a probability density, apart from the normalization factor, motivating the use of an analogous notation.

The normalization factor is a global factor, meaning that techniques that have been developed to evaluate moments in probability theory can be applied to our problem as well. This will allow us to compute all the moments for a given Gaussian function specified by the precision matrix A and the center $\boldsymbol{a}$.

4. Putting Everything Together

Combining the two ingredients, we can now formulate the complete algorithm for the evaluation of the above-mentioned integrals, namely:

$$I_{nlm} = \int_{\mathbb{R}^3} \mathrm{d}^3 r e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{a})^T A(\boldsymbol{r}-\boldsymbol{a})} r^{2n} R_l^m(\boldsymbol{r}) \tag{A27}$$

$$= \int_{\mathbb{R}^3} \mathrm{d}^3 r e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{a})^T A(\boldsymbol{r}-\boldsymbol{a})} \sum_{n_0,n_1,n_2} T^{n,lm}_{n_0,n_1,n_2} x^{n_0} y^{n_1} z^{n_2} \tag{A28}$$

$$= \sum_{n_0,n_1,n_2} T^{n,lm}_{n_0,n_1,n_2} \int_{\mathbb{R}^3} \mathrm{d}^3 r e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{a})^T A(\boldsymbol{r}-\boldsymbol{a})} x^{n_0} y^{n_1} z^{n_2} \tag{A29}$$

$$= \sum_{n_0,n_1,n_2} T^{n,lm}_{n_0,n_1,n_2} \langle x^{n_0} y^{n_1} z^{n_2} \rangle \tag{A30}$$

We can therefore see that the computation of our features is completely determined by the two ingredients mentioned before, namely:

1. the transformation coefficients $T^{n,lm}_{n_0,n_1 n_2}$ that express $r^{2n} R_l^m$ using monomials
2. the moments $\langle x^{n_0} y^{n_1} z^{n_2} \rangle$ of the Gaussian

In the following two subsections, we will explain how to perform these two steps, respectively.

5. Evaluation of Transformation Coefficients

To compute the transformation coefficients $T^{nlm}_{n0,n1,n2}$ for high orders of n, we utilize the recursive structure of the radial term of $r^{2n} R_l^m$:

$$r^{2(n+1)} R_l^m = r^2 r^{2n} R_l^m = (x^2 + y^2 + z^2) r^{2n} R_l^m \tag{A31}$$

Thus, if all transformation coefficients $T^{nlm}_{n_0,n_1,n_2}$ at some n are known, we obtain those for higher n by running the iteration:

$$\forall (l,m,n_0,n_1,n_2) \text{ with } \quad n_0+n_1+n_2 = l+2n \tag{A32}$$

$$T^{n+1,lm}_{(n_0+2),n_1,n_2} += T^{n,lm}_{n_0,n_1,n_2}$$

$$T^{n+1,lm}_{n_0,(n_1+2),n_2} += T^{n,lm}_{n_0,n_1,n_2}$$

$$T^{n+1,lm}_{n_0,n_1,(n_2+2)} += T^{n,lm}_{n_0,n_1,n_2}$$

After the iteration, all coefficients at the radial channel $n+1$ will have the correct values. To initialize this recurrence relation, we set the transformation coefficients $T^{n,lm}_{n_0,n_1,n_2} = 0$ for $n > 0$. For $n = 0$, $T^{n,lm}_{n_0,n_1,n_2}$ corresponds with the coefficients of R_l^m, as determined by a different set of recurrence relationships[63].

6. Computing the Moments

In this subsection, we explain how we can compute the moments

$$\langle x^{n_0} y^{n_1} z^{n_2} \rangle = \int_{\mathbb{R}^3} \mathrm{d}^3 r e^{-\frac{1}{2}(\boldsymbol{r}-\boldsymbol{a})^T \mathbf{A}(\boldsymbol{r}-\boldsymbol{a})} x^{n_0} y^{n_1} z^{n_2}. \tag{A33}$$

for a given precision matrix A and center $\boldsymbol{a}$.
As we will show, this step will be significantly easier if the matrix $\mathbf{A}$ is diagonal, i.e. if we work in the basis of the principal axes. In our current implementation, we still use the original coordinate frame in which the matrix $\mathbf{A}$ keeps its general form. Despite the slightly more complicated computation of the moments, this will reduce some computational cost associated with the rotation of the coordinate frame (summing over the Wigner matrix) later on. Nevertheless, there might still be some benefits to have an implementation that uses the diagonalization step. We will, therefore, start by discussing the simpler diagonal case, as well as extensions that would become possible for this special case. We then move on to the algorithm for general A, which is what we are using in the current implementation.

Diagonal Case In the diagonal case, the integral separates into an x-, y- and z-dependent part. The total integral can thus easily be evaluated by using the explicit formula for the one-dimensional Gaussian integral, namely:

$$\langle x^{n_0} y^{n_1} z^{n_2} \rangle = \int_{\mathbb{R}^3} \mathrm{d}^3 r x^{n_1} y^{n_2} z^{n_3} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_1^2}+\frac{y^2}{\sigma_2^2}+\frac{z^2}{\sigma_3^2}\right)} \tag{A34}$$

$$= \left(2\sigma_1^2\right)^{n_1'} \Gamma(n_1') \left(2\sigma_2^2\right)^{n_2'} \Gamma(n_2') \left(2\sigma_3^2\right)^{n_3'} \Gamma(n_3'). \tag{A35}$$

An advantage of the diagonal formalism is that it can easily be adapted to non-Gaussian densities. We could, for instance, consider densities of the form

$$g_0(\boldsymbol{r}) = \exp\left[-\left(\frac{|x|}{\sigma_x}+\frac{|y|}{\sigma_y}+\frac{|z|}{\sigma_z}\right)\right] \tag{A36}$$

This could be used to have densities that have the symmetry of a cube (if all σ_j are equal) or other rectangular shapes. While integrating these in spherical coordinates would be a nightmare due to the absolute values, in Cartesian coordinates, the computation is relatively simple once the problem is brought into this diagonal form. There could, therefore, be some value in having an implementation that works for the diagonalized case. It should be noted, however, that we are not directly working with the density. Instead, we are approximating the density with our basis functions. If the resolution of our basis functions is low, it might not be possible to properly distinguish Gaussians from rectangular densities. This should be kept in mind before spending too much time developing complicated schemes with minimal additional improvements.

General Case Here, we use an iterative updating scheme. For later convenience, we define the covariance matrix $C = A^{-1}$ and work with it instead. To keep the notation closer to the implementation, we will index its components as $0,1,2$ rather than $1,2,3$, corresponding to the coordinate axes x,y,z.
We can initialize the first few moments, up to a global factor, by

$$\langle 1 \rangle = \langle x^0 y^0 z^0 \rangle = 1 \tag{A37}$$

$$\langle x \rangle = a_x \tag{A38}$$

$$\langle y \rangle = a_y \tag{A39}$$

$$\langle z \rangle = a_z \tag{A40}$$

We can compute all higher-order moments by using three recurrence relations[19], one in the x-, y-, and z-directions, respectively. These are given by:
x-iteration:

$$\langle x^{n_0+1} y^{n_1} z^{n_2} \rangle = a_0 \langle x^{n_0} y^{n_1} z^{n_2} \rangle + C_{00} n_0 \langle x^{n_0-1} y^{n_1} z^{n_2} \rangle \tag{A41}$$

$$+ C_{01} n_1 \langle x^{n_0} y^{n_1-1} z^{n_2} \rangle + C_{02} n_2 \langle x^{n_0} y^{n_1} z^{n_2-1} \rangle \tag{A42}$$

y-iteration:

$$\langle x^{n_0} y^{n_1+1} z^{n_2} \rangle = a_1 \langle x^{n_0} y^{n_1} z^{n_2} \rangle + C_{10} n_0 \langle x^{n_0-1} y^{n_1} z^{n_2} \rangle \tag{A43}$$

$$+ C_{11} n_1 \langle x^{n_0} y^{n_1-1} z^{n_2} \rangle + C_{12} n_2 \langle x^{n_0} y^{n_1} z^{n_2-1} \rangle \tag{A44}$$

z-iteration:

$$\langle x^{n_0} y^{n_1} z^{n_2+1} \rangle = a_2 \langle x^{n_0} y^{n_1} z^{n_2} \rangle + C_{20} n_0 \langle x^{n_0-1} y^{n_1} z^{n_2} \rangle \tag{A45}$$

$$+ C_{21} n_1 \langle x^{n_0} y^{n_1-1} z^{n_2} \rangle + C_{22} n_2 \langle x^{n_0} y^{n_1} z^{n_2-1} \rangle. \tag{A46}$$

Warning: Please note that for any of the above iterations, it is possible that some exponents will become negative. A concrete example, in which we naively use the above formulae, would be the evaluation of $\langle x^2 \rangle$ from the x-iteration, which would give

$$\langle x^2 \rangle = a_0 \langle x^1 \rangle + C_{00} \cdot 1 \cdot \langle 1 \rangle + C_{01} \cdot 0 \cdot \langle x y^{-1} \rangle + C_{02} \cdot 0 \cdot \langle x z^{-1} \rangle \tag{A47}$$

In such cases, the terms containing negative exponents, strictly speaking, do not exist. As the multiplication by zero suggests, however, these terms do not contribute to the final result. The correct iteration in this case therefore is

$$\langle x^2 \rangle = a_0 \langle x^1 \rangle + C_{00} n_0 \langle 1 \rangle. \tag{A48}$$

In practice, depending on the specific implementation, it might not even be necessary to make special cases for this if all array elements are initialized to zero. Please also note that the initializations only apply to normalized Gaussians that have a proper interpretation as a probability density. For other normalizations, the coefficients will all be multiplied by the same global factor. Thus, in practice, it makes sense to include this global factor in the initialization, since the iterative scheme will then automatically ensure that all higher order moments have the correct prefactor as well.

7. Orthonormalization

The coefficients $\langle nlm|\rho_i\rangle$ from the above procedures result from using non-orthonormal bases. While not strictly necessary, working in orthonormal bases ensures minimal overlap in mutual information between features[64]. This discussion only applies to basis functions that are square integrable (e.g. GTO); non-square-integrable basis functions (e.g. monomial basis) cannot be orthonormalized. Below, we discuss how to scale our coefficients, taking the GTO basis as a representative example for any square-integrable basis.

We first choose the coefficients of our density expansion $\langle nlm|\rho_i\rangle$ to use the unnormalized GTO basis $R_{nl}(r) = r^{l+2n}e^{\frac{-r^2}{2\sigma^2}}$ for our radial expansion. The general unnormalized GTO $\phi_d = r^d e^{\frac{-r^2}{2\sigma_d^2}}$ has a finite square-integral over $\mathbb{R}^3$:

$$I_d = \int_0^\infty |\phi_d(r)|^2 * r^2 dr = 2^{-1}\sigma_d^{2d+3} * \Gamma(\frac{2d+3}{2})$$

and can hence be normalized with the constant $N_d = 1/\sqrt{I_d}$. In other words, our normalized GTO is $\Phi_d = N_d * \phi_d$, and $\int_0^\infty |\Phi_d(r)|^2 * r^2 dr = 1$. After normalizing all the bases, we can take orthogonalize them using Löwdin Symmetric Orthonormalization[25]. We first find the overlap matrix between two normalized GTOs:

$$G_{ij} = \int_0^\infty \Phi_i \Phi_j r^2 dr.$$

The orthonormalization matrix is the inverse square root of the overlap matrix $G^{-1/2}$, which is guaranteed to exist because the overlap matrix is a gram matrix and is hence symmetric positive definite. Specifically, $G^{-1/2}$ is calculated by diagonalizing G, then taking the recipricol of the square root of the diagonal matrix: $G = MDM^T$, $G^{-1/2} = MD^{-1/2}M^T$. Then, we can apply this matrix to obtain a set of orthonormal GTO basis vectors $\hat{\Phi}_i$: $\hat{\Phi}_i = G_{ij}^{-1/2}\Phi_j$. We note that in our previous procedure, we could in theory construct an overlap matrix from unnormalized GTOs, but practically, calculating the overlap between unnormalized GTOs of high order, then inverting them to find the orthonormalization matrix, is not numerically stable.

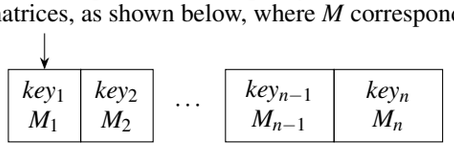
8. Tricks for Efficient Implementation

The current implementation of AniSOAP is performance limited, both by inefficient recalculations of the Clebsch-Gordan Matrices, and by the large number of computations and nested iterations performed in Python when calculating high-order moments. Below we outline two strategies under active development to address these inefficiencies. The computational costs of the current implementation and the proposed future implementations are shown in Fig. 7.

a. Caching the Clebsch-Gordan (CG) Matrix

The construction of Clebsch-Gordan matrices depends on only the hyperparameter l_{max}. Originally, the matrices were stored within a Python dictionary indexed by (l_1, l_2, L). Since $0 \leq l_1, l_2 \leq l_{max}$ and $|l_1 - l_2| \leq L \leq \min(l_{max}, (l_1 + l_2))$, the number of matrices required to compute grows as $\mathcal{O}(l_{max}^3)$, which is expensive even for modest values of l_{max}.

However, since the construction of CG matrices only depends on l_{max}, we can cache the matrices and re-use them internally. For caching, we use a simple cyclic list that stores `(key, value)` pair with `key` storing l_{max} and `value` storing corresponding matrices, as shown below, where M corresponds to the appropriate CG matrices.



The cache has a finite and fixed number of `(key, value)` pairs it can store to prevent memory overflow, and whenever the cache is full, it decides the entry for replacement using an algorithm that mimics the CLOCK algorithm for page replacement[65]. In accordance with the algorithm, the `key` for the CG list also contains a replacement bit to store the time-indexed usage of the entry. The format of the CLOCK algorithm does place implicit limits on l_{max}, such that $l_{max} \leq 2^{31} - 1$, although this value is beyond practical usage. The utility of caching these CG matrices is apparent when performing repeated higher-body order expansions with the same l_{max}, such as in a molecular dynamics simulation.

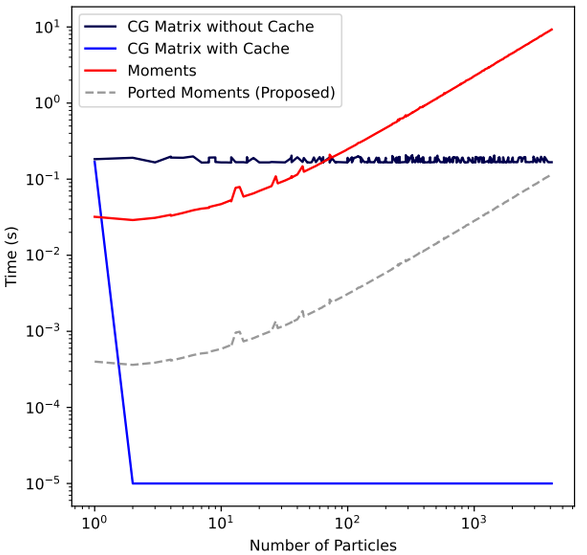


FIG. 7. **Time measurement of major bottlenecks** with and without caching. The same computation was performed for all N by N by M crystal structures with N and M both ranging from 1 to 16, and with fixed $l_{max} = 6, \sigma = 5.0, r_{cut} = 1.0$. With caching, after the first computation, all subsequent computations of Clebsch-Gordan matrices are near instantaneous ($\sim 10\mu s$, the time required to access a list). The gray dashed line show the expected ∼80x speedup when we port the moments calculation to a lower-level language like Rust. The red and gray lines approach a slope of 1 on the log-log plot, indicating linear scaling with system size.

b. *Moments Array*

Given the simplicity of the moments calculations and the frequency of invoking this section of code, we are currently porting this functionality to Rust[66], which gives the benefits of a compiled language while retaining easy interfacing with Python.

While the Rust code is almost a one-to-one translation, there were two minor changes. Firstly, the computation of the inverse of dilation matrix $\boldsymbol{D}$ (10) is changed. By definition, $\boldsymbol{D}$ is a 3 by 3 symmetric matrix. Therefore, in Rust we compute the analytical formula for inverting this matrix to avoid unnecessary operations while minimizing package dependency. Secondly, the code was re-organized in Rust to maintain clarity in the new syntax.

c. *Results of Optimization*

With the CG-matrix caching and proposed porting of the moments above, we have performed benchmark timings (Fig. 7) to compare the speed of execution. These benchmarks were computed on the High-Performance Computers (HPC) within the Wisconsin Center for High-Throughput Computation. Each HPC had a 3GHz AMD EPYC 7763 64-Core Processor and 514 GB Total RAM. Note that none of this code is currently parallelized and therefore only a single core was used, yielding timings that are similar to what is obtained on a local machine.

Appendix B: Gay-Berne Case-Study

1. Gay-Berne Interactions

Continuing our discussion of Eq. (23), the second term, η_{12}, is a function of each ellipsoids' geometry and orientation but not a function of distance, and is given as follows:

$$\eta_{12}(\mathscr{R}_1,\boldsymbol{S}_1,\mathscr{R}_2,\boldsymbol{S}_2) = \left[\frac{2s_1s_2}{\det \boldsymbol{\Xi}_{12}(\mathscr{R}_1,\mathscr{R}_2)]}\right]^{\nu/2} \tag{B1}$$

$$s_i = [a_ib_i + c_ic_i][a_ib_i]^{1/2} \tag{B2}$$

This correction term describes the interaction strength between two ellipsoids at 0 separation, whose influence is tuned by parameter ν.

The last term, χ_{12}, is only a function of each ellipsoid's individual orientations ($\mathscr{R}_1,\mathscr{R}_2$) and relative orientation ($\hat{r_{12}}$), but not of their geometries ($\boldsymbol{S}_1,\boldsymbol{S}_2$). It is given as follows:

$$\chi_{12}(\mathscr{R}_1,\mathscr{R}_2,\hat{r}_{12}) = \left[\hat{r}_{12}^T\boldsymbol{B}_{12}^{-1}(\mathscr{R}_1,\mathscr{R}_2)\hat{r}_{12}\right]^{\mu} \tag{B3}$$

$$\boldsymbol{B}_{12}(\mathscr{R}_1,\mathscr{R}_2) = \mathscr{R}_1^T\boldsymbol{E}_1\mathscr{R}_1 + \mathscr{R}_2^T\boldsymbol{E}_2\mathscr{R}_2 \tag{B4}$$

$$\boldsymbol{E}_i = \begin{pmatrix} e_{ai}^{-1/\mu} & 0 & 0 \\ 0 & e_{bi}^{-1/\mu} & 0 \\ 0 & 0 & e_{ci}^{-1/\mu} \end{pmatrix} \tag{B5}$$

Note that for Gay-Berne, the rotations $\mathscr{R}_i$ transform from lab frame to the body frame, while AniSOAP defines rotations to transform from the body frame to the lab frame. Hence, $\mathscr{R}_{i,AniSOAP} \equiv \mathscr{R}_{i,GB}^{-1}$. The above definitions use $\mathscr{R}_i \equiv \mathscr{R}_{i,GB}$.

χ_{12} corrects for the well-depth by interpolating the relative well depth between the side-to-side, face-to-face, and end-to-end interactions, given by e_{ai}, e_{bi}, e_{ci}. Generally, these pole-pole relative well depths are arbitrarily fitted, but can be assumed to be equal to the Gaussian curvature of the ellipsoids at each pole, provided that $\mu = 1$:

$$\boldsymbol{E}_i = \sigma \begin{pmatrix} \frac{a_i}{b_ic_i} & 0 & 0 \\ 0 & \frac{b_i}{a_ic_i} & 0 \\ 0 & 0 & \frac{c_i}{a_ib_i} \end{pmatrix} \tag{B6}$$

In our case-study with $a_i = 1, b_i = 1.5, c_i = 2$ ellipsoids, we set $\mu = 1$, enabling the use of B6 to calculate $\boldsymbol{E}_i$. We furthermore set $\nu = 1$, completely specifying the Gay-Berne hyperparameters.