RESEARCH ARTICLE | JUNE 24 2024

Molecular dynamics simulations of anisotropic particles accelerated by neural-net predicted interactions

Special Collection: 2024 JCP Emerging Investigators Special Collection

B. Ruşen Argun ; Yu Fu ; Antonia Statt

J. Chem. Phys. 160, 244901 (2024)
https://doi.org/10.1063/5.0206636
CHORUS

The Journal of Chemical Physics ARTICLE pubs.aip.org/aip/jcp

Molecular dynamics simulations of anisotropic particles accelerated by neural-net predicted interactions

Cite as: J. Chem. Phys. 160, 244901 (2024); doi: 10.1063/5.0206636
Submitted: 3 March 2024 • Accepted: 24 May 2024 •
Published Online: 24 June 2024

B. Ruşen Argun,[1] Yu Fu,[2] and Antonia Statt[3,a)]

AFFILIATIONS

[1] Mechanical Engineering, Grainger College of Engineering, University of Illinois, Urbana-Champaign, Champaign, Illinois 61801, USA
[2] Physics, Grainger College of Engineering, University of Illinois, Urbana-Champaign, Champaign, Illinois 61801, USA
[3] Materials Science and Engineering, Grainger College of Engineering, University of Illinois, Urbana-Champaign, Champaign, Illinois 61801, USA

Note: This paper is part of the 2024 JCP Emerging Investigators Special Collection.
[a)] **Author to whom correspondence should be addressed:** statt@illinois.edu

ABSTRACT

Rigid bodies, made of smaller composite beads, are commonly used to simulate anisotropic particles with molecular dynamics or Monte Carlo methods. To accurately represent the particle shape and to obtain smooth and realistic effective pair interactions between two rigid bodies, each body may need to contain hundreds of spherical beads. Given an interacting pair of particles, traditional molecular dynamics methods calculate all the inter-body distances between the beads of the rigid bodies within a certain distance. For a system containing many anisotropic particles, these distance calculations are computationally costly and limit the attainable system size and simulation time. However, the effective interaction between two rigid particles should only depend on the distance between their center of masses and their relative orientation. Therefore, a function capable of directly mapping the center of mass distance and orientation to the interaction energy between the two rigid bodies would completely bypass inter-bead distance calculations. It is challenging to derive such a general function analytically for almost any non-spherical rigid body. In this study, we have trained neural nets, powerful tools to fit nonlinear functions to complex datasets, to achieve this task. The pair configuration (center of mass distance and relative orientation) is taken as an input, and the energy, forces, and torques between two rigid particles are predicted directly. We show that molecular dynamics simulations of cubes and cylinders performed with forces and torques obtained from the gradients of the energy neural-nets quantitatively match traditional simulations that use composite rigid bodies. Both structural quantities and dynamic measures are in agreement, while achieving up to 23 times speedup over traditional molecular dynamics, depending on hardware and system size. The method presented here can, in principle, be applied to any irregular concave or convex shape with any pair interaction, provided that sufficient training data can be obtained.

I. INTRODUCTION

Advances in chemistry have made it possible to synthesize many non-spherical colloidal particles.[1] Those anisotropic colloidal building blocks can self-assemble into various targeted nano-structures with desirable structural, optical, or electronic properties.[2,3] In the environment, pollutants such as microplastics or engineered nanoparticles can occur in different shapes.[4,5] In addition, particles with non-spherical morphologies are central to several biological processes.[6] Thus, capturing anisotropic features of particles is essential to accurately simulate and understand the fundamentals of a wide range of phenomena in science and engineering.

Traditional particle based simulations, such as molecular dynamics (MD) and Monte Carlo (MC) simulations, often treat particles as perfect spheres; however, there are several methods available that can take non-spherical geometric features of particles into account. For athermal systems, where particles interact with hard

repulsion only, calculating the pairwise potential reduces to the task of finding whether the two particles or shapes overlap or not. In such systems, where the self-assembly is driven by entropic effects, powerful methods and analytical solutions to calculate the interactions for many geometries exist.[7–10] Donev[11] introduced an event-driven MD algorithm that can efficiently calculate the overlap between non-spherical geometries. This method has been used to find closed-packing structures and phase behavior of hard superballs.[12,13] De Michele[14] developed an alternative event-driven approach to simulate systems of generic convex hard rigid bodies. Damasceno *et al.*[15] and others[16–18] have simulated a wide range of polyhedra with hard-particle MC and investigated their self-assembly and phase behavior using different methods to calculate the overlap or contact between the shapes.[19] However, for systems where the self-assembly is driven by enthalpic forces (i.e., attractive interactions), different methods are required. The Gay–Berne[20] potential is commonly used to simulate ellipsoids with pairwise attractive interactions, depending on the orientations of the particles in addition to the distance between their center of masses. It is convenient for MD simulations, as forces and torques can be calculated analytically; however, it is limited to ellipsoid shapes only. The anisotropic Lennard-Jones potential[21] extends this to polyhedra.

By using a composite bead approach, *any* particle shape with attractive and hard repulsive interactions can be simulated. Due to its flexibility in both shape and interaction, this method is commonly used. Here, smaller, spherical beads, often called composite beads, are held together with stiff harmonic bonds and angle potentials to keep the geometry of the larger composite shape the same throughout the simulation. Alternatively, rigid body constraints between the composite beads can be applied. Popular MD software applications, such as HOOMD-blue[22–24] and LAMMPS,[25] have implemented rigid body constraints to simulate non-spherical particles or molecules made out of smaller beads.[23,26] The composite bead approach is flexible, as almost any concave or convex shape can be simulated with it. The self-assembly behavior of nanoparticles can be affected by small morphological details.[27] To obtain a precise geometry including sharp corners and edges, the shape often needs to contain many beads. The interaction between two bodies or shapes is now given by the double sum of all pairwise interactions between the composite beads that make up the two bodies. Thus, rigid bodies containing many composite beads significantly increase the computational cost in both coarse-grained and atomistic simulations. A similar computational cost challenge arises, for example, to calculate the total van der Waals (vdW) interaction between two non-spherical particles. Most notably, Lee and Arya[28] devised an analytical method to approximate the vdW interactions between two cubic rigid bodies bypassing the cumbersome double sum. Their method allows for MC simulations of cubes, but it is limited in its application to MD since forces and torques cannot be obtained with it. There are several similar efforts in the literature to calculate the vdW interaction for non-spherical bodies.[29–33] Recent work[34] has calculated vdW forces and torques for cubes with a numerical method. Ramasubramani *et al.*[21] introduced a Lennard-Jones-like potential that can be used for different polyhedra.

Machine Learning (ML) techniques are already applied to spherically symmetric particles, especially to bridge quantum mechanical calculations and classical atomistic MD simulations.[35,36] Some possibilities that ML offers for anisotropic interactions have recently been discovered. Campos-Villalobos *et al.*[37] developed symmetry functions that can be used to construct ML potentials for cylindrically symmetric particles. Anisotropic and rigid molecules like benzene were coarse-grained by matching forces and torques to accelerate all-atom condensed-phase MD simulations using a linear model[38] and neural networks.[39] Recent work[40] has used several ML models to detect whether or not two composite rigid bodies are overlapping, bypassing explicit distance calculations between primary beads of the rigid bodies. However, this method does not allow attractive interactions.

In this work, we use fully connected feed-forward neural nets to predict the total interaction between two example geometries of composite rigid bodies: cubes and cylinders. We focus on forces and torques, since these are the necessary inputs for a MD simulation. Here, we describe our generic method of data sampling and processing and test the neural-net predictions for simple example particle shapes. We show that the MD simulations performed with forces and torques from the trained neural nets accurately reproduce structural and dynamic properties of traditional rigid body simulations, which were performed with explicit distance calculations. Depending on the system simulated (number of particles, temperature, etc.) and hardware available, neural-net assisted simulations are faster than traditional MD simulations. Our method is entirely agnostic to the particle shape and can be, in principle, applied to any irregular composite particle geometry with a wide range of both attractive and repulsive pair interactions.

II. METHODS AND MOTIVATION

To calculate the total potential U between a pair of composite-rigid bodies that are made out of smaller beads (as in Fig. 1) that interact with pairwise potentials $u_{bead}(r)$, the distances between each pair of beads need to be calculated,

$$U = \sum_{i=1}^{N} \sum_{j=1}^{N} u_{bead}(r_{ij}).$$

For two rigid bodies, each consisting of N beads, calculating N^2 distances r_{ij} can be computationally costly for high values of N. For

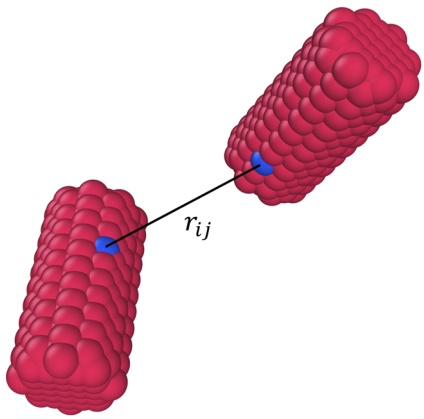

FIG. 1. A pair of cylindrical rigid bodies made out of 639 smaller beads.

example, the shapes investigated here contain 666 and 639 beads for the cube and cylinder, respectively. The cylinder is shown in Fig. 1, and a depiction of the cube can be found in the supplementary material. The number of beads N used to represent a rigid-body depends on several factors, including the geometry of the particle and how accurate one would like to reproduce that geometry. For example, to perform a systematic study of particle shapes that range from a perfect cube to a sphere with several super-ball shapes in between, edges and corners of the cube must be smoothed out gradually as the shape approaches to sphere and this requires a high resolution of beads. Simulation methods that are used to incorporate a background fluid and hydrodynamic interactions such as multi-particle collision dynamics[41] may also require a high resolution of shape surface for accurate coupling between the colloid and solvent particles.[42,43] Another factor is the type of interaction between the shapes. For hard-core repulsion interactions or WCA-like[44] interactions, where there is no attraction, one can place beads at the surface of the shape only and leave the core empty, thereby reducing N and saving computational time. However, this approach may not work for attractive interactions. By only having surface beads, or shapes with a small number of beads, it may not be possible to reproduce the desired effective interaction between the two shapes in such cases. N may need to be on the order of 10^2 to accurately represent the shape and reproduce the desired effective interaction. We refer to the supplementary material for a detailed illustration of this issue.

In the following, we outline an alternative route to reduce computational costs associated with the composite bead method by bypassing explicit distance calculations, while keeping the full flexibility of the different shapes that can be used. Any pair configuration of 3D bodies can be expressed as a function of six numbers. The position and orientation of the first body define a coordinate frame. The position of the second body $\mathbf{x}(x_1, x_2, x_3)$ and its orientation $\mathbf{\Omega}(\Omega_1,\ \Omega_2,\ \Omega_3)$ in the relative coordinate frame of the first body are then sufficient to uniquely define any pair configuration. Thus, the total interaction between the two composite-rigid bodies is essentially a function of six numbers,

$$U = g(\mathbf{x}, \mathbf{\Omega}) = g(x_1, x_2, x_3, \Omega_1, \Omega_2, \Omega_3).$$

However, this function is quite complex and its general analytical solution is not available even for simple shapes such as cubes or non-aligned cylinders.[28] We follow a data-driven approach and use neural nets (g_{NN}) to approximate the function g relating the pair configuration directly to the interaction energy, force, or torque,

$$U = \sum_{i=1}^{N}\sum_{j=1}^{N} u_{bead}(r_{ij}) = g(\mathbf{x}, \mathbf{\Omega}) \approx g_{NN_U}(\mathbf{x}, \mathbf{\Omega}),$$

thus allowing us to skip the costly distance calculations at the expense of the inference and training cost of a neural net. We demonstrate the method on two particle geometries: cylinder (Fig. 1) and cube (Fig. 1 of the supplementary material).

We are using half-power half-cosine as a base function,

$$\mathrm{u_{bead}}(r) = \begin{cases} \varepsilon((b-r)/\sigma)^p - 2A, & r \le b \\ -A\cos\left(c(r-b)\right) - A, & r > b \\ 0, & r > \pi/c + b \end{cases} \quad (1)$$

for the bead-to-bead potential as detailed in Sec. 4 of the supplementary material with the following parameters: $A = 0.000\,35\varepsilon$, $b = 1.463\,75\sigma$, $c = 3.3833\sigma^{-1}$, and $p = 2.5$ for both cylinders and cubes. The cutoff distance is $r_{cut} = \pi/c + b$. When summed up over the beads of the two interacting cubes, we effectively obtain a Lennard-Jones-like interaction between the two cubes as illustrated in Fig. 3. We used the size of the smaller composite beads σ, as a unit of length. Energy is given in units of ε. In the remainder of this work, we use reduced units.

Training and testing data for the neural net were sampled from traditional composite-rigid body MD simulations that were performed at a wide range of temperatures, starting from $T = 0.2\varepsilon/k_B$ up to $2.0\varepsilon/k_B$. Both energetically favorable (i.e., pairs that are strongly attracting, for example two cubes with their faces aligned) and unfavorable (i.e., pairs that are slightly overlapping or touching, physically less likely to occur) pair configurations can be efficiently sampled at low and high temperatures, respectively. Sampled body pairs and their calculated energy, force, and torque vectors are then subjected to geometric transformations and operations in order to express each pair as a function of $\mathbf{\Omega}$ and $\mathbf{x}$ and to make use of the geometrical symmetries of the shapes. This process is explained in detail in Sec. 1 of the supplementary material, and the code is publicly available at https://github.com/stattlab/NNAssistedRigidBodyMD. Here, we only mention that unlike other 3D bodies, a pair of cylinders can be expressed as a function of four numbers only $g(\mathbf{x}(x_1, x_2), \mathbf{\Omega}(\Omega_1,\ \Omega_2))$, thanks to their cylindrical symmetry.

Training a single neural-net that can predict the energy would be sufficient to perform Monte Carlo simulations. For MD simulations, force $\mathbf{f} = (f_1, f_2, f_3)$ and torque $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3)$ vectors are required to integrate the equations of motion forward in time. We have used two different methods to obtain $\mathbf{f}$ and $\boldsymbol{\tau}$. First, for cylinders, five separate neural nets were trained for each component of $\mathbf{f} = (f_1, f_2, f_3)$ and (τ_1, τ_2) and used directly in the simulation. The torque around the axis of the cylinder, τ_3, is not needed. This leads to the following approximations:

$$f_i \approx g_{NN_{f_i}}, \quad \tau_i \approx g_{NN_{\tau_i}} \quad \text{with} \quad i = 1, 2, 3.$$

Second, for cubes of side length ~3.2σ, the forces and torques were calculated from the gradients of the energy neural-net as explained in Sec. 6 of the supplementary material. The second method can provide the same accuracy as the first one, and it is more practical to implement. In addition, using one neural net for energy also helps us ensure that the components of the forces and torques obey energy conversation. The method was not tested for non-thermostatted simulations and should be used with caution in NVE due to possible jumps and kinks in the predicted energy landscape (Fig. 9 of the supplementary material), which may result in violation of overall energy conservation. Simple fully connected feed-forward neural nets of width 60 and depth 6 are used for the regression tasks in the cube system. The details can be found in the supplementary material.

III. RESULTS AND DISCUSSION

Generally, accurate force–torque prediction from our model should result in correct molecular dynamics simulations. Before comparing the traditional and neural-net assisted simulation results, we are presenting a more direct measure of ML accuracy, namely parity plots obtained from the test data. In Fig. 2, the predicted values

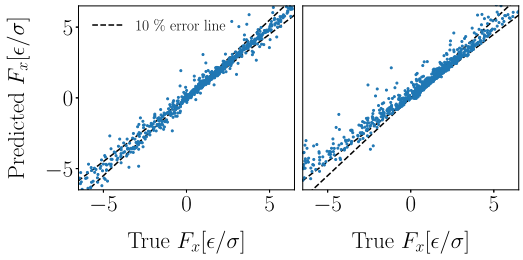

FIG. 2. Parity plot of true vs predicted f_x values for the cylinder (left) and cube (right) systems. For the cylinder, f_x is the direct output of the force neural net g_{NN_f}; for the cube, it is obtained from the gradient of the energy neural net g_{NN_U}. The dashed lines indicate the 10% error line.

of the x-component of the net force (f_x) vs the true value between two cylinders and cubes are plotted. The other force components follow the same trends. For both shapes, data points outside the 10% error lines were frequent, especially for values close to zero. We note that using separate neural networks to predict f_i for cubes, instead of taking the gradient of g_{NN_U}, results in a much lower rate of error, as we show in Sec. 5 of the supplementary material. However, since the accuracy of the MD simulation results is similar for both methods, the results obtained from cube simulations assisted by $g_{NN_{f_i}}$ and $g_{NN_{\tau_i}}$ are not presented here.

In Fig. 3, traditionally calculated and neural-net predicted energies of cubes are compared. Here, we show the energies of three different relative orientations as a function of their center of mass distances. Overall, there is good agreement over a wide range of center of mass distances, with noticeable deviations for small distances and high interaction energies. These short distances are less likely to occur in a molecular dynamics simulation but might play a significant role in simulations under highly dense conditions or temperatures above those used to generate the training dataset.

The next step is to compare the results of neural-net assisted simulations with traditional molecular dynamics simulations. For this, we performed two sets of simulations, one using HOOMD-blue 2.9.4 with traditional molecular dynamics simulations of rigid bodies in the NVT ensemble and the other with the C++ code using neural-net forces and torques for the molecular dynamics simulation.

The temperature was kept constant using a Nosé–Hoover thermostat with a time step of 0.005 τ. We equilibrated the system for 2500τ, where $\tau = \sigma\sqrt{m/\varepsilon}$ is the unit of time. The particle volume fraction $\phi = N * V_{\text{particle}}/V$ of all simulations was set to $\phi = 0.16$. The cubes have an approximate side length of 3.2σ, and the cylinders are $\sim 2.75\sigma$ in diameter and 5.3σ long.

To assess the accuracy of our simulations, a measure for the structure of the system was used, the pair correlation function $g(r)$ of the center of mass of the simulated shapes. In Fig. 4, $g(r)$ of cylinders was plotted for both nn-assisted and traditional MD simulations. While qualitative agreement between the two curves can be observed, and nn-assisted simulations reproduced the peaks at correct positions, the height of the first peak is underestimated. The first peak in $g(r)$ corresponds to the pair configuration where two cylinders are aligned along their main axis and are close to each other. We attribute the mismatch at the first peak to the assumption in nn-assisted simulations that the cylinders are symmetric around their main axis. Even though the cylinders were comprised of many composite beads for the traditional MD simulations, the surface was still rough and rotations around the main axis might at close distances result in different interactions. This roughness is effectively coarse-grained out in the neural-net assisted simulations, since the neural net cannot account for rotations around the main axis by the design of the input variables. This effect is the strongest when two cylinders are closest to each other, almost touching, and decays as the distance increased, as shown in Fig. S6 of the supplementary material.

Next, we tested the structural accuracy of the cube simulations. In Fig. 5, the $g(r)$ curves for the center of mass of cubes are plotted for simulations performed at temperatures of $T = 0.5\varepsilon/k_B$, $0.6\varepsilon/k_B$, and $0.7\varepsilon/k_B$. There is quantitative match at all temperatures simulated. The nn-assisted simulations accurately capture the significant structural changes observed in this temperature window, ranging from a non-aggregated suspension of cubes at high temperatures $T = 0.7\varepsilon/k_B$, down to an aggregated face-centered, cubic structure at low temperatures $T = 0.5\varepsilon/k_B$. Unlike in the case of cylinders, where we ignore rotations around the main axis, rotations in the cube configurations are represented explicitly and we speculate that this is part of the reason for the improved match of the first peak in $g(r)$.

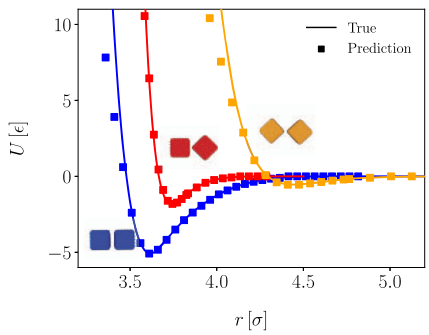

FIG. 3. Interaction energy $U(r)$ as a function of center of mass distance r for three different cube configurations, as shown next to the curves. Each cube had an approximate side length of 3.2σ and is composed of 666 small spherical beads.

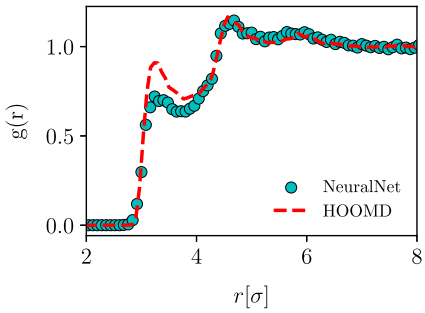

FIG. 4. Pair correlation function $g(r)$ at $T = 1\varepsilon/k_b$ for a cylinder system. The traditional simulation result is shown as dashed line, and the nn-assisted simulation result is shown as filled symbols.

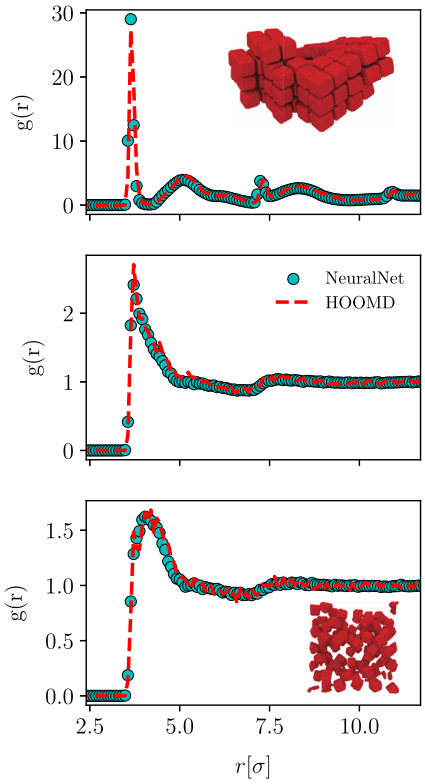

FIG. 5. Pair correlation functions $g(r)$ for three different temperatures: $0.5\varepsilon/k_B$ (top), $0.6\varepsilon/k_B$ (middle), and $0.7\varepsilon/k_B$ (bottom) for cubes. The dashed lines show the result from traditional simulations, and the filled symbols show the result of the nn-assisted simulations. The insets are representative snapshots of the corresponding structures at these temperatures.

We also assessed the accuracy of dynamics in the neural-net assisted simulations. For consistency, the mass of the cube rigid-body is set to be the same in HOOMD and neural-net assisted simulations. In Fig. 6, mean square displacements (MSDs) of cubes at different temperatures for both neural-net assisted and traditional simulations are shown. At all temperatures, good agreement can be found. The estimates for diffusion coefficients are obtained using the slope of MSD vs time data, at longer times. The diffusion coefficients are $0.16\sigma^2/\tau$, $0.54\sigma^2/\tau$, $1.36\sigma^2/\tau$, and $1.88\sigma^2/\tau$ for HOOMD simulations and $0.13\sigma^2/\tau$, $0.33\sigma^2/\tau$, $1.12\sigma^2/\tau$, and $1.58\sigma^2/\tau$ for nn-assisted simulations at temperatures of $0.3\varepsilon/k_B$, $0.5\varepsilon/k_B$, $0.7\varepsilon/k_B$, and $1.0\varepsilon/k_B$, respectively. Even though there is a qualitative match, the diffusion coefficient is smaller in nn-assisted simulations at every temperature simulated. In addition to MSD, mean square rotation (MSR) is also a fundamental measure of the kinetics for non-spherical particles. MSR is the orientational-angular analog of the MSD, and its calculation is explained in detail in Sec. 5 of the supplementary material. In Fig. 6, the MSRs of the cubes at different temperatures are plotted. We observe a reasonable match, except at lower temperatures (0.5 and 0.3). Cubes in the nn-assisted simulations are rotating slightly faster compared to traditional MD simulations. It is not clear why we observe this behavior, and future investigations will be needed.

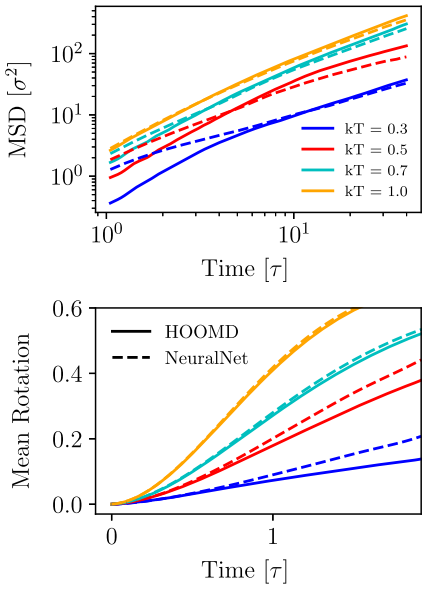

FIG. 6. Mean squared displacement (top) and mean squared rotation (bottom) of cubes at different temperatures. The solid lines represent the traditional simulations, whereas the dashed lines show the nn-assisted results.

Finally, the speed, or performance, of the neural-net assisted simulations was compared with HOOMD-blue simulation performance. Depending on the hardware used and on the number N of cubes simulated, nn-assisted simulations can significantly increase the performance of simulations, as shown in Table I. We performed all simulations on a single RTX 2070 Super GPU with 8 GB memory and a single A100 GPU with 80 GB memory and report the time steps per second. An acceleration factor between the two simulation methods on each GPU was computed. Aside from the smallest system ($N = 500$) on the A100, all nn-assisted simulations are faster than their corresponding HOOMD-blue simulations. In addition, on the RTX 2070, HOOMD rigid-body simulations run out of the 8 GB memory for systems larger than $N \approx 1300$.

TABLE I. Time steps per second and acceleration factor of nn-assisted simulations compared to HOOMD-blue for various system sizes N. The ellipses $(\cdots)$ indicate that GPU memory was not sufficient to run the simulation. Bold lines indicate speedup.

GPU-method \ N	500	1000	1500	2000	3000	4000
RTX-2070—HOOMD	9.5	4.6	⋯	⋯	⋯	⋯
RTX-2070—NN-assisted	143	109	90	75	64	45
RTX-2070—speedup	**x15**	**x23**	⋯	⋯	⋯	⋯
A100—HOOMD	190	90	60	45	30	23
A100—NN-assisted	153	125	108	91	78	60
A100—speedup	**x0.8**	**x1.4**	**x1.8**	**x2.0**	**x2.6**	**x2.6**

IV. CONCLUSIONS

In this work, we present a flexible neural-net assisted method to accelerate molecular dynamics simulations of non-spherical composite rigid bodies of any irregular shape. Here, we have used cubes and cylinders as examples; however, the method has no restrictions on the shape of the particles as long as sufficient training data can be obtained. Our method also allows any pair interactions, including repulsive and attractive forces. We observed good agreement in structural and dynamic properties between the nn-assisted and traditional rigid-body composite simulations over a range of temperatures. We tested the performance of the current method and observed a speedup of x15 to x23 on a single RTX 2070 Super NVIDIA GPU and a speedup of x0.8 to x2.6 on an A100 NVIDIA GPU. Depending on the system of interest and available hardware, this speedup can be significant.

Our method has several shortcomings that prevent further speedup, which can be addressed in future work. Ideally, the descriptors (for our model, these are the input parameters of the neural-net) should be easy to calculate from the raw representation of the system, i.e., positions and quaternions.[37] However, the geometric operations that we perform to consider the symmetries of a pair of cubes or cylinders are computationally quite costly, reducing the performance. A second disadvantage is the use of a neural-net, where inference is associated with significant computational cost, especially considering that inference is needed at every single time step.

We note that cylinders have an advantage where any pair configuration can be expressed with four numbers instead of six due to the cylindrical symmetry. This symmetry greatly improved the performance of the neural-nets or, equivalently, reduces the difficulty of fitting. However, this symmetry assumption led to a slight loss in accuracy, particularly of parallel cylinder configurations at short distances. Depending on the desired density and level of surface roughness (i.e., how coarse-grained the cylinder is), these inaccurate interaction predictions might be significant.

Our method, as presented here, is completely flexible in the input training data. Consequently, it can be easily extended to small nanoparticle systems that include ligands, charges, atomistic interactions, and other effects. Generally, as particle sizes get smaller, the error in effective interactions between particles increases, due to classical approximations and coarse-graining.[45] We speculate that our approach could potentially attain the level of accuracy comparable to atomistic simulations by training the neural nets using nanoparticles made out of atoms. Similarly, for small nanoparticles, surface ligands play a significant role in self-assembly, and neural-nets can be trained with effective forces and torques between nanoparticles with explicit ligands.[37,46] Self-assembly pathways of charged colloids are also highly dependent on morphology.[47] It would be possible to include Coulombic interactions by adding charged beads to the rigid bodies, which would be learned by the neural nets as a part of their effective interactions. Depletion interactions of nanoparticles depend strongly on their shape, and this method could provide an efficient way of developing effective depletion pair interactions by using traditional simulations with explicit depletants.

In this paper, the effective interaction we used was relatively short-ranged. It decayed to zero around $1.4a$, where a is the size of the shape, which is shorter than a typical Lennard-Jones interaction that would decay to zero around 2.5 to $3a$. Especially for nanoparticles, relatively long-ranged interactions can be relevant in self-assembly.[27] Longer-range shape interactions would significantly increase the potential speedup of the nn-assisted method over traditional simulations. Unfortunately, more data sampling and possibly longer training times would be required. Similarly, more irregular shapes may require more composite beads for an accurate representation, which would lead to a larger potential benefit for the nn-assisted method. However, the regression task would be more complicated as well. Future research and methods development will push the limits of the method shown here, and we hope to include different physical interactions, longer-ranged interactions, and more complicated geometries, including concave shapes.

SUPPLEMENTARY MATERIAL

The supplementary material contains the details of the geometric operations performed to take the symmetries of pairs of cubes and cylinders into account. A discussion of the cylindrical symmetry assumption for cylinders is included. An example to demonstrate why many beads can be necessary for a rigid body is also included. We show how one can obtain torques and forces from the energy neural net. The code for running the nn-assisted simulations is available at https://github.com/stattlab/NNAssistedRigidBodyMD.

ACKNOWLEDGMENTS

This work was supported by the funding from the Molecule Maker Lab Institute (MMLI): An AI Research Institutes program supported by NSF under Award No. 2019897. This work made use of the Illinois Campus Cluster, a computing resource that was operated by the Illinois Campus Cluster Program (ICCP) in conjunction with the National Center for Supercomputing Applications (NCSA), which was supported by funds from the University of Illinois at Urbana-Champaign.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

B. Rușen Argun: Data curation (equal); Methodology (equal); Software (equal); Supervision (supporting); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Yu Fu**: Data curation (supporting); Software (supporting); Validation (supporting); Visualization (supporting); Writing – original draft (supporting). **Antonia Statt**: Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Resources (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available within the article and its supplementary material.

REFERENCES

[1] S. Sacanna, M. Korpics, K. Rodriguez, L. Colón-Meléndez, S.-H. Kim, D. J. Pine, and G.-R. Yi, Nat. Commun. **4**, 1688 (2013).
[2] Z. Cai, Z. Li, S. Ravaine, M. He, Y. Song, Y. Yin, H. Zheng, J. Teng, and A. Zhang, Chem. Soc. Rev. **50**, 5898 (2021).
[3] J. Henzie, M. Grünwald, A. Widmer-Cooper, P. L. Geissler, and P. Yang, Nat. Mater. **11**, 131 (2012).
[4] B. R. Argun and A. Statt, Soft Matter **19**, 8081 (2023).
[5] X. Wang, N. Bolan, D. C. Tsang, B. Sarkar, L. Bradney, and Y. Li, J. Hazard. Mater. **402**, 123496 (2021).
[6] C. M. Roth, B. L. Neal, and A. M. Lenhoff, Biophys. J. **70**, 977 (1996).
[7] R. Blaak, D. Frenkel, and B. M. Mulder, J. Chem. Phys. **110**, 11652 (1999).
[8] A. G. Orellana, E. Romani, and C. De Michele, Eur. Phys. J. E **41**, 51 (2018).
[9] A. G. Orellana and C. D. Michele, ACM Trans. Math. Software **46**, 1 (2020).
[10] D. Eberly, *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics* (CRC Press, 2006).
[11] A. Donev, *Jammed Packings of Hard Particles* (Princeton University, 2006).
[12] R. Ni, A. P. Gantapara, J. de Graaf, R. van Roij, and M. Dijkstra, Soft Matter **8**, 8826 (2012).
[13] A. Donev, I. Cisse, D. Sachs, E. A. Variano, F. H. Stillinger, R. Connelly, S. Torquato, and P. M. Chaikin, Science **303**, 990 (2004).
[14] C. De Michele, J. Comput. Phys. **229**, 3276 (2010).
[15] P. F. Damasceno, M. Engel, and S. C. Glotzer, Science **337**, 453 (2012).
[16] J. A. Anderson, M. Eric Irrgang, and S. C. Glotzer, Comput. Phys. Commun. **204**, 21 (2016).
[17] Y. Geng, G. van Anders, P. M. Dodd, J. Dshemuchadse, and S. C. Glotzer, Sci. Adv. **5**, eaaw0514 (2019).
[18] G. van Anders, D. Klotsa, A. S. Karas, P. M. Dodd, and S. C. Glotzer, ACS Nano **9**, 9542 (2015).
[19] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, IEEE J. Rob. Autom. **4**, 193 (1988).
[20] R. Berardi, C. Fava, and C. Zannoni, Chem. Phys. Lett. **297**, 8 (1998).
[21] V. Ramasubramani, T. Vo, J. A. Anderson, and S. C. Glotzer, J. Chem. Phys. **153**, 084106 (2020).
[22] J. A. Anderson, J. Glaser, and S. C. Glotzer, Comput. Mater. Sci. **173**, 109363 (2020).
[23] T. D. Nguyen, C. L. Phillips, J. A. Anderson, and S. C. Glotzer, Comput. Phys. Commun. **182**, 2307 (2011).
[24] J. Glaser, X. Zha, J. A. Anderson, S. C. Glotzer, and A. Travesset, Comput. Mater. Sci. **173**, 109430 (2020).
[25] S. Plimpton, J. Comput. Phys. **117**, 1 (1995).
[26] T. D. Nguyen and S. J. Plimpton, Comput. Phys. Commun. **243**, 12 (2019).
[27] J. Kim, Z. Ou, M. R. Jones, X. Song, and Q. Chen, Nat. Commun. **8**, 761 (2017).
[28] B. H.-j. Lee and G. Arya, Nanoscale Horiz. **5**, 1628 (2020).
[29] A. Priye and W. H. Marlow, J. Phys. D: Appl. Phys. **46**, 425306 (2013).
[30] R. P. Jaiswal and S. P. Beaudoin, Langmuir **28**, 8359 (2012).
[31] Y. Hallez, Colloids Surf., A **414**, 466 (2012).
[32] P. Yang and X. Qian, J. Colloid Interface Sci. **337**, 594 (2009).
[33] H. Maeda and Y. Maeda, Langmuir **31**, 7251 (2015).
[34] J. Lee, E. Nakouzi, J. Heo, B. A. Legg, G. K. Schenter, D. Li, C. Park, H. Ma, and J. Chun, J. Colloid Interface Sci. **652**, 1974 (2023).
[35] V. L. Deringer, M. A. Caro, and G. Csányi, Adv. Mater. **31**, 1902765 (2019).
[36] A. M. Tokita and J. Behler, J. Chem. Phys. **159**, 121501 (2023).
[37] G. Campos-Villalobos, G. Giunta, S. Marín-Aguilar, and M. Dijkstra, J. Chem. Phys. **157**, 024902 (2022).
[38] H. T. Nguyen and D. M. Huang, J. Chem. Phys. **156**, 184118 (2022).
[39] M. O. Wilson and D. M. Huang, J. Chem. Phys. **159**, 024110 (2023).
[40] S. Bag, A. Jha, and F. Müller-Plathe, "Machine learning assisted Monte Carlo simulation: Efficient overlap determination for nonspherical hard bodies," Adv. Theory Simul. **6**, 2300520 (2023).
[41] A. Malevanets and R. Kapral, J. Chem. Phys. **110**, 8605 (1999).
[42] S. Poblete, A. Wysocki, G. Gompper, and R. G. Winkler, Phys. Rev. E **90**, 033314 (2014).
[43] Y. M. Wani, P. G. Kovakas, A. Nikoubashman, and M. P. Howard, J. Chem. Phys. **156**, 024901 (2022).
[44] J. D. Weeks, D. Chandler, and H. C. Andersen, J. Chem. Phys. **54**, 5237 (1971).
[45] C. A. Silvera Batista, R. G. Larson, and N. A. Kotov, Science **350**, 1242477 (2015).
[46] G. Giunta, G. Campos-Villalobos, and M. Dijkstra, ACS Nano **17**, 23391 (2023).
[47] M. Rosenberg, F. Dekker, J. G. Donaldson, A. P. Philipse, and S. S. Kantorovich, Soft Matter **16**, 4451 (2020).