

Machine Learning Assisted Monte Carlo Simulation: Efficient Overlap Determination for Nonspherical Hard Bodies

Saientan Bag,* Ayush Jha, and Florian Müller-Plathe

Standard molecular dynamics (MD) and Monte Carlo (MC) simulations deal with spherical particles. Extending the standard simulation methodologies to the nonspherical objects is non-trivial. To circumvent this problem, nonspherical bodies are often treated as a collection of constituent spherical objects. As the number of these constituent objects becomes large, the computational burden to simulate the system also increases. Here, an alternative way is proposed to simulate nonspherical rigid bodies having pairwise repulsive interactions. This approach is based on a machine learning (ML)-based model, which predicts the overlap between two nonspherical bodies. The ML model is easy to train and the computation cost of its implementation remains independent of the number of constituent spheres used to represent a nonspherical rigid body. When used in MC simulation, this method is faster than the standard implementation, where overlap determination is based on calculating the distance between constituent spheres. This proposed ML-based MC method produces similar structural features (in comparison to the standard implementation) in both two and three dimensions, and can qualitatively capture the isotropic to nematic transition of rigid rods in three dimensions. It is believed that this work is a step toward a time-efficient simulation of non-spherical rigid bodies.

standard MD and MC simulations are mostly pertinent to systems of spherical particles. On the other hand, the simulation of nonspherical particles is relevant in a great variety of contexts, e.g., to study the effect of crowding in various biological processes,^[3] or to understand the self-assembly of nonspherical magnetic particles^[4] for biotechnological applications.

Extending the MD and MC simulations to nonspherical particles is not straightforward. Even for a very simple case of a system of nonspherical particles with pairwise hard repulsion, there exist no simple simulation methodologies. Simulation of such a system requires a method to determine the overlap of two nonspherical particles with arbitrary relative position and orientation. Except for a few standard cases, there is no exact way to determine the overlap between two rigid bodies. For example, for two rigid spherocylinders, an analytical formula^[5] for the overlap determination can be written by considering the minimum distance between two line segments. This problem (overlap


determination for nonspherical bodies) is often circumvented by considering the rigid body as a collection of spherical particles^[6] (or disks in two dimensions) and calculating the overlap between the spherical particles (or disk). However, this method of overlap calculation becomes expensive as the number of constituent spheres (or disks) increases, making the simulation computationally demanding.

In this paper, we propose a data-driven approach to overcome this problem. We trained different Machine Learning (ML) classifiers to determine if there is an overlap between two rigid bodies given their position and orientation. After training, the overlap detection time is independent of the number of constituent spheres (or disks), thus allowing longer simulations of large collections of rigid bodies. We consider rigid, hard bodies of four different 2D shapes (Circle, Triangle, Rod, and Star) and detail the ML model building in two dimensions. We further perform MC simulation with these ML models and compare the obtained structural properties of the systems with the standard MC simulation. As a pilot study, we also extend our approach to three dimensions and demonstrate the isotropic to nematic transition of rigid rods. We believe that this will generate interest in the

1. Introduction

In the last 50 years, we have seen the development of classical molecular simulation techniques, such as Molecular Dynamics (MD) and Monte Carlo (MC) simulation.^[1] These simulation methodologies can quite successfully calculate the structural and dynamical properties of a variety of systems.^[2] However, the

S. Bag, A. Jha, F. Müller-Plathe
Eduard-Zintl-Institut für Anorganische und Physikalische Chemie
Technische Universität Darmstadt
Peter-Grünberg-Str. 8, 64287 Darmstadt, Germany
E-mail: s.bag@theo.chemie.tu-darmstadt.de

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/adts.202300520>

© 2023 The Authors. Advanced Theory and Simulations published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

DOI: 10.1002/adts.202300520

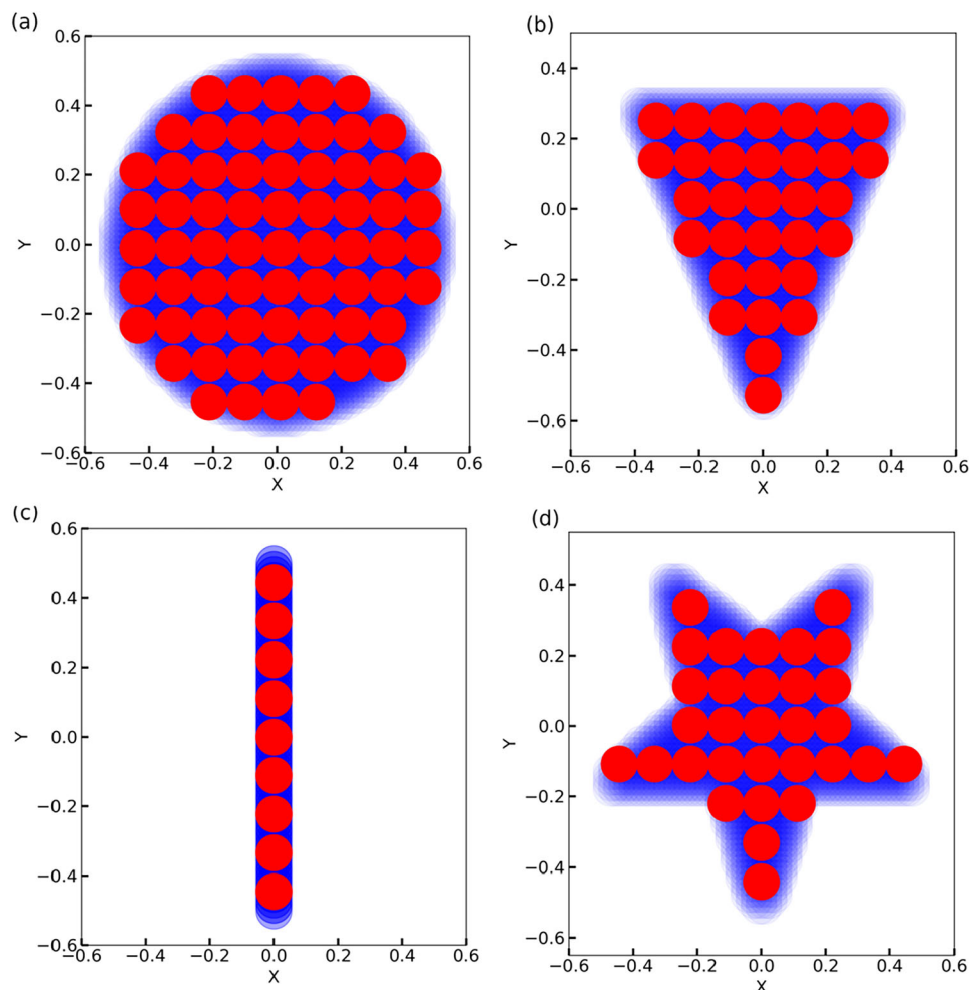


Figure 1. a–d) Rigid bodies (the blue background) of different geometric shapes modeled by a series of small disks (shown in red). The number of disks required to represent the (a) circle, (b) triangle (c) rod, and (d) star is 67, 32, 9, and 31, respectively.

molecular simulation community as an alternative way to simulate collections of nonspherical rigid bodies.

2. Results and Discussion

2.1. In Two Dimension (2-d)

We considered rigid bodies of four different shapes, namely circle, triangle, rod, and star (see **Figure 1**). To determine if there is an overlap between the two bodies, the rigid bodies were modeled by constituent small disks of the same size as shown in **Figure 1**. Different numbers of disks are required to represent bodies of different shapes. As the size of the disk becomes smaller, they trace more accurately the peripheral shape of the rigid body (see **Figure 1**).

To determine if there is an overlap between two bodies, we calculated the distance between all disk pairs with one disk belonging to body A and the other belonging to body B (see **Figure 2a**). If there are N constituent disks for a rigid body, then the overlap determination requires N^2 distance calculations. If any of these N^2 distances are smaller than the diameter of the disk, then we

treated bodies A and B as overlapping. Consequently, for large N , the computation time steeply increases. Therefore, we designed a ML model to predict the overlap between the two bodies. It should be noted that while it is certainly possible to devise a more efficient overlap detection technique with weaker N dependence than N^2 , it will always be dependent on N to some extent. On the other hand, the overlap detection time for the ML model we designed remains independent of N . The input and output of the ML model are described in **Figure 2b** below.

It is worth mentioning that we could use the disks only to represent the periphery of the rigid body (keeping the central part vacant), which would require a lower number of constituent disks, as shown in **Figure S1** (Supporting Information). This kind of construction would perform fine in two dimensions, but in three dimensions, we would actually need the disk in the core of the rigid bodies.

To generate the dataset for the ML model, we randomly kept two rigid bodies at different random positions and orientations and determined the overlap by calculating the distance between the constituent disks representing them (see **Figure 2**). We kept the x and y component of the distance vector (dx , dy) between 0

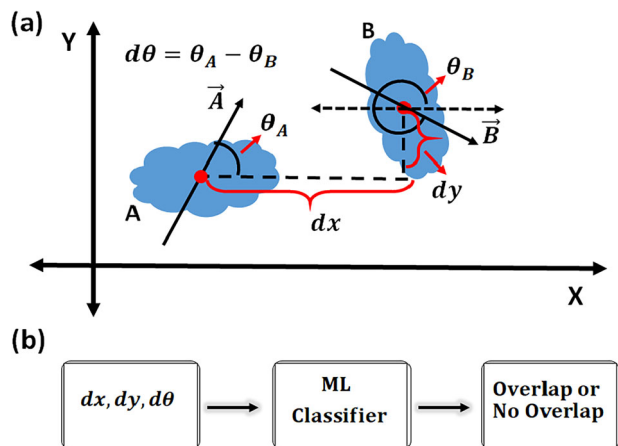


Figure 2. a,b) Schematic diagram describing the input and output of a ML classifier model to predict the overlap between two rigid bodies A and B in two dimensions. a) The position and the orientation of the body are represented by the center of geometry (shown as a red dot) and a vector A attached with the rigid body. As the body moves and rotates the vector A moves and rotates with it. b) The distance between the centers of geometry and the relative angle between these body fixed vectors (A and B) are used to determine whether there is an overlap between them.

and 10 (see Figure 2), with the diameters of the objects being ≈ 1 . The relative orientation ($d\theta$) was chosen between 0 and 2π . We trained various ML classification models which take dx , dy and $d\theta$ as input and predict if there is overlap or not (see Figure 2b). Therefore, this ML task is a binary classification problem. However, the use of this kind of ML classification model is justified only if it accurately predicts the overlap in less computation time. To find out the best possible ML models in terms of computation time, we scanned through a series of ML classifiers, namely 1) Nearest Neighbors Classifier,^[7] 2) A support vector machine (SVM) with Radial Basis Function (RBF) kernel,^[8–10] 3) Adaptive Boosting^[11] (AdaBoost), 4) Random Forest,^[12] 5) Decision Tree,^[13] 6) Quadratic Discriminant Analysis (QDA),^[14] 7) Gradient Boosting Classifier,^[15,16] and 8) Gaussian Naive Bayes.^[17] For extensive details on theory behind the ML classifier models, see Refs.[7–17], for the Python implementation of the ML models, see Ref.[18], and for the exact codes used in this work see our git-hub (https://github.com/saiantan/overlap_non_circular/tree/master) repository.

We generated rigid rods of varying sizes and determined the overlap between them by explicitly calculating the distance of the constituent disks and by training the ML models described in the previous paragraph. All the calculations were repeated 10 times, and their mean is presented in Figure 3, with the error bars being their standard deviations. Figure 3 shows that the computation time for the ML methods remains constant but increases almost quadratically for the case of “Explicit distance calculation”. Four of the eight ML models provide a computationally cheap estimation of “overlap” in comparison to the “Explicit distance calculation” technique. Therefore, we selected these four ML models (Decision Trees, QDA, Gradient Boosting, and Naive Bayes) for further analysis. All the calculations described above were performed in a single CPU of a local desktop computer.

We checked the accuracy of these four ML models in predicting the overlap between rigid bodies of four different shapes (see Figure 1). As presented in Figure 1, the circle, triangle, rod, and star-shaped objects were modeled by 67, 32, 9, and 31 disks, respectively. Using the strategy described previously, we generated 120 000 data for the ML models to be trained for each case. Of the 1 20 000 data, we kept 20 000 data points (test data) to verify the accuracy of the ML model, while the remaining 100 000 data points (training data) were used to train it. To generate the learning curve for the ML models, we increased the number of data points in the training data set (maximum number of available training data points 1 00 000) in steps of 25 and trained a ML model. The trained ML model was used to predict the overlap for the test data points, and the predictions were compared with the actual values to calculate the prediction accuracy.

In both the training and test data sets, we kept an equal number of data of two categories (overlap and no overlap). The learning curves (model prediction accuracy on the test dataset as a function of number of data in the training set) for the four ML models are shown in Figure 4.

Figure 4 shows that with the increase of training data, the ML models initially get better, resulting in an increase in prediction accuracy (in the test dataset). After an optimal number of training data, the accuracy of the ML models stops improving and the learning curve reaches a plateau. The prediction accuracy of the ML models at this stage is the best that we can achieve, and we report this accuracy to be the prediction accuracy of the ML model further in this manuscript.

Among the four ML models studied, the gradient boosting classifier shows the best accuracy for all four types of rigid bodies. The performance of the gradient boosting classifier is the best for the “circle” case with 98% accuracy, which is expected because the ML model can simply learn the distance between the centers of geometry to determine if there is overlap or not. However, as the rigid body becomes more circular, the accuracy of the gradient boosting classifier decreases, with 86% accuracy for the “rod” case (see Figure 4c). The results of the “star” and “triangle” cases lie in between the “rod” case, with prediction accuracy (for the gradient boosting classifier model) values of $\approx 95\%$ and $\approx 92\%$, respectively. The “star,” which is more circularly symmetric than “triangle,” generates a better gradient-boosting classifier model compared to “triangle”. In the circular case, the other three models (Naive Bayes, QDA, and Decision Tree) show similar performance with a maximum accuracy of 92%. In the cases of triangle, rod, and star, the performance of Decision Tree is the worst among all the four ML models, and the performance of QDA and Naive Bayes is between the Gradient boosting (best performing) and decision tree (worst performing). We also trained the ML models and generated the learning curves for the cases where only the outlines of the rigid bodies were represented by the disks (Figure S1, Supporting Information). We found similar learning curves in this case too, as shown in Figure S2 (Supporting Information).

To understand the predictions of the gradient boosting classifier (the best-performing ML model) more deeply, we calculated the confusion matrix (of the best ML model) for the test predictions with 2000 test data points (1000 in each category) shown in Figure 5. In the case of a circle, the gradient boosting classifier categorizes the “Overlap” and “No Overlap” with $\approx 98\%$ accuracy.

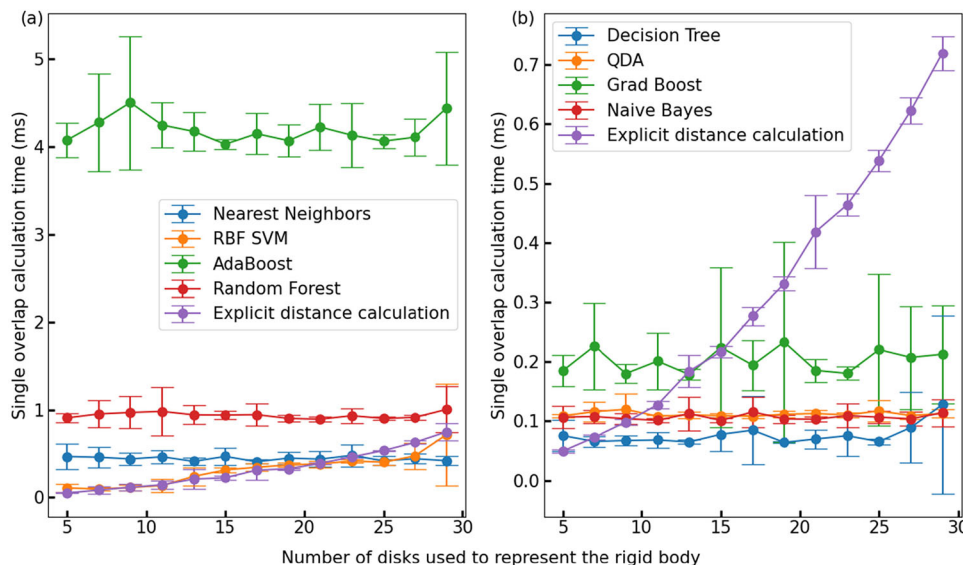


Figure 3. Single overlap calculation time between two rigid bodies as a function of the number of disks used to represent them. Eight different types of ML models and the explicit distance (calculation of the distances between the constituents disks) calculation techniques were used to determine the overlap. The calculation time increases quadratically with the number of disks when the overlap was determined using “explicit distance” calculation while the corresponding times in the case of the ML models remain constant.

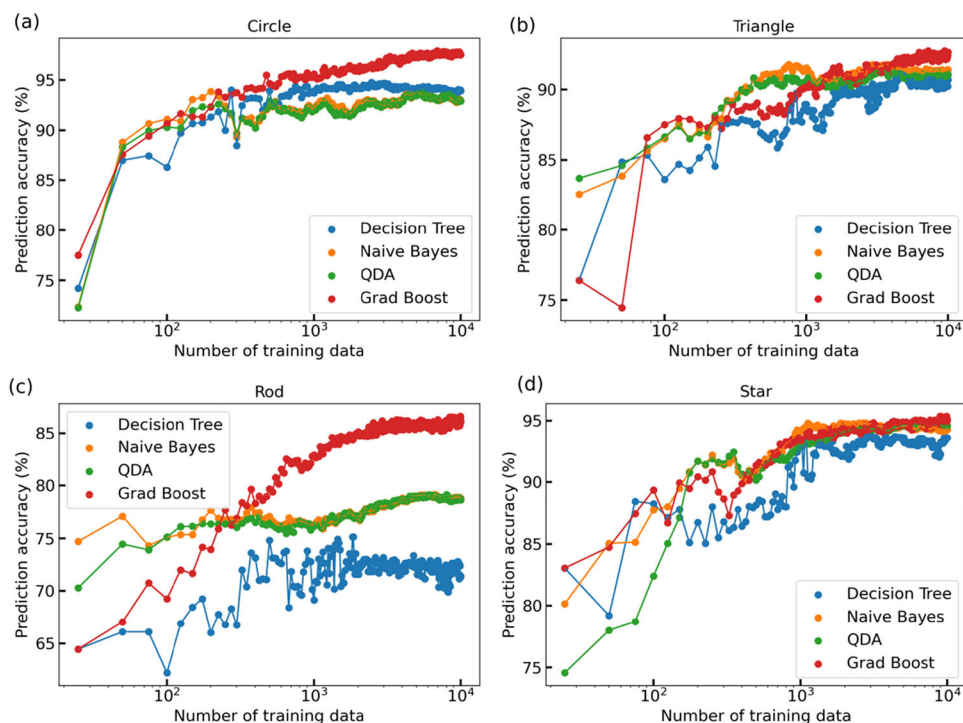


Figure 4. Overlap (between two rigid bodies) prediction accuracy in test dataset as a function of number of data in the training data set for rigid bodies of four different shapes (a) Circle, (b) Triangle, (c) Rod, and (d) Star. The number of disks used to represent each object is shown in Figure 1. The overlap predictions were done using 4 different ML models: Decision Tree, Quadratic Discriminant Analysis (QDA), Naive Bayes classifier, and Gradient Boosting (Grad Boost) classifier. a,b,d) In the case of circles, stars, and triangles all four models reach a maximum accuracy of more than 90% to predict the overlap between two bodies. c) For the rod case the Gradient Boosting performs the best with $\approx 86\%$ maximum accuracy while Decision Tree is worst with only a maximum accuracy of $\approx 72\%$.

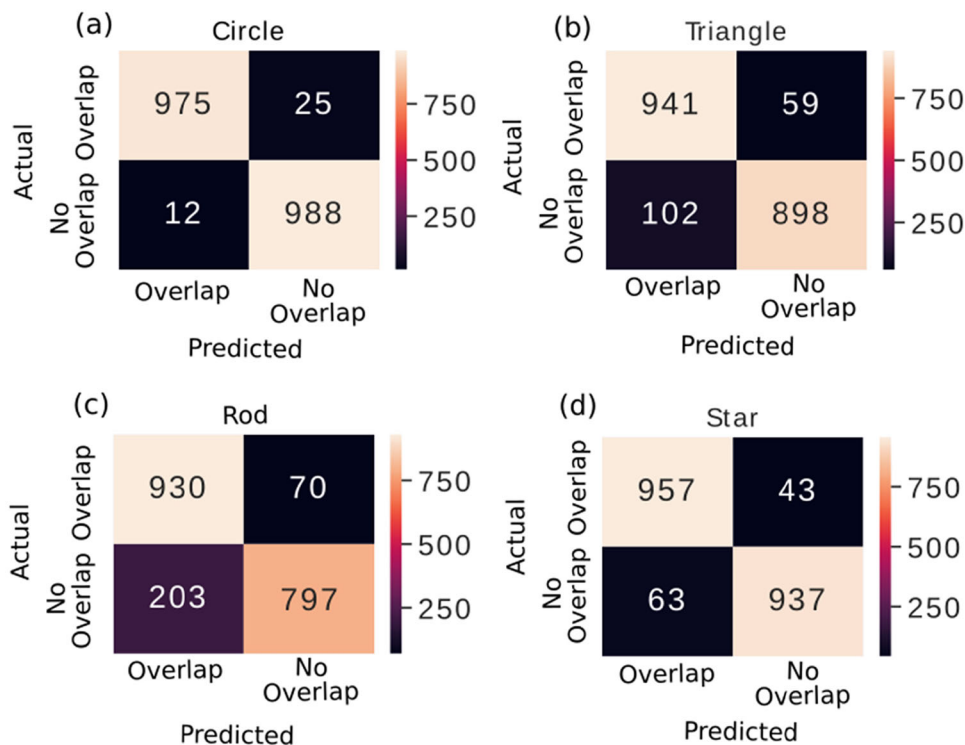


Figure 5. a–d) Confusion matrices for the gradient boosting classifier model in prediction of the 2000 test data points having 1000 points in each category: “Overlap” and “No Overlap”. a) In the case of a circle, the gradient boosting classifier categorizes the “Overlap” and “No Overlap” with $\approx 98\%$ accuracy. In this case, the classifier showed slightly better performance in predicting “No Overlap” in comparison to “Overlap”. b–d) In all other three cases, “Overlap” is predicted better than “No overlap” which is more often wrongly predicted as “Overlap”.

The gradient boosting classifier performs slightly better at predicting “No Overlap” in comparison to “Overlap”. In all the other three cases (triangle, rod, and star), the gradient boosting classifier model mistakenly identifies “No Overlap” to be “Overlap” more frequently than “Overlap” as being “No Overlap”. Thus, the gradient boosting classifier model tends to overpredict “overlap”, with inaccuracy increasing as the object deviates from the circular shape: circle < star < triangle < rod.

So far, we have established that ML classifiers can be trained to accurately detect the overlap between two nonspherical rigid bodies and that these ML models can also be computationally efficient. However, it is still not clear what the implications of these ML models are in actual simulations where one needs to perform many overlap calculations between many pairs. To estimate the effect of the ML models in an actual simulation in terms of computational time and in predicting the structure of the system, we performed MC simulations with the following details. We took a system of 64 rigid bodies of the same type and randomly arranged them in a square simulation box. The system was prepared with an area fraction of 0.20, which corresponds to a number density (n) of 0.31, 0.64, 0.66, and 2.92 for circle, triangle, star, and rod, respectively. The simulation box was periodic in both x and y directions. During the random arrangement, we made sure that the rigid bodies did not overlap. We randomly picked a rigid body and proposed a MC move. A MC move is a combination of translation (of amount tx and ty in x and y direction, respectively) and a rotation (of amount $r\theta$) of the rigid body. Here, tx and ty were chosen to be a random number between -0.4 and $+0.4$, while $r\theta$

is randomly chosen between -10° and 10° . If the rigid body’s new position and orientation overlapped with the other rigid bodies, then we discarded this move, keeping the body at its previous position and orientation. In the case of “No overlap”, the move was accepted. We repeated this move and performed a simulation of 10^5 MC steps. MC simulations were done by using the ML models to determine the overlap. As a reference, we also performed MC simulations by explicitly calculating the distance between the constituent disks to determine overlap. We repeated the above MC simulations for all four different types of rigid bodies and compared the simulation time in **Table 1**.

As expected, the simulation time for the ML models remains almost unchanged for the four shapes, while in the case of the reference calculation, it largely varies as the total number of disks (used to represent the rigid body) changes. In the case of “circle”, ML classifiers allow MC simulations to be ≈ 25 times faster. In the case of “star” and “triangle”, the ML model-guided MC is roughly six times faster. In the rod case, they are of the same order, but it is beneficial to use “Decision Trees”, “QDA”, and “Naive Bayes” as ML models when considering the computational time. Now, it is quite clear that using the ML-based overlap determination provides a benefit in terms of computational time. It is important to note that we could speed up the MC simulation by constructing a neighbor and a cell list. In general, this would, however, speed up both the ML-assisted MC and the standard MC by the same factor.

We further checked how comparable the MC-generated structures of the systems are in these two cases: ML-based overlap

Table 1. Comparison of time taken on a single CPU to perform 10^5 MC steps with 64 rigid bodies. All calculations were repeated 10 times and the average of these 10 calculation times are presented here with the standard deviation of these 10 values as the error bar.

Rigid Body Type	Total number of disks required to represent the 64 rigid body	Time taken (single CPU) for 10^5 MC steps: Explicit Distance Calculation	Time taken (single CPU) for 10^5 MC steps: ML models			
			Decision Trees	QDA	Naive Bayes	Gradient Boosting
Circle	4288	24 945 ± 828 s	389 ± 86 s	636 ± 126 s	650 ± 170 s	1144 ± 210 s
Triangle	2048	6058 ± 160 s	475 ± 86 s	655 ± 118 s	771 ± 148 s	1173 ± 236 s
Rod	576	708 ± 34 s	460 ± 79 s	685 ± 114 s	627 ± 78 s	1288 ± 216 s
Star	1984	5717 ± 196 s	486 ± 71 s	695 ± 135 s	782 ± 163 s	1309 ± 235 s

determination versus standard calculation deciding the overlap. The comparison is presented in **Figure 6**. All the calculations were repeated 10 times, and their mean is presented in Figure 6, with the error bars being their standard deviations. Here, we only show the results with the Gradient Boosting classifier as a ML model. The results with all four computationally cheap ML models are shown in Figure S3 (Supporting Information). From the generated MC trajectory, we calculated the pair correlation function^[19] $g(r)$, defined as

$$g(r) = \frac{1}{2\pi r N n} \left\langle \sum_i^N \sum_j^N \delta(r - r_{ij}) \right\rangle \quad (1)$$

Here, r_{ij} is the distance between the center of geometry of two rigid bodies i and j , N is the total number of rigid bodies, and n is the number density of the system.

We also show simulation snapshots (see bottom panels of Figure 6a–d) after the 10^5 MC steps in both cases. In the circular case, the gradient boosting classifier is quite accurate (see Figures 4a and 5a) at predicting the overlap. Therefore, the ML-assisted MC simulation yields very similar $g(r)$ (see Figure 6a) and the equilibrated snapshots as the reference MC simulation. In the triangular case, the gradient boosting classifier predicts the overlap with 92% accuracy (see Figure 4b), which allows the particles to overlap a little (snapshots in Figure 6b). This overlap causes the amplification of the first peak in $g(r)$ for the ML case. In the rod case, the imperfect (86% accuracy) ML classifier causes the rod to frequently overlap, as seen in the snapshots (highlighted in green in Figure 6c). With the star-shaped particle, we observe the amplification of the first peak of $g(r)$ for the ML case (in comparison to the reference calculation), which is very similar to what we see (Figure 4b) with the triangle shaped particle. This similarity is expected because the accuracy (see Figure 4) of the ML classifier is very similar for the star and triangle-shaped particles. When compared with the star and triangle, we find that the rod generates better $g(r)$ (Figure 6) in spite of having the worst (among the four shapes studied) ML classifier. This might be because the rod is relatively more circular compared to the other four shapes, potentially leading to a reduced influence of overlap on $g(r)$ in this scenario. It is worth noting that we performed the MC simulations and evaluated the pair correlation functions at a fixed area fraction of 0.2. To verify the effects of particle density on the pair correlation function, we also repeated the MC simulation with another area fraction of 0.09 for the rod case where the ML model performed the worst. We find that at this area fraction

too the pair correlation functions match well with the reference calculation as shown in Figure S4 (Supporting Information).

The pair-correlation functions computed in the previous paragraph did not have any information regarding the orientation of the rigid bodies. To see how the orientation of the bodies is correlated, we further evaluated the orientational correlation between two rigid bodies in the systems. We first identified all the pairs of rigid bodies that are separated (distance between the center of geometry) by distance r and then calculated Pearson's correlation coefficient ($R(r)$) between the orientational angles of the pairs.

We further averaged the $R(r)$ values over the snapshots generated from the MC simulations. All the calculations (MC simulations and computations of $R(r)$ with the MC generated snapshots) were repeated 10 times, and their mean is presented in **Figure 7**, with the error bars being their standard deviations.

The vanishing $R(r)$ values in the case of a circle (see Figure 7a), triangle (see Figure 7b), and star (see Figure 7d) indicate an absence of any correlation between particle orientations, irrespective of their proximity. In contrast, when examining rod-like particles, a positive correlation emerges, evident by the positive R values, but only when they are within a distance of 0.5 between their center of geometry. This positive correlation arises because closely positioned rod pairs tend to maintain parallel alignment, as illustrated in Figure S5 (Supporting Information). It is important to note that this alignment constraint does not apply to other shapes with more circular characteristics. Notably, the angle-angle correlation derived from the ML-assisted MC simulation coincides with the findings from the standard MC simulation across all four cases.

2.2. Extension to Three Dimensions (3-d)

We generalized our approach in three dimensions (3-d). To uniquely represent the orientation of a rigid body in 3-d, we used the three Euler angles (α , β , and γ) of the principal axis (corresponding to the largest eigenvalues of the gyration tensor) of the rigid body with respect to three mutually orthogonal directions (x , y , and z) of the Cartesian coordinate system. The position of the rigid body was represented by its center of geometry. The rigid bodies were modeled with several constituent small spheres of the same size. To determine the overlap between the two bodies, we followed the same protocol as the 2-d cases, replacing the role of the disks with spheres. As above, we generated the dataset for the ML models (to predict the overlap) by keeping two rigid bodies at different random relative positions and orientations and

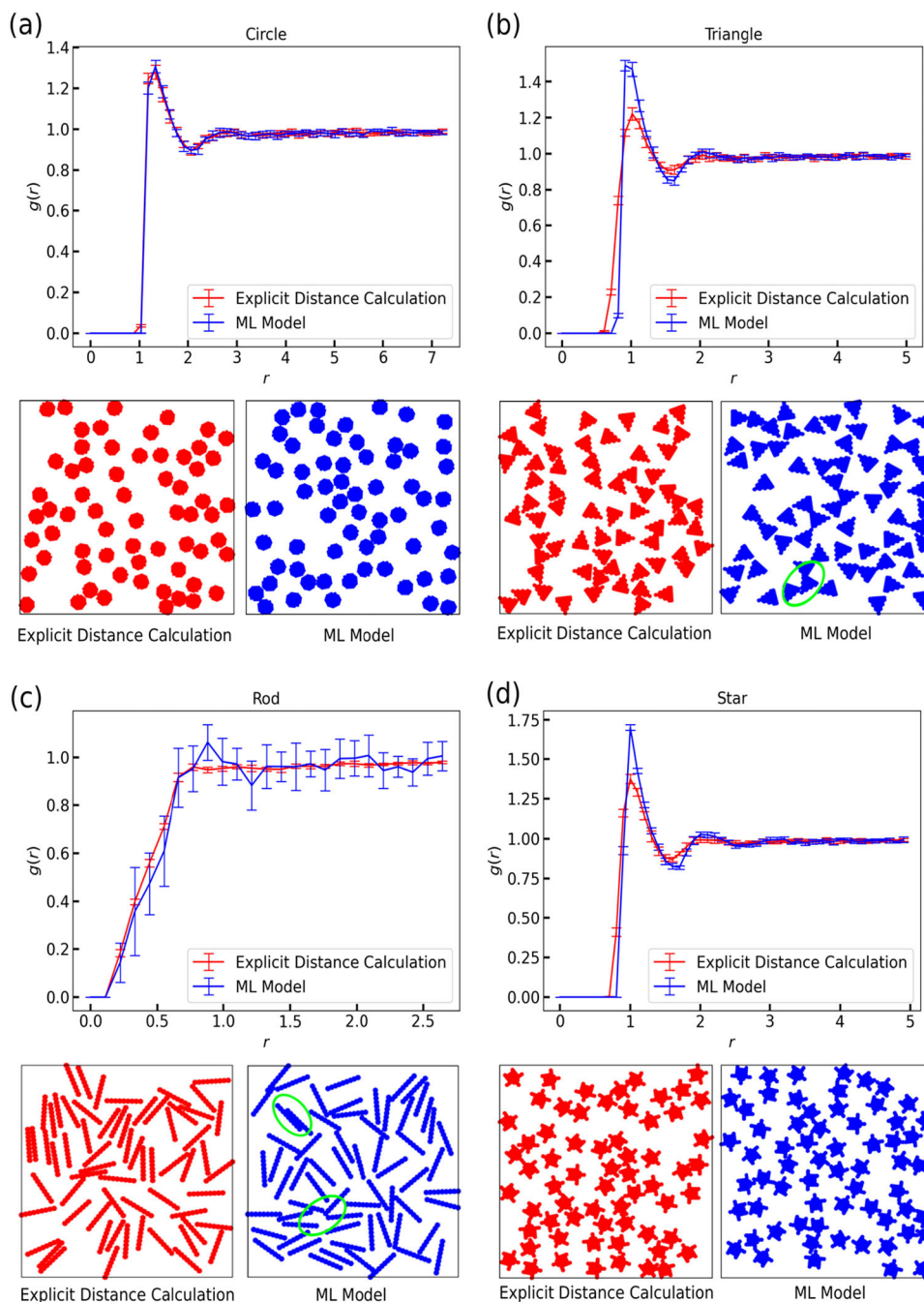


Figure 6. a–d) Pair correlation function ($g(r)$, top panels) and the equilibrated simulation snapshots (bottom panels) obtained from 10^5 steps of MC Simulation with 64 rigid bodies. MC simulations were performed alternatively using the ML model (Gradient Boosting classifier) for overlap determination or using explicit distance calculation. a) In the case of the circle, the two types of MC simulations (ML assisted and standard) generated identical $g(r)$ and equilibrated simulation snapshots because of the very accurate ML classifier. b) In the case of the triangle, the ML model allows the particle to overlap a little, which is reflected in the simulation snapshot (highlighted in green) as well as the first peak of $g(r)$ gets amplified for the ML case. c) The rods frequently overlap (highlighted in green) as seen in the snapshots because of the imperfect ML model. Although the resultant $g(r)$ are very similar. d) The snapshots and $g(r)$ for the star-shaped particle show similar behavior as the triangle-shaped particle.

determined the overlap by calculating the distance between the constituent spheres. The relative position (dx , dy , and dz) between the two bodies was kept between 0 and 10 in x , y , and z direction, respectively. The relative orientation was described by the difference ($d\alpha$, $d\beta$, and $d\gamma$) in Euler angles of the princi-

pal axis of two bodies. For the training data generation, $d\alpha$, $d\beta$, and $d\gamma$ were randomly chosen between 0 and 2π , 0 and π , and 0 and 2π , respectively. ML classification models were trained to predict the overlap between two bodies with their relative position (dx , dy , and $d\gamma$) and orientation ($d\alpha$, $d\beta$, and $d\gamma$) as input

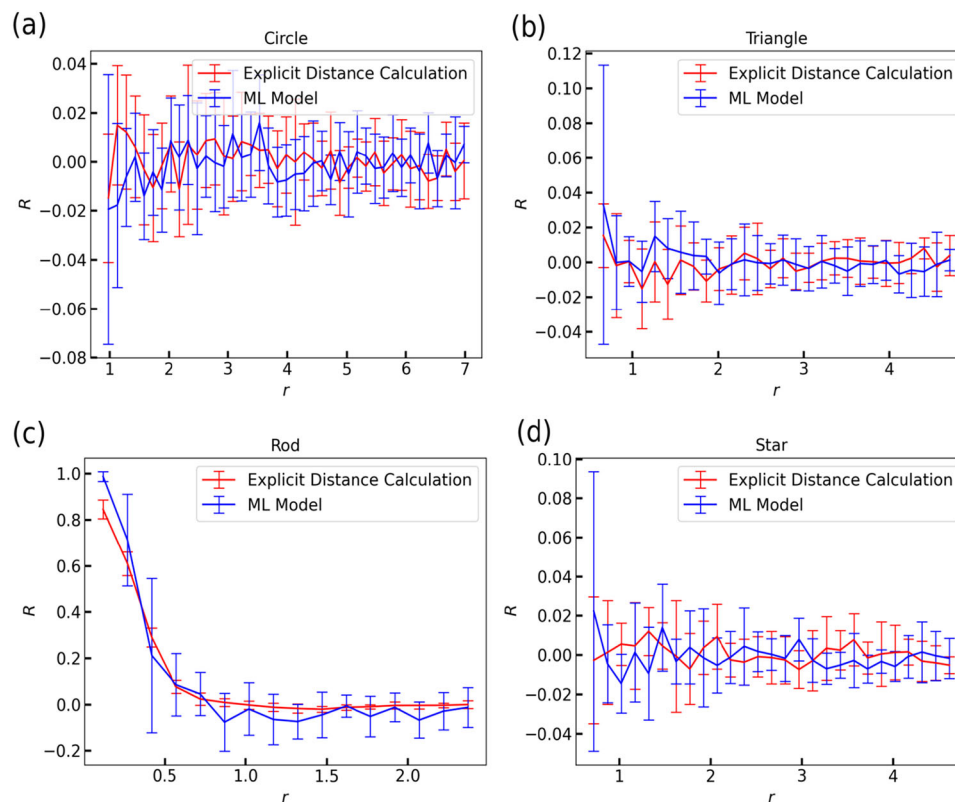


Figure 7. a–d) Pearson’s correlation coefficient ($R(r)$) between the orientational angle of the rigid bodies as a function of the distance between their center of geometries.

to the ML models. To demonstrate this ML-based overlap calculation technique in 3-d, we considered, as an example, rods that are easy to model as a collection of spheres. In addition, rigid rod models are relevant in studying various physical systems of interest, like rod-like virus particles,^[20] long liquid crystalline polymer chains,^[21] etc. As in the 2-d case, the rigid rods were modeled by nine constituted spheres (see Figure 1c). Therefore, the length (L) to diameter (D) ratio for the rods was eight.

We trained four different ML models (decision tree, Gaussian naive Bayes, quadratic discriminant analysis, gradient boosting) for the overlap of rigid rods and then used those trained models to perform MC simulations. The learning curves for the ML classifiers are shown in Figure 8.

As shown in Figure 8, the QDA, Naive Bayes classifier, and Gradient Boosting classifier show very similar performance with a maximum accuracy of $\approx 75\%$, while Decision Tree performs slightly worse than (maximum accuracy of 73%) the other three. A comparison of the learning curves (see Figure 4c) of the 2-d cases reveals that the performance of the best ML classifier (77% accuracy) in the 3-d case is worse than that of the best classifier obtained (85% accuracy) in the 2-d case. This result is unsurprising given that in 3-d cases, rods have more degrees of freedom, making the task of the ML classifier more difficult.

In order to study the liquid crystalline behavior, we then arranged 64 rigid rods in a 3-d simulation box in the isotropic phase with a number density of 0.25 which corresponds to a volume fraction (ρ) of 0.001 and a reduced density (ρ^*) of 0.012. The volume fraction (ρ) is defined as the ratio of the volume occu-

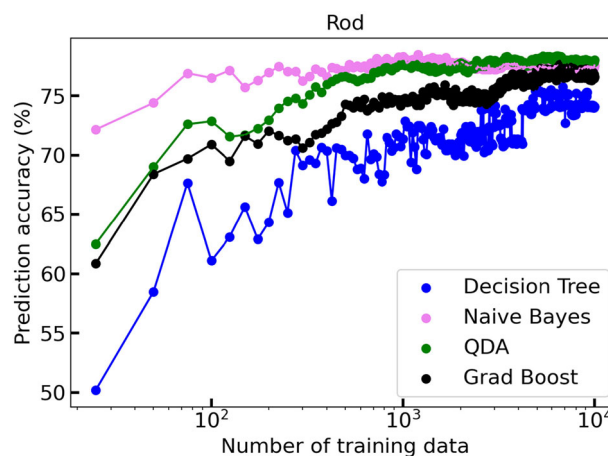


Figure 8. Overlap (between two rigid rods) prediction (by the ML classifiers) accuracy in the test dataset as a function of the number of data in the training data set. Four different ML models, namely, Decision Tree, Quadratic Discriminant Analysis, Naive Bayes classifier, and Gradient Boosting classifier were used to determine the overlap.

ried by the rods (modeled as spherocylinders constructed by a string of 9 nonoverlapping spheres) to the total volume of the system. The reduced density (ρ^*) of the system is defined^[22] as $\rho^* = \frac{\rho}{\rho_{cp}}$. Here, ρ_{cp} is the density of regular close-packing spherocylinders (with Length L and diameter D), which can be calculated as:

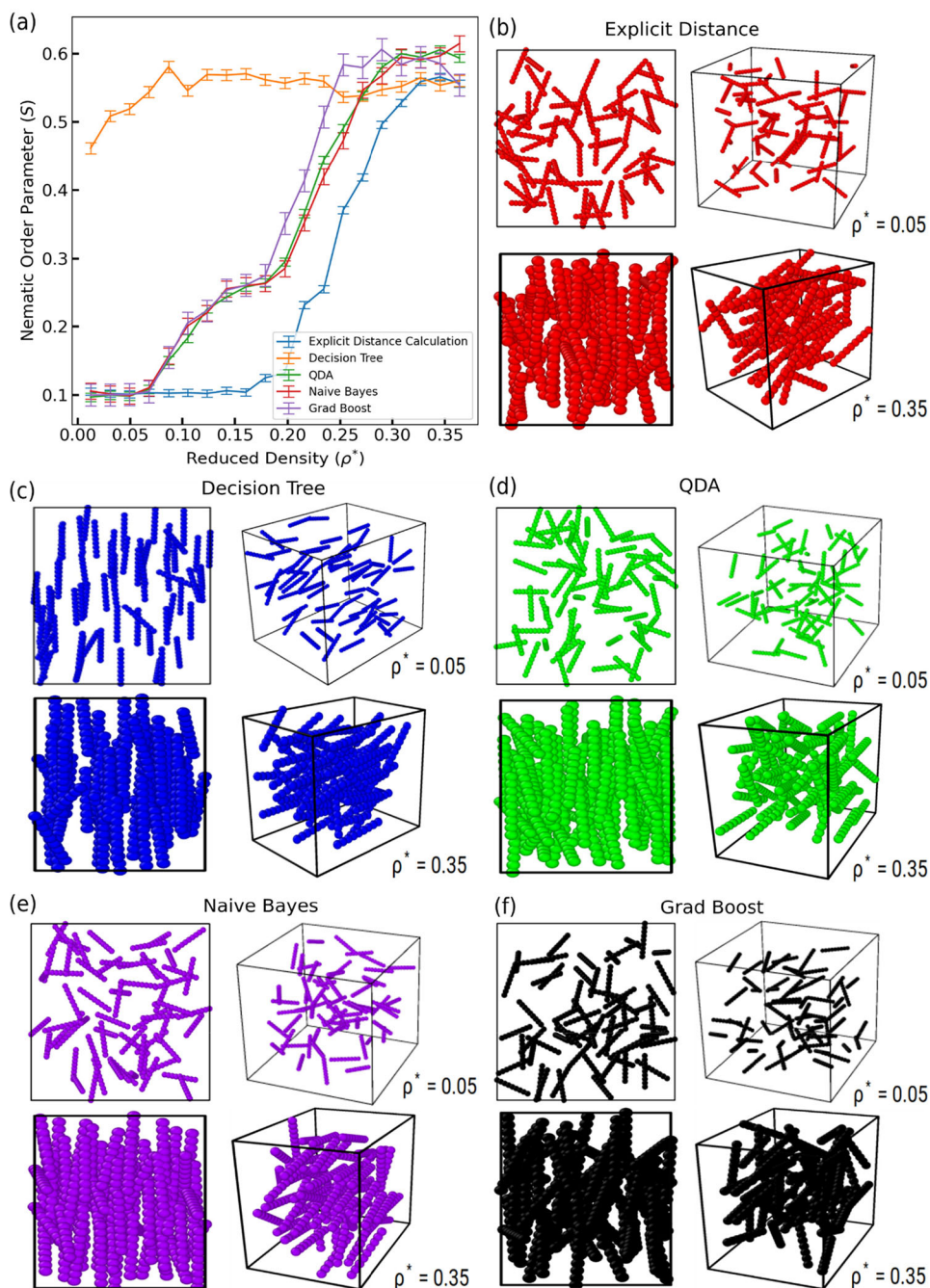


Figure 9. a) Nematic order parameter for a system of 64 rigid rods as a function of reduced density. The nematic order parameters were calculated using the conformations generated by Monte Carlo (MC) simulation. Four Different machine learning (ML) models were used to calculate the overlap between the rigid bodies during the MC simulations. Results are compared with the standard MC implementation where the distances between the constituted spheres of the rods were explicitly calculated to determine the overlap (between the rods). b–f) Simulation snapshots at two different reduced densities (0.05 and 0.35) obtained from the MC simulation with explicit distance calculation and four different ML models: Decision Tree, Quadratic Discriminant Analysis (QDA), Naive Bayes, and Gradient Boost. Both the top (left) and side view (right) of the simulation boxes are shown.

$\rho_{cp} = \frac{2}{\sqrt{2+(L/D)\sqrt{3}}}$. To perform MC simulation with this system we randomly chose a rigid body and proposed its MC move. A MC move is a combination of translation (of amount t_x , t_y , and t_z in x , y , and z direction, respectively) and a rotation. The rotational move is a combination of random change in the Euler angles α

and γ (of amount $r\alpha$, $r\gamma$) along with a random change in $\cos(\beta)$ (of amount $r\cos(\beta)$) rather than in β itself. The MC moves were designed to obey the detailed balance condition as prescribed by Allen and Tildesley.^[23] This MC move was accepted if this new position and orientation of the body did not overlap with the other rigid bodies. Otherwise, the MC move was discarded, and

Table 2. Comparison of CPU time taken to perform 10^5 MC steps with 64 rigid rods in 3-d.

Time taken (single CPU) for 10^5 MC steps: Explicit Distance Calculation	Time taken (single CPU) for 10^5 MC steps: ML models			
	Decision Trees	QDA	Naive Bayes	Gradient Boosting
1034	357 s	1119 s	565 s	1028 s

the body remained at its previous position and orientation. Here, tx , ty , and tz were chosen to be a random number between $-dt$ and $+dt$, while ra and ry were randomly chosen between $-\delta$ and $+\delta$ degrees and $rcos(\beta)$ was randomly chosen between $-dcos(\beta)$ and $+dcos(\beta)$. To determine the optimal values of the dt , δ and $dcos(\beta)$, we started with high values of 0.5, 70, and 0.34 for dt , δ and $dcos(\beta)$, respectively, and measured the acceptance percentage of the proposed MC move. We continued to decrease the values of, δ and $dcos(\beta)$ until the acceptance exceeded 50%. With this optimal value of, δ , and $dcos(\beta)$, we performed MC simulations of 10^5 steps. After 10^5 MC steps, the simulation box was compressed equally in all three directions such that reduced density increased by 0.018. With this system, we again performed a MC simulation of 10^5 steps. We repeated this consecutive box compression and MC simulation 20 times up to a reduced density of 0.35 and measured the structural order of the rigid rods in the system. To characterize the orientational order of the rods in the system, we calculated the nematic order parameter^[22,24–27] defined by $S = \langle \frac{3}{2} \cos^2 \theta - \frac{1}{2} \rangle$. Here, θ is the angle between the director of the nematic phase and the long axis of the rod. The direction of the director was defined as the average direction of the long axes of the rods in the nematic liquid crystalline phase observed at reduced density (ρ^*) of greater than 0.3. The “ $\langle \rangle$ ” symbol was used to indicate ensemble average. The complete phase diagram of the hard spherocylinder as a function of the shape anisotropy (ratio of length (L) and diameter (D)) has been previously mapped by Bolhuis et al.^[22] using MC simulations. For a L/D ratio of > 3.7 , an isotropic to nematic transition of the spherocylinders was reported as the system was compressed. Our system, having an L/D ratio of 8, should exhibit this transition. To capture this transition, we calculated the nematic order parameter of the system as we compressed it. The entire simulation routine described above was repeated using different ML models to calculate the overlap between the rods during the MC simulation. The results from all these simulations are summarized (see **Figure 9**) and compared with the results from the standard MC simulation, where the overlap calculations were done by explicitly calculating the distance between the constituent spheres of the rigid body.

In the case of standard MC simulation for a low reduced density, the rods are found to be in an isotropic phase with a nematic order parameter of ≈ 0.1 . The order parameter starts increasing at a reduced density of ≈ 0.2 , indicating the onset of the transition to the nematic phase. At the reduced density of ≈ 0.3 , the nematic order parameter reaches ≈ 0.5 , indicating the end of the isotropic to nematic transition. The behavior of the order parameter as a function of the reduced density is qualitatively similar to that found by Bolhuis et al.^[22] and others,^[24,25,28–30] who find the isotropic to nematic transition at reduced densities between ≈ 0.5 and ≈ 0.6 . However, instead of a very sharp transition (indicative of first-order phase transition) to the nematic phase, we find a rather gradual transition to the nematic phase. This grad-

ual transition may be a result of the very small system size (64 rods) studied here. Since our main goal was to compare the ML classifier to the standard calculation, we did not find it necessary to simulate a larger system at present. All ML models except the “Decision Tree” show the isotropic to nematic transition, but the onset and end of transition differed from the standard calculation. The QDA, Grad Boost, and Naive bias classifiers predict the onset of the transition early and end somewhat late, making the transition less sharp in comparison to the actual calculation. They all show an artefact around the transition density, namely a shoulder-like profile instead of an uninterrupted increase. This deviation is understandable because these three ML models have similar accuracy (see **Figure 8**) in predicting the overlap. The simulation snapshots at different reduced densities from the MC simulations are presented in **Figure 9b–f**. In the case of Decision Tree, where isotropic-nematic transitions are not seen shows that rods are aligned (see **Figure 9c**) with each other even at a reduced density of 0.05. The snapshots for Standard MC (see **Figure 9b**), QDA (see **Figure 9d**), Naive bias (**Figure 9e**), and Grad Boost (see **Figure 9f**) show that at a reduced density of 0.05, the rods are more isotropically arranged while at a reduced density of 0.35, we see the alignment of the rod indicative of Nematic phase.

We again estimated the simulation time (see **Table 2**) for the 10^5 MC steps with overlap (between two rigid bodies) calculated using different ML models and compared the simulation time of standard MC implementation.

Table 2 shows that the Naive Bayes and Decision Tree provide cheaper simulation time in comparison to standard implementation. Based on the preceding discussion, we conclude that the performance of the ML classifier for the rod case is not very inspiring, both in terms of computation time and prediction accuracy. Additionally, the ML classifiers performed the worst for the rod case in 2-d (see **Figures 4** and **5**), while it was relatively better for the other shapes. In the future, it will be interesting to see how the ML classifiers work for other 3-d shapes.

It is very important to note that the CPU times reported in this manuscript strongly depend on the software library used for the ML models as well as the hardware used to run the codes. The time comparison might change in case the ML model was hardwired in C or Fortran, rather than using very general Python libraries. Therefore, we have mentioned complete details of hardware and software libraries in Supporting Information.

3. Conclusion

Simulating nonspherical particles is relevant in a variety of physical situations to predict the correct structural and dynamic properties of a system. In the context of coarse-grained molecular dynamics simulation, the shape anisotropy of the particles is sometimes captured by considering ellipsoidal beads with the anisotropic Gay–Berne^[31] potential or Kern-Fenkel^[32] potential

between them. For more complex shapes, representing the non-spherical bodies as a collection of spheres is the state of the art.^[6] As mentioned earlier, one of the pitfalls of this method is the higher computational costs due to the increase of total number of degrees of freedom in the system. This has been addressed in the past by designing parallel and scalable algorithms that are already available in the standard MD package like LAMMPS.^[6] On the contrary, our ML model (after training) works with the nonspherical particles themselves, causing no increase in the degrees of freedom of the system. As a proof of concept, we developed various ML models (to detect overlap between two non-spherical bodies) for hard repulsive interaction between the rigid bodies and used those in MC simulations to calculate the structural properties of the systems. We scanned through a series of ML models and found that the Gradient Boosting Classifier is generally the best to determine the overlap in two dimensions. Moreover, within the limited set of nonspherical shapes studied in this paper, we find that our ML-based MC method, reproduces the structural features generated from the traditional approach well. So far, we have not attempted to parallelize our codes in different hardware architectures. Therefore, using our ML approach along with the most efficient parallelization algorithms available in the literature^[6] seems to be the best choice at the moment to simulate nonspherical rigid bodies. We have also conducted a test in three dimensions on fluids of rod-like spherocylinders. Here, Naive Bayes is performing best in reproducing the nematic order parameter. Speedwise, the ML models in three dimensions range from equal to the standard reference MC calculation to three times as fast.

In the granular matter physics community, there has also been interest in the simulation of non-spherical particles in the presence of friction. In this context, Discrete Element Method (DEM) simulation^[33–36] is commonly done by modeling the non-spherical particles as a collection of spherical beads or as a polygon mesh. To make the DEM simulation computationally more affordable, Spellings et al.^[37] have previously designed a GPU-accelerated version of the methods. Our ML model is still not useful in this context since it only predicts the overlap between the particles. In the future, we are planning to extend our ML model to predict the amount of overlap, which can be useful to perform DEM simulations with friction. Our ML models can be straightforwardly generalized by converting the classification problem to a regression one where the output of the ML model is the amount of overlap.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

S.B. acknowledges Danna De Boer for careful reading of the manuscript. S.B. also acknowledges financial support from TU Darmstadt in terms of the “Career Bridging Grant” funded by the Hessian Ministry of Science and Arts (HMWK). A.J. acknowledges the German Academic Exchange Service (DAAD) for granting the WISE scholarship to pursue the summer internship at TU Darmstadt.

Open access funding enabled and organized by Projekt DEAL.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Keywords

machine learning (ML), Monte Carlo (MC), non-spherical particles

Received: July 19, 2023
Published online: August 31, 2023

- [1] D. Frenkel, B. Smit, in *Understanding Molecular Simulation: From Algorithms to Applications*, Elsevier, Amsterdam, Netherlands **2001**.
- [2] S. A. Hollingsworth, R. O. Dror, *Neuron* **2018**, 99, 1129.
- [3] R. J. Ellis, *Trends Biochem. Sci.* **2001**, 26, 597.
- [4] J. W. Tavasoli, P. Bauër, M. Fermigier, D. Bartolo, J. Heuvingh, O. d. Roure, *Soft Matter* **2013**, 9, 9103.
- [5] M. P. Allen, G. T. Evans, D. Frenkel, B. M. Mulder, in *Advances in Chemical Physics*, (Eds.: I. Prigogine, S. A. Rice), Vol. 86, Wiley, New Jersey, USA **1993**.
- [6] T. D. Nguyen, S. J. Plimpton, *Comput. Phys. Commun.* **2019**, 243, 12.
- [7] T. Cover, P. Hart, *IEEE Trans. Inf. Theory* **1967**, 13, 21.
- [8] C.-C. Chang, C.-J. L. Lin, *ACM Trans. Intell. Syst. Technol.* **2011**, 2, 1.
- [9] K. Crammer, Y. Singer, *J. Mach. Learn. Res.* **2002**, 2, 265.
- [10] T.-F. Wu, C.-J. Lin, R. C. Weng, *J. Mach. Learn. Res.* **2004**, 5, 975.
- [11] Y. Freund, R. E. Schapire, *J. Comput. Syst. Sci.* **1997**, 55, 119.
- [12] L. Breiman, *Mach. Learn.* **2001**, 45, 5.
- [13] L. Breiman, in *Classification and Regression Trees*, Routledge L. Breiman, in Classification and Regression Trees, New York, USA **2017**.
- [14] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, 2nd Ed, Springer Series in Statistics, Springer, New York, NY, USA **2009**.
- [15] J. H. Friedman, *Ann. Stat.* **2001**, 29, 1189.
- [16] J. H. Friedman, *Comput. Stat. Data Anal.* **2002**, 38, 367.
- [17] J. D. M. Rennie, L. Shih, J. Teevan, D. R. Karger, presented at Proc. of the Twentieth Intl. Conf. on Machine Learning, AAAI Press, Washington, DC, USA **2003**, pp 616–623.
- [18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. S.-L. Duchesnay, *J. Mach. Learn. Res.* **2011**, 12, 2825.
- [19] S. Bag, R. Mandal, *Soft Matter* **2021**, 17, 8322.
- [20] M. P. Lettinga, E. Barry, Z. Dogic, *Europhys. Lett.* **2005**, 71, 692.
- [21] A. R. Khokhlov, A. N. Semenov, *Phys. Stat. Mech. Its Appl.* **1981**, 108, 546.
- [22] P. Bolhuis, D. Frenkel, *J. Chem. Phys.* **1997**, 106, 666.
- [23] M. P. Allen, D. J. Tildesley, in *Computer Simulation of Liquids*, Clarendon Press, Oxford **1989**.
- [24] M. C. Bott, J. M. Brader, R. Wittmann, F. Winterhalter, M. Marechal, A. Sharma, *Phys. Rev. E* **2018**, 98, 012601.
- [25] A. Gschwind, M. Klopotek, Y. Ai, M. Oettel, *Phys. Rev. E* **2017**, 96, 012104.
- [26] R. Hentschke, J. I. Herzfeld, *Phys. Rev. A* **1991**, 44, 1148.

- [27] L.-T. Yan, in *Self-Assembling Systems: Theory and Simulation*, Wiley, New Jersey, USA **2016**.
- [28] S. C. McGrother, D. C. Williamson, G. Jackson, *J. Chem. Phys.* **1996**, *104*, 6755.
- [29] Y. G. Tao, W. K. Den Otter, J. K. G. Dhont, W. J. Briels, *J. Chem. Phys.* **2006**, *124*, 134906.
- [30] A. Samborski, G. T. Evans, C. P. Mason, M. P. Allen, *Mol. Phys.* **1994**, *81*, 263.
- [31] R. Berardi, C. Fava, C. Zannoni, *Chem. Phys. Lett.* **1998**, *297*, 8.
- [32] N. Kern, D. Frenkel, *J. Chem. Phys.* **2003**, *118*, 9882.
- [33] J. Ghaboussi, R. Barbosa, *Int. J. Numer. Anal. Methods Geomech.* **1990**, *14*, 451.
- [34] F. Alonso-Marroquín, Y. Wang, *Granul. Matter* **2009**, *11*, 317.
- [35] J. Wang, H. S. Yu, P. Langston, F. Fraige, *Granul. Matter* **2011**, *13*, 1.
- [36] F. Y. Fraige, P. A. Langston, A. J. Matchett, J. Dodds, *Particuology* **2008**, *6*, 455.
- [37] M. Spellings, R. L. Marson, J. A. Anderson, S. C. G. P. U. Glotzer, *J. Comput. Phys.* **2017**, *334*, 460.