ADVANCED
THEORY AND
SIMULATIONS

Supporting Information

for *Adv. Theory Simul.*, DOI 10.1002/adts.202300520

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

Saientan Bag, Ayush Jha and Florian Müller-Plathe*

Supporting Information for "Machine Learning Assisted Monte Carlo Simulation: Efficient Overlap determination for Nonspherical Hard Bodies"

Saientan Bag, Ayush Jha and Florian Müller-Plathe*

Eduard-Zintl-Institut für Anorganische und Physikalische Chemie, Technische Universität Darmstadt, Alarich-Weiss-Str. 8, 64287 Darmstadt, Germany.

*email: s.bag@theo.chemie.tu-darmstadt.de

Rigid Body Represented by Collection of Disks

Rigid bodies of four different shapes (circle, triangle, rod and star) were modeled as many constituent small disks of the same size. Only the periphery of the rigid bodies were represented here with the disk. Representing only the periphery in contrast to the whole object (see Figure 1 in the main text) reduces the number of constituent disks (see Figure S1) except the rod case.

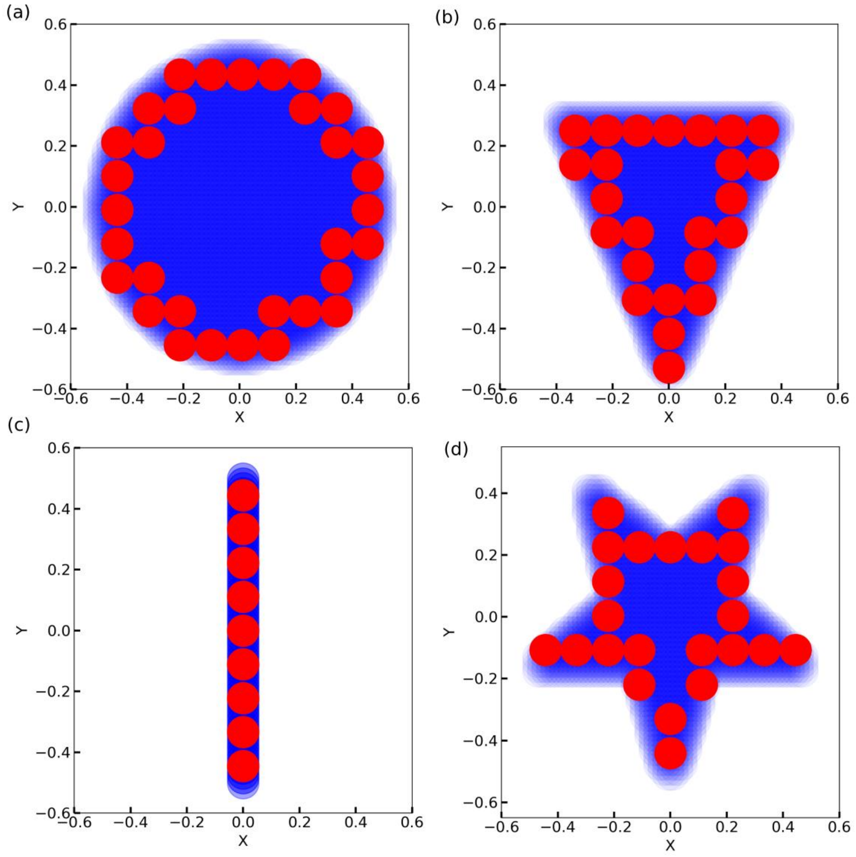

Figure S1: (a-d) Rigid body (the blue background) with different geometric shapes (circle,

triangle, rod and star) modelled by a series of small disks (shown in red). The disks were used only to represent the boundary of the rigid bodies.

Learning Curves for the Machine Learning (ML) models

To determine the overlap between two rigid bodies (see Figure S1), we calculated the distance between constituent disks (see Figure S1). We designed Machine Learning (ML) models to predict the overlap between two rigid bodies. We followed exactly the same protocol to determine the overlap and build the ML model as mentioned in the main text. The learning curves for the ML models are shown in Figure S2.

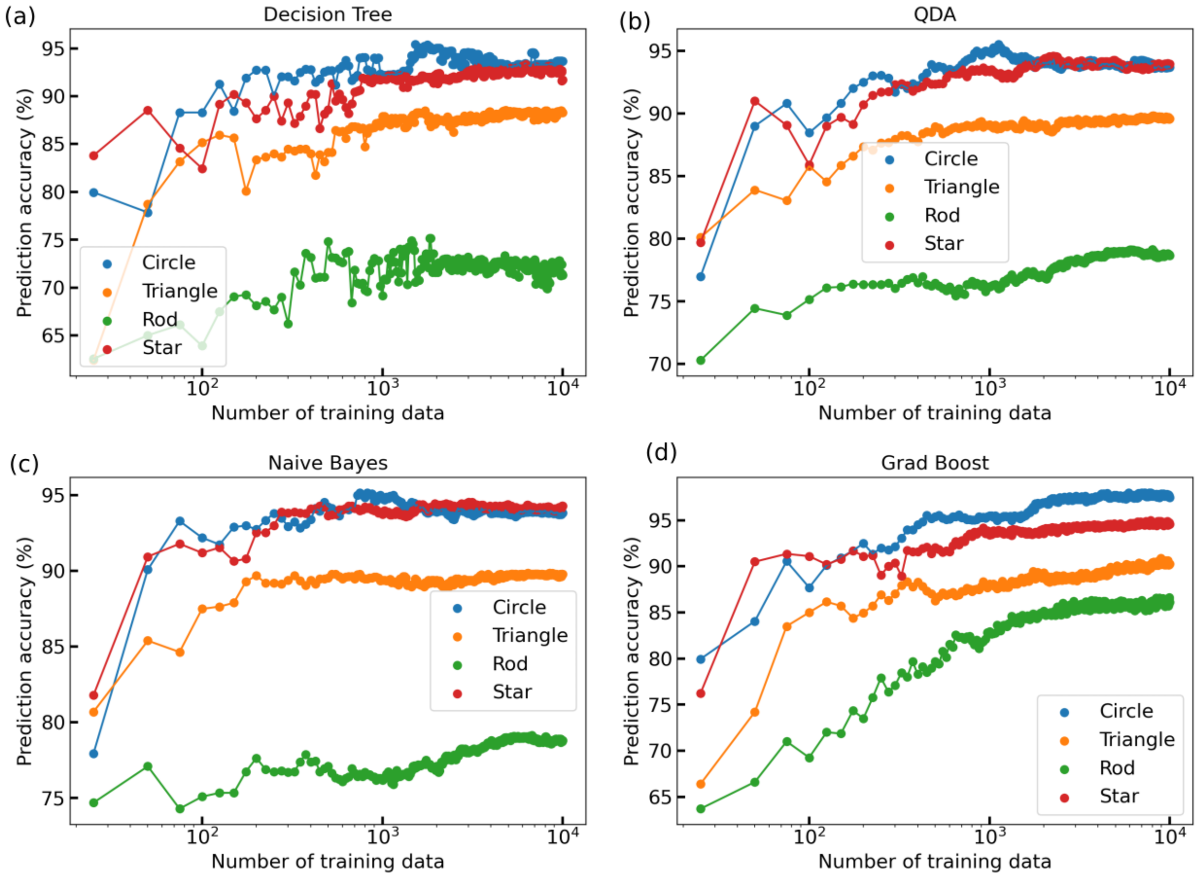

Figure S2: Overlap prediction accuracy in test dataset as a function of number of training data for different rigid body cases. The overlap predictions were done using 4 different machine learning models (a) Decision Tree, (b) Quadratic Decision Analysis (QDA) (c) Naive Bayes and (d)Gradient Boosting.

Structural Properties of the Systems of 64 Rigid Bodies calculated using MC Simulation

We performed Monte Carlo (MC) simulation using the machine learning (ML) models (see Figure 1 to 4 in the main text) to detect the overlap between the rigid bodies. The details of the MC simulation were presented in the main text in-detail. Pair correlation functions calculated from the MC generated trajectories are shown in Figure S3 below. All the systems were prepared at a fraction of occupied surface area of 0.20.

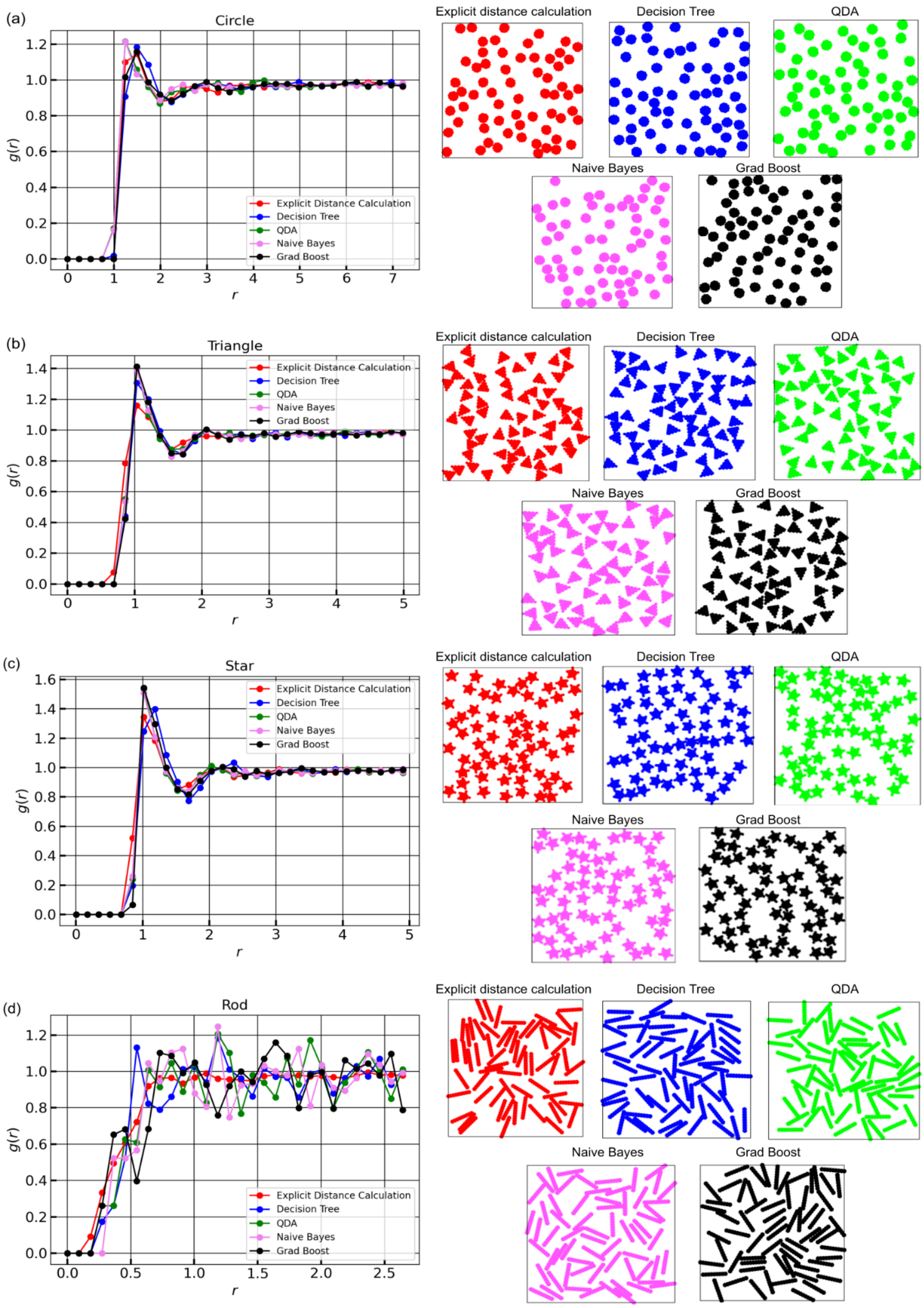

Figure S3: (a-d) Pair correlation function (left panel) and the equilibrated simulation snapshots (right panels) obtained from 10^5 steps of MC Simulation with 64 rigid bodies of different geometric shapes: (a) circle (b) triangle (c) star and (d) rod. MC simulations were performed using the ML model for overlap determination. The results are compared with reference calculations.

We also simulated a system of 64 rigid rods at a number density of 1.02 (corresponding to an area fraction of 0.09). The equilibrated simulation snapshots and the calculated pair correlation function are shown below.

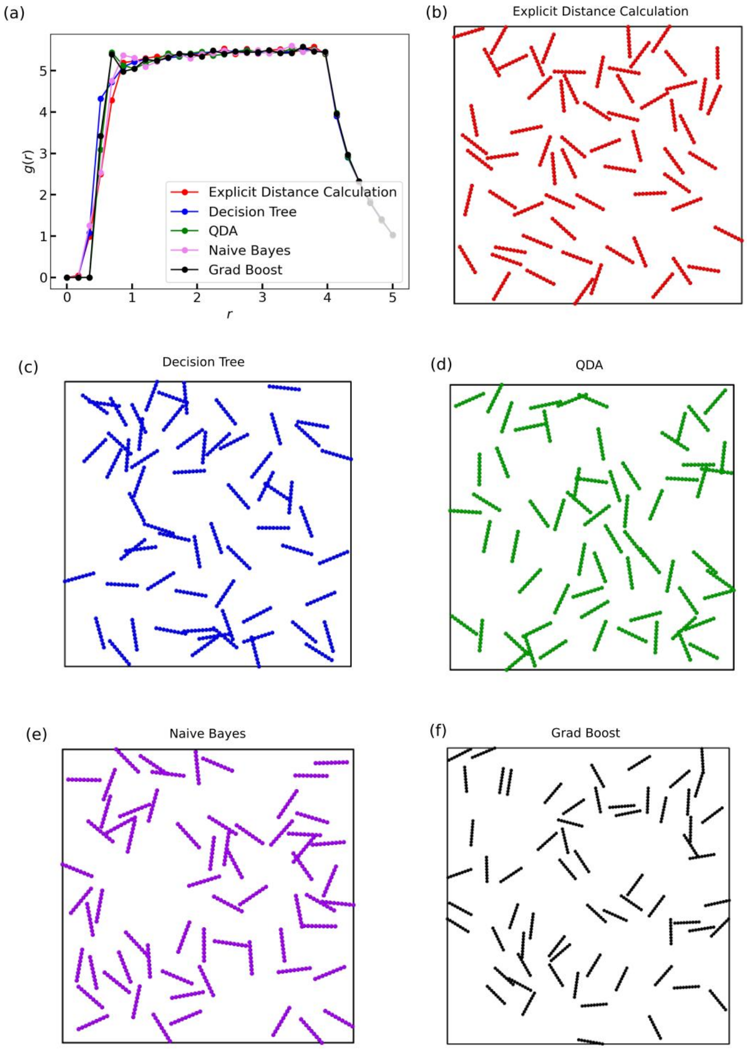

Figure S4: (a) Pair correlation function (top panel) and the (b-f) equilibrated simulation snapshots obtained from 10^5 steps of MC Simulation with 64 rigid rods at area fraction of 0.09. MC simulations were performed using the ML models (b-f) for overlap determination as well as the using (b) explicit distance calculation.

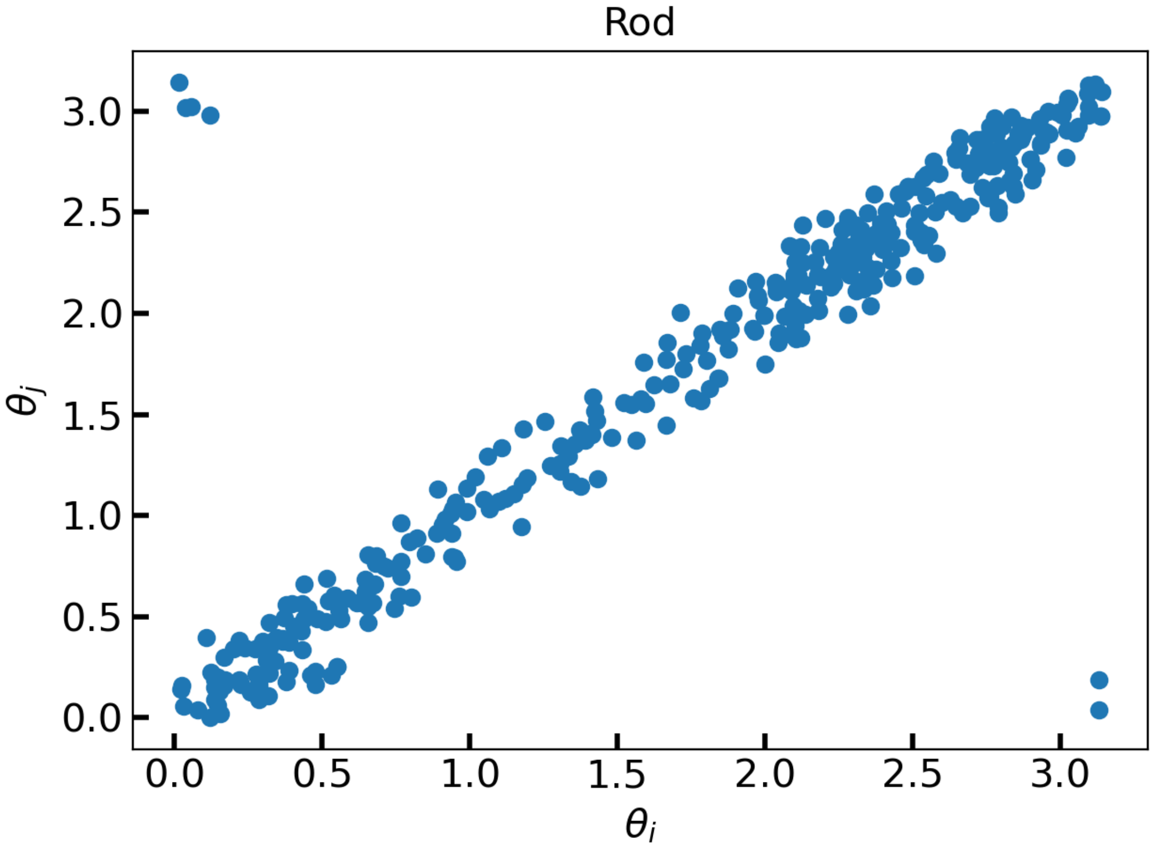

Figure S5: Correlation between orientational angles (in radian) of pairs of rods with a distance of 0.12 between the center of geometries. Linear correlation between the angles indicated the tendency of the rods to stay parallel to each other.

Details of Hardware Used to Run the Codes

All the codes were run in a single cpu of a local desktop computer with the following details

```
Architecture:           x86_64
CPU op-mode(s):         32-bit, 64-bit
Byte Order:             Little Endian
Address sizes:          39 bits physical, 48 bits virtual
CPU(s):                 8
On-line CPU(s) list:    0-7
Thread(s) per core:     2
Core(s) per socket:     4
Socket(s):              1
NUMA node(s):           1
Vendor ID:              Genuine Intel
CPU family:             6
Model:                  158
Model name:             Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz
Stepping:               9
CPU MHz:                3600.000
CPU max MHz:            4200,0000
CPU min MHz:            800,0000
BogoMIPS:               7200.00
Virtualization:         VT-x
L1d cache:              128 KiB
L1i cache:              128 KiB
L2 cache:               1 MiB
L3 cache:               8 MiB
NUMA node0 CPU(s):      0-7
```

All the codes were written in Python and compiled using Python 3.9.7.
For the ML models, we used sklearn: 1.0.1