A Multipronged Computational Approach to Nanoparticle Design: Characterization, Alchemical Modeling and Advanced Simulation

by

Shih-Kuang Lee

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Material Science and Engineering and Scientific Computing)
in The University of Michigan
2026

Doctoral Committee:

Professor Sharon C. Glotzer, Chair
Professor Rebecca K. Lindsey
Professor Liang Qi
Professor Wenhao Sun

Shih-Kuang Lee

shihkual@umich.edu

ORCID iD: 0009-0002-4677-336X

© Shih-Kuang Lee 2026

DEDICATION

For my beloved father, mother, and brother.

ACKNOWLEDGEMENTS

Pursuing a PhD has been, by far, the most impactful decision of my life. Throughout this six-year journey, I have not only gained a profound understanding of scientific research but have also fundamentally reshaped my worldview and my perspective on what it means to live a meaningful life. Most importantly, the people I encountered along the way have been the heart of this experience; they are the reason I have become the person I am today.

First, I want to thank my advisor, **Professor Sharon Glotzer**. When she welcomed me into her lab at the end of my first year, everything changed for me. At that time, I was unsure about my place in academia, but her support gave me the confidence to try bold ideas. I am grateful for the freedom she gave me and for the chance to grow in such a lively research group. It has been a privilege to learn from her and work with the talented scientists she brought together. As I finish this chapter, I feel ready for what comes next because of the lessons she taught me.

I also want to thank **Professor Liang Qi** for his important support during my first year. Because of him, I was able to come to this wonderful university and start my life as a doctoral student in Ann Arbor. Starting a PhD during the COVID-19 pandemic was very challenging, and I am truly grateful that he chose to mentor me during such a tough time. Without his help at the beginning, none of this would have happened.

During my time in the Glotzer group, I was lucky to be surrounded by mentors

and mentees who became the core of my PhD experience. **Dr. Sun-Ting Tsai** has been my most influential mentor. With his guidance on my first study, I learned how to write scientific papers and gained knowledge that was key to the rest of my dissertation. He helped me develop the mindset and intuition needed for good research. Most importantly, he showed me what a great mentor is: not just a brilliant scientist, but also a generous and helpful colleague. He taught me how to build a healthy and productive relationship with an advisor, and I am truly grateful for that. It has been a joy to work with him over the years.

I am also very grateful to have worked with **Melody Zhang**, who started as a PhD student I mentored in both the Glotzer Lab and Lindsey Lab. As my co-author on the second part of this dissertation, she has become a strong and independent researcher. From her, I learned how important it is to stay curious and connect with others. Our mentorship helped both of us grow, and I truly enjoyed working together. I will always remember the time we spent plotting hundreds of figures for the paper and her great sense of humor, which always made me laugh. I hope my guidance helps her face the challenges ahead as she gets closer to graduation. I wish her all the best!

I also want to thank the colleagues I worked with on projects in the Glotzer Lab: Tim Moore, Domagoj Fijan, Philipp Schönhöfer, Joshua Anderson, and Joseph Burkhart. Your teamwork and ideas were essential to this work. I am especially grateful to our lab managers, Karen Coulter and Lori Bolchalk. You really keep our group running, and without your hard work, none of this would have been possible.

Finally, I want to end these acknowledgements by thanking my parents, one a doctor and the other a PhD candidate, and my brother. Thank you for your constant love and support from far away in Taiwan. You have been my foundation from the

very start.

TABLE OF CONTENTS

DEDICATION . . . . . **ii**

ACKNOWLEDGEMENTS . . . . . **iii**

LIST OF FIGURES . . . . . **viii**

LIST OF TABLES . . . . . **xiv**

ABSTRACT . . . . . **xv**

CHAPTER

I. Introduction 1

1.1 Prologue 1
1.2 Order Parameters: Reducing Complexity in Many-Body Dynamics 3
1.3 Bias Simulation: Overcoming Barriers in Configuration Space 5
1.4 Digital Alchemy: Inverse Design via Extended Ensembles 7
1.5 Outlines 9

II. Classification of Complex Local Environments in Systems of Particle Shapes Through Shape-Symmetry Encoded Data Augmentation 11

2.1 Introduction 11
2.2 Methods 14
2.2.1 Per-Particle Quantities as Input Features 14
2.2.2 Shape-Symmetry Encoded Data Augmentation 17
2.2.3 Multilayer Perceptron (MLP) as Local Environment Classifier . . . 20
2.2.4 Data Preparation 22
2.3 Results and Discussion 25
2.3.1 Test Case 1: Simple Cubic Crystals Assembled by Hard Cubes . . 25
2.3.2 Test Cases 2 and 3: Self-Assembly of 2D and 3D Patchy Particles . 28
2.3.3 Test Cases 4 and 5: Self-Assembly of Prolate Hexagonal Bipyramids 31
2.3.4 Test Cases 6 and 7: Self-Assembly of Truncated Polyhedra 33
2.4 Conclusion 37

III. A Generalized Machine-Learning Framework for Developing Alchemical Many-Body Interaction Models for Polymer Grafted Nanoparticles 38

3.1 Introduction 39
3.2 Methods 43
3.2.1 PMF Generation 44
3.2.2 The ChIMES Machine-Learned Interaction Model 46
3.2.3 Simulation Protocols 50

3.3 Results and Discussion . . . 51
3.3.1 PMF Generation . . . 51
3.3.2 OS ChIMES Models . . . 58
3.3.3 OS X-ChIMES Models for Inverse Design . . . 66
3.3.4 Digital Alchemy with X-ChIMES Model . . . 69
3.4 Conclusion . . . 73
3.5 Supplementary Information . . . 74
3.5.1 One-Site Coarse-Grain (OS-CG) Mapping . . . 74
3.5.2 Martini Model Description . . . 75
3.5.3 Background on the Forward-Reverse (FR) Method . . . 76
3.5.4 ChIMES Fitting using Truncated Singular Value Decomposition (TSVD) . . . 82
3.5.5 One-Site (OS) ChIMES Model Prediction . . . 85
3.5.6 Condensed Phase Simulation . . . 85
3.5.7 One-Site (OS) Extended ChIMES (X-ChIMES) Model . . . 89
3.5.8 Application of the Forward-Reverse (FR) Method in Steered Molecular Dynamics (SMD) Simulations . . . 90
3.5.9 ChIMES Model Fitting . . . 95
3.5.10 OS ChIMES Simulations . . . 96

IV. Exploratory Digital Alchemy for Colloidal Crystal Discovery . . . 97

4.1 Introduction . . . 97
4.2 Methods . . . 99
4.2.1 Digital Alchemy (DA) . . . 99
4.2.2 Exploratory Digital Alchemy (EDA) . . . 101
4.2.3 Collective Variable (CV) to Sample Crystal Structures . . . 103
4.3 Results and Discussion . . . 104
4.3.1 Lennard-Jones Gauss Potential (LJGP) . . . 106
4.3.2 Oscillating Pair Potential (OPP) . . . 109
4.4 Conclusion . . . 114

V. Towards A New Generation of Molecular Simulation Engine: Successor of HOOMD-blue in Rust . . . 115

5.1 Introduction . . . 115
5.2 The Landscape of Molecular Simulation Engines . . . 117
5.3 Why Rust for Molecular Simulation? . . . 119
5.4 HOOMD-rs Implementation . . . 120
5.4.1 Modular and Flexible MD Symplectic Integrator . . . 123
5.4.2 ChIMES Machine Learning Potential Module . . . 127
5.5 Conclusion . . . 132

VI. Conclusion and Outlook . . . 133

6.1 Summary of Contributions . . . 133
6.2 General Significance . . . 134
6.3 Outlook and Future Work . . . 135

BIBLIOGRAPHY . . . **137**

LIST OF FIGURES

Figure

2.1 **Systems of particle shapes** The particle systems studied in this paper: **(a)** cube, **(b)** patchy triangle, **(c)** patchy triangular prism [1], **(d)** hexagonal bipyramid with aspect ratio $\alpha = 3.0$, **(e)** hexagonal bipyramid with $\alpha = 1.28$, **(f)** truncated tetrahedron, and **(g)** truncated octahedron. In (a) and (b), the transparent blue region decorating the particles indicates the attractive patch, while the transparent red region indicates a repulsive patch. Also shown are the shape symmetry-related factors D_θ, D_ϕ, and mirror plane normal vector $\mathbf{n}$ utilized for each system. Each particle pictured here is oriented such that it is in the reference orientation given by the quaternion $\mathbf{q}_0 = (1, 0, 0, 0)$. 14

2.2 **Rotation in the local environment** The blue bipyramid represents the target particle, and the white bipyramid represents a neighboring particle. 15

2.3 **The data augmentation and the architecture of Multilayer Perceptron (MLP)** The diagram of the data augmentation and the MLP architecture used to classify local particle environment. The raw local environment data is augmented through sets of rotations and passed to the input layer of MLP. The input layer has $N_b + 4N_b + 2N_b$ units as described in Equation 2.9, and the hidden layer contains 256 units. The number of output layers depends on the system. For example, we will present results for a system of particles undergoing a two-steps crystallization transition, where the output layer is composed of three units representing the probability that a given particle belongs to one of three predefined phases. 15

2.4 **Simple cubic lattice consisted of cubes**. Summary of the MLP classifier's classification results, training details, and accuracy tests, in which we use the simple cubic system as a test case. **(a)** The MLP classifier's classification results on three different snapshots. **(b)** The MLP classifier's classification results on the self-assembly trajectory and classification, compared to the Steinhardt order parameter Q_4 (dotted line) calculated as ground truth. For visualization purposes, solid lines represent the MLP classifier trained on the data accounting for symmetry, while dashed lines are for the MLP classifier trained on the data without accounting for symmetry. Annotations I, II, and III indicate the three corresponding snapshots in (a) for the classifier trained on augmented data. **(c)** Learning curve of the MLP classifier used in (a) with (solid line) and without (dashed line) data augmentation **(d)** MLP classification accuracy plotted versus increasing number of training epochs **(e)** MLP classification accuracy plotted versus increasing number of neighbors N_b in the local particle neighborhood. In (a)-(d), we used fixed $N_b = 6$. In (a)-(c) and (e), we used 30 training epochs. 26

2.5 **Kagome lattice of patchy triangles** Summary of the MLP classifier's classification results on the test assembly trajectory. **(a)** The MLP classifier's classification results on the three snapshots. **(b)** The MLP classifier's classification results on the entire trajectory. For visualization purposes, solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. The annotations I, II, and III correspond to the three snapshots in (a) for the classifier trained on augmented data. 29

2.6 **Dimer-diamond phase consisted of anti-aligned patchy triangular prism [1]** The MLP classifier's classification results on the testing self-assembly trajectory, in which we identify the crystallization dimer-diamond structure. Each Wyckoff site comprises two anti-aligned patchy prisms forming a gyrobifastigium. **(a)** The MLP classifier's classification results on the three snapshots. **(b)** The MLP classifier's classification results on the whole trajectory. For visualization purposes, solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. The annotates I, II, and III indicate the corresponding snapshots in (a) for the classifier trained on augmented data. 30

2.7 **Hexagonal bipyramid systems** The MLP classifier's classification results on the assembly trajectories for two systems of hexagonal bipyramids. Both systems exhibit a two-step transition when crystallizing from an initial disordered fluid phase. **(a)** and **(c)** The MLP classifier's classification on six snapshots of the hexagonal bipyramid systems with (a) $\alpha = 3.0$ and (b) $\alpha = 1.28$, as well as the corresponding radial distribution functions at the bottom. **(b)** and **(d)** The MLP classifier's classification results on the entire trajectories. The annotations I, II, III, and IV, V, VI correspond to the snapshots in (a) and (b), respectively, for the classifier trained on augmented data. For visualization purposes, solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. 32

2.8 **Diamond structure of truncated tetrahedrons** The MLP classifier's classification results on the self-assembly test trajectory, in which we identify the crystallization of the diamond structure from an initially disordered fluid phase. **(a)** The MLP classifier's classification results on snapshots taken at three points along the trajectory. **(b)** The MLP classifier's classification results on the entire trajectory. Solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. The annotations I, II, and III indicate the corresponding snapshots in (a) for the classifier trained on augmented data. . . . 34

2.9 **High-pressure lithium phase of truncated octahedrons** The MLP classifier's classification results on the self-assembly test trajectory, in which we identify the crystallization of the high-pressure lithium phase from an initially disordered fluid phase. **(a)** The MLP classifier's classification results on the three snapshots. **(b)** The MLP classifier's classification results on the entire trajectory. Solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. The annotations I, II, and III indicate the corresponding snapshots in (a) for the classifier trained on augmented data. 35

3.1 **Polymer-grafted nanoparticle (PGN) coarse-graining scheme and alchemical PGN interaction surface, W. Top**: This work treats PGN interactions at coarse-grained (CG) resolution. Two CG resolutions are considered: multi-site (MS), where approximately 4 heavy atoms are mapped to one interaction bead, and one-site (OS) resolution, where an entire PGN is treated as a single interaction site. **Bottom**: This work details the extension of OS-CG models to enable the description of W as a function of both inter-PGN separation distance r and polymer length α. 40

3.2 **Comparison of PMFs generated via umbrella sampling (US) and forward-reverse (FR) steered molecular dynamics (SMD) for PGNs with $\alpha = 4$ and 8.** SMD pulling velocities are chosen such that PMFs calculated based on the FR method are consistent with those obtained by the US method. 52

3.3 **Two-body PMFs $W_{\mathrm{2B}}^{\mathrm{MS\text{-}CG}}$ for PGNs with $\alpha = 3$ to 8.** 54

3.4 **Overview of many-body PMFs. Top**: Schematic representation of three-body SMD simulation via FR method. The three-body SMD simulations are performed by positioning PGN 1 and 2 at various separation distances r_{12} while systematically pulling PGN 3 from different initial positions. **Bottom**: The left and middle figures compare the two-body PMFs $W_{\mathrm{2B}}^{\mathrm{MS\text{-}CG}}$ (dashed lines) and many-body PMFs $W_{\mathrm{mB}}^{\mathrm{MS\text{-}CG}}$ (solid lines) for PGNs with α=4 and 8. Each PMF is a function of an isosceles trimer configuration as illustrated in the right figure, where r is the two equal sides (magenta arrows) and r_{12} is the fixed base (black arrow). The deviation between $W_{\mathrm{2B}}^{\mathrm{MS\text{-}CG}}$ and $W_{\mathrm{mB}}^{\mathrm{MS\text{-}CG}}$ indicates the presence of non-negligible three-body interactions. 55

3.5 **Three-body PMF $W_{\mathrm{3B}}^{\mathrm{MS\text{-}CG}}$ and many-body PMF $W_{\mathrm{mB}}^{\mathrm{MS\text{-}CG}}$ heatmaps for $\alpha = 4$ and 8 PGNs with r_{12} fixed at 6 nm.** The contour interval is 80 kcal/mol. 57

3.6 **Performance of ChIMES model $W^{\mathrm{OS\text{-}CG}}$ fits for PGNs with $\alpha = 4$ and 8.** Parity plots compare ChIMES model predictions with ground truth data (top: $W_{\mathrm{2B}}^{\mathrm{MS\text{-}CG}}$; bottom: $W_{\mathrm{mB}}^{\mathrm{MS\text{-}CG}}$). The red line is the $y = x$ line. 59

3.7 **OS-CG model validation with PGN trimers. Top**: Initial trimer configurations and relaxation of trimers initialized from different configurations for $\alpha =$ 4 and 8. Transparent lines indicate individual replica simulations, and the solid lines with square markers represent the average values over all five replicas. **Bottom**: $W_{\mathrm{mB}}^{\mathrm{OS\text{-}CG}}$ for $\alpha = 4$ and 8 as a function of trimer configuration with fixed inter-PGN distance r_{12} and various r_{13} and r_{23}. The starred locations in PMFs correspond to schematic drawings of PGN configurations on the right side. The contour interval is 42.5 kcal/mol. 61

3.8 **Pressure-volume relationship predicted by MS-CG and OS-CG models for PGNs with $\alpha = 4$ and 8.** 63

3.9 **Radial distribution functions (RDFs) for simulations using MS-CG and many-body OS-CG models at different pressures, for PGNs with $\alpha = 4$ and 8.** For OS-CG models, orange transparent lines represent individual replica simulations. Almost every OS-CG replica simulation result overlaps, except the simulations in the top right corner. The blue dashed lines indicate the averaged RDFs calculated from five replica simulations using the MS-CG models. 64

3.10 **Performance of the fits of two-body extended ChIMES (X-ChIMES) model $W_{\mathbf{2B}}^{\mathbf{X\text{-}OS\text{-}CG}}$. Top**: Parity plot comparing two-body X-ChIMES model predictions with ground truth data $W_{2\mathrm{B}}^{\text{MS-CG}}$. We use $\alpha = 5$ and 7 (i.e., hold-out validation data) to demonstrate X-ChIMES' excellent performance. **Bottom**: Two-body X-ChIMES model predictions as a function of inter-PGN distance r and polymer length α, illustrating simultaneous predictions of two-body PMFs of all six PGNs. The small inset figure depicts $\alpha = 6$ slice where the red line represents $W_{2\mathrm{B}}^{\text{X-OS-CG}}$ and black dots represent $W_{2\mathrm{B}}^{\text{MS-CG}}$. The contour interval is 80 kcal/mol. 68

3.11 **Performance of the fits of three-body extended ChIMES (X-ChIMES) model $W_{\mathbf{3B}}^{\mathbf{X\text{-}OS\text{-}CG}}$. Top**: Parity plot comparing three-body X-ChIMES model predictions with ground truth data $W_{3\mathrm{B}}^{\text{MS-CG}}$. We use $\alpha = 5$ and 7 (i.e., hold-out validation data) to demonstrate X-ChIMES' excellent performance. **Bottom**: three-body X-ChIMES model predictions at different α as a function of trimer configurations, i.e., fixed $r_{12} = 6$ nm and varied r_{13} and r_{23}. $\alpha = 4$ and 8 are included for X-ChIMES model training and $\alpha = 5$ and 7 are excluded for hold-out validation. The contour interval is 100 kcal/mol. 70

3.12 **Alchemical Monte-Carlo (Alch-MC) simulation trajectories.** We illustrate the Alch-MC simulation trajectories using many-body X-OS-CG model $W_{\mathrm{mB}}^{\text{X-OS-CG}}$ by plotting α evolution over Alch-MC sweeps. The blue lines with square markers indicate the averaged trajectories of five replicas in each set of simulations, with each initialized at $\alpha = 3$, 5, and 7. The black dashed lines indicate t_{end} during the Alch-MC sweep, where we turn off the biased potential for the target crystal, described in equation 1.8. **Top**: Alch-MC simulations for target FCC crystals with decreasing lattice constant a_0. **Bottom**: Alch-MC simulations for target BCC crystal with decreasing a_0. 72

3.13 **Forward-reverse (FR) method demonstration and SMD velocity convergence test for PGNs with $\boldsymbol{\alpha = 4}$ and 8.** To ensure the accuracy of the FR method, a sufficiently slow SMD pulling velocity must be selected. **Top**: Various pulling velocities are tested for PGNs with $\alpha = 4$ and 8. **Bottom**: PMF calculated using the FR method with the selected pulling velocity. 81

3.14 **Illustration of two- and many-body PMFs for PGNs with $\boldsymbol{\alpha = 3}$.** The spacing between contour lines for the PMFs is 40 kcal/mol, and 10 kcal/mol for the error map. 82

3.15 **Illustration of two- and many-body PMFs for PGNs with $\boldsymbol{\alpha = 4}$.** The spacing between contour lines for the PMFs is 40 kcal/mol, and 10 kcal/mol for the error map. 83

3.16 **Illustration of two- and many-body PMFs for PGNs with $\boldsymbol{\alpha = 5}$.** The spacing between contour lines for the PMFs is 40 kcal/mol, and 10 kcal/mol for the error map. 83

3.17 **Illustration of two- and many-body PMFs for PGNs with $\boldsymbol{\alpha = 6}$.** The spacing between contour lines for the PMFs is 40 kcal/mol, and 10 kcal/mol for the error map. 84

3.18 **Illustration of two- and many-body PMFs for PGNs with $\boldsymbol{\alpha = 7}$.** The spacing between contour lines for the PMFs is 40 kcal/mol, and 10 kcal/mol for the error map. 84

3.19 **Illustration of two- and many-body PMFs for PGNs with $\alpha = 8$.** The spacing between contour lines for the PMFs is 40 kcal/mol, and 10 kcal/mol for the error map. 85

3.20 **Steinhardt order parameter $\overline{q}_6$ for simulations of PGNs described by the OS-CG model initialized on an FCC lattice with $\alpha = 8$.** Analysis is performed for the same set of simulations demonstrated in the top right of Figures 8 and 9 in the main text. Blue solid lines and square markers give results for five replica simulations and the average of each simulation, respectively. The black dashed lines indicate $\overline{q}_6$ for FCC crystals. The red dash-dotted lines indicate $\overline{q}_6$ for BCC crystals. 86

3.21 **Snapshots of expanding box simulation for PGNs with $\alpha = 4$ and 8.** . . 87

3.22 **The approximated phase diagram of PGNs with $\alpha = 3$ - 8 using X-ChIMES model.** 88

3.23 **The normalized partial number integral $\overline{N}(r)$ of isolated PGNs for calculating ligand capping layer thickness λ**. We illustrate the $\overline{N}(r)$ between core nanoparticles and the polymer beads, emphasized at the distance between core nanoparticles and polymer beads r where $\overline{N}(r)$ reaches 1. 88

3.24 **Performance of the two- and three-body extended ChIMES (X-ChIMES) model fits based on training data**. **Left**: two-body X-ChIMES parity plot with Pearson correlation value of $R^2 \sim 0.9999$. **Right**: three-body X-ChIMES parity plot with Pearson correlation value of $R^2 \sim 0.9343$. 89

3.25 **Single point energy calculation of FCC and BCC crystals using X-ChIMES models.** 90

3.26 **Alchemical Monte-Carlo (Alch-MC) simulation trajectories.** We illustrate the energy evolution of Alch-MC simulation trajectories using many-body X-OS-CG model $W_{\mathrm{mB}}^{\mathrm{X\text{-}OS\text{-}CG}}$ by plotting the total energy E evolution over Alch-MC sweeps. 91

4.1 **Graphical abstract** The Exploratory Digital Alchemy (EDA) modifies the original Digital Alchemy (DA) framework, which applies the harmonic constraints (dashed line on the left) on crystal structures, causing the elevation of generalized free energy (red region), and forces the system to equilibrate towards the pre-determined state for inverse design. Instead, EDA combines the generalized ensemble of DA with Metadynamics, which iteratively accumulates Gaussian biases (green region) and enables exploration of the correspondence between crystal structure and particle design. 98

4.2 **Two-dimensional Lennard-Jones-Gauss Potential (LJGP) system** (A) Thermodynamic phases discovered by the EDA method, including: **I** square lattice, **II** hexagonal lattice, and **III** disordered phase. **I**' and **II**' represent the same crystal structure with different lattice constants (B) The LJGP energy profile at different r_0. (C) Helmholtz free energy surface F (FES) as a function of r_0 and temperature. (D), (E) and (F) The points of collective variables (r_0, $\overline{w}_6$) sampled by the EDA method and the corresponding FES at various temperatures. The minima on the FES are indicated by the corresponding phases in (A). The colorbar indicates the FES value $\Phi(r_0, \overline{w}_6)$ in units of $k_B T$. 105

4.3 **Equilibrium phases obtained from cooling simulations for the Lennard-Jones-Gauss Potential system with $r_0 = 1.8$ to 2.1.** The four snapshots indicated by the grey box in the middle are the r_0 values that we intend to confirm that the rhombic lattice is unstable at a finite temperature. 106

4.4 **Three-dimensional Oscillating Pair Potential (OPP) system** (A) and (B) The OPP energy profile for $k = 6$ and 8.5 respectively, at different ϕ. (C) and (D) The points of collective variables (ϕ, $\overline{w}_l$) sampled by the EDA method and the corresponding free energy surface (FES), for $k = 6$ and 8.5, respectively. In each subfigure of (C) and (D), the top row plots the diffraction pattern and bond-orientational order diagram for the three crystals discovered, and the FES minima at the bottom row are indicated by the corresponding phases. The colorbars indicate the FES value $\Phi(\phi, \overline{w}_l)$ in units of k_BT. 110

4.5 **Transition between A15 and σ phases at $k = 8.5$** (A) The perfect A15 phase. (E) The perfect σ phase. (B) The beginning (C), intermediate, and final stage (D) of the transition. In (B), the stacking fault is indicated by the black dashed line. The special motifs of A15 and the σ phase are highlighted by the brown square and green pentagons, respectively. 111

4.6 **Stability test for Oscillating Pair Potential system** (A) The evolution of the unbiased EDA simulation snapshots at $k = 6$, starting from the equilibrium $hP2$ phase at $k_BT = 0.1$ and $\phi = 4$. (B) The evolution of the unbiased EDA simulation snapshots at $k = 8.5$, starting from the equilibrium $cP54$ phase at $k_BT = 0.1$ and $\phi = 4$. 112

5.1 Component architecture of the HOOMD-rs. 121

5.2 **The error in energy conservation** The error bars indicate the maximum and minimum values of the samples. 125

5.3 **The kinetic energy distribution** The green curve is the ground truth gamma distribution of Gamma($\alpha = N_{\text{dof}}/2, \theta = k_BT$), and the bar plots are sampled from either HOOMD-blue or HOOMD-rs. 126

5.4 **Component architecture of the hoomd-chimes crate in Figure 5.1** The elliptical nodes represent Rust types, and the square nodes represent Rust traits. The Builder/Parser is the top-level type under hoomd-chimes that I expect the user would use. 129

5.5 **Example documentation of *hoomd-chimes*** A snapshot of documentation of the ChebyshevExpansion type in Figure 5.4. 130

5.6 **The radial distribution function of two-body carbon model using ChIMES model in HOOMD-rs** 131

LIST OF TABLES

Table

3.1 Acronym Guide 43

3.2 ChIMES and extended ChIMES (X-CHIMES) fitting parameters 59

3.3 PGNs capping layer thickness λ for particles with different polymer lengths α . . . 66

3.4 interaction parameter 76

4.1 Metadynamics parameters used in this study. 104

ABSTRACT

The self-assembly of colloidal nanoparticles into complex structures is a foundational aspect of nanotechnology. It enables the design of advanced materials with tailored properties. Recently, building self-assembled structures from sub-micron-sized particle building blocks has garnered significant attention because of their broad potential applications in nanotechnology. Despite these promising advances, several challenges remain in (1) elucidating and, eventually, engineering assembly pathways to optimize target structures via robust self-assembly structures characterization techniques. (2) Researchers must also develop effective design strategies to help navigate through the vast design space of highly customizable building blocks. This dissertation advances the field through interconnected studies. These leverage machine learning (ML), physics-informed coarse-grained modeling, and novel forward design strategies to investigate and optimize colloidal nanoparticle assembly. It also describes the development of a new molecular simulation engine. By addressing critical challenges in structural characterization, modeling tunable interactions, exploring the relationship between particle design and structures, and developing a new simulation tool for nanoparticle design, this work provides innovative tools and frameworks to enhance the design of self-assembled nanomaterials.

The first study introduces a physics-agnostic ML approach to analyze local environments in colloidal self-assembly [J. Chem. Phys. 160, 154102 (2024)]. A multilayer perceptron (MLP) classifier, trained with a shape symmetry-encoded data

augmentation technique, identifies structural order in systems of diverse particle shapes without relying on traditional symmetry functions. Evaluated across scenarios like cubic structures, patchy particle systems, hexagonal bipyramids, and truncated shapes, this classifier demonstrates robust performance and flexibility, offering a powerful tool for structure identification in particle-based systems where orientations are critical.

The second study, a collaborative effort, develops an alchemical machine learning-interaction model (ML-IAM) to describe polymer-grafted nanoparticle (PGN) interactions [J. Chem. Theory Comput. 2025, 21, 19, 9853–9867]. It accounts for both interparticle distances and tunable attributes like polymer length and grafting density. This model extends the ML-IAM ChIMES model. The resulting extended ChIMES (X-ChIMES) model is trained on PMF data for PGNs with varied polymer ligand length. We enable efficient training data generation using the forward-reverse steered molecular dynamics approach in HOOMD-blue. We demonstrate the efficacy of ChIMES for generating CG models for PGNs with fixed design attributes. The development of the X-ChIMES model and its application enhances digital alchemy inverse-design simulations using HOOMD-blue and the ChIMES Calculator. This is the first application of ChIMES to model CG systems and to couple with an inverse design method to target nanomaterial self-assembly.

The third study presents Exploratory Digital Alchemy (EDA), an enhanced forward design framework. EDA builds on the generalized ensemble of the Digital Alchemy (DA) method by integrating the self-exploration power of Well-Tempered Metadynamics. This helps overcome DA's limitation of requiring a predefined target structure. Applied to systems with the Lennard-Jones-Gauss and Oscillating Pair Potentials, EDA identifies stable and metastable structures, along with their optimal

particle design. EDA can also reveal metastability among phases in the generalized ensemble. This fusion of DA with exploratory sampling transforms it into an effective forward design scheme and enables the discovery of novel colloidal crystals.

Finally, this dissertation presents my work on developing a next-generation molecular simulation engine called HOOMD-rs. This engine is a spiritual and technical successor to HOOMD-blue, but is implemented from the ground up in Rust. The HOOMD-blue design concept is redefined and adopted in HOOMD-rs to enhance flexibility in customizing simulations. By leveraging Rust's unique memory safety features and expressive type system, HOOMD-rs aims not only to replicate existing functionality, but to redefine what is possible in high-performance scientific computing.

Together, these studies provide a comprehensive toolkit for advancing colloidal nanoparticle design, with broad implications for materials science and nanotechnology.

CHAPTER I

Introduction

1.1 Prologue

The evolution of materials science has long been defined by the manipulation of atomic and molecular building blocks. However, the past two decades have witnessed a paradigm shift: the rise of nanoparticles as programmable atom equivalents (PAEs) [2, 3]. Unlike atoms, which are constrained by fixed electronic structure and valence, nanoparticles offer near-infinite tunability in size, shape, composition, surface chemistry, and interaction anisotropy. This malleability enables the bottom-up construction of complex, functional nanomaterials through self-assembly-the spontaneous organization of disordered components into ordered structures driven solely by thermodynamics [4].

Self-assembly is not a new phenomenon. It has been discovered and studied across scales: biomolecule self-assembly to produce target structures through adjusting the intermolecular interaction [5], including nanofibers [6, 7], virus-like particles [8, 9, 10], and DNA nanostructures assembled by DNA origami [11, 12]. The work of Mirkin and Alivisatos introduced DNA-functionalized gold nanoparticles as programmable building blocks, yielding crystalline lattices with programmable symmetry and lattice constants [13, 14].

In addition, extending this concept beyond sequence-specific interactions, shape anisotropy alone can drive assembly into complex structures. Simulations and experiments have shown that hard polyhedra—particles interacting via excluded volume—self-assemble into dense packings with liquid crystalline, quasicrystalline, and crystalline phases [15]. Octahedra assemble into Minkowski lattice phase with denest packing [16], dodecagonal quasicrystals assembled by hard tetrahedra [17], and elongated polyhedra yield smectic and plastic crystal phases [18]. These findings, rigorously mapped by the Glotzer group, reveal that shape entropy dominates assembly in hard particle systems [19, 20, 21].

Beyond shape, surface patterning introduces directional interactions. Patchy particles, the colloids decorated with attractive or repulsive patches—mimic molecular orbitals or valence [22]. Hydrophobic patches drive aggregation via capillary forces [23], DNA-coated patches enable programmable bonding [24], and Janus particles with asymmetric surface chemistry form clusters, chains, and gels [25]. A breakthrough in conceptualizing these interactions came with entropic bonding [26, 27], where directional depletion forces between faceted particles create bond-like specificity without enthalpic attraction.

This extraordinary malleability of nanoparticles, while powerful, poses a fundamental inverse design challenge: identifying the optimal nanoparticle attributes—shape, interaction potential, or surface chemistry—that reliably produce a target self-assembled structure (SAS). The vast parameter space renders brute-force forward design inefficient, akin to searching for a needle in a needlestack. Inverse design frameworks address this by autonomously optimizing building blocks to maximize the stability of a user-specified SAS. Notable approaches include entropy maximization [28, 29] and Digital Alchemy (DA) [30], a statistically rigorous method developed in the

Glotzer Lab that treats particle attributes as thermodynamic variables within extended ensembles. DA has successfully optimized nanoparticle shape [30, 31], patch patterns [32], and interaction potentials [33, 34].

Despite its robustness, the DA framework exhibits several limitations that motivate this dissertation. First, robust structure identification is lacking, particularly for systems involving particle shapes with orientation degrees of freedom and the resulting complex crystal structures. Second, DA relies on closed-form interaction functionals, which are not always available for chemically realistic systems such as polymer-grafted nanoparticles. Third, as an inverse design method, DA requires prior knowledge of the target structure, which is unsuitable for exploratory studies where the target structure is unknown. Fourth, simulations based on DA necessitate sampling fluctuations of the design parameter to optimize it. The design parameter does not possess a universal meaning, unlike particle position or orientation, across every system. Therefore, a highly customizable simulation engine is required to accommodate various DA-related simulations. This dissertation addresses these challenges through three synergistic studies that advance analysis, modeling, and exploration in colloidal self-assembly.

1.2 Order Parameters: Reducing Complexity in Many-Body Dynamics

In a typical self-assembly simualtion, involving systems with $N \sim 10^3$ to 10^5 particles, each with translational and rotational degrees of freedom (DOFs). The full configuration space has dimensionality $6N$, causing direct analysis intractable. Moreover, most DOFs are irrelevant to macroscopic events like nucleation or phase coexistence. To extract physical insight, we project the system onto a low-dimensional manifold as order parameters (OPs), $s = f(\mathbf{R}, \mathbf{q})$, where $\mathbf{R}$ and $\mathbf{q}$ are particle po-

sitions and orientations, and the f is the function that performs the projection. Ideally, OPs should have low dimensionality, such that $s \in \mathbb{R}^d$ with $d \ll 6N$. A classic example of OP for studying crystal nucleation and crystalization is the Steinhardt parameters [35], rotationally invariant combinations of spherical harmonics, $Q_l = \sqrt{\frac{4\pi}{2l+1}\sum_{m=-l}^{l}|q_{lm}|^2}$, where q_{lm} are bond orientations using spherical harmonics expansion. For a shape-anisotropic particle system, traditional OPs like the Steinhardt parameters would fail. Imagine that octahedrons may form the same body-centered cubic crystal that has identical Q_6, despite one of them could be a plastic crystal, i.e., particles can freely rotate while staying at the lattice sites. Q_6 cannot capture such orientation disordering. This motivates machine learning classifiers in this dissertation (Chapter II) that use raw geometric features (distances, angles, orientations) to distinguish local environments.

Having discussed the OPs as low dimensional variable that captures important macroscopic event. Let's now turn our focus to the Boltzmann distribution as a fucntion of OPs, which is the heart of the statistical mechanics description of any given system. In statistical mechanics, the Boltzmann distribution over OPs is:

$$\pi(s) = \int d\mathbf{R}\ \delta(s - f(\mathbf{R}))\pi(\mathbf{R}), \tag{1.1}$$

where $\pi(\mathbf{R}) = e^{-\beta U(\mathbf{R})}/Z$ is the canonical distribution with the potential energy $U(\mathbf{R})$ and Z is the canonical partition function with the temperature $1/\beta$. Now, the free energy surface (FES) can be written as,

$$F(s) = -\frac{1}{\beta}\ln \pi(s) \tag{1.2}$$

This form of free energy surface as a function of OPs first appeared in Landau's theory of phase transitions, often referred to as Landau free energy or a similar concept called the potential of mean force (PMF). The FES governs the system's

thermodynamics, such that its minima correspond to stable phases and the barrier between stables phases determined the transition rate via the Arrhenius equation. Accurate OPs are thus essential for both structure identification and free energy calculation.

1.3 Bias Simulation: Overcoming Barriers in Configuration Space

Computing the FES along an order parameter s requires accurate sampling of the Boltzmann distribution $\pi(s)$. To perform sampling, we would run a long molecular dynamics (MD) or Monte Carlo (MC) simulation, record the value of s with certain frequency, and build a histogram. Assuming the simulations are ergodic, we estimate the Boltzmann distribution using the histogram as:

$$\pi(s) = \lim_{N\to\infty} \frac{\sum_{i=1}^{N} \delta(s - f(\mathbf{R}_i))}{N} \tag{1.3}$$

where N is the number of sampled configurations. Taking the negative logarithm of distribution estimate then yields the FES estimate as: $F(s) = -\frac{1}{\beta} \ln \pi(s)$. In practice, this approach fails for most systems due to the non-ergodic simulation that violates the assumption I made. Consider a system whose free energy barriers are tens of $k_B T$—separate metastable states (e.g., liquid vs. crystal). The time to cross such a barrier scales exponentially according to the Arrhenius equation: $\tau \sim \tau_0 e^{\beta \Delta F}$, where $\tau_0 \sim$ picoseconds. For $\Delta F = 20 k_B T$, this means $\tau \sim$ milliseconds to seconds—far beyond the reach of most of the simulations.

Bias sampling methods solve the non-ergodicity problem by modifying the potential energy landscape to accelerate rare events. Instead of sampling the true Hamiltonian $H = K(\mathbf{p}) + U(\mathbf{R})$, we introduce a bias potential $V(s)$ that depends only on the order parameter:

$$H_{\text{bias}} = K(\mathbf{p}) + U(\mathbf{R}) + V(s) \tag{1.4}$$

This distorts the probability distribution to $\pi_{\text{bias}}(s)$, but the unbiased distribution can be recovered using the importance sampling reweighting scheme:

$$\pi(s) = \frac{\pi_{\text{bias}}(s)\, e^{\beta V(s)}}{\int ds'\, \pi_{\text{bias}}(s')\, e^{\beta V(s')}}.$$

Here, I briefly review three methods which I used in this dissertation and to illustrate the spectrum of bias strategies.

1. **Umbrella Sampling (US)**: US divides the OP space into overlapping windows and applies a harmonic restraint in each as: $V_i(s) = \frac{1}{2}k(s - s_i)^2$. By running many short simulations at different s_i, we obtain biased histograms that are later stitched together using the Weighted Histogram Analysis Method (WHAM) [36]. WHAM solves a self-consistent equation to produce a smooth, unbiased $\pi(s)$ across the full range.

2. **Steered Molecular Dynamics (SMD) and Jarzynski Equality**: Instead of sampling equilibrium in fixed windows, SMD pulls the system along s using a time-dependent restraint: $V(s,t) = \frac{1}{2}k\left(s - s_0 - ut\right)^2$, where u is the pulling speed. The non-equilibrium work w done on the system is recorded, and Jarzynski's equality [37] connects the non-equilibrium work to equilibrium free energy: $e^{-\beta\Delta F} = \langle e^{-\beta w}\rangle$. Averaging over many fast pulling trajectories yields ΔF. The forward-reverse (FR) method [38] improves accuracy by combining forward (pulling apart) and reverse (pushing together) paths.

3. **Metadynamics (MetaD)**: MetaD takes a dynamic, adaptive approach. It builds a history-dependent bias potential by depositing small Gaussian hills whenever

the system visits a state s_t during the course of simulation as:

$$V(s, t+\tau) = V(s,t) + W_t \exp\left[-\frac{(s-s_t)^2}{2\sigma^2}\right] \tag{1.5}$$

with adaptive height:

$$W_t = \omega e^{-\beta V(\mathbf{s}_t)/(\gamma-1)} \tag{1.6}$$

where $V(s,0) = 0$ and the update of bias potential is performed periodically every τ timestep. The s_t are the set of collective variables calculated from the system evaluated at timestep t, ω is the Gaussian height, σ controls the width of the Gaussian bias, and γ is a user-defined parameter controlling the adaptation of the Gaussian height. In well-tempered MetaD [39], the bias eventually cancels $(1 - 1/\gamma)F(s)$, encouraging the system to explore high free energy regions. The result is a flattened effective landscape in s-space, with enhanced diffusion over barriers. Well-tempered MetaD has a unqiue reweighting scheme to recover the true FES [40].

In this dissertation, I will demonstrate that SMD generates accurate two- and three-body PMFs for polymer-grafted nanoparticles (Chapter III). Additionally, the Meta-inspired scheme powers a new forward design framework (Chapter IV).

1.4 Digital Alchemy: Inverse Design via Extended Ensembles

Among inverse design methods, Digital Alchemy (DA) [30] stands out for its elegance and statistical mechanical foundation, and can be viewed as one type of bias simulation techniques. Rather than guessing particle attributes and simulating forward, DA inverts the problem: it treats design variables as thermodynamic DOFs and uses simulation to discover the optimal building block that stabilizes a target self-assembled structure (SAS). The core idea is the sampling in the extended thermodynamics ensemble. Imagine gradually changing a sphere into a cube, or tuning

the depth of an attractive well. In DA, such changes are not ad hoc—they are continuous thermodynamic paths in an extended phase space. A design parameter α (e.g., shape truncation, potential width, grafting density) is promoted to a dynamic variable, paired with a conjugate alchemical potential μ_α. The system now evolves in $(\mathbf{r}, \mathbf{p}, \alpha)$ phase space. In the extended ensemble of DA, the system is described by a biased alchemical Hamiltonian:

$$H(\boldsymbol{r}, \boldsymbol{p}, \alpha) = K(\boldsymbol{p}) + U(\boldsymbol{r}, \alpha) + V^{(\text{crystal})}(\boldsymbol{r}, t) \tag{1.7}$$

where K is the kinetic energy, U is a physical interaction potential, now α-dependent, and $V^{(\text{crystal})}$ is the time-dependent harmonic lattice constraint that pins particles to the target SAS lattice sites, similar to the one in US method:

$$V^{(\text{crystal})}(\boldsymbol{r}, t) = \frac{1}{2} k(t) \sum ||\boldsymbol{r} - \boldsymbol{r}^{(\text{crystal})}||^2 \tag{1.8}$$

The spring constant $k(t)$ decreases linearly from k_0 to zero over a time t_{end} to release the constraint eventually:

$$k(t) = \begin{cases} k_0(t_{\text{end}} - t)/t_{\text{end}}, & 0 \le t < t_{\text{end}} \\ 0, & t \ge t_{\text{end}} \end{cases} \tag{1.9}$$

This anneals the bias, ensuring the final state reflects the true free energy landscape. While the MD realization is possible [41], conventionally, the sampling is performed using Alchemical Monte Carlo (Alch-MC) in the NVTμ_α ensemble. Proposed moves include: 1. Particle translation/rotation moves like standard MC method, and 2. α moves as alchemical moves. The acceptance of Alch-MC move P_{acc} follows an extended Metropolis criterion : $P_{\text{acc}} = \min\left[1, \exp\left(-\beta[\Delta H - \mu_\alpha \Delta\alpha]\right)\right]$. Intuitively, the goal of applying DA via Alch-MC is to optimize α, which corresponds to a maximized likelihood of producing the specified SAS. To achieve this, we maximize

the extended Boltzmann distribution π_α:

$$\pi_\alpha = \frac{1}{\mathcal{Z}} \exp\left[-\beta(H - \mu_\alpha \alpha)\right] \tag{1.10}$$

where $\mathcal{Z}$ is the extended partition function that governs the extended ensemble.

1.5 Outlines

The remainder of this dissertation is organized as follows: I begin with a study in Chapter II, which introduces a physics-agnostic ML approach to analyze local environments in systems of particle shapes. A multilayer perceptron classifier, trained with a shape symmetry-encoded data augmentation technique, identifies structural order in systems of diverse particle shapes without relying on traditional symmetry functions. Then, in Chapter III, I detail my second study, as a collaborative effort, which develops an alchemical machine learning-interaction model to describe polymer-grafted nanoparticle interactions as a function of both interparticle distances and tunable attributes such as polymer length and grafting density simultaneously. Such a framework can be used to perform inverse design of PGN. Next, I demonstrate Explortory Digital Alchemy in Chapter IV, an enhanced forward design framework that builds on the generalized ensemble of the Digital Alchemy method by integrating the self-exploration power of Well-Tempered Metadynamics to overcome DA's limitation of requiring a predefined target structure, allowing for the simultaneous discovery of stable phases and their optimal building blocks. In Chapter V, I present my work that contributes to the development of a next-generation molecular simulation engine, entitled HOOMD-rs, a spiritual and technical successor to HOOMD-blue. It introduces a performant, memory-safe, and highly customizable simulation framework built in Rust. Finally, Chapter VI provides a summary of the major contributions of this dissertation and offers an outlook on future research directions for

the computational design of nanomaterials.

CHAPTER II

Classification of Complex Local Environments in Systems of Particle Shapes Through Shape-Symmetry Encoded Data Augmentation

This chapter is reproduced from *Classification of complex local environments in systems of particle shapes through shape-symmetry encoded data augmentation* [42], a paper published in *The Journal of Chemical Physics* by Shih-Kuang (Alex) Lee, Sun-Ting Tsai, and Sharon C. Glotzer †. († indicates the corresponding author)

2.1 Introduction

Self-assembly is studied extensively in such fields as physics, chemistry, materials science, chemical engineering and biology [43, 44]. A fundamental process involving thermodynamics and kinetics, self-assembly refers to the formation of ordered structures from individual building blocks or particles without direction from an external field. In recent years, the development of self-assembled structures from sub-micron-sized particle building blocks has attracted considerable attention due to their potential applications in nanotechnology [45, 46, 47, 48, 49, 50, 51, 52]. An important challenge in elucidating and, eventually, engineering assembly pathways to optimize target structures is defining appropriate order parameters that quantify local order in the assembling structures along the pathway [53, 54, 55, 56, 57]. Defining suitable

local order parameters is particularly challenging when the self-assembling structure is complex (e.g. a large unit cell, possessing chirality, etc.) or when competing polymorphs or pre-nucleation motifs emerge along the pathway [58, 59, 60].

Previous studies have attempted to identify suitable order parameters for self-assembly, including using the radial distribution function, the bond-orientational order parameter [61, 62, 57, 63], and Voronoi tessellation [64, 61, 65]. While these order parameters have been used extensively and successfully in capturing some aspects of the assembly process, they have limitations. For example, the radial distribution function (RDF) considers only the pairwise (two-point) correlation between particles and does not capture higher-order correlations. The bond-orientational order parameter is sensitive to local structure but is less effective in detecting global ordering. Methods using the Voronoi tessellation can elucidate local structure but are less effective in describing long-range order. In particular, these approaches are insufficient for interrogating, e.g., the self-assembly of a system of truncated tetrahedra into a crystal with 432-particle unit cell [59].

Machine learning (ML) is becoming an increasingly popular approach for discovering order parameters useful in the study of self-assembly [66, 67, 68, 54, 69, 55, 56, 70]. One ML approach uses existing order parameters that combine roto-translation-invariant symmetry functions as input descriptors and approximates optimal order parameters [54, 71]. However, an immediate shortcoming of this type of supervised ML method is its limited classification capabilities, which are constrained by its input descriptors. If none of the input descriptors can classify a certain crystal structure, the machine learning methods will fail to classify that crystal structure. Training deep neural networks (NNs) to classify crystal phases is another increasingly common approach [72, 73, 57, 63, 74, 75]. Deep NNs can identify nonlinearity and are

especially promising in constructing order parameters from particle features, such as instantaneous positions along a trajectory. Since thermal fluctuations often lead to noisy data, however, applying symmetry functions to encode particle positions is usually inevitable [76]. These pre-engineered symmetry functions can work well for some systems, but finding symmetry functions that encode particle orientations is non-trivial. Finally, graph neural networks (GNNs) have also become popular for classifying crystal structures [77, 78, 79]. By treating crystal structures as connected graphs, GNNs preserve permutation symmetry. However, they often require additional manipulation to deal with rotational and translational symmetry [71]. Although equivariant GNNs have recently been proposed [80], the question of how to incorporate equivariant properties for particle shapes remains to be addressed. Common to all of these ML approaches is the tendency for increasing complexity in the approach as the building blocks and the structures they form become increasingly complex.

In this work, we show how we can use the most basic NN, a multilayer perceptron (MLP), as a local environment classifier in systems of particle shapes. Because we use the MLP classifier to quantify the local environment around each particle, the classifier is permutation-invariant, precluding the use of more sophisticated network structures such as GNNs [79]. Thus, instead of employing conventional symmetry functions to transform per-particle quantities to input descriptors, we analyze step-by-step the symmetry of a particle's shape and propose a straightforward shape-symmetry encoded data augmentation method that allows our MLP classifier to operate on per-particle features directly. This data augmentation method ensures the roto-translation invariance of input features to the global environment and accounts for the particle shape's symmetry. Importantly, the simplicity of our data

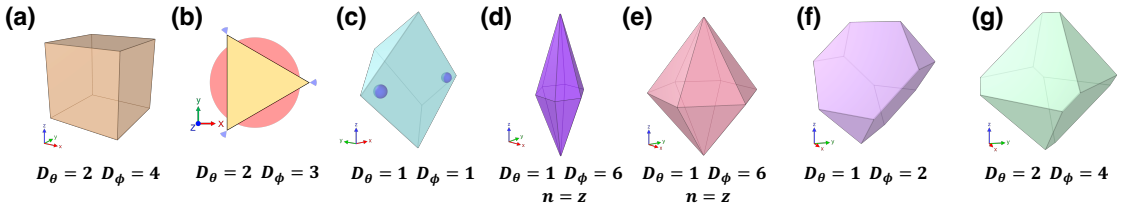

Figure 2.1: **Systems of particle shapes** The particle systems studied in this paper: **(a)** cube, **(b)** patchy triangle, **(c)** patchy triangular prism [1], **(d)** hexagonal bipyramid with aspect ratio $\alpha = 3.0$, **(e)** hexagonal bipyramid with $\alpha = 1.28$, **(f)** truncated tetrahedron, and **(g)** truncated octahedron. In (a) and (b), the transparent blue region decorating the particles indicates the attractive patch, while the transparent red region indicates a repulsive patch. Also shown are the shape symmetry-related factors D_θ, D_ϕ, and mirror plane normal vector $\mathbf{n}$ utilized for each system. Each particle pictured here is oriented such that it is in the reference orientation given by the quaternion $\mathbf{q}_0 = (1, 0, 0, 0)$.

augmentation method necessitates minimal manipulation of the raw data, and also leads to a substantial improvement in the classification performance and flexibility when distinguishing different thermodynamic phases in our test systems. In this way, our approach can provide a simple, powerful, physics-agnostic alternative to conventional order parameters when studying self-assembly.

2.2 Methods

We construct a classifier for the local environment around a particle using an MLP that incorporates continuous, discrete and shape symmetry information. By building an MLP classifier that learns to classify each particle, we ensure that the classification is permutation invariant. We first extract features from each particle's local environment and then use these features to train MLP classifiers to accurately classify particles based on their local environment.

2.2.1 Per-Particle Quantities as Input Features

To account for translational symmetry and describe the local environment of a target particle in a system of like particles, we employ a set of interparticle quantities

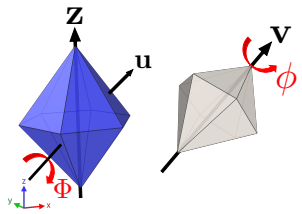

Figure 2.2: **Rotation in the local environment** The blue bipyramid represents the target particle, and the white bipyramid represents a neighboring particle.

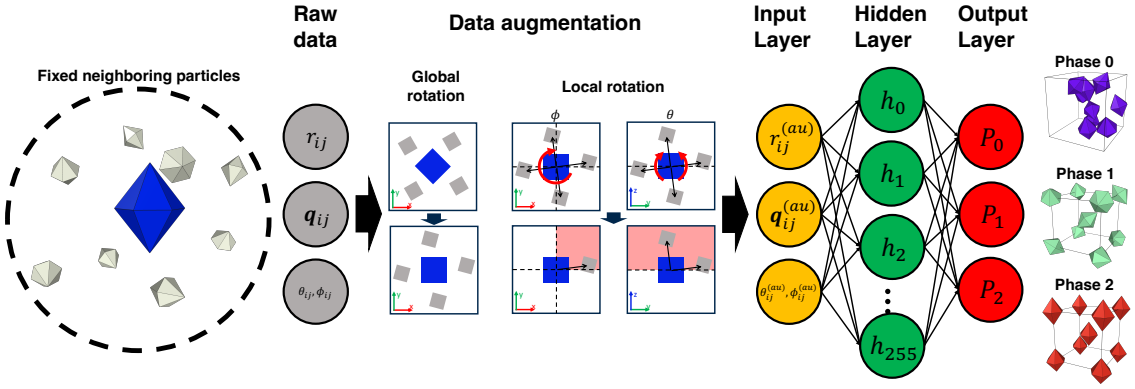

Figure 2.3: **The data augmentation and the architecture of Multilayer Perceptron (MLP)** The diagram of the data augmentation and the MLP architecture used to classify local particle environment. The raw local environment data is augmented through sets of rotations and passed to the input layer of MLP. The input layer has N_b+$4N_b$+$2N_b$ units as described in Equation 2.9, and the hidden layer contains 256 units. The number of output layers depends on the system. For example, we will present results for a system of particles undergoing a two-steps crystallization transition, where the output layer is composed of three units representing the probability that a given particle belongs to one of three predefined phases.

defined with respect to the surrounding neighborhood. Specifically, we calculate the **relative positions** $\mathbf{r}_{ij}$ and **relative orientations** defined by quaternion $\mathbf{q}_{ij}$ of a predetermined number of neighboring particles j in relation to the target particle i. These quantities serve as a unique fingerprint of the local environment surrounding the target particle [81, 57] and are defined below.

We define the relative position $\mathbf{r}_{ij}$ using spherical coordinates $(r_{ij}, \theta_{ij}, \phi_{ij})$, where the relative distances r_{ij} between a fixed number N_b of neighboring particles denoted by the index j (that is, the size of the neighborhood) and the target particle i contain

information about the local density distribution. The angular part (θ_{ij}, ϕ_{ij}) define the local bond angles. Note that for a two-dimensional system, θ_{ij} is always 0. However, despite the invariance of the relative distances r_{ij} to arbitrary translation operations, these distances can still be affected by thermal fluctuations and the length scale of the system, which can impact the transferability of our classifier. Thus, we normalized the relative distances:

$$r_{ij}^{(\mathrm{au})} = \frac{r_{ij}}{\max_{j \in N_b} r_{ij}} \tag{2.1}$$

where the superscript $^{(\mathrm{au})}$ denotes the augmented features.

The relative quaternion $\mathbf{q}_{ij}$ is commonly used to define the relative orientation of a neighboring particle j with respect to the target particle i. The quaternion plays an important role in both simulating and analyzing crystallization of a system of anisotropic particles [82, 83] and in defining the space group of crystals of certain molecular systems. To describe a particle's orientation, a reference orientation first needs to be established using the important symmetry axis of the particle along the orthogonal basis of Cartesian coordinates. We denote the reference orientation by $\mathbf{q}_0 = (1, 0, 0, 0)$, which is equivalent to the identity rotation with respect to the reference orientation. In Figure 2.1, we illustrate the particles studied in this paper by placing them in the predefined reference orientation. Given this reference orientation, we can define subsequent orientations of each particle by performing a 3-dimensional spatial rotation, which can be expressed as a rotation quaternion $\mathbf{q} = (C, Su^x, Su^y, Su^z)$ where $(C, S) = (\cos(\theta/2), \sin(\theta/2))$. This $\mathbf{q}$ thus represents a rotation angle θ from the reference orientation about the axis $\mathbf{u}$. The relative orientation between target particle i and its neighborhood j can then be expressed as the rotation quaternion via the conventional rotation from i to j as:

$$\mathbf{q}_{ij} = \mathbf{q}_j \mathbf{q}_i^{-1} \tag{2.2}$$

By utilizing relative position and orientation, we can accurately capture each particle's local environment while maintaining translational invariance. We have yet to discuss the property of invariance to an arbitrary rotation of the system, which can influence the angular parts θ_{ij}, ϕ_{ij} that are used to define relative position and $\mathbf{q}_{ij}$. Conventionally, the rotational invariance is achieved by randomly rotating the training set in each training epoch [78]; this step is referred to as data augmentation. In the next section, we present a more robust approach for training a classifier invariant to an arbitrary system rotation of particle environments.

2.2.2 Shape-Symmetry Encoded Data Augmentation

In the previous section, we introduced relative positions and quaternions as input features. These input features are designed so they are invariant to the translation of the system. However, arbitrary rotation of the system can also limit the transferability of the classifier, which is traditionally addressed using data augmentation techniques [69]. Note that our data augmentation is different from the data augmentation used in the training of convolution neural networks, where random orientations are applied to generate modified copies of image data to allow the network to recognize different symmetry of the images. For example, in image processing, one duplicates the image but with random rotations to create a training dataset to prevent possible overfitting and enhance the transferability of the classifier. Recent developments in equivariant NNs (ENNs) also allow for input without data augmentation [80, 84, 85]. For systems of point particles, the final crystal structure can be used to define the reference orientation. Here we exploit particle shape and use

the user-defined reference orientation represented by quaternions to define the local environment. To ensure rotational invariance, we independently rotate each target particle and its local environment onto the predefined reference orientation, such that each target particle is in the predefined reference orientation as shown in Figure 2.1 prior to calculating the per-particle quantities.

In addition to translational and rotational symmetries associated with the system of particles, the individual particles can possess symmetries. With a sophisticated design, the ENNs can process these additional symmetries during training. However, we propose a simple yet effective way to achieve the same result here. We perform an additional data augmentation that encodes the particle's shape and interaction anisotropy (patchiness). The new angular part of the relative position and relative quaternion after this data augmentation is denoted as $(\theta_{ij}^{(\mathrm{au})}, \phi_{ij}^{(\mathrm{au})})$ and $\mathbf{q}_{ij}^{(\mathrm{au})}$. Through simple geometric reasoning, the augmented angular part can be calculated easily as follows:

$$(\theta_{ij}^{(\mathrm{au})}, \phi_{ij}^{(\mathrm{au})}) = \left(\frac{\theta_{ij}}{\pi} \bmod \frac{1}{D_\theta}, \frac{\phi_{ij}}{\pi} \bmod \frac{2}{D_\phi} \right) \tag{2.3}$$

where $\bmod$ is the modulo operator, and D_ϕ and D_θ are discrete integers that assume the value, N of the N-fold rotational symmetry along ϕ and θ-direction with respect to the particles' shape when it is in the reference orientation. In Figure 2.1, we show D_ϕ and D_θ for each of the corresponding hard shapes and patchy particles in their reference orientations.

The orientation quaternion $\mathbf{q}_{ij}$ ican be written as a rotation of angle Φ along an axis $\mathbf{u}$, as shown in Figure 2.2 It can be proved that the same quaternion can also be written as the rotation of angle ϕ along axis $\mathbf{v}$,

$$\mathbf{q}_{ij} = e^{i\frac{\Phi}{2}\mathbf{u}} = e^{i\frac{\phi}{2}\mathbf{v}}\mathbf{Q} \tag{2.4}$$

where $\mathbf{v} = \mathbf{Q}\mathbf{z}\mathbf{Q}^{-1}$ defines the symmetry axis of the particle. The symmetry axis is defined as the z-axis when each particle is in its reference orientation and rotates in the same way the particle rotates. The augmented quaternion is then calculated as

$$\mathbf{q}_{ij}^{(\mathrm{au})} = e^{i\frac{\phi^{(\mathrm{au})}}{2}\mathbf{v}^{(\mathrm{au})}} \tag{2.5}$$

where $\phi^{(\mathrm{au})} = \phi \bmod \frac{2\pi}{D_\phi}$ and

$$\mathbf{v}^{(\mathrm{au})} = \mathbf{v}_{\perp\mathbf{z}} + \cos\theta^{(\mathrm{au})}\mathbf{z} \tag{2.6}$$

where $\mathbf{v}_{\perp\mathbf{z}}$ is defined as the component perpendicular to $\mathbf{z}$ and $\theta^{(\mathrm{au})} = \arccos(\mathbf{v}\cdot\mathbf{z})$ $\bmod \frac{\pi}{D_\theta}$.

Under the same rule, it is also straightforward to consider mirror symmetry. For the angular part, in general, we can calculate augmented relative positions $\mathbf{r}_{ij}^{(\mathrm{au})}$ before separating relative positions into distance r_{ij} and angular parts (θ_{ij}, ϕ_{ij}):

$$\mathbf{r}_{ij}^{(\mathrm{au})} = (\mathbf{r}_{ij})_{\perp\mathbf{n}} + \|\mathbf{r}_{ij}\cdot\mathbf{n}\|\mathbf{n} \tag{2.7}$$

where $(\mathbf{r}_{ij})_{\perp\mathbf{n}} = \mathbf{n}\times(\mathbf{r}_{ij}\times\mathbf{n})$ is defined as the component perpendicular to $\mathbf{n}$. In practice, since the mirror plane of a hexagonal bipyramid is along the $\mathbf{z}$-axis, we need only to take the absolute value of the $\mathbf{z}$-coordinate before separating relative positions into distance and angular parts. For the quaternion part, we need to separate the symmetry axis into normal and parallel parts with respect to the plane normal vector $\mathbf{n}$:

$$\mathbf{v}^{(\mathrm{au})} = \mathbf{v}_{\perp\mathbf{n}} + \|\mathbf{v}\cdot\mathbf{n}\|\mathbf{n} \tag{2.8}$$

where $\mathbf{v}_{\perp\mathbf{n}} = \mathbf{n}\times(\mathbf{v}\times\mathbf{n})$ is defined as the component perpendicular to $\mathbf{n}$.

2.2.3 Multilayer Perceptron (MLP) as Local Environment Classifier

The local symmetry is broken during a self-assembly process as the local environment around a particle changes. To monitor the change in the local environment, we utilize a fully connected NN, commonly known as a multilayer perceptron (MLP), to classify this local environment, as illustrated in Figure 2.3. There are several advantages of using a simple MLP instead of a more advanced GNNs or ENNs. First, an MLP handles large datasets with simplicity and effectiveness. An MLP allows us to train the machine even on a personal laptop, suitable for a quick, on-the-fly test. Additionally, an MLP can be easily extended to incorporate additional features or classify particles in other colloidal systems. Second, a simple MLP can be greatly accelerated by harnessing the power of modern graphical processing units (GPUs), which can facilitate future research, such as using MLPs as order parameters to study the assembly pathways of disparate systems forming the same structure, or in enhanced sampling methods such as umbrella sampling or metadynamics whose algorithms demand efficient calculation of order parameters and their derivatives. Third, despite its simple network structure, a MLP still provides sufficient nonlinearity to build a powerful classifier from fundamental features, e.g., particle coordinates and orientations. Its classification ability is decent enough to map local environmental fingerprints of particle shapes to thermodynamic phases.

The input to the MLP classifier is a set of feature vectors $(r_{ij}^{(\mathrm{au})}, \mathbf{q}_{ij}^{(\mathrm{au})}, \theta_{ij}^{(\mathrm{au})}, \phi_{ij}^{(\mathrm{au})})$, as described in Section 2.2.1 and 2.2.2. These feature vectors are arranged and concatenated to a 1-dimensional vector $\mathbf{x}_i$ with the relative position and relative quaternion sequentially placed in ascending order as follows,

$$\mathbf{x}_i = \mathrm{sort}_{r_{ij}^{(\mathrm{au})}}(..., r_{ij}^{(\mathrm{au})}, \mathbf{q}_{ij}^{(\mathrm{au})}, \theta_{ij}^{(\mathrm{au})}, \ \phi_{ij}^{(\mathrm{au})}, \ ...) \tag{2.9}$$

where j runs over all neighbors and therefore $\mathbf{x}_i \in \mathbb{R}^{N_b+4N_b+2N_b}$. As shown in Figure 2.3, the MLP architecture includes one hidden layer consisting of 256 neurons. The input layer of the network takes the feature vectors $\mathbf{x}_i$ and propagates them forward to the hidden layer, which performs a linear operation followed by a non-linear activation function σ:

$$\mathbf{h} = \sigma(\mathbf{W}\mathbf{x}_i + \mathbf{b}) \tag{2.10}$$

The activation function comprises a linear operation defined by a layer-wise matrix multiplication with trainable weight matrix $\mathbf{W}$ and bias vector $\mathbf{b}$.

The hidden layer $l = 0$ takes input feature vectors $\mathbf{x}_i$ directly, and therefore has dimensions $\mathbf{W} \in \mathbb{R}^{\mathcal{M}\times\mathcal{N}}$ and $\mathbf{b} \in \mathbb{R}^{\mathcal{M}}$, where $\mathcal{M} = 256$ and $\mathcal{N} = N_b + 4N_b + 2N_b$. After this linear operation, we used a rectified linear unit (ReLU) as the activation function σ. The output from the hidden layer is then converted to output nodes $\mathbf{y} \in \mathbb{R}^{\mathcal{C}}$ in the output layer,

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}^{(o)}\mathbf{h} + \mathbf{b}^{(o)}) \tag{2.11}$$

where $\mathbf{W}^{(o)} \in \mathbb{R}^{\mathcal{M}\times\mathcal{C}}$ and $\mathbf{b}^{(o)} \in \mathbb{R}^{\mathcal{C}}$ are the output weight matrix and output bias, respectively, and $\mathcal{C}$ is the number of pre-defined classes. The softmax function in Equation 2.11 is a mathematical function that converts a vector of real numbers into a probability distribution. The MLP classifiers are then trained using the optimizer Adam [86], with a learning rate of 7.5×10^{-4}, and with an error metric given by the cross-entropy loss L, defined as:

$$L = -\sum_{n=1}^{\mathcal{C}} \log \hat{\mathbf{y}}_n \tag{2.12}$$

The cross-entropy loss L, or log loss, measures the performance of a probabilistic classification model whose output is a probability distribution. For simplicity, all of our MLP classifiers are trained with 30 epochs, where the number of epochs is defined as the number of times the optimization algorithm goes through all training samples. The classifier and training algorithm are both implemented using PyTorch [87].

2.2.4 Data Preparation

We generated independent training and testing trajectories for all seven test systems. Our training trajectories comprise fully equilibrated phases that were initiated from synthetic structures or self-assembled and annealed at various thermodynamic conditions. Equally spaced snapshots of the training trajectories were used to generate the training sets, from which we randomly drew the validation sets. The partitioning ratio of training and validation sets was fixed at 4:1, and each training set comprised a minimum of 20,000 local environments. The testing trajectories comprised self-assembly runs in which at least one phase transition was observed in each run.

We utilized the Hard Particle Monte Carlo (HPMC) and Molecular Dynamics (MD) modules of HOOMD-blue [88] to simulate two-dimensional and three-dimensional convex hard particles. Interaction patchiness was implemented using the Just-In-Time (JIT) compilation module under HPMC. We simulated the equilibration, annealing and self-assembly processes within both the canonical (NVT) and isobaric-isothermal (NPT) ensemble. To simulate the hard particles using MD, we employed the anisotropic Weeks-Chandler-Andersen (AWCA) potential in HOOMD-blue [89].

After we obtained the trajectories, we used the freud analysis package [90] to construct the neighbor list used to calculate per-particle input features. Other analysis

functions of freud were also used, such as the radial distribution function (RDF) and Steinhardt order parameters [35]. The quaternion algebra is handled by rowan [83]. Snapshot images are rendered using Ovito [91]. Because our model is a supervised model, the model's ability to classify the phases is largely influenced by the labeling strategy during training. Here, our labeling strategy labels all particles within the same crystal the same. Additionally, we confirmed via calculation of the diffraction pattern and bond order diagram that our training trajectories contain mostly the reference local environments.

However, we note that one can never be 100% sure in any type of supervised ML model that all configurations that appear in the test trajectory will be in the training trajectories. Indeed, there is always a chance that small local fluctuations in particle arrangements will make our labeling less accurate, and thus will be missed by the model. In the systems used as test cases, we know a priori what equilibrium phases form and train on those phases. We then confirmed via the diffraction pattern and bond order diagram that our training dataset contains mostly the reference environments and that any spontaneous fluctuations that create locally the coordination of a different phase do not persist.

Each test case was simulated as follows:

- **Cubes** For training trajectories, three independent equilibrated HPMC training trajectories were prepared, with one dense fluid trajectory and two crystal trajectories equilibrated at packing fractions of 0.244 and 0.751, respectively. The testing trajectory was generated by compressing the system from packing fraction 0.244 to 0.864 using HPMC simulation and then equilibrating.

- **Patchy triangles** For training trajectories, three independent equilibrated training HPMC trajectories were prepared, with one dense fluid trajectory and two

kagome lattice trajectories. The kagome lattice is a two-dimensional crystal composed of corner-sharing triangles that has been discovered to be assembled by triblock Janus particles [92, 32]. The fluid phase was equilibrated above the nucleation temperature ($k_BT \gtrapprox 0.105$), while the kagome lattice was initialized from a perfect kagome lattice with randomly placed guest particles and equilibrated at $k_BT = 0.105$. The testing trajectory was prepared by quenching from $k_BT = 0.3$ to 0.1 and equilibrating using HPMC.

- **Patchy prisms** For training trajectories, three independent equilibrated HPMC training trajectories were prepared, with one dense fluid trajectory and two crystal trajectories. The testing trajectory was generated by equilibrating an initially disordered system. Here, we use the system from Ref. [1] where the distance between the attractive patches and the face center of the prism is 0.8.

- **Hexagonal bipyramids with aspect ratio** $\alpha = 1.28$ For training trajectories, three independent equilibrated training MD trajectories were prepared, with one dense fluid trajectory, one plastic crystal trajectory, and one body-centered tetragonal (BCT) crystal trajectory [18]. The systems were equilibrated at packing fractions 0.464 and 0.569 for the dense fluid and plastic crystal, respectively.

- **Hexagonal bipyramids with aspect ratio** $\alpha = 3.0$ For training trajectories, three independent equilibrated training MD trajectories were prepared, with one dense fluid trajectory, one liquid crystal trajectory, and one triclinic crystal trajectory [18] equilibrated at packing fractions of 0.4, 0.51, and 0.661, respectively. The testing trajectory was prepared by quenching the system from a reduced pressure P^* of 0.5 to 10 and equilibrating it using MD.

- **Truncated tetrahedrons and octahedrons** For truncated tetrahedrons, three

independent HPMC training trajectories were prepared, with one dense fluid trajectory and two crystal trajectories. The dense fluid trajectories were equilibrated at packing fractions 0.347 for truncated tetrahedrons and 0.524 for truncated octahedrons. For truncated tetrahedrons, the two training diamond crystal trajectories were equilibrated at packing fractions of 0.561. For the truncated octahedrons, the two training high-pressure lithium crystal trajectories were prepared by slowly annealing a self-assembled crystal and equilibrating at a packing fraction of 0.606. The HPMC testing trajectories were prepared by quenching the systems from a reduced pressure P^* from 0.5 to 10 for truncated tetrahedrons (using NPT) and a packing fraction from 0.14 to 0.62 for truncated octahedrons (using NVT), followed by equilibration.

2.3 Results and Discussion

We test the performance of our MLP on seven different systems, beginning with the simplest case of hard cubes that self-assemble into a simple cubic lattice.

2.3.1 Test Case 1: Simple Cubic Crystals Assembled by Hard Cubes

As a demonstration of our method, we first show a simple classification test on a simple cubic structure self-assembled from the fluid phase of hard cubes upon an increase in packing fraction. As shown in Figure 2.1, the cube exhibits 2-fold and 4-fold rotational symmetries along the θ- and ϕ-directions. Using Equation 2.3 and Equation 2.5, we can explicitly consider these symmetries in the data augmentation step and generate an appropriate training dataset that accounts for symmetry. From the three snapshots, each representing a different stage in the self-assembly process, in Figure 2.4 (a), we see that as the packing fraction increases, the number of local environments classified as locally cubic also increases.

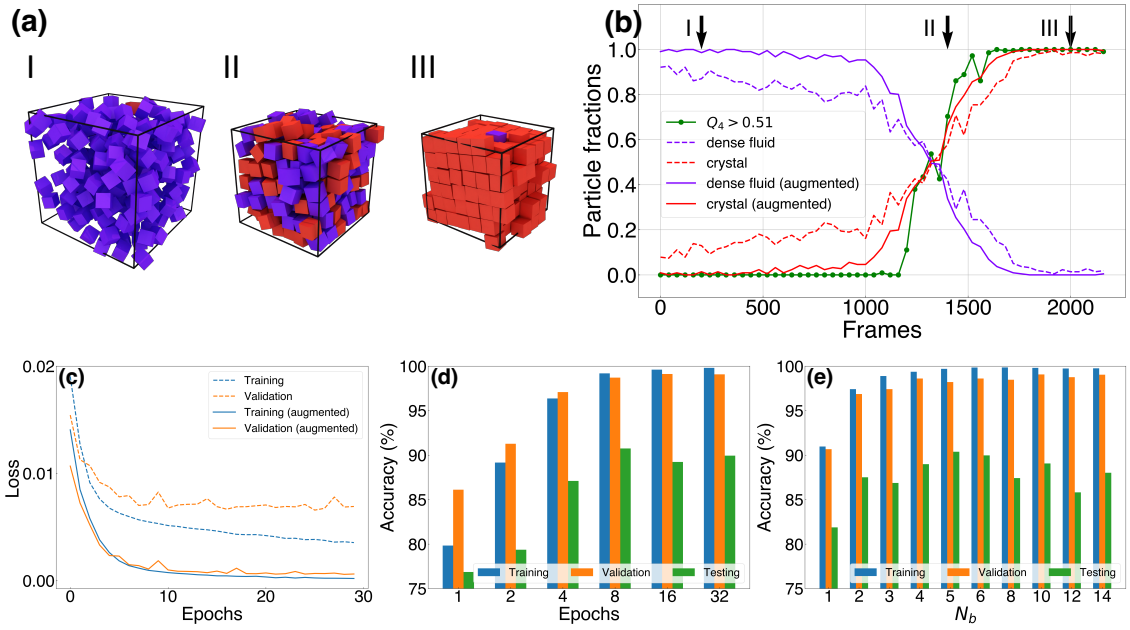

Figure 2.4: **Simple cubic lattice consisted of cubes**. Summary of the MLP classifier's classification results, training details, and accuracy tests, in which we use the simple cubic system as a test case. **(a)** The MLP classifier's classification results on three different snapshots. **(b)** The MLP classifier's classification results on the self-assembly trajectory and classification, compared to the Steinhardt order parameter Q_4 (dotted line) calculated as ground truth. For visualization purposes, solid lines represent the MLP classifier trained on the data accounting for symmetry, while dashed lines are for the MLP classifier trained on the data without accounting for symmetry. Annotations I, II, and III indicate the three corresponding snapshots in (a) for the classifier trained on augmented data. **(c)** Learning curve of the MLP classifier used in (a) with (solid line) and without (dashed line) data augmentation **(d)** MLP classification accuracy plotted versus increasing number of training epochs **(e)** MLP classification accuracy plotted versus increasing number of neighbors N_b in the local particle neighborhood. In (a)-(d), we used fixed $N_b = 6$. In (a)-(c) and (e), we used 30 training epochs.

To further quantify the extent to which data augmentation using symmetry improves the MLP's classification abilities, we compare the classification results by the classifier trained with and without data augmentation. In Figure 2.4 (b), the particle fraction is defined as the number of particles being classified as a certain local environment divided by the total number of particles in the system. There, we also plot the conventional Steinhardt order parameter Q_l [35] with $l = 4$ for comparison. The order parameter Q_l is defined by averaging $Q_{l,i}$ over each particle i in a system

defined as:

$$Q_{l,i} = \sqrt{\frac{4\pi}{2l+1} \sum_{m=-l}^{l} Q_{lm,i} Q^*_{lm,i}} \tag{2.13}$$

$$Q_{lm,i} = \frac{1}{N_b} \sum_{j=1}^{N_b} Y_{lm,\,ij}(\mathbf{r}_{ij}) \tag{2.14}$$

where $Y_{lm,\,ij}$ is the spherical harmonic calculated by the relative position of the target particle i and its neighbors j. For clarification, Q_l differs from the quaternion $\mathbf{Q}$ in equation 2.4.

Note that when using Q_4, we need to select a threshold value (manually selected as 0.51 in this case) based on human intuition or visualization, or we can apply another ML method to determine this threshold, such as support vector machine (SVM), which maximizes the margin. While in the MLP classifier, the threshold is determined solely by the MLP. It can be clearly seen that without data augmentation based on symmetry, misclassification occurs primarily in the fluid phase. In Figure 2.4 (c), we see from the convergence behavior of the loss values that the augmented dataset generally converges faster and better.

To ensure that the augmented data provides sufficient information for the model to recognize the difference between different local environments and predict all possible local fluctuations as the same label, a sufficiently large number of neighbors must be included to build each training dataset. If the number of neighbors is not large enough, a clear drop of training and validation accuracy is seen, as depicted in Figure 2.4 (e).

Figure 2.4 (d) and (e) shows the results of accuracy tests performed on two other aspects. First, in Figure 2.4(d), we observe that accuracy increases with the number of training epochs. We observe that the accuracy converges at around eight epochs and persists without severe overfitting until 32 epochs. Second, in Figure 2.4 (e), we

show the change in accuracy by increasing N_b. The accuracy converges after $N_b = 5$, roughly the number of first nearest neighbors in a cubic lattice. This provides an excellent initial guess for N_b in preparing the training set of our classifier. Furthermore, for the test accuracy, which is defined with respect to the Q_l, 85-90% of the time our model will classify a particle the same as the classification using Q_l. This level of accuracy is sufficient to detect phase transitions also detected by Q_l. Importantly, our model (i) will also detect phase transitions in known systems where Steinhardt OPs (which contain no orientation information) and other commonly used OPs fail, and (ii) process particles' raw positions and orientations with a set of linear operations while Steinhardt OPs use non-linear functions such as spherical harmonics. This linearity in OPs will be important when one would like to apply them to various biased simulations, where gradients of OPs are required. We have also included training and testing time in the Supplementary Information. These tests show that our classifier is robust with appropriate augmentation, training epochs, and N_b. In the next section, we will look at more complicated crystals formed by hard particle shapes with different symmetries.

2.3.2 Test Cases 2 and 3: Self-Assembly of 2D and 3D Patchy Particles

For this second test case, we employ our classifier to classify 2D and 3D systems of patchy particles. In the 2D case depicted in Figure 2.5 (a), each particle is a rigid equilateral triangle (Figure 2.1 (b)). The patchiness is realized by decorating each particle with a Kern-Frenkel attractive patch [93] at each of the three vertices. Additionally, we apply three repulsive patches centered on each of the particle's edges to negatively design against undesirable phases. The guest particles inside the kagome lattice make finding a reliable order parameter that distinguishes the guest particles from non-guest kagome particles necessary. Our simulation results show that during

self-assembly three distinct local environments emerge corresponding to a fluid-like, guest particle, and kagome lattice environments . We demonstrate the classification

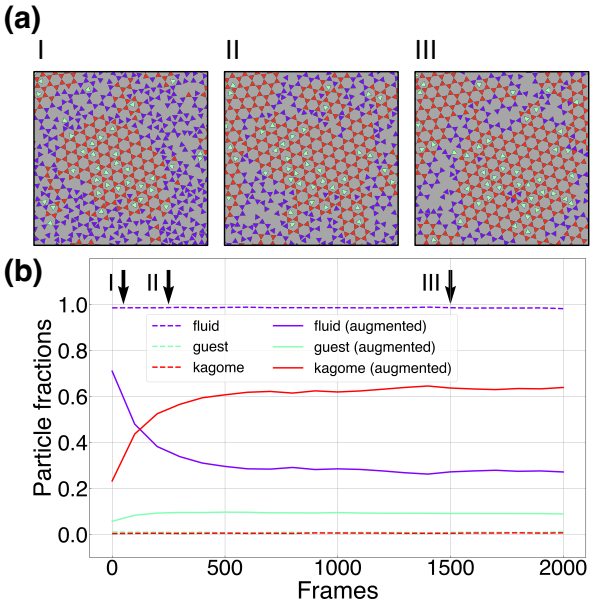

Figure 2.5: **Kagome lattice of patchy triangles** Summary of the MLP classifier's classification results on the test assembly trajectory. **(a)** The MLP classifier's classification results on the three snapshots. **(b)** The MLP classifier's classification results on the entire trajectory. For visualization purposes, solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. The annotations I, II, and III correspond to the three snapshots in (a) for the classifier trained on augmented data.

result of our MLP classifier on the test case assembly pathway of the kagome lattice in Figure 2.5 (b), where the light green color indicates the local environment is classified as a guest particle. It can be seen that the system nucleates and forms a kagome lattice cluster within the fluid phase. Inside the kagome lattice cluster, several guest triangles are enclosed by six surrounding triangles. As the assembly simulation proceeds, we observe the coalescence of multiple small clusters into a single crystallite. Once the majority of particles in the system are in kagome lattice phase, the number of guest particles remains unchanged, which is also captured by the MLP classifier. It should be noted that some guest particles are misclassified as belonging to the fluid phase when they are too close to the surrounding particles

within the kagome lattice. On the other hand, without data augmentation, the classifier only discovered fluid phase. This inability to distinguish local environments without data augmentation is consistent with our observation in hard cubes that when the crystal phase forms, the MLP classifier without data augmentation tends to underestimate the local environments of the ordered phases.

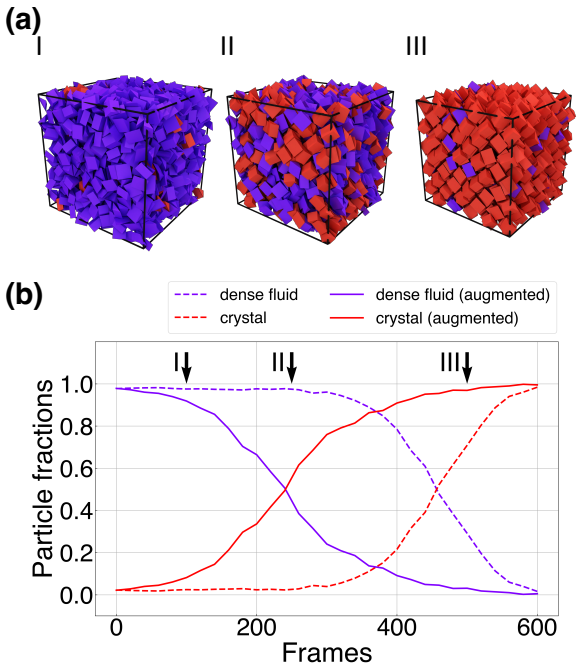

Figure 2.6: **Dimer-diamond phase consisted of anti-aligned patchy triangular prism [1]** The MLP classifier's classification results on the testing self-assembly trajectory, in which we identify the crystallization dimer-diamond structure. Each Wyckoff site comprises two anti-aligned patchy prisms forming a gyrobifastigium. **(a)** The MLP classifier's classification results on the three snapshots. **(b)** The MLP classifier's classification results on the whole trajectory. For visualization purposes, solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. The annotates I, II, and III indicate the corresponding snapshots in (a) for the classifier trained on augmented data.

The second test case is the 3-dimensional dimer diamond self-assembled from the fluid phase of patchy triangular prisms [1] (See Figure 2.1 (c)). In this test example, the triangular prisms pair up and arrange the pairs into the dimer-diamond structure. Since we treat each triangular prism as a particle, it is insufficient for the MLP classifier to classify only the diamond structure. It is crucial to recognize the pairing

motif to classify the crystal phase.

In Figure 2.6 (b), we show the trajectories of particle fractions identified by the classifier trained with or without data augmentation. While the initial and final particle fractions are now the same, the classifier trained with or without data augmentation exhibits different transition behaviors: the non-augmented classifier identifies half of the crystal particles much later than the augmented classifier.

2.3.3 Test Cases 4 and 5: Self-Assembly of Prolate Hexagonal Bipyramids

In the previous section, we showed two different patchy particle systems for which distinguishing the different local environments during assembly is crucial for observing that, in both 2D and 3D, these patchy shapes follow similar assembly pathways. In this section, we focus on two different assembly pathways using geometrically similar building blocks. Both crystals are self-assembled entropically by hard hexagonal bipyramids, but the bipyramid's aspect ratio ($\alpha = 1.28$ vs. 3.0), defined as the ratio of the particle's height h to its circumcircle diameter of base d, i.e., $\alpha = h/d$, greatly influences the intermediate and final products. For $\alpha = 3.0$ orientational order develops prior to translational order, while for $\alpha = 1.28$ we observe the opposite. According to a recent study [18], the self-assembly pathway from a fluid to the final crystal under slow compression involves an intermediate phase – a plastic crystal BCT phase for $\alpha = 1.28$ and a liquid crystal phase for $\alpha = 3.0$. The final products are BCT and triclinic phases for $\alpha = 1.28$ and $\alpha = 3.0$, respectively.

Because only the aspect ratio, and not symmetry, is different for the two shapes, we prepared only one classifier trained on six different synthetically prepared phases of hexagonal bipyramids for the two aspect ratios. We labeled the disordered phase of both hexagonal bipyramid systems as the same dense fluid phase. Because we consider only the particles' relative positions and quaternions as input, there is no

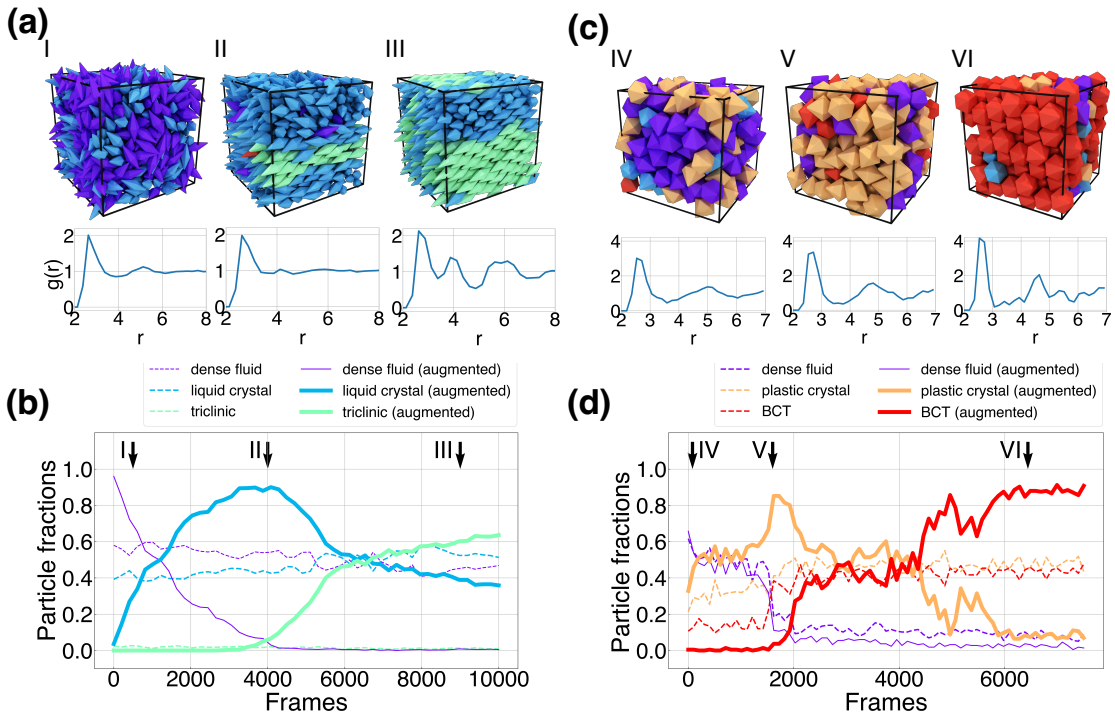

Figure 2.7: **Hexagonal bipyramid systems** The MLP classifier's classification results on the assembly trajectories for two systems of hexagonal bipyramids. Both systems exhibit a two-step transition when crystallizing from an initial disordered fluid phase. **(a)** and **(c)** The MLP classifier's classification on six snapshots of the hexagonal bipyramid systems with (a) $\alpha = 3.0$ and (b) $\alpha = 1.28$, as well as the corresponding radial distribution functions at the bottom. **(b)** and **(d)** The MLP classifier's classification results on the entire trajectories. The annotations I, II, III, and IV, V, VI correspond to the snapshots in (a) and (b), respectively, for the classifier trained on augmented data. For visualization purposes, solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively.

difference in the form of the feature vectors except that they describe different local environments.

We first show the classification results of the MLP classifier on the test case trajectory of hexagonal bipyramids with $\alpha = 3.0$ in Figure 2.7 (a) and (b). The dense fluid, liquid crystalline, and triclinic crystal phases are labeled purple, light blue, and light green, respectively. We also calculated the RDF for the three snapshots of Figure 2.7 (a). In Figure 2.7 (b), our MLP classifier reveals that the system starts to transform into the liquid crystalline phase immediately, followed by slow growth

of the triclinic phase. This can also be seen in snapshots I, II, and III of Figure 2.7 (a). By comparing snapshots I and II, it is evident that the first transition involves only the orientational, and not yet the translational, ordering of particles, leading to a small difference between the two RDFs. Only after the liquid crystalline phase is sufficiently developed does the system order translationally to produce the triclinic crystal; this subsequent behavior is supported by snapshots II and III, as well as their RDFs.

Furthermore, our MLP classifier trained on the augmented data for the hexagonal bipyramids with $\alpha = 1.28$ was used to detect a similar two-step transition, and the outcomes are illustrated in Figure 2.7 (c) and (d). The plastic crystal and BCT phases are indicated by orange and red colors, respectively. In this instance, the translational ordering of the particles occurs before the orientational ordering, which corresponds to a transition from the dense fluid phase to a plastic crystal phase, followed by a transition to the final BCT phase. It is worth noting that both snapshots IV and V of Figure 2.7 (c) possess nearly identical RDFs, meaning the local density and structure are very similar. Despite this, our MLP classifier captures a significant difference in the particle fraction corresponding to dense fluid and plastic crystal. Without data augmentation, the MLP classifier cannot identify particle fractions that match our observed snapshots. In particular, the non-augmented MLP classifier underestimates the particle fractions of the final crystalline phases in both cases. Since the final crystalline phases have both translational and orientational ordering, the MLP classifier performs better with augmented data.

2.3.4 Test Cases 6 and 7: Self-Assembly of Truncated Polyhedra

As a final test of our classifier, we consider systems that are particularly challenging to identify using order parameters. As an example, we consider self-assembled

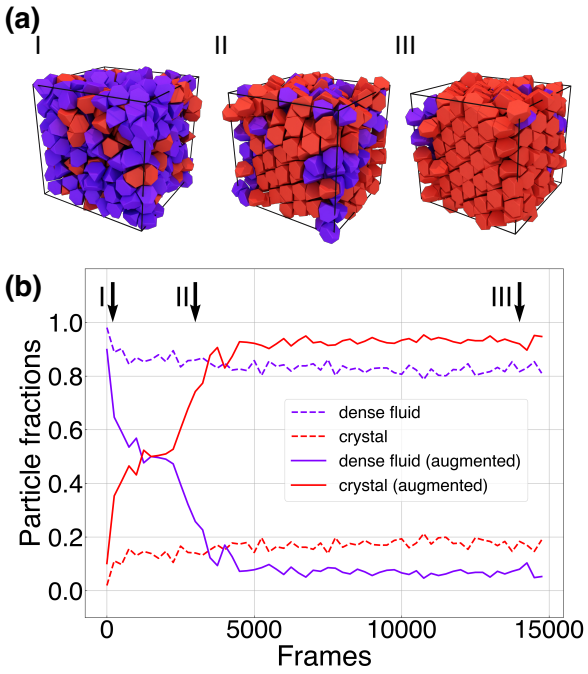

Figure 2.8: **Diamond structure of truncated tetrahedrons** The MLP classifier's classification results on the self-assembly test trajectory, in which we identify the crystallization of the diamond structure from an initially disordered fluid phase. **(a)** The MLP classifier's classification results on snapshots taken at three points along the trajectory. **(b)** The MLP classifier's classification results on the entire trajectory. Solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. The annotations I, II, and III indicate the corresponding snapshots in (a) for the classifier trained on augmented data.

systems of truncated tetrahedrons. Damasceno et al. [15] showed that tetrahedrons with varying degrees of truncation self-assemble into a wide range of complex crystal structures, including diamond structures and high-pressure lithium phases. Here, we investigate two different truncated tetrahedron systems with intermediate and large amounts of truncation. Because the tetrahedron gradually transforms to an octahedron with increasing vertex truncation, we refer to the system with intermediate truncation as the truncated tetrahedron and the system with high truncation as the truncated octahedron, to avoid confusion. Our simulations show results consistent with those of Damasceno et al. When we compressed the systems to a packing fraction between 0.5 and 0.6, they self-assembled into cubic diamond structures and

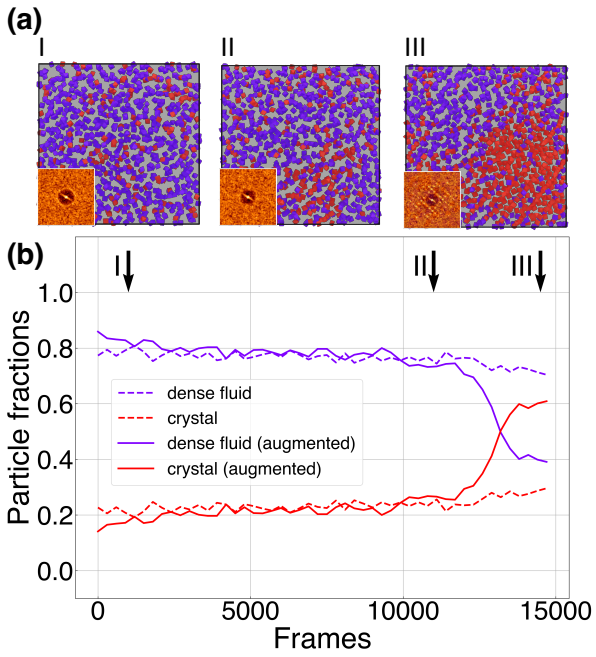

Figure 2.9: **High-pressure lithium phase of truncated octahedrons** The MLP classifier's classification results on the self-assembly test trajectory, in which we identify the crystallization of the high-pressure lithium phase from an initially disordered fluid phase. **(a)** The MLP classifier's classification results on the three snapshots. **(b)** The MLP classifier's classification results on the entire trajectory. Solid and dashed lines represent the MLP classifier trained on the data with and without augmentation, respectively. The annotations I, II, and III indicate the corresponding snapshots in (a) for the classifier trained on augmented data.

high-pressure lithium phases for truncated tetrahedrons and truncated octahedrons, respectively.

In Figure 2.8 (a) and (b), we demonstrate the classification results of the MLP classifier for the truncated tetrahedrons. In Figure 2.8 (b), during the early simulation stage, the classifier trained on augmented data identifies an abrupt increase in the number of particles classified as belonging to a cubic diamond local environment. Subsequently, the diamond structure grows rapidly and remains stable. However, the classifier trained on the data without augmentation reports a slow growth of the diamond structure and fails to identify the phase transition.

In Figure 2.9 (b), our MLP classifier trained on augmented data first detected a

progressively increasing high-pressure lithium-like local environment before the simulation reached frame12000. After that, our classifier identified that the high-pressure lithium phase reached the critical nucleus size, followed by rapid crystal growth. Thus the MLP classifier discovered a homogeneous nucleation process followed by crystal growth. This nucleation process can also be seen in Figure 2.9 (a). From the classification-based coloring in snapshot I, we observed many small sub-critical crystal nuclei form and dissolve in the early stages. In snapshot II, shortly before the rapid growth of the crystal, the MLP classifier identified a noticeable amount of crystal-like local environment in the lower part of the box, which we expect to be the critical nucleus. At the end of the simulation, we can see that the crystal has stabilized in the lower right corner of the box in snapshot III, where it is in coexistence with a dense fluid phase.

For comparison, we demonstrate the classifier's classification ability without data augmentation. As expected, the MLP classifier trained on the data without augmentation failed to recognize the rapid growth of the crystal. The inability of the classifier to identify the formation of complex crystals is similar to the case of truncated tetrahedrons, as shown in Figure **??** (a) and (b), and thus for both test cases we see that data augmentation highly improves the performance of the MLP classifier. We can rationalize this performance difference between training on augmented vs. non-augmented data by examining the information contributed by each component of the particle feature vector. Since we started both simulations from very dense fluid phases, there are no significant changes in densities that are encoded in $r_{ij}^{(au)}$ during the formation of crystals. Therefore, the information provided by $r_{ij}^{(au)}$ is insufficient to recognize the formation of crystals. As a result, whether the MLP classifier can correctly learn to identify the formation of crystals largely depends on whether θ_{ij},

ϕ_{ij} and $\mathbf{q}_{ij}$ are augmented.

2.4 Conclusion

To summarize, we have developed a simple, yet powerful, physics-agnostic local environment classifier specifically designed for systems of particle shapes utilizing a multilayer perceptron. As demonstrated in this paper, our method is applicable to a range of enthalpically and entropically patchy particle systems. Importantly, our MLP classifier does not need conventional roto-translation invariant symmetry functions to transform per-particle quantities to input descriptors. Instead, it directly takes particle positions and orientations as input features, complemented by shape-encoded data augmentation. To demonstrate robustness and flexibility, our classifier's performance was assessed on a variety of self-assembling systems, including hard cubes, 2D and 3D patchy shapes, hexagonal bipyramids with two different aspect ratios, and two different truncated shapes. The data augmentation method we used is straightforward and easily transferable to other systems involving particle orientations, such as molecular or coarse-grained systems. As a result, our approach should be useful in classifying different polymorphs formed by molecules by properly defining orientation through quaternions. We emphasize that we do not intend for our model to be used on an unknown dataset. Likewise, Steinhardt and other order parameters (OP) commonly used in soft matter studies are not used on unknown datasets. Rather, they are used to automate the detection of phase transitions in systems where the initial and final phases are known. In our paper, we are attempting to show that our supervised ML model is able to detect phase transitions in known systems nearly as well as those order parameters.

CHAPTER III

A Generalized Machine-Learning Framework for Developing Alchemical Many-Body Interaction Models for Polymer Grafted Nanoparticles

This chapter is reproduced from *A Generalized Machine-Learning Framework for Developing Alchemical Many-Body Interaction Models for Polymer Grafted Nanoparticles* [34], a paper published in *The Journal of Chemical Theory and Computation* by Melody Y. Zhang *, Shih-Kuang (Alex) Lee *, Sharon C. Glotzer, and Rebecca K. Lindsey †. (* denotes the first authors and † indicates the corresponding author) The Glotzer Group's contributions to this work mainly include applying the forward-reversed method to perform potential of mean force calculations, providing theoretical and technical support for the Digital Alchemy method, and designing simulations using the HOOMD-blue simulation engine, along with performing their analysis. The Lindsey group performed polymer-grafted nanoparticle design, ChIMES potential model fitting, and provided technical support, X-ChIMES model design and fitting, and simulations using both ChIMES and X-ChIMES models, along with their analysis.

3.1 Introduction

Polymer-grafted nanoparticles (PGNs) provide an endlessly customizable base unit that can be used to build a manifold of complex self-assembled structures with technological applications spanning catalysis [94], drug delivery [95], optics [96], battery technologies [97], and more. These particles are broadly comprised of an inner core nanoparticle (e.g., SiO_2, Ag, Au, or CdSe) with chains such as alkyl thiol ligands grafted onto the surface, allowing precise modulation of interparticle interactions through mechanisms such as hybridized quantum states [98], covalent interactions [99], and solvent-particle interactions [100]. PGNs can self-assemble into a variety of structures, each exhibiting unique and tunable properties pertaining to thermoelectric behavior [101], structural order and symmetry [102], electric permittivity [103], and selective permeability [104]. However, this tunability gives rise to the practical challenge of identifying specific PGN configurations (e.g., polymer chain length, grafting density, chemical functionality, and solvent composition) that are suitable for producing a target self-assembled structure (SAS).

Significant effort has been directed toward addressing this challenge, with early work taking a "forward design" approach, utilizing theoretical frameworks and simulations to predict SAS given an input PGN description. For studies of this nature, accuracy of the underlying models describing inter-PGN interactions is critical. For instance, initial studies using the hard-sphere model revealed a strong correlation between binary particle size and resulting packing fractions [105]. However, phase diagrams produced by these models are not entirely consistent with experiments [106, 107] due to model's inability to account for properties of soft and compressible polymer ligand layers. More recent studies that employ adjustable

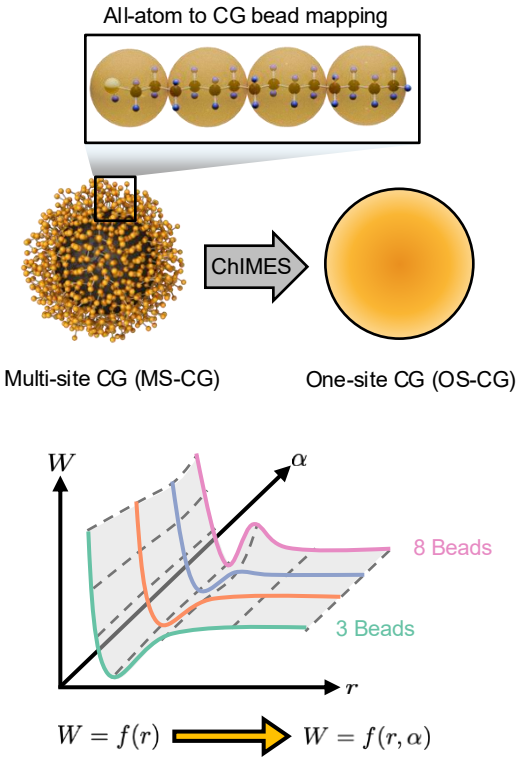

Figure 3.1: **Polymer-grafted nanoparticle (PGN) coarse-graining scheme and alchemical PGN interaction surface, W. Top**: This work treats PGN interactions at coarse-grained (CG) resolution. Two CG resolutions are considered: multi-site (MS), where approximately 4 heavy atoms are mapped to one interaction bead, and one-site (OS) resolution, where an entire PGN is treated as a single interaction site. **Bottom**: This work details the extension of OS-CG models to enable the description of W as a function of both inter-PGN separation distance r and polymer length α.

short-range pairwise inverse-power law potentials have produced more realistic predictions [108, 109, 110], and theoretical approaches that implicitly or explicitly account for polymer distribution around the PGN core have enabled predictions with yet better experimental agreement [111, 112, 113, 114].

While effective, the forward design approach can be inefficient when an SAS is known *a priori*. In such cases, users must analyze an assortment of PGNs until they identify one that can form the desired SAS. In these cases, an "inverse design" approach becomes attractive, wherein the design space is autonomously and efficiently searched to identify an optimal PGN building block for producing a user-specified

SAS. Inverse design can be achieved through a variety of approaches. For instance, a machine-learning (ML)-based landscape engineering method that employs diffusion maps and hybrid Monte Carlo (MC) simulation has been used to optimize factors such as patch placement and strength on colloidal nanoparticles to ensure the formation of target structures [115]. Similarly, genetic algorithm-based reinforcement learning schemes have shown promise in reverse-engineering colloidal crystals and quasicrystals [116]. Alternatively, the relative entropy method defines a target probability distribution of the desired microstructure and samples interaction parameters that maximize the likelihood of embodying the target system [117, 118].

Of the multitude of available inverse design schemes, the statistical mechanics-based digital alchemy (DA) method is a particularly attractive approach [30, 31]. DA can be used to optimize PGN design for a SAS by treating particle attributes such as shape or effective interactions as thermodynamic variables that can be adjusted to minimize the free energy of the target SAS, maximizing likelihood of its formation. [31, 32]. Compared to other inverse design methods, DA is an intuitive, statistically and mechanically rigorous, straightforward to implement, and efficient method. DA has been employed in a variety of self-assembly studies, including the inverse design of symmetric hard particles for assembling target colloidal crystals [31], shape-directed solid–solid phase transitions in colloidal systems [119], robust optimization of pair potentials through iterative information maximization [120], optimization of a generalized Fourier potential [33], alchemical molecular dynamics to optimize double well potential [41], and energy landscape analysis for the design of interaction potentials [121].

Nevertheless, DA has not yet been applied to systems that exhibit many-body interactions. For instance, PGNs with longer polymer chains can introduce significant

steric effects at short range, resulting in three-body or higher-order interactions [112, 122, 123, 124, 125, 126]. Moreover, in its current formulation, DA provides high-level information on optimal PGNs, such as effective interparticle interaction curves or PGN shape, but not detailed insights into specific physical properties of optimal PGNs, such as polymer length or grafting density, which govern effective interactions. Practically, this means experimentally realizing these recommendations requires trial-and-error, i.e., to synthesize PGNs that exhibit the DA-recommended properties. To bridge this gap, DA requires a high-dimensional "alchemical" interaction surface that depends on both system configuration and design parameters directly correlating with the physical properties. Conventional molecular mechanics (MM) models lack the ability to model such an interaction surface that incorporates design parameter descriptions. In contrast, ML-based methods are well-suited for learning complex, high-dimensional interaction surfaces [127, 128, 129], offering a promising route to enhancing DA.

To this end, we extended the ChIMES ML interatomic model to establish X-ChIMES, a generalized ML framework for producing coarse-grained (CG) PGN design surfaces describing how effective PGN interactions change as a function of physical properties, which can be integrated into inverse design tools like DA to provide chemically detailed information on optimal PGNs.

The remainder of this paper is organized as follows: We begin with a high-level overview of our X-ChIMES, followed by description of our training data generation strategy and an overview of ChIMES in Section 3.2. We then present results for generating ChIMES inter-particle interaction models in Section 3.3.1 and 3.3.2. We follow with description of our X-ChIMES extension, its application to MD simulations of PGN systems in Section 3.3.3. We then demonstrate use of the resulting

X-ChIMES model in DA calculations in Section 3.3.3. We conclude by discussing implications of the X-ChIMES method and future directions. Key acronyms used throughout the paper are given in Table 3.1.

Table 3.1: Acronym Guide

Acronym	Description
PGN	polymer-grafted nanoparticle
CG	coarse-grain
SAS	self-assembled structure
DA	digital alchemy
ML-IAM	machine-learned interaction model
MS-CG	multi-site coarse-grained
OS-CG	one-site coarse-grained
PMF	potential of mean force
X-ChIMES	extended ChIMES
X-OS-CG	extended OS-CG
SMD	steered molecular dynamics
FR	forward-reverse
Alch-MC	alchemical Monte Carlo

3.2 Methods

A central goal of this work is to develop PGN design surfaces that describe how inter-PGN interactions vary with both inter-PGN distances and PGN attributes. We aim to describe inter-PGN interactions at the resolution of one-PGN to one interaction site (i.e., OS-CG resolution) due to the significant increase in computational efficiency afforded compared to finer-grained descriptions. For fixed PGN attributes, interparticle interaction surfaces can exhibit a variety of shapes. Hence, rather than describing these interactions through conventional molecular mechanics-inspired models, we focus on the development of ML-based solutions. To account for the loss of all conformational degrees of freedom at the OS-CG resolution, our ML interparticle models describe PMF rather than potential energy between particles [130, 131]. Hence, in the following sections, we provide (1) description of the PGNs studied and our approach to inter-PGN PMF generation, (2) description of the ML models developed in this work, and (3) general simulation protocols.

3.2.1 PMF Generation

As shown in Figure 3.1, we treat polymer-grafted nanoparticles with varied polymer lengths as a testbed system. For efficiency, the ground-truth multi-site coarse-grained (i.e., MS-CG) description is based on the Martini 3 model [132]. Each PGN comprises a 4.7 nm diameter core coated with 300 polymers, resulting in a grafting density of 4.32 chains/nm^2. PGN core-core and core-bead interactions are described through a shifted Lennard-Jones (LJ) potential while PGN ligand bead-bead interactions are modeled as Martini 3 C1 beads, which interact via LJ potential. Bead bond and angle bending interactions are described with a harmonic potential. Additional details of the ground-truth model, including parameters, are provided in Supplementary Information Sections 3.5.1, 3.5.2, and Table 3.4.

PGN PMF calculations are non-trivial due to the many conformational degrees of freedom that must be accounted for. One approach is to sample particle forces in their equilibrium configurations. After collecting sufficient force data, an ML model can be used to learn PMFs indirectly. This approach is used by Giunta et al. [125] to enable simulation of ligand-stabilized nanoparticles. Such a data-driven method is reliable but slow to converge. An alternative strategy is to use enhanced sampling techniques. For example, umbrella sampling (US) [133] coupled with weighted histogram analysis method (WHAM) [134] is a well-established and robust equilibrium-enhanced sampling technique, where PMFs are calculated by analyzing the biased probability distribution sampled around the pre-selected windows distributed along the reaction coordinate. Nourian *et al.* used US to calculate PMFs for PGNs with various polymer architectures (e.g., linear, dendron, and comb) and found that precisely-tuned comb polymers introduce enhanced interaction range and repulsive strength between PGNs [135]. Zhou et al. used the blue moon sampling method to sample and inte-

grate the mean PGN force at various cluster configurations to calculate the accurate pairwise and three-body PMFs for studying the self-assembly of aggregate and dispersed phases [126]. While effective, the primary drawback of these equilibrium enhanced sampling methods can be computationally expensive and data-intensive.

Non-equilibrium methods can also be used for PMF calculation, e.g., Steered Molecular Dynamics (SMD) based on the Jarzynski Equality (JE) formulation. SMD is an enhanced sampling method that applies a moving harmonic potential to steer systems along a chosen reaction coordinate in non-equilibrium molecular simulations. In SMD, the system is pulled across the selected reaction coordinate to calculate the PMFs using the average of the exponentiated work. Park *et al.* [136, 137] used the SMD method to calculate the exponentiated work and generate PMFs of deca-alanine oligopeptide molecules.

However, accurate calculation of PMFs often requires a large number of non-equilibrium SMD simulations to obtain a reliable estimation of the mean exponentiated work done along the reaction coordinate. Consequently, such a technique is rarely applied in the PMF calculation for PGN systems containing large particle degrees-of-freedom due to the significant computational burden. As a result, such an approach is limited to simpler systems where sampling efficiency is less challenging [138, 139, 140].

In the present work, we use the JE formulation to generate PMFs between PGNs via SMD. To overcome the challenge of high computational cost, we employ the forward-reverse (FR) method proposed by Kosztin et al. [38] to approximate PMFs. This method employs SMD to generate two time-reversed symmetric paths (i.e., forward and reverse paths) such that the PMFs are correlated to the difference between mean forward work and mean negative reversed work, thereby significantly reducing

the number of required SMD runs without directly invoking JE.

It has been shown [112, 122, 123, 124, 125, 126] that PGNs with large capping layer thicknesses can exhibit non-negligible many-body effects. Hence, we generate PMF data for both two- and three-body PGN interactions by implementing the FR method into the many-body PMF grid sampling scheme outlined in the work by Zhou *et al.* [126], i.e., leveraging the symmetry of three-PGN clusters to design minimal grids for training data sampling. Additional details and theoretical derivations of these methods are provided in Supplementary Information Section 3.5.3.

3.2.2 The ChIMES Machine-Learned Interaction Model

In the present work, the physics-informed ChIMES [141] machine-learned interaction model (ML-IAM) is used to describe OS-CG resolution inter-PGN interactions and is extended to produce corresponding X-ChIMES PGN design surfaces. Of the multitude of available ML-IAMs, we select ChIMES due to its simplicity, extensibility, inherent capability for describing many-body interactions, and capability to generate high accuracy [142, 143, 144, 127] models using considerably less training data than more complex ML-IAM [127, 145].

The ChIMES model, originally developed for treating systems at atomistic resolution, describes system energy through an explicit many-body cluster expansion:

$$E_{\text{total}} = \sum_{i_1}^{n_a} {}^{1}E_{i_1} + \sum_{i_1>i_2}^{n_a} {}^{2}E_{i_1 i_2} + \ldots + \sum_{i_1>i_2\ldots i_{n\text{B}-1}>i_{n\text{B}}}^{n_a} {}^{n\text{B}}E_{i_1 i_2 .. i_{n_B}} \tag{3.1}$$

where E_{total} represents the total energy, consisting of single-particle energy ${}^{1}E_{i_1}$, pairwise energy ${}^{2}E_{i_1 i_2}$, and up to maximum-bodied contributions ${}^{n\text{B}}E_{i_1 i_2 .. i_{n_\text{B}}}$. nB

represents n-body interactions, $i_1, i_2, ..., i_n$ represents particle indices, and n_a is the total number of particles in the system.

The single-particle energy is included when ChIMES models are trained on first-principles-derived data, which explicitly account for single-particle contributions to overall system energy and therefore exhibit a compositional dependence even in the case of a fully non-interacting system. In this study, this term is neglected.

The ChIMES two-body interactions between a pair of particles i and j are given by the following:

$$^2E_{ij} = f_{\mathrm{p}} + f_{\mathrm{s}} \sum_{A=1}^{\mathcal{O}_{2\mathrm{B}}} c_A T_A(s_{ij}) \tag{3.2}$$

where T_{A} is a Chebyshev polynomial of subscripted order, $O_{2\mathrm{B}}$ is the highest polynomial order, and c_{A} is a set of model fitting coefficients. f_{s} is a "Tersoff" style [146] smooth cutoff function, which ensures the two-body interactions smoothly decay to zero at the outer cutoff, $r_{\mathrm{c,out}}$. As discussed elsewhere [141, 147], Chebyshev polynomials exhibit several properties which make them convenient basis functions for modeling interparticle interactions. This function incorporates a hyperparameter f_o that governs the function's activation point [127]. f_{p} is a penalty function designed to prevent the system from sampling distances below the model inner cutoff [127], $r_{\mathrm{c,in}}$. s_{ij} is the transformed pair distance, which maps interparticle distances r to the $[-1, 1]$ domain over Chebyshev polynomials are defined. The transformation from r to s as achieved through the following equation:

$$s = (x - x_{\mathrm{avg}})/x_{\mathrm{diff}} \tag{3.3}$$

where

$$x_{\mathrm{avg}} = 0.5(x_{\mathrm{c,out}} + x_{\mathrm{c,in}}) \tag{3.4}$$

$$x_{\text{diff}} = 0.5|x_{\text{c,out}} - x_{\text{c,in}}| \tag{3.5}$$

A Morse-style transformation [148, 149, 150] is commonly used, where x is taken as $\exp(-r/\lambda_{\text{Morse}})$ [141]. Morse-style transformation has the effect of exaggerating changes at short interparticle distances and reducing effective changes in large separation distances. λ_{Morse} is typically set to the pair distance corresponding to the first peak in the radial pair distribution function (RDF).

The ChIMES three-body energy is defined as the following:

$$\begin{aligned} {}^3E_{ijk} =& f_{\text{s}}(r_{ij}) f_{\text{s}}(r_{ik}) f_{\text{s}}(r_{jk}) \\ & \sum_{A=0}^{\mathcal{O}_{\text{3B}}} \sum_{B=0}^{*} \sum_{\Gamma=0} c_{AB\Gamma} T_A(s_{ij}) T_B(s_{ik}) T_\Gamma(s_{jk}) \end{aligned} \tag{3.6}$$

For many-body interactions to be valid, all pair distances must fall within the model cutoffs, forming a fully connected graph between constituent particles [151]. As indicated by the asterisk (*) below the triple summation symbol, a sufficient number of non-zero polynomial orders must be present within the triple sum to qualify as a true many-body interaction. A three-body interaction is considered valid if all three distances are within the model cutoffs, and the polynomial orders of at least two of the constituent pairs must begin at one, since the first Chebyshev polynomial is a constant and does not vary with the interparticle distance.

In this work, ChIMES models are generated by minimizing the loss function between model prediction and ground truth:

$$\mathcal{L} = \frac{1}{N_{\text{train}}} \| W^{\text{OS-CG}} - W^{\text{MS-CG}} \|_2^2 \tag{3.7}$$

where N_{train} is the total number of training data points, $W^{\text{OS-CG}}$ is the OS-CG ChIMES model, and $W^{\text{MS-CG}}$ is the ground truth MS-CG PMF training data.

Since ChIMES is entirely linear in its fitted parameters, the model optimization problem can be recast as the following overdetermined matrix equation:

$$\boldsymbol{A}\boldsymbol{c} = \boldsymbol{b}_{\text{MS-CG}} \tag{3.8}$$

where $\boldsymbol{b}_{\text{MS-CG}}$ is the vector that consists $W^{\text{MS-CG}}$ for each training configuration, $\boldsymbol{c}$ is the vector consisting model parameters to be determined, and the elements of design matrix $\boldsymbol{A}$ are given by the following:

$$A_{ab} = \frac{\partial W_a^{\text{OS-CG}}}{\partial c_b} \tag{3.9}$$

where a represents index over PMF components, i.e., sampled dimers or trimers, and b is the index over permutationally invariant model parameters. In this study, model parameters are determined via truncated singular value decomposition (TSVD), where the singular values are set to zero if the corresponding ratios between those and the largest singular value are below $\kappa = 10^{-5}$. Additional information on TSVD is provided in Supplementary Information Section 3.5.4.

In this work, we extend ChIMES to incorporate an additional dimension, α, which represents a tunable PGN attribute. Henceforth, we refer to this "alchemical" model as X-ChIMES. In this study, α corresponds to polymer length in terms of the number of MS-CG beads. The X-ChIMES model embeds α dimension by introducing an additional product of polynomials to the energy expression as shown in Equation 3.10. We note that this strategy can be extended to an arbitrary number of alchemical dimensions through the introduction of additional corresponding polynomials to the product. Consequently, the potential energy surface describing the system is lifted to a higher dimensionality, incorporating information on both inter-PGN distance

and polymer length. The X-ChIMES two- and three-body energies are described as follows:

$$^2E_{\alpha ij} = f_{\mathrm{p}} + f_{\mathrm{s}} \sum_{A=0}^{\mathcal{O}_{\alpha,2\mathrm{B}}} \sum_{B=1}^{\mathcal{O}_{r,2\mathrm{B}}} c_{AB} T_A(s_\alpha) T_B(s_{r,ij}) \tag{3.10}$$

and

$$\begin{aligned} ^3E_{\alpha ijk} = f_{\mathrm{s}}(r_{ij}) f_{\mathrm{s}}(r_{ik}) f_{\mathrm{s}}(r_{jk}) \sum_{A=0}^{\mathcal{O}_{\alpha,3\mathrm{B}}} \\ \sum_{B=0}^{\mathcal{O}_{r,3\mathrm{B}}} \sum_{\substack{\Gamma=0 \\ *}} \sum_{\Delta=0} c_{AB\Gamma\Delta} T_A(s_\alpha) T_B(s_{r,ij}) T_\Gamma(s_{r,ik}) T_\Delta(s_{r,jk}) \end{aligned} \tag{3.11}$$

where $s_{r,ij}$, $\mathcal{O}_{r,2\mathrm{B}}$, and $\mathcal{O}_{r,3\mathrm{B}}$ are equivalent to s_{ij}. We note that the use of a product of Chebyshev polynomials for the alchemical and distance dimensions allows treating coupling between the two dimensions, which can be necessary when changes in changes in α (ligand properties) manifest changes in the potential energy surface that go beyond simple (e.g., linear) scaling.

$\mathcal{O}_{2\mathrm{B}}$, and $\mathcal{O}_{3\mathrm{B}}$ in Equations 3.2 and 3.6, and are used to differentiate from s_α, $\mathcal{O}_{\alpha,2\mathrm{B}}$, and $\mathcal{O}_{\alpha,3\mathrm{B}}$, which are introduced to account for the new alchemical dimension. In contrast to the distance dimension which applies a Morse-style transformation to map inter-PGN distances to the $[-1, 1]$ domain, we employ a direct transformation for the α dimensions (i.e., $x = \alpha$ in Equations 3.3 - 3.5), since we do not know *a priori* how distance polynomials should be modulated in response to changing α.

3.2.3 Simulation Protocols

Unless otherwise stated, all simulations were performed in HOOMD-blue [88] . The MD simulations were conducted at $T = 300$ K with a timestep of 2.5 and 5.0 fs for MS-CG and OS-CG systems, respectively. For thermostatting, the Nosé-Hoover thermostat [152] was only applied for SMD simulations, since a deterministic integration is required to calculate PMFs via the FR method. For all other

MD simulations, the Bussi-Donadio-Parrinello thermostat [153] was used, which is a stochastic thermostat and guarantees the correct sampling of the canonical ensemble. A Langevin piston was employed to control pressure in isobaric-isothermal (NPT) simulations [154]. Custom force modules in HOOMD-blue were implemented to perform PMF calculations, including SMD and US simulations. Furthermore, we performed alchemical Monte Carlo (Alch-MC) simulations to inversely design the target SAS. The details about Alch-MC simulations are outlined in Section 3.3.2.

ChIMES models were generated using the ChIMES LSQ package [155]. To enable DA simulations with the X-ChIMES model, a custom HOOMD-blue module was developed using the alchemical Monte Carlo (Alch-MC) algorithm [30, 31]. Subsequently, the ChIMES Calculator package [156] was used to evaluate the ChIMES interaction potential to conduct simulations.

Trajectory analyses were carried out using freud [157], and snapshots were rendered using OVITO [91]. signac [158, 159] and row [160] were implemented for simulation workflow and data management.

We provide detailed instructions for the packages we developed in Supplementary Information Section 3.5.8, and 3.5.9.

3.3 Results and Discussion

3.3.1 PMF Generation

FR Method and Benchmarks

We sampled PMF data for generating ChIMES and X-ChIMES models via SMD simulations with the FR method. Two-body PMFs, $W_{\mathrm{2B}}^{\mathrm{MS\text{-}CG}}$, were generated through the following FR SMD protocols: First, two PGNs of a given α were prepared at a separation distance large enough such that the PGNs were non-interacting. Then, a strong harmonic guiding potential was used to pull the PGNs toward an inner

cutoff distance with a constant pulling velocity. Once the innermost inter-PGN distance was reached, the PGNs were equilibrated at that distance for a sufficient time. Subsequently, the PGNs were pulled by the same harmonic guiding potential but in the reverse path (i.e., reversed pulling velocity) to their initial separation distance.

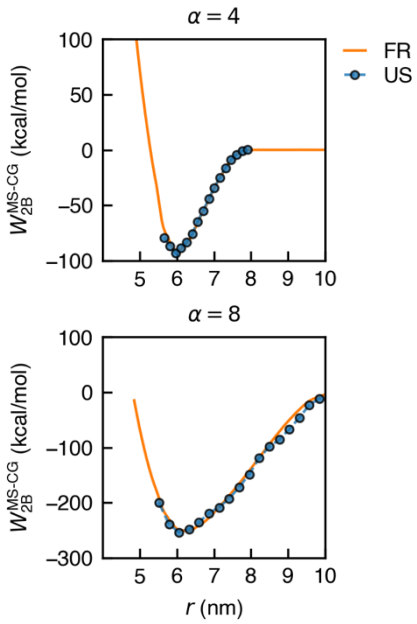

Figure 3.2: **Comparison of PMFs generated via umbrella sampling (US) and forward-reverse (FR) steered molecular dynamics (SMD) for PGNs with $\alpha = 4$ and 8**. SMD pulling velocities are chosen such that PMFs calculated based on the FR method are consistent with those obtained by the US method.

The FR method relies on the stiff spring approximation [161], which requires sufficiently slow pulling velocities to ensure (1) the positions of the core particles remain in the vicinity of the reaction coordinate, which is the equilibrium distance of the guiding potential, and (2) the inter-PGN forces among the cores and polymer beads remain relatively small compared to the force exerted by the guiding potential, allowing them to be treated as perturbations to the system's potential energy. A challenge faced by directly applying JE is that simulations become considerably more

expensive as the pulling velocity is decreased. Other free energy methods such as US circumvent this challenge but require considerably more computational effort due to the need for a large number of individual sampling simulations. Hence, a series of FR SMD simulations were conducted for pulling velocities varying from 1×10^{-5} to 2.5×10^{-7} nm/fs (Figure S1) to identify optimal values for our system. For each velocity, we ran five simulations to calculate the average work as required by the FR method. Our selections were validated by comparing the resulting PMFs against those obtained via US with WHAM. A total of 223 windows at spacings of 0.025 nm were used to generate a single PMF. Such a large number of windows is required to ensure adequate histogram overlap and successful PMF generation. For each window, we ran 4 million steps of MD simulation to obtain the biased equilibrium distribution. In Figure 3.2, we show the PMFs of PGNs with 4 and 8 Martini 3 C1 beads per polymer chain (i.e., $\alpha = 4$ and 8). We observe strong agreement between two-body PMFs obtained from US and FR SMD simulations with our selected pulling velocities of 1×10^{-6} nm/fs and 2.5×10^{-7} nm/fs for $\alpha = 4$ and 8, respectively. We can then use the validated pulling velocities to calculate two-body PMFs for PGNs with $\alpha = 3$ to 8, each obtained from five SMD simulations. Notably, for any given polymer length, the five combined SMD simulations achieve the same level of convergence while requiring an order of magnitude less simulation time than a single set of 223 US simulations.

PMF Training Data Sampling

PGNs with different α values have different effective interaction ranges (e.g., particles with larger values of α interact over longer ranges than those with smaller α). Hence, different initial separation distances were used depending on the α values of PGNs. We use 10 nm for $\alpha = 3$ to 6, 11 nm for $\alpha = 7$, and 12 nm for $\alpha = 8$. The

resulting two-body PMFs are shown in Figure 3.3. As expected, we find an increase in attractive interactions and equilibrium inter-PGN distances in the PMFs as the polymer length increases.

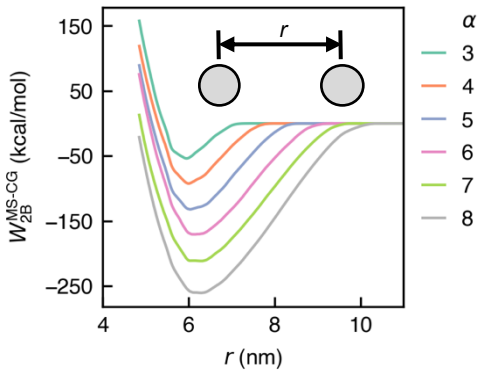

Figure 3.3: **Two-body PMFs $W_{2B}^{MS\text{-}CG}$ for PGNs with $\alpha = 3$ to 8.**

In the present work, we consider many-body interactions up to three-body interactions, in addition to two-body interactions. The procedure for generating many-body PMFs is illustrated in the top panel of Figure 3.4. We generate the initial configuration by placing three PGNs at separation distances of $r_{0,12}$, $r_{0,13}$, and $r_{0,23}$, where $r_{0,13}$ and $r_{0,23}$ are large enough to be treated as the zero energy point for three-body PMFs. Following the strategy outlined by Zhou et al. [126], non-degenerate sets of initial configurations for the three PGNs are considered such that PGN 1 and 2 are placed symmetrically along the x-axis with respect to the origin. For each fixed $r_{0,12}$ value, the position of PGN 3 is shifted by a fixed distance d_x along the positive x-axis. This approach assumes that the resulting three-body PMFs inherit cylindrical symmetry, indicating that the energy remains constant when PGN 3 is positioned symmetrically about the y-axis.

For PGNs with $\alpha = 3$ to 6, we selected a total of 319 different initial configurations with unique combinations of 11 initial r_{12} values and 29 d_x values ranging between

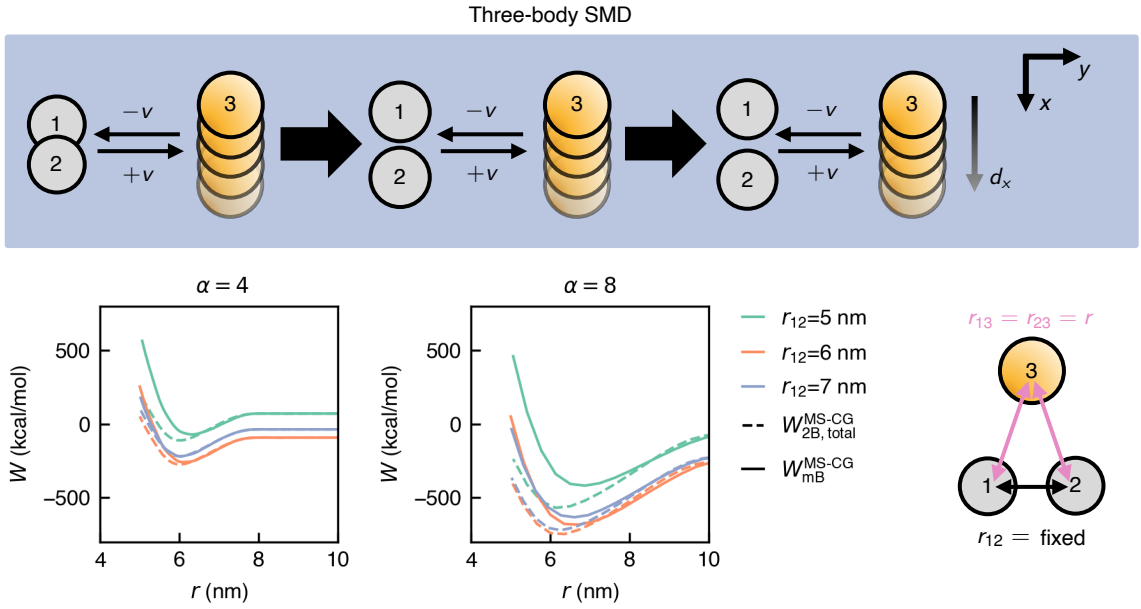

Figure 3.4: **Overview of many-body PMFs. Top**: Schematic representation of three-body SMD simulation via FR method. The three-body SMD simulations are performed by positioning PGN 1 and 2 at various separation distances r_{12} while systematically pulling PGN 3 from different initial positions. **Bottom**: The left and middle figures compare the two-body PMFs $W_{2B}^{MS\text{-}CG}$ (dashed lines) and many-body PMFs $W_{mB}^{MS\text{-}CG}$ (solid lines) for PGNs with α=4 and 8. Each PMF is a function of an isosceles trimer configuration as illustrated in the right figure, where r is the two equal sides (magenta arrows) and r_{12} is the fixed base (black arrow). The deviation between $W_{2B}^{MS\text{-}CG}$ and $W_{mB}^{MS\text{-}CG}$ indicates the presence of non-negligible three-body interactions.

0 and 9 nm. For PGNs with $\alpha = 7$ and 8 that exhibit a longer interaction range, we selected a total of 672 initial configurations with unique combinations of 21 r_{12} values and 32 d_x values ranging between 0 and 14 nm. These values provide sufficient sampling points to produce smooth PMFs. The three-body SMD simulations are similar to the two-body SMD simulations: PGNs 1 and 2 are fixed at their initial positions using strong harmonic potentials, and PGN 3 is pulled along the reaction coordinate, which is the path perpendicular to the x-axis from its initial position, to integrate work done on PGN 3 for PMF calculations.

As noted by Zhou et al. [126], the resulting PMF from these SMD simulations, $\tilde{W}_{3B}^{MS\text{-}CG}$, includes both the three-body PMF (i.e., $W_{3B}^{MS\text{-}CG}$) and two of the two-body

PMFs (i.e., $W_{\text{2B}}^{\text{MS-CG}}(r_{13})$ and $W_{\text{2B}}^{\text{MS-CG}}(r_{23})$), for interactions between PGN 1 and 3 and PGN 2 and 3, accordingly:

$$\begin{aligned}\tilde{W}_{\text{3B}}^{\text{MS-CG}} =& W_{\text{3B}}^{\text{MS-CG}}(r_{12}, r_{13}, r_{23}) + \\ & W_{\text{2B}}^{\text{MS-CG}}(r_{13}) + W_{\text{2B}}^{\text{MS-CG}}(r_{23})\end{aligned} \tag{3.12}$$

We note that $W_{\text{2B}}^{\text{MS-CG}}(r_{12})$ has a value of zero and is excluded, since the pulling direction of SMD is always perpendicular to the axis connecting PGN 1 and 2. Based on Equation 3.12, we can calculate the pure three-body interaction term $W_{\text{3B}}^{\text{MS-CG}}$ by subtracting the $W_{\text{2B}}^{\text{MS-CG}}(r_{13})$ and $W_{\text{2B}}^{\text{MS-CG}}(r_{23})$ terms from $\tilde{W}_{\text{3B}}^{\text{MS-CG}}$. We will show in Section 3.3.2 that the many-body PMF for PGNs considered in this work is dominated by two- and three-body interactions. Therefore, we can write:

$$\begin{aligned}W_{\text{mB}}^{\text{MS-CG}} =& W_{\text{3B}}^{\text{MS-CG}}(r_{12}, r_{13}, r_{23}) + W_{\text{2B}}^{\text{MS-CG}}(r_{12}) + \\ & W_{\text{2B}}^{\text{MS-CG}}(r_{13}) + W_{\text{2B}}^{\text{MS-CG}}(r_{23})\end{aligned} \tag{3.13}$$

To evaluate the significance of three-body interactions, Figure 3.4 bottom panel compares 2-dimensional slices of $W_{\text{2B, total}}^{\text{MS-CG}}$ and $W_{\text{mB}}^{\text{MS-CG}}$ for PGNs with $\alpha = 4$ and 8 by placing these PGNs on the vertices of isosceles triangles, where r_{13} and r_{23} are set to be equal, denoted as r, and r_{12} are fixed at various distances. After sampling these configurations, we plot two- and many-body PMFs as a function of r. The total two-body PMFs, $W_{\text{2B, total}}^{\text{MS-CG}}$, are obtained by summing the contributions from all three components of two-body interactions, i.e., $W_{\text{2B, total}}^{\text{MS-CG}} = W_{\text{2B}}^{\text{MS-CG}}(r_{12}) + W_{\text{2B}}^{\text{MS-CG}}(r_{13}) + W_{\text{2B}}^{\text{MS-CG}}(r_{23})$. The many-body PMFs are determined by summing the two- and three-body PMFs, i.e., $W_{\text{2B, total}}^{\text{MS-CG}} + W_{\text{3B}}^{\text{MS-CG}}$. We observe results consistent with the two-body PMF calculations, where the magnitude of $W_{\text{mB}}^{\text{MS-CG}}$ increases when polymer length is increased from $\alpha = 4$ to 8. Moreover, three-body contributions are evident from the deviation between $W_{\text{2B, total}}^{\text{MS-CG}}$ and

$W_{\mathrm{mB}}^{\mathrm{MS\text{-}CG}}$, where $W_{\mathrm{2B,\,total}}^{\mathrm{MS\text{-}CG}}$ alone underestimates the inter-PGN interactions. We observe more comparable energy profiles between $W_{\mathrm{2B}}^{\mathrm{MS\text{-}CG}}$ and $W_{\mathrm{mB}}^{\mathrm{MS\text{-}CG}}$ for $\alpha = 4$, indicating weaker three-body interactions at shorter polymer length. In contrast, PGNs with $\alpha = 8$ exhibit stronger and longer-ranged three-body repulsive interactions. The intensified three-body contribution is consistent with the results reported in literature [112, 122, 123, 124, 125, 126], which suggests that PGNs with longer capping layers generally exhibit stronger repulsive many-body interactions due to the complex ligand entanglement and exclusionary effects.

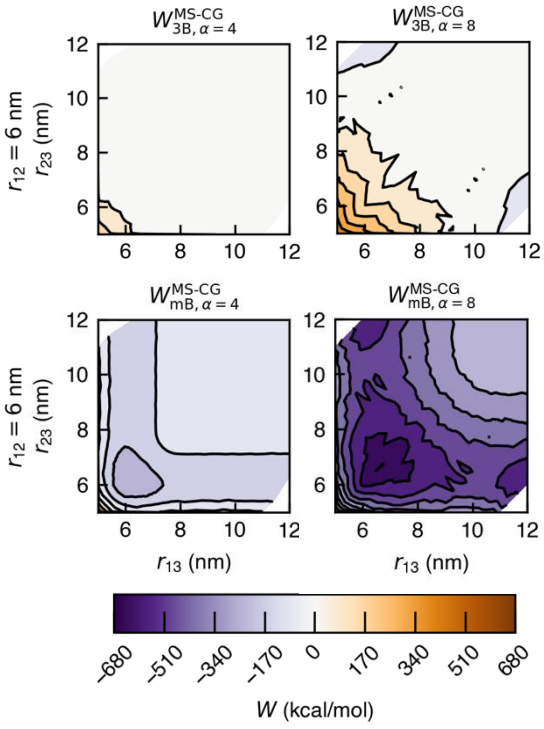

Figure 3.5: **Three-body PMF $W_{\mathbf{3B}}^{\mathbf{MS\text{-}CG}}$ and many-body PMF $W_{\mathbf{mB}}^{\mathbf{MS\text{-}CG}}$ heatmaps for $\alpha = 4$ and 8 PGNs with r_{12} fixed at 6 nm.** The contour interval is 80 kcal/mol.

A more detailed view of the three-body PMFs is provided in Figure 3.5. Overall, we find that many-body interactions are more attractive and exhibit additional minima in systems with larger α. However, three-body interactions introduce repulsion at small inter-PGN separation distances that grows with increasing α. We showcase

many-body PMF data across a broader range of r_{12} at all α values in Figures S2-S7.

3.3.2 OS ChIMES Models

This study represents the first application of ChIMES to CG systems and also presents the unique challenge of generating models describing many-body interactions between nanoparticles. Hence, before exploring the development and application of X-ChIMES models, we first assess the efficacy and accuracy of ChIMES for describing PGNs with *fixed* α.

In the following section, we will use the terms OS-CG and ChIMES interchangeably. We generated a total of 12 individual OS-CG models corresponding to PGNs of $\alpha = 3$ to 8, described through either two-body interactions:

$$W_{\mathrm{2B}}^{\text{OS-CG}}(r_{ij}) = \sum_{i<j} {}^{2}E_{ij}, \tag{3.14}$$

or many-body interactions that are the sum of two-body and three-body interactions:

$$\begin{aligned} W_{\mathrm{mB}}^{\text{OS-CG}}(r_{ij}, r_{ik}, r_{jk}) = W_{\mathrm{2B}}^{\text{OS-CG}}(r_{ij}) + W_{\mathrm{2B}}^{\text{OS-CG}}(r_{ik}) \\ + W_{\mathrm{2B}}^{\text{OS-CG}}(r_{jk}) + \sum_{i<j<k} {}^{3}E_{ijk}. \end{aligned} \tag{3.15}$$

Models were trained on two-body PMFs (i.e., $W_{\mathrm{2B}}^{\text{MS-CG}}$) and three-body PMFs (i.e., $W_{\mathrm{3B}}^{\text{MS-CG}}$) described in Section 3.3.1. Model hyperparameters are provided in Table 3.2. Polynomial orders were selected to achieve the best fit with the lowest possible polynomial orders to maximize model efficiency; we note that larger polynomial orders were used for PGNs with greater α to accommodate their more featured PMF surfaces (see, e.g., Fig. 3.5).

Figure 3.6 provides the performance of two- and many-body models for systems with $\alpha = 4$ and 8; we observe excellent model performance across all models with distinct α. Overall, we find Pearson correlation values of $R^2 \sim 0.9998$ and ~ 0.9999

Table 3.2: ChIMES and extended ChIMES (X-CHIMES) fitting parameters

Hyperparameter	ChIMES		X-ChIMES
	$\alpha = 3-6$	$\alpha = 7-8$	
$\mathcal{O}_{r,\mathrm{2B}}$	12	12	8
$\mathcal{O}_{r,\mathrm{3B}}$	4	7	3
$\mathcal{O}_{\alpha,\mathrm{2B}}$	-	-	3
$\mathcal{O}_{\alpha,\mathrm{3B}}$	-	-	2
$r_{\mathrm{c,in}}$	4.85	4.85	4.85
$r_{\mathrm{c,out}}$	10.0	12.0	12.0
$\alpha_{\mathrm{c,in}}$	-	-	3
$\alpha_{\mathrm{c,out}}$	-	-	8
λ_{Morse}	6.0	6.5	6.0
κ	10^{-5}	10^{-5}	10^{-5}
f_o	0.75	0.75	0.75

Units: nm

for two-body $\alpha = 4$ and 8 models, and ~ 0.9994 and ~ 0.9939 for many-body $\alpha = 4$ and 8 models. A more comprehensive overview of $W_{\mathrm{mB}}^{\mathrm{OS\text{-}CG}}$ including PES heatmaps and error analysis for a broader range of r_{12} at all α values is provided in Supplementary Information Figures 3.14 to 3.19.

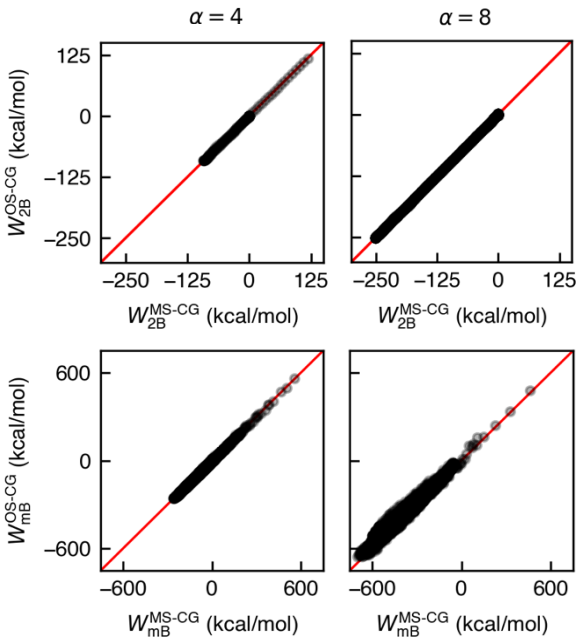

Figure 3.6: **Performance of ChIMES model $W^{\mathrm{OS\text{-}CG}}$ fits for PGNs with $\alpha = 4$ and 8.** Parity plots compare ChIMES model predictions with ground truth data (top: $W_{\mathrm{2B}}^{\mathrm{MS\text{-}CG}}$; bottom: $W_{\mathrm{mB}}^{\mathrm{MS\text{-}CG}}$). The red line is the $y = x$ line.

We further validate the performance of the many-body OS-CG models by comparing ChIMES-predicted and MS-CG equilibrium structures for PGN trimers initialized

with different configurations. As shown in Figure 3.7, trimer configurations were initialized with a 1-2-3 PGN-PGN-PGN angle θ_0 of 120° and 90°; the trimers were then equilibrated in NVT MD simulations. For each θ_0, we conducted five independent replica simulations. We note that due to the reduction in the degrees of freedom in the OS-CG resolution, simulation time cannot be directly compared between the MS-CG and OS-CG resolutions.

Figure 3.7 provides the evolution of three PGN distances over each simulation, i.e., $r_{12}, r_{13},$ and r_{23}. For PGNs with $\alpha = 4$, we observe that for the $\theta_0 = 90°$ case, both MS-CG and OS-CG models equilibrated to an equilateral triangular trimer configuration with $(r_{12}, r_{13}, r_{23}) \simeq (6.0, 6.0, 6.0)$ nm, corresponding to the energy minima shown in both training data and OS-CG model in the bottom left of Figure 3.5 and Figure 3.7. However, for the $\theta_0 = 90°$ case, the MS-CG model equilibrated to a triangular trimer configuration with $(r_{12}, r_{13}, r_{23}) \simeq (6.0, 7.0, 6.0)$ nm, while the OS-CG model reached a similar configuration as the one initialized at $\theta_0 = 90°$. This discrepancy is attributed to the timescale differences between the MS-CG and OS-CG models. We anticipate that, given a longer simulation time, the MS-CG model will ultimately reach the same equilibrium configuration as the OS-CG model. For these systems, equilibration in the OS-CG model is achieved within 2 million MD steps irrespective of the starting configuration. Specifically, the OS-CG model initialized with $\theta_0 = 120°$ is significantly faster than the MS-CG model.

For $\alpha = 8$, the top right of Figure 3.7 shows that trimers initialized at θ_0 of 90° using the MS-CG model equilibrate to a larger equilateral triangular trimer configuration with $(r_{12}, r_{13}, r_{23}) \simeq (6.5, 6.5, 6.5)$ nm within 10 ns, corresponding to the global energy minima of both $W_{\text{mB}}^{\text{MS-CG}}$ and $W_{\text{mB}}^{\text{OS-CG}}$ in the bottom right of Figure 3.5 and the bottom middle of Figure 3.7. However, the OS-CG models equilibrate to a

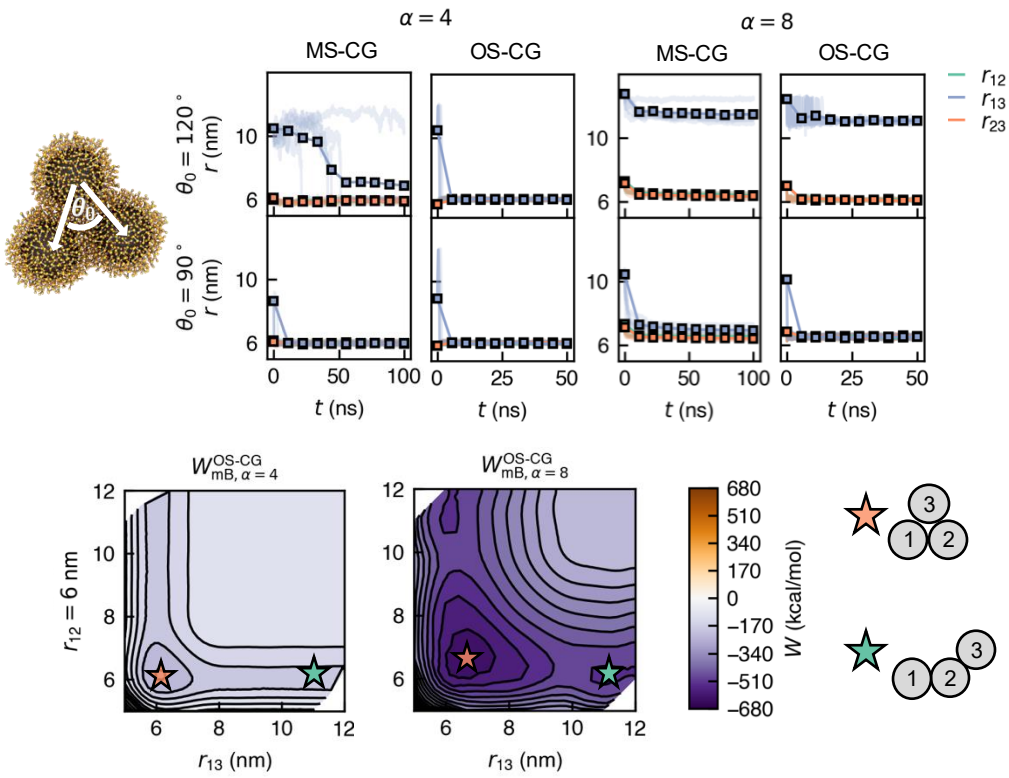

Figure 3.7: **OS-CG model validation with PGN trimers. Top**: Initial trimer configurations and relaxation of trimers initialized from different configurations for $\alpha = 4$ and 8. Transparent lines indicate individual replica simulations, and the solid lines with square markers represent the average values over all five replicas. **Bottom**: $W_{\mathrm{mB}}^{\mathrm{OS\text{-}CG}}$ for $\alpha = 4$ and 8 as a function of trimer configuration with fixed inter-PGN distance r_{12} and various r_{13} and r_{23}. The starred locations in PMFs correspond to schematic drawings of PGN configurations on the right side. The contour interval is 42.5 kcal/mol.

similar trimer configuration with $(r_{12}, r_{13}, r_{23}) \simeq (6.0, 6.0, 6.0)$ nm. Again, we anticipate that, given a longer simulation time, the MS-CG model will ultimately reach the same equilibrium configuration as the OS-CG model. In contrast, $\alpha = 8$ trimers initialized at $\theta_0 = 120°$ equilibrate to a configuration with the larger enclosed angle of $(r_{12}, r_{13}, r_{23}) \simeq (6.0, 11.0, 6.0)$ nm in simulations using both the MS-CG and OS-CG models, corresponding to the second lowest PMF minimum as illustrated in the bottom middle of Figure 3.7. This local minimum corresponds to a nearly linear trimer configuration illustrated by the green star symbol in the bottom right

of Figure 3.7, which explains why the r_{23} distance remained at approximately 11 nm for the trimers initialized at $\theta_0 = 120°$. Such local minima are not observed in the $\alpha = 4$ heat map in bottom left of Figure 3.7, which aligns with the observation that for $\alpha = 4$ trimers initiated at $\theta_0 = 120°$, both the r_{13} and r_{23} converged to the energy minimum location (i.e., around 6.0 nm). Ultimately, this study demonstrates (1) fidelity of the PMF calculated with the FR method, (2) effectiveness of ChIMES for learning highly coarse-grained systems with many-body interactions, and (3) acceleration of simulation dynamics afforded by the OS-CG model due to the reduction of overall conformational degrees of freedom.

The performance of the OS-CG models was further evaluated by comparing equation of state (EOS) predictions for PGNs in a BCC and FCC lattice with those generated via the MS-CG models. To generate this data, a series of NPT MD simulations were run at pressures of 10, 100, and 1000 atm for 37.5 and 5 ns for the MS-CG and OS-CG models, respectively. As is shown in Figure 3.8, we find that for $\alpha = 4$, many-body OS-CG model predictions are in good agreement with MS-CG predictions at all pressures. In contrast, the two-body OS-CG model underpredicts equilibrium volume, and for the BCC system at 1000 atm, becomes unstable due to the absence of repulsive three-body interactions, allowing particles to sample unphysical distances below the model inner cutoff.

For PGNs with $\alpha = 8$, two-body OS-CG simulations are only stable at the lowest pressure and severely underpredict the volume of the MS-CG system. The many-body OS-CG model exhibits the expected behavior, i.e., becoming considerably denser with increasing pressure. However, the MS-CG model only exhibits small changes in system volume over the duration of the simulations due to the explicit treatment of longer polymer ligands, which causes considerably slower relaxation dy-

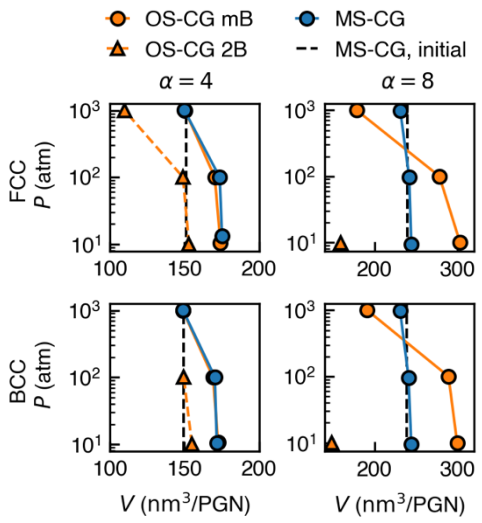

Figure 3.8: **Pressure-volume relationship predicted by MS-CG and OS-CG models for PGNs with $\alpha = 4$ and 8.**

namics. Consequently, the system remains in the initial metastable state throughout the simulation.

Figure 3.9 provides the corresponding RDF for each NPT simulation. We find excellent agreement between the many-body MS-CG and OS-CG models for PGNs with $\alpha = 4$ in both FCC and BCC structures. However, for $\alpha = 8$, a mismatch in both the FCC and BCC structures is observed due to the MS-CG systems being kinetically trapped in their initial metastable state.

The top right of Figure 3.9 also highlights the intriguing behavior of PGNs with $\alpha = 8$ initialized in the FCC lattice sites, where OS-CG RDF peaks vanish at 10 and 100 atm. Upon visualizing the simulation trajectories, at 10 atm, two out of the five replicas contain a mixture of FCC, BCC, and liquid-like structure at late simulation times, while the rest remain stable in the FCC structure. Additionally, for 100 atm, all five replica simulations show the tendency to transition from FCC to BCC, resulting in a FCC/BCC/liquid mixed state. In contrast, at 1000 atm, all five replicas remain stable in the FCC structure. This behavior occurs because at lower

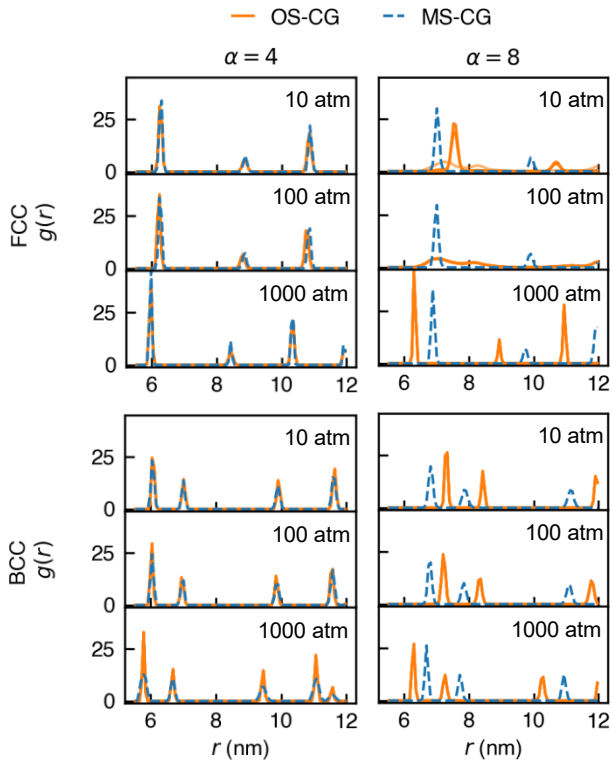

Figure 3.9: **Radial distribution functions (RDFs) for simulations using MS-CG and many-body OS-CG models at different pressures, for PGNs with $\alpha = 4$ and 8**. For OS-CG models, orange transparent lines represent individual replica simulations. Almost every OS-CG replica simulation result overlaps, except the simulations in the top right corner. The blue dashed lines indicate the averaged RDFs calculated from five replica simulations using the MS-CG models.

pressures, systems can undergo an FCC to BCC transition, while at higher pressures, they are more likely to become kinetically trapped in their initial crystal structures. To verify our observation, we calculated the Steinhardt order parameter [62] $\overline{q}_6$ for the NPT MD simulation trajectories of $\alpha = 8$ FCC structures, and we present the evolution of $\overline{q}_6$ during simulations in Figure 3.20.

To qualitatively investigate our observations and to probe the most stable structure self-assembled by these PGNs, we equilibrate an $\alpha = 8$ FCC crystal in a box with linearly expanding box length. By continuously increasing the box length without rescaling particle positions, we allow the kinetically trapped system to relax and

reconfigure into its energetically favorable structure at lower pressures. We initialized the FCC crystal structure with lattice constant a_0 equal to 9.2 nm and gradually increased the box volume over the first 10 ns of the total 30 ns simulation time. We illustrate the snapshots during the expanding box simulation for $\alpha = 4$ and 8 in the Supplementary Information Figure 3.21. During the box expansion, the FCC structure first melts into a liquid droplet. As the simulation progresses, BCC structure begins to nucleate within the liquid droplet and continues to grow. Eventually, the system converts into a liquid droplet with a BCC structure at the core, which demonstrates the potential of a BCC structure self-assembled from $\alpha = 8$ PGNs. For PGNs with $\alpha = 4$, we performed the same simulation in a linearly expanding box and observed that the $\alpha = 4$ system is stable in a liquid droplet with co-existing FCC and BCC structure at the core. This result, together with the RDF data, shows that FCC and BCC structures have comparable stability for $\alpha = 4$. To complement our findings, we provide the approximated phase diagram in the Supplementary Information Figure 3.22 by comparing the enthalpy at $T = 0$ K between the perfect FCC and BCC crystals at the same statepoint via single-point energy calculation. We note that our observed tendency for phase transformation in PGNs with $\alpha = 8$ qualitatively aligns with the OTM model developed by Travesset [113], which explore the relationship between the free energy of PGNs at zero pressure and ligand capping layer thickness λ, defined as the ratio of the maximal polymer length L to the core radius R, i.e., L/R. Note that this λ differs from the ChIMES fitting parameter λ_{Morse} mentioned in Section 3.2.2. Travesset demonstrates that FCC and BCC structures are the most stable crystal configurations for PGNs, with a transition point from FCC to BCC occurring at $\lambda \approx 0.66$ and a flat free energy surface for the remaining λ values. This result is intuitive in the sense that an increase in λ

corresponds to a thicker polymer grafting layer, and thus, the less dense BCC structures are favored over the more compact FCC structures. The transition from FCC to BCC also signifies a shift from the regime of two-body interaction dominance to where three-body interactions become significant.

Table 3.3 provides λ values for our PGNs, determined based on the partial number integral between the MS-CG core particles and the MS-CG polymer beads of each PGN, as shown in Supplementary Information Figure 3.9. Beginning with $\alpha = 5$, λ exceeds 0.66, resulting in more significant three-body interactions. This analysis is consistent with Figure 3.5, where three-body interactions are weaker for $\alpha = 4$ but prominent for $\alpha = 8$. As previously noted, we did not observe a transition from FCC to BCC for PGNs with $\alpha = 4$. In light of the λ value calculations, we attribute this behavior to the comparable stability of the FCC and BCC structures at $\alpha = 4$. Specifically, the λ value of 0.64 is close to the transition threshold of 0.66, which quantitatively explains the FCC/BCC co-existing structure for $\alpha = 4$ in the expanding box simulation described above.

Table 3.3: PGNs capping layer thickness λ for particles with different polymer lengths α

α					
3	4	5	6	7	8
0.45	0.64	0.85	1.02	1.21	1.33

3.3.3 OS X-ChIMES Models for Inverse Design

We now focus on development of an extended OS-CG model (X-OS-CG), i.e., an X-ChIMES model that unifies the description of inter-PGN interactions with varying ligand length α. We note that, in this section, we will use the terms X-OS-CG and X-ChIMES interchangeably. These models are defined as:

$$W_{2B}^{\text{X-OS-CG}}(\alpha, r_{ij}) = \sum_{i<j} {}^{2}E_{\alpha ij} \tag{3.16}$$

$$
\begin{aligned}
W_{\mathrm{mB}}^{\text{X-OS-CG}}(\alpha, r_{ij}, r_{ik}, r_{jk}) &= W_{\mathrm{2B}}^{\text{X-OS-CG}}(\alpha, r_{ij}) \\
&+ W_{\mathrm{2B}}^{\text{X-OS-CG}}(\alpha, r_{ik}) + W_{\mathrm{2B}}^{\text{X-OS-CG}}(\alpha, r_{jk}) \\
&+ \sum_{i<j<k} {}^{3}E_{\alpha ijk}
\end{aligned}
\tag{3.17}
$$

where ${}^{2}E_{\alpha ij}$ and ${}^{3}E_{\alpha ijk}$ are two- and three-body X-ChIMES energies defined in Equation 3.10 and 3.11.

The X-OS-CG model was generated by separately fitting two- and three-body PMFs, $W_{\mathrm{2B}}^{\text{X-OS-CG}}$ and $W_{\mathrm{3B}}^{\text{X-OS-CG}}$, and taking $W_{\mathrm{mB}}^{\text{X-OS-CG}} = W_{\mathrm{2B}}^{\text{X-OS-CG}} + W_{\mathrm{3B}}^{\text{X-OS-CG}}$. The resulting model treats α as a continuous variable, e.g., where $\alpha = 2.5$ can produce valid energy predictions. We remind the reader that α corresponds to the number of beads in each MS-CG PGN ligand. Therefore, within the context of our α, these fractional bead counts can account for the fact that each bead is coarse-grained such that multiple repeat units are contained in a single bead. Moreover, the use of a continuous α dimension makes resulting models smoothly differentiable, allowing for use in general molecular simulation where forces are computed based on potential energy.

To evaluate the effectiveness of X-OS-CG, a model was trained on PMFs with $\alpha = 3,\ 4,\ 6,$ and 8 while $\alpha = 5$ and 7 are held out for validation. Hyperparameter optimization was performed to identify model polynomial orders affording a balance in model accuracy and complexity. For this study, we found optimal polynomial orders to be $\mathcal{O}_{s,\mathrm{2B}} = 8$ and $\mathcal{O}_{\alpha,\mathrm{2B}} = 3$ for two-body interactions and $\mathcal{O}_{s,\mathrm{3B}} = 3$ and $\mathcal{O}_{\alpha,\mathrm{3B}} = 2$ for three-body interactions.

When fitting $W_{\mathrm{2B}}^{\text{X-OS-CG}}$ and $W_{\mathrm{3B}}^{\text{X-OS-CG}}$, a global inner radial cutoff of 4.85 nm and an outer radial cutoff of 12.0 nm was used. A Morse transformation was used on r_{ij} with $\lambda_{\mathrm{Morse}} = 6.0$ nm, approximately corresponding to the most attractive inter-

PGN distance. A direct transformation was used on α. Inner and outer cutoffs for the α dimension were set to 3 and 8, corresponding to the minimum and maximum number of beads per polymer chain over which models were trained. The complete list of fitting parameters is in Table 3.2.

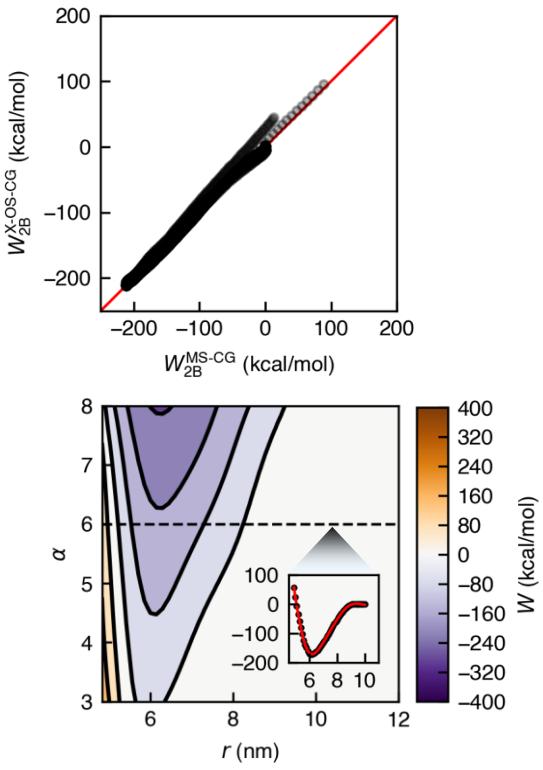

Figure 3.10: **Performance of the fits of two-body extended ChIMES (X-ChIMES) model $W_{2B}^{X\text{-}OS\text{-}CG}$. Top**: Parity plot comparing two-body X-ChIMES model predictions with ground truth data $W_{2B}^{MS\text{-}CG}$. We use $\alpha = 5$ and 7 (i.e., hold-out validation data) to demonstrate X-ChIMES' excellent performance. **Bottom**: Two-body X-ChIMES model predictions as a function of inter-PGN distance r and polymer length α, illustrating simultaneous predictions of two-body PMFs of all six PGNs. The small inset figure depicts $\alpha = 6$ slice where the red line represents $W_{2B}^{X\text{-}OS\text{-}CG}$ and black dots represent $W_{2B}^{MS\text{-}CG}$. The contour interval is 80 kcal/mol.

Figures 3.10 and 3.11 provide validation of the two- and three-body components of the X-OS-CG models in terms of parity plots corresponding to the *validation dataset*. Overall, we find excellent agreement with Pearson correlation values R^2 of ~ 0.9957 and ~ 0.9699, respectively. Training parity plots are given in Figure S12.

Figures 3.10 and 3.11 also provide corresponding PMF heatmaps. We observe expected trends, with $W_{\mathrm{2B}}^{\mathrm{X\text{-}OS\text{-}CG}}$ exhibiting a continuously increasing PMF well depth, well width, and well location near $r = 6$ nm as α increases. Figure 3.11 bottom panel shows $W_{\mathrm{3B}}^{\mathrm{X\text{-}OS\text{-}CG}}$ for different α values near the optimal bonding distance, i.e., $r_{12} = 6$ nm, where $\alpha = 4$ and 8 correspond to the training dataset and $\alpha = 5$ and 7 correspond to the validation dataset. $W_{\mathrm{3B}}^{\mathrm{X\text{-}OS\text{-}CG}}$ shows a smooth three-body energy landscape and the expected trend showing intensified three-body interaction strength with increased polymer length α. These findings emphasize the robustness and predictive power of the newly developed X-ChIMES framework in unifying both two-body and three-body interactions across a range of polymer lengths.

3.3.4 Digital Alchemy with X-ChIMES Model

Here, we provide an example application of DA applied to our X-ChIMES model. This application focuses on the Monte Carlo realization of DA, henceforth referred to as Alch-MC. Intuitively, the goal of applying DA via Alch-MC is to optimize α, which corresponds to a maximized likelihood of producing the specified SAS. To achieve this, we maximize the extended Boltzmann distribution as described in Equation 1.10.

We choose FCC and BCC crystals as the target SAS where the interactions between PGNs are modeled using the $W_{\mathrm{mB}}^{\mathrm{X\text{-}OS\text{-}CG}}$. To perform the Alch-MC simulation, we first packed PGNs onto FCC and BCC lattice sites using the biased Hamiltonian outlined in Equation 1.7 to constrain the PGNs onto lattice sites. We note that the potential energy term $U(\boldsymbol{r}, \alpha)$ is now $W_{\mathrm{mB}}^{\mathrm{X\text{-}OS\text{-}CG}}$. Each Alch-MC sweep involves two types of MC moves: configurational moves and alchemical moves. Configurational moves attempt to randomly perturb every particle's position, while alchemical moves attempt to randomly adjust the α value. Both MC moves are based on the Metropo-

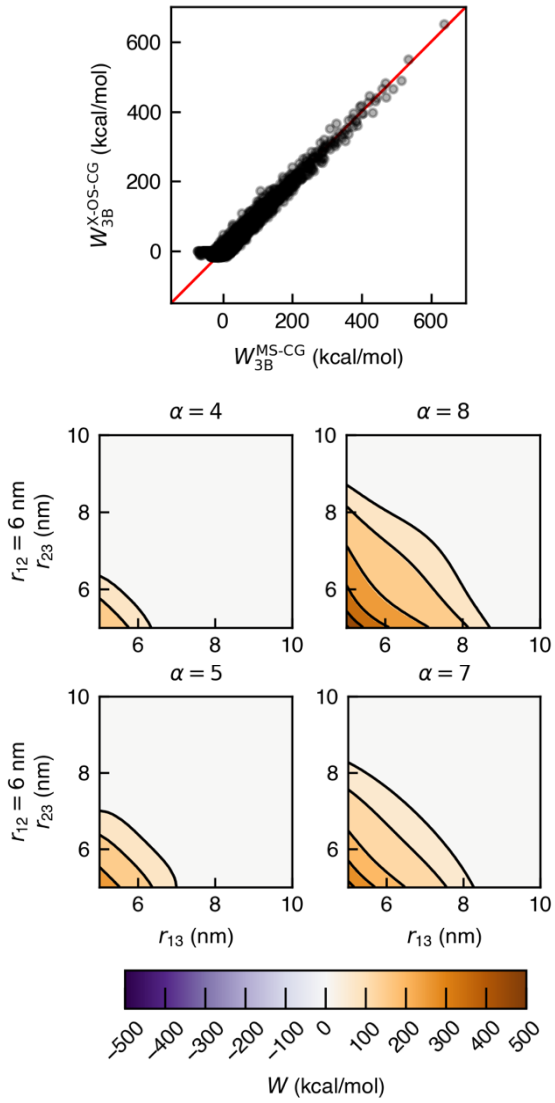

Figure 3.11: **Performance of the fits of three-body extended ChIMES (X-ChIMES) model $W_{3B}^{X\text{-}OS\text{-}CG}$. Top**: Parity plot comparing three-body X-ChIMES model predictions with ground truth data $W_{3B}^{MS\text{-}CG}$. We use $\alpha = 5$ and 7 (i.e., hold-out validation data) to demonstrate X-ChIMES' excellent performance. **Bottom**: three-body X-ChIMES model predictions at different α as a function of trimer configurations, i.e., fixed $r_{12} = 6$ nm and varied r_{13} and r_{23}. $\alpha = 4$ and 8 are included for X-ChIMES model training and $\alpha = 5$ and 7 are excluded for hold-out validation. The contour interval is 100 kcal/mol.

lis acceptance criterion. We perform a total of 10000 Alch-MC sweeps. The biased potential was turned off at $t_{\text{end}} = 7500$ sweeps to ensure structural stability at the final optimized α.

$W_{\mathrm{mB}}^{\text{X-OS-CG}}$ consists of attractive two-body interactions dominating at large inter-PGN distances and repulsive three-body interactions being prominent at shorter distances. Due to the interplay between the range of the two- and three-body interactions and the observation that both types of interactions intensify with increasing polymer length, we expect the optimized α value to change with changing lattice constant a_0.

We first initialized FCC and BCC crystals with three different lattice constants: $a_0 = 8.5$, 7.6, and 7.1 nm for FCC crystals and $a_0 = 6.9$, 6.2, and 5.8 nm for BCC crystals. Next, we conducted Alch-MC simulations with the initial α guesses of 3, 5, and 7, as shown in Figure 3.12. For each pair of a_0 and initial α, we ran five replica simulations. The top row of Figure 3.12 illustrates α evolution during Alch-MC simulations for FCC crystals with various a_0. We observe that at the largest a_0 tested, the optimized α is either 3 or 8, depending on the initial guess of α. Even though $\alpha = 8$ corresponds to the global energy minimum, the system with an initial α of 3 is not able to overcome the energy barrier to reach $\alpha = 8$, illustrating the possibility of multiple local optima. At the smaller a_0 tested, we observe a unique optimum equal to 3, regardless of the initial guess. This result is expected since these densely packed structures with small a_0 prefer smaller α whose repulsive three-body interactions are weaker. Furthermore, such a transition between optimal α due to the target structure density features a unique transition from a regime dominated by strong three-body interactions to two-body interactions. Additionally, we observe similar results for BCC systems in the bottom row of Figure 3.12. Our observations here are supported by the single-point energy calculation illustrated in Supplementary Information Figure 3.25.

Evolution of the energy in DA simulations is shown in Supplementary Information

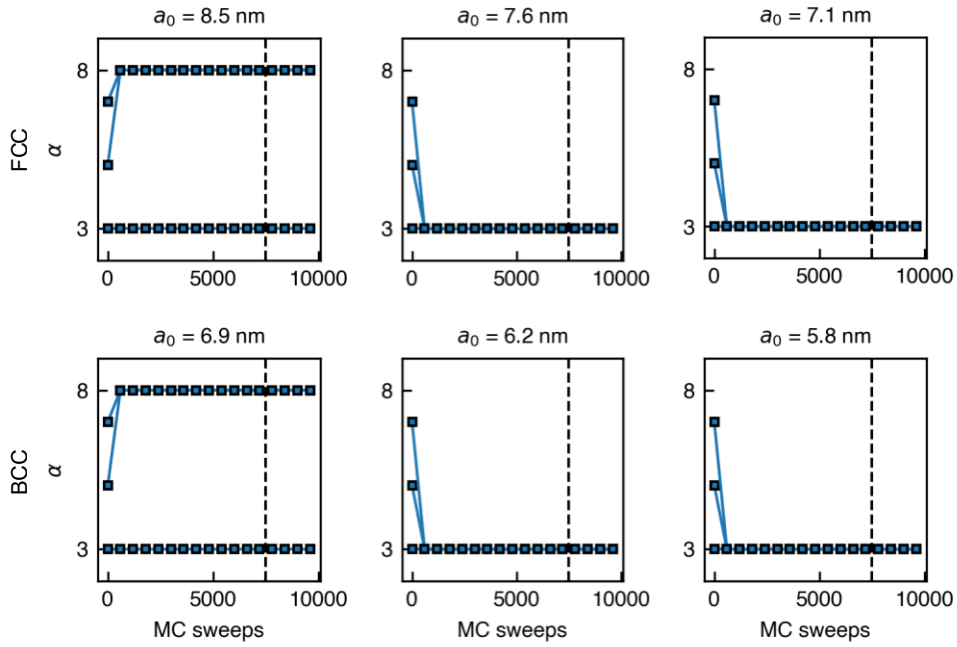

Figure 3.12: **Alchemical Monte-Carlo (Alch-MC) simulation trajectories.** We illustrate the Alch-MC simulation trajectories using many-body X-OS-CG model $W_{\mathrm{mB}}^{\text{X-OS-CG}}$ by plotting α evolution over Alch-MC sweeps. The blue lines with square markers indicate the averaged trajectories of five replicas in each set of simulations, with each initialized at $\alpha = 3$, 5, and 7. The black dashed lines indicate t_{end} during the Alch-MC sweep, where we turn off the biased potential for the target crystal, described in equation 1.8. **Top**: Alch-MC simulations for target FCC crystals with decreasing lattice constant a_0. **Bottom**: Alch-MC simulations for target BCC crystal with decreasing a_0.

Figure 3.26. For both FCC and BCC structures at larger lattice constants, systems initialized with $\alpha = 3$ exhibit relatively constant energy profiles over the course of the simulation. This behavior is consistent with that observed in DA simulations, suggesting that the system may be trapped in a local minimum and unable to overcome the energy to reach the global energy minima at that lattice constant. For systems with smaller lattice constants, all DA simulation trajectories converge to the global minima regardless of the initial α value, which corresponds to the DA simulation results.

3.4 Conclusion

Inverse design algorithms like DA represent powerful tools for optimizing the self-assembly of PGNs for target applications. The predictive power of these tools can be significantly enhanced by building more accurate, efficient, and chemically aware design models to guide the exploration. To meet these needs, this work explores the use of the ChIMES ML-IAM [141] to generate highly coarse-grained yet highly accurate models for PGN interactions based on data generated at the MS-CG resolution. We show that these models yield predictions in excellent agreement with the ground truth data, yet are orders of magnitude more computationally efficient. Moreover, we present an efficient sampling strategy for generating PMF training data needed to train these PGN interaction models by integrating the FR method [38] into the efficient many-body grid sampling scheme [126]. We then demonstrate an extension of the ChIMES CG ML-IAM to describe how these interactions change as a function of PGN attributes - polymer length in this case - and demonstrate how these models can be used within DA simulations for inverse design of target SAS.

Our simulations of FCC and BCC PGN structures are found to exhibit interesting behaviors. Namely, for $\alpha = 8$, our OS-CG model predicted that systems initialized as FCC crystals exhibited a tendency to undergo phase transition to BCC at 10 atm and 100 atm due to the strong repulsive three-body interactions in condensed FCC crystal structures. These results are not observed in the MS-CG simulation due to the limited timescales accessible to this resolution. Our observations are consistent with predictions of Travesset's OTM model [113], where PGNs with longer polymer ligands favor the BCC structure during the self-assembly process. These results demonstrate that the acceleration achieved through our highly coarse-grained OS-

CG representation affords access to the longer simulation timescales that can be relevant to system relaxation. Furthermore, our findings on phase transition tendencies highlight the interplay between two-body and three-body interactions and their dependence on the inter-PGN distance.

Our X-ChIMES DA [30, 31] application demonstrates that our models preserve physical specificity that is often lost during the CG process. Using PGNs with distinct polymer lengths as a prototypical design parameter, we demonstrate the model's potential for nanoparticle design because the optimized α using DA corresponds to a specific chemical or physical feature. Ultimately, this work lays a foundation for further advancements in physics-informed CG ML-IAMs to facilitate the design and discovery of novel nanomaterials.

3.5 Supplementary Information

3.5.1 One-Site Coarse-Grain (OS-CG) Mapping

The multi-site CG (MS-CG) models describe interactions between PGNs comprised of n+N particles, where n represents the number of polymer beads, and N represents the number of core particles. Bead positions are denoted as $\boldsymbol{r}^{n+N} = [\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_n, \boldsymbol{r}_{n+1}, ..., \boldsymbol{r}_{n+N}]$. The configurational Boltzmann distribution of PGNs is expressed as $p(\boldsymbol{r}^{n+N})$. Our goal is to establish a CG mapping from the MS-CG to OS-CG resolution, reducing the number of PGNs from n+N to N, where $N \ll n$. The OS-CG model now has positions denoted as $\boldsymbol{r}^N = [\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N]$. This CG mapping is realized through a generic operation, ξ, such that $\boldsymbol{r}^N = \xi(\boldsymbol{r}^{n+N})$. We choose the core particle positions as our OS-CG sites, implying that the function ξ serves as a simple filter to the MS-CG core particle positions.

To generate an accurate OS-CG model using the aforementioned mapping, it is essential to ensure that the resulting OS-CG model is thermodynamically consis-

tent with the original MS-CG model. Such consistency can be demonstrated by projecting the configurational Boltzmann distribution of the MS-CG model into the low-dimensional representation of the OS-CG model as follows:

$$p^{\text{OS-CG}}(\boldsymbol{r}^N) = \frac{\int p(\boldsymbol{r}^{n+N})\delta(\boldsymbol{r}^N - \xi(\boldsymbol{r}^{n+N}))d\boldsymbol{r}^{n+N}}{\int p(\boldsymbol{r}^{n+N})d\boldsymbol{r}^{n+N}} \tag{3.18}$$

where $p^{\text{OS-CG}}(\boldsymbol{r}^N)$ represents the Boltzmann distribution of the OS-CG model. Consequently, our focus shifts to obtaining the PMF, denoted as $W^{\text{OS-CG}}$, that yields $p^{\text{OS-CG}}(\boldsymbol{r}^N)$ described by the following equation:

$$W^{\text{OS-CG}}(\boldsymbol{r}^N) = -k_{\text{B}}T \ln p^{\text{OS-CG}}(\boldsymbol{r}^N) + \text{c}_0 \tag{3.19}$$

where k_{B} is the Boltzmann constant, T is temperature, and c_0 is an arbitrary constant. As a result of the CG mapping, we treat $W^{\text{OS-CG}}$ as the potential energy of the OS-CG model to account for the loss in configurational entropy due to the CG process.

3.5.2 Martini Model Description

The non-bonded interaction between C1 beads are defined as:

$$U_{\text{LJ,C1-C1}} = 4\epsilon_{\text{C1-C1}} \left[\left(\frac{\sigma_{\text{C1-C1}}}{r_{\text{C1-C1}}} \right)^{12} - \left(\frac{\sigma_{\text{C1-C1}}}{r_{\text{C1-C1}}} \right)^{6} \right] \tag{3.20}$$

where the Lennard-Jones (LJ) parameters $\epsilon_{\text{C1-C1}}$ and $\sigma_{\text{C1-C1}}$ are based on the Martini 3 model [132]. Additionally, we add shifted LJ (SLJ) potential between core particles to C1 beads and other core particles to prevent polymer beads from moving inside the PGN cores, described by U_{SLJ}:

$$U_{\text{SLJ}} = 4\epsilon \left[\left(\frac{\sigma}{r - \Delta} \right)^{12} - \left(\frac{\sigma}{r - \Delta} \right)^{6} \right], \; r > \Delta \tag{3.21}$$

where the shifted distance Δ equals the core radius of 2.35 nm. Moreover, for bonded interactions, we use a harmonic stretching potential U_{bond} to model the connected core-bead and bead-bead interactions:

$$U_{\text{bond}} = \frac{1}{2} k_{\text{bond}} (r - r_0)^2 \tag{3.22}$$

where k_{bond} is the spring constant and r_0 is the equilibrium distance of the core-bead or bead-bead connections. Additionally, the cosine squared angle potential is applied to constrain the angle bending between three connected beads, described by the U_{angle}:

$$U_\theta = \frac{1}{2} k_\theta (\cos\theta_{\text{beads}} - \cos\theta_0)^2 \tag{3.23}$$

where k_θ defines the bending rigidity and θ_0 is the equilibrium bonding angle formed by three connected beads. The value of θ_0 is set to π to ensure that the alkane-based polymer chains maintain their linear structure and do not coil back to the core particle. We summarize the interaction parameters in Table 3.4:

Table 3.4: interaction parameter

interaction type	parameter 1	parameter 2	parameter 3
$U_{\text{LJ,C1-C1}}$	$\sigma_{\text{C1-C1}} = 0.47$	$\epsilon_{\text{C1-C1}} = 0.81$	-
$U_{\text{SLJ,core-core}}$	$\sigma = 0.47$	$\epsilon = 0.81$	$\Delta = 2.35$
$U_{\text{SLJ,core-C1}}$	$\sigma = 0.47$	$\epsilon = 0.81$	$\Delta = 2.35$
$U_{\text{bond,core-C1}}$	$k_{\text{bond}} = 1998$	$r_0 = 2.35$	-
$U_{\text{bond,C1-C1}}$	$k_{\text{bond}} = 1998$	$r_0 = 0.47$	-
$U_{\theta,\text{C1-C1-C1}}$	$k_\theta = 2.98$	$\theta_0 = \pi$	-

Units: nm, kcal/mol, kcal/mol-nm^2, radian

3.5.3 Background on the Forward-Reverse (FR) Method

The FR method relies heavily on the stiff spring approximation [161], which mitigates the challenges associated with directly applying the Jarzynski Equality (JE).

To revisit the idea of the stiff string approximation, we consider a one-dimensional system with a single particle, where the position x serves as the reaction coordinate. The particle resides in a thermal bath at temperature β^{-1} and experiences an external, time-dependent harmonic guiding potential $V(x,t)$ with spring constant k whose equilibrium point moves at a constant velocity v.

$$V(x,t) = \frac{k}{2}(x - vt)^2 \tag{3.24}$$

The effect of this harmonic guiding potential can be illustrated by visualizing the particle attached to a constantly moving spring, which is analogous to the harmonic guiding potential used in the SMD simulation. Furthermore, we assume that the motion of this particle follows overdamped Langevin dynamics:

$$\frac{dx}{dt} = -\frac{k}{\gamma}(x - vt) + \mathcal{V}(t) \tag{3.25}$$

Where γ is the friction constant, and $\mathcal{V}(t)$ represents a delta-correlated white noise with a variance of $2/\beta\gamma$ that satisfies $\langle \mathcal{V}(t) \rangle = 0$ and $\langle \mathcal{V}(t), \mathcal{V}(t') \rangle = \frac{2}{\beta\gamma}\delta(t - t')$. This white noise arises from random fluctuations due to thermal interactions with the surrounding environment. To describe the system evolution, we are interested in the work w done by the external potential on the particle over a time interval, starting from $t = 0$ up to an arbitrary time t. We solve for the work w or the power dw/dt as follows:

$$\frac{dw}{dt} = \frac{dV}{dt} = -kv(x - vt) \tag{3.26}$$

Using Equations 3.25 and 3.26, we ask the following question: *Given that the particle position x and the work done on the particle w are joint random variables evolving in time, what is the corresponding joint probability distribution $p(x, w, t)$ describing this statistical ensemble?* To determine this joint probability distribution, we solve

the corresponding Fokker-Planck equation, which describes the time evolution of the joint probability for our system governed by the overdamped Langevin dynamics. The corresponding Fokker-Planck equation can be derived as follows:

$$\frac{\partial p}{\partial t} = \frac{k}{\gamma}\frac{\partial}{\partial \tilde{x}}(\tilde{x}p) + v\frac{\partial p}{\partial \tilde{x}} + vk\tilde{x}\frac{\partial p}{\partial w} + \frac{1}{\beta\gamma}\frac{\partial^2 p}{\partial \tilde{x}^2} \tag{3.27}$$

We substitute the particle position x with the reference position $\tilde{x} = x - vt$ by setting the equilibrium point of the external potential as the origin. Additionally, our system satisfies the initial condition that the particle starts at position $\tilde{x} = \tilde{x}_i$ without work being performed on it. Thus, the joint distribution of position and work at initial time step, $p(\tilde{x}, w, t = 0|\tilde{x}_i)$, can be expressed as the product of two delta functions, $\delta(\tilde{x} - \tilde{x}_i)$ and $\delta(w)$, which indicates that at t=0, the position $\tilde{x}$ is exactly $\tilde{x}_i$ and no work has been performed yet.

$$p(\tilde{x}, w, t = 0|\tilde{x}_i) = \delta(\tilde{x} - \tilde{x}_i)\delta(w) \tag{3.28}$$

Given this initial condition, we can solve Equation **??** and find that the conditional distribution $p(\tilde{x}, w|\tilde{x}_i)$ is a multivariate Gaussian [161]. Furthermore, we can obtain the distribution of position $\tilde{x}$ and work w by marginalizing p, which also follow a Gaussian form:

$$p(\tilde{x}, t|\tilde{x}_i) = \int dw p(\tilde{x}, w, t|\tilde{x}_i) = \frac{1}{\sqrt{2\pi\sigma_{\tilde{x}}^2}} \exp\left[\frac{(\tilde{x} - \langle\tilde{x}\rangle)}{2\sigma_{\tilde{x}}^2}\right] \tag{3.29}$$

$$p(w, t|\tilde{x}_i) = \int d\tilde{x} p(\tilde{x}, w, t|\tilde{x}_i) = \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left[\frac{(w - \langle w\rangle)^2}{2\sigma_w^2}\right] \tag{3.30}$$

Where $\langle\tilde{x}\rangle$ is the mean reference position at time t, $\langle\tilde{w}\rangle$ is the mean work at time t, σ_x^2 is the variance of reference position at time t, and σ_w^2 is the variance of work at time t. Without losing generality, we solve for the conditional distribution p by perturbing the system with small amounts of potential energy relative to the harmonic

potential. This perturbation does not alter the fact that p has the form of a multivariate Gaussian. Furthermore, the Gaussian work distribution offers advantageous mathematical properties when deriving the PMFs. This approach is known as the stiff-spring approximation.

The FR method [38] combines the conclusions drawn from the stiff-spring approximation with the Crooks fluctuation theorem to calculate PMFs. The Crooks fluctuation theorem [162] states that the ratio of the probability of the work done on the forward path to that on the reverse path is proportional to the total dissipative work done:

$$\frac{p_{\mathrm{F}}(w)}{p_{\mathrm{R}}(-w)} = \exp(w - W^{\text{MS-CG}}) \tag{3.31}$$

Where w represents the work done on the system, $W^{\text{MS-CG}}$ represents PMF, and $(w - W^{\text{MS-CG}})$ corresponds the dissipative work. We plug in the forward and reverse work distribution and derive the following equations for the $W^{\text{MS-CG}}$:

$$W^{\text{MS-CG}} = (\overline{w}_{\mathrm{F}} - \overline{w}_{\mathrm{R}})/2 \tag{3.32}$$

This method accurately calculates the system's PMF using the time-reversal symmetry of non-equilibrium forward and reverse paths. It leverages the existence of a true mean of $W^{\text{MS-CG}}$ around which the Gaussian work distributions of the forward and reverse processes are centered.

Similarly, the conclusion derived from Equation 3.32 can also be obtained using JE through the cumulant expansion. JE is expressed as:

$$W^{\text{MS-CG}} = -\frac{1}{\beta} \ln \left(\langle \exp \left[-\beta w \right] \rangle \right) \tag{3.33}$$

The accuracy of calculating $W^{\text{MS-CG}}$ crucially depends on the precision of the expected value calculated on the right-hand side of Equation 3.33. Through the cumu-

lant expansion, the right-hand side can be expressed as an infinite series in terms of the probability cumulants of w:

$$-\frac{1}{\beta}\ln\left(\langle\exp\left[-\beta w\right]\rangle\right) = \overline{w} - \frac{\beta}{2!}\sigma_w^2 + ... \tag{3.34}$$

In general, this infinite series is intractable without knowledge of the form of the work distribution, due to countless possible integration paths to consider, making JE inapplicable in most scenarios. However, if the work distribution can be approximated by a Gaussian, in which only the first two cumulants (mean and variance) are significant, the higher-order terms can be neglected. The corresponding $W^{\text{MS-CG}}$ can then be expressed as:

$$W^{\text{MS-CG}} = \overline{w}_F - \frac{\beta}{2!}\sigma_{w_F}^2 \tag{3.35}$$

where we obtain the cumulant approximation of $W^{\text{MS-CG}}$ [163]. Similarly, along the reverse path, W^{CG} becomes:

$$W^{\text{MS-CG}} = -\overline{w}_R + \frac{\beta}{2!}\sigma_{w_R}^2 \tag{3.36}$$

By combining Equations 3.35 and 3.36 and assuming that the variances of the forward and reverse work distributions are equal by symmetry, we recover Equation 3.32:

$$2W^{\text{MS-CG}} = \overline{w}_F - \overline{w}_R + \frac{\beta}{2!}(-\sigma_{w_F}^2 + \sigma_{w_R}^2) \approx \overline{w}_F - \overline{w}_R \tag{3.37}$$

It is worth noting that this FR method is robust in the sense that (1) it does not require a large number of samples, and (2) a simple linear integration path can be chosen to accurately sample $W^{\text{MS-CG}}$.

To better illustrate the FR method, we performed the SMD simulations with the FR method with a pulling velocity of 10^{-6} nm/fs and compared the PMFs against the ones directly calculated from the forward SMD simulations. Figure **??** top figures

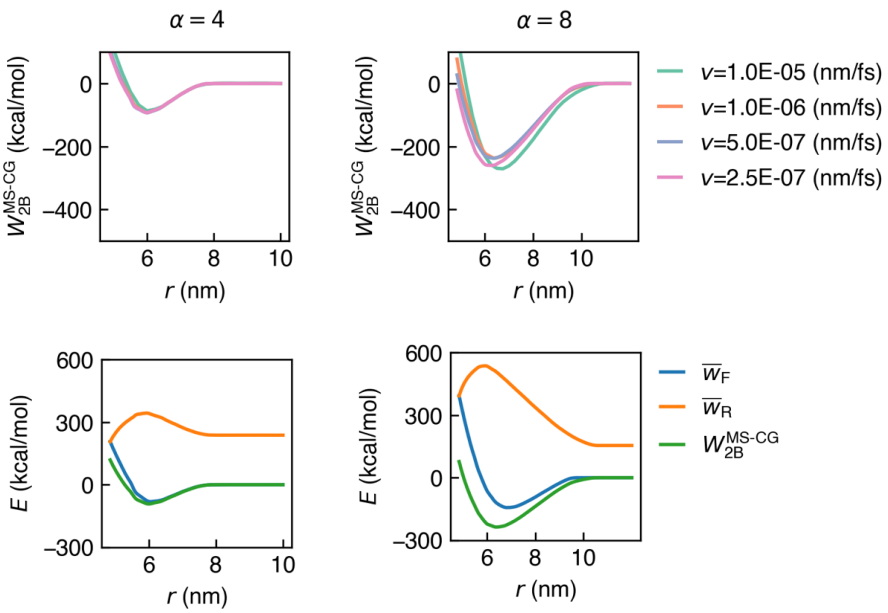

Figure 3.13: **Forward-reverse (FR) method demonstration and SMD velocity convergence test for PGNs with $\alpha = 4$ and 8.** To ensure the accuracy of the FR method, a sufficiently slow SMD pulling velocity must be selected. **Top**: Various pulling velocities are tested for PGNs with $\alpha = 4$ and 8. **Bottom**: PMF calculated using the FR method with the selected pulling velocity.

illustrate SMD pulling velocity convergence tests for PGNs with $\alpha = 4$ and 8. In Figure **??** bottom figures, the blue curve represents the mean forward work, which serves as a first-order approximation to the PMFs of the PGNs. The averaged exponentiated work in JE is estimated by averaging the mean forward work (blue) and the negative mean reverse work (orange) using Equation 3.32. We observe a noticeable discrepancy between the first-order approximation and the results obtained from the FR method at shorter inter-PGN distances, which intensifies with increasing polymer length. Theoretically, the first-order approximation of the PMF with an infinitely slow pulling velocity would yield identical outcomes to the FR method. However, such sampling conditions are computationally impractical, making the FR method a more feasible and efficient approach for PMF sampling.

3.5.4 ChIMES Fitting using Truncated Singular Value Decomposition (TSVD)

When solving the $Ac = b$ linear system, b corresponds to the ground truth values for energy, force, or stress; in this work, only the energy, i.e., PMF, is included. ChIMES by default solves for c through the truncated singular value decomposition (TSVD), with an optimization factor κ that determines the threshold for dropping small singular values to achieve denoising effects and numerical stability when solving the linear system. Using SVD, A can be decomposed into $U\Sigma V^T$, where U is the left singular vector matrix, V is the right singular vector matrix, and Σ is the singular value matrix. Σ is a diagonal matrix with entries in the descending order from $\sigma_{\max}$ to $\sigma_{\min}$, where $\sigma_{\max}$ and $\sigma_{\min}$ are the maximum and minimum singular value of A, respectively. Any singular values below $\kappa \times \sigma_{max}$ are set to zero. TSVD solves for $\tilde{A}c = b$ where $\tilde{A} = U\tilde{\Sigma}V^T$ with the solution of c as $V\tilde{\Sigma}^{-1}U^T b$. $\tilde{\Sigma}$ is a truncated singular value matrix, making $\tilde{A}$ a low rank approximation of the original matrix A.

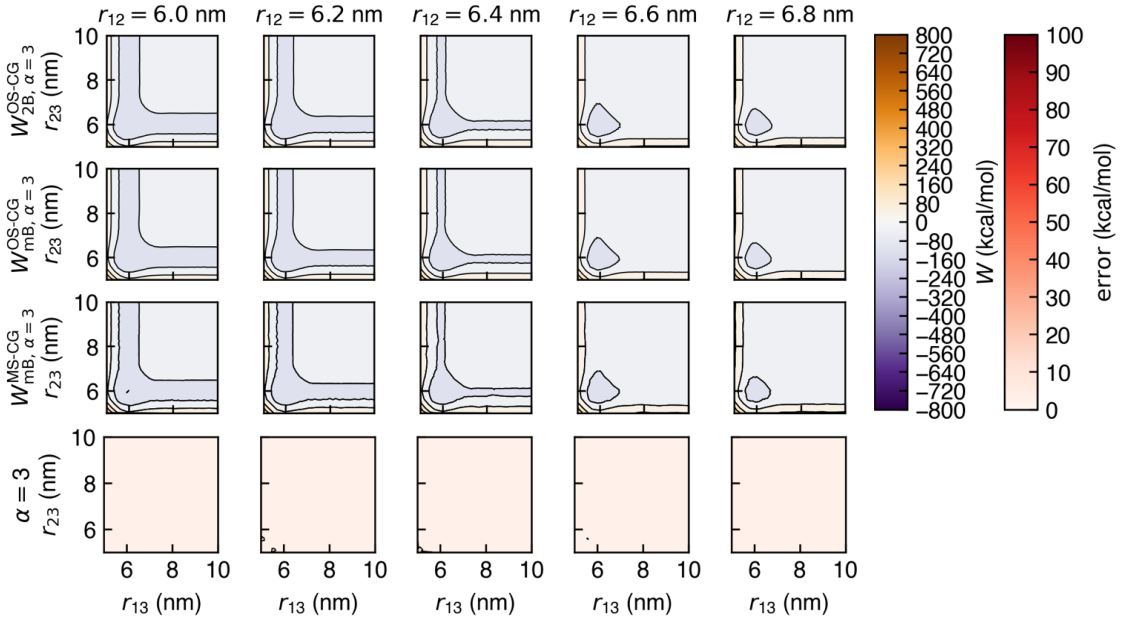

Figure 3.14: **Illustration of two- and many-body PMFs for PGNs with $\alpha = 3$.** The spacing between contour lines for the PMFs is 40 kcal/mol, and 10 kcal/mol for the error map.

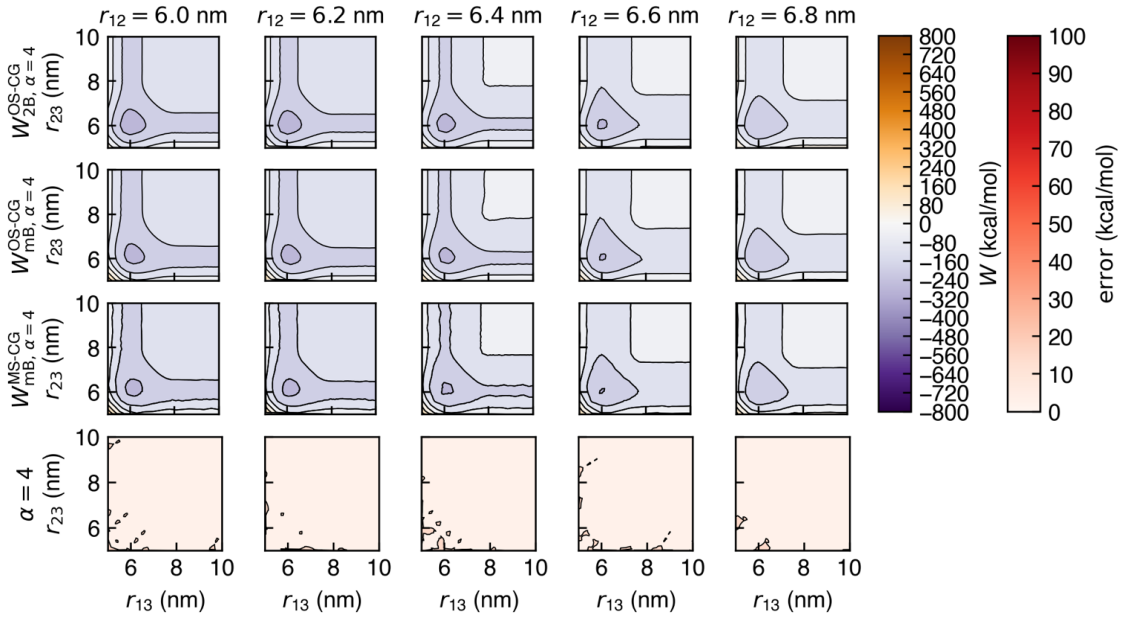

Figure 3.15: **Illustration of two- and many-body PMFs for PGNs with $\alpha = 4$.** The spacing between contour lines for the PMFs is 40 kcal/mol, and 10 kcal/mol for the error map.

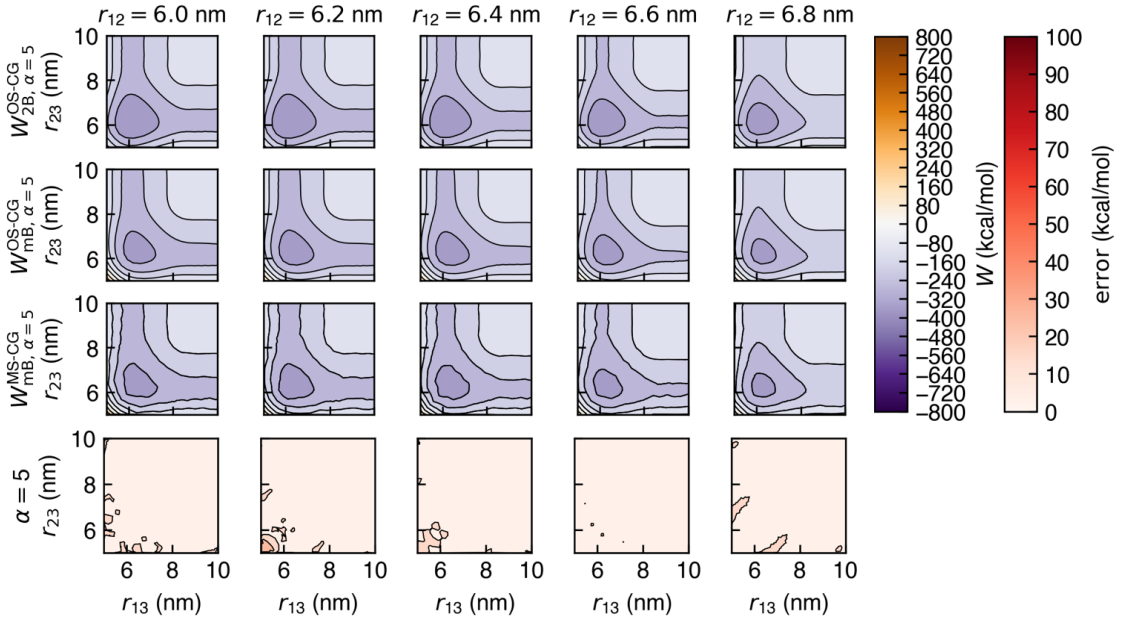

Figure 3.16: **Illustration of two- and many-body PMFs for PGNs with $\alpha = 5$.** The spacing between contour lines for the PMFs is 40 kcal/mol, and 10 kcal/mol for the error map.

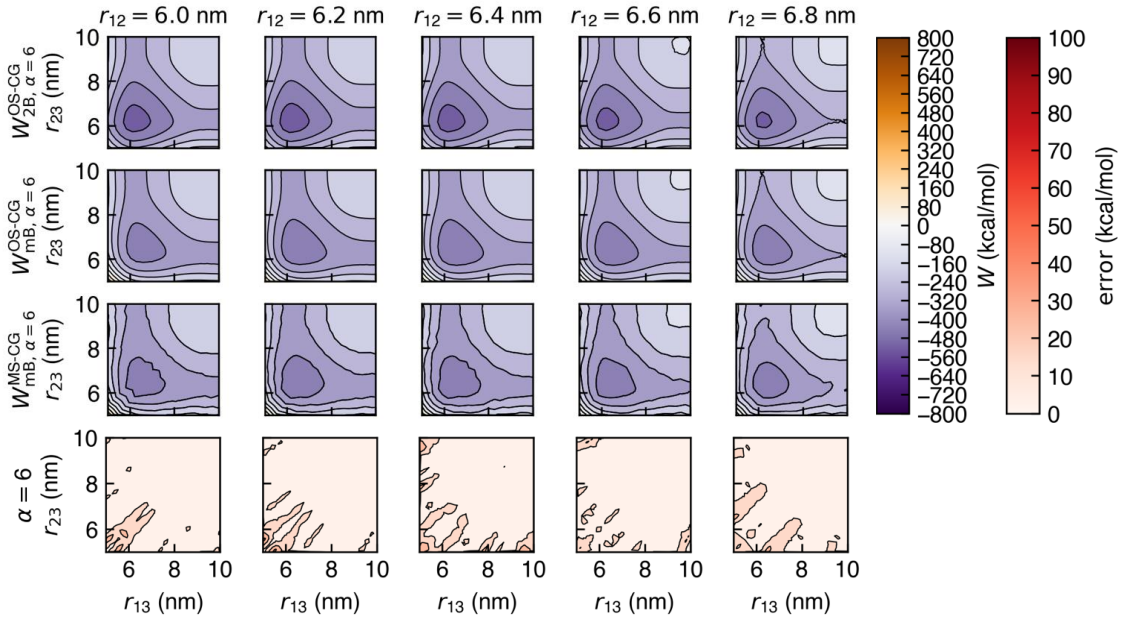

Figure 3.17: **Illustration of two- and many-body PMFs for PGNs with $\alpha = 6$.** The spacing between contour lines for the PMFs is 40 kcal/mol, and 10 kcal/mol for the error map.

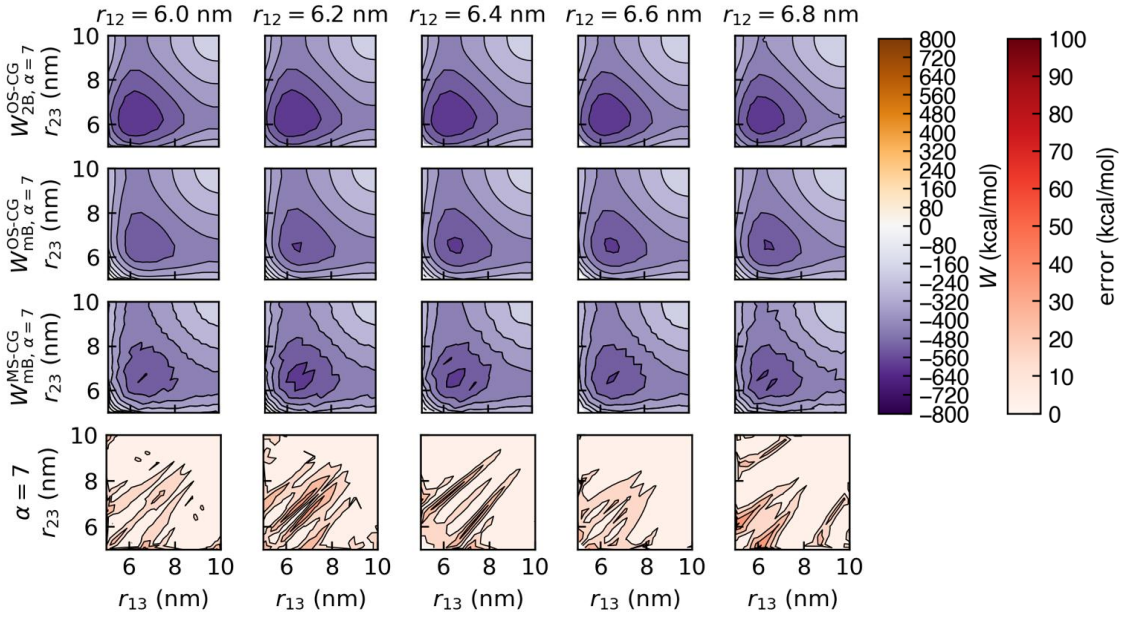

Figure 3.18: **Illustration of two- and many-body PMFs for PGNs with $\alpha = 7$.** The spacing between contour lines for the PMFs is 40 kcal/mol, and 10 kcal/mol for the error map.

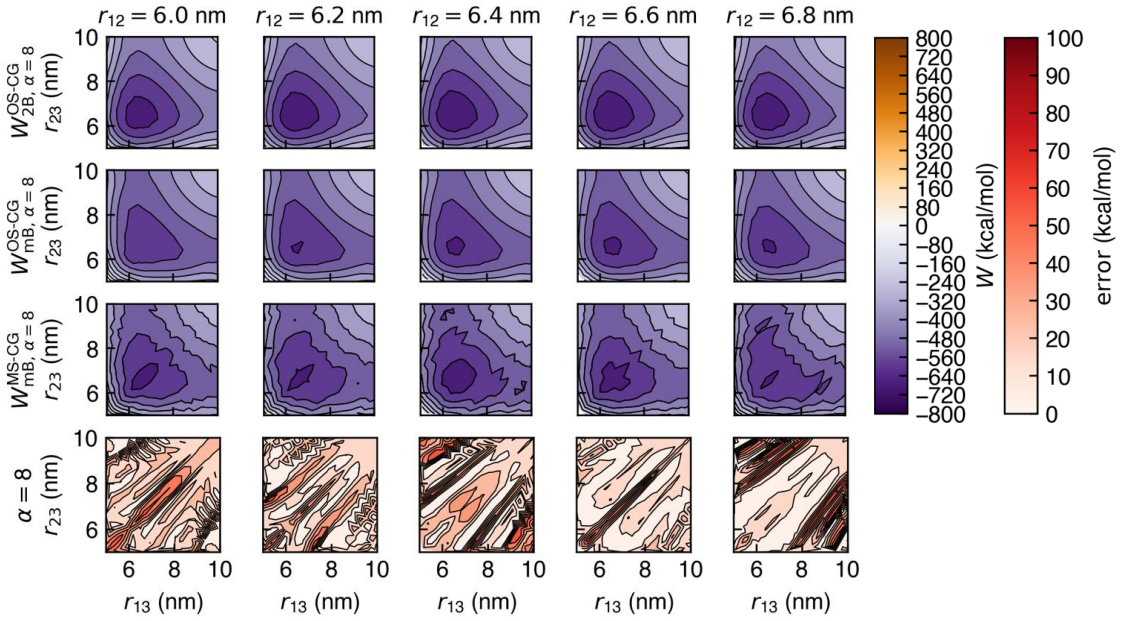

Figure 3.19: **Illustration of two- and many-body PMFs for PGNs with $\alpha = 8$.** The spacing between contour lines for the PMFs is 40 kcal/mol, and 10 kcal/mol for the error map.

3.5.5 One-Site (OS) ChIMES Model Prediction

In Figure 3.14-3.19, we provide a more comprehensive illustration of ChIMES-predicted two- and many-body PMFs (i.e., $W_{\mathrm{2B}}^{\mathrm{OS\text{-}CG}}$ and $W_{\mathrm{mB}}^{\mathrm{OS\text{-}CG}}$), compared with the ground truth many-body MS-CG PMFs (i.e., $W_{\mathrm{mB}}^{\mathrm{MS\text{-}CG}}$), and the absolute error in the ChIMES prediction (red heatmaps) for each PGN studied in this work.

3.5.6 Condensed Phase Simulation

Here, we provide the results that support our observation in the main text Section 3.3.2.

To further verify our observation of the phase transition, we employ the Steinhardt order parameter, $\overline{q}_l$, with l=6, for probing FCC, BCC, and liquid structures [62], as shown in Figure 3.20. For all five replicas at 10 atm, three remain at $\overline{q}_6 \approx 0.57$, corresponding to stable FCC structures, while the remaining two transition towards

$\overline{q}_6 \approx 0.38$, which indicates existence of liquid structure, since $\overline{q}_6$ for liquid is close to 0. This calculation aligns with our previous observation where two out of the five replicas transform from FCC to an FCC/BCC/liquid mixture. At 100 atm, $\overline{q}_6$ for all replicas decreases to $\overline{q}_6 \approx 0.35$, which is consistent with the observation of FCC/BCC/liquid composition upon equilibrium. At 1000 atm, all replicas remain $\overline{q}_6 \approx 0.57$ throughout the simulation, which is consistent with the observation that all the systems are kinetically trapped in the initial FCC structures at 1000 atm.

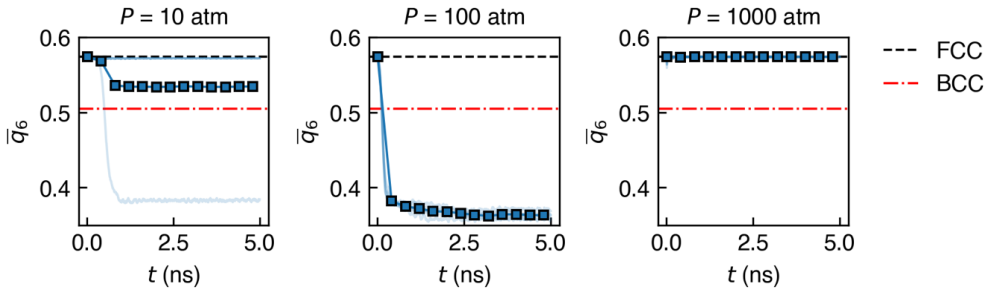

Figure 3.20: **Steinhardt order parameter $\overline{q}_6$ for simulations of PGNs described by the OS-CG model initialized on an FCC lattice with $\alpha = 8$.** Analysis is performed for the same set of simulations demonstrated in the top right of Figures 8 and 9 in the main text. Blue solid lines and square markers give results for five replica simulations and the average of each simulation, respectively. The black dashed lines indicate $\overline{q}_6$ for FCC crystals. The red dash-dotted lines indicate $\overline{q}_6$ for BCC crystals.

In figure 3.21, we illustrate three snapshots during the expanding box simulations using OS-CG models for PGNs with $\alpha = 4$ and 8, respectively. The particles are classified and colored as either FCC, BCC, HCP crystal, or others using the polyhedral template matching (PTM) method [65] implemented in Ovito [91]. We use the root-mean-square deviation cutoff of 0.1 as a threshold to compare the particle's local environment with the ideal structural template.

To investigate the ideal crystal structure assembled by our PGNs across low to high pressure, we estimate the phase diagram in Figure 3.22. We approximate the phase diagram by arranging the PGNs at ideal lattice sites of FCC and BCC at

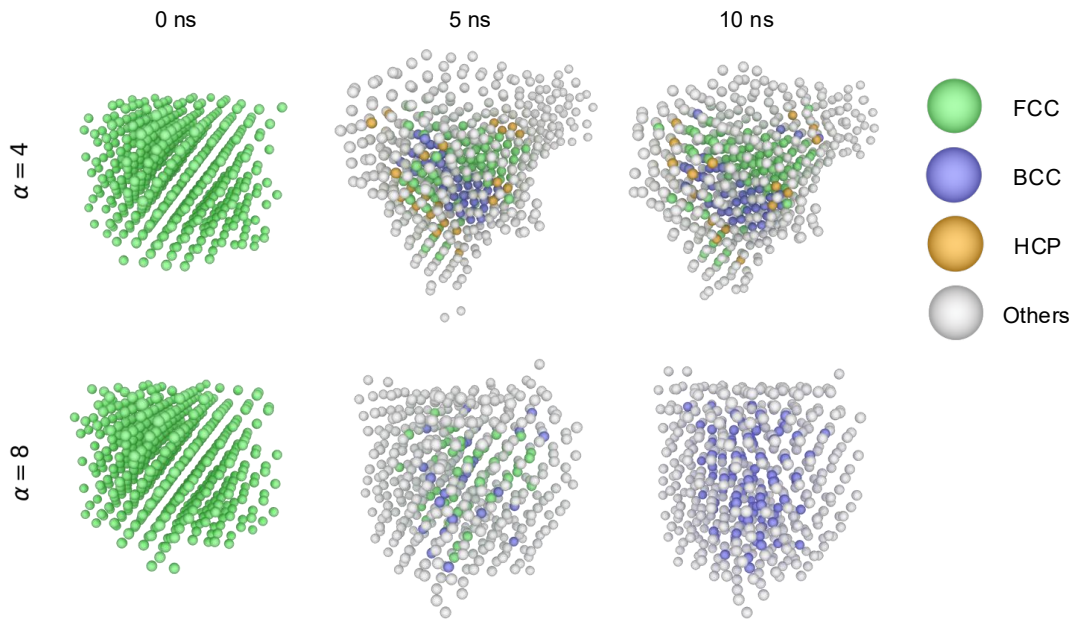

Figure 3.21: **Snapshots of expanding box simulation for PGNs with $\alpha = 4$ and 8.**

different lattice constant a_0 and varying the $\alpha = 3$ to 8. Then, we perform single-point energy calculations on the selected state points. Specifically, we calculate the corresponding potential energy U using the OS-CG model $W^{\text{OS-CG}}$ and adding the product of pressure and volume PV. For the P term, we only evaluate the configuration contribution. The resulting $U + PV$ corresponds to the enthalpy at $T = 0$ K for the OS-CG model. Finally, we compare whether FCC or BCC possesses a lower enthalpy to determine the phase diagram. To smooth the resulting phase diagram, we employ the Clough-Tocher two-dimensional interpolator[164, 165, 166] implemented in the Python scipy package to interpolate the enthalpy in two dimensions using a piecewise cubic, C1-smooth, curvature-minimizing interpolator. The left figure is the phase diagram plotted as a function of α and the nearest neighbor distance between particles, r_{near}. The right figure is the phase diagram plotted as a function of α and P.

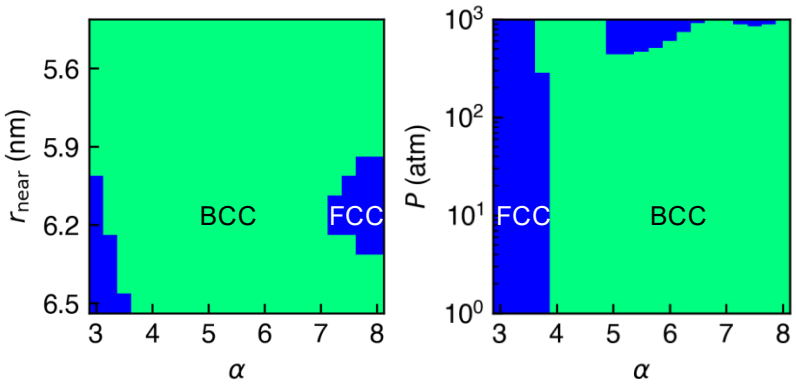

Figure 3.22: **The approximated phase diagram of PGNs with α = 3 - 8 using X-ChIMES model.**

To calculate the PGNs' capping layer thickness λ, we generated the normalized partial number integral $\overline{N}(r)$ between core nanoparticles and polymer beads for all of the PGNs studied in this work. λ is obtained by inspecting where the $\overline{N}(r)$ firstly reaches one, indicating the outermost distance between core particles and polymer beads. The result is shown in Figure 3.23.

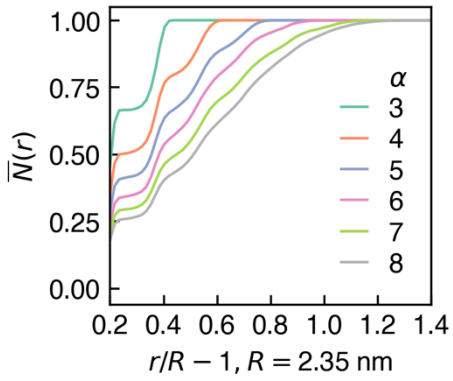

Figure 3.23: **The normalized partial number integral $\overline{N}(r)$ of isolated PGNs for calculating ligand capping layer thickness λ.** We illustrate the $\overline{N}(r)$ between core nanoparticles and the polymer beads, emphasized at the distance between core nanoparticles and polymer beads r where $\overline{N}(r)$ reaches 1.

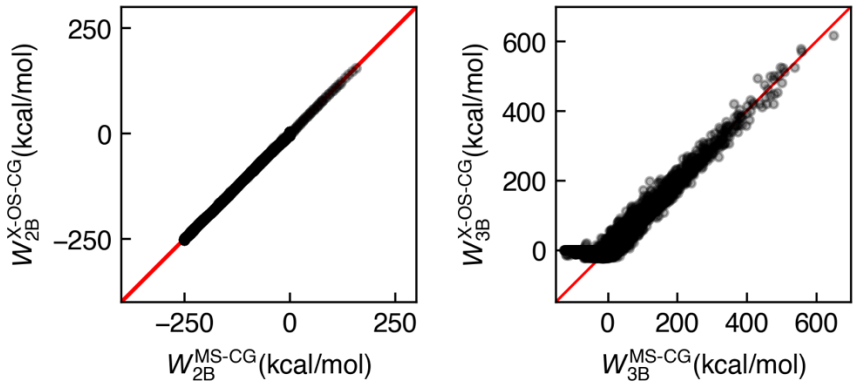

Figure 3.24: **Performance of the two- and three-body extended ChIMES (X-ChIMES) model fits based on training data**. **Left**: two-body X-ChIMES parity plot with Pearson correlation value of $R^2 \sim 0.9999$. **Right**: three-body X-ChIMES parity plot with Pearson correlation value of $R^2 \sim 0.9343$.

3.5.7 One-Site (OS) Extended ChIMES (X-ChIMES) Model

Here, we demonstrate the performance of two- and many-body X-ChIMES model training results by illustrating the parity plots, comparing the X-ChIMES model prediction denoted as $W_{2B}^{\text{X-OS-CG}}$ and $W_{3B}^{\text{X-OS-CG}}$ and the training data $W_{2B}^{\text{MS-CG}}$ and $W_{3B}^{\text{MS-CG}}$ for $\alpha = 3, 4, 6$ and 8 in Figure 3.24. We note a cluster of data points aggregates at $W_{3B}^{\text{X-OS-CG}} = 0$ in the three-body parity plot, attributed to that X-ChIMES model overestimates PMFs at intermediate inter-PGN distances of $r_{23}, r_{13} \sim$ 7 - 9 nm.

To better understand the stability of FCC and BCC crystals assembled by PGNs, we conduct single-point energy calculations of crystals by placing the PGNs on perfect lattice sites using the many-body X-ChIMES model. We scan through three different lattice constants a_0 for FCC and BCC lattices with $\alpha = 3$ to 8, and the results are demonstrated in Figure 3.25. Furthermore, we use these results to approximate the phase diagram by comparing crystal energy at the same first nearest distance between PGNs (i.e., r_{near}) and the same α, shown in Figure 3.22. We note

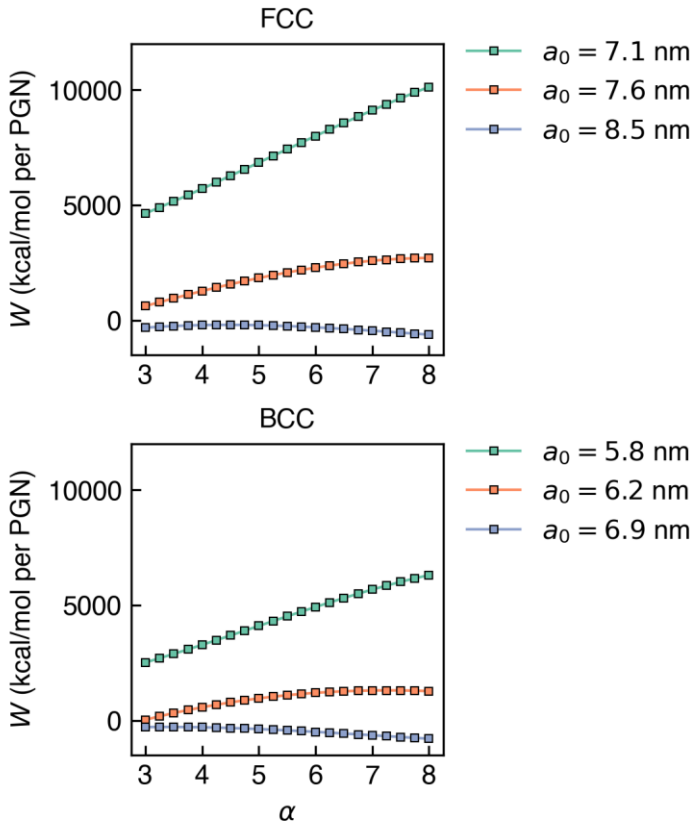

Figure 3.25: **Single point energy calculation of FCC and BCC crystals using X-ChIMES models.**

that the equilibrium distances of r_{near} are near 6 nm across all the α in this work.

3.5.8 Application of the Forward-Reverse (FR) Method in Steered Molecular Dynamics (SMD) Simulations

As an initial step, we take our fine-grained models and calculate the PMFs as functions of CG PGN positions. Please refer to the Github repository [167] for our SMD modules implemented under the HOOMD-Blue Python Package. In the repository, we provide a simple example of calculating the two- and three-body PMF between LJ particles at a low temperature. The keys to successful calculation of PMFs with the FR method via SMD simulations are as follows: (1) To ensure the

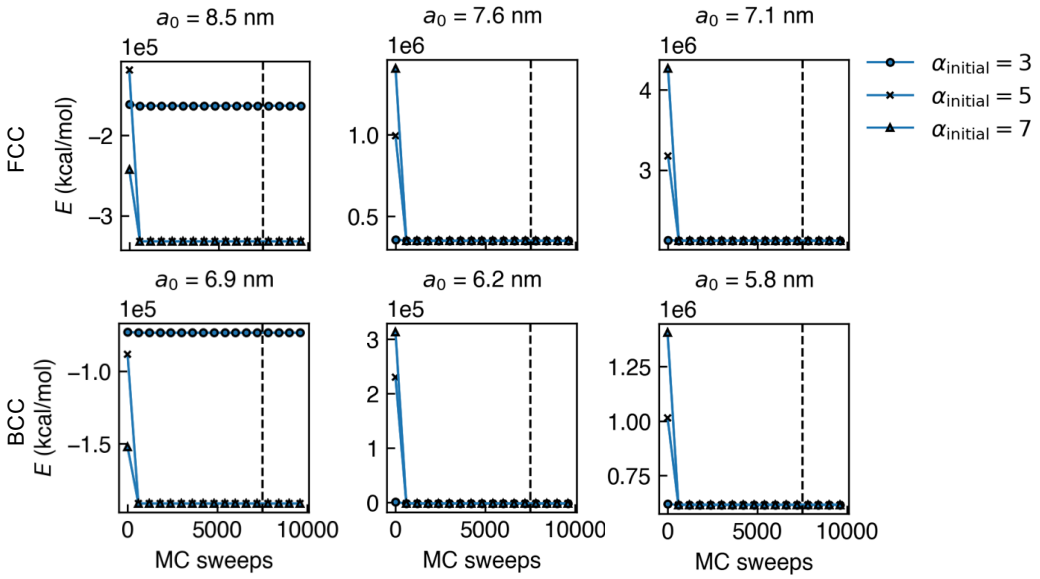

Figure 3.26: **Alchemical Monte-Carlo (Alch-MC) simulation trajectories.** We illustrate the energy evolution of Alch-MC simulation trajectories using many-body X-OS-CG model $W_{\mathrm{mB}}^{\text{X-OS-CG}}$ by plotting the total energy E evolution over Alch-MC sweeps.

stiff-string approximation holds, a strong enough spring constant k should be chosen. The rule to determine the value of k is to estimate the highest expected PMF barrier $W_{\max}$ that the system needs to overcome during SMD pullings [38] and set k as:

$$k \simeq \max\left(\frac{2c}{\delta x}, \frac{2W_{\max}}{\delta x}\right) \tag{3.38}$$

Where c is a constant ≥ 1 and δx is the largest expected deviation between the instantaneous spring length and the actual reaction coordinate of the system, i.e., the interparticle distance between coarse-grained PGNs. (2) A slow enough SMD pulling velocity should be used to guarantee that the work distribution remains Gaussian throughout the entire SMD simulation. This can be done by conducting umbrella sampling simulations to calculate the PMF profile in the cheapest manner, e.g., two-body PMFs, and use them as a benchmark. (3). During the SMD simulation with the FR method, sufficient time should be allowed between the forward and reverse pullings to equilibrate the system at the innermost distance, and the equilibrium time

varies with respect to the system. (4) In our SMD module, users need to provide the *group_index*. There are two indices for two-body SMD (see Pseudocode III.1) and three indices for three-body SMD. These indices determine how the spring(s) is connected within the system, i.e., how the system is coarse-grained. In our paper, we use the core particle position as the CG position. Thus, we only need to provide the core particle index for each PGN, which means that the spring is attached to the core particles only. Alternatively, users can input a group of particle indexes that represent a PGN, and the spring is connected to the center-of-mass position, which is the center-of-mass of the group. (5) A few SMD simulations should be conducted to calculate the averaged forward work and reverse work, which contribute to the calculation of PMFs. For simplicity, we showcase the PMF calculation of LJ particles in the repository. Due to the fact that the LJ particle system is extremely simple, one set of SMD simulations is a sufficient approximation.

Pseudocode III.1: Two body SMD example

```
import hoomd
from src import smd
...
forces = []
smd_force = smd.SMD_constV_couple(
    sim=simulation,
    group1_idx=[0],
    group2_idx=[1],
    dt=timestep,
    pulling_velocity=pulling_velocity,
    spring_constant=spring_constant,
    pulling_direction=pulling_direction
)
forces.append(smd_force)
simulation.operation.integrator.force = forces
```

Special treatment should be implemented when conducting three-body SMD simulations. In Pseudocode III.2, we provide an example usage of our three-body SMD module. Intuitively, the three-body SMD simulation is performed on a system that contains three groups of particles at the fine-grained resolution. Two of the groups are fixed, and the other one is pulled along a user-defined reaction coordinate toward point c, denoted as r_c. In our paper, as illustrated in Figure 4, r_c is always on

the axis connecting the system represented with *group1_index* and *group2_index*. The three-body SMD module is constructed such that the particle group labeled by *group3_index* is pulled toward r_c. r_c can be user-defined and by default, is the midpoint between the center-of-mass position of the two groups represented with *group1_index* and *group2_index*. During SMD pulling, the group being pulled experiences a static field exerted by the other two fixed groups. We note that the forward and reverse work are functions of the distance between r_c and the center-of-mass of the particles represented by *group3_index*. The resulting work contains two-body effects and must be removed when generating pure three-body PMFs. Additionally, when conducting three-body SMD simulation, the center-of-mass position of the groups represented by *group1_index* and *group2_index* should be fixed using a strong spring with spring constant of k_{fix}, while the other spring applied on the group labeled by *group3_index* have a spring constant of k_{pull}. The three-body SMD module requires users to input two spring constants, k_{fix} and k_{pull}.

Pseudocode III.2: Three body SMD example

```
import hoomd
from src import smd
...
forces = []
smd_force = smd.SMD_constV_threebody(
    sim=simulation,
    group1_idx=[0],
    group2_idx=[1],
    group3_idx=[2],
    dt=timestep,
    pulling_velocity=pulling_velocity,
    spring_constant_fix=k_fix,
    spring_constant_pull=k_pull,
    r_c=[0., 0., 0.],
    r_fix=[[r1, r2]],
    pulling_direction=pulling_direction
)
forces.append(smd_force)
simulation.operation.integrator.force = forces
```

Here, we describe the output log file of two- and three-body SMD simulations. The log file is formulated in the form of a text file, with each column containing different outputs. An example of a two-body SMD log file is shown below:

Pseudocode III.3: Example of two-body SMD log file

```
t      force_x    force_y    force_z    total_force    spring_length    d          work
```

```
0      0.00050    0.00000    0.00000    0.00050        2.99999          3.00000     5.00000e-09
1000   4.13573   -2.13900   -2.14892    4.13573        2.98999          3.07330    -0.00556
2000  -2.28561    0.67227    0.67539   -2.28561        2.97999          2.93434    -0.00713
3000  -2.58838    0.48048    0.48271   -2.58838        2.96999          2.91825    -8.59829e-04
```

Where t is the timestep, the three components of the pulling force are *force_x*, *force_y*, *force_z*, respectively, *total_force* is the projection of the pulling force along the pulling direction, *spring_length* is the pulling spring length, d is the actual distance between particle group 1 and 2, and the *work* is the work done on particle group 2. Below, we showcase the output log file of a three-body SMD simulation. The structure of the log file is similar to that of a two-body SMD simulation, but it contains more information since the system is more complex. For illustration purposes, we split the log file into two sections:

Pseudocode III.4: Example of three-body SMD log file - first part

```
t      r1_deviation      r2_deviation      pull_force_x      pull_force_y      pull_force_z
0      7.04195e-09       7.04195e-09       0.00100           0.00000           0.00000
1000   0.00780           0.00960           2.78402           3.78896          -0.37520
2000   0.01003           0.00940          -0.96305          -1.38644           0.13693
3000   0.00531           0.00568          -1.69557          -1.91916           0.18827
```

Pseudocode III.5: Example of three-body SMD log file - second part

```
total_pull_force      spring_length      d3c         work
0.00100               2.99999            3.00000     1.00000e-08
2.78402               2.98999            3.01807     0.00111
-0.96305              2.97999            2.97039     0.00172
-1.69557              2.96999            2.95310    -0.00148
```

Where t is the timestep, *r1_deviation* and *r2_deviation* are positional deviations of particle groups 1 and 2 from the desired distance *r_fix* that is specified upon module initialization. The three components of the pulling force are *pull_force_x*, *pull_force_y*, *pull_force_z*, respectively. *pull_total_force* is the projection of pulling force along pulling direction, *spring_length* is the pulling spring length, *d3c* is the actual distance between particle group 3 and r_c, and *work* is the work done on particle group 3.

3.5.9 ChIMES Model Fitting

After we parse the SMD output log file into the xyzf file format that ChIMES uses to formulate the training data, we generate ChIMES models for each PGN configuration using the general ChIMES workflow outlines in the ChIMES LSQ documentation (`https://chimes-lsq.readthedocs.io/en/latest/index.html`). Unlike the energy-, force-, and stress-matching approach commonly used in ChIMES model fitting, our models are generated through learning PMF training data. Therefore, we implemented different bash commands in the job submission script to parse out PMF data during the model training process. After generating the A matrix and b vector, we made the following modifications to the submission script to prepare energy-only data for ChIMES fitting, shown in III.6. Line 1 first combines *A.txt* and *b-labeled.txt* line-by-line and then uses the *awk* command to search for lines containing "+1" in the combined file. In the ChIMES convention, "+1" labels correspond to energy data. Once these lines are identified, the last field ($NF) and the second-to-last field ($(NF-1)) are removed, leaving only the corresponding lines in *A.txt*, which are written to a new file called *A_ener.txt*. Line 3 performs a similar operation: it searches for lines in *b-labeled.txt* where the first column is "+1" ($1=="+1"), corresponding to the lines that contain energy data. The energy values, stored in the second column ($2), are written to a new file named *b_ener.txt*. Once *A_ener.txt* and *b_ener.txt* are generated, they can be used to solve for model fitting parameter file *params.txt* by specifying the –*A* and –*B* flags, as shown in Line 6. Finally, Line 9 generates *compare.txt*, which evaluates model performance by inspecting the target energy recovery.

Pseudocode III.6: ChIMES energy fitting submission script

```
1 # Generate files needed to fit to energy only
2 paste A.txt b-labeled.txt | awk '/+1/{$NF=""; $(NF-1)="";print}' > A_ener.txt
3 awk '{if($1=="+1"){print $2}}' b-labeled.txt > b_ener.txt
```

```
4
5# solve for A matrix
6python3 /path/to/chimes_lsq-myfork/build/chimes_lsq.py --A A_ener.txt --b b_ener.txt >
      params.txt
7
8# combined data for model performance evaluation via parity plot
9paste b_ener.txt force.txt > compare.txt
```

3.5.10 OS ChIMES Simulations

After we fit the traditional ChIMES model (i.e., OS-CG model) and obtain the *param.txt* file mentioned in Section 3.5.9, which contains ChIMES potential parameters, users have three choices to conduct simulations. The first two options are to use the LAMMPS simulation package [168] or the ChIMSE MD module, where a small MD simulation can be conducted for validation purposes. The usage of which is well-documented in the ChIMES Calculator package [156]. The third option is to use the HOOMD-Blue package [88], supplemented with the Python module we developed (link), which incorporates the ChIMES Calculator that interfaces with HOOMD-Blue to evaluate energy, force, and virial. A pseudocode that uses our Python module with HOOMD-Blue is illustrated below:

Pseudocode III.7: HOOMD-ChIMES module

```
1import hoomd
2from src.hoomd_chimes_addons import ChIMES_manybody
3...
4forces = []
5chimes = ChIMES_manybody(
6     sim=simulation,
7     chimes_api_path="/path/to/chimes_calculator/serial_interface/api",
8     chimes_wrapper_path="/path/to/chimes_calculator/build",
9     param_file_path="./param.txt",
10     N_particles=N_particles
11)
12forces.append(chimes)
13simulation.operation.integrator.force = forces
```

CHAPTER IV

Exploratory Digital Alchemy for Colloidal Crystal Discovery

This chapter is reproduced from a paper in preparation by Shih-Kuang (Alex) Lee, Sun-Ting Tsai, and Sharon C. Glotzer †. († indicates the corresponding author)

4.1 Introduction

Colloidal self-assembly – the spontaneous organization of particles at the micrometer and nanometer scales – underpins advances in materials science [43], nanotechnology [52], consumer and beauty [169, 170], and drug delivery [95]. The ability to control this process enables the design of materials and products with tailored structures and properties. Digital Alchemy (DA) is an inverse design framework rooted in statistical mechanics that optimizes colloidal particle attributes—such as shape and interparticle interactions—to achieve desired crystal structures [30, 31, 32]. DA performs simulations in a generalized thermodynamic ensemble, treating particle attributes as thermodynamic variables to target specific self-assembled structures [30]. Using methods like Hard Particle Monte Carlo (HPMC) [30, 31] or Molecular Dynamics (MD) [41, 33], DA has successfully linked diverse particle designs, including shapes [171, 31], force field parameters [33, 34], and patchy interactions [32], to targeted crystal structures. However, as an inverse design method, DA's reliance on *a*

priori knowledge of the target structure limits its applicability in exploratory studies, particularly when presented with a novel set of interaction potentials where the target structure is unknown. Additionally, DA only provides nominal optima for particle design based on each simulation. This approach may encounter difficulties due to metastability between particle design parameters and crystal structures: the same set of design parameters can lead to different crystal phases, and conversely, the same crystal phase can be stabilized by different design parameters.

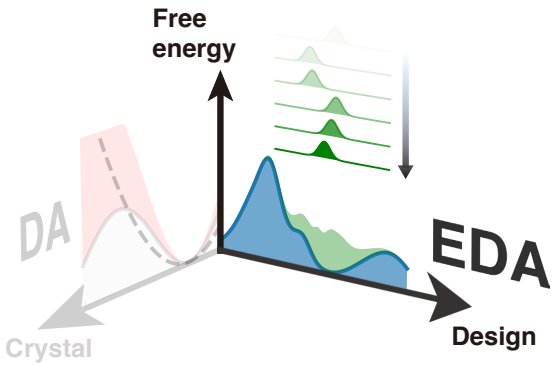

Figure 4.1: **Graphical abstract** The Exploratory Digital Alchemy (EDA) modifies the original Digital Alchemy (DA) framework, which applies the harmonic constraints (dashed line on the left) on crystal structures, causing the elevation of generalized free energy (red region), and forces the system to equilibrate towards the pre-determined state for inverse design. Instead, EDA combines the generalized ensemble of DA with Metadynamics, which iteratively accumulates Gaussian biases (green region) and enables exploration of the correspondence between crystal structure and particle design.

In this paper, we introduce Exploratory Digital Alchemy (EDA), a novel framework that extends the original DA method by combining it with Metadynamics (MetaD) [39, 172]. Unlike DA's inverse approach, which relies on the constraints of the target structure, EDA releases these constraints and enables forward design by evolving colloidal assembly structures and particle designs across stable phases. EDA harnesses the self-exploration mechanics of MetaD to explore stable and metastable phases within a generalized thermodynamic ensemble and identify optimal particle design. That is, EDA allows for a thermodynamically rigorous exploration of particle

design space.

We demonstrate the EDA framework by applying it to explore the design spaces defined by two double-well pairwise potentials: the two-dimensional Lennard- Jones-Gauss Potential (2D-LJGP) and the three-dimensional Oscillating Pair Potential (3D-OPP) systems, both of which have been shown to assemble a wide range of different crystal structures using a grid search method [173, 174]. We demonstrate the simultaneous discovery of crystal structures and their corresponding optimal potential parameter during the evolution of the system by analyzing the free energy surface (FES) of the generalized ensemble. Our method accurately identifies the same crystal structures that have been previously reported. Furthermore, we uncover the metastability among phases in the generalized ensemble, where identical crystal structures can be stabilized by the same optimal potential parameters, or vice versa. By integrating MetaD's self-exploration feature with the generalized ensemble framework of DA, EDA serves as a powerful tool for discovering colloidal assembly structures and optimizing colloidal particle design, significantly enhancing the original DA framework. The trajectory data used to generate the results in this paper can be accessed through the University of Michigan's Deep Blue Data repository via the DOI: `https://doi.org/10.7302/34xw-w788`.

4.2 Methods

4.2.1 Digital Alchemy (DA)

In DA, particle attributes are treated as thermodynamic variables that fluctuate in a generalized ensemble. These additional thermodynamic variables act as additional degrees of freedom and, because changes in these variables correspond to changes in the particles (e.g., their shape or interparticle interactions), they are referred to as alchemical parameters. So-called "alchemical potentials" – the conjugate variables to

the alchemical parameters – are obtained by taking the derivative of the generalized free energy with respect to the alchemical parameter while other thermodynamic variables are held fixed. Additional details may be found in Refs [30, 31, 32, 171, 41, 33, 31, 34].

Consider a system of isotropic point particles interacting with a pair potential $U_\alpha(r)$, in which there is one alchemical parameter α (this can be easily generalized to multiple alchemical parameters). The goal of DA is to find the value of α that minimizes the free energy of the system of point particles in the target crystal structure, in the hope that by using this optimized alchemical parameter, the system will self-assemble into the target structure from an initially disordered state.

To target a desired crystal structure, the biased partition function of the generalized ensemble $\mathcal{Z}$, i.e., biased NVTμ_α ensemble, is defined as [30, 175]

$$\mathcal{Z} = \sum_{X,\alpha} e^{-\beta[U_\alpha(X) - \mu_\alpha N\alpha - \lambda\Lambda]} \tag{4.1}$$

where X denotes a microstate of the system, $\beta = 1/k_BT$ is the inverse of the product of the Boltzmann constant k_B and temperature T, and N is the number of particles. The alchemical potential, μ_α, is the thermodynamic conjugate of α. The Lagrange multiplier $\lambda > 0$ is associated with a constraint Λ that drives the system toward the target crystal structure. In a conventional DA simulation, one applies a spring constant $K = \lambda\beta$ and constraint $\Lambda = \|\mathbf{r} - \mathbf{r}^{(0)}\|^2$, where $\mathbf{r}$ and $\mathbf{r}^{(0)}$ are the particle positions in the system and the reference position of the target crystal, respectively. The $e^{\beta\lambda\Lambda}$ term in Equation 4.1 can then be seen as constraining the system to sample the variants of an Einstein crystal [30].

To sample the partition function $\mathcal{Z}$ defined in Equation 4.1 and find the optimal α, we apply the Metropolis algorithm as follows for determining the acceptance ratio

of a move from alchemical parameter α_1 to α_2 in a Monte Carlo simulation:

$$a_{\alpha_1 \to \alpha_2} = \min\left\{1, \frac{e^{-\beta U_{\alpha_2}(X)}}{e^{-\beta U_{\alpha_1}(X)}}\right\} \tag{4.2}$$

The simulation is performed with fluctuating α at constant $\mu_\alpha = 0$ [30]. We note that while the MD sampling of the partition function $\mathcal{Z}$ is possible, as demonstrated in the Refs [41, 33] where the alchemostat is implemented following the Nosé-Hoover's formulation [152] to control the μ_α, we choose to perform MC sampling with a more straightforward implementation. This choice simplifies the development process of a new method, such as the EDA in this paper.

4.2.2 Exploratory Digital Alchemy (EDA)

Instead of imposing constraints to bias the sampling towards a specific crystal structure, we propose an exploratory version of digital alchemy (EDA) that samples *all microstates* (including all thermodynamically stable and metastable states) defined by $\{X, \alpha\}$ in the generalized ensemble. To perform EDA, we apply an α-dependent bias potential to facilitate and promote the exploration of different crystal structures in the generalized ensemble. Equation 4.1 then becomes

$$\mathcal{Z} = \sum_{X,\alpha} e^{-\beta[U_\alpha(X) + V(\mathbf{s})]} \tag{4.3}$$

where μ_α is again set to 0 and $V(\mathbf{s})$ is the bias potential constructed as a functional of a set of collective variables $\mathbf{s} = \mathbf{s}(X, \alpha)$. In this work we apply a history-dependent bias potential that has been extensively applied in Well-Tempered Metadynamics (MetaD) and is constructed iteratively in the following way [39, 176, 177, 178]

$$\begin{aligned} V(\mathbf{s}, t+1) &= V(\mathbf{s}, t) + W_t \exp\left[-\frac{\|\mathbf{s} - \mathbf{s}_t\|^2}{2\sigma^2}\right] \\ W_t &= \omega e^{-\beta V(\mathbf{s}_t)/(\gamma - 1)} \end{aligned} \tag{4.4}$$

where $V(\mathbf{s}, 0) = 0$ and the update of the bias potential is performed periodically every τ timestep. The $\mathbf{s}_t$ are the set of collective variables calculated from the system evaluated at timestep t, ω is the Gaussian height, σ controls the width of the Gaussian bias, and γ is a user-defined parameter controlling the adaptation of the Gaussian height. Empirically, the DA method sometimes suffers from sensitivity to the initial alchemical parameters it begins with [34]. This is primarily due to high barriers present in alchemical space that isolate local minima of α, resulting in a non-ergodicity problem. Therefore, to maximize the exploration capability of the bias potential, we will set $\mathbf{s} = \alpha$, where α are the alchemical parameters that parameterize the interacting potentials in the system. Equation 4.2 then becomes

$$a_{\alpha_1 \to \alpha_2} = \min\left\{1, e^{\beta(\Delta U + \Delta V)}\right\} \tag{4.5}$$

where $\Delta U = U_{\alpha_2}(X) - U_{\alpha_1}(X)$ and $\Delta V = V(\alpha_2, t+1) - V(\alpha_1, t)$. Following the standard approach of free energy estimation used in metadynamics [177] we can also reweight the sampled collective variable $\xi = \xi(X, \alpha)$ and estimate the unbiased Boltzmann distribution using the time-dependent bias potential $V(\mathbf{s}, t)$ constructed on-the-fly [40]. Note that the ξ for analyzing the statistics can be different from the $\mathbf{s}$ which is used to define the bias potential

$$\langle \xi(X, \alpha) \rangle = \langle \xi(X, \alpha) e^{-\beta[V(\mathbf{s}(t), t) - c(t)]} \rangle_V \tag{4.6}$$

$$e^{-\beta c(t)} \equiv \frac{\int e^{-\frac{\gamma}{\gamma - 1}\beta V(\mathbf{s}, t)} d\mathbf{s}}{\int e^{-\frac{1}{\gamma - 1}\beta V(\mathbf{s}, t)} d\mathbf{s}} \tag{4.7}$$

where $\langle \cdot \rangle_V$ represents the average over the biased ensemble.

With the help of Equation 4.5-4.7, we are able to explore the generalized free energy φ, defined as

$$\varphi = -\frac{1}{\beta} \ln \sum_{X, \alpha} e^{-\beta U_\alpha(X)} \tag{4.8}$$

In practice, we usually consider the free energy surface as a function of the collective variables, sometimes called potential of mean force (PMF), Φ

$$\Phi(\alpha, w) = -\frac{1}{\beta} \ln \sum_{X',\alpha'} e^{-\beta U_{\alpha'}(X')} \delta(w - w(X')) \delta(\alpha - \alpha') \tag{4.9}$$

where $w = w(X)$ is the collective variable, often defined to identify the metastable states sampled during the simulation. In this paper, we will use the term free energy surface (FES) and PMF interchangeably. Additionally, we will apply our method to sample the Helmholtz free energy, F, over all possible α

$$F(\alpha) = -\frac{1}{\beta} \ln \int e^{-\beta \Phi(\alpha, w)} dw \tag{4.10}$$

.

4.2.3 Collective Variable (CV) to Sample Crystal Structures

Because the size, symmetry and lattice constant of the sampled crystal structures tend to constantly fluctuate as a result of α fluctuating, we need a robust collective variable to explore crystal structures that does not rely on a radial cutoff function. Here, we used the Minkowski structure metric [61] (MSM), where the neighboring particles are weighted and determined based on the Voronoi tessellation, as the order parameter for our collective variable. The MSM $w_{\ell,i}$ of particle i is considered an improved version of the Steinhardt OP [62] and is defined as:

$$w_{\ell,i} = \sqrt{\frac{4\pi}{2\ell + 1} \sum_{m=-\ell}^{\ell} w_{\ell m,i} w^*_{\ell m,i}} \tag{4.11}$$

$$w_{\ell m,i} = \frac{1}{\sum_{j=1}^{N_b} A_{ij}} \sum_{j=1}^{N_b} A_{ij} Y_{\ell m}(\boldsymbol{r}_{ij}) \tag{4.12}$$

where A_{ij} is the surface area of the Voronoi cell facet separating the two neighboring particles i and j, $Y_{\ell m}$ is a spherical harmonic, and $\boldsymbol{r}_{ij}$ is the relative position vector

between neighboring particles. Note that for a two-dimensional system, the calculation of MSM simplifies as the polar angle, ϕ_{ij}, alone is sufficient to define the bond orientation:

$$w_{\ell,i} = \left| \frac{1}{\sum_{j=1}^{N_b} A_{ij}} \sum_{j=1}^{N_b} A_{ij} e^{i\ell\phi_{ij}} \right| \tag{4.13}$$

Depending on the crystal structures sampled, other local OPs may be useful as CVs, such as the recently proposed Point Group Order Parameter [179]. Continuous CVs are also possible [180].

4.3 Results and Discussion

Metadynamics parameters				
Pair potential	ω (k_BT)	σ	τ	γ
LJGP, $k_BT = 0.1$	300	0.01	1000	2000
LJGP, $k_BT = 0.3$	100	0.01	1000	2000
LJGP, $k_BT = 0.6$	50	0.01	1000	2000
OPP, $k = 6$	1000	0.05	10000	5000
OPP, $k = 8.5$	3000	0.05	10000	8000

Table 4.1: Metadynamics parameters used in this study.

In this section, we demonstrate the utility of EDA in broadening the search for optimal particle interactions that facilitate self-assembly of many thermodynamically stable and metastable structures in a single simulation. As examples, we apply EDA to systems of particles interacting via two types of double-well potentials: (i) the two-dimensional Lennard-Jones-Gauss Potential (LJGP) [181, 173] and (ii) the three-dimensional Oscillating Pair Potential (OPP) [182, 183]. Systems of particles interacting via these two pair potentials self-assemble into a wide range of different crystal structures and thus are well-suited to evaluate the exploratory power of EDA.

We performed the Monte Carlo (MC) simulations to sample the NVT and NVTμ_α ensembles using the HPMC integrator implemented in HOOMD-blue [88] version 5.0,

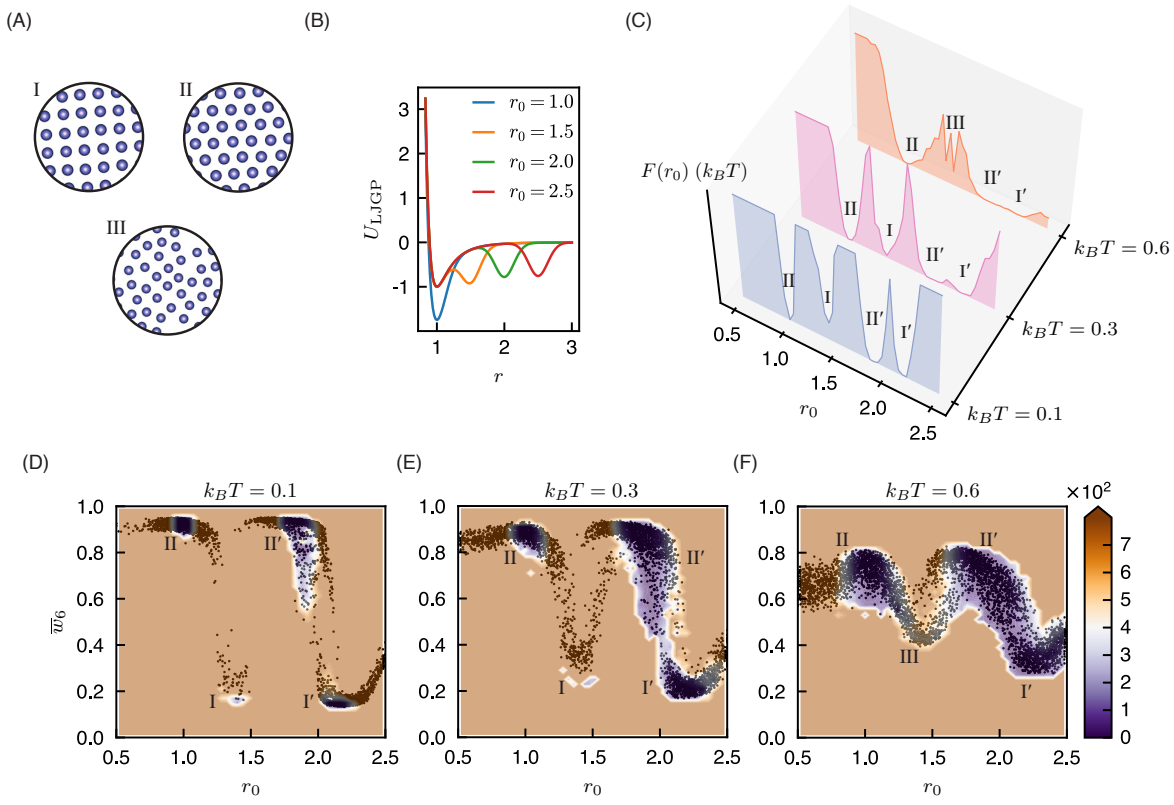

Figure 4.2: **Two-dimensional Lennard-Jones-Gauss Potential (LJGP) system** (A) Thermodynamic phases discovered by the EDA method, including: **I** square lattice, **II** hexagonal lattice, and **III** disordered phase. **I**' and **II**' represent the same crystal structure with different lattice constants (B) The LJGP energy profile at different r_0. (C) Helmholtz free energy surface F (FES) as a function of r_0 and temperature. (D), (E) and (F) The points of collective variables (r_0, $\overline{w}_6$) sampled by the EDA method and the corresponding FES at various temperatures. The minima on the FES are indicated by the corresponding phases in (A). The colorbar indicates the FES value $\Phi(r_0, \overline{w}_6)$ in units of k_BT.

which can be applied to particle systems interacting via a pair potential. For both systems, EDA simulations are performed by initializing a random system with a fixed initial guess of the alchemical parameter, followed by a cooling simulation to a low enough temperature; then, we start the EDA simulations that perturb the alchemical parameter and iteratively accumulate the bias potential as described by Equation 4.4. The alchemical parameter MC move and the bias potential calculation are implemented using the custom updater module of HOOMD-blue. We perform the MSM calculation and other structural analyses, including diffraction pattern

analysis and bond-orientational order calculations, using freud [157]. The simulation snapshots are rendered using Ovito [91]. The trajectory data used to generate the results in this paper can be accessed through the University of Michigan's Deep Blue Data repository via the DOI: `https://doi.org/10.7302/34xw-w788`.

4.3.1 Lennard-Jones Gauss Potential (LJGP)

As our first test case, the Lennard-Jones Gauss Potential (LJGP) is a pairwise double-well potential constructed by adding a negative Gaussian function to the scaled Lennard-Jones potential, with the potential energy formulated as follows [181, 173]

$$U_{\mathrm{LJGP}}(r) = \left[\left(\frac{1}{r}\right)^{12} - 2\left(\frac{1}{r}\right)^{6}\right] - \epsilon_{\mathrm{G}} \exp\left(-\frac{(r-r_0)^2}{2\sigma_{\mathrm{G}}^2}\right) \tag{4.14}$$

Here $\epsilon_G = 0.75$, $\sigma_G^2 = 0.02$, r is the interparticle distance, and r_0 is the center of the Gaussian function and controls the position of the second potential energy well. The Lennard-Jones component in the LJGP has an energy minimum at $r = d = 1$. We performed EDA simulations with 400 particles in 2D interacting via the LJGP with a fixed volume of $60d^2$ at temperatures $k_BT = 0.1$, 0.3 and 0.6. For the purposes of statistical analysis, we ran five independent replicas at each temperature.

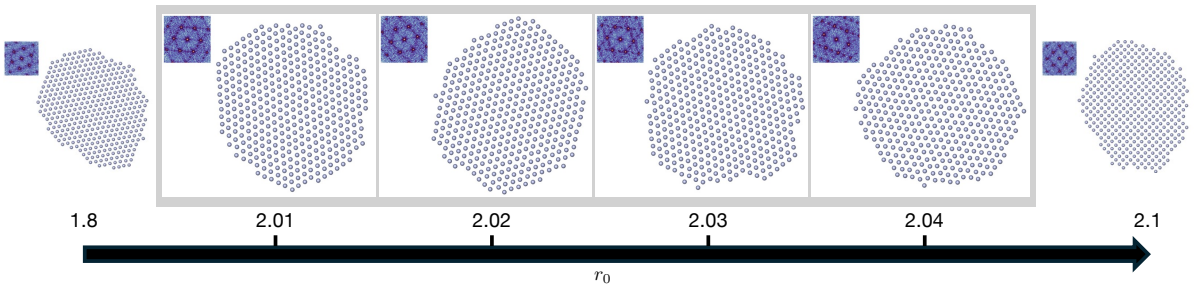

Figure 4.3: **Equilibrium phases obtained from cooling simulations for the Lennard-Jones-Gauss Potential system with $r_0 = 1.8$ to 2.1.** The four snapshots indicated by the grey box in the middle are the r_0 values that we intend to confirm that the rhombic lattice is unstable at a finite temperature.

The EDA simulations were implemented with r_0 as the fluctuating thermodynamic alchemical variable. The bias potential is constructed as a function of r_0 following the approach presented in Section 4.2.2. For each replica, we prepared the simulations by first setting $r_0 = 1.5$ as an initial guess. We then cooled the system from $k_BT = 1.0$ to the target temperatures using 100000 MC sweeps. We calculated $\overline{w}_6$ to identify the thermodynamic phases arising during the simulations and then constructed the unbiased Boltzmann distribution and FES at each target temperature as a function of $(\overline{w}_6, r_0)$, where $\overline{w}_\ell$ is defined as the system average of w_ℓ calculated for each individual particle. The potential energy profiles of different r_0 are also illustrated in Figure 4.2 (B).

Our findings for the LJGP system are reported in Figure 4.2. At $k_BT = 0.1$, we observed two stable, ordered structures: a square lattice (Figure 4.2 (A) I) and a hexagonal lattice (Figure 4.2 (A) II). The square lattice is stable at two different ranges of r_0: $r_0 = 1.3$ to 1.5 and $r_0 = 2.0$ to 2.5. These stability ranges are labeled as I and I$'$ and correspond to square lattices with different lattice constants. The hexagonal lattice is stable within two different ranges r_0: $r_0 = 0.5$ to 1.3 and $r_0 = 1.5$ to 2.0, which are labeled as II and II$'$ and correspond to hexagonal lattices with different lattice constants. Figure 4.2 (C) shows the FES profile $F(r_0)$ by integrating out $\overline{w}_6$. We note that the $F(r_0)$ in Figure 4.2 (C) is smoothed via the Kernel Density Approximation with Gaussian kernels to first smooth the reweighted probability density function and calculate the $F(r_0)$. At $k_BT = 0.1$, the free energy basin of square lattice I is isolated by high barriers, and its free energy minimum is relatively higher than the other phases. As k_BT is increased above 0.3, the square lattice I becomes more stable and the free energy minimum becomes more and the barriers that separate it from the other phase become lower, as shown by the pink FES in

Figure 4.2 (C). Same results for $k_BT = 0.1$ and 0.3 can also be seen from Figure 4.2 (D) and (E) that the square lattice I is less stable than the other phases: the free energy basin of I is substantially higher than the other phases. However, although at $k_BT = 0.6$ there is still a visible free energy minimum, it slightly narrows and splits, and the system becomes disordered (indicated by III), as shown by the orange FES in Figure 4.2 (B) and the upward shift of the energy basin location indicated by III in (F) that used to be I in (E). Additionally, the free energy barrier between the square lattice I$'$ and the hexagonal lattice II$'$ disappears at that temperature, at which we expect that the transitions become easier at higher k_BT. We can also see the disappearance of energy barrier in the 2D plot of the FES (Figure 4.2 (F)) that the barrier between I$'$ and II$'$ nearly vanishes as k_BT is increased to 0.6.

We compare our results at $k_BT = 0.1$ to the zero temperature phase diagram reported by Engel et. al. [173], in which an exhaustive grid search of LJGP potential parameters (ϵ_G, r_0) was performed. The stable crystal structures identified at $k_BT = 0.1$ and shown in Figure 4.2 mostly align with Engel's results, with the exception of the rhombic lattice, which was reported to be stable within a narrow range of $r_0 = 2.0$ to 2.04 but is not observed with our method. We attribute this discrepancy to the fact that our simulations were performed at a finite temperature. We confirm using standard MC simulations, starting from the same initial configuration as in the EDA simulation and cooling the system, starting from $k_BT = 1.0$ down to $k_BT = 0.1$ using 2 million steps. We initiated the simulations from multiple points within the range of $r_0 = 2.0$ to 2.04, and we only observe the hexagonal crystal as the dominant equilibrium phase. We demonstrate the snapshot of those equilibrium phases and their corresponding diffraction pattern in Figure 4.3.

4.3.2 Oscillating Pair Potential (OPP)

In this subsection, we present our findings of the double-well pair potential, OPP, in three-dimensional systems. The Oscillating Pair Potential (OPP) was originally proposed to approximate electronic interaction of intermetallic systems and is based on the form of the Mihalkovič Henley interaction potential [182, 183]

$$U_{\mathrm{OPP}}(r) = \frac{1}{r^{15}} + \frac{\cos\left(k(r-1)+\phi\right)}{r^3} \tag{4.15}$$

where k and ϕ are the wavenumber and phase shift, respectively, of a damped oscillation. The oscillation is truncated and shifted to zero at the maximum following the second attractive well. We show the potential energy profile in Figure 4.4 (A) and (B).

We performed EDA simulations with point particles interacting via the OPP and let the potential parameter ϕ fluctuate while the wave number k is kept fixed. The bias potential is constructed as a function of ϕ, which we use as a collective variable defined within the interval $(0, 2\pi)$. We prepared the simulations at two different wave numbers $k = 6$ and $k = 8.5$ for comparison. For $k = 6$, we performed five independent simulations with 1000 particles each, and for $k = 8.5$, we performed five independent simulations with 810 particles each. Each simulation was initialized with different ϕ values, set as integers from 1 to 5. We also prepared the systems by cooling them from $k_BT = 0.5$ to $k_BT = 0.1$ over 5 million MC sweeps. The sampled points via EDA and the corresponding FES are constructed as a function of $(\overline{w}_6, \phi)$ for $k = 6$ and as a function of $(\overline{w}_5, \phi)$ for $k = 8.5$, and shown in Figure 4.4 (C) and (D).

We first present the results obtained with wave number $k = 6$, shown in Figure 4.4 (A) and (C). Using $\overline{w}_6$ as the collective variable, we identified four distinct

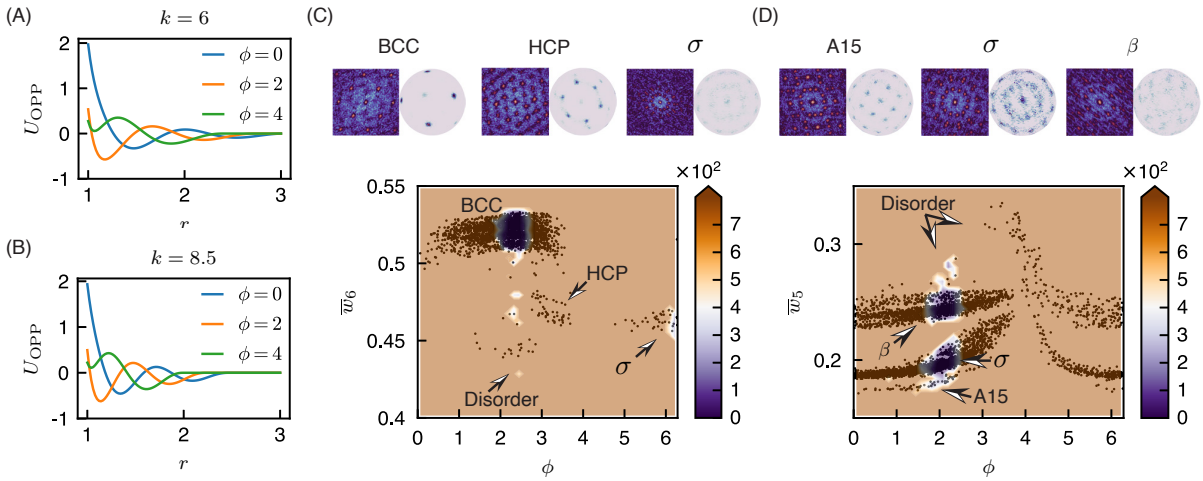

Figure 4.4: **Three-dimensional Oscillating Pair Potential (OPP) system** (A) and (B) The OPP energy profile for $k = 6$ and 8.5 respectively, at different ϕ. (C) and (D) The points of collective variables $(\phi, \overline{w}_l)$ sampled by the EDA method and the corresponding free energy surface (FES), for $k = 6$ and 8.5, respectively. In each subfigure of (C) and (D), the top row plots the diffraction pattern and bond-orientational order diagram for the three crystals discovered, and the FES minima at the bottom row are indicated by the corresponding phases. The colorbars indicate the FES value $\Phi(\phi, \overline{w}_l)$ in units of $k_B T$.

thermodynamic phases: body-centered cubic (BCC), hexagonal close-packed (HCP), a disordered phase, and σ phase, a Frank-Kasper phase with complex unit cell. Of the phases identified, the BCC crystal appears to be the most stable and possesses the lowest free energy located at $\overline{w}_6 \approx 0.52$. The FES indicates that the optimal value of the alchemical parameter r_0 for stabilizing BCC is near 2.3. As r_0 approaches 3, we observed several transitions from BCC to HCP ($\overline{w}_6 \approx 0.475$); however, the missing free energy minimum where HCP is observed shows that HCP is significantly less stable than BCC. The σ phase ($\overline{w}_6 \approx 0.46$), though still less stable than BCC, is stable with a noticeable free energy minimum when r_0 is near 6.0.

Next, we apply EDA to the OPP system with wave number $k = 8.5$. As shown in Figure 4.4 (B) and (D), we observed four phases at the values of the collective variable indicated in parentheses: The A15 phase ($\overline{w}_5 \approx 0.18$), σ phase ($\overline{w}_5 \approx 0.19$), β-manganese crystal ($\overline{w}_5 \approx 0.24$), and a disordered phase. The three crystal struc-

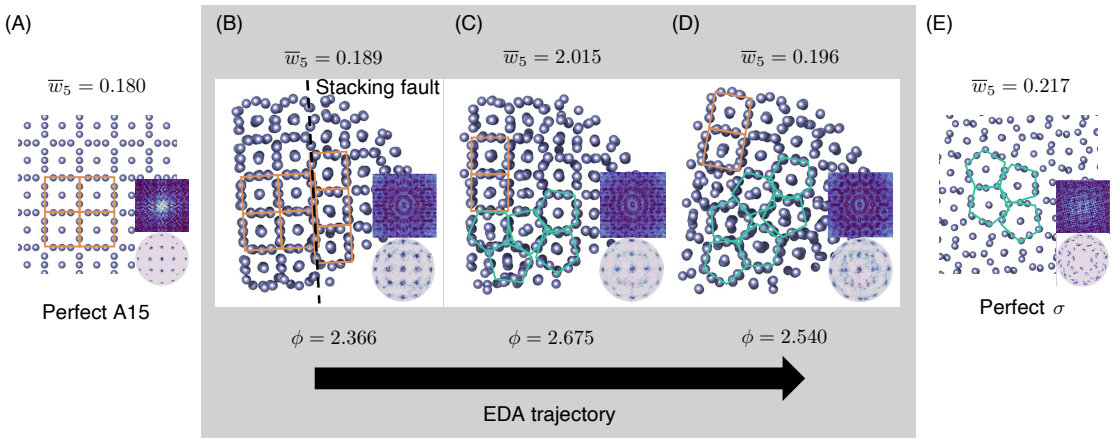

Figure 4.5: **Transition between A15 and σ phases at** $k = 8.5$ (A) The perfect A15 phase. (E) The perfect σ phase. (B) The beginning (C), intermediate, and final stage (D) of the transition. In (B), the stacking fault is indicated by the black dashed line. The special motifs of A15 and the σ phase are highlighted by the brown square and green pentagons, respectively.

tures are stable when the alchemical parameter $\phi \approx 2.0$; each is associated with a corresponding free energy minimum. Due to the lower free energy barrier between A15 and σ phases, we also observed several transitions back and forth between them during the simulations, with the σ phase being more stable. We illustrate the transition between A15 and σ phases in Figure 4.5. The transition begins with the defect formation, such as a stacking fault in the A15 phase, which creates instability, as shown in Figure 4.5 (B). Subsequently, when the ϕ value evolves to approximately 2.5, the incommensurable A15 phase and the interaction introduce local lattice distortion, causing the transformation from the square brown motifs of A15 to the pentagon-like green motifs of σ phase. This leads to the transformation of the A15 phase into the σ phase near the defects, as depicted in Figure 4.5 (C). Finally, in Figure 4.5 (D), the σ phase starts to expand into the surrounding structure. For reference, we also demonstrate the perfect A15 and σ phases in Figure 4.5 (A) and (E). The relatively low free energy barrier between the σ and A15 phases may be due

to their similar local structural motifs. As reported by Dshemuchadse et al. [174], both phases have a high coordination number of 13.5 and contain similar motifs in their unit cell, with one particle surrounded by two alternating hexagons formed by the other 12 particles. On the other hand, we observed no transitions between β-manganese and the other two crystals, which is consistent with the significantly high free energy barrier ($> 700 k_B T$!) around the β-manganese basin. Although sampling transition states between crystal phases and estimating the corresponding free energy barriers are beyond the scope of this paper and are left as future work, we note that the poor sampling of transitions between crystal phases can be improved by constructing the bias potential as a function of the pair $(\overline{w}_5, \phi)$.

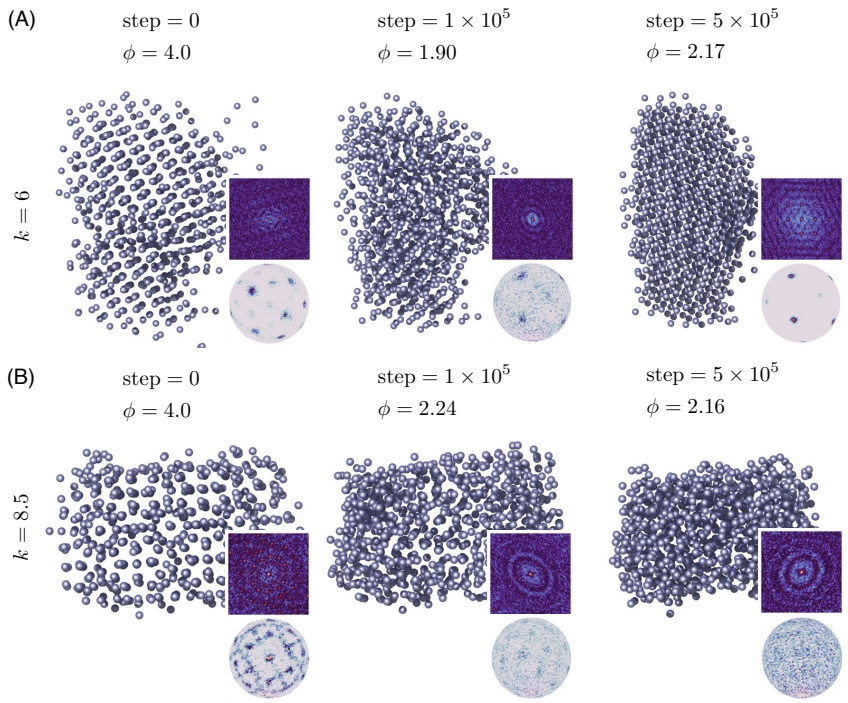

Figure 4.6: **Stability test for Oscillating Pair Potential system** (A) The evolution of the unbiased EDA simulation snapshots at $k = 6$, starting from the equilibrium $hP2$ phase at $k_B T = 0.1$ and $\phi = 4$. (B) The evolution of the unbiased EDA simulation snapshots at $k = 8.5$, starting from the equilibrium $cP54$ phase at $k_B T = 0.1$ and $\phi = 4$.

We can compare our findings to those reported in a detailed mapping of the OPP

phase diagram for different values of (k, ϕ) reported in Dshemuchadse et al. [174]. The EDA simulations discovered crystals at consistent values of ϕ for the two values of k we studied. However, instead of the $hP2$ (compressed hcp/$hP2$ Mg) and $cP54$ (clathrate-I) phases reported by Dshemuchadse et al. [174], EDA found a disordered phase at the same location of $\phi = 4$. This discrepancy can be explained by the instability of the complex crystals for the values of ϕ and k we explored, as Dshemuchadse et al. [174] found $hP2$ and $cP54$ to be stable only within a very narrow ϕ regime. In Figure 4.6, We perform stability tests to confirm by setting up both crystals, running a quick EDA simulation, and setting the bias potential to zero, i.e., allowing ϕ to fluctuate freely, and found that neither crystal is stable in our exploratory regions.

For $k = 8.5$, the three main crystals (A15, σ, and β-manganese phases) discovered by EDA exhibit comparable stability, each with nearly the same $\phi \approx 2$. In other words, a single particle design with $\phi \approx 2$ can stabilize multiple phases simultaneously. In fact, the free energy plot in FIG. 4.4 (D) demonstrates that the sampled points for these three crystals distribute as three nearly horizontal strips along the ϕ direction, indicating that all three phases can remain metastable across a broad range of ϕ. This result highlights the advantage of EDA over the DA method, as EDA provides a more comprehensive perspective on metastability within the generalized ensemble and identifies all metastable phases for each particle design. To confirm the stability of crystals obtained through the EDA simulations, we conducted a stability test using the $k = 8.5$ OPP system, initialized from sampled snapshots of three crystals at $\phi = 2$ in the middle of the EDA simulation. We then performed a standard NVT MC simulation. Each system remained stable after running 5 million MC simulation steps. This suggests that all the sampled crystals are either thermodynamically stable or metastable at $\phi = 2$.

4.4 Conclusion

In summary, we proposed and demonstrated a new approach – Exploratory Digital Alchemy (EDA) – to discovering thermodynamically stable and metastable crystal structures within design space defined by pair potentials and their alchemical parameters. To improve the original Digital Alchemy (DA) framework by incorporating a self-exploration feature, we employed a history-dependent bias potential commonly used in the well-known metadynamics method. We applied EDA to systems of point particles interacting with multi-well pair potentials: a two-dimensional system with the Lennard-Jones Gauss Potential (LJGP) and a three-dimensional system with the Oscillating Pair Potential (OPP). For LJGP, we successfully reproduced the stable phases previously reported at low temperature. In addition, we observed transitions between metastable phases at higher temperatures for the first time. For OPP, we tested EDA with two different wave numbers k. At a lower k, EDA produced three metastable crystal phases with two simple lattice structures and one Frank-Kasper phase. At higher k, EDA produced three metastable Frank-Kasper phases. Our work tackles the sampling problem in design space by introducing an enhanced sampling method to digital alchemy. The combination of the two methods also enables quantitative analysis of the free energy surface as a function of the alchemical degree of freedom.

CHAPTER V

Towards A New Generation of Molecular Simulation Engine: Successor of HOOMD-blue in Rust

This chapter is a collaborative work of the HOOMD-rs developers in Glotzer Lab: Joshua Anderson, Jen Bradley, Joseph Burkhart, Domagoj Fijan, Shih-Kuang Lee, Michelle Thran, Trevor Teague, and Melody Zhang. The development of the HOOMD-rs is initiated and managed by Joshua Anderson. The main contributors to Section 5.4.1 are Shih-Kuang Lee and Joseph Burkhart. The main contributors to Section 5.4.2 are Shih-Kuang Lee and Melody Zhang.

5.1 Introduction

The field of molecular simulation has experienced significant evolution over the past few decades. It has advanced from simple hard-sphere [184, 185, 186] and pairwise potentials, like the Lennard-Jones potential [187, 188], on serial machines to highly scalable, GPU-accelerated engines capable of simulating millions of particles [189, 190]. However, despite these advancements, a persistent tension persists within the discipline, particularly regarding trade-offs among performance, correctness, maintainability, and flexibility. Especially, the flexibility to include new methods and customize simulations is crucial in the rapidly growing soft matter commu-

nity. The most widely used molecular simulation packages—such as LAMMPS [168], GROMACS [191], and HOOMD-blue [88] from the Glotzer Lab—were developed in an era where performance and scientific correctness were the primary focuses, while other aspects were considered secondary. These tools, mostly written in C and C++, have made significant contributions to science, but they also have notable deficiencies. These include, but are not limited to: a lack of memory safety [192]—a fundamental issue with software based on C or C++ that potentially can corrupt the results generated, undermining the scientific correctness, steep learning curves for LAMMPS and GROMACS, and the accumulation of technical difficulties as new modules are added to the existing codebase.

This chapter presents a fundamental departure from that paradigm: my works contribute to the development of a next-generation molecular simulation engine, entitled HOOMD-rs, conceived as a spiritual and technical successor to HOOMD-blue, but implemented from the ground up in Rust [193]—a modern systems programming language that promises to reconcile performance with safety in a way that C++ never could. Besides, the HOOMD-blue design concept is redefined and adopted in HOOMD-rs to enhance the flexibility for customizing simulations. By leveraging Rust's unique combination of memory safety without garbage collection, expressive, and flexible type system, we aim not merely to replicate existing functionality, but to redefine what is possible in high-performance scientific computing. The motivation is both practical and philosophical. Practically, the increasing complexity of modeling the nanoparticle system. For example, the implementation of newly developed X-ChIMES to aid inverse design or the extended ensemble sampling in Digital Alchemy demands robust, debuggable, and extensible codebases that can evolve over time. Philosophically, I believe that the tools scientists use to model the physical

world should themselves be governed by rigorous, verifiable principles. Rust offers a path toward that ideal: a language in which entire classes of bugs are eliminated at compile time, in which parallel performance is safe by default, and in which abstraction does not come at the cost of efficiency.

The remainder of this chapter is organized as follows: I begin by reviewing the aforementioned choices of simulation engines: LAMMPS, GROMACS, and HOOMD-blue in Section 5.2. Then I briefly explain the reason to choose Rust to build the HOOMD-rs in Section 5.3. Finally, in Section 5.4.1 and 5.4.2, I demonstrate my key contribution to the Molecular Dynamics (MD) symplectic integrators and ChIMES potential modules under the HOOMD-rs.

5.2 The Landscape of Molecular Simulation Engines

To understand the significance of this Rust-based development, it is essential to survey the current ecosystem of molecular simulation engines and identify the gaps that motivate this work.

1. LAMMPS [168], developed at Sandia National Laboratories, remains the de facto standard for large-scale classical simulations. Its strength lies in its vast reservoir of implementation, supporting simulations across a broad spectrum of communities, including material science, soft matter, and biomolecules, and its ability to execute parallel simulations on hundreds of cores. Moreover, it offers comprehensive documentation and a large community that contributes to code development. However, its C++ codebase is massive, its build system is complex, and its potential memory safety issues make development work require a very experienced user.

2. GROMACS [191], originally developed in the Department of Biophysical Chemistry at the University of Groningen, optimized for biomolecular simulation, achieves

extraordinary performance through hand-tuned assembly and single-instruction multiple-data intrinsics interface. Yet its architecture reflects decades of incremental evolution: a mix of C++98, C++11, and modern C++ features, with significant technical debt in its force computation kernels. GPU offloading, while powerful, is implemented via CUDA and requires separate code paths. Besides, using GROMACS primarily involves command-line tools, which can be less intuitive for beginners.

3. HOOMD-blue [88], developed in the Glotzer Lab at the University of Michigan, represented a remarkable rethinking of molecular simulation software when it was introduced in 2007. By exposing a high-level Python application programming interface (API) with a high-performance C++/CUDA backend supporting both CPU and GPU execution, it dramatically lowered the barrier to learning molecular simulation techniques and conducting GPU-accelerated simulations. Additionally, HOOMD-blue's Python API allows users to define and execute simulations and combine them with other tools in the scientific Python ecosystem. Its built-in custom updater and force functionalities can be used to customize the simulation by changing the system's microstate or adding extra force with a Python script. Such flexibility is crucial for developing the method within the rapidly growing soft matter community, such as the HOOMD-blue modules developed in the Chapter III. Still, its core is written in C++ and CUDA, and maintaining it has become increasingly challenging. More critically, manual memory management remains a source of potential crashes and memory leaks. Besides, the customizability of simulations is constrained by the computational costs associated with communication between the C++/CUDA backend and the Python API, making customized mid- to large-scale simulations that rely heavily on user-defined Python scripts expensive.

5.3 Why Rust for Molecular Simulation?

Traditional molecular simulation engines are written in C++ because it offers fine-grained control over memory layout, cache behavior, and instruction-level optimization—qualities essential for achieving peak floating-point throughput on modern CPUs and GPUs. However, this control comes at a steep price. For instance, memory safety in C++ is opt-in and fragile: a single misplaced pointer can silently corrupt simulation data, a use-after-free can crash a week-long run, and a data race in a parallel force loop can produce irreproducible results. Moreover, the complexity of modern C++—with its template metaprogramming, concepts, and ever-evolving standards—has made large codebases increasingly challenging to maintain. Error messages are cryptic, and onboarding new contributors requires months of apprenticeship. As computational science becomes more interdisciplinary, these issues create significant barriers to entry. Such frustrations are common in large-scale simulation groups, such as the Glotzer Lab.

Rust was designed to solve precisely these problems. At its core is an ownership model that tracks the lifetime and exclusivity of every variable at compile time. The borrow checker enforces strict rules: data can have either one mutable reference or any number of immutable references, but never both simultaneously. This eliminates use-after-free, double-free, and data races by construction. For molecular simulation, these guarantees are transformative. For example, force computation in large-scale simulation has to be parallelized and inherently memory-intensive; in Rust, spawning thousands of threads to evaluate pairwise interactions is not only efficient but provably safe. The language's Send and Sync traits ensure that only data that can be safely shared across threads is shared, preventing an entire class of con-

currency bugs that troubles C++ developers. Uniquely, unlike languages relying on a garbage collector to maintain memory safety, Rust achieves this with zero runtime overhead—there is no tracing, no stop-the-world pauses, and no hidden allocations.

Beyond safety, Rust offers expressiveness without sacrifice. Generic programming via traits and const generics allows simulation parameters to be encoded at compile time, enabling the compiler to automatically optimize the program. The powerful typing system, combined with the guaranteed memory safety, allows us to create a highly customizable, modular, and correct simulation engine.

In short, Rust allows us to write code that is as fast as hand-tuned C++, as safe as Python without garbage collector, as maintainable as a modern high-level language, and can be as flexible and customizable as HOOMD-blue with Python API. For a domain where flexibility and correctness are essential, and performance is crucial, this combination is unprecedented.

5.4 HOOMD-rs Implementation

To briefly introduce HOOMD-rs, I start with an overview of the architecture design. Notice that the HOOMD-rs is still in construction, so the architecture may change over time. The HOOMD-rs is structured as a Rust package (hoomd-rs) with several library crates; in Rust, library crates do not compile to an executable but instead provide functionality intended to be shared with other projects. I illustrate the important components in Figure 5.1:

These crates provide various utilities for performing molecular simulations:

1. **hoomd-utility**: Implementations of new data types that ensure values are physically reasonable, such as positive real numbers.

2. **hoomd-vector**: Implementations that are used to describe vectorial quantities,

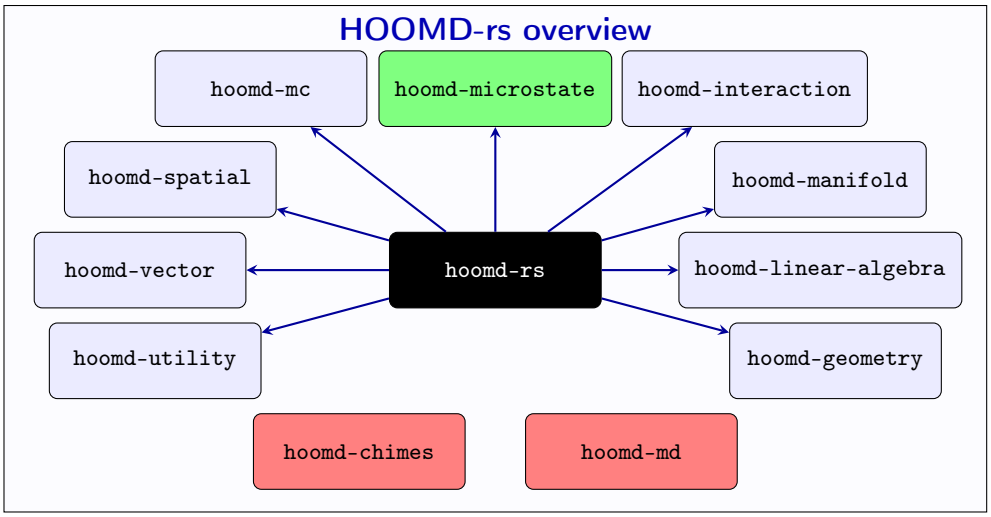

Figure 5.1: Component architecture of the HOOMD-rs.

such as positions, and to perform vector algebra, such as dot and cross product.

3. **hoomd-linear-algebra**: Implementations of methods related to matrices, with limited functionality that mimics numpy, such as singular value decomposition.

4. **hoomd-spatial**: Particle neighborlist techniques to accelerate interaction calculation.

5. **hoomd-manifold**: Methods for running particle simulations in non-Euclidean curved space.

6. **hoomd-geometry**: Implementations for describing geometric shapes. Use case includes simulation box geometry and particle shapes.

7. **hoomd-interaction**: Implementations of potential energy functions, including functionality to describe systems of hard particle shapes.

8. **hoomd-mc**: Implementations of Monte Carlo integrators to perturb the system's configuration to sample the Boltzmann distribution, including the hard particle Monte Carlo (HPMC) method [88] in HOOMD-blue.

The main breaking change that makes HOOMD-rs a more robust and flexible simulation engine is that HOOMD-rs is a "microstate-centric" simulation engine. The

hoomd-microstat crate implements such an idea and implements the *microstate* type. The *microstate* type in hoomd-rs encapsulates all fundamental thermodynamic quantities associated with a system, the same concept as in statistical mechanics: particle positions, momenta, rigid-body configurations, and simulation box geometry, etc. In HOOMD-rs, every operation the user intends to apply changes only the *microstate*, and it should return the updated or identical *microstate* to the user without saving copies for other purposes. The user would have full control over their system, maximizing flexibility for customizing a simulation. Furthermore, we share responsibility for managing the interplay between different simulation sub-modules with the user; there will be no built-in manager to help the user manage these interplays. For example, to perform simulations of particle shape in the canonical ensemble using HPMC method, primarily, the user would define and use *microstate* that describes a system of particle shapes, *interaction* that evaluates the volume exclusion effects, and an HPMC *integrator* for integrating particle positions and orientations. Then, the user will iterate the integration step until it meets a user-defined convergence criterion. Within each iteration, the user will pass the *microstate* to the *integrator*. Internally, the *integrator* uses the *interaction* to evaluate the volume exclusion effects to accept or reject a perturbation to the system based on Metropolis acceptance rule. Finally, *integrator* returns the updated *microstate*. When the *microstate* is passed to the *integrator*, all the associated modules would perform necessary actions on the *microstate*, without saving any copies of the *microstate* for other purposes. Such treatment ensures that the *microstate*, which is the most crucial component for later analysis, will not be silently or accidentally changed due to Rust's strict ownership system; hence, correctness is guaranteed.

Next, I will showcase the development of the Molecular Dynamics (MD) integrator

and the ChIMES potential modules under HOOMD-rs as one of the library crates.

5.4.1 Modular and Flexible MD Symplectic Integrator

The **hoomd-md** crate provides the essential methods for performing MD simulations. Future updates will introduce additional methods; however, the current implementation includes three primary components: 1. the constant-volume integrator, 2. the thermostat, and 3. the method to initialize momenta. The constant-volume integrator serves as the core of the MD method, enabling the integration of Newton's equations of motion to produce the microcanonical (NVE) ensemble. When a thermostat is used, it will yield the canonical (NVT) ensemble. My implementation of the constant-volume integrator follows the formulation in Tuckerman et al. [194], resulting in a phase-space-volume-preserving, symplectic integrator. I provide pseudo-code for illustrating the MD integration in each iteration from the user side:

Pseudocode V.1: MD integration

```
...

integrator.integrate_translation_step_one(
    &mut microstate,
    &mut thermostat,
    &macrostate,
);

integrator.integrate_rotation_step_one(
    &mut microstate,
    &mut thermostat,
    &macrostate,
);

integrator.update_force_and_torque(&mut microstate, &interaction);

integrator.integrate_translation_step_two(
    &mut microstate,
    &mut thermostat,
    &macrostate,
);

integrator.integrate_rotation_step_two(
    &mut microstate,
    &mut thermostat,
    &macrostate,
);

...
```

The integrator implements five methods used here: *integrate_translation_step_one* and *two*, *integrate_rotation_step_one* and *two*, and *update_force_and_torque*. The *macrostate* represents the important macroscopic quantity set points, associated with the ensemble that the user intends to sample. When running an NVE ensemble simulation, *macrostate* functions as a placeholder that does not store any data. When running an NVT ensemble, *macrostate* holds the temperature set point for the *thermostat* to maintain the system's temperature. Consistent with HOOMD-rs design principles, the user retains ownership of both the microstate and the macrostate. The *thermostat* would not own and know the temperature set point; the user must provide it to perform extra treatment to maintain the temperature. Now, to perform one iteration that advances the system from time t to $t+dt$. Following the sympletic integration scheme, I apply the *integrator.integrate_translation_step_one* and *integrator.integrate_rotation_step_one* on *microstate* to advance the momentum and angular momentum to time $t + dt/2$ and the position and orientation to $t + dt$. For NVT simulation, before updating the momenta and angular momenta, the *integrator* will use the provided *thermostat* to advance the thermostat state to time $t + dt/2$ and rescale the momenta and angular momenta to maintain the temperature. After the position and orientation are updated, I evaluate the force and torque at time $t + dt$ using *integrator.update_force_and_torque* and store the results in the *microstate*, with the chosen *interaction*. Finally, I advance the *microstate's* momentum and angular momentum from time $t + dt/2$ to $t + dt$; then, if necessary, rescale the momenta and angular momenta, and advance the thermostat state. The key that makes this integrator robust to newly introduced methods and flexible is that the force and torque calculation is isolated from the microstate integration. Whenever a user intends to invoke extra force or torque calculation, for example, an external

field, bias force, or active torque that makes the particle actively rotate, a user would only need to modify the *update_force_and_torque* method, and this modification would guarantee not to affect elsewhere in the integrator due to Rust's ownership system, less likely to introduce error accidentally and easier to debug.

Additionally, I perform simulations to examine the integrator by calculating the error in energy conservation versus the discretized simulation step dt, and comparing it with HOOMD-blue. To quantify the error of energy conservation, following the method in the work [195], the energy can be viewed as a noisy time series $A(t)$. The $A(t)$ can be modeled as a linear function with a Gaussian noise as: $A(t) = bt+c+\epsilon(t)$, where the slope b and intercept c can be obtained via linear regression, and ϵ is the Gaussian noise with zero mean and standard deviation σ, which can be estimated from samples. Here, define a quantity, drift $\delta = bt_{\max}/\sigma$, where $bt_{\max}$ can be seen as the maximum expected drift in the energy when the simulation reaches the time $t_{\max}$. In Figure 5.2, I demonstrate the δ and $bt_{\max}$ as a function of various squared simulation step dt^2.

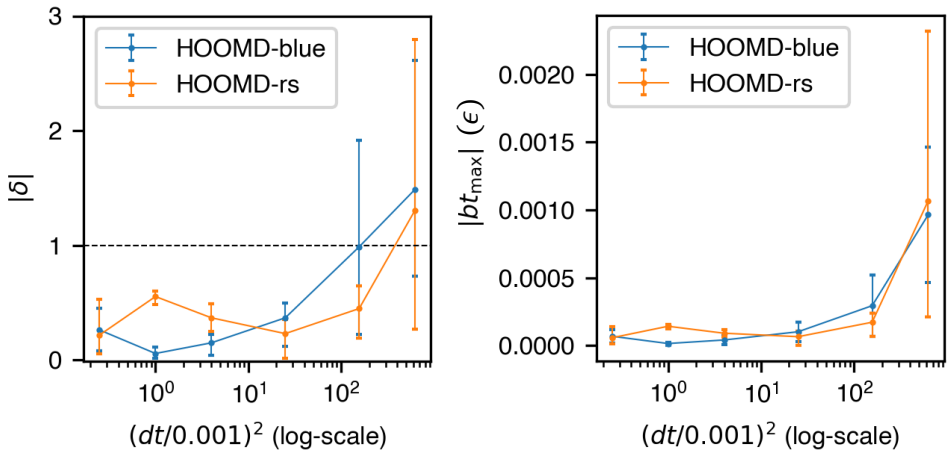

Figure 5.2: **The error in energy conservation** The error bars indicate the maximum and minimum values of the samples.

I use a two-dimensional Lennard-Jones-Gauss potential system, with the first

well depth of ϵ, and 100 particles at $k_BT = 0.15\epsilon$. Five NVT simulations are run at each dt to obtain statistics for both simulation engines. Simulations are run for N steps, where $dt \times N = 1000 = t_{\max}$. Notice that since the NVT simulations are run using the Nosé-Hoover thermostat [152], the energy measured here is the total energy plus the energy possessed by the virtual thermal reservoir. The original paper claims that when the magnitude of δ exceeds one, it indicates potential problems in the conducted simulations, such as an improper implementation or unreasonable simulation parameters. The results show that both simulation engine has comparable drift. The drift starts to exceed one when an unreasonably large dt is used here, i.e, the maximum I use here is $dt = 0.025$. Given that the $|bt_{max}|$ is small when I use a reasonable dt, I attribute the large drift to the unreasonable choice of simulation step, rather than erroneous implementation.

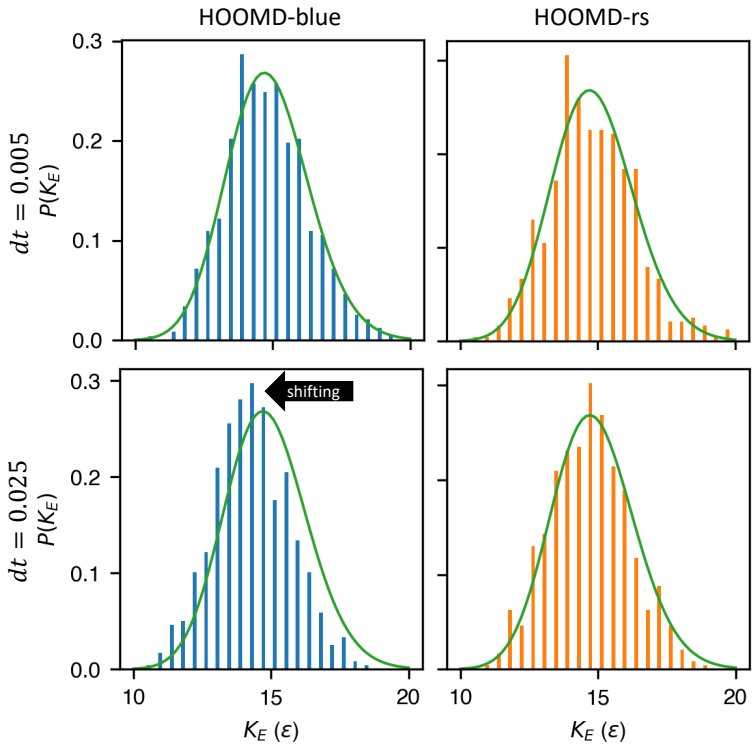

Figure 5.3: **The kinetic energy distribution** The green curve is the ground truth gamma distribution of Gamma($\alpha = N_{\text{dof}}/2, \theta = k_BT$), and the bar plots are sampled from either HOOMD-blue or HOOMD-rs.

Next, I examine the kinetic energy distribution sampled by both simulation engines. The kinetic energy distribution must have the form of a gamma distribution of Gamma($\alpha = N_{\text{dof}}/2, \theta = k_B T$) for the NVT ensemble, where α is the shape parameter, θ is the scale parameter, and N_{dof} is the number of degrees of freedom. Figure 5.3 shows the sampled kinetic energy distribution (bar plots) compared with the ground truth gamma distribution (green curve). The results show that at $dt = 0.005$, both simulation engines reproduce the correct distribution, i.e., the correct NVT ensemble. However, at the largest $dt = 0.025$, while HOOMD-rs still produces a correct distribution, HOOMD-blue shows a noticeable leftward shift, indicating that the NVT ensemble is not correctly sampled and that the system has a temperature lower than the set point on average. To this point, it is the first step to confirm that the MD HOOMD-rs implementation is correct, but it still requires more stringent tests in the future, including tests of rotational degrees of freedom. Moreover, it requires further investigation into why HOOMD-blue would produce more erroneous results, potentially due to an incorrect implementation.

5.4.2 ChIMES Machine Learning Potential Module

In this section, I demonstrate the ChIMES potential modules implemented in HOOMD-rs. As I described in the Section 3.3.2, ChIMES is a physics-informed machine learning potential model. It's a general-purpose potential model that can describe a wide range of systems. By integrating this model into HOOMD-rs, it would empower HOOMD-rs to simulate a broader spectrum of systems, including the atomistic system at quantum mechanical precision and with the cost of classical simulations [141, 127]. Furthermore, as demonstrated in Chapter III, the ChIMES model has proven to be both useful and robust for describing coarse-grained nanoparticle systems, such as polymer-grafted nanoparticles. I anticipate that more researchers

in the soft matter community will adopt the ChIMES model for simulating nanoparticle systems, which remains the primary focus of HOOMD-rs. These considerations motivate my work in this section. Note that the goal of this work is not to build a module for both potential fitting functionality and potential energy calculation. The goal is only to implement the potential energy calculation and to exploit potential fitting results obtained elsewhere, such as those from ChIMES-LSQ [155]. Currently, only the two-body potential energy calculation is implemented.

The ChIMES model was mainly implemented in the LAMMPS [156] as a USER package and can be compiled with a specific version of LAMMPS as its *pair* style. Here, the model has been restructured and reimplemented as a library crate in HOOMD-rs, named **hoomd-chimes**. I aim to make HOOMD-rs support the ChIMES model natively, so a user will not need to take any extra compilation steps to use it. I illustrate the detailed component architectures of **hoomd-chimes** in Figure 5.4. The yellow and green nodes represent the Rust types and traits implemented within the **hoomd-chimes** crate, and the blue nodes are the Rust types and traits in **hoomd-interaction** as described in Section 5.4. The Rust trait is a unique concept in Rust for types that share the same behaviour. In the Figure, when an arrow points from a type to a trait, it means the type implements the trait that it points to. When the arrow points in the opposite direction, it means the trait is used within a type. If one type points to another, it means the latter type uses the former. I structure the diagram in Figure 5.4 to describe the module hierarchy in **hoomd-chimes**, with the lowest-level, basic implementation at the top. As can be seen, the ChIMES model is now highly modular, with each component representing an important building block of the ChIMES potential in Equation 3.14, such as the penalty function, smoothing function, and Chebyshev polynomial, etc. Such high

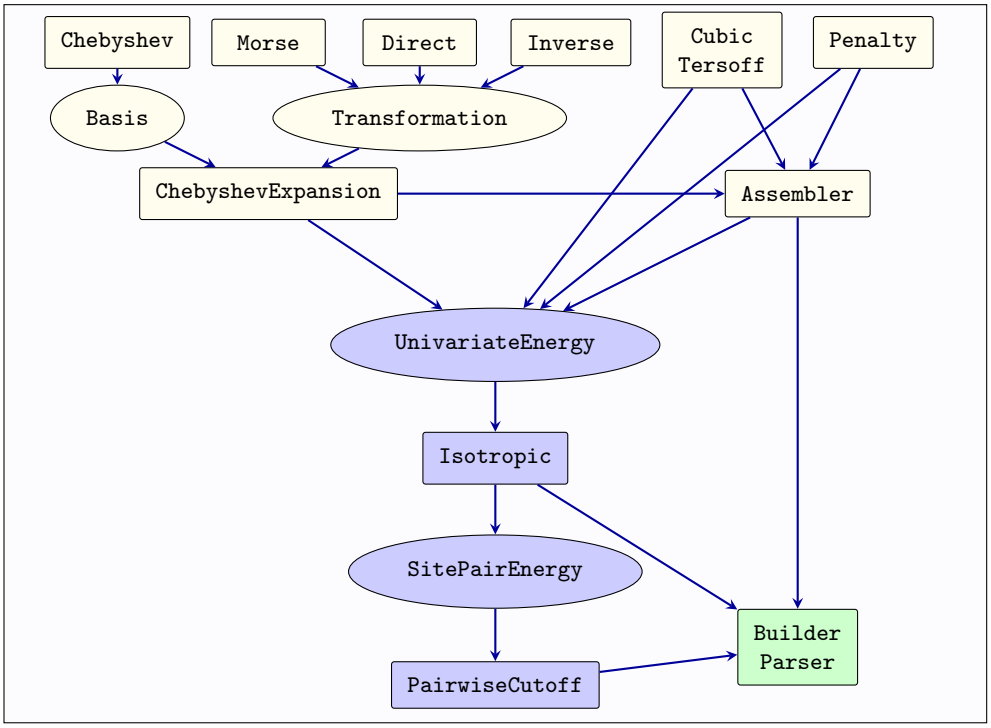

Figure 5.4: **Component architecture of the hoomd-chimes crate in Figure 5.1** The elliptical nodes represent Rust types, and the square nodes represent Rust traits. The Builder/Parser is the top-level type under hoomd-chimes that I expect the user would use.

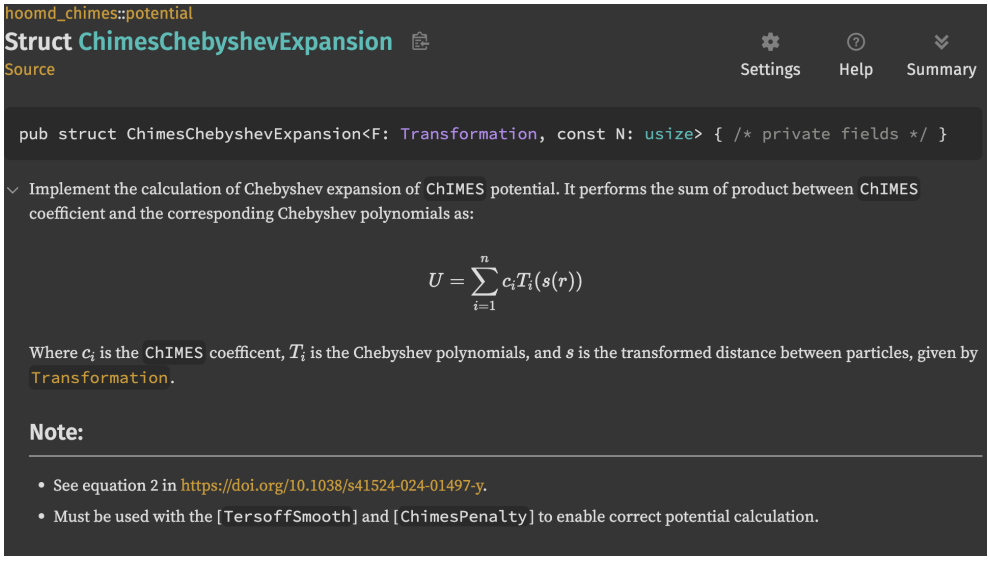

Figure 5.5: **Example documentation of *hoomd-chimes*** A snapshot of documentation of the ChebyshevExpansion type in Figure 5.4.

modularity allows a user to learn the detailed components of the ChIMES model without treating it as a machine learning black box. This purpose is also supported by detailed documentation, as shown in the Figure 5.5. Besides, the **hoomd-chimes** is heavily dependent on the **hoomd-interaction**, which is the core library crate that handles the potential calculation for HOOMD-rs. This treatment ensures that the ChIMES model integrates seamlessly into the architecture of the entire HOOMD-rs, so a user will need minimal effort to adopt and use it. In addition, to use the ChIMES model, an essential procedure is the potential fitting to obtain the optimal fitting parameters. The ChIMES model stores parameters in the TXT format [155]. To enable convenient use, I implement the builder/parser shown in Figure 5.4 (green node) to parse the potential parameters from the TXT file. Then, the Builder/Parser will use the remaining components of **hoomd-chimes** to construct a PairwiseCutoff type in **hoomd-interaction**, which will be ready for use in MC or MD simulations. I envision that a user will primarily use the Builder/Parser, but can also build the ChIMES potential using the provided basic components.

Here, I demonstrate a use case example of the **hoomd-chimes** using the two-body carbon model published by Lindsey et al. [141]. In the original publication, the simulations were run using LAMMPS's MD module. The NVT simulations were performed using the Nosé-Hoover thermostat at $T = 5000$ K and the density of 2.43 g/cm^3. It results in a liquid carbon state at equilibrium. To reproduce the result, I use the MC module of HOOMD-rs to perform the NVT simulation at the same temperature of $k_BT \simeq 9.94$ kcal/mol and the same density, i.e., 512 particles in a cubic box with the edge length $\simeq 16.137$ Å. I run the simulation for 1 million MC sweeps. The perturbation on each particle's position is performed in every MC sweep according to the Metropolis criterion. To ensure the system is in equilibrium and to compare with the results by Lindsey et al., I use the last 500000 sweeps to average the system and calculate the radial distribution function (RDF) in Figure 5.6.

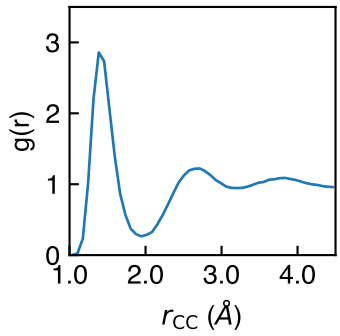

Figure 5.6: **The radial distribution function of two-body carbon model using ChIMES model in HOOMD-rs**

My RDF perfectly reproduces the RDF in Figure 3 of the original paper [141] - the blue curve labeled as ChIMES-2B. Note that, according to Lindsey et al. [141], accurately simulating the carbon system requires a many-body model. Here, the two-body carbon model I used is not to simulate the actual carbon system, but to validate the correctness of the two-body ChIMES model I implemented in HOOMD-

rs, which is an important first step for the future implementation of the many-body model part of ChIMES.

5.5 Conclusion

To conclude, I demonstrate the development of a new molecular simulation engine, entitled HOOMD-rs, implemented using Rust by the Glotzer Lab. I briefly review three powerful simulation engines, LAMMPS, GROMACS, and HOOMD-blue, that are currently standard in various fields. Then, since these engines are built in C++ and CUDA, they may face certain issues, particularly memory-safety concerns that could corrupt simulation results. Additionally, their flexibility may not sufficiently meet the simulation customization requirements of the soft-matter community. These limitations are the reasons for developing HOOMD-rs as a spiritual successor to HOOMD-blue, using Rust to address these challenges. Furthermore, I briefly introduce the current state of HOOMD-rs and detail my contributions to the Molecular Dynamics (MD) module and the implementation of the ChIMES potential module in HOOMD-rs. Validation simulations are conducted to demonstrate the correctness of both implementations. In the future, more rigorous test simulations should be done on both modules. Besides, I envision that many additional modules will be implemented in HOOMD-rs to construct a mature, robust new-generation simulation engine, such as the multi-thread simulations, constant-pressure MD integrator, the many-body ChIMES model, and Digital Alchemy simulations. With these advancements, the newly developed HOOMD-rs—prioritizing performance, correctness, and flexibility—aims to contribute to the soft-matter community and facilitate future research, just as HOOMD-blue has done to date.

CHAPTER VI

Conclusion and Outlook

6.1 Summary of Contributions

This dissertation advances the computational toolkit for nanomaterial design by addressing critical challenges in characterization, modeling, and discovery. Each study contributes to the overarching objective of establishing a programmable design-to-discovery pipeline for colloidal systems.

In **Chapter II**, I addressed the limitations of traditional order parameters by developing a physics-agnostic machine learning (ML) classifier. Utilizing a novel shape-symmetry encoded data augmentation technique, I demonstrated that a multilayer perceptron (MLP) can accurately identify complex local environments across diverse particle shapes, such as truncated polyhedra and patchy particles, without requiring pre-engineered symmetry functions.

In **Chapter III**, I introduced the **X-ChIMES** model, which extends the many-body machine learning potential model to enable inverse design of polymer-grafted nanoparticles (PGNs). Additionally, I developed an efficient method for calculating the Potential of Mean Force (PMF) in PGNs using the Forward-Reversed Method. This framework supports simultaneous modeling of interaction potentials across a continuous design space, such as varying polymer lengths, and facilitates inverse

design via Digital Alchemy (DA).

In **Chapter IV**, I advanced the DA framework into **Exploratory Digital Alchemy (EDA)**. By integrating Well-Tempered Metadynamics, I removed the need for a predefined target structure. This advancement enables the system to autonomously discover stable and metastable phases while optimizing building-block attributes, thereby establishing an intelligent forward-design framework. I demonstrated the versatility of EDA using Lennard-Jones-Gauss and Oscillating Pair Potential systems.

Finally, **Chapter V** presents the development of **HOOMD-rs**, a next-generation simulation engine implemented in Rust. This work establishes a foundation for future scientific computing by combining high performance with memory safety and modularity. My contributions include implementing the core Molecular Dynamics (MD) modules and the ChIMES potential modules, ensuring the engine is equipped with a machine learning potential.

6.2 General Significance

The overarching impact of this work arises from the synergy among its components. The ML classifier from Chapter II serves as the analytical tool for exploratory frameworks in Chapter IV and for broader challenges in identifying crystal structures assembled by anisotropic colloids. The X-ChIMES models from Chapter III sketch the design space. Potentially, these innovations can be integrated into a high-performance, customizable engine such as HOOMD-rs, to build a comprehensive ecosystem for navigating the vast design space of colloidal nanotechnology. This multifaceted approach to nanomaterial characterization and design provides a critical toolkit for addressing current limitations in nanomaterial discovery.

6.3 Outlook and Future Work

While this dissertation establishes a robust foundation, several promising avenues for future research remain:

The supervised MLP classifier described in Chapter II is highly effective within its training domain. It is currently best suited for rapid analysis of large-scale datasets where target structures are known *a priori*, such as when sampling thousands of independent nucleation and crystallization pathways to obtain statistical insights. Due to its simplicity and promising classification performance, our method may be applied to study nucleation pathways in colloidal systems using biased simulation techniques, such as umbrella sampling and metadynamics [78]. Such applications will be explored in future works. Besides, our framework can be easily extended to a system with multiple particle types, which can assemble more complex crystal structures [196]. Such an extension can be achieved by including information about the neighboring particle type as part of the local environment descriptor, along with augmented relative positions and orientations.

However, the generalizability of this approach could be enhanced by training on a more comprehensive library of metastable and equilibrium crystal structures. Looking ahead, transitioning toward unsupervised learning represents the next frontier. A Variational Autoencoder (VAE)-based [197, 198, 199, 200] or a diffusion model [201] could be employed to learn latent representations of local environments without human labels. Such a model could detect novel phases absent from the training set and serve as a generative tool to assist in sampling rare nucleation events. Integrating the data augmentation techniques developed in this work with a generative model would likely yield a robust, symmetry-aware discovery tool.

For the X-ChIMES model in Chapter III, a natural extension is to construct a higher-dimensional design surface. Beyond polymer length, future models could incorporate chemical attributes such as bead types (hydrophilic or hydrophobic) or solvent conditions. Besides, integrating Exploratory Digital Alchemy (EDA) with X-ChIMES would be transformative. Applying EDA to PGNs would enable the discovery of phases in systems where the interaction landscape is inherently many-body and computationally intensive to map manually. In addition, to enable the ChIMES model to describe a broader range of colloidal particles, particularly anisotropic particles, it is essential to add an anisotropic basis function component to the current ChIMES model, which is presently constructed as an isotropic potential model. Several attempts have been made to achieve this goal, including the anisotropic scalar basis function used by Campos-Villalobos et al. [202] and a generalized version of the ChIMES model that incorporates anisotropic basis functions [203]. These approaches could potentially be adopted in the X-ChIMES model to enable the inverse design of anisotropic particles.

The development of HOOMD-rs described in Chapter V represents an ongoing effort to create a more robust and customizable simulation infrastructure. Future work should focus on expanding the crate ecosystem to include advanced techniques utilized throughout this dissertation. Specifically, implementing modules for collective variable calculations, Steered Molecular Dynamics, and a comprehensive suite of Metadynamics tools would enable HOOMD-rs to perform rare-event sampling. Additionally, expanding the many-body potential modules to support the full ChIMES framework would allow HOOMD-rs to leverage the capabilities of the ChIMES machine learning potential.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Yuan Zhou, Rose K Cersonsky, and Sharon C Glotzer. A route to hierarchical assembly of colloidal diamond. *Soft Matter*, 18(2):304–311, 2022.

[2] Devleena Samanta, Wenjie Zhou, Sasha B Ebrahimi, Sarah Hurst Petrosko, and Chad A Mirkin. Programmable matter: the nanoparticle atom and dna bond. *Advanced Materials*, 34(12):2107875, 2022.

[3] Mayank Agrawal and Sharon C Glotzer. Scale-free, programmable design of morphable chain loops of kilobots and colloidal motors. *Proceedings of the National Academy of Sciences*, 117(16):8700–8710, 2020.

[4] Michael A. Boles, Michael Engel, and Dmitri V. Talapin. Self-Assembly of Colloidal Nanocrystals: From Intricate Structures to Functional Materials. *Chem. Rev.*, 116(18):11220–11289, September 2016.

[5] Li Wang, Coucong Gong, Xinzhu Yuan, and Gang Wei. Controlling the Self-Assembly of Biomolecules into Functional Nanomaterials through Internal Interactions and External Stimulations: A Review. *Nanomaterials*, 9(2):285, February 2019.

[6] Jinhui Wang, Xiaojia Zhao, Jingfeng Li, Xiao Kuang, Yuqian Fan, Gang Wei, and Zhiqiang Su. Electrostatic Assembly of Peptide Nanofiber–Biomimetic Silver Nanowires onto Graphene for Electrochemical Sensors. *ACS Macro Lett.*, 3(6):529–533, June 2014.

[7] Qi Li, Yi Jia, Luru Dai, Yang Yang, and Junbai Li. Controlled Rod Nanostructured Assembly of Diphenylalanine and Their Optical Waveguide Properties. *ACS Nano*, 9(3):2689–2695, March 2015.

[8] Soubantika Palchoudhury, Ziyou Zhou, Karthik Ramasamy, Franklin Okirie, Peter E. Prevelige, and Arunava Gupta. Self-assembly of P22 protein cages with polyamidoamine dendrimer and inorganic nanoparticles. *J. Mater. Res.*, 32(2):465–472, January 2017.

[9] Kimberly McCoy, Masaki Uchida, Byeongdu Lee, and Trevor Douglas. Templated Assembly of a Functional Ordered Protein Macromolecular Framework from P22 Virus-like Particles. *ACS Nano*, 12(4):3541–3550, April 2018.

[10] Masaki Uchida, Kimberly McCoy, Masafumi Fukuto, Lin Yang, Hideyuki Yoshimura, Heini M. Miettinen, Ben LaFrance, Dustin P. Patterson, Benjamin Schwarz, Jonathan A. Karty, Peter E. Prevelige, Byeongdu Lee, and Trevor Douglas. Modular Self-Assembly of Protein Cage Lattices for Multistep Catalysis. *ACS Nano*, 12(2):942–953, February 2018.

[11] Zhe Li, Minghui Liu, Lei Wang, Jeanette Nangreave, Hao Yan, and Yan Liu. Molecular Behavior of DNA Origami in Higher-Order Self-Assembly. *J. Am. Chem. Soc.*, 132(38):13545–13552, September 2010.

[12] Thomas Tørring, Niels V. Voigt, Jeanette Nangreave, Hao Yan, and Kurt V. Gothelf. DNA origami: a quantum leap for self-assembly of complex structures. *Chem. Soc. Rev.*, 40(12):5636, 2011.

[13] Chad A Mirkin and Sarah Hurst Petrosko. Inspired beyond nature: three decades of spherical nucleic acids and colloidal crystal engineering with dna, 2023.

[14] A Paul Alivisatos, Kai P Johnsson, Xiaogang Peng, Troy E Wilson, Colin J Loweth, Marcel P Bruchez Jr, and Peter G Schultz. Organization of'nanocrystal molecules' using dna. *Nature*, 382(6592):609–611, 1996.

[15] Pablo F Damasceno, Michael Engel, and Sharon C Glotzer. Crystalline assemblies and densest packings of a family of truncated tetrahedra and the role of directional entropic forces. *Acs Nano*, 6(1):609–614, 2012.

[16] Joel Henzie, Michael Grünwald, Asaph Widmer-Cooper, Phillip L Geissler, and Peidong Yang. Self-assembly of uniform polyhedral silver nanocrystals into densest packings and exotic superlattices. *Nature materials*, 11(2):131–137, 2012.

[17] Kwanghwi Je, Sangmin Lee, Erin G Teich, Michael Engel, and Sharon C Glotzer. Entropic formation of a thermodynamically stable colloidal quasicrystal with negligible phason strain. *Proceedings of the National Academy of Sciences*, 118(7):e2011799118, 2021.

[18] Yein Lim, Sangmin Lee, and Sharon C Glotzer. Engineering the thermodynamic stability and metastability of mesophases of colloidal bipyramids through shape entropy. *ACS nano*, 2023.

[19] Greg van Anders, Daphne Klotsa, N. Khalid Ahmed, Michael Engel, and Sharon C. Glotzer. Understanding shape entropy through local dense packing. *Proc Natl Acad Sci USA*, 111(45):E4812–E4821, November 2014.

[20] Daphne Klotsa, Elizabeth R. Chen, Michael Engel, and Sharon C. Glotzer. Intermediate crystalline structures of colloids in shape space. *Soft Matter*, 14(43):8692–8697, 2018.

[21] Duanduan Wan, Chrisy Xiyu Du, Greg van Anders, and Sharon C. Glotzer. Fcc & bcc phase transitions in convex and concave hard particle systems. *J. Phys. Chem. B*, 123(42):9038–9043, October 2019.

[22] Zhenli Zhang and Sharon C Glotzer. Self-assembly of patchy particles. *Nano letters*, 4(8):1407–1413, 2004.

[23] Wenjie Fei, Yang Gu, and Kyle JM Bishop. Active colloidal particles at fluid-fluid interfaces. *Current opinion in colloid & interface science*, 32:57–68, 2017.

[24] Lang Feng, Rémi Dreyfus, Ruojie Sha, Nadrian C Seeman, and Paul M Chaikin. Dna patchy particles. *Adv. Mater*, 25(20):2779–2783, 2013.

[25] Gabriel Loget and Alexander Kuhn. Bulk synthesis of janus objects and asymmetric patchy particles. *Journal of Materials Chemistry*, 22(31):15457–15474, 2012.

[26] Greg van Anders, N Khalid Ahmed, Ross Smith, Michael Engel, and Sharon C Glotzer. Entropically patchy particles: engineering valence through shape entropy. *Acs Nano*, 8(1):931–940, 2014.

[27] Eric S Harper, Greg van Anders, and Sharon C Glotzer. The entropic bond in colloidal crystals. *Proceedings of the National Academy of Sciences*, 116(34):16703–16710, 2019.

[28] Rajneesh Kumar, Gabriele M. Coli, Marjolein Dijkstra, and Srikanth Sastry. Inverse design of charged colloidal particle interactions for self assembly into specified crystal structures. *J. Chem. Phys.*, 151(8):084109, August 2019.

[29] Uyen Tu Lieu and Natsuhiko Yoshinaga. Inverse design of two-dimensional structure by self-assembly of patchy particles. *J. Chem. Phys.*, 156(5):054901, February 2022.

[30] Greg van Anders, Daphne Klotsa, Andrew S Karas, Paul M Dodd, and Sharon C Glotzer. Digital alchemy for materials design: Colloids and beyond. *Acs Nano*, 9(10):9542–9553, 2015.

[31] Yina Geng, Greg van Anders, Paul M Dodd, Julia Dshemuchadse, and Sharon C Glotzer. Engineering entropy for the inverse design of colloidal crystals from hard shapes. *Science advances*, 5(7):eaaw0514, 2019.

[32] Luis Y Rivera-Rivera, Timothy C Moore, and Sharon C Glotzer. Inverse design of triblock janus spheres for self-assembly of complex structures in the crystallization slot via digital alchemy. *Soft Matter*, 19(15):2726–2736, 2023.

[33] Pengji Zhou and Sharon C Glotzer. Inverse design of isotropic pair potentials using digital alchemy with a generalized fourier potential. *The European Physical Journal B*, 94(12):243, 2021.

[34] Melody Yiyuan Zhang, Shih-Kuang Alex Lee, Sharon C Glotzer, and Rebecca K Lindsey. A generalized machine-learning framework for developing alchemical many-body interaction models for polymer-grafted nanoparticles. *Journal of Chemical Theory and Computation*, 21(19):9853–9867, 2025.

[35] Paul J Steinhardt, David R Nelson, and Marco Ronchetti. Bond-orientational order in liquids and glasses. *Physical Review B*, 28(2):784, 1983.

[36] Shankar Kumar, John M Rosenberg, Djamal Bouzida, Robert H Swendsen, and Peter A Kollman. The weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. *Journal of computational chemistry*, 13(8):1011–1021, 1992.

[37] Christopher Jarzynski. Nonequilibrium equality for free energy differences. *Physical Review Letters*, 78(14):2690, 1997.

[38] Ioan Kosztin, Bogdan Barz, and Lorant Janosi. Calculating potentials of mean force and diffusion coefficients from nonequilibrium processes without jarzynski's equality. *The Journal of chemical physics*, 124(6), 2006.

[39] Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proceedings of the national academy of sciences*, 99(20):12562–12566, 2002.

[40] Pratyush Tiwary and Michele Parrinello. A time-independent free energy estimator for metadynamics. *The Journal of Physical Chemistry B*, 119(3):736–742, 2015.

[41] Pengji Zhou, James C Proctor, Greg van Anders, and Sharon C Glotzer. Alchemical molecular dynamics for inverse design. *Molecular Physics*, 117(23-24):3968–3980, 2019.

[42] Shih-Kuang Alex Lee, Sun-Ting Tsai, and Sharon C Glotzer. Classification of complex local environments in systems of particle shapes through shape symmetry-encoded data augmentation. *The Journal of Chemical Physics*, 160(15), 2024.

[43] Michael A Boles, Michael Engel, and Dmitri V Talapin. Self-assembly of colloidal nanocrystals: From intricate structures to functional materials. *Chemical reviews*, 116(18):11220–11289, 2016.

[44] George M Whitesides and Mila Boncheva. Beyond molecules: Self-assembly of mesoscopic and macroscopic components. *Proceedings of the National Academy of Sciences*, 99(8):4769–4774, 2002.

[45] Zhe Li, Minghui Liu, Lei Wang, Jeanette Nangreave, Hao Yan, and Yan Liu. Molecular behavior of dna origami in higher-order self-assembly. *Journal of the American Chemical Society*, 132(38):13545–13552, 2010.

[46] Thomas Tørring, Niels V Voigt, Jeanette Nangreave, Hao Yan, and Kurt V Gothelf. Dna origami: a quantum leap for self-assembly of complex structures. *Chemical Society Reviews*, 40(12):5636–5646, 2011.

[47] Jinhui Wang, Xiaojia Zhao, Jingfeng Li, Xiao Kuang, Yuqian Fan, Gang Wei, and Zhiqiang Su. Electrostatic assembly of peptide nanofiber–biomimetic silver nanowires onto graphene for electrochemical sensors. *ACS Macro Letters*, 3(6):529–533, 2014.

[48] Qi Li, Yi Jia, Luru Dai, Yang Yang, and Junbai Li. Controlled rod nanostructured assembly of diphenylalanine and their optical waveguide properties. *ACS nano*, 9(3):2689–2695, 2015.

[49] Soubantika Palchoudhury, Ziyou Zhou, Karthik Ramasamy, Franklin Okirie, Peter E Prevelige, and Arunava Gupta. Self-assembly of p22 protein cages with polyamidoamine dendrimer and inorganic nanoparticles. *Journal of Materials Research*, 32(2):465–472, 2017.

[50] Masaki Uchida, Kimberly McCoy, Masafumi Fukuto, Lin Yang, Hideyuki Yoshimura, Heini M Miettinen, Ben LaFrance, Dustin P Patterson, Benjamin Schwarz, Jonathan A Karty, et al. Modular self-assembly of protein cage lattices for multistep catalysis. *ACS nano*, 12(2):942–953, 2018.

[51] Kimberly McCoy, Masaki Uchida, Byeongdu Lee, and Trevor Douglas. Templated assembly of a functional ordered protein macromolecular framework from p22 virus-like particles. *ACS nano*, 12(4):3541–3550, 2018.

[52] Li Wang, Coucong Gong, Xinzhu Yuan, and Gang Wei. Controlling the self-assembly of biomolecules into functional nanomaterials through internal interactions and external stimulations: A review. *Nanomaterials*, 9(2):285, 2019.

[53] J Wang and AL Ferguson. Nonlinear machine learning in simulations of soft and biological materials. *Molecular Simulation*, 44(13-14):1090–1107, 2018.

[54] Matthew Spellings and Sharon C Glotzer. Machine learning for crystal identification and discovery. *AIChE Journal*, 64(6):2198–2206, 2018.

[55] Carl S Adorf, Timothy C Moore, Yannah JU Melle, and Sharon C Glotzer. Analysis of self-assembly pathways with unsupervised machine learning algorithms. *The Journal of Physical Chemistry B*, 124(1):69–78, 2019.

[56] Robin Van Damme, Gabriele M Coli, Rene Van Roij, and Marjolein Dijkstra. Classifying crystals of rounded tetrahedra and determining their order parameters using dimensionality reduction. *ACS nano*, 14(11):15144–15153, 2020.

[57] Gabriele M Coli and Marjolein Dijkstra. An artificial neural network reveals the nucleation mechanism of a binary colloidal ab13 crystal. *ACS nano*, 15(3):4335–4346, 2021.

[58] Erin G Teich, Greg van Anders, and Sharon C Glotzer. Identity crisis in alchemical space drives the entropic colloidal glass transition. *Nature communications*, 10(1):64, 2019.

[59] Sangmin Lee, Erin G Teich, Michael Engel, and Sharon C Glotzer. Entropic colloidal crystallization pathways via fluid–fluid transitions and multidimensional prenucleation motifs. *Proceedings of the National Academy of Sciences*, 116(30):14843–14851, 2019.

[60] John E Carpenter and Michael Grünwald. Pre-nucleation clusters predict crystal structures in models of chiral molecules. *Journal of the American Chemical Society*, 143(51):21580–21593, 2021.

[61] Walter Mickel, Sebastian C Kapfer, Gerd E Schröder-Turk, and Klaus Mecke. Shortcomings of the bond orientational order parameters for the analysis of disordered particulate matter. *The Journal of chemical physics*, 138(4):044501, 2013.

[62] Wolfgang Lechner and Christoph Dellago. Accurate determination of crystal structures based on averaged local bond order parameters. *The Journal of chemical physics*, 129(11):114707, 2008.

[63] Runfang Mao, Jared O'Leary, Ali Mesbah, and Jeetain Mittal. A deep learning framework discovers compositional order and self-assembly pathways in binary colloidal mixtures. *JACS Au*, 2(8):1818–1828, 2022.

[64] Alexander Stukowski. Structure identification methods for atomistic simulations of crystalline materials. *Modelling and Simulation in Materials Science and Engineering*, 20(4):045021, 2012.

[65] Peter Mahler Larsen, Søren Schmidt, and Jakob Schiøtz. Robust structural identification via polyhedral template matching. *Modelling and Simulation in Materials Science and Engineering*, 24(5):055007, 2016.

[66] Andrew W Long and Andrew L Ferguson. Nonlinear machine learning of patchy colloid self-assembly pathways and mechanisms. *The Journal of Physical Chemistry B*, 118(15):4228–4244, 2014.

[67] Wesley F Reinhart, Andrew W Long, Michael P Howard, Andrew L Ferguson, and Athanassios Z Panagiotopoulos. Machine learning for autonomous crystal structure identification. *Soft Matter*, 13(27):4733–4745, 2017.

[68] Andrew L Ferguson. Machine learning and data science in soft materials engineering. *Journal of Physics: Condensed Matter*, 30(4):043002, 2017.

[69] Ryan S DeFever, Colin Targonski, Steven W Hall, Melissa C Smith, and Sapna Sarupria. A generalized deep learning approach for local structure identification in molecular simulations. *Chemical science*, 10(32):7503–7515, 2019.

[70] Wesley F Reinhart. Unsupervised learning of atomic environments from simple features. *Computational Materials Science*, 196:110511, 2021.

[71] QHwan Kim, Joon-Hyuk Ko, Sunghoon Kim, and Wonho Jhe. Gcicenet: a graph convolutional network for accurate classification of water phases. *Physical Chemistry Chemical Physics*, 22(45):26340–26350, 2020.

[72] Angelo Ziletti, Devinder Kumar, Matthias Scheffler, and Luca M Ghiringhelli. Insightful classification of crystal structures using deep learning. *Nature communications*, 9(1):2775, 2018.

[73] Nathaniel Charest, Michael Tro, Michael T Bowers, and Joan-Emma Shea. Latent models of molecular dynamics data: Automatic order parameter generation for peptide fibrillization. *The Journal of Physical Chemistry B*, 124(37):8012–8022, 2020.

[74] Abhik Chakraborty and Raksha Sharma. A deep crystal structure identification system for x-ray diffraction patterns. *The Visual Computer*, 38(4):1275–1282, 2022.

[75] Andrés F Lizano-Villalobos and Xun Tang. Convolutional neural network-based colloidal self-assembly state classification. *Soft Matter*, 2023.

[76] Philipp Geiger and Christoph Dellago. Neural networks for local structure detection in polymorphic systems. *The Journal of chemical physics*, 139(16):164105, 2013.

[77] Ziyue Zou and Pratyush Tiwary. Enhanced sampling of crystal nucleation with graph representation learnt variables. *arXiv preprint arXiv:2310.07927*, 2023.

[78] Florian M Dietrich, Xavier R Advincula, Gianpaolo Gobbo, Michael A Bellucci, and Matteo Salvalaglio. Machine learning nucleation collective variables with graph neural networks. *Journal of Chemical Theory and Computation*, 20(4):1600–1611, 2023.

[79] Suvo Banik, Debdas Dhabal, Henry Chan, Sukriti Manna, Mathew Cherukara, Valeria Molinero, and Subramanian KRS Sankaranarayanan. Cegann: Crystal edge graph attention neural network for multiscale classification of materials environment. *npj Computational Materials*, 9(1):23, 2023.

[80] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[81] Aaron S Keys, Christopher R Iacovella, and Sharon C Glotzer. Characterizing complex particle morphologies through shape matching: Descriptors, applications, and algorithms. *Journal of Computational Physics*, 230(17):6438–6463, 2011.

[82] Amir Haji-Akbari and Sharon C Glotzer. Strong orientational coordinates and orientational order parameters for symmetric objects. *Journal of Physics A: Mathematical and Theoretical*, 48(48):485201, 2015.

[83] Vyas Ramasubramani and Sharon C Glotzer. rowan: A python package for working with quaternions. *Journal of Open Source Software*, 3(27):787, 2018.

[84] Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International Conference on Machine Learning*, pages 3318–3328. PMLR, 2021.

[85] Dongdong Chen, Julián Tachella, and Mike E Davies. Robust equivariant imaging: a fully unsupervised framework for learning to image from noisy and partial measurements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5647–5656, 2022.

[86] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

[87] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[88] Joshua A Anderson, Jens Glaser, and Sharon C Glotzer. Hoomd-blue: A python package for high-performance molecular dynamics and hard particle monte carlo simulations. *Computational Materials Science*, 173:109363, 2020.

[89] Vyas Ramasubramani, Thi Vo, Joshua A Anderson, and Sharon C Glotzer. A mean-field approach to simulating anisotropic particles. *The Journal of Chemical Physics*, 153(8):084106, 2020.

[90] Vyas Ramasubramani, Bradley D Dice, Eric S Harper, Matthew P Spellings, Joshua A Anderson, and Sharon C Glotzer. freud: A software suite for high throughput analysis of particle simulation data. *Computer Physics Communications*, 254:107275, 2020.

[91] Alexander Stukowski. Visualization and analysis of atomistic simulation data with ovito–the open visualization tool. *Modelling and simulation in materials science and engineering*, 18(1):015012, 2009.

[92] Qian Chen, Sung Chul Bae, and Steve Granick. Directed self-assembly of a colloidal kagome lattice. *Nature*, 469(7330):381–384, 2011.

[93] Norbert Kern and Daan Frenkel. Fluid–fluid coexistence in colloidal systems with short-ranged strongly directional attraction. *The Journal of chemical physics*, 118(21):9882–9889, 2003.

[94] András Sápi, Thangamani Rajkumar, János Kiss, Ákos Kukovecz, Zoltán Kónya, and Gabor A Somorjai. Metallic nanoparticles in heterogeneous catalysis. *Catalysis letters*, 151:2153–2175, 2021.

[95] Michael J Mitchell, Margaret M Billingsley, Rebecca M Haley, Marissa E Wechsler, Nicholas A Peppas, and Robert Langer. Engineering precision nanoparticles for drug delivery. *Nature reviews drug discovery*, 20(2):101–124, 2021.

[96] K Lance Kelly, Eduardo Coronado, Lin Lin Zhao, and George C Schatz. The optical properties of metal nanoparticles: the influence of size, shape, and dielectric environment, 2003.

[97] Maksym V Kovalenko, Liberato Manna, Andreu Cabot, Zeger Hens, Dmitri V Talapin, Cherie R Kagan, Victor I Klimov, Andrey L Rogach, Peter Reiss, Delia J Milliron, et al. Prospects of nanoscience with nanocrystals. *ACS nano*, 9(2):1012–1057, 2015.

[98] Somnath Koley, Jiabin Cui, Yossef E Panfil, and Uri Banin. Coupled colloidal quantum dot molecules. *Accounts of Chemical Research*, 54(5):1178–1188, 2021.

[99] Kuldeep Mahato, Sahil Nagpal, Mahero Ayesha Shah, Ananya Srivastava, Pawan Kumar Maurya, Shounak Roy, Amit Jaiswal, Renu Singh, and Pranjal Chandra. Gold nanoparticle surface engineering strategies and their applications in biomedicine and diagnostics. *3 Biotech*, 9:1–19, 2019.

[100] Ahyoung Kim, Shan Zhou, Lehan Yao, Stacey Ni, Binbin Luo, Charles E Sing, and Qian Chen. Tip-patched nanoprisms from formation of ligand islands. *Journal of the American Chemical Society*, 141(30):11796–11800, 2019.

[101] Christine Fiedler, Tobias Kleinhanns, Maria Garcia, Seungho Lee, Mariano Calcabrini, and Maria Ibáñez. Solution-processed inorganic thermoelectric materials: Opportunities and challenges. *Chemistry of Materials*, 34(19):8471–8489, 2022.

[102] Pan Gu, Jiangping Xu, and Jintao Zhu. Self-assembly of polymer-grafted inorganic nanoparticles into three-dimensional superlattices. *Giant*, 12:100123, 2022.

[103] Bhausaheb V Tawade, Maninderjeet Singh, Ikeoluwa E Apata, Jagadesh Veerasamy, Nihar Pradhan, Alamgir Karim, Jack F Douglas, and Dharmaraj Raghavan. Polymer-grafted nanoparticles with variable grafting densities for high energy density polymeric nanocomposite dielectric capacitors. *JACS Au*, 3(5):1365–1375, 2023.

[104] Zongyu Wang, Hao Chen, Yangyang Wang, Jihua Chen, Mark A Arnould, Bin Hu, Ilja Popovs, Shannon M Mahurin, and Sheng Dai. Polymer-grafted porous silica nanoparticles with enhanced co2 permeability and mechanical performance. *ACS Applied Materials & Interfaces*, 13(23):27411–27418, 2021.

[105] MJ Murray and JV Sanders. Close-packed structures of spheres of two different sizes ii. the packing densities of likely arrangements. *Philosophical Magazine A*, 42(6):721–740, 1980.

[106] Elena V Shevchenko, Dmitri V Talapin, Nicholas A Kotov, Stephen O'Brien, and Christopher B Murray. Structural diversity in binary nanoparticle superlattices. *Nature*, 439(7072):55–59, 2006.

[107] Xingchen Ye, Jun Chen, and Christopher B Murray. Polymorphism in self-assembled ab6 binary nanocrystal superlattices. *Journal of the American Chemical Society*, 133(8):2613–2620, 2011.

[108] Alex Travesset. Phase diagram of power law and lennard-jones systems: Crystal phases. *The Journal of chemical physics*, 141(16), 2014.

[109] Alex Travesset. Binary nanoparticle superlattices of soft-particle systems. *Proceedings of the National Academy of Sciences*, 112(31):9563–9567, 2015.

[110] R Allen LaCour, Carl Simon Adorf, Julia Dshemuchadse, and Sharon C Glotzer. Influence of softness on the stability of binary colloidal crystals. *ACS nano*, 13(12):13829–13842, 2019.

[111] Uzi Landman and WD Luedtke. Small is different: energetic, structural, thermal, and mechanical properties of passivated nanocluster assemblies. *Faraday Discussions*, 125:1–22, 2004.

[112] Philipp Schapotschnikow and Thijs JH Vlugt. Understanding interactions between capped nanocrystals: Three-body and chain packing effects. *The Journal of chemical physics*, 131(12), 2009.

[113] A Travesset. Topological structure prediction in binary nanoparticle superlattices. *Soft Matter*, 13(1):147–157, 2017.

[114] Leandro L Missoni and Mario Tagliazucchi. The phase behavior of nanoparticle superlattices in the presence of a solvent. *ACS nano*, 14(5):5649–5658, 2020.

[115] Andrew W Long and Andrew L Ferguson. Rational design of patchy colloids via landscape engineering. *Molecular Systems Design & Engineering*, 3(1):49–65, 2018.

[116] Gabriele M Coli, Emanuele Boattini, Laura Filion, and Marjolein Dijkstra. Inverse design of soft materials via a deep learning–based evolutionary strategy. *Science advances*, 8(3):eabj6731, 2022.

[117] RB Jadrich, BA Lindquist, and TM Truskett. Probabilistic inverse design for self-assembling materials. *The Journal of Chemical Physics*, 146(18), 2017.

[118] Zachary M Sherman, Michael P Howard, Beth A Lindquist, Ryan B Jadrich, and Thomas M Truskett. Inverse methods for design of soft materials. *The Journal of chemical physics*, 152(14), 2020.

[119] Chrisy Xiyu Du, Greg van Anders, Richmond S Newman, and Sharon C Glotzer. Shape-driven solid–solid transitions in colloids. *Proceedings of the National Academy of Sciences*, 114(20):E3892–E3899, 2017.

[120] Carl S Adorf, James Antonaglia, Julia Dshemuchadse, and Sharon C Glotzer. Inverse design of simple pair potentials for the self-assembly of complex structures. *The Journal of chemical physics*, 149(20), 2018.

[121] John WR Morgan and Sharon C Glotzer. The alchemical energy landscape for a pentameric cluster. *The Journal of Chemical Physics*, 152(1), 2020.

[122] Emily R Liepold, Alex Smith, Binhua Lin, Juan De Pablo, and Stuart A Rice. Pair and many-body interactions between ligated au nanoparticles. *The Journal of chemical physics*, 150(4), 2019.

[123] Tommy Waltmann, Curt Waltmann, Nathan Horst, and Alex Travesset. Many body effects and icosahedral order in superlattice self-assembly. *Journal of the American Chemical Society*, 140(26):8236–8245, 2018.

[124] Tsung-Yeh Tang and Gaurav Arya. Anisotropic three-particle interactions between spherical polymer-grafted nanoparticles in a polymer matrix. *Macromolecules*, 50(3):1167–1183, 2017.

[125] Giuliana Giunta, Gerardo Campos-Villalobos, and Marjolein Dijkstra. Coarse-grained many-body potentials of ligand-stabilized nanoparticles from machine-learned mean forces. *ACS nano*, 17(23):23391–23404, 2023.

[126] Yilong Zhou, Sigbjørn Løland Bore, Andrea R Tao, Francesco Paesani, and Gaurav Arya. Many-body potential for simulating the self-assembly of polymer-grafted nanoparticles in a polymer matrix. *npj Computational Materials*, 9(1):224, 2023.

[127] Rebecca K Lindsey, Sorin Bastea, Sebastien Hamel, Yanjun Lyu, Nir Goldman, and Vincenzo Lordi. Chimes carbon 2.0: A transferable machine-learned interatomic model harnessing multifidelity training data. *npj Computational Materials*, 11(1):26, 2025.

[128] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[129] Volker L Deringer, Albert P Bartók, Noam Bernstein, David M Wilkins, Michele Ceriotti, and Gábor Csányi. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16):10073–10141, 2021.

[130] Marissa G Saunders and Gregory A Voth. Coarse-graining methods for computational biology. *Annual review of biophysics*, 42(1):73–93, 2013.

[131] Jaehyeok Jin, Alexander J Pak, Aleksander EP Durumeric, Timothy D Loose, and Gregory A Voth. Bottom-up coarse-graining: Principles and perspectives. *Journal of chemical theory and computation*, 18(10):5759–5791, 2022.

[132] Paulo CT Souza, Riccardo Alessandri, Jonathan Barnoud, Sebastian Thallmair, Ignacio Faustino, Fabian Grünewald, Ilias Patmanidis, Haleh Abdizadeh, Bart MH Bruininks, Tsjerk A Wassenaar, et al. Martini 3: a general purpose force field for coarse-grained molecular dynamics. *Nature methods*, 18(4):382–388, 2021.

[133] Johannes Kästner. Umbrella sampling. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(6):932–942, 2011.

[134] Alan Grossfield. Wham: the weighted histogram analysis method, 2012.

[135] Pouria Nourian, Jimmy Lawrence, and Andrew J Peters. Interparticle interactions of dendrimer, comb, and linear grafted nanoparticles via coarse-grained molecular dynamics simulations. *Macromolecules*, 2024.

[136] Sanghyun Park, Fatemeh Khalili-Araghi, Emad Tajkhorshid, and Klaus Schulten. Free energy calculation from steered molecular dynamics simulations using jarzynski's equality. *The Journal of chemical physics*, 119(6):3559–3566, 2003.

[137] Sanghyun Park and Klaus Schulten. Calculating potentials of mean force from steered molecular dynamics simulations. *The Journal of chemical physics*, 120(13):5946–5961, 2004.

[138] Hui Lu and Klaus Schulten. Steered molecular dynamics simulations of force-induced protein domain unfolding. *Proteins: Structure, Function, and Bioinformatics*, 35(4):453–463, 1999.

[139] Hsieh Chen, Shannon L Eichmann, and Nancy A Burnham. Understanding calcium-mediated adhesion of nanomaterials in reservoir fluids by insights from molecular dynamics simulations. *Scientific Reports*, 9(1):10763, 2019.

[140] Tatiane Aparecida Rocha Gonçalves, Lucas Andrade Silva, Alexandre Moni Pereira, Antonio Eduardo Clark Peres, and Julio Cesar Guedes Correia. Advancing iron ore slimes magnetic separation with functionalized nanoparticles: Molecular insights from free energy simulations. *Applied Surface Science*, 642:158498, 2024.

[141] Rebecca K Lindsey, Laurence E Fried, and Nir Goldman. Chimes: A force matched potential with explicit three-body interactions for molten carbon. *Journal of chemical theory and computation*, 13(12):6222–6229, 2017.

[142] Rebecca K Lindsey, Cong Huy Pham, Nir Goldman, Sorin Bastea, and Laurence E Fried. Machine-learning a solution for reactive atomistic simulations of energetic materials. *Propellants, Explosives, Pyrotechnics*, 47(8):e202200001, 2022.

[143] Rebecca K Lindsey, Sorin Bastea, Yanjun Lyu, Sebastien Hamel, Nir Goldman, and Laurence E Fried. Chemical evolution in nitrogen shocked beyond the molecular stability limit. *The Journal of Chemical Physics*, 159(8), 2023.

[144] Nir Goldman, Laurence E Fried, Rebecca K Lindsey, C Huy Pham, and Riccardo Dettori. Enhancing the accuracy of density functional tight binding models through chimes many-body interaction potentials. *The Journal of Chemical Physics*, 158(14), 2023.

[145] Cong Huy Pham, Rebecca K Lindsey, Laurence E Fried, and Nir Goldman. Calculation of the detonation state of hn3 with quantum accuracy. *The Journal of Chemical Physics*, 153(22), 2020.

[146] JJPRB Tersoff. Modeling solid-state chemistry: Interatomic potentials for multicomponent systems. *Physical review B*, 39(8):5566, 1989.

[147] Rebecca K Lindsey, Nir Goldman, Laurence E Fried, and Sorin Bastea. Many-body reactive force field development for carbon condensation in c/o systems under extreme conditions. *The Journal of Chemical Physics*, 153(5), 2020.

[148] Yimin Wang, Benjamin C Shepler, Bastiaan J Braams, and Joel M Bowman. Full-dimensional, ab initio potential energy and dipole moment surfaces for water. *The Journal of chemical physics*, 131(5), 2009.

[149] Yimin Wang, Xinchuan Huang, Benjamin C Shepler, Bastiaan J Braams, and Joel M Bowman. Flexible, ab initio potential, and dipole moment surfaces for water. i. tests and applications for clusters up to the 22-mer. *The Journal of chemical physics*, 134(9), 2011.

[150] Bastiaan J Braams and Joel M Bowman. Permutationally invariant potential energy surfaces in high dimensionality. *International Reviews in Physical Chemistry*, 28(4):577–606, 2009.

[151] Lindsey R. Laubach B. Cluster-graph fingerprinting: A framework for quantitative analysis of machine-learned interatomic model training and simulation data. *ChemRxiv.*, 2025.

[152] Glenn J Martyna, Douglas J Tobias, and Michael L Klein. Constant pressure molecular dynamics algorithms. *J. chem. Phys*, 101(4177):10–1063, 1994.

[153] Giovanni Bussi, Davide Donadio, and Michele Parrinello. Canonical sampling through velocity rescaling. *The Journal of chemical physics*, 126(1), 2007.

[154] Scott E Feller, Yuhong Zhang, Richard W Pastor, and Bernard R Brooks. Constant pressure molecular dynamics simulation: The langevin piston method. *The Journal of chemical physics*, 103(11):4613–4621, 1995.

[155] Rebecca K. Lindsey, Nir Goldman, and Laurence E Fried. Chimes-lsq. `https://github.com/rk-lindsey/chimes_lsq.git`.

[156] Rebecca K. Lindsey, Nir Goldman, and Laurence E Fried. Chimes-calculator. `https://github.com/rk-lindsey/chimes_calculator.git`.

[157] Vyas Ramasubramani, Bradley D. Dice, Eric S. Harper, Matthew P. Spellings, Joshua A. Anderson, and Sharon C. Glotzer. freud: A software suite for high throughput analysis of particle simulation data. *Computer Physics Communications*, 254:107275, 2020.

[158] Carl S Adorf, Paul M Dodd, Vyas Ramasubramani, and Sharon C Glotzer. Simple data and workflow management with the signac framework. *Computational Materials Science*, 146:220–229, 2018.

[159] Vyas Ramasubramani, Carl Adorf, Paul Dodd, Bradley Dice, and Sharon Glotzer. signac: A python framework for data and workflow management. In *Proceedings of the Python in Science Conference*, 2018.

[160] Joshua A. Anderson, Jen Bradley, Joseph Burkhart, Kate Jensen, Tim Moore, and Corwin Kerr. Row. `https://github.com/glotzerlab/row.git`, 2024.

[161] O Mazonka and C Jarzynski. Exactly solvable model illustrating far-from-equilibrium predictions. *arXiv preprint cond-mat/9912121*, 1999.

[162] Gavin E Crooks. Path-ensemble averages in systems driven far from equilibrium. *Physical review E*, 61(3):2361, 2000.

[163] Jeff Gore, Felix Ritort, and Carlos Bustamante. Bias and error in estimates of equilibrium free-energy differences from nonequilibrium measurements. *Proceedings of the National Academy of Sciences*, 100(22):12564–12569, 2003.

[164] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[165] `http://www.qhull.org/`.

[166] Peter Alfeld. A trivariate clough—tocher scheme for tetrahedral data. *Computer Aided Geometric Design*, 1(2):169–181, 1984.

[167] Sharon C. Glotzer Melody Y. Zhang, Shih-Kuang (Alex) Lee and Rebecca K. Lindsey. Hoomd chimes. `https://github.com/glotzerlab/xchimes`.

[168] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comp. Phys. Comm.*, 271:108171, 2022.

[169] Christina AR Picken, Orla Buensoz, Paul D Price, Christopher Fidge, Laurie Points, and Michael P Shaver. Sustainable formulation polymers for home, beauty and personal care: challenges and opportunities. *Chemical Science*, 14(45):12926–12940, 2023.

[170] Gustavo S Luengo, Fabien Leonforte, Andrew Greaves, Ramon G Rubio, and Eduardo Guzman. Physico-chemical challenges on the self-assembly of natural and bio-based ingredients on hair surfaces: towards sustainable haircare formulations. *Green Chemistry*, 25(20):7863–7882, 2023.

[171] Yina Geng, Greg van Anders, and Sharon C Glotzer. Synthesizable nanoparticle eigenshapes for colloidal crystals. *Nanoscale*, 13(31):13301–13309, 2021.

[172] Wei-Tse Hsu, Valerio Piomponi, Pascal T Merz, Giovanni Bussi, and Michael R Shirts. Alchemical metadynamics: Adding alchemical variables to metadynamics to enhance sampling in free energy calculations. *Journal of Chemical Theory and Computation*, 19(6):1805–1817, 2023.

[173] Michael Engel and Hans-Rainer Trebin. Self-assembly of monatomic complex crystals and quasicrystals with a double-well interaction potential. *Physical Review Letters*, 98:225505, Jun 2007.

[174] Julia Dshemuchadse, Pablo F Damasceno, Carolyn L Phillips, Michael Engel, and Sharon C Glotzer. Moving beyond the constraints of chemistry via crystal structure discovery with isotropic multiwell pair potentials. *Proceedings of the National Academy of Sciences*, 118(21):e2024034118, 2021.

[175] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[176] Alessandro Barducci, Giovanni Bussi, and Michele Parrinello. Well-tempered metadynamics: a smoothly converging and tunable free-energy method. *Physical Review Letters*, 100(2):020603, 2008.

[177] Omar Valsson, Pratyush Tiwary, and Michele Parrinello. Enhancing important fluctuations: Rare events and metadynamics from a conceptual viewpoint. *Annual Review of Physical Chemistry*, 67(1):159–184, 2016.

[178] Giovanni Bussi and Alessandro Laio. Using metadynamics to explore complex free-energy landscapes. *Nature Reviews Physics*, 2(4):200–212, 2020.

[179] Domagoj Fijan, Maria R Ward Rashidi, and Sharon C Glotzer. Quantifying local point-group-symmetry order in complex particle systems. *arXiv preprint arXiv:2509.12665*, 2025.

[180] Charlotte Shiqi Zhao, Sun-Ting Tsai, and Sharon C Glotzer. Hybrid monte carlo metadynamics (hybridmc-metad). *The Journal of Chemical Physics*, 164(2), 2026.

[181] Mikael C Rechtsman, Frank H Stillinger, and Salvatore Torquato. Optimized interactions for targeted self-assembly: application to a honeycomb lattice. *Physical Review Letters*, 95(22):228301, 2005.

[182] Marek Mihalkovič and CL Henley. Empirical oscillating potentials for alloys from ab initio fits and the prediction of quasicrystal-related structures in the al-cu-sc system. *Physical Review B—Condensed Matter and Materials Physics*, 85(9):092102, 2012.

[183] Måns Elenius, Fredrik HM Zetterling, Mikhail Dzugutov, Mikhail Dzugutov, Daniel C Fredrickson, and Sven Lidin. Structural model for octagonal quasicrystals derived from octagonal symmetry elements arising in β-mn crystallization of a simple monatomic liquid. *Physical Review B—Condensed Matter and Materials Physics*, 79(14):144201, 2009.

[184] Stefan Auer and Daan Frenkel. Prediction of absolute crystal-nucleation rate in hard-sphere colloids. *Nature*, 409(6823):1020–1023, 2001.

[185] Stefan Auer and Daan Frenkel. Crystallization of weakly charged colloidal spheres: a numerical study. *Journal of Physics: Condensed Matter*, 14(33):7667, 2002.

[186] Stefan Auer and Daan Frenkel. Numerical prediction of absolute crystallization rates in hard-sphere colloids. *The Journal of chemical physics*, 120(6):3015–3029, 2004.

[187] Pieter Rein ten Wolde and Daan Frenkel. Computer simulation study of gas–liquid nucleation in a lennard-jones system. *The Journal of chemical physics*, 109(22):9901–9918, 1998.

[188] Matthias Mecke, Jochen Winkelmann, and Johann Fischer. Molecular dynamics simulation of the liquid–vapor interface: The lennard-jones fluid. *The Journal of chemical physics*, 107(21):9264–9270, 1997.

[189] Joshua A Anderson, Eric Jankowski, Thomas L Grubb, Michael Engel, and Sharon C Glotzer. Massively parallel monte carlo for many-particle simulations on gpus. *Journal of Computational Physics*, 254:27–38, 2013.

[190] Jens Glaser, Trung Dac Nguyen, Joshua A Anderson, Pak Lui, Filippo Spiga, Jaime A Millan, David C Morse, and Sharon C Glotzer. Strong scaling of general-purpose molecular dynamics simulations on gpus. *Computer Physics Communications*, 192:97–107, 2015.

[191] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C Smith, Berk Hess, and Erik Lindahl. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25, 2015.

[192] William Bugden and Ayman Alahmar. The safety and performance of prominent programming languages. *International Journal of Software Engineering and Knowledge Engineering*, 32(05):713–744, 2022.

[193] William Bugden and Ayman Alahmar. Rust: The programming language for safety and performance. *arXiv preprint arXiv:2206.05503*, 2022.

[194] Mark E Tuckerman, José Alejandre, Roberto López-Rendón, Andrea L Jochim, and Glenn J Martyna. A liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal–isobaric ensemble. *Journal of Physics A: Mathematical and General*, 39(19):5629, 2006.

[195] Ruslan L Davidchack. Discretization errors in molecular dynamics simulations with deterministic and stochastic thermostats. *Journal of Computational Physics*, 229(24):9323–9346, 2010.

[196] Rahul Nag, Nina Rouvière, Jaime Gabriel Trazo, Jules Marcone, Nika Kutalia, Claire Goldmann, Marianne Impéror-Clerc, Damien Alloyeau, Doru Constantin, and Cyrille Hamon. Depletion-induced tunable assembly of complementary platonic solids. *Nano Letters*, 24(51):16368–16373, 2024.

[197] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[198] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[199] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

[200] Yan Zhu, Ya-Dong Wu, Ge Bai, Dong-Sheng Wang, Yuexuan Wang, and Giulio Chiribella. Flexible learning of quantum states with generative query neural networks. *Nature communications*, 13(1):6222, 2022.

[201] Tian Xie, Xiang Fu, Octavian-Eugen Ganea, Regina Barzilay, and Tommi Jaakkola. Crystal diffusion variational autoencoder for periodic material generation. *arXiv preprint arXiv:2110.06197*, 2021.

[202] Gerardo Campos-Villalobos, Rodolfo Subert, Giuliana Giunta, and Marjolein Dijkstra. Machine-learned coarse-grained potentials for particles with anisotropic shapes and interactions. *npj Computational Materials*, 10(1):228, 2024.

[203] Mohammadreza Fakhraei, Chris A Kieslich, and Michael P Howard. Approximation of anisotropic pair potentials using multivariate interpolation. *The Journal of Physical Chemistry B*, 129(27):6985–6996, 2025.